ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 20-03-29
j.inst.Korean.electr.electron.eng.Vol.24,No.3,895~900,September 2020
http://dx.doi.org/10.7471/ikeee.2020.24.3.895
219

딥러닝을 하드웨어 가속기를 위한 저전력 BSPE Core 구현

Implementation of low power BSPE Core for deep learning hardware accelerators

조 철 원*, 이 광 엽**, 남 기 훈★

Cheol-Won Jo*, Kwang-Yeob Lee**, Ki-Hun Nam★

Abstract

In this paper, BSPE replaced the existing multiplication algorithm that consumes a lot of power. Hardware resources are reduced by using a bit-serial multiplier, and variable integer data is used to reduce memory usage. In addition, MOA resource usage and power usage were reduced by applying LOA (Lower-part OR Approximation) to MOA (Multi Operand Adder) used to add partial sums. Therefore, compared to the existing MBS (Multiplication by Barrel Shifter), hardware resource reduction of 44% and power consumption of 42% were reduced. Also, we propose a hardware architecture design for BSPE Core.

요 약

본 논문에서 BSPE는 전력이 많이 소모되는 기존의 곱셈 알고리즘을 대체했다. Bit-serial Multiplier를 이용해 하드웨어 자원을 줄였으며, 메모리 사용량을 줄이기 위해 가변적인 정수 형태의 데이터를 사용한다. 또한, 부분 합을 더하는 MOA(Multi Operand Adder)에 LOA(Lower-part OR Approximation)를 적용해서 MOA의 자원 사용량 및 전력사용량을 줄였다. 따라서 기존 MBS(Multiplication by Barrel Shifter)보다 하드웨어 자원과 전력이 각각 44%와 42%가 감소했다. 또한, BSPE Core를 위한 hardware architecture design을 제안한다.

Key words : Deep Learning, quantization, BSPE, LOA, Overlapping Computation

Ⅰ. 서론

딥러닝 알고리즘의 연구가 활발히 이루어지면서 이러한 알고리즘을 이용한 어플리케이션들이 다양한 분야와 장치에서 활용되고 있다. 사용자들이 주로 사용하는 디바이스가 PC에서 모바일로 이동하면서 모바일에서 딥러닝 어플리케이션을 수행하기 위한 노력이 동반되고 있다[5][6][7].

모바일 또는 엣지 디바이스에서 딥러닝 어플리케이션을 수행하기에는 다양한 한계점이 있다. 하드웨어의 자원이 한정적이며, 전력 사용의 제한이 있다. 또한, 네트워크의 연결이 원활하지 않으며 계산

*Dept. of Computer Eng., Seokyeong University
**Dept. of Electronics and Computer Eng., Seokyeong University
★Corresponding author
E-mail : namkh@skuniv.ac.kr, Tel : +82-2-940-7667
※ Acknowledgment
Manuscript received Sep. 18, 2020; revised Sep. 27, 2020; accepted Sep. 28, 2020.
This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP). (No. 2016-0-00204, Development of mobile GPU hardware for photo-realistic realtime virtual reality)

집약적인 딥러닝 알고리즘을 수행하기에는 부담이 크다.

위와 같은 한계점을 극복하기 위해 전용 하드웨어 가속기의 연구가 활발히 진행되고 있다. 딥러닝 전용 하드웨어 가속기는 GP-GPU(General Purpose computing on Graphics Processing Units)보다 자원 대비 연산 효율과 전성비가 좋아 모바일 또는 엣지 디바이스에서 딥러닝 애플리케이션을 수행하기에 적합하다.

본 논문에서는 저전력으로 딥러닝 알고리즘을 수행하는 BSPE Core를 제안한다. BSPE Core 내부에는 m개의 BSPE가 있어 최대 m개의 데이터를 이용해서 MAC 연산을 수행할 수 있다. BSPE는 Bit-serial Multiplier[8]를 기반으로 기존의 곱셈 알고리즘을 대체한다.

딥러닝 알고리즘의 특성상 MAC 연산의 출력값을 더해 부분 합을 구하는 동작이 많다. 따라서, 딥러닝 하드웨어 가속기에서 MOA는 약 69%를 차지한다.[9] 본 논문의 MOA는 딥러닝 알고리즘의 성능에는 영향을 끼치지 않고 하드웨어 자원의 소모를 줄이기 위해 LOA를 적용해 MOA의 크기를 줄였다. 본 논문은 BSPE를 활용한 BSPE Core에서의 효율적인 연산을 위한 hardware design을 제안한다.

II. 본론

1. Deep learning basics

딥러닝 알고리즘의 기본 연산은 수식 1과 같다.

$$a_i^l = \sigma((\sum_j a_i^{l-1} \cdot w_{ij}^l) + b_i^l) \quad (1)$$

Activation과 weight는 곱의 합 연산을 통해 더해지며 bias와 더해진다. 더해진 값은 최종 activation function을 통해 다음 layer의 activation 값이 된다.

Table 1. Definition of terms in Equation 1.
표 1. 수식 1의 용어 정의

Shape parameter	Description
a	activation
w	weight
b	bias
o	output
σ	activation function
i	column
j	row

2. MBS(Multiplication by Barrel Shifter)

MBS[9]는 기존의 곱셈기를 대체하는 곱셈 알고리즘이며 구조는 그림 1과 같다. 2개의 Bit-Brick[11]을 이용해서 기존의 곱셈 알고리즘을 대체한다. 입력으로 들어온 weight는 Booth's algorithms에 의해 인코딩된다. 인코딩된 weight는 나뉘어 Barrel Shifter로 전달된다. 전달된 weight에 의해 activation이 시프트된다. 이때, 출력값이 음수일 때 2의 보수 연산을 위해 시프트 출력값에 1을 더해줘야 한다. 하지만 MBS는 값이 음수일 때 1bit의 INV1과 INV2를 이용해 부분 합을 따로 구해 최종 부분 합과 더해져 출력값이 된다.

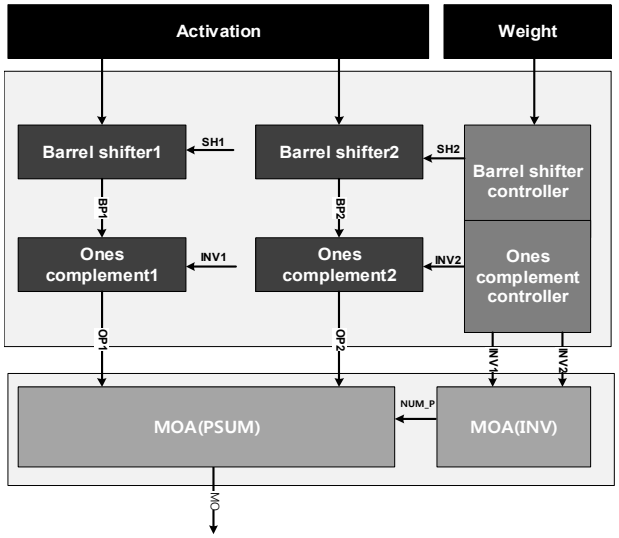

Fig. 1. Architecture of MBS.
그림 1. MBS(Multiplication by Barrel Shifter)의 구조

MBS는 가중치 인코딩과 Barrel shifter를 이용해 효율적으로 곱셈 연산을 하지만 한계가 있다. 첫째, 가중치를 인코딩하기 때문에 정해진 precision을 넘은 데이터는 사용할 수 없다. 이로 인해 사용할 수 있는 데이터의 정밀도는 한정된다. 둘째, 인코딩을 할 수 없는 가중치는 근사화를 한다. 정수 형태의 데이터를 사용하는 MBS는 인코딩 테이블에 의해 weight가 ±11 또는 13일 경우 각각 10과 12로 근사화 한다. Weight의 정밀도가 낮을 때는 인코딩을 할 수 없는 데이터는 적어지지만 정밀도가 높아짐에 따라 인코딩을 할 수 없는 데이터의 양이 늘어 근사화하는 비율이 높아진다. 셋째, Barrel Shifter를

사용해 하드웨어 자원을 많이 사용한다. Precision 높아짐에 따라 Barrel Shifter가 추가되어 높은 정밀도를 요구하는 어플리케이션을 수행하기 위해서는 많은 하드웨어 자원을 요구한다. 또한, 가중치 인코딩 과정 또한 복잡해진다.

이러한 한계점을 보완하기 위해 본 논문에서는 BSPE(Bit-Serial Processing Element)를 제안한다.

3. 제안하는 BSPE(Bit-Serial Processing Element)

BSPE는 MBS의 인코딩 방법 대신 Bit-Serial Multiplier를 이용해 곱셈 연산을 수행하며 그림 2와 같다.

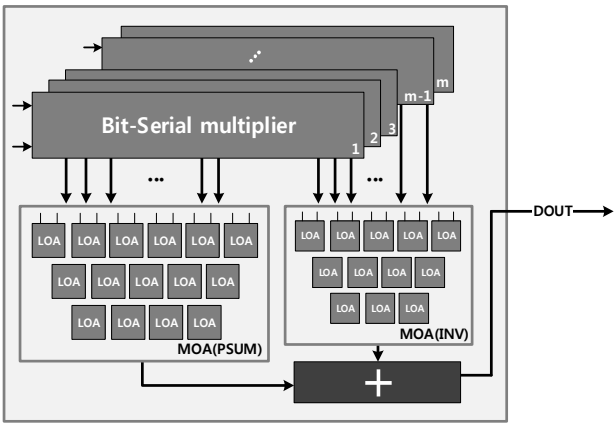

Fig. 2. Architecture of BSPE.
그림 2. BSPE의 구조

가. BSPE의 구조

BSPE는 Bit-Serial Multiplier에 Activation과 weight가 입력으로 들어가며, n-bit의 weight는 0번째 bit부터 n-1번째 비트로 차례대로 입력으로 들어가 곱셈 연산을 수행한다. 음수인 경우는 MBS와 마찬가지로 INV와 MOA(INV)를 이용해 음수 연산을 수행한다.

가중치 인코딩과 Barrel Shifter 대신 Bit-Serial Multiplier를 이용해 가변적인 precision의 데이터를 사용할 수 있으며, 인코딩을 할 수 없는 데이터를 근사화시키는 과정을 제거했다.

또한, MBS는 2개의 barrel-shifter의 출력값과 2개의 INV를 더하기 때문에 더 많은 Adder Tree를 소모하지만 BSPE는 1개의 출력값과 1개의 INV가 출력으로 나오기 때문에 이들의 합을 구하는 Adder Tree의 규모가 작아져 자원 소모량을 줄일 수 있었다.

나. LOA(Lower-part OR Approximation adder)

기존의 adder tree의 경우 n-bit의 full adder를 사용해서 덧셈을 수행한다. BSPE에서 제안하는 LOA는 딥러닝 알고리즘의 특징인 fault tolerance를 adder tree에 적용하며 그림 3과 같다.

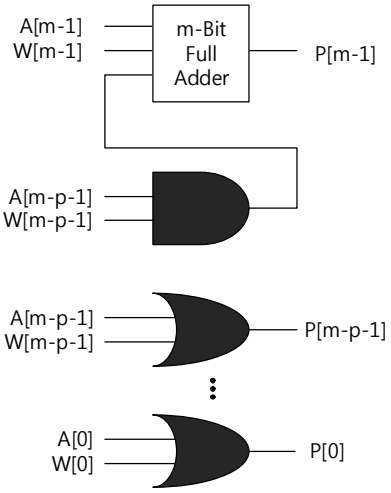

Fig. 3. Data prefetching and overlapping.
그림 3. 데이터 프리페칭과 오버래핑

M-bit의 adder에서 p-bit 미만은 full adder를 사용하지 않고 or-gate로 대체하며 p-bit 이상은 full adder를 사용한다.

전가산기는 2개의 xor-gate, 2개의 and-gate, 1개의 or-gate로 구성된다. 따라서 or-gate로 대체하면 총 2개의 xor-gate와 2개의 and-gate를 절약할 수 있다.

Table 1. Comparison between MBS and BSPE.
표 1. MBS와 BSPE의 비교

	MBS	BSPE	BSPE(LOA)
Total Power(mW)	16.1067	9.1745	9.1508
Chip Area	102968	63041	60106
Total Gates	11,726	7,179	6,845

표 1은 MBS와 BSPE를 CMOS 180nm공정 200MHz에서 합성한 결과이다.

4. Hardware architecture design for CNN

BSPE를 기반으로 합성곱 신경망에 응용하기 위해 BSPE에 최적하는 CNN 구조를 다음과 같이 제안한다.

가. Overlapping Computation

Overlapping Computation[12]은 서로 겹치는 데이터를 제외한 새로 필요한 데이터만을 가져온다. 컨벌루션 레이어는 그림 4와 같이 Current tile의 연산을 수행한 후 Next tile로 stride를 옮겨서 연산을 수행한다. 이때, 일반적인 방법은 Current tile의 데이터인 R1~R5의 데이터를 이용해 컨벌루션 연산을 한 후 다시 R2~R5의 데이터를 가져온다.

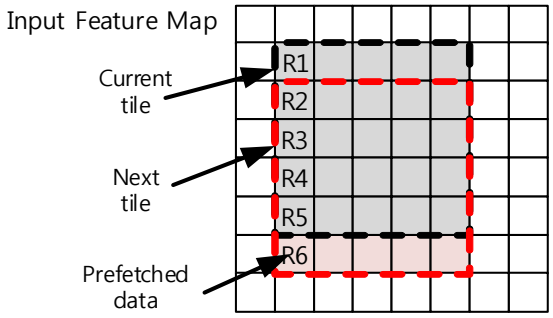

Fig. 4. Data prefetching and overlapping.
그림 4. 데이터 프리페칭과 오버래핑

하지만 Overlapping Computation을 적용해서 Next tile의 모든 데이터를 가져오는 것이 아닌 R6 데이터만을 가져와서 연산을 수행한다.

나. Data Prefetching

Current tile의 연산이 종료된 후에 Overlapping Computation에 의해 R6 데이터를 가져온다. R6를 가져오는 동안 연산기는 동작을 수행하지 않는다. 따라서 Data Prefetching 기법을 이용해 낭비되는 사이클을 제거했다.

Current tile을 연산하는 도중 R6 데이터를 미리 가져와 Scratch Pad에 저장해 Current tile의 연산이 종료되자마자 Next tile의 연산을 수행한다.

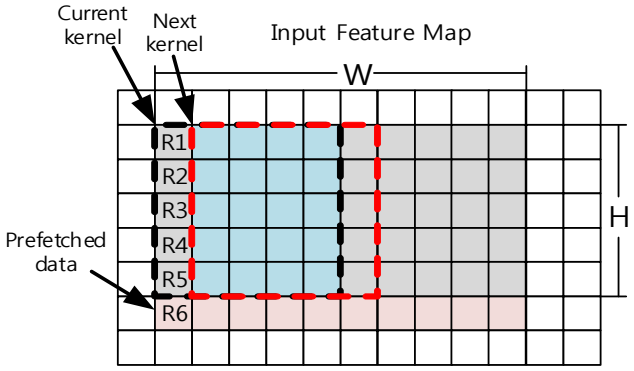

Fig. 5. Data reuse through variable data tiling.
그림 5. 가변적 데이터 타일링을 통한 데이터 재사용

다. Overlapping computation for Row data reuse

그림 5에서 W는 데이터 타일링의 width를 의미한다. 본 구조에서는 데이터를 overlapping computation row 단위로 수행한다. 만약 W를 본 실험의 예시에서처럼 kernel size 단위로 tiling을 한다면 파란색으로 표기된 부분은 Next kernel을 연산할 때 다시 가져온다.

따라서 본 논문에서는 가변적 tiling width를 두어 row 단위의 데이터도 재사용할 수 있다. Row stationary data reuse를 사용하지 않을 때 W*H를 수행하려면 데이터를 30번 전송해야 한다. 하지만 재사용을 통해 데이터를 R1~R5까지 5번만 전송해서 W*H 크기의 데이터를 연산할 수 있다.

4. BSPE Core

BSPE Core는 그림 6과 같다. Core controller와 Activation buffer, Weight buffer가 존재한다. Activation buffer와 Weight buffer는 Core controller로부터 값을 받아 BSPE로 데이터를 전송한다. 데이터를 전송받은 BSPE는 Core controller로부터 연산 시작 신호를 받아 연산을 수행한다.

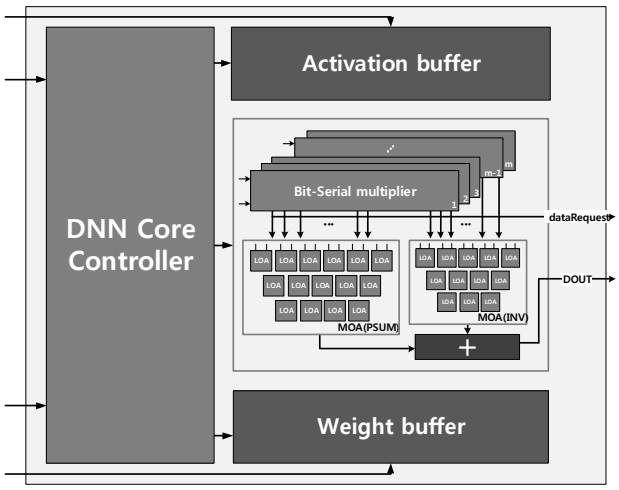

Fig. 6. BSPE Core.
그림 6. BSPE Core

III. 실험

본 실험에서 사용된 파라미터는 표 2와 같다. Activation은 8-bit를 사용했으며 weight는 5-bit를 사용했다. 또한, BSPE 내부의 곱셈기의 개수는 25개를 사용했다. 따라서 최대 5×5의 크기의 커널을 지원할 수 있다.

Table 2. Used parameter on this experiment.
표 2. 실험에서 사용된 파라미터

	parameter
Activation precision	8
Weight precision	5
m	25

Xilinx Vertex7 707 FPGA Board에서 50MHz로 합성을 수행했을 때 25개의 데이터를 연산하는데 180ns이 소요된다.

Table 3. Synthesis result of BSPE Core.
표 3. BSPE Core 합성 결과

	BSPE Core
Total Power(mW)	50.1152
Chip Area	429822
Total Gates	48,955

BSPE Core를 CMOS 180nm공정에서 100MHz의 동작 주파수로 합성한 결과는 표 3과 같다.

Ⅳ. 결론

본 논문은 기존의 곱셈 알고리즘인 MBS보다 적은 하드웨어 자원과 전력을 소모하여 컨벌루션 연산을 수행하는 BSPE를 제시했다. MBS를 Bit-Serial Multiplier로 대체하고 MOA에 LOA를 적용해서 하드웨어 자원과 전력 사용을 낮췄다. 결과적으로 하드웨어 자원과 전력이 각각 44%와 42%가 감소했다.

또한, BSPE를 이용한 BSPE Core를 위한 하드웨어 구조를 제안한다. 가변적인 타일링 길이와 col, row 단위의 overlapping computation을 이용해서 데이터 전송 횟수를 최소화했으며, 다음 연산할 tile 데이터를 prefetch해서 latency hiding을 통해 소모하는 사이클을 최소화했다.

References

[1] C. W. Cho, G. Y. Lee, “Low power for deep learning hardware accelerators Bit-Serial Multiplier based Processing Element,” *IKEEE Conference*, 2020.

[2] C. W. Cho, G. Y. Lee, “Bit-Serial multiplier based Neural Processing Element with Approximate adder tree,” *International SoC Design Conference (ISOCC)*, 2020.

[3] Mahdiani, Hamid Reza, et al. “Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” *IEEE Transactions on Circuits and Systems I: Regular Papers,* Vol.57, No.4 pp.850-862, 2009. DOI: 10.1109/TCSI.2009.2027626

[4] Abdelouahab, Kamel, Maxime Pelcat, and Francois Berry. “The challenge of multi-operand adders in CNNs on FPGAs: how not to solve it!,” *Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation.* pp.157-160, 2018. DOI: 10.1145/3229631.3235024

[5] Chen, Tianshi, et al. “Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning,” *ACM SIGARCH Computer Architecture News,* Vol.42, No.1, pp.269-284, 2014. DOI: 10.1145/2541940.2541967

[6] Chen, Yu-Hsin, et al. “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks,” *IEEE journal of solid-state circuits,* Vol.52, No.1 pp.127-138, 2016. DOI: 10.1109/JSSC.2016.2616357

[7] Jouppi, Norman P., et al. “In-datacenter performance analysis of a tensor processing unit,” *Proceedings of the 44th Annual International Symposium on Computer Architecture,* Vol.45, No.2, 2017. DOI: 10.1145/3140659.3080246

[8] Lee, Jinmook, et al. “UNPU: A 50.6 TOPS/W unified deep neural network accelerator with 1b-to-16b fully-variable weight bit-precision,” *2018 IEEE International Solid-State Circuits Conference-(ISSCC). IEEE,* 2018. DOI: 10.1109/ISSCC.2018.8310262

[9] Abdelouahab, Kamel, Maxime Pelcat, and Francois Berry. “The challenge of multi-operand adders in CNNs on FPGAs: how not to solve it!,” *Proceedings of the 18th International Conference*

on Embedded Computer Systems: Architectures, Modeling, and Simulation. pp.187-160, 2018.
DOI: 10.1145/3229631.3235024
[10] Park, Hyunbin, Dohyun Kim, and Shiho Kim. "Digital Neuron: A Hardware Inference Accelerator for Convolutional Deep Neural Networks," arXiv preprint arXiv:1812.07517, 2018.
[11] Sharma, Hardik, et al. "Bit fusion: Bit-level dynamically composable architecture for accelerating deep neural network," *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA). IEEE*, 2018.
DOI: 10.1109/ISCA.2018.00069
[12] Alwani, Manoj, et al. "Fused-layer CNN accelerators," *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE,* 2016.
DOI: 10.5555/3195638.3195664

BIOGRAPHY

Cheol-Won Jo (Member)

2019 : BS degree in Computer Engineering, Seokyeong University.
2019~present : MS degree in Electronics and Computer Engineering, Seokyeong University.

Kwang-Yeob Lee (Member)

1985 : BS degree in Electronics Engineering, Sogang University
1987 : MS degree in Electronics Engineering, Yonsei University.
1994 : PhD degree in Electronics Engineering, Yonsei University.
1989~1995.2 : Senior Researcher, Hyundai Electronics Inc.
1995.3~present : Professor, Dept. of Computer Engineering, Seokyeong University

Ki-Hun Nam (Member)

2000 : BS degree in Computer Science, Seokyeong University.
2005 : MS degree in Computer Science, Seokyeong University.
2006 : PhD degree in Computer Science, Seokyeong University.
2006.3~2017.2 : Software Department Adjunct Professor
2009.10~2011.2 : Researcher, Display Research Institute, Hanyang University
2011.3~2017.2 : Visiting Professor, Department of Computer Engineering, Seokyeong University
2017.3~present : Assistant Professor, Department of Computer Engineering, Seokyeong University