합성곱과 행렬 곱셈 연산을 지원하는 효율적 연산기 설계 및 구현

정보근, 김형원*
충북대학교

bogeunjung@chungbuk.ac.kr, *hwkim@chunbuk.ac.kr

Design and Implementation of an Efficient Computational Unit Supporting Convolution and Matrix Multiplication Operations

Bo-Geun Jung, Hyung-Won Kim*
Chungbuk National University, Cheongju, Korea

요 약

본 논문에서는 합성곱 (Convolution)과 행렬 곱셈 (Matrix Multiplication) 연산을 지원하는 효율적인 연산 엔진을 제안한다. 전통적으로 합성곱 신경망 (CNN)은 지역 패턴 학습에 강점을 보이지만 전역적 문맥 처리는 어려운 반면, Transformer 는 전역적 문맥을 잘 포착하나 데이터와 계산 자원이 많이 필요하다. 제안된 연산 엔진은 3x3 Processing Element (PE) 배열을 Systolic Array 구조로 배치하여 이 두 가지 연산을 효율적으로 처리할 수 있다. 이 구조는 다양한 합성곱과 행렬 곱셈 연산을 지원하며, 데이터 전송 메커니즘 및 PE 의 구조 재구성을 통해 하드웨어 최적화를 달성한다. 결과적으로, 이 설계는 단일 구조에서 두 연산을 효과적으로 처리하여 하드웨어 자원 활용에 큰 이점을 제공한다.

Ⅰ. 서 론

딥러닝의 이미지 처리 및 비전 분야에서는 과거에 주로 합성곱 신경망 (Convolutional Neural Network)이 연구되었다. CNN 은 이미지의 로컬 패턴을 효과적으로 학습하는 데 강점이 있으나, 전역적인 정보 처리가 상대적으로 어려운 단점이 있다. 이에 반해, 최근 주목받고 있는 Vision Transformer[1]는 Self-Attention 계층을 통해 CNN 의 이러한 한계를 보완하며, 이미지의 전역적 문맥을 잘 포착할 수 있다. 그러나 Transformer 는 많은 데이터와 계산 자원이 필요하다는 단점이 존재한다. 이러한 배경에서 CNN 과 Transformer 의 장점을 결합하여 높은 정확도를 유지하면서도 경량화 된 모델들이 등장하고 있다. 예를 들어, 최신 객체 인지 모델인 YOLOv10[2]은 CNN 기반의 연산에 Transformer 구조의 일부인 Self-Attention 구조를 결합했다. 그 결과, YOLOv10-m 모델은 CNN 기반의 YOLOv8-m[3] 모델보다 파라미터 수가 40% 적으면서도 모델의 정확도 (AP^{val})가 0.5%포인트 더 높았고, Transformer 기반의 RT-DETR-R34[4] 모델보다 파라미터 수가 50.3% 적으면서도 모델의 정확도는 2.2%포인트 더 높게 나타났다[2].

CNN 에서는 대부분의 연산이 합성곱 (Convolution)으로 이루어진다. 반면, Transformer 는 합성곱 연산 대신 Self-Attention 계층에서 행렬 곱셈 (Matrix Multiplication) 연산을 사용한다. 따라서 이들을 System on a Chip (SoC), Field Programmable Gate Array (FPGA)와 같은 하드웨어 플랫폼에서 가속하기 위해서는 합성곱과 행렬 곱셈 연산 모두를 효율적으로 처리할 수 있는 연산기의 구조가 요구된다. 특히 하나의 연산기 구조에서 합성곱과 행렬 곱셈 연산 모두를 수행할 수 있다면 Area 와 Power 의 오버헤드를 크게 감소시킬 수 있다. 이를 위해 CNN 과 Transformer 의 장점을 융합하여 두 연산을 하나의 구조에서 효과적으로 처리하는 방법이 연구되고 있으며, 이는 하드웨어 최적화 측면에서도 큰 이점을 제공한다. 본 논문에서는 이러한 요구를 충족시키기 위해 합성곱과 행렬 곱셈 연산을 둘 다 지원하는 연산기를 제안한다.

Ⅱ. 본론

그림 1 은 본 논문에서 제안하는 연산기의 구조를 보여준다. 빨간색 박스는 단일 입력 채널 배열이 합성곱 연산을 수행할 때의 연산기 배치를, 파란색 박스는 행렬 곱셈 연산을 수행할 때의 연산기 배치를 보여준다. 제안하는 연산기는 합성곱 연산에 특화된 기존 3x3 Processing Element (PE) 구조를 기반으로 하며, 4 개의 단일 입력 채널 배열 (Single Input Channel Array)이 병렬로 배치되어 4 개의 입력 채널과 16 개의 출력 채널을 동시에 처리하는 4x16x3x3 PE 구조를 갖춘다.

그림 2 는 제안하는 Processing Element 의 구조를 보여준다. 인접한 PE 에 입력 데이터와 가중치를 전달하기 위해 각각의 레지스터가 있으며, 연산 결과를 누적하기 위한 누산 레지스터도 포함되어 있다. 연산은 곱셈기와 덧셈기를 통해 진행되며, 멀티플렉서의 refresh 신호를 이용해 누적된 값을 초기화 할 수 있다.

합성곱 연산에서는 PE 가 커널의 한 행을 연산한 후, 부분합을 인접한 대각선 방향의 PE 로 전달하여 다음 행의 연산을 수행한다. 이를 통해 일반적인 연산기보다

데이터 재사용을 극대화할 수 있다. 제안된 연산기는 3x3 커널 stride 1, 3x3 커널 stride 2, 그리고 1x1 커널 stride 1 의 합성곱 연산을 지원한다.

행렬 곱셈 연산의 경우, PE 의 구조를 재구성하여 단일 채널 배열이 정사각형 형태로 변환되어 12x12 PE 배열을 형성한다. 이 배열은 Systolic Array 로 확장되어 4x12x12 의 Systolic Array 구조로 연산이 수행된다. 입력 행렬은 Systolic Array 의 크기에 맞춰 12x12 단위로 분할되어 연산이 진행되며, Output Stationary 방식으로 처리된다. 이 방식에서는 각 PE 가 연산을 수행하면서 인접한 PE 와 데이터를 주고받으며, 각 PE 의 내부에서 데이터 간의 곱셈 결과를 부분합에 누적한다. 왼쪽 최상단 PE 부터 대각선 방향으로 동일한 위치에 있는 PE 들이 데이터를 동일한 순서로 누적하므로, PE_select 신호와 refresh 신호를 공유하여 연산 결과를 출력으로 전송하고, 누산 레지스터를 초기화한다.

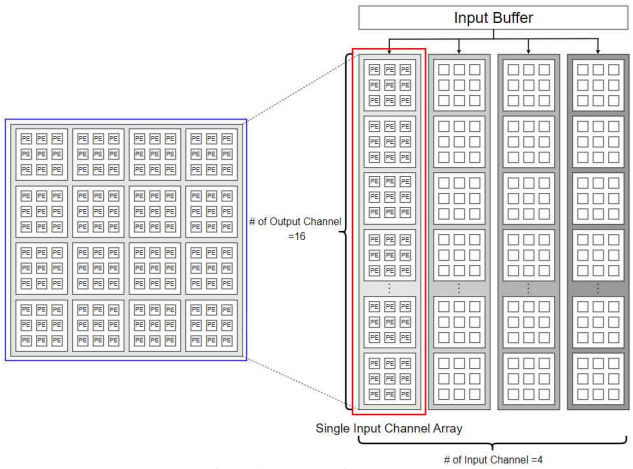

그림 1. 제안하는 연산기의 구조도

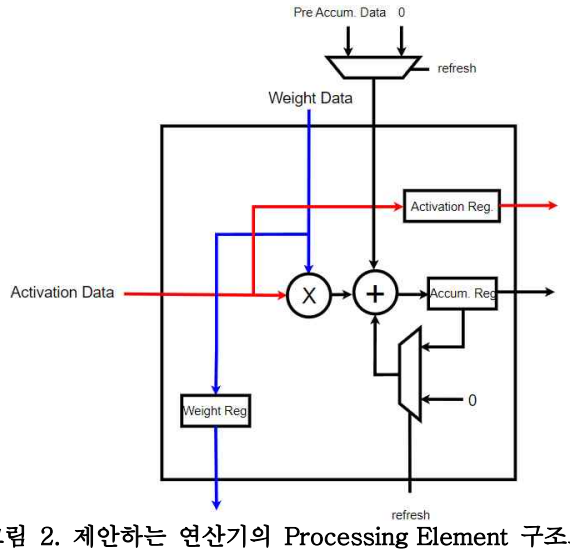

그림 2. 제안하는 연산기의 Processing Element 구조도

Ⅲ. 결론

본 논문에서는 합성곱과 행렬 곱셈 연산을 둘 다 지원하는 연산기를 제안하였다. 3x3 PE 배열을 기반으로 한 Systolic Array 구조를 활용하여 두 연산을 효율적으로 처리하고, 데이터 재사용을 극대화하며 다양한 커널과 stride 를 지원한다. 또한 레지스터와 멀티플렉서를 추가하고, 연산기 재배열 및 데이터 전송 메커니즘을 통해 하드웨어 자원 최적화를 달성하였다. 제안된 연산기는 향후 다양한 최신 모델에 효과적으로 활용될 가능성을 지닌다. 앞으로의 연구에서는 제안된 연산기의 성능을 실제 모델에 적용하고, 추가적인 최적화 방법을 탐색할 예정이다.

ACKNOWLEDGMENT

This work was supported by Regional Leading Research Center (RLRC) of the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2022R1A5A8026986) and supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-01304, Development of Self-Learnable Mobile Recursive Neural Network Processor Technology). It was also supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Communication Technology Research Center support program (IITP-2024-2020-0-01462) supervised by the IITP (Institute or Information & communications Technology Planning & Evaluation).

참 고 문 헌

[1] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

[2] Wang, Ao, et al. "Yolov10: Real-time end-to-end object detection." *arXiv preprint arXiv:2405.14458* (2024).

[3] Jocher Glenn. Yolov8. https: // github. com/ ultralytics/ ultralytics/ tree/ main , 2023

[4] Zhao, Yian, et al. "Detrs beat yolos on real-time object detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.