AI 가속기 설계 영역 탐색에 대한 연구

이동주 [1], 백윤흥 [1]
[1] 서울대학교 전기정보공학부, 반도체공동연구소

djlee@sor.snu.ac.kr, ypaek@snu.ac.kr

A Study on Design Space Exploration on AI accelerator

Dong-Ju Lee[1], Yun-Heung Paek[1]
[1]Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center(ISRC), Seoul National University

요 약

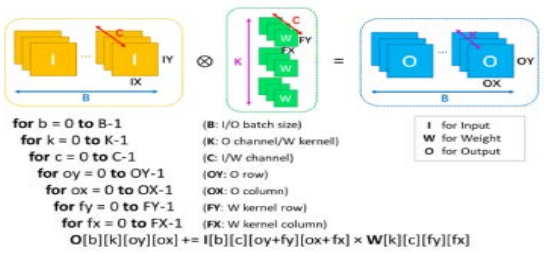
AI 가속기는 머신 러닝 및 딥 러닝을 포함한 인공 지능 및 기계 학습 응용 프로그램의 연산을 더 빠르게 수행하도록 설계된 일종의 하드웨어 가속기 또는 컴퓨터 시스템이다. 가속기를 설계하기 위해선 설계 영역 탐색(Design Space Exploration)을 하여야 하고 여러 인공지능 중에서도 합성 곱 신경망(CNN)에 대한 설계 영역 탐색을 소개한다.

1. 서론

딥 러닝 알고리즘은 일종의 인공 지능 알고리즘으로서 현재 패턴인식, 데이터 마이닝을 이용한 다양한 분야에 활용되고 있다. 최근 딥 러닝 모델은 점점 발전되어 왔고 그 정확도가 늘어나면서 깊이나 복잡도는 늘어나게 되었다. 이러한 복잡한 모델을 한정된 하드웨어 크기와 자원 안에서 실행하기 위하여 AI 가속기에 대한 연구가 활발하게 진행되고 있다. 가속기를 설계하기 위해서 설계 영역 탐색(Design Space Exploration)이 이루어져야 하는데 그 방법을 소개한다.

2. 합성 곱 신경망의 코드 구조

딥 러닝 알고리즘에는 여러 가지 종류가 있는데 그 중에서도 합성 곱 신경망의 합성 곱 층에 대한 코드는 그림 1 과 같이 7 개의 반복문으로 구성되어 있다. 반복문이라는 특성에 의해 for 문은 순서가 뒤바뀌어도 같은 연산을 수행하게 된다.



(그림 1) 합성 곱 층 구조[1].

3. 설계 영역 탐색

설계 영역 탐색은 합성 곱 신경망의 코드를 어떻게 하드웨어에 대입하여 실행하여야 최적의 효율이 나올지를 탐색하는 것이다. 먼저 입력으로 합성 곱 신경망의 구조, 하드웨어 제약조건 (메모리, 연산 칩 배열 구조 등) 이 주어지면 그에 맞는 연산 수행 순서, 코드 배열 방법 등이 출력으로 나오게 된다.

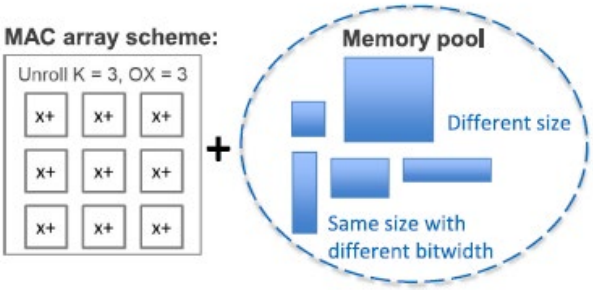
3-1 단층 설계 영역 탐색[1]

연산 수행 순서는 7 개의 각 반복문 고리의 모든 순열 구조를 탐색하게 된다. 또 가장 안쪽에 반복문에 있는 코드는 하드웨어에서 직접적으로 실행되는 코드인데 이를 하드웨어 연산칩의 활용도를 최대화하는 방향으로 연산을 골고루 실행할 수 있도록 확장한다. 하지만 모든 영역을 탐색하는 것은 매우 많은 시간과 제약이 존재하므로 가능성 없는 설계 영역을 제거함으로써 효율적으로 영역을 탐색하는 방법을 제시한다. 먼저 반복문 고리의 순서를 모두 탐색하기 위해선 7! = 5040 만큼의 경우의 수가 존재하기에 효율적인 탐색이 필요하다. 7 개의 반복문은 입력, 출력, 커널 데이터의 입장에서 서로 간의 관련성이 존재한다. 데이터가 이동하게 되면 많은 시간이 소요되므로, 데이터의 이동을 최소화하는 방향으로 반복문의 순서가 부분적으로 고정되는 방향으로 탐색을 진행한다.
또 연산을 연산 칩 배열 구조에 효율적으로 배치하기 위해서 각 반복문의 크기를 소인수 분해하여 그 조합으로 최적의 활용도를 찾는다.

	B	K	C	OY	OX	FY	FX
W	✗	✓	✓	✗	✗	✓	✓
I	✓	✗	✓	?^{IY}	?^{IX}	?^{IY}	?^{IX}
O	✓	✓	✗	✓	✓	✗	✗

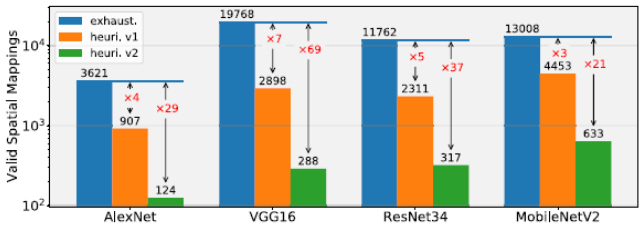
✓ relevant (r)
✗ irrelevant (ir)
? partially relevant (pr)
?$^{IX/IY}$ partially relevant to IX/IY

(그림 2) 반복 문 관계도 표[1].



(그림 3) 연산 배열 방법 예시[1].

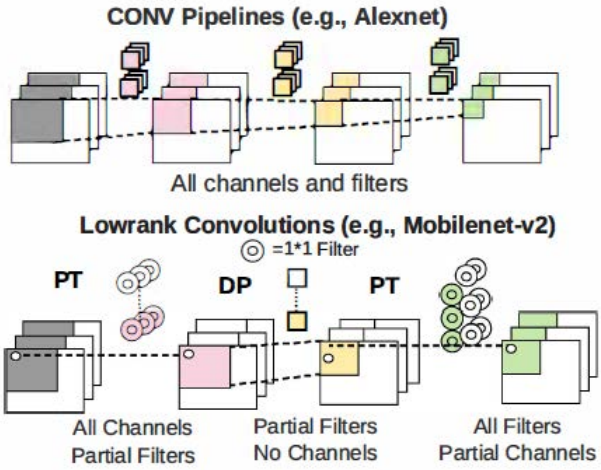
설계영역을 일부 제거하여 탐색함으로써 모든 영역을 탐색하는 것보다 탐색시간을 (그림 4)와 같이 유의미하게 단축시킬 수 있었다.



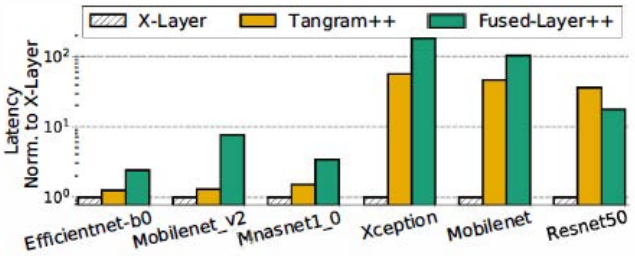
(그림 4) 탐색방법별 탐색량 비교[1]

3-2 다층 설계 영역 탐색[2]

최근에는 휴대 전자 기기 상에서 인공지능 연산을 수행하기 위해 여러 가지 딥 러닝 모델들이 개발되고 있다. MobileNet 이 대표적인 모델 중 하나이다. 이 모델의 특징은 합성 곱 연산을 최대한 줄이기 위하여 기존의 합성 곱을 변형하여 새로운 연산 방식의 합성 곱을 여러 층에 걸쳐 활용하였다는 것이다. 각 입력 채널마다 다른 커널을 한 번씩만 연산하는 깊이 방향 합성 곱(Depthwise Convolution)과 입력 채널을 한 개의 채널로 압축시켜주는 채널방향 합성 곱(Pointwise Convolution)을 연속적으로 활용하여 연산수를 줄이는 방법을 채택하였다. 3-1 에서는 한 개의 합성 곱 층에 대한 최적화 기법에 대해 다뤘는데 그 한계점으로서 특정 합성 곱(Depthwise, Pointwise Convolution)에 대해서는 연산 처리가 빠르기 때문에 메모리에 데이터가 저장되고 불러오는 시간이 데이터를 연산하는 시간보다 많이 소요된다는 단점이 있다. 이 문제를 해결하기 위해서 [2]는 이전 층의 데이터의 출력값이 다음 층의 입력 값으로 활용된다는 특성을 이용해 연산 수행 순서를 재배치하였다. (그림 5)와 같이 각 층의 연산을 부분으로 쪼개 그 데이터가 이용되는 여러 개의 층을 먼저 연산하는 방법이다.



(그림 5) 다층 연산 수행 방식



(그림 6) 다층 연산 실험결과[2]

결과적으로 앞서 소개한 MobileNet 에서 다른 유사 가속기 논문의 모델들 보다 약 1.4 배 4 배 빨라졌다.

4. 결론

인공 지능 모델이 발전하면서 그 연산의 깊이와 양이 계속해서 증가하고 있다. 증가하는 연산과 복잡성 때문에 이를 최적화하기 위해 AI 가속기에 대한 연구가 활발히 진행되고 있고 그중 설계 영역 탐색이 중요한 요소로 작용하고 있다. [1]에서는 합성 곱 신경망을 목표로 하는 설계 영역 탐색 방법을 소개하였고 효과적으로 탐색 영역을 줄여 나가는 방법을 제시하였다. 하지만 연산수를 줄이는 특정한 합성 곱을 이용한 모델에서는 메모리 측면에서 이슈가 있어 좋지 못한 성능을 보여주고 있다. 이 해결책으로 [2]는 이전 층의 데이터의 출력 값이 다음층의 입력 값으로 활용된다는 점을 이용하여 데이터가 메모리에 저장되고 불러오는 횟수를 단축하는 방법을 제시하였다.

5. ACKNOWLEDGEMENT

이 논문은 2022 년도 BK21 FOUR 정보기술 미래인재 교육연구단에 의하여 지원되었음.

참고문헌

[1] Linyan Mei, Pouya Houshmand, Vikram Jain, Sebastian Giraldo, and Marian Verhelst “ZigZag: Enlarging Joint Architecture-Mapping Design Space Exploration for DNN Accelerators” ,IEEETC, pp. 1160–1174, 2021

[2] Naveen Vedula , Reza Hojabr, Ahmad Khonsari and Arrvindh Shriraman “X-Layer: Building Composable Pipelined Dataflows for Low-Rank Convolutions”,30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Atlanta, GA, USA, 2021, pp. 103-115