Proceedings of the 2nd International Conference on Functional Materials and Civil Engineering
DOI: 10.54254/2755-2721/70/20241029

Convolutional neural network model compression method

Likai Chu

Sichuan University-Pittsburgh Institute, Chengdu, China

2022141520261@stu.scu.edu.cn

Abstract. In the field of deep learning, Convolutional Neural Networks (CNN) has become a focal point due to its multi-layered structure and wide application. The success of deep learning is due to the model has more layers and more parameters, which gives it a stronger nonlinear fitting ability. Traditionally, CNNs are primarily run on Central Processing Units (CPUs) and Graphics Processing Units (GPUs). However, CPUs have lower computational power, and GPUs consume a lot of energy. In contrast, Field-Programmable Gate Arrays (FPGAs) offer high parallelism, low power consumption, flexible programming, and rapid development cycles. These combined advantages make FPGAs more suitable for the forward inference processes of deep learning compared to other platforms. However, CNNs has the characteristics of parameter redundancy, the storage cost of deploying it on FPGA is too high. In order to apply CNNs to FPGA, we need to optimize CNNs for compression. Because after compression, This paper analyzes several specific cases of model compression for convolutional neural networks, and summarizes and compares the efficient methods of model compression. The results show that the model compression methods are mainly divided into the structure change of convolutional neural network and the quantization of parameters. The two compression methods can be cascaded at the same time to achieve a better optimization effect.

Keywords: Convolutional Neural Networks, model compression, finer network model, network pruning, model quantization

1. Introduction

In the field of deep learning, Convolutional Neural Networks (CNN) has become the focus of research due to its multi-level structure and wide application. Traditionally, CNNs are primarily run on Central Processing Units (CPUs) and Graphics Processing Units (GPUs). However, CPUs have lower computational power, and GPUs consume a lot of energy. In contrast, Field-Programmable Gate Arrays (FPGAs) offer high parallelism, low power consumption, flexible programming, and rapid development cycles. These combined advantages make FPGAs more suitable for the forward inference processes of deep learning compared to other platforms. Nevertheless, deploying CNNs on FPGAs presents challenges such as high storage requirements, external memory bandwidth limitations, and significant computational demands, especially as future models become more complex. Consequently, how to compress and optimize CNNs without sacrificing model performance becomes a crucial preprocessing step for their deployment on FPGAs [1]. Gong et.al proposed that deep neural networks generally contain parameter redundancy. Therefore, model compression techniques should be employed to streamline network models, thereby reducing the parameter and computational complexity of the network [2]. Currently, major model compression methods can be roughly categorized into the following

types: finer network model design, network pruning, model quantization, and network decomposition [3]. Finer network model design involves reducing network parameters and computational complexity through careful structuring of the network architecture. Network pruning entails removing unimportant parameters from the model, retaining only the essential ones. Model quantization involves converting high-precision floating-point parameters into low-bit fixed-point data to reduce model storage space and computational complexity. This paper focuses on the specific cases of these methods, analyzes, summarizes and summarizes the optimization methods of each case, and then compares them to find out the similarities and differences between different compression methods, and finally summarizes to obtain an efficient compression optimization method for Convolutional Neural Networks.
Table 1 shows the compression methods studied in this paper.

Table 1. Compression method generalization

Compression method	Compressed description	advantages
Finer network model design	reducing network parameters and computational complexity through careful structuring of the network architecture	Optimize the network structure and simplify the calculation
Network pruning	removing unimportant parameters from the model, retaining only the essential ones	Greatly reduce the size of parameters while ensuring accuracy
Model quantization	converting high-precision floating-point parameters into low-bit fixed-point data to reduce model storage space and computational complexity	The model volume can be greatly reduced with little loss of accuracy

This paper is mainly divided into three chapters. The first chapter is the introduction of the paper, which only introduces the background of convolutional neural network compression optimization, research status and the structure of the paper. In chapter 2, we introduce the relative methods of convolutional neural network modulus compression and the specific cases of each method. The third chapter of the article makes a specific analysis, comparison and summary of each case, and draws a conclusion. The last chapter is conclusion.

2. Model compression

2.1. Classification of model compression

Chen Siang in his paper said the design, compression and acceleration of convolutional neural networks are very challenging. First of all, in order to meet the performance requirements of complex intelligent applications, neural networks often contain a huge number of more than one million parameters, and it is not feasible to find redundant parameters or build efficient network structures through greedy generation selection. Therefore, how to effectively judge whether parameters or structures are important is the key to network design and compression [4]. For this problem, Guo Qingbei presented a series of methods in his paper. These include network pruning, network quantization, low-rank decomposition, knowledge distillation, lightweight neural framework design, and so on. And neural framework search [5]. In this paper, the research on compression and acceleration technology of deep neural networks is mainly focused on finer network model design, network pruning and model quantization.

2.2. Analysis on the problems

(1) Finer network model design

Finer network model design involves reducing network parameters and computational complexity through careful structuring of the network architecture. Chen Siang mentioned that the design of an efficient convolutional network architecture can be seen as a trade-off between how to set these parameters to achieve better performance, the number of parameters, and the amount of computation [4].

Mei Qichang adopted a computational approach based on vector inner products and split convolution, achieving efficient computation while reducing data caching space [6]. Chen et.al proposed an algorithm called k-means to realize the lightweight of convolutional neural networks [7]. Azam Ghanbari and Mehdi Modarressi present a new computation-reuse aware accelerator for Convolutional Neural Networks called CORN-C, this approach minimizes redundant calculations by utilizing the result of one arithmetic operation for multiple subsequent and concurrent operations with repetitive inputs, thereby enhancing the energy efficiency of neural networks [8].

(2) Network pruning

Wang Jong said network pruning refers to removing redundant parameters from a pre-trained neural network model based on certain criteria for determining the importance of parameters and ensuring that the performance of the network model is not seriously affected [9]. Wang proposed a global adaptive pruning algorithm based on dual DDPG, using two DDPG agents to calculate the global scale coefficient and global deviation coefficient respectively. The experimental results show that the dual DDPG algorithm is not only efficient, but also more accurate and stable [9]. Sun Jianhui designed an adaptive pruning strategy to score channels from both global information and local information, and then the channels with low scores are eliminated [10]. Zhao proposed a Pseudo Pruning method that the traditional alternate pruning and training methods are abandoned, and the two methods are decouple, that is, network pruning and network training are carried out at the same time, and the compression effect of the network is improved by optimizing the precision difference before and after pruning [11]. Sun Put forward a combination of the whole Information and local information channel importance scoring method, and design a score based on the results Pruning strategies that automatically determine the proportion of each layer cut [10]. Zheng et.al proposed a novel weight-based filter pruning criterion. This criterion considers both direct and indirect influences, evaluating their importance for pruning. It models the impacts of these two influences based on weights and introduces a penalty term to constrain uncertainty, thereby enhancing the performance of the pruning model [12].

(3) Model quantization

Quantization is a numerical representation that uses a low-precision way to store and calculate. It was initially used to compress text, images and other data, and now it is applied in the field of neural networks to discretize and sparse the weight parameters of networks, so as to achieve the purpose of network compression and reduce the amount of computation. Generally speaking, the full-precision floating-point number, that is, FP32, is used as a general data operation and storage method for various parameters in the construction and training of neural networks [9].

Sun used the hybrid fine quantization scheme allows each layer to use different quantization bits. A gradual quantization strategy is also proposed, which can select the most suitable precision for each layer parameter in a short time [10]. Luo et. al designed a quantization neural network for real-time classification of ECG data. The weight quantization force is two-digit integer and the performance is achieved by knowledge distillation method [13]. By establishing a joint compression framework, Wang et.al proposed a deep network compression method based on low-rank decomposition and vector quantization, which can achieve further quantization on the basis of the low-rank structure of the network, thus achieving a greater compression ratio [14].

3. More specific analysis

3.1. Detailed case analysis

(1) Finer network model design

This paper uses Mei's method as a discussion. This method conducts convolution calculations of different kernel sizes through two steps: channel-wise multiplication and addition, and pixel-wise accumulation. Unlike the storage logic of 2D convolution, the storage method based on vector inner products does not require matrix transformation, thereby avoiding additional memory overhead. Data is stored in BRAM (Block RAM), and simple logic control enables convolution calculations with different kernel sizes and different strides [6].

Mei Qi chang also designed efficient and flexible convolution computation units based on vector inner products and split convolution. These units are suitable for the unified computation of convolution kernels of different sizes and consist of multiplication-addition arrays and output feature spectrum accumulation units. In terms of spatial exploration, the maximum parallelism of the convolution computation units is explored. For the channel-wise multiplication-addition array, starting from the FPGA's underlying hard core DSP (Digital Signal Processor), a quantization method based on 8 bits was designed, and DSP double multiplication operations were implemented. Additionally, cascading 16 DSPs was achieved, saving LUT (Look-Up Table) resources for the addition trees required for channel-wise addition [6].

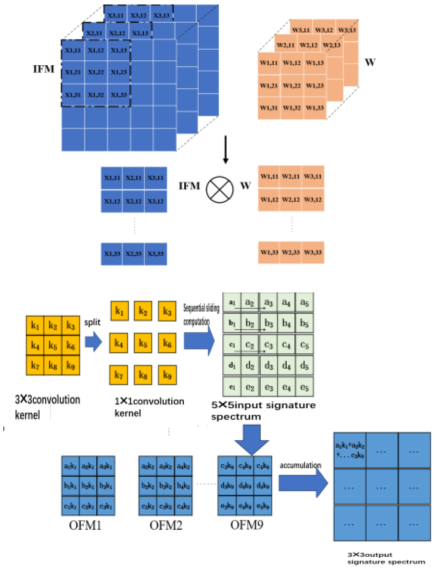
IFM
W
IFM
W
split
Sequential sliding computation
3X3convolution kernel
1X1convolution kernel
5X5input signature spectrum
accumulation
OFM1
OFM2
OFM9
3X3output signature spectrum

Figure 1. Vector inner products and split convolution

(2) Network pruning

This part uses Wang's example. His pruning algorithm mainly uses two

DDPGS to calculate the global size coefficient and the global deviation coefficient respectively. After obtaining these two coefficients, the local importance of a channel in a convolution layer can be transformed into the global importance. Then select the corresponding threshold value and remove the channel whose importance is less than the threshold value to complete the pruning. The formula is as follows.

$$\alpha = a_t^1 = DDPG^1(s_t^1) \in R^L$$
$$\alpha = a_t^2 = DDPG^2(s_t^2) \in R^L$$

at 1and at 2 as the global scale coefficient and global deviation coefficient of the network model [9].

(3) Model quantization

This part adopts sun's method. Sun adopted a two-stage hybrid precision quantization algorithm. First, the input feature weights are quantified to mixed precision, then the performance of the convolutional neural network is fine-tuned, and then the feature map is quantified. Sun also proposed a cascaded scheme involving quantization followed by pruning. Initially, Sun trained CNNs undergo quantization to obtain weight parameters with mixed precision, and all generated feature maps are quantized during forward inference. Then, the quantized CNNs from the previous step are scored, using quantized values for both weights and feature maps. Subsequently, channels with low scores are pruned, and parameters corresponding to pruned channels are completely removed. The result is a pruned CNN with weights represented in mixed precision [10].

3.2. Suggestions

Following the cascaded approach proposed by Sun Jianhui from Shanghai Jiao Tong University, the aforementioned three model compression methods can be further cascaded. For example, although Mei Qichang from Nanjing University of Posts and Telecommunications did not specifically design a quantization method, the input features required by his designed network model computation method are also quantized after model quantization. Therefore, they can be cascaded with the model quantization method. Additionally, both finer network model design and network pruning, regardless of their nature, involve changes to the network structure, thus they can also find common ground.

4. Conclusion

The results show that the model compression methods are mainly divided into the structure change of convolutional neural network and the quantization of parameters. The two compression methods can be cascaded at the same time to achieve a better optimization effect. Finer network model design is to learn from the existing deep neural network structure, the classical network module is used to re-construct the neural network, not to simplify the existing network model. Because of its simple model, low storage space occupation, simplified calculation and other properties, it is classified into the model compression research category. Lightweight network design is also for a specific task, specially designed for mobile device network model, although the design of this network makes deep learning landing, can be widely used in smart devices, but it has a single task, poor generalization of the shortcomings, also makes deep learning this technology is overqualified. Network pruning is one of the most effective methods used in the research of deep neural network model compression. However, different clipping methods are aimed at the classification of specific tasks and cannot be applied to multi-objective tasks. Due to the plasticity of neural network itself, almost all clipping methods can achieve the effect of network compression while ensuring certain accuracy. Traditional clipping methods need to iterate repeatedly on multiple threshold values to be measured, set sensitivity manually, and fine-tune parameters, which is not only time-consuming but also computation-intensive. At the same time, it is difficult to find a suitable threshold value because the weight country value is shared among all layers of the network. Parameter quantization algorithm, no matter it is binary quantization, ternary quantization or multi-value quantization, its essence is to map multiple weights to one value, achieve weight sharing, thus reducing storage and operation overhead.

Parametric quantization is a mainstream model compression technique, which can greatly reduce model volume with little loss of accuracy. On the one hand, it is difficult to realize quantization and the accuracy is unstable. After quantization, it is difficult to make other changes. On the other hand, the universality is poor, often requires specific hardware support, a quantitative method needs to develop a set of special runtime, increasing the difficulty of implementation and maintenance costs.

References

[1] Seojin J ,Wei L ,Yongbeom C . Convolutional Neural Network Model Compression Method for Software—Hardware Co-Design [J]. Information, 2022, 13 (10): 451-451. (10): 451-451.

[2] Zhang S ,Gao Y . Hybrid multi-objective evolutionary model compression with convolutional neural networks [J]. Results in Engineering, 2024, 21 101751-.

[3] Xie, J., Lin, S., Zhang, Y., & Luo, L. (2023). Compressing convolutional neural networks with cheap convolutions and online distillation. Displays, 78, 102428–102428. https://doi.org/10.1016/j.displa.2023.102428

[4] Chen Sian, Research on Model Compression and Hardware Acceleration of Convolutional Neural Networks [D]. Zhejiang University, 2021. DOI:10.27461/d.cnki.gzjdx.2021.001680.

[5] Guo Qingbei, Research on Compression and Acceleration Techniques of Deep Convolutional Neural Networks [D]. Jiangnan University, 2021. DOI:10.27169/d.cnki.gwqgu.2021.001906.

[6] Mei Qichang, Research and Application of FPGA-based Convolutional Neural Network Accelerator [D]. Nanjing University of Posts and Telecommunications, 2023. DOI:10.27251/d.cnki.gnjdc.2023.000260.

[7] Chen Guilin, Wang Guanwu, Wang Kang, et al. KCNN: A Lightweight Neural Network Method and Hardware Implementation Architecture [J/OL]. Journal of Computer Research and Development, 1-10[2024-05-23]. http://kns.cnki.net/kcms/detail/11.1777.tp.20240219.1308.002.html.

[8] Ghanbari, A., & Modarressi, M. (2022). Energy-efficient acceleration of convolutional neural networks using computation reuse. Journal of Systems Architecture, 126, 102490. https://doi.org/10.1016/j.sysarc.2022.102490

[9] Wang Jiong, Research on Model Compression Methods of Deep Neural Networks Based on Structured Pruning [D]. Nanjing University of Posts and Telecommunications, 2022. DOI:10.27251/d.cnki.gnjdc.2022.001412.

[10] Sun Jianhui, Research on Key Technologies for Hardware Implementation of Convolutional Neural Networks [D]. Shanghai Jiao Tong University, 2020. DOI:10.27307/d.cnki.gsjtu.2020.003022.

[11] Zhao Jiacheng, Research on Model Acceleration for Convolutional Neural Networks [D]. Nanjing University, 2021. DOI:10.27235/d.cnki.gnjiu.2021.002191.

[12] Zheng, Y., Sun, P., Ren, Q., Xu, W., & Zhu, D. (2024). A novel and efficient model pruning method for deep convolutional neural networks by evaluating the direct and indirect effects of filters. Neurocomputing, 569, 127124–127124. https://doi.org/10.1016/j.neucom.2023.127124

[13] Luo Deyu, Guo Qianxi, Zhang Huaicheng, et al. FPGA Deployment of Quantized Convolutional Neural Networks Based on Knowledge Distillation [J]. Electronics Technology Application, 2024, 50(04): 97-101. DOI:10.16157/j.issn.0258-7998.234479.

[14] Wang Dongwei, Liu Baichen, Han Zhi, et al. Deep Network Compression Method Based on Low-Rank Decomposition and Vector Quantization [J/OL]. Computer Application, 1-9[2024-05-23]. http://kns.cnki.net/kcms/detail/51.1307.TP.20230927.1508.006.html.st numbered section of the paper.