Xilinx DPU를 사용한 CNN 추론 분석

김채영*, 서태원**
*고려대학교 반도체시스템공학과
**고려대학교 컴퓨터학과
e-mail: chaeyuiop@korea.ac.kr

Analysis of CNN Inference Using Xilinx DPU

Chaeyoung Kim*, Taeweon Suh**
*Dept of Semiconductor System Engineering, Korea University
**Dept of Computer Science and Engineering, Korea University

요 약

지능형 IoT 애플리케이션들을 효과적으로 사용하기 위해서는 추론 엔진을 Edge device로 포팅하는 것이 필수적이다. 그러나 컴퓨팅 자원이 제한적인 Edge 환경에서 computational cost가 상당히 큰 CNN 추론을 실시간으로 하는 것은 쉽지 않다. 이에, CNN 추론의 하드웨어 가속화의 필요성이 제기되어 활발한 연구가 진행되고 있으며, Xilinx, Intel 등에서도 하드웨어 가속화를 도와주는 툴을 개발하여 지속적으로 업그레이드하고 있다. 본 연구에서는 CIFAR-10 데이터베이스의 테스트 이미지 10,000개를 Xilinx 사의 CNN 추론 엔진인 DPU를 사용하여 Zynq UltraScale+ 보드에서 추론해보고, DPU 아키텍처에 따른 결과를 비교·분석했다. 병렬처리 수준을 높게 한 DPU는 그렇지 않은 DPU보다 소비전력 및 자원 사용량이 3배 이상 높았지만, 1.65배 좋은 성능을 보여 Trade-off 관계를 확인할 수 있었다.

1. 서론

최근 인공지능을 활용한 애플리케이션에 대한 연구가 각 분야에서 활발하게 진행되고 있다. 특히 이미지 인식에 뛰어난 성능을 보이는 CNN(Convolutional Neural Network)가 주목을 받으며, 이것을 사용한 스마트 보안, 로보틱스, 헬스케어 등의 지능형 IoT 애플리케이션들이 등장했다. 이것을 실제로 효과적으로 사용하기 위해서는 Edge device에 CNN 추론 엔진을 포팅하는 것이 필수적이다. 그러나 CNN은 computational cost가 매우 높아, 컴퓨팅 자원이 제한적인 IoT 환경에서 정해진 시간 안에 추론이 이루어지게 하기가 쉽지 않다. 이를 가속화하기 위해 전용 하드웨어에 대한 연구가 진행되고 있으며, Edge device에서도 인공신경망 추론을 할 수 있도록 다양한 도구들이 등장하고 있는데 그 중 대표적인 것이 Xilinx의 DPU(Deep Learning Processor Unit)[1]과 Intel의 OpenVINO[2]이다. 본 연구에서는 Xilinx UltraScale+ 보드에서 서로 다른 아키텍처의 DPU를 구현하여 이미지 데이터베이스로 잘 알려진 CIFAR-10의 이미지들을 추론한 결과를 비교·분석했다.

2. 선행 연구

컴퓨팅 파워의 기하급수적인 향상과 인터넷 보급 등으로 데이터가 폭발적으로 늘어나자 영상, 이미지, 음성 등에서 의미 있는 정보를 얻는 방법들이 연구되기 시작했다. 여러 연구 방법 중에서도 이미지 인식에 있어 사람과 비슷한 수준에까지 오른 CNN이 크게 주목받고 있다.[3] CNN은 가중치들로 이루어진 커널과 이미지에 합성곱 연산을 해서 이미지를 특정 카테고리로 분류한다. 각 이미지 픽셀마다 다른 가중치를 필요로 하는 완전연결 신경망과 달리 CNN의 Convolution 계층은 여러 개의 가중치로 이루어진 하나의 커널을 다른 픽셀들과 공유한다. 학습해야 하는 파라미터 수가 줄어듦에 따라 더 깊은 네트워크로 학습이 가능해져 성능을 더욱 높일 수 있다.[4] 그럼에도 불구하고 여전히 연산량이 많고 학습 및 추론 소요시간이 길어 CNN 추론 단계를 가속화하는 전용 하드웨어의 필요성이 제기되었다. GEMM Transformation, Winograd Transform, FFT와 같은 알고리즘 최적화, Datapath 최적화, Quantization 등을 이용한 CNN 모델 최적화 등의 관점에서 하드웨어 가속기에 대한 연구가 진행되어 왔으며, 최근에는 하드웨어에 대한 깊은 지식이 없는 사람들도 손쉽게 하드웨어 가속기를 구현할 수 있도록 Xilinx DPU와 Intel의 OpenVINO를 비롯한 툴들이 발표되었다.

3. Xilinx DPU 및 DNNDK

DPU란 CNN 추론에 최적화된 전용 프로세서 유닛으로서, Xilinx 사의 Zynq-7000 또는 Zynq UltraScale+ 보드의 PL에 구현하여 사용할 수 있다. 프로그램이 가능하

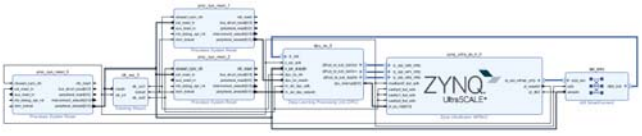
이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2019-0-00533, 컴퓨터 프로세서의 구조적 보안 취약점 검증 및 공격 탐지대응)

여 Convolution 계층의 커널 크기, stride, padding, 그리고 Max pooling, Fully connected 계층에서의 파라미터들까지 모두 직접 설정해서 원하는 네트워크의 구조를 쉽게 만들어낼 수 있다. 성능을 높이기 위해 On-chip 메모리를 가지고 있으며 이 안의 데이터를 최대한 재사용하여 메모리 대역폭을 줄인다. 또한, DPU의 Processing Elements(PE)들은 파이프라인 구조로 이루어져 있어 계산 효율이 높다.[5]

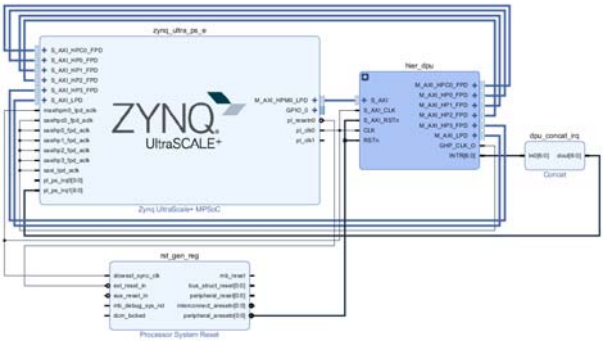
Xilinx에서는 DPU를 편리하게 사용할 수 있도록 DNNDK(Deep Neural Network Development Kit)라는 프레임워크를 함께 제공하고 있다. DECENT Compression tool, DNNC 컴파일러, DNNAS 어셈블러/시뮬레이터와 프로파일러를 포함하는 Full-stack SDK이다. C/C++ 기반 프로그래밍 API를 제공함으로써 로우레벨 아키텍처에 대한 지식이 많지 않은 사용자들도 이 툴을 사용해서 하드웨어 가속화된 딥러닝 추론 애플리케이션을 쉽게 개발할 수 있다. DNNC 컴파일러로 Tensorflow 또는 Caffe 모델을 컴파일하면 output으로 ELF 포맷의 DPU 커널이 나오는데, 여기에 DPU 전용 명령어들과 CNN 모델의 파라미터들이 담겨 있다. DNNC는 DPU로 실행할 수 있는 부분을 DPU 커널로 만들고 그렇지 못한 부분에 대한 정보도 보여주는데, 이 부분은 사용자가 C/C++를 사용해서 직접 구현해야 한다. 따라서 전체적인 연산의 Flow가 DPU, CPU를 넘나들게 된다.[6]

4. 실험 및 결과 분석

본 연구에서는 Zynq UltraScale+ 보드에 서로 다른 아키텍처의 DPU IP를 추가하여 CIFAR-10 이미지 데이터베이스의 테스트 이미지 10,000개를 추론한 결과를 분석했다. DPU IP는 아키텍처에 따라 B1152, B2304, B4096 등으로 나뉘는데, 아키텍처마다 병렬처리 수준이 다르다. DPU의 Convolution unit은 한 클럭에 여러 pixel을 동시에 처리할 수 있으며, input channel, output channel에서도 병렬처리가 이루어진다. 이 연구에서 사용한 B1152는 한 클럭당 4 pixel까지, 그리고 input, output 채널에서 각각 12개까지 병렬처리할 수 있다. 비교군으로 사용한 B4096은 Pixel parallelism이 8, input/output channel parallelism이 16으로 B1152보다 병렬처리 수준이 더 높다. 이 실험에서는 Ultra96 보드에 B1152를, ZCU102에 B4096을 추가해 비교하였다.[7] Ultra96가 ZCU102보다 가용 자원이 더 적으므로, Ultra96에 병렬처리 수준이 낮은 DPU를, ZCU102에 병렬처리 수준이 높은 DPU를 구현해 결과의 차이를 극대화해 보이고자 했다. 양 보드에 구현한 DPU IP는 모두 1.4.0 버전을 사용하였으며, 각 DPU IP를 추가한 하드웨어 블록 다이어그램과 자원 사용량은 (그림 1, 2), <표 1>과 같다. DPU IP는 PL에 추가하며, PS와 연결된다. DPU에는 input 이미지, output, 임시 정보들을 저장하기 위한 On-chip 메모리 공간이 있으나 Vivado 2018.2의 블록 다이어그램에서는 확인할 수 없다.



(그림 1) DPU IP를 추가한 Ultra96의 하드웨어 구조

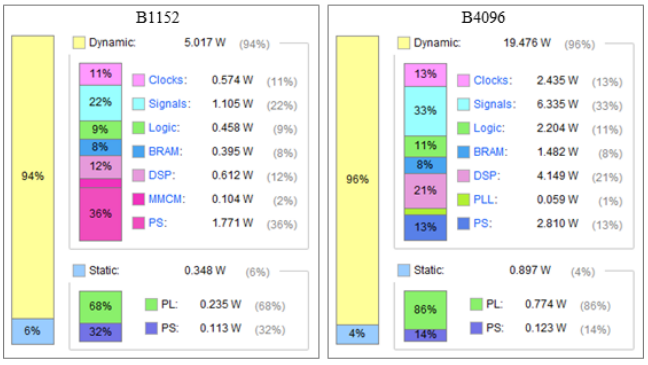


(그림 2) DPU IP를 추가한 ZCU102의 하드웨어 구조

Resource	B1152 on Ultra96	B4096 on ZCU102
LUT	32,736	111,952
FF	51,210	210,424
BRAM	123	519
DSP	211	1,399

<표 1> DPU 아키텍처에 따른 H/W 자원 사용량

완성된 하드웨어 분석 결과, B1152를 추가한 Ultra96에 비해 B4096을 추가한 ZCU102의 하드웨어 자원 사용량이 전체적으로 3배 이상 높았다. 특히 On-chip 메모리인 BRAM 블록은 4배 이상 많았으며, DSP는 6.6배에 달했다.<표 1> 소비전력에 있어서는, B1152를 구현한 하드웨어가 Dynamic power 5.017W, Static power 0.348W로 총 5.365W를, B4096을 구현한 하드웨어가 Dynamic power 19.476W, Static power 0.897W로 총 20.373W의 전력을 소비한다.(그림 3)



(그림 3) B1152와 B4096을 올린 하드웨어의 소비전력

각 보드를 MicroUSB 케이블로 같은 PC에 연결해서 CIFAR-10 이미지 데이터베이스의 테스트 이미지 10,000장을 추론한 결과, B4096은 272.865초, B1152는 450.525초가 걸렸다. B4096은 초당 36.6482개의 이미지를, B1152는 22.1963개의 이미지를 추론하는 셈이다. B4096이 파워를 3.8배 이상, 하드웨어 자원을 3배가량 더 쓰는 대신, 1.65배 더 좋은 성능을 보였다. B4096을 올린 ZCU102에서 연산에 필요한 자원들을 더 많이 사용할 뿐만 아니라 연산에 필요한 파라미터 등을 저장해둘 On-chip 메모리 용량이 4배 이상 크기 때문에 계산이 빠르다. 또한, B4096의 더 높은 병렬처리 수준이 Peak operation 수에 기여한 것으로 보인다.

5. 결론

Xilinx DPU를 사용하여 하드웨어 구조에 대한 큰 고민 없이 좋은 성능의 CNN 추론 엔진을 구현해볼 수 있었다. 하드웨어에 대한 깊은 이해와 상당한 개발 시간이 요구되었던 CNN 하드웨어 가속화가 하드웨어 설계에 대한 지식이 많지 않은 사람에게도 가능한 선택지가 되어 앞으로 다양한 분야에서 CNN 추론 엔진이 구현되어 사용될 수 있을 것으로 보인다. 또한, 실험을 통해 DPU 아키텍처에 따라 하드웨어 자원 사용량, 소비전력 및 성능 사이의 Trade-off 관계를 확인할 수 있었다. 추론 엔진 설계 시 이러한 Trade-off 관계를 고려하여 애플리케이션에 맞는 아키텍처를 사용해야 한다. 실시간 Throughput이 중요한 애플리케이션에서는 파워를 많이 쓰더라도 성능이 좋은 DPU를, Edge device의 크기가 제한적인 경우 B1152 또는 병렬처리 수준을 더 낮게 설정한 DPU 아키텍처를 쓰는 것이 적절할 것이다.

참고문헌

[1] “Edge AI Platform.” Xilinx, https://www.xilinx.com/products/design-tools/ai-inference/edge-ai-platform.html.

[2] “Intel® Distribution of OpenVINO Toolkit.” Intel, https://software.intel.com/en-us/openvino-toolkit.

[3] Kamel Abdelouahab, et al. “Accelerating CNN inference on FPGAs: A Survey.” January 2018.

[4] Habibi A., Jahani H. “Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification.” Springer International Publishing, 2017.

[5] Xilinx. DPU for Convolutional Neural Network: DPU IP Product Guide(PG338), v3.0. 2019. [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/dpu/v3_0/pg338-dpu.pdf

[6] Xilinx. DNNDK User Guide(UG1327), v1.5. 2019. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug1327-dnndk-user-guide.pdf

[7] Xilinx. Xilinx/Edge-AI-Platform-Tutorials. 2019. https://github.com/Xilinx/Edge-AI-Platform-Tutorials.