2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines

Accelerator Design with Effective Resource Utilization for Binary Convolutional Neural Networks on an FPGA

Sunwoong Kim* and Rob A. Rutenbar†
*Department of Computer Science, University of Illinois at Urbana-Champaign
†Department of Computer Science and Department of Electrical and Computer Engineering, University of Pittsburgh
Email: sunwoong@illinois.edu, rutenbar@pitt.edu

A number of recent deep learning studies have decreased the precision of operations and operands to reduce computation and storage while minimizing loss of accuracy. In binary convolutional neural networks (BCNNs) that lower the precision to essentially single-bit binary, arithmetic operations such as real-valued multiplication are replaced by bitwise operations and the memory size for the weights and feature maps is greatly reduced [1]. Since FPGAs have hundreds of thousands or millions of LUTs, higher parallelism than in GPUs can be exploited, which is a good opportunity to accelerate training and inference.

In the proposed BCNN architecture, the XNOR operation and accumulation between 2D-weight (e.g. 3×3) and 2D-input feature map (ifmap) (e.g. 32×32) are processed per cycle. For example, the structure for CIFAR10 datasets [2]-[4], where 32 is the maximum width (and height) of ifmaps, deploys 32×32-processing elements (PEs).

Although the use of 32×32-PEs achieves high-throughput, there is a resource utilization issue after pooling. If pooling uses a 2×2 filter, the width and height of ifmaps for the next layer decrease by half. For high resource utilization, the next consecutive operations are forwarded and processed in parallel. For example, the width (and height) of ifmaps for the fourth convolution (Conv) layer for CIFAR10 datasets is 16 that is half of 32. Therefore, 3-consecutive 16×16-ifmaps and 3×3-weights are forwarded, and XNOR operation and accumulation between 3×3×4-weight and 16×16×4-ifmap are processed in parallel. Depending on layers, different weight arrangements and boundary controls are applied.

To further increase the resource utilization, fully-connected (FC) layers reuse the existing PEs designed for Conv layers. For example, the first FC layer performs matrix multiplication between 1×8192-ifmap and 8192×1024-weight. Firstly, the left-most 1×8-ifmap and the upper left-most 8×1-weight are entered into the PE(0, 0). Since there are nine XNOR gates on the single PE, one XNOR gate is not used and the 9-to-1 adder acts like an 8-to-1 adder. Lastly, the PE(31, 31) receives the same 1×8-ifmap and the upper right-most 8×1-weight. Since 32×32-PEs work in parallel, matrix multiplication between 1×8-ifmap and 8×1024-weight is performed per cycle. To complete the first FC layer, 1024 (=8192/8) cycles are required.

The proposed design is compared with three previous BCNN inference designs for CIFAR10 datasets. Zhao *et al.* exploit three engines for the first Conv layer, the other Conv layers, and FC layers [2]. Umuroglu *et al.* propose a streaming architecture and allocate hardware resources depending on FPS and network requirements [3]. They implement separate computing engines with different configurations for respective layers. Note that Umuroglu *et al.* show the highest FPS value of 21,900 as shown in the eighth row of Table I, but the number of filters is half of other designs. Li *et al.* propose a BCNN inference design with high throughput and energy efficiency, using architectural unfolding and pipelining schemes [4]. The tenth row of Table I presents GOPS/kLUT results related to the resource utilization. Even though the proposed design uses a single engine without optimal configurations on each Conv or FC layer, it achieves the highest GOPS/kLUT value of 61.6.

Table I
COMPARISON WITH PREVIOUS BCNN INFERENCE DESIGNS ON FPGAS

	[2]	[3]	[4]	Proposed
Device	XC7Z020	XC7Z045	XC7VX690	XCVU190
kLUT	47/53	46/219	342/433	61/1,074
kFF	46/106	N/A	71/607	45/2,148
BRAM (Mbit)	3.3/4.9	6.5/19.2	18.1/37.1	13.9/132.9
DSP	3/220	0/900	1,096/2,800	0/1,800
Freq. (MHz)	143	200	90	240
FPS	168	21,900	6,218	3,044
GOPS	208	2,465	7,663	3,756
GOPS/kLUT	4.4	53.3	22.4	61.6
Power (W)	4.7	3.6	8.2	5.9

REFERENCES

[1] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or −1," in *arXiv preprint arXiv: 1602.02830*, Feb. 2016.

[2] R. Zhao et al., "Accelerating Binarized Convolutional Neural Networks with Software-Programmable FPGAs," in *FPGA*, 2017.

[3] Y. Umuroglu et al., "FINN: A Framework for Fast, Scalable Binarized Neural Network Inference," in *FPGA*, 2017.

[4] Y. Li, Z. Liu, K. Xu, H. Yu, and F. Ren, "A GPU-Outperforming FPGA Accelerator Architecture for Binary Convolutional Neural Networks," in *arXiv preprint arXiv: 1702.06392*, Feb. 2017.

2576-2621/18/$31.00 ©2018 IEEE
DOI 10.1109/FCCM.2018.00052