시스톨릭 어레이를 위한 저전력 희소 데이터 프로세싱 유닛 설계

박주동, 공준호
경북대학교 전자전기공학부

bagjd24@knu.ac.kr, joonho.kong@knu.ac.kr

Design of Low-Power Sparse Data Processing Unit for Systolic Array

Judong Park, Joonho Kong
School of Electronic and Electrical Engineering, Kyungpook National University

요 약

최근 인공지능 애플리케이션이 많이 사용되고 이러한 애플리케이션에서 데이터 희소성이 높아지고 있어 이러한 희소 데이터를 효율적으로 처리하기 위한 하드웨어 구조들이 많이 소개되고 있다. 본 논문에서는 희소 데이터 처리 시 전력 소모량을 낮출 수 있는 새로운 하드웨어 구조를 제안한다. 일반적인 인공지능 하드웨어에서 많이 사용되는 시스톨릭 어레이 구조를 기반으로 하며, 제안된 저전력 PE 가 희소 데이터 처리시 희소하지 않은 데이터 처리 시보다 최대 2 배의 전력 소모량을 줄일 수 있는 것으로 나타났다.

1. 서론

최근 컴퓨터 비전, 머신 러닝 등의 애플리케이션이 많이 사용되고 있다. 컴퓨팅 성능의 향상과 알고리즘의 지속적 발전으로 인해 심층신경망 (deep neural network: DNN) 이 인공지능 애플리케이션의 핵심적 역할을 담당하고 있다. 이러한 연산의 핵심은 행렬 곱셈 연산으로써 행렬 곱셈을 효율적으로 수행하기 위한 하드웨어의 필요성이 대두되었다. Google Tensor Processing Unit (TPU) [1] 등은 이러한 필요에 의해 나온 하드웨어로 실제 Google 데이터센터에서 사용되고 있으며, 딥러닝 연산을 수행함에 있어 기존 GPU 대비 더 나은 전력당 성능을 보여주고 있다.

심층신경망 연산에 있어 최근 데이터의 희소성 (sparsity)에 대한 고려가 매우 중요한 요소가 되었다. 특히, 딥러닝 중 많이 사용되는 합성곱신경망 (convolutional neural network: CNN)의 경우 Rectified Linear Unit(ReLU) 활성화 함수로 인한 피쳐맵에서의 희소성과 가중치 가지치기 (weight pruning)으로 인한 가중치 값에서의 희소성으로 인해 매우 높은 데이터 희소성을 보여주고 있다. 이처럼, 일반적인 심층신경망 연산에서 희소 행렬 연산이 매우 빈번하게 일어날 수 있으며 [2], 희소행렬 연산에서 0 값을 곱하거나 더하는 연산은 수행 후 결과가 0 이 되거나 값이 변하지 않으므로 연산 자체가 무효한 (ineffectual) 경우가 많다. 이 경우 무효한 연산을 수행하지 않고 건너뛰거나 생략할 수 있다면 이에 소요되는 전력을 줄일 수 있다.

본 논문에서는 심층신경망 연산에 사용될 수 있는 시스톨릭 어레이 구조에 사용되는 프로세싱 요소 (processing element: PE) 구조를 변형하여 희소 값 연산 시에 전력 소모량을 줄인 새로운 PE 구조를 제안한다. 본 논문에서는 이를 위해 기존의 PE 구조에서 새롭게 멀티플렉서 (MUX) 및 비교기 등을 추가하여 희소 값의 연산 시 동적 전력 소모량을 줄일 수 있게끔 하였다. 제안된 PE 구조는 실험 결과 희소하지 않은 값을 연산할 때보다 희소 값 연산 시에 전력 소모량을 상당량 절감할 수 있다는 것을 보여주었다.

2. 저전력 프로세싱 요소 구조 및 설계

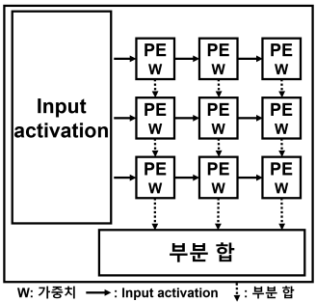

그림 1 weight stationary 데이터 플로우

CNN 의 convolution 연산을 위해서는 곱셈과 덧셈 연산이 필요하다 [3]. 본 논문에서는 그림 1 처럼 곱셈과 덧셈 연산을 수행할 수 있는 2 차원 배열 모양의 PE 들로 이루어진 시스톨릭 어레이를 이용하여 곱셈 및 덧셈 연산을 수행한다

Weight stationary 데이터 플로우 방식[3]은 각 PE 들에 입력되는 가중치 값이 외부 메모리로부터 내부 레지스터로 로드되는 횟수를 줄이고 가중치 값의 PE 내 재사용성을 높이고자 하는 방식이다. 기존 CNN 구조에서는 이러한 방식이 유리할 수 있는데 이는 가중치를 공유하는 CNN 의 연산 방식에 기인한다. 본 논문에서는 weight stationary 방식을 기본 데이터 플로우로 사용한다.

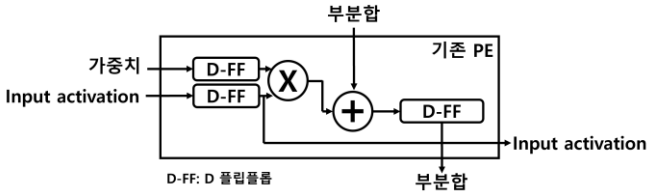

그림 2 기존 PE 구조

그림 2 는 기존의 weight stationary 방식 PE 구조를 나타낸 것이다. 시스톨릭 어레이를 구성하는 PE 는 먼저 곱셈부에 입력된 가중치와 input activation 값을 곱한다. 그 다음으로, 그림과 같이 입력되는 부분 합 (partial sum)과 이전 곱셈 결과를 더하여 또 다른 부분 합을 출력한다.

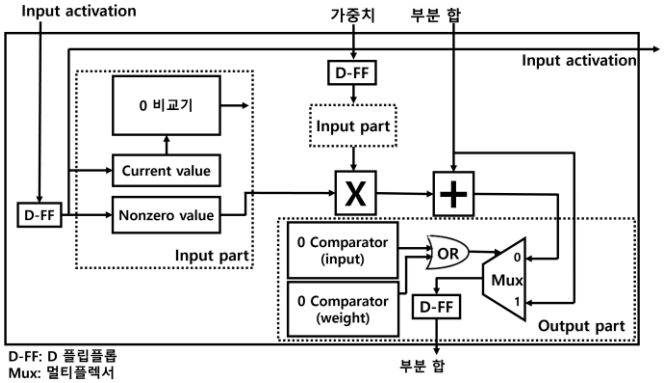

그림 3 설계한 저전력 PE 구조

본 논문에서 제안하는 저전력 PE (그림 3)는 0 값 비교기를 추가하여 input, 가중치 포트에 0 이 들어오는 경우 값을 그대로 유지하여 곱셈기에 게이트 스위칭이 일어나지 않도록 하며, 이는 동적 전력 소모량을 줄이도록 설계되었다. Output 출력 부분은 input 또는 가중치에 0 값이 입력되는 경우 덧셈기를 우회하여 출력이 바로 아웃풋 포트로 전해질 수 있도록 설계하였다.

3. 전력 소모량 측정 방법

각 PE 가 소모하는 동적 전력을 측정하기 위해 Synopsys 사의 Design Compiler [4]와 VCS [5]를 사용하였다. Design Compiler 는 Register Transfer Level(RTL) 코드를 특정 타깃의 라이브러리에 맞는 Gate-Level Netlist 로 합성하는 툴이다. Design Compiler 에서 로직 합성 후 평균 전력 소모량을 측정할 수 있다 [4]. VCS 는 테스트벤치, 합성한 코드를 활용하여 시뮬레이션을 수행하며, 스위칭 활동을 기록하는 데 사용된다 [5].

Design Compiler 와 VCS 를 활용하여 동적 전력 소모량을 측정하는 과정을 그림 4 에 정리하였다. 시뮬레이션 하는 동안 내부 스위칭 변화 기록에 필요한 switching activity interchange format (saif) 파일을 이용하여, 테스트벤치의 입력에 따른 스위칭 활동을 기록하고 Design Compiler 를 통해 해당 파일을 읽어 동적 전력 소모량을 측정할 수 있다. 이러한 측정 방법론은 [6]에서 제안된 방법과 유사하다.

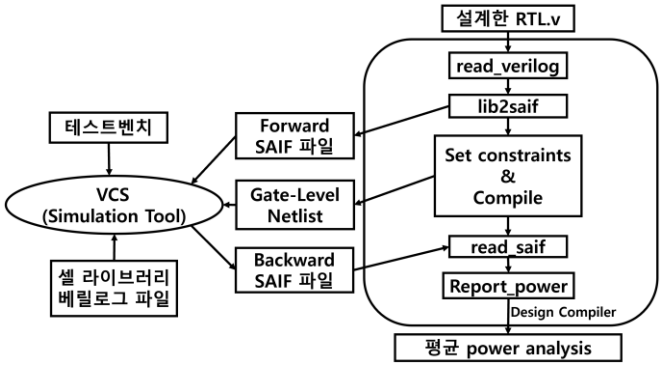

그림 4 전력 소모량 측정 과정[6]

4. 실험을 위한 환경 및 파라미터 설정

실험을 위해 본 논문에서 제안한 PE 구조를 Verilog hardware description language (HDL)로 구현하였으며, 시뮬레이션을 위한 테스트벤치도 Verilog HDL 로 구현하였다. 로직 합성 시에 타깃 클럭 주파수는 100MHz 로 설정하였다. 로직 합성 시의 라이브러리는 Synopsys 사에서 제공하는 32nm 공정 라이브러리를 사용하였다.

시스톨릭 어레이에서는 PE 를 격자 구조로 배치하여 사용하는 것이 일반적이지만 본 논문에서는 하나의 PE 에 대해서만 전력 소모량을 측정하였다. 인풋 값과 가중치 값은 8-비트 정수형 값을 사용하고 부분합은 24-비트를 담을 수 있도록 설계하였다.

5. 실험 결과

이전 입력	현재 입력	평균 Dynamic Power
Sparse	Sparse	32.9893 ㎼
Sparse	Dense	36.4193 ㎼
Dense	Sparse	39.9158 ㎼
Dense	Dense	67.3100 ㎼

표 1 입력에 따른 동적 전력 소모량

표 1 은 PE 가 소모하는 동적 전력을 4 가지 경우로 나누어서 보여준다. Sparse 입력의 경우, 가중치, 인풋, 부분 합 모두 0 이 사용되는 경우이고, dense 입력의 경우, 테스트벤치 상에서 Verilog 커맨드인 $urandom 을 사용하여 위의 3 개의 입력에 0 이 아닌 임의의 값을 입력하였을 경우를 의미한다. 현재 입력과 이전 입력이 같은 경우, 1000 번의 임의의 값을 입력한 평균값 사용했고 현재 입력과 이전 입력이 다른 경우, 1 번의 임의값을 입력한 평균값을 사용했다.

실험 결과에서 보이듯이, dense-dense (이전 입력-현재 입력)의 경우 소모하는 전력이 제일 높은 것을 볼 수 있으며, 이는 sparse-sparse 경우 대비 2.0 배 높은 전력 소모량을 보인다. 반면 sparse 값이 이전이나 이후에 포함되는 경우 전력 소모량이 dense-dense 경우와 비교해 최대 45.8%의 전력 소모량 감소를 보였다. 실험 결과에 비추어 볼 때 본 논문에서 제안된 PE 구조가 sparse 데이터 처리 시 전력 소모량을 줄일 수 있음을 확인할 수 있다.

6. 결론

본 논문에서는 희소 데이터를 저전력으로 처리하는 시스톨릭 어레이에서 활용될 수 있는 저전력 PE 구조를 제안하였다. 설계된 PE 구조는 비교기를 통하여 0 값을 판별하며, MUX 를 통해 값을 우회시켜 전력 소모량을 줄인다. 우리는 새로운 PE 구조를 Verilog HDL 로 구현하였고, Design Compiler 를 통해 합성하였으며, 희소하지 않은 데이터 처리의 경우 대비 최대 2 배의 전력 소모량 감소 효과를 확인할 수 있었다.

7. 사사

이 논문은 2022 년도 정부(교육부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임 (No. 2021R1I1A3A04037455).

참고문헌

[1] Jouppi, Norman P., et al. "In-datacenter performance analysis of a tensor processing unit." Proceedings of the 44th annual international symposium on computer architecture. 2017. pp.1-12

[2] Kim, Dongyoung, Junwhan Ahn, and Sungjoo Yoo. "Zena: Zero-aware neural network accelerator." IEEE Design & Test 35.1. 2017. pp. 39-46.

[3] Sze, Vivienne, et al. "Efficient processing of deep neural networks: A tutorial and survey." Proceedings of the IEEE 105.12(2017), pp.2295-2329, 2017

[4] Synopsys. “Power Compiler User Guide”. Synopsys. 2016

[5] Synopsys. “VCS/VCSi User Guide”. Synopsys. 2016

[6] 이동건 (Donggeon Lee), 추상호 (Sangho Chu), 김슬아 (Seul-a Kim),and 김호원 (Howon Kim). "SHA-3 후보들의 H/W 구현에 대한 전력 소모량 추정." 한국정보처리학회 학술대회논문집 17.2. pp:1183-1185. 2010