Explore Efficient LUT-based Architecture for Quantized Convolutional Neural Networks on FPGA

Yanpeng Cao, Chengcheng Wang, Yongming Tang
Joint International Research Laboratory of Information Display and Visualization, Southeast University, Nanjing, China
Email: {220181334, 213162565, tym}@seu.edu.cn

***Abstract*—The vast computations of the convolutional neural network have limited the speed of the forward inference running in hardware. In recent years, network quantization technique has made it possible to quantize network into low bit-wide and retain the original performance simultaneously, while the complexity of the quantized network is still considerable. FPGA is a highly parallelized platform, which contains a mass of configurable logic resources. We study on the feasibility of implementing convolution calculation based on pure LUTs, introduce the shift multipliers and addition trees, and propose an efficient architecture for QNN on FPGA. With the optimization of Winograd algorithm for QNN, we demonstrate that our scheme significantly reduces the number of multipliers and saves the usage of LUT resources by 2.25× at least without using DSP resources. As a result, our LUT-based architecture for QNN shortens the latency up to 19.3× and represents more effective performance compared to other methods.**

I. Proposed Architecture

Exploring efficient hardware architecture for quantized neural networks [1], QNN for short, is necessary to eliminate the bottleneck of high-density computing requirements. Figure 1 illustrates the schematic diagram about the proposed LUT-based architecture, named Wino-Conv, from bottom to top. Our design can be divided into the following aspects:

1) Due to the data streams of the convolutional layer in the quantized network are low bit-width, the bottom units can be constructed with LUT-based shift multipliers to replace the DSP multipliers.
2) From architecture-wise, we introduce the Winograd minimal algorithm [2] to optimize the inner kernels.
3) From the perspective of trade-off between the resource utilization and convolutional throughput, we explore the optimal parameters in the search space.
4) Finally, we optimize the parallelism of the inter kernels from memory-wise.

II. Experimental Results

For the proposed Wino-Conv architecture, the latency of single convolutional operation is obtained by controlled experiments compared to the LUT-based Direct-Conv operation and the DSP-based convolutional operation. We compute the value of speed up of convolutional computation between these three methods. For the Direct-Conv based on LUTs, its accelerating speedup fluctuates slightly by the factors of output size and data bit-width, which is basically stable around $3.0\times \sim 5.2\times$. While, there is an obvious accelerating gain for the proposed Wino-Conv. When the output size gets larger, Wino-Conv will greatly improve the speed performance from 6.5× up to 19.3×. For an optimal setting for resource utilization, when $m = 4$, there is still a speedup of 12.5×, which remains an absolute advantage compared to DSP-based method.

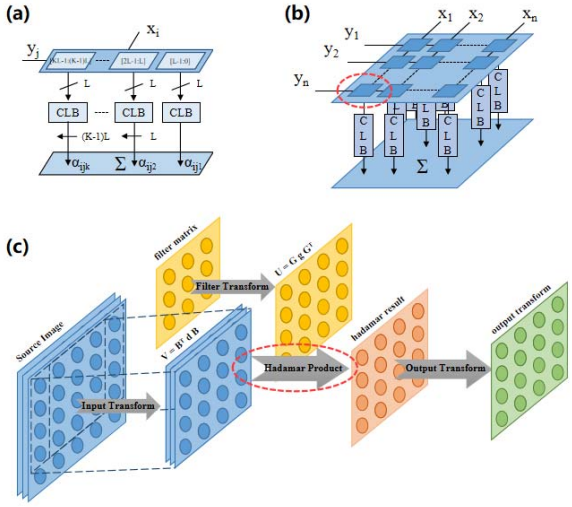

Fig. 1. Overview of the proposed Wino-Conv. (a) Shift multiplication; (b) Hadamar product; (c) The processing flow of winograd convolution calculation.

III. Conclusion

In this work, we were committed to explore a convolutional accelerator without DSPs on FPGA. Results show the distinct LUT reductions and computation speedup, which makes our method an ideal candidate for the future quantized CNN architecture. Besides, our convolutional accelerator can be easily embedded into the existing networks.

Acknowledgment

The authors would like to thank the Academic Colleges and Universities Innovation Program 2.0 (grant number BP0719013) supported by the Ministry of Education of the People's Republic of China.

References

[1] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research,* vol. 18, no. 1, pp. 6869-6898, 2017.
[2] S. Winograd, *Arithmetic complexity of computations.* Siam, 1980.

2576-2621/20/$31.00 ©2020 IEEE
DOI 10.1109/FCCM48280.2020.00065