An Energy-efficient Deep Neural Network Training Processor with Bit-slice-level Reconfigurability and Sparsity Exploitation

Donghyeon Han, Dongseok Im, Gwangtae Park, Youngwoo Kim, Seokchan Song, Juhyoung Lee, and Hoi-Jun Yoo

School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea, E-mail: hdh4797@kaist.ac.kr

Abstract

This paper presents an energy-efficient deep neural network (DNN) training processor through the four key features: 1) Layer-wise Adaptive bit-Precision Scaling (LAPS) with 2) In-Out Slice Skipping (IOSS) core, 3) double-buffered Reconfigurable Accumulation Network (RAN), 4) momentum-ADAM unified OPTimizer Core (OPTC). Thanks to the bit-slice-level scalability and zero-slice skipping, it shows 5.9 × higher energy-efficiency compared with the state-of-the-art on-chip-learning processor (OCLPs).

1 Introduction

Training DNN requires a significant amount of computation which is possible only on cloud servers. Therefore, most of the DNN accelerators for mobile applications only support DNN inference (INF). However, OCLP for edge or mobile applications are recently receiving increasing attention due to its ability to deliver personalized applications using user-specific data, and continuously compensate for accuracy degradation by adapting to environmental variations in real-time. Furthermore, distributed learning based on OCLP can accelerate training while protecting privacy by avoiding the upload of user's private data to the servers.

The implementation of OCLP on edge and mobile platforms is extremely challenging because of two major difficulties. At first, it needs high throughput > 3 TOPS because it requires not only feed-forward (FF) but also error-propagation (EP) and weight gradient (WG) stages with multiple inputs and iterations. Secondly, a high bit-precision (HBP) > FP16 is required to precisely represent the error in EP stage. Unlike INF, most OCLPs shows low efficiency due to HBP computing. Thus, conventional OCLPs tried to compensate efficiency through the in- and out- word-level sparsity exploitation induced by ReLU. However, it induces high logic complexity and still shows lower energy-efficiency compared with the conventional INF processors. Moreover, the efficiency improvement is limited when the other non-linear function is used instead of ReLU.

In this paper, a maximum of 50.3TOPS/W, OCLP is proposed to solve these difficulties with the following 4 key features. Firstly, stochastic rounding (SR, [1]) and LAPS enable low bit-precision (LBP) training and minimize energy consumption by discarding useless computations. Secondly, quad-bit-slice-level sparsity is exploited even with various activation functions. Third, double-buffered RAN with long-term (LT) and short-term (ST) data management enables reconfigurable core allocation while maintaining high throughput. At last, unified OPTC is designed to support two typical types of optimizer during the training.

2 Processor Architecture

Fig. 1 shows the overall architecture of the processor which consists of 32 bit-slice training cores (BSTCs), a peripheral training assistant (PTA), 2 OPTCs, and a TOP RISC controller. Each BSTC has 4 processing element (PE) rows, input-activation (IA) pre-fetchers, 4KB IA memory, and buffers to store the weights and output activations (OAs). Every BSTC is connected with both the data NoC and the 2-D mesh-type RAN. An accumulation switch (ACC SW) placed in the BSTC includes a dynamic fixed-point (DFXP, [2]) unit which performs SR to convolution results to quantize it as target lower bit-precision. The SR is realized by adding noise to the LSB with a pseudo-random number generator. The LAPS unit is indicated in the PTA with a 120 KB peripheral memory (PMEM). As shown in Fig. 2, the proposed processor modifies existing task division methodologies [2] and uses modified methods to support not only the FF stage but also EP and WG stage within limited on-chip and off-chip memory bandwidth.

Fig. 3 illustrates LAPS which searches the required precision of each layer automatically by calculating the difference between LBP and HBP convolution results. The difference calculation occurs once every epoch and the revised precision is reflected for the remaining iterations. BSTC, which adopts a bit-slice serial architecture (BSSA), enables SR and LAPS to enhance the training speed and efficiency further. It uses 4b × 4b MAC as a base unit and receives a 4b-slice at a time after dividing HBP data into a number of 4b-slices. The BSTC can respond to the bit-precision changes determined by LAPS and it can minimize the throughput degradation appeared by LAPS-based active precision searching. Compared to the training of ResNet-9 with DFXP [2], the BSTC with SR and LAPS shows 3.19× higher energy-efficiency.

2021 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS) | 978-1-6654-1503-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/COOLCHIPS52128.2021.9410324

 978-1-6654-1503-3/21/$31.00 ©2021 IEEE

The IOSS architecture consists of sparsity-aware input-slice-skipping (ISS, Fig. 4) and precision-aware output-slice-skipping (OSS, Fig. 5) functionalities. ISS uses IA pre-fetchers to detect the zero-slices (ZS), and the internal ZS counter generates ZS length which is used as the address of the weight buffer. Only the remaining non-zero-slices (NZS) are stacked in a FIFO then fed into the PE arrays sequentially. ISS, together with BSSA, shows a 7.9× higher throughput at 90% slice-sparsity in 8b training cases. The OSS omits useless ACCs, and the OSS controller skips the ACC based on the precision information of the current and next layers stored in the register file. With the OSS, the throughputs of 8b and 16b convolution are improved by 33% and 67%, respectively. Combining ISS and OSS, the IOSS compensates for the throughput degradation that occurred in HBP training leading to 82.9% higher energy-efficiency compared with the GANPU [6]. Since the IOSS utilizes slice-level sparsity, it is still effective even with the non-ReLUs such as tanh and sigmoid.

Proposed processor modifies RAN [6] so that it can be applied to the DFXP based training. In addition, it includes double-buffer for pipeline structure between BSTC and ACC SW. A buffer consists of LT-MEM and ST-MEM. The ACC data is divided and stored in two distinct buffers depending on whether it is an incomplete result or not. The data stored in ST-MEM can be directly used as the next layer input but the data in LT-MEM should be aggregated with the ACC results in the adjacent input tiles before it transferred to the next layer.

The processor also includes OPTCs which can support both momentum and ADAM based optimization. Unlike momentum, the ADAM optimizer requires complex arithmetic operations such as division and square-root. Additionally, it increases both internal and external memory access by 40%. To resolve this problem, the proposed processor adopts signadam++ [3] which realizes ADAM optimizer by adopting a sign-only gradient in the momentum operations and removing the other complex operations. Thanks to signadam++, OPTC can unify both momentum and ADAM optimizer operations within minimized calculation units (Fig. 7).

3 Implementation Results & Conclusion

As shown in Fig. 8, the proposed processor is fabricated in 28nm CMOS technology and occupying a 12.96 mm^2 area. It operates at 0.58-to-1.04V supply voltage with 2-to-250 MHz core frequency. The peak performance and energy-efficiency are shown in 4b training with 12.3 TOPS and 50.3 TOPS/W respectively. The efficiency of the > 4b training varies from 1.4 TOPS/W (16b without IOSS) to 33.3 TOPS/W (8b with IOSS, 90% slice-sparsity) affected by its precision and input slice sparsity.

Table 1 shows the measurement results of the proposed processor and its comparison table. A DNN training benchmark with three different image classification datasets and widely-used DNNs such as ResNet-9, 18, VGG-16, and SENet are used for the comparison with previous OCLPs. SR improves the training energy-efficiency by × 4.5 thanks to the reduction of required precision. In addition, IOSS further increases its efficiency by × 2.4. At last, LAPS enhances the energy-efficiency more by × 1.5. Based on the benchmark result, the proposed processor shows at least 5.9× higher energy-efficiency compared with the previous OCLPs [4-7]. Specifically, the processor shows 85.1% energy saving during CIFAR-100 training. (Fig. 9)

In conclusion, an adaptive DNN training processor with SR and LAPS as well as IOSS can achieve ultra-low bit-width (< 8b) training and state-of-the-art energy-efficiency, 50.3TOPS/W.

References

[1] S. Gupta et al., "Deep learning with limited numerical precision," 2015 International Conference on International Conference on Machine Learning (ICML), vol 37, pp.1737–1746.

[2] D. Shin, J. Lee, J. Lee and H. Yoo, "14.2 DNPU: An 8.1TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks," 2017 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, 2017, pp. 240-241.

[3] D. Wang et al., "signADAM++: Learning Confidences for Deep Neural Networks," 2019 International Conference on Data Mining Workshops (ICDMW), Beijing, China, 2019, pp. 186-195.

[4] D. Han et al., "A 1.32 TOPS/W Energy Efficient Deep Neural Network Learning Processor with Direct Feedback Alignment based Heterogeneous Core Architecture," 2019 Symposium on VLSI Circuits, Kyoto, Japan, 2019, pp. C304-C305.

[5] J. Lee, et al., "7.7 LNPU: A 25.3TFLOPS/W Sparse Deep-Neural-Network Learning Processor with Fine-Grained Mixed Precision of FP8-FP16," 2019 IEEE International Solid- State Circuits Conference - (ISSCC), San Francisco, CA, USA, 2019, pp. 142-144.

[6] S. Kang et al., "7.4 GANPU: A 135TFLOPS/W Multi-DNN Training Processor for GANs with Speculative Dual-Sparsity Exploitation," 2020 IEEE International Solid- State Circuits Conference - (ISSCC), San Francisco, CA, USA, 2020, pp. 140-142.

[7] J. Oh et al., "A 3.0 TFLOPS 0.62V Scalable Processor Core for High Compute Utilization AI Training and Inference," 2020 IEEE Symposium on VLSI Circuits, Honolulu, HI, USA, 2020, pp. 1-2.

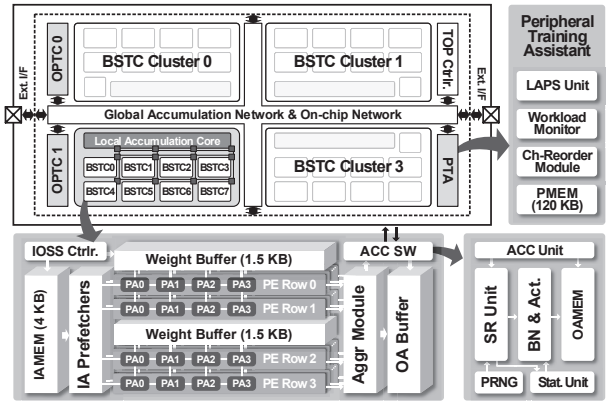


Fig.1 Overall Architecture of the Proposed Processor

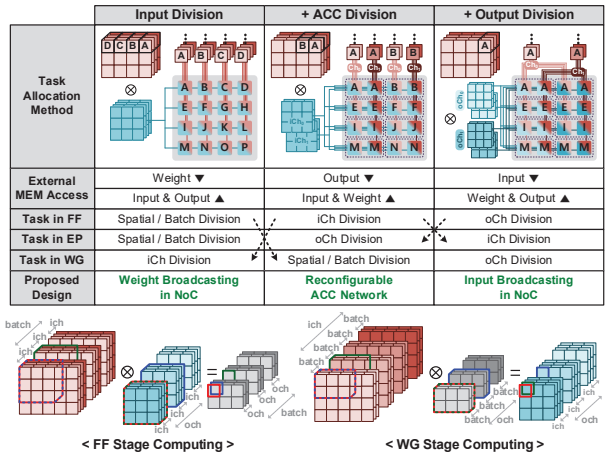


Fig.2 Task Allocation Methodologies during 3 Training Stages

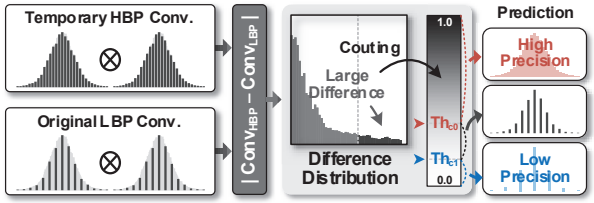


Fig.3 Overview of LAPS based Active Precision Optimization

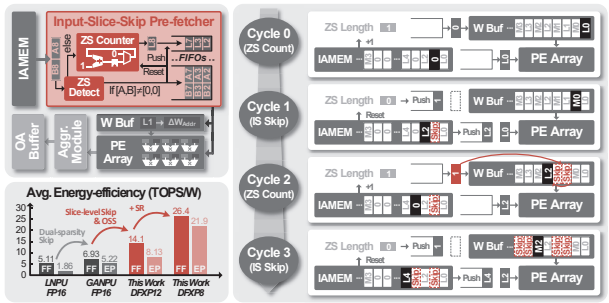


Fig.4 Input-slice Skipping Architecture and its Operation

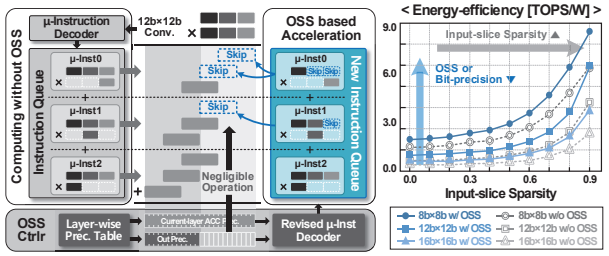


Fig.5 Output-slice Skipping based Efficiency Improvement

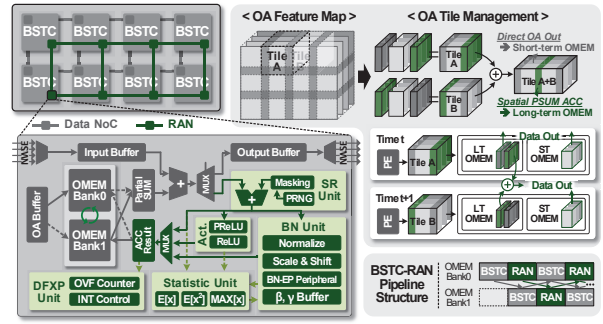


Fig.6 Double-buffered RAN with LT and ST Data Management

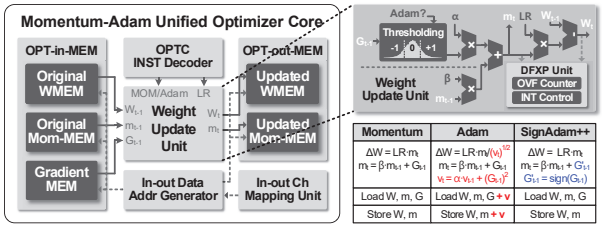
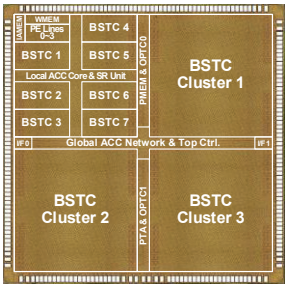


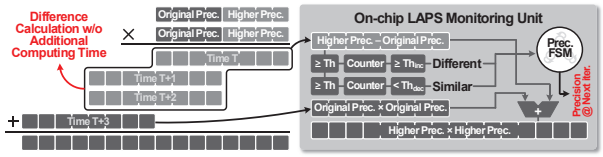
Fig.7 Momentum-ADAM Unified OPTC with SignAdam++



Specifications			
Technology	Samsung 28nm 1P8M CMOS		
Die Area	3.6mm × 3.6mm (12.96mm²)		
SRAM	552 KB		
Op. Condition	0.58V (@ 2MHz) ~ 1.04V (@ 250MHz)		
Data Type	DFXP + SR (4/8/12/16)-bit × (4/8/12/16)-bit		
Area Efficiency	59-to-948 GOPS/mm²		
Power [mW]	In-slice Sparsity	0%*	90%**
	250MHz, 1.04V	1162	500
	2MHz, 0.58V	4.39	1.95
Peak Energy Efficiency [TOPS/W]	16b×16b	1.40	14.9
	8b×8b	5.60	33.3
	4b×4b	22.4	50.3

I/O Voltage = 1.8V, 1 MAC = 2 OP ** w/o Output slice Skip ** w/ Output slice Skip*

Fig.8 Chip Photograph and Performance Summary



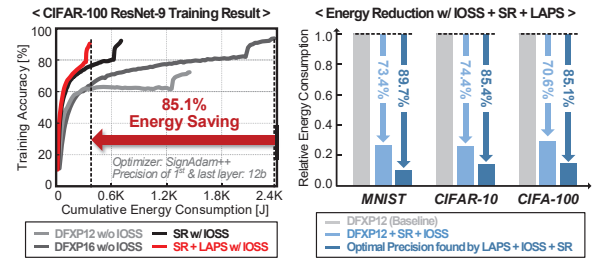


Fig.9 DNN Training Acceleration by Proposed Processor with LAPS based Optimal Precision Search and IOSS

		Tesla V100	[4]	[5]	[6]	[7]	This Work
Low-precision Train		X	X	X	X	X	O
Skipping Technique		-	X	IA skip	IA & OA skip	X	IS & OS skip
Process [nm]		12	65	65	65	14	28
Die Area [mm²]		815	5.76	16	32.4	9.8	12.96
Supply Voltage [V]		-	0.78-1.1	0.78-1.1	0.70-1.1	0.54-0.80	0.58-1.04
Max Frequency [MHz]		1530	200	200	200	1500	250
Precision	W	FP 16/32/64	FXP16	FP 8/16	FP 8/16	FP 16/32	SR+DFXP 4/8/12/16
	IA	FP 16/32/64	FXP13	FP 8/16	FP 8/16	FP 16/32	SR+DFXP 4/8/12/16
Avg. Energy Efficiency @ Training[1),2)] [TOPS/W or TFLOPS/W]	FF	0.42 (FP16)	1.05 (FXP16)	5.11 (FP16)	6.93 (FP16)	1.4 (FP16)	34.17 (SR + LAPS)
	EP	0.42 (FP16)	1.05 (FXP16)	1.86 (FP16)	5.22 (FP16)	1.4 (FP16)	37.09 (SR + LAPS)

1) ResNet-9, 18, VGG-16, SENet Training 2) Operating Condition: MAX Efficiency

Table 1 CNN Training Benchmark Result (Comparison Table)