FPGA기반 이진 신경망 연산 가속 시스템 경량 설계 및 구현

이준호, 박종욱, 윤동욱, 김호원*

*부산대학교

junho@islab.re.kr, jonguk@islab.re.kr, dongwook@islab.re.kr, *howonkim@pusan.ac.kr

FPGA-based Binarized Neural Network Acceleration System Lightweight Design and Implementation

Lee Jun Ho*, Park Jong Uk*, Yun Dong Wook*, Kim Ho Won*

*Pusan National Univ.

요 약

DNN은 보안 금융 음향 의료 뿐만아니라 IoT 서비스와 접목된 AIoT 까지 다양한 분야로 확장되고 있다. 하지만 AIoT 디바이스는 한정된 자원을 갖고있어 CNN(Convolutional Neural Network)와 같이 모델의 파라미터가 차지하는 메모리 공간이 크고 부동소수점 연산이 많은 모델을 추론하기에는 어려움이 있다. BNN(Binarized Neural Network)은 모델의 파라미터가 차지하는 메모리 공간을 최소화 할 수 있기 때문에 저사양 임베디드 디바이스에서 구현하기에 적합하다. 이에 본 논문에서는 저사양 임베디드 디바이스에서도 원활한 AI 모델의 추론을 위해 FPGA에 기반한 이진 신경망의 연산 가속 시스템을 경량 설계 및 구현하였고, Avnet Ultra96 v2 FPGA Board에서 구현 결과 CIFAR-10 모델에서 최대 17%의 하드웨어 자원을 사용하였으며, 10,000장의 테스트 데이터를 추론 하였을 때 약 82.42%의 추론 정확도를 얻을 수 있었다.

Ⅰ. 서 론

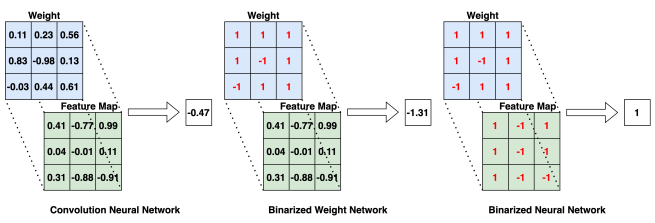
DNN(Deep Neural Network)은 보안, 금융, 음향, 의료 뿐만 아니라 IoT 서비스와 접목된 AIoT(Artificial Intelligence of Things) 까지 다양한 분야로 확장되고 있다. 하지만 AIoT 디바이스는 한정된 자원을 갖고 있어 CNN(Convolutional Neural Network)과 같이 모델의 파라미터가 차지하는 메모리 공간이 크고 부동소수점 연산이 많은 모델을 추론하기에는 어려움이 있다. 이때 모델의 경량화를 위해 모델 파라미터 및 레이어 중간 값에 대한 양자화(Quantization)를 수행한다. BNN(Binarized Neural Network)은 이러한 양자화 기법을 기반으로 연구된 신경망이며, 모델 파라미터가 차지하는 메모리 공간을 최소화 할 수 있기 때문에 저사양 임베디드 디바이스에서 구현하기 적합하다. 이에 본 논문에서는 FPGA에 기반한 이진 신경망의 연산 가속 시스템을 경량 설계 및 구현 한다.

Ⅱ. 본론

본 절에서는 BNN(Binarized Neural Network)의 배경지식과 제안하는 가속기의 구조 및 구현결과를 설명한다.

2.1. 배경 지식

BNN[1]은 기존의 CNN의 Feature Map, Weight의 저장 공간 최소화를 위해 이진화(Binarization)하여 저장하는 신경망이다. [그림 1]과 같이 이진화하는 대상에 따라 BNN, BWN(Binarized Weight Network)으로 구분할 수 있다.



[그림 1] 양자화 대상에 따른 CNN, BWN, BNN의 구분

이러한 이진화 기법은 학습 단계에서부터 적용할 수 있으며 Deterministic과 Stochastic으로 구분할 수 있다. Deterministic의 경우 입력 값이 0보다 크거나 같을 경우엔 +1, 입력 값이 0보다 작을 경우에는 -1로 이진화하는 방법이고, Stochastic의 경우 입력 값이 1 이상일 경우는 +1, 입력 값이 -1 이하일 경우는 -1로 결과 값이 출력되지만 그사이 값일 경우에는 확률적으로 값을 정하게 되는데, 이때 난수발생 과정이 포함되어 있다. Deterministic 이진화는 하드웨어에서 입력 값의 부호 비트(Sign Bit)만 가지고 출력 값을 결정할 수 있으므로 경량 연산 구조에 적합하다. 추가적으로 -1, +1로 구성되어 있는 Feature Map과 Weight를 0, +1의 값으로 치환하여 연산하는 구조도 제시되고 있으며 해당 구조는 0, +1의 값만 가지게 되므로 하드웨어에서 1bit의 크기를 가지고 표현할 수 있게 된다[2]. 본 연구에서도 -1, +1의 값을 0, +1로 치환하여 연산하는 구조에 기반하여 가속기를 구현하였다.

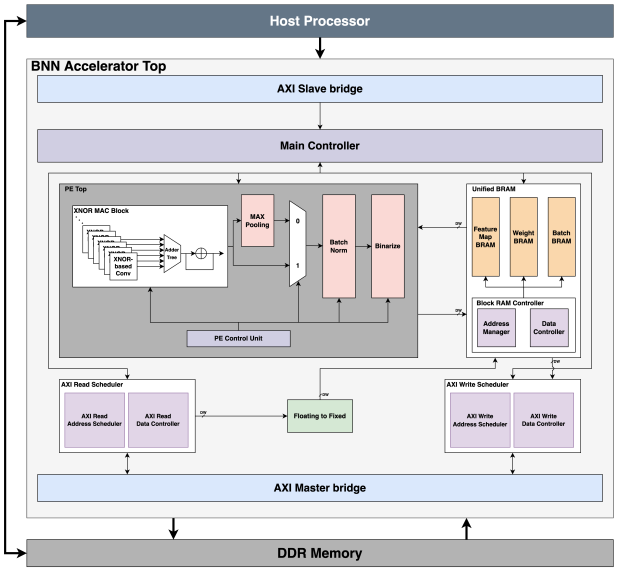
2.2 제안 구조

본 연구에서는 CIFAR-10 데이터 셋 기반 BNN 연산 가속 시스템을 구현하였고, 학습 및 추론을 위한 ConvNet 구조는 [표 1]과 같다. *Cin*, *Win*, *Cout*, *Wout*, Weight와 Input은 각 레이어의 입력 및 출력 채널 수와 크기 및 Weight와 Feature Map의 bit수를 나타낸 것이다. 이때 첫 번째 레이어는 32비트 고정소수점(Fixed Point)으로 Image를 입력 받기 때문에 해당 Bit Width만큼 곱해진 형태로 계산된다.

[표 1] BNN 학습 및 추론을 위한 ConvNet

Layer	C_{in}	C_{out}	W_{in}	W_{out}	Weight	Input
Conv1	3	128	32	32	3.38K	96K
Conv2	128	128	32	16	144K	128K
Conv3	128	256	16	16	288K	32K
Conv4	256	256	16	8	576K	64K
Conv5	256	512	8	8	1.13M	16K
Conv6	512	512	8	4	2.25M	32K
FC1	8,192	1,024	1	1	8M	8K
FC2	1,024	1,024	1	1	1M	1K
FC3	1,024	10	1	1	10K	1K

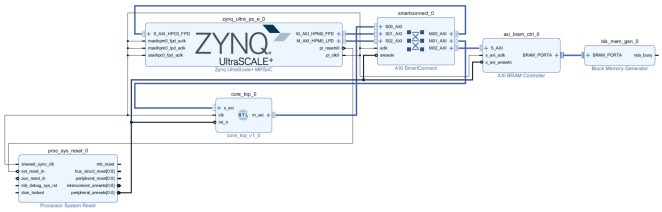
[그림 2]는 본 연구에서 제안하는 경량 BNN 가속기의 전체 구조를 나타낸 그림이다. 가속기 내부에는 고정소수점(Fixed Point) 기반 연산을 수행하기 때문에, 부동소수점(Floating Point)을 고정소수점으로 변환 하기위한 모듈이 존재한다. 그리고 Weight, Batch Normalization, Layer 결과 값을 저장하는 BRAM(Block RAM)과 데이터 및 주소를 컨트롤하기 위한 BRAM Controller가 존재한다. 또한 Convolution, Max pooling 등 Layer 연산을 수행하기 위한 PE(Processing Element)가 있고, AXI Interface를 이용하여 데이터의 통신이 이루어지므로 AXI Slave, Master Bridge와 이것을 제어하기 위한 별도의 Scheduler가 존재한다.



[그림 2] 제안하는 BNN 가속 시스템의 전체 구조

2.3 구현 및 구현 결과

경량 BNN 가속 시스템 구현을 위해 본 연구에서는 Xilinx Zynq Ultrascale+ ZU3EG-A484 FPGA 칩이 내장된 Avnet Ultra96 v2 Board를 사용한다. Host로는 Ultra 96에서 지원하는 Petalinux 기반 PS(Processing System)를 사용하였고, [그림 3]과 같이 가속기와 PS 간 통신을 위해 AXI Interface를 구성하였다.



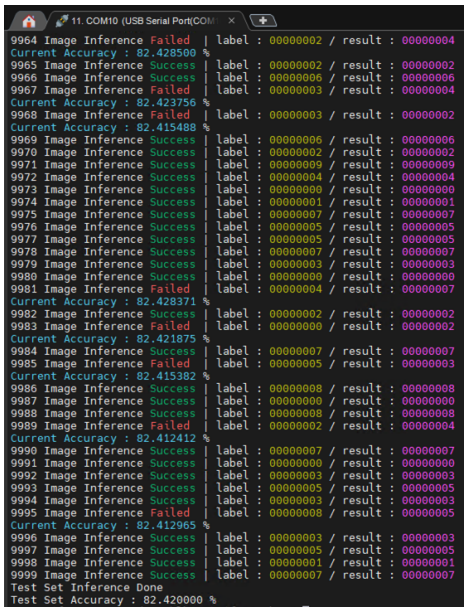
[그림 3] BNN 가속 시스템 구현 디자인

먼저 Host Processor에서는 Model의 Weight, Batch Normalization Parameter, Image Data 등 파라미터들을 DDR Memory에 Write 한 후, AXI Interface를 이용하여 가속기 내부 CSR(Control Status Register)에 파라미터들의 Base Address와 Start 신호를 Write 한다. 가속기에서는 CSR 값에 따라 추론을 위한 데이터들을 DDR Memory에서 Read 한 뒤 추론을 시작하며 출력을 DDR Memory에 Write 한다. Host processor에서는 해당 값을 읽고 후처리하여 최종 결과를 출력하며 [표 2]는 이에따른 자원 사용량을 나타낸 것이다.

[표 2] 경량 BNN 가속기 구현에 따른 자원 사용량

	LUT	Register	BRAM	DSP
Utilized	12,580	3,300	40.5	32
Available	70,560	141,120	216	360
Percentage	17.82%	2.34%	18.75%	8.89%

또한 [그림 4]와 같이 CIFAR-10 데이터셋의 10,000장의 테스트 셋을 이용하여 추론하였을 때 82.42%의 정확도를 얻은 것을 확인하였다.



[그림 4] CIFAR-10 테스트 셋 추론 결과

Ⅲ. 결론

본 연구에서는 저사양 임베디드 디바이스에서도 원활한 구성이 가능하도록 BNN 저면적화 연산 가속기를 설계 및 구현 하였다. 또한 파이프라인 구조를 적용하였기 때문에 처리량 측면에서 이점을 얻을 수 있었다. 구현에 따른 합성 결과 하드웨어 자원은 최대 17% 가량 사용되었으며 저사양 FPGA에서도 충분히 구현이 가능할 것으로 보인다. 온 보드 검증 결과 CIFAR-10 데이터 셋 기준 10,000장의 테스트 이미지에 대해 반복 추론을 한 결과 82.42%의 추론 정확도를 얻을 수 있었다. 모델 구조 수정을 통한 정확도 개선, 고정 소수점 변환 오차 최소화, 레이어 중간 값에 대한 Bit Width 최적화 시 가속기의 자원 사용량을 보다 더 줄일 수 있을 것으로 보인다.

ACKNOWLEDGMENT

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00903, 고신뢰 온-디바이스 딥러닝 가속기 설계를 위한 물리채널 기반 취약점 검증 및 대응기술 개발)

참 고 문 헌

[1] Courbariaux, Matthieu, et al. "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1." arXiv preprint arXiv:1602.02830 (2016).

[2] RASTEGARI, Mohammad, et al. Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision. Springer, Cham, 2016. p. 525-542.