A High-performance Inference Accelerator Exploiting Patterned Sparsity in CNNs

Ning Li, Leibo Liu, Shaojun Wei, Shouyi Yin*
Department of Microelectronics and Nanoelectronics, Tsinghua University, Beijing, China
Email: {n-li17, wsj}@mails.tsinghua.edu.cn, {liulb, yinsy}@tsinghua.edu.cn

***Abstract*—Convolutional neural networks (CNNs) have emerged as the critical technology for deep learning with significantly growing computation and memory demands. Model compression has been widely acknowledged as an effective way to achieve acceleration on CNNs. However, most proposed architectures of FPGA are inefficient for compressed models that contain a large amount of zero operations. In this work, we propose a sparse CNNs inference accelerator on FPGA utilizing uniform sparsity introduced by pattern pruning to achieve high energy efficiency. Our architecture maintains the sparse weights in a compressed format to reduce the storage demands and displays a flexible kernel-stationary dataflow to enable the extensive data reusing. In addition, we design flexible computing arrays which can be dynamically reconfigured to balance workload with low overheads. Specially, the on-chip memory applies a novel data buffering structure with slightly rearranged sequences to address the challenge of access conflict. The experiments show that our accelerator can achieve $316.4GOP/s \sim 343.5GOP/s$ for VGG-16 and ResNet-50.**

I. Introduction

To achieve the balance between the accuracy of network and the friendliness of hardware, our architecture utilizes the uniform sparsity introduced by pattern pruning and connectivity pruning [1]. PCONV [1] produces the same sparsity ratio in each filter and a limited number of pattern shapes. The existing FPGA-based architectures focus on accelerating the calculation of common CNNs [2]–[4]. Such accelerators are designed for a regular calculation loop that can not eliminate inefficient operations in sparse networks. Moreover, it is wasteful for on-chip memory to store the uncompressed parameters and easy to occur severe access conflict.

II. Dataflow and Architecture

We design an efficient dataflow for pruned convolutional layers based on compressed and encoded weights. There are three features in our dataflow: (i) we gather the efficient kernels for the same channel (K_1, K_3, K_5, K_9) in one group and apply a channel-wise loop unrolling to promote input reusability; (ii) the output parallelism is dynamically adjustable to balance workload on computing resources; (iii) the kernel-stationary dataflow improves the reuse of weights and decreases the amount of indexes decoding at the same time. Our overall architecture is displayed in Fig 1.

III. Experiment and Evaluation

To evaluate the performance of our architecture, we implemented a hardware prototype on Xilinx ZC706 platform running at 200MHz. 1GB DDR3 DRAMs are used. The experiments show that our accelerator can achieve $316.4GOP/s \sim 343.5GOP/s$ for VGG-16 and ResNet-50 with different sparsity. And the computing efficiency has been improved $14.7\% \sim 44\%$ by applying our dynamic parallelism and buffer design.

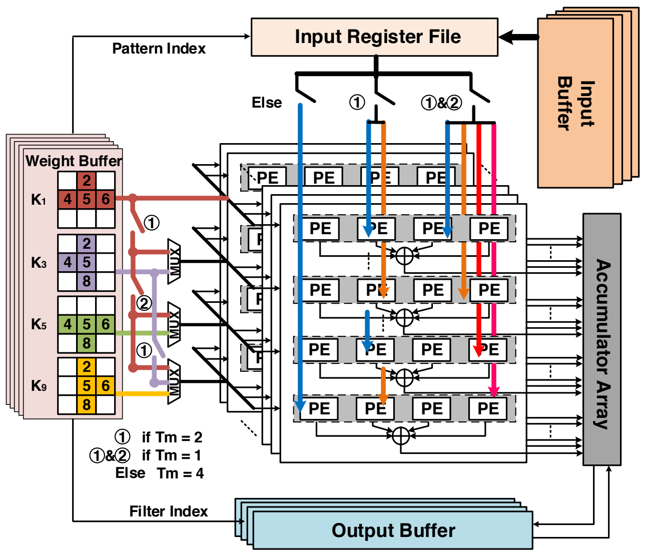

Fig. 1. The overall architecture

Acknowledgment

This work was supported in part by the National Key Research and Development Project (No. 2018YFB2202600), the NSFC under Grant U19B2041 and 61774094, and the China Major ST Project (No.2018ZX01031101-002).

References

[1] X. Ma, F.-M. Guo, W. Niu, X. Lin, J. Tang, K. Ma, B. Ren, and Y. Wang, "Pconv: The missing but desirable sparsity in dnn weight pruning for real-time execution on mobile devices," 09 2019.
[2] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing fpga-based accelerator design for deep convolutional neural networks," FPGA '15, 2015.
[3] Y. Ma, Y. Cao, S. Vrudhula, and J.-s. Seo, "Optimizing loop operation and dataflow in fpga acceleration of deep convolutional neural networks," FPGA '17, 2017.
[4] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, and et al., "Going deeper with embedded fpga platform for convolutional neural network," FPGA '16, 2016.

2576-2621/20/$31.00 ©2020 IEEE
DOI 10.1109/FCCM48280.2020.00076

IEEE computer society

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on November 27,2020 at 14:14:27 UTC from IEEE Xplore. Restrictions apply.