(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0078186
(43) 공개일자 2024년06월03일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2023.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
Z05S 12/00 (2022.08)
(21) 출원번호 10-2022-0160902
(22) 출원일자 2022년11월25일
심사청구일자 2022년11월25일

(71) 출원인
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)
(72) 발명자
이영주
경상북도 포항시 남구 청암로 77
권혁준
경상북도 포항시 남구 청암로 77
김영석
경상북도 포항시 남구 청암로 77
(74) 대리인
특허법인(유한)아이시스

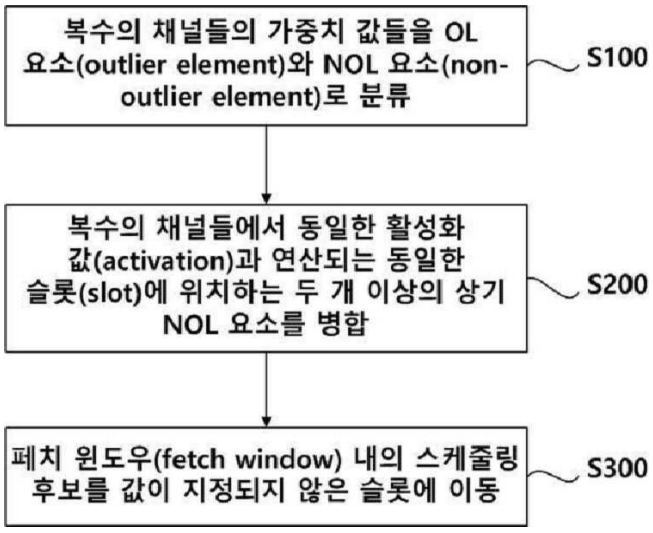
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 **컨벌루션 레이어의 연산 최적화 방법 및 컨벌루션 신경망 가속기**

(57) 요 약

본 실시예는 복수의 채널들을 포함하는 컨벌루션 레이어의 연산 최적화 방법으로, 상기 최적화 방법은: 상기 복수의 채널들의 가중치 값들을 OL 요소(outlier element)와 NOL 요소(non-outlier element)로 분류하는 분류 단계와, 복수의 채널들에서 동일한 활성화 값(activation)과 연산되는 동일한 슬롯(slot)에 위치하는 두 개 이상의 상기 NOL 요소를 병합하는 병합 단계 및 병합 단계 이후 상기 페치 윈도우(fetch window) 내의 스케줄링 후보 요소를 값이 지정되지 않은 슬롯에 이동하는 스케줄링 단계를 포함한다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업
과제고유번호 1711174849
과제번호 2022R1A2C2092521
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 개인기초연구(과기정통부)
연구과제명 차세대 초고효율 6G 베이스밴드 시스템을 위한 알고리즘-하드웨어 융합최적화
기 여 율 40/100
과제수행기관명 포항공과대학교
연구기간 2022.09.01 ~ 2023.02.28
이 발명을 지원한 국가연구개발사업
과제고유번호 1711160341
과제번호 2020-0-01461-003
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성
연구과제명 지능형 의료영상 진단 솔루션 개발
기 여 율 30/100
과제수행기관명 아주대학교산학협력단
연구기간 2022.01.01 ~ 2022.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711159913
과제번호 2020-0-01612-003
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성
연구과제명 Grand ICT연구센터(금오공과대학교)
기 여 율 30/100
과제수행기관명 금오공과대학교 산학협력단
연구기간 2022.01.01 ~ 2022.12.31

명 세 서

청구범위

청구항 1

복수의 채널들을 포함하는 컨벌루션 레이어의 연산 최적화 방법으로, 상기 최적화 방법은:

상기 복수의 채널들의 가중치 값들을 OL 요소(outlier element)와 NOL 요소(non-outlier element)로 분류하는 분류 단계와,

복수의 채널들에서 동일한 활성화 값(activation)과 연산되는 동일한 슬롯(slot)에 위치하는 두 개 이상의 상기 NOL 요소를 병합하는 병합 단계 및

병합 단계 이후 상기 페치 윈도우(fetch window) 내의 스케줄링 후보 요소를 값이 지정되지 않은 슬롯에 이동하는 스케줄링 단계를 포함하는 연산 최적화 방법.

청구항 2

제1항에 있어서,

상기 OL 요소와 상기 NOL 요소는 모두 이진수로 표현 가능하고,

상기 OL 요소를 표현하는 비트 수는, 상기 NOL 요소를 표현하는 비트 수의 n 배인 연산 최적화 방법. (n: 2 이상의 정수)

청구항 3

제2항에 있어서,

상기 슬롯에 할당된 비트수는,

상기 OL 요소의 비트수에 상응하는 연산 최적화 방법.

청구항 4

제1항에 있어서,

상기 병합 단계에서,

동일한 상기 슬롯에 병합되는 복수의 NOL 요소들의 비트 수의 합이 상기 슬롯에 지정된 비트 수보다 클 때,

상기 슬롯에 지정된 비트수를 초과하는 개수의 상기 NOL 요소들은 후속하는 사이클에서 연산되도록 사이클을 추가하는 연산 최적화 방법.

청구항 5

제4항에 있어서,

상기 병합 단계에서,

상기 슬롯에 지정된 비트수에 부합하는 개수의 상기 NOL 요소들을 상기 슬롯에 병합하는 연산 최적화 방법.

청구항 6

제1항에 있어서,

상기 페치 윈도우는,

상기 동일한 사이클 다음 j 개의 사이클에서 가중치 값과 연산되는 요소들을 포함하는 연산 최적화 방법.(j: 자연수)

청구항 7

제6항에 있어서,

상기 j는 2인 연산 최적화 방법.

청구항 8

제1항에 있어서,

상기 스케줄링 후보는,

상기 동일한 사이클 다음 사이클(h 번째 사이클)에서,

상기 값이 지정되지 않은 슬롯의 열(i 번째 열)과 동일한 열, i-1번째 열 및 i+1번째 열에 위치한 슬롯에 포함된 요소를 포함하는 연산 최적화 방법.

청구항 9

제8항에 있어서,

상기 스케줄링 후보는,

h+1 번째 사이클에서,

상기 값이 지정되지 않은 슬롯의 열(i 번째 열)과 동일한 열, i-2번째 열 및 i+2번째 열에 위치한 슬롯에 포함된 요소를 더 포함하는 연산 최적화 방법.

청구항 10

복수개의 연산 요소들을 포함하는 컨벌루션 신경망 가속기로, 상기 신경망 가속기는:

활성화 값을 저장하는 활성화 버퍼(activation buffer)와,

복수의 채널들이 병합되어 형성된 NOL(non-outlier) 가중치를 저장하는 가중치 버퍼(weight buffer)를 포함하고,

상기 연산 요소들 각각은:

상기 활성화 값과 NOL(non-outlier) 가중치의 곱을 연산하는 곱셈부(multiplication unit); 및

상기 곱셈부의 출력을 상기 병합된 채널에 따라 서로 다른 누적부로 출력하는 다중화부(MUX unit) 및

병합된 복수의 채널들별로 상기 곱셈 결과를 누적하는 누적부(accumulation unit)을 포함하는 신경망 가속기.

청구항 11

제10항에 있어서,

상기 곱셈부는,

상기 NOL(non-outlier) 가중치와 상기 활성화 값을 입력받아 곱셈 연산하는 곱셈기를 포함하는 신경망 가속기.

청구항 12

제11항에 있어서,

상기 곱셈부는,

하나 이상의 곱셈기를 더 포함하며,

상기 곱셈기와 상기 하나 이상의 곱셈기는,

각각 OL 가중치의 일부와 상기 활성화 값을 입력받고

각각 상기 OL 가중치와 상기 활성화 값과의 곱셈 연산의 부분곱을 연산하는 신경망 가속기.

청구항 13

제12항에 있어서,

상기 곱셈기는,

상기 부분곱의 자리수가 정렬되도록 부분곱을 시프트하는 시프터(shfiter)를 더 포함하는 신경망 가속기.

청구항 14

제10항에 있어서,

상기 다중화부(MUX unit)는

병합된 상기 채널들의 개수에 상응하는 복수의 단위 다중화 부들을 포함하며,

상기 단위 다중화부들 각각은

상기 곱셈부의 연산 결과와 논리 0 및 선택 신호가 제공되는 상기 복수의 다중화기(multiplexer)들을 포함하고,

상기 다중화기들은 선택 신호에 따라 제어되어 상기 곱셈부의 연산 결과를 출력하는 신경망 가속기.

청구항 15

제10항에 있어서,

상기 신경망 가속기는,

상기 곰셈부를 복수개 포함하며,

상기 다중화부(MUX unit)는 병합된 상기 채널들의 개수에 상응하는 복수의 단위 다중화 부들을 포함하며,

상기 단위 다중화부들 각각은

선택 신호와, 논리 0과, 상기 복수의 곱셈부 중 어느 하나가 연산한 연산 결과의 일부 및 상기 복수의 곱셈부들이 각각 연산한 연산 결과의 나머지 일부를 모두 입력받는 다중화기(multiplexer)들을 포함하고,

상기 다중화기들은 선택 신호에 따라 제어되어 상기 곱셈부의 연산 결과를 출력하는 신경망 가속기.

청구항 16

제10항에 있어서,

상기 신경망 가속기는,

상기 활성화 버퍼(activation buffer)가 출력한 값을 저장하는 활성화 레지스터와 상기 가중치 버퍼(weight buffer)가 출력한 값을 저장하는 가중치 레지스터와,

상기 곱셈부 출력을 저장하는 곱셈부 레지스터 및

상기 누적부 출력을 저장하는 누적부 레지스터를 더 포함하는 신경망 가속기.

청구항 17

제16항에 있어서,

상기 활성화 레지스터와 상기 가중치 레지스터,

상기 곱셈부 레지스터 및

상기 누적부 레지스터는 파이프라인 방식으로 동작하는 신경망 가속기.

발명의 설명

기 술 분 야

[0001] 본 개시는 일반적으로 컨벌루션 레이어의 연산 최적화 방법 및 컨벌루션 신경망 가속기와 관련된다.

배 경 기 술

[0002] 신경망의 일종인 컨벌루션 신경망(CNN, convolutional neural network)은 행렬의 컨벌루션 연산 등을 이용하여 이미지 또는 음성 등의 데이터로부터 수치적 특징을 추출하는데 강점을 가지는 기계 학습 네트워크이다. 이러한 CNN은 하나의 입력을 사용하여 여러 채널에 대한 계산 결과를 내보내는 컨벌루션 층을 포함하는 여러 층(layer)을 가지며, 하나의 층은 여러 채널(channel)을 포함한다. 이러한 특성으로 인해 컨벌루션 신경망은 많은 층과 채널을 이루는 방대한 양의 가중치를 저장하며, 반복적인 연산을 수행한다.

[0003] 효율적인 연산을 위해 저장하는 가중치의 양과 연산 횟수를 줄이도록 데이터 양자화(quantization), 프루닝(pruning) 등의 최적화 방법을 널리 사용하고 있다. 양자화는 데이터를 표현하는 해상도를 낮추는 것으로, 데이터의 정확도를 희생하나 연산되는 데이터 크기를 효과적으로 감소시킬 수 있다. 프루닝은 최종 연산 결과에 미치는 영향이 적은 가중치를 제거하여 필요한 저장 공간을 줄이고 연산 결과에 미치는 영향이 적은 연산을 건너뛰는 효과를 얻을 수 있다.

[0004] 희소행렬은 위의 두 기법으로 인해 값이 0인 데이터가 대부분을 차지하는 데이터 행렬을 의미하며, 양자화와 프루닝으로 최적화하여 크기가 줄어든 행렬 연산을 위한 하드웨어들은 한 사이클에 병렬적으로 배치한 연산기들로 동시에 여러 데이터를 연산하는 데이터 병렬화 및 가지치기로 인해 하드웨어 포화도를 높이고, 연산을 가속할 수 있도록 스케줄링을 이용한다.

발명의 내용

해결하려는 과제

[0005] 양자화와 프루닝을 모두 적용한 컨벌루션 신경망이 등장하고 하고 있으나, 양자화와 프루닝 기법이 모두 적용된 신경망에 적합한 하드웨어 가속기 개발이 활발히 이루어지지 못하여 최적화의 이득을 최대화하지 못하는 상태이다. 즉, 기존의 하드웨어는 하드웨어 포화도를 높였지만, 희소 행렬에서 스케줄링의 효과가 미미해지는 한계점을 가진다.

[0006] 본 실시예로 해결하고자 하는 과제 중 하나는 상기한 종래 기술의 난점을 극복하기 위한 것으로 양자화와 프루닝 기법이 모두 적용된 신경망에 대하여 최적화를 이룰 수 있는 기술을 제공하는 것이다.

과제의 해결 수단

[0007] 본 실시예는 복수의 채널들을 포함하는 컨벌루션 레이어의 연산 최적화 방법으로, 상기 최적화 방법은: 상기 복수의 채널들의 가중치 값들을 OL 요소(outlier element)와 NOL 요소(non-outlier element)로 분류하는 분류 단계와, 복수의 채널들에서 동일한 활성화 값(activation)과 연산되는 동일한 슬롯(slot)에 위치하는 두 개 이상의 상기 NOL 요소를 병합하는 병합 단계 및 병합 단계 이후 상기 페치 윈도우(fetch window) 내의 스케줄링 후보 요소를 값이 지정되지 않은 슬롯에 이동하는 스케줄링 단계를 포함한다.

[0008] 본 실시예의 어느 한 측면에 의하면, 상기 OL 요소와 상기 NOL 요소는 모두 이진수로 표현 가능하고, 상기 OL 요소를 표현하는 비트 수는, 상기 NOL 요소를 표현하는 비트 수의 n 배이다. (n: 2 이상의 정수)

[0009] 본 실시예의 어느 한 측면에 의하면, 상기 슬롯에 할당된 비트수는, 상기 OL 요소의 비트수에 상응한다.

[0010] 본 실시예의 어느 한 측면에 의하면, 상기 병합 단계에서, 동일한 상기 슬롯에 병합되는 복수의 NOL 요소들의 비트 수의 합이 상기 슬롯에 지정된 비트 수보다 클 때, 상기 슬롯에 지정된 비트수를 초과하는 개수의 상기 NOL 요소들은 후속하는 사이클에서 연산되도록 사이클을 추가한다.

[0011] 본 실시예의 어느 한 측면에 의하면, 상기 병합 단계에서, 상기 슬롯에 지정된 비트수에 부합하는 개수의 상기 NOL 요소들을 상기 슬롯에 병합한다.

[0012] 본 실시예의 어느 한 측면에 의하면, 상기 페치 윈도우는, 상기 동일한 사이클 다음 j 개의 사이클에서 가중치 값과 연산되는 요소들을 포함한다.(j: 자연수)

[0013] 본 실시예의 어느 한 측면에 의하면, 상기 j는 2이다.

[0014] 본 실시예의 어느 한 측면에 의하면, 상기 스케줄링 후보는, 상기 동일한 사이클 다음 사이클(h 번째 사이클)에서, 상기 값이 지정되지 않은 슬롯의 열(i 번째 열)과 동일한 열, i-1번째 열 및 i+1번째 열에 위치한 슬롯에

포함된 요소를 포함한다.

[0015] 본 실시예의 어느 한 측면에 의하면, h+1 번째 사이클에서, 상기 값이 지정되지 않은 슬롯의 열(i 번째 열)과 동일한 열, i-2번째 열 및 i+2번째 열에 위치한 슬롯에 포함된 요소를 더 포함한다.

[0016] 본 실시예는 복수개의 연산 요소들을 포함하는 컨벌루션 신경망 가속기로, 상기 신경망 가속기는: 활성화 값을 저장하는 활성화 버퍼(activation buffer)와, 복수의 채널들이 병합되어 형성된 NOL(non-outlier) 가중치를 저장하는 가중치 버퍼(weight buffer)를 포함하고, 상기 연산 요소들 각각은: 상기 활성화 값과 NOL(non-outlier) 가중치의 곱을 연산하는 곱셈부(multiplication unit); 및 상기 곱셈부의 출력을 상기 병합된 채널에 따라 서로 다른 누적부로 출력하는 다중화부(MUX unit) 및 병합된 복수의 채널들별로 상기 곱셈 결과를 누적하는 누적부(accumulation unit)을 포함한다.

[0017] 본 실시예의 어느 한 측면에 의하면, 상기 곱셈부는, 상기 NOL(non-outlier) 가중치와 상기 활성화 값을 입력받아 곱셈 연산하는 곱셈기를 포함한다.

[0018] 본 실시예의 어느 한 측면에 의하면, 상기 곱셈부는, 하나 이상의 곱셈기를 더 포함하며, 상기 곱셈기와 상기 하나 이상의 곱셈기는, 각각 OL 가중치의 일부와 상기 활성화 값을 입력받고 각각 상기 OL 가중치와 상기 활성화 값과의 곱셈 연산의 부분곱을 연산한다.

[0019] 본 실시예의 어느 한 측면에 의하면, 상기 곱셈기는, 상기 부분곱의 자리수가 정렬되도록 부분곱을 시프트하는 시프터(shfiter)를 더 포함한다.

[0020] 본 실시예의 어느 한 측면에 의하면, 상기 다중화부(MUX unit)는 병합된 상기 채널들의 개수에 상응하는 복수의 단위 다중화 부들을 포함하며, 상기 단위 다중화부들 각각은 상기 곱셈부의 연산 결과와 논리 0 및 선택 신호가 제공되는 상기 복수의 다중화기(multiplexer)들을 포함하고, 상기 다중화기들은 선택 신호에 따라 제어되어 상기 곱셈부의 연산 결과를 출력한다.

[0021] 본 실시예의 어느 한 측면에 의하면, 상기 신경망 가속기는, 상기 곰셈부를 복수개 포함하며, 상기 다중화부(MUX unit)는 병합된 상기 채널들의 개수에 상응하는 복수의 단위 다중화 부들을 포함하며, 상기 단위 다중화부들 각각은 선택 신호와, 논리 0과, 상기 복수의 곱셈부 중 어느 하나가 연산한 연산 결과의 일부 및 상기 복수의 곱셈부들이 각각 연산한 연산 결과의 나머지 일부를 모두 입력받는 다중화기(multiplexer)들을 포함하고, 상기 다중화기들은 선택 신호에 따라 제어되어 상기 곱셈부의 연산 결과를 출력한다.

[0022] 본 실시예의 어느 한 측면에 의하면, 상기 신경망 가속기는, 상기 활성화 버퍼(activation buffer)가 출력한 값을 저장하는 활성화 레지스터와 상기 가중치 버퍼(weight buffer)가 출력한 값을 저장하는 가중치 레지스터와, 상기 곱셈부 출력을 저장하는 곱셈부 레지스터 및 상기 누적부 출력을 저장하는 누적부 레지스터를 더 포함한다.

[0023] 본 실시예의 어느 한 측면에 의하면, 상기 활성화 레지스터와 상기 가중치 레지스터, 상기 곱셈부 레지스터 및 상기 누적부 레지스터는 파이프라인 방식으로 동작한다.

발명의 효과

[0024] 본 실시예에 의하면 신경망 가속기가 연산 효율적, 에너지 효율적으로 동작할 수 있는 연산 최적화 방법과 그에 따라 동작하는 신경망 가속기가 제공된다.

도면의 간단한 설명

[0025] 도 1은 본 실시예에 의한 복수의 채널들을 포함하는 컨벌루션 레이어의 연산 최적화 방법의 개요를 도시한 도면이다.

도 2는 컨벌루션 레이어에 포함된 복수의 채널들의 분류 단계를 수행한 결과를 예시적으로 도시한 도면이다.

도 3은 병합 단계가 수행된 결과를 예시한 도면이다.

도 4는 스케줄링 단계의 수행 과정을 예시한 도면이다.

도 5는 스케줄링이 완료된 상태를 예시한 도면이다.

도 6은 본 실시예에 의한 신경망 가속기의 개요를 도시한 블록도이다.

도 7은 연산 요소의 제1 실시예를 설명하기 위한 개요도이다.

도 8은 연산 요소의 제2 실시예를 설명하기 위한 개요도이다.

도 9는 다중화기를 포함하는 단위 다중화부의 개요를 도시한 도면이다.

도 10는 본 실시예에 의한 가속기의 동작 성능을 VGG-16, ResNet-18 및 MobileNetV2의 세가지 컨벌루션 신경망으로 평가하여 나타낸 그래프이다.

도 11은 아래는 본 실시예에 의한 가속기의 에너지 효율을 나타내는 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0026] 이하에서는 첨부된 도면들을 참조하여 본 실시예를 설명한다. 도 1은 본 실시예에 의한 복수의 채널들을 포함하는 컨벌루션 레이어의 연산 최적화 방법의 개요를 도시한 도면이다. 도 1을 참조하면, 최적화 방법은: 복수의 채널들의 가중치 값들을 OL 요소(outlier element)와 NOL 요소(non-outlier element)로 분류하는 분류 단계(S100)와, 복수의 채널들에서 동일한 활성화 값(activation)과 연산되는 동일한 슬롯(slot)에 위치하는 두 개 이상의 NOL 요소들을 병합하는 병합 단계(S200) 및 병합 단계 이후 페치 윈도우(fetch window) 내의 스케줄링 후보 요소를 값이 지정되지 않은 슬롯에 이동하는 스케줄링 단계(S300)를 포함한다.

[0027] 도 2는 컨벌루션 레이어에 포함된 복수의 채널들의 분류 단계(S100)를 수행한 결과를 예시적으로 도시한 도면이다. 도 1 및 도 2를 참조하면, 미리 정해진 문턱값보다 작은 가중치 값을 가지는 요소들을 제거하는 프루닝(pruning), 각각의 요소의 비트 해상도를 감소시키는 양자화(quantization)을 수행하면 각 채널의 요소들의 다수는 제거되어 결과적으로 희소 행렬(sparse matrix) 형태를 가진다. 희소 행렬을 그대로 이용하여 활성화 값(activation)과 연산을 수행하면 연산 속도와 연산의 효율성 측면에서 낭비가 발생한다.

[0028] 본 실시예에서, 각 채널들의 가중치 요소들을 OL(outlier) 요소와 NOL(nonoutlier) 요소로 분류한다. 일 예로, OL 요소를 표현하는 비트 수는, 상기 NOL 요소를 표현하는 비트 수의 n 배(n: 2 이상의 자연수)일 수 있으며, OL 요소의 비트수는 채널을 이루는 각각의 슬롯(S)에 지정된 비트수에 상응할 수 있다.

[0029] 일 예로, 슬롯에 지정된 비트수가 8 비트이면, OL 요소의 비트수는 8 비트일 수 있고, NOL 요소의 비트수는 4 비트일 수 있다. 다른 예로, 슬롯에 지정된 비트수가 15 비트이면, OL 요소의 비트수는 15비트일 수 있고, NOL 요소의 비트수는 5 비트일 수 있다.

[0030] 도시된 분류 결과에서, OL 요소 보다 적은 비트 수로 표현될 수 있는 NOL 요소인 가중치를 Wnol로 표시하였고, NOL 요소보다 큰 비트수로 표현될 수 있는 OL 요소인 가중치를 Wol로 표시하였다. 또한, 도 2로 예시된 채널들에서 굵은 선으로 표시된 슬롯들은 다른 채널에서 동일한 위치에 0이 아닌 요소가 존재하는 것을 나타낸다.

[0031] 도 2로 예시된 실시예에서, 각 채널은 모두 세 개의 행을 가지는 것으로 도시되었으나, 이는 실시예일 따름이며, 각 채널은 세 개의 이상의 행을 가지는 것이 일반적이다. 또한, 각 행들은 스케줄링(S300) 단계가 수행된 이후, 동일한 연산 사이클에서 활성화 값과 연산이 수행된다.

[0032] 도 3은 병합 단계(S200)가 수행된 결과를 예시한 도면이다. 도 3을 참조하면, 복수의 채널들에서 동일한 활성화 값(activation)과 연산되는 동일한 슬롯(slot)에 위치하는 두 개 이상의 NOL 요소들을 병합한다(S200). 각 채널에서 동일한 위치의 NOL 요소들을 서로 병합한다. 도 3은 슬롯(S)에 지정된 비트 수가 8 비트이고, OL 요소들의 비트 수가 8 비트이며, NOL 요소의 비트 수가 4 비트인 경우를 예시한다. 따라서, 병합 단계(S200)에서 하나의 슬롯에는 최대 두 개의 NOL 요소들이 병합되어 배치될 수 있다. 또한, 하나의 슬롯에는 하나의 OL 요소가 위치할 수 있다.

[0033] 병합 단계(S200)에서 생성되는 가중치 행렬의 0 행(row 0)의 2 열(col 2)요소에는 두 개의 Wnol 요소들이 서로 동일한 위치에 위치하므로 동일한 슬롯(S)으로 병합될 수 없다. 따라서, 다른 하나의 Wnol 요소는 추가되는 행(row a) 내에서 동일한 열인 2 열(col 2)에 배치된다.

[0034] 동일한 행에 포함된 요소들은 가중치들과 동일한 사이클에서 연산되므로 행이 추가되면 연산을 위한 사이클이 추가된다. 따라서, 연산의 속도가 지연될 수 있. 그러나, 병합 단계(S200)와 후술할 스케줄링 단계(S300)를 수행함에 따라 연산 속도가 느려지는 것을 최소화할 수 있다.

[0035] 마찬가지로, 채널 a(Ch. a)와 채널 a+1(Ch. a+1)의 3행 1열의 원소들은 모두 OL 원소들로 병합 단계(S200)에서 서로 동일한 슬롯에 배치될 수 없다. 따라서, 추가 행(row a)를 형성하여 어느 하나의 원소를 추가 행(rowa)에

배치한다.

[0036] 도 4는 스케줄링 단계(S300)의 수행 과정을 예시한 도면이다. 도 4를 참조하면, 스케줄링 단계(S300)에서 페치 윈도우(fetch window, FW) 내에 위치한 스케줄링 후보 요소를 이동하여 스케줄링을 수행한다. 스케줄링을 수행하여 행렬의 밀집도(density)를 향상시킬 수 있다.

[0037] 스케줄링을 수행하기 위한 페치 윈도우(FW)의 크기는 사용자가 지정할 수 있다. 페치 윈도우에 포함되는 행의 개수가 증가할수록 행렬의 밀집도와 연산의 효율성이 증가하나, 스케줄링을 수행하는 하드웨어의 부담도 마찬가지로 증가한다. 따라서, 적절한 선에서의 트레이드 오프(trade-off)가 필요하며, 도시된 실시예는 스케줄링의 대상이 되는 행인 row 0의 다음 두 행인 row 1, row 2로 페치 윈도우를 설정한다.

[0038] 도 4에서 스케줄링 단계(S300)에서 스케줄링 후보 요소들의 이동 경로를 화살표로 표시하였다. 스케줄링 후보 요소는 스케줄링 대상 행인 row 0의 다음 행에서 값이 지정되지 않은 빈 슬롯의 열(i 번째 열)과 동일한 열, i-1번째 열 및 i+1번째 열에 위치한 슬롯에 포함된 요소일 수 있다. 또한, 스케줄링 후보는 스케줄링 대상 행인 row 0의 다음 두번째 행에서 값이 지정되지 않은 빈 슬롯의 열(i 번째 열)과 동일한 열, i-2번째 열 및 i+2번째 열에 위치한 슬롯에 포함된 요소일 수 있다. 스케줄링 후보를 연산할 때 열의 개수를 넘어서는 경우가 있을 수 있으나, 이러한 경우에는 연산된 열 번호에서 모든 열들의 개수를 빼거나 더하여 스케줄링 후보를 연산할 수 있다.

[0039] 도 4로 예시된 것과 같이 row 0의 col 3의 슬롯은 값이 지정되지 않았다. 따라서 스케줄링 후보 요소인 row 2, col 5의 NOL 요소인 Wnol이 스케줄링 되어 이동할 수 있다. row 0, col 1의 슬롯은 값이 지정되지 않았다. 따라서, 스케줄링 후보 요소인 row 2, col1의 NOL 요소인 Wnol이 스케줄링 되어 이동할 수 있다. 나아가, row 1, col 2의 요소도 row 0, col1 로 스케줄링되어 이동할 수 있다. 따라서, row 0, col1에는 두 개의 NOL 가중치 요소가 병합되어 위치할 수 있다.

[0040] row 0, col 0의 슬롯도 값이 지정되지 않았으며, 페치 윈도우(FW) 내 동일한 열 또는 대각 관계에서 스케줄링 후보 요소가 없다. 그러나, row 2, col 6의 슬롯은 대각방향으로 이동하여 col 8로 이동할 수 있으나, col 8은 열의 개수를 넘는다. 따라서, col 8에서 열들의 개수인 8을 빼면 row 2, col 6에 위치하는 Wnol 요소들은 row 0, col 0로 이동되어 스케줄링될 수 있다. 이와 같이 row 0로 이동할 수 있는 요소들을 모두 이동하면, 페치 윈도우(FW) 내에는 요소들이 없다. 따라서, 행에 요소가 없는 경우에는 해당 행을 삭제하여 연산을 수행하지 않도록 하고 0 행에 대한 스케줄링을 완료한다.

[0041] 0 행에 대한 스케줄링을 완료하면 후속하는 3 행(row 3)에 대한 스케줄링을 수행할 수 있다. row 3, col 1의 슬롯은 값이 지정되지 않았으며, row 4, col 1이 스케줄링 후보가 될 수 있다. 따라서, row 4, col 1의 Wol 가중치 요소를 row 3, col 1로 이동하고, 페치 윈도우 내의 4 행(row 4)을 삭제하여 3 행(row 3)의 스케줄링을 완료하며, 이러한 상태는 도 5로 예시된 것과 같다.

[0042] 일 실시예에서, 스케줄링 단계는 동일한 사이클에서 연산되는 하나의 행에 동일한 채널의 요소가 절반 이상을 넘지않도록 수행된다. 도 5로 예시된 예에서, 하나의 행에 여덟 개의 슬롯이 있으며, 이들 슬롯 중에서 네 개의 슬롯을 동일한 채널의 요소들(예를 들어, 채널 a의 네 개의 OL 요소와 하나의 NOL 요소)이 점유하지 못하도록 스케줄링 된다. 이와 같이 스케줄링을 수행함으로써 후술할 실시예와 같이 크리티컬 딜레이(critical delay)를 감소시켜 연산 속도를 향상시킬 수 있다.

[0043] 위에서 설명된 실시예는 방법의 형태로 설명되었으며, 본 실시예에 의한 방법은 기계에서 실행될 수 있는 프로그램으로 구현될 수 있다. 그러나 상술한 설명은 본 실시예의 방법이 하드웨어에서 실시간으로 데이터가 입력되어 온 더 플라이(on-the-fly) 형태로 실시되는 것을 배제하는 것이 아니다.

[0044] 상술한 실시예에 의하면 희소 행렬 형태를 가지는 복수의 채널들에 대하여 분류 단계, 병합 단계 및 스케줄링 단계를 수행하여 높은 밀집도를 가지도록 할 수 있고, 이로부터 높은 연산 속도 및 연산 효율을 얻을 수 있으며, 나아가 전력 소모도 감소시킬 수 있다는 장점이 제공된다.

[0045] 이하에서는 첨부된 도 6 내지 도 9를 참조하여 본 실시예에 의한 신경망 가속기를 설명한다. 도 6은 본 실시예에 의한 신경망 가속기의 개요를 도시한 블록도이다. 도 6을 참조하면, 신경망 가속기는 가중치(weight)와 활성화 값(activation)을 제공받고, 저장하며, 복수의 타일(tile)들에 제공하는 전역 버퍼(global buffer)와, 전역 버퍼로부터 제공된 가중치와 활성화값 으로 연산을 수행하는 복수의 타일(tile)들을 포함한다.

[0046] 각각의 타일(tile)들은 제공된 가중치를 저장하는 가중치 버퍼(weight buffer, 200)와 활성화 값을 저장하는 활

성화 버퍼(activation buffer, 100) 및 가중치 버퍼(weight buffer)가 제공하는 가중치와 활성화 버퍼가 제공하는 활성화 값으로부터 연산을 수행하는 복수의 연산 요소(processing element, 300)들을 포함한다.

[0047] 도 7은 연산 요소(300)의 제1 실시예를 설명하기 위한 개요도이다. 도 7을 참조하면, 연산 요소(300)는 활성화 값과 NOL(non-outlier) 가중치의 곱을 연산하는 곱셈부(multiplication unit, 310) 및 상기 곱셈부의 출력을 상기 병합된 채널에 따라 서로 다른 누적부로 출력하는 다중화부(MUX unit, 320) 및 병합된 복수의 채널들별로 상기 곱셈 결과를 누적하는 누적부(accumulation unit, 330)을 포함한다.

[0048] 가중치 버퍼(200)가 곱셈부(310)에 출력하는 데이터들은 두 개의 NOL 가중치 데이터이거나, 하나의 OL 가중치 데이터일 수 있다. 하나의 NOL 가중치 데이터의 비트 수를 Nw, 하나의 활성화 값의 비트 수를 Na라 하면 곱셈부(310)가 출력한 데이터는 2개의 NOL 가중치와 활성화 값의 곱셈으로부터의 얻어진 두 개의 연산 결과 이거나, OL 가중치와 활성화 값 곱셈으로부터의 얻은 상부 부분곱 및 하부 부분 곱이다.

[0049] 일 실시예로, OL 가중치와 활성화 값과의 연산 결과는 하부 부분곱과 상부 부분곱을 이루는 비트들은 서로 자리수로 정렬한 후 누적되어야 하므로, 시프터(314)를 이용하여 상부 부분곱을 시프트하여 출력한다. 곱셈부(310)의 출력들은 곱셈 레지스터(410)에 출력된다.

[0050] 다중화부(320)는 복수의 단위 다중화부(322)들을 포함하며, 단위 다중화부(322) 각각은 복수의 다중화기(MUX)를 포함한다. 단위 다중화부(322)들의 개수는 컨벌루션 레이어에 포함된 채널들의 개수에 상응할 수 있으며, 이로부터 곱셈부(310)의 연산 결과를 각 채널별로 분할하여 각각의 단위 누적부(332)로 출력할 수 있다.

[0051] 단위 다중화부(322)는 복수의 다중화기(MUX)를 포함하고, 다중화기(MUX)의 입력에는 “0” 또는 곱셈부(310)가 NOL 가중치와 활성화 값을 곱셈 연산한 결과가 입력된다. 다중화기(MUX) 각각에 채널 선택 신호(sel)를 제공하여 각 다중화기(MUX)의 출력을 제어한다. 일 예로, 채널 선택 신호(sel)에 의하여 단위 다중화부(322)에서 미리 정해진 채널의 가중치와 활성화 값의 곱을 선택하여 출력하도록 하고, 다중화기(MUX)에 입력으로 제공된 값이 미리 정해진 채널의 가중치와 활성화 값의 곱에 상응하지 않으면 선택 신호(sel)는 다중화기(MUX)가 “0”을 출력하도록 제어한다. 즉, 다중화부(320) 내에 포함된 복수의 단위 다중화부(322)들은 동일한 곱셈 결과를 입력받고, 각각 채널 선택 신호(sel)로 지정된 채널의 연산 결과만을 출력한다.

[0052] 일 예로, 단위 다중화부(322)에 포함된 MUX0의 입력으로 채널 0의 NOL 가중치와 활성화 값의 곱셈 결과가 입력되고, MUX1의 입력으로 채널 1의 NOL 가중치와 활성화 값의 곱셈 결과가 입력되는 경우를 가정한다. MUX0가 NOL 가중치와 활성화 값의 곱셈 결과를 출력하고, MUX1은 0을 출력하도록 채널 선택 신호(sel)가 제공될 수 있다.

[0053] 다른 예로, 단위 다중화부(322)에 포함된 MUX0의 입력으로 채널 0의 OL 가중치와 활성화값의 곱셈 결과 중 상위 부분곱이 입력되고, MUX1의 입력으로 채널 0의 OL 가중치와 활성화값의 곱셈 결과 중 하위 부분곱이 입력되되는 경우를 가정한다. 상위 부분곱은 시프터(314)에 의하여 정렬이 수행되었다. MUX0가 상위 부분곱을 출력하고, MUX1은 하위 부분곱을 출력하도록 채널 선택 신호(sel)가 제공될 수 있다.

[0054] 각각의 단위 다중화부(322)가 출력한 값들은 각 채널별 가중치와 활성화값이 곱해진 결과이고, 이들은 각각의 단위 누적부(332)에 출력되고 누적되어 누적 레지스터(420)에 제공된다.

[0055] 도 8은 연산 요소(300)의 제2 실시예를 설명하기 위한 개요도이다. 도 7을 참조하여 설명된 실시예와 동일하거나 유사한 요소에 대한 설명은 실시예의 간결한설명을 위하여 생략할 수 있다. 도 8을 참조하면, 연산 요소(300)는 활성화 값과 NOL(non-outlier) 가중치의 곱을 연산하는 곱셈부(multiplication unit, 310) 및 상기 곱셈부의 출력을 상기 병합된 채널에 따라 서로 다른 누적부로 출력하는 다중화부(MUX unit, 320) 및 병합된 복수의 채널들별로 상기 곱셈 결과를 누적하는 누적부(accumulation unit, 330)을 포함한다. 다중화부(321)는 복수의 단위 다중화부(323)들을 포함하며, 단위 다중화부(323) 각각은 복수의 다중화기(MUX)를 포함한다.

[0056] 도 9는 다중화기(MUX)를 포함하는 단위 다중화부(323)의 개요를 도시한 도면이다. 도 8 및 도 9를 참조하면, 단위 다중화부(323)은 (M+2) 대 1의 다중화기(MUX)를 포함한다. M은 도 8로 예시된 곱셈부(310)의 개수에 상응하는 자연수로, 도 5에서 예시된 단일한 사이클에서 연산될 수 있는 슬롯들의 개수에 상응한다.

[0057] 각각의 단위 다중화부(323)에 포함된 다중화기(MUXa, MUXb, MUXc, MUXd)는 곱셈부(310)가 출력하는 M 개의 상위 연산 결과(U_0, U_1, ..., U_{M-1}), 하나의 하위 연산 결과(L) 및 0이 입력된다. 다중화기는 각각의 입력은 채널 선택 신호(sel)로 제어되어 입력 중 어느 하나를 출력한다.

[0058] 일 예로, 단위 다중화부(323)가 채널 1의 OL 가중치와 활성화 값이 연산된 결과를 출력하여야 하는 경우를 가정

한다. 채널 1의 OL 가중치와 활성화 값이 연산된 결과의 상위 부분곱이 U_1으로 출력되고, 채널 1의 OL 가중치와 활성화 값이 연산된 결과의 하위 부분곱이 L_1으로 출력될 때, MUXb에는 하위 부분곱 L_1을 출력하도록 채널 선택 신호(sel)가 제공된다. 단위 다중화부(323)에 포함된 다른 다중화기 중 어느 하나인 MUXa는 상위 부분곱 U_1을 출력하도록 선택 신호(sel)가 제공된다. 따라서, 단위 다중화부는 상위 부분곱과 하위 부분곱을 후속하는 누적기에 출력하고, 누적기는 이들을 누적하여 해당 채널의 부분합을 형성할 수 있다. 단위 다중화부(323)에 지정되지 않은 다른 채널의 입력이 제공된 다중화기들은 0을 출력하도록 선택 신호가 제공될 수 있다.

[0059] 다중화부(320)에 포함된 단위 다중화부(323)들의 개수는 컨벌루션 레이어에 포함된 채널들의 개수에 상응할 수 있으며, 이로부터 곱셈부(310)의 연산 결과를 각 채널별로 분할하여 각각의 단위 누적부(332)로 출력할 수 있은 위에서 설명된 실시예와 같다.

[0060] 위에서 설명된 바와 같이 동일한 사이클에서 연산되는 슬롯들에서 어느 하나의 채널이 과반을 넘지 않도록 스케줄링된다. 따라서, 도 8 및 도 9로 예시된 실시예에 따른 가속기가 무리 없이 동작할 수 있다.

[0061] 나아가, 전체 2M 개의 입력이 필요한 종래 기술에 비하여 감소한 입력의 개수를 절반인 M 개로 감소시킬 수 있어 하드웨어 복잡도를 감소시킬 수 있으며, 이로부터 연산 속도를 향상시킬 수 있다는 장점이 제공된다.

[0062] 또한, 본 실시예에 의한 연산 요소(300)는 활성화 레지스터(100), 가중치 레지스터(200), 곱셈 레지스터(410) 및 누적 레지스터(420)는 파이프라인 방식으로 동작하므로, 높은 처리량을 얻을 수 있다는 장점이 제공된다.

[0063] **실험예 및 모의 실험예**

[0064] 도 10는 본 실시예에 의한 가속기의 동작 성능을 VGG-16, ResNet-18 및 MobileNetV2의 세가지 컨벌루션 신경망으로 평가하여 나타낸 그래프이다. 검은색 네모는 OL 가중치, NOL 가중치를 구분하지 않은 경우, 파란 동그라미는 OL 가중치와 NOL 가중치로 부분하여 스케줄링을 수행한 경우이고, 보라색 삼각형, 붉은색 역삼각형 및 초록색 마름모꼴은 본 실시예에 따라 각각 두 채널, 네 채널 및 여섯 채널을 병합한 경우를 예시한다.

[0065] 도 10로 도시된 바와 같이 행렬의 희소도(pruning ratio)가 높아짐에 따라 종래 기술에 비해 본 실시예의 성능 향상 폭이 이상적인 성능 향상 폭을 더 잘 따라 가는 것을 확인할 수 있다.

[0066] 도 11은 아래는 본 실시예에 의한 가속기의 에너지 효율을 나타내는 그래프로, 단위 전력당 얼마나 많은 연산을 수행하는지 도시한다. 도 10로 예시된 것과 마찬가지로, VGG-16, ResNet-18 및 MobileNetV2의 세가지 컨벌루션 신경망으로 평가하였다. 검은색 네모는 아무런 스케줄링을 수행하지 않은 경우, 파란 동그라미는 종래 기술로 스케줄링을 수행한 경우이고, 보라색 삼각형, 붉은색 역삼각형 및 초록색 마름모꼴은 본 실시예에 따라 각각 두 채널, 네 채널 및 여섯 채널을 병합한 경우를 예시한다.

[0067] 희소도가 낮은 구간에서는 종래 기술과 제안하는 기술의 일부가 비슷한 성능을 보여주지만, 희소도가 높아짐에 따라 제안하는 기술이 종래 기술에 비해 더 우수한 에너지 효율을 보이는 것을 알 수 있다.

[0068] 본 발명에 대한 이해를 돕기 위하여 도면에 도시된 실시 예를 참고로 설명되었으나, 이는 실시를 위한 실시예로, 예시적인 것에 불과하며, 당해 분야에서 통상적 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 특허청구범위에 의해 정해져야 할 것이다.

부호의 설명

[0069] 1: 신경망 가속기

100: 활성화 레지스터 200: 가중치 레지스터

300: 연산 요소

310: 곱셈부 312: 곱셈기

314: 시프터 320: 다중화부

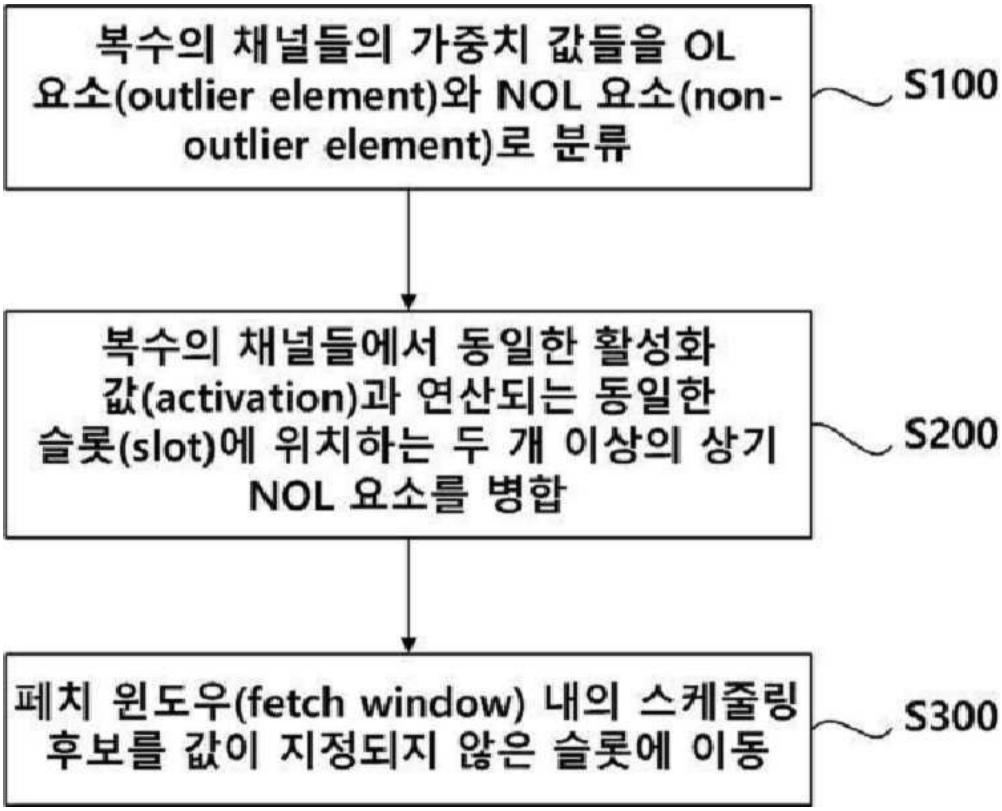
321: 다중화부 322: 단위 다중화부

323: 단위 다중화부 330: 누적부

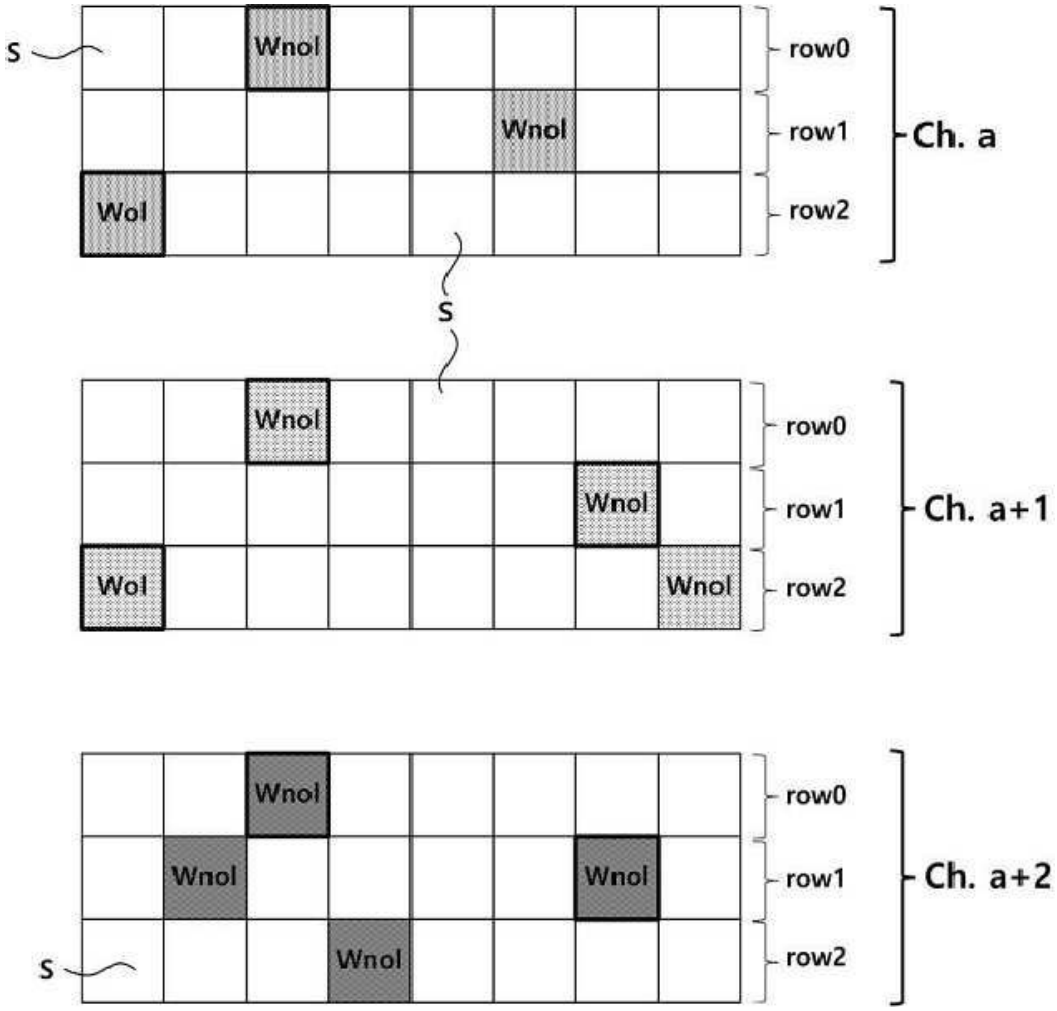
332: 단위 누적부

도면

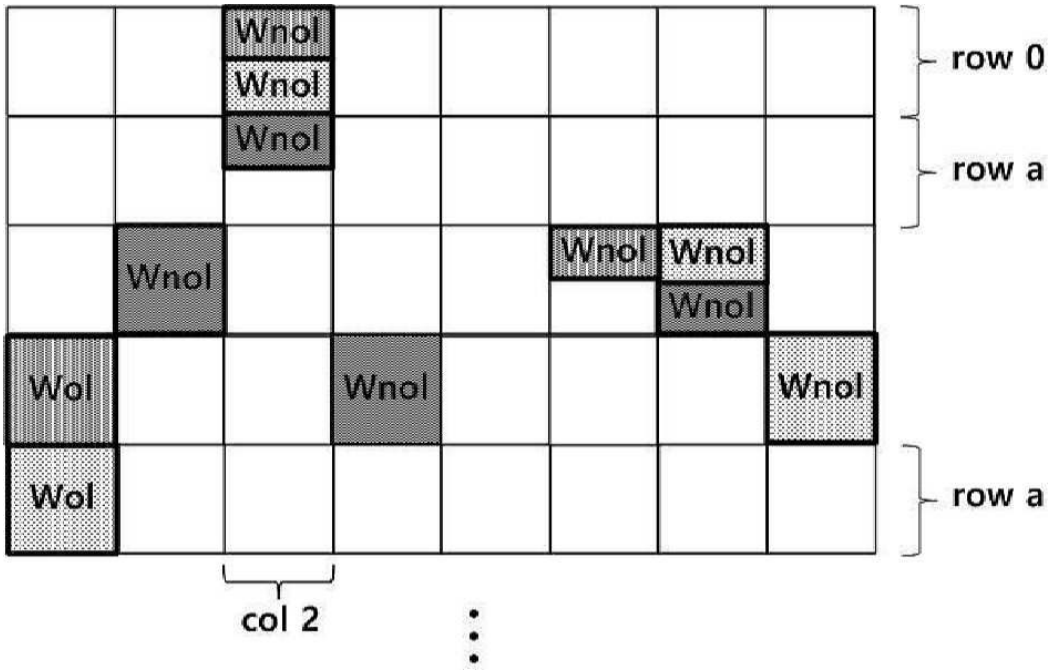
도면1



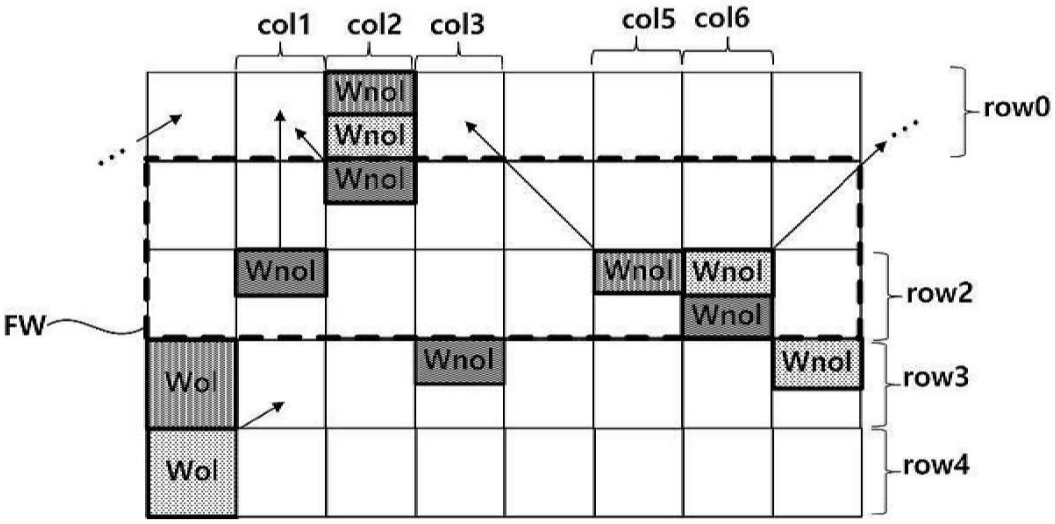
도면2


S
Wnol
Wol
row0
row1
row2
Ch. a
S
Wnol
Wol
Ch. a+1
S
Wnol
Ch. a+2

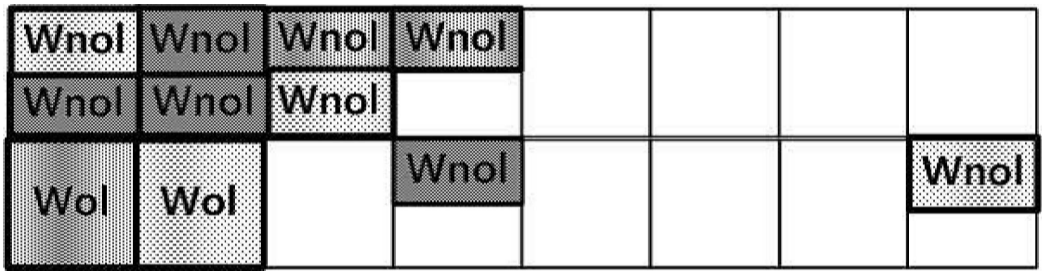
도면3


Wnol
Wol
row 0
row a
row a
col 2

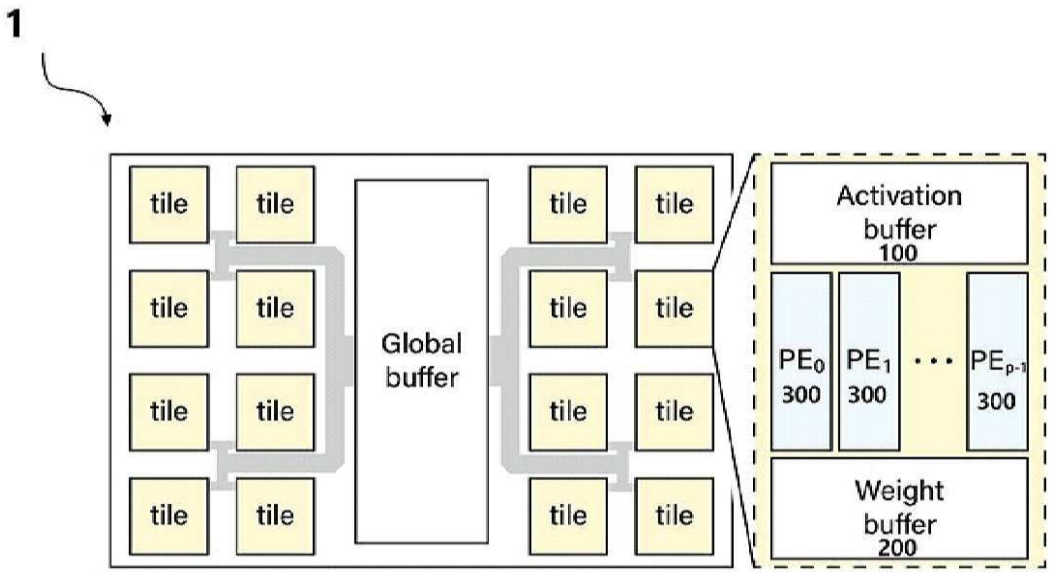
도면4


col1
col2
col3
col5
col6
row0
row2
row3
row4
FW
Wnol
Wol

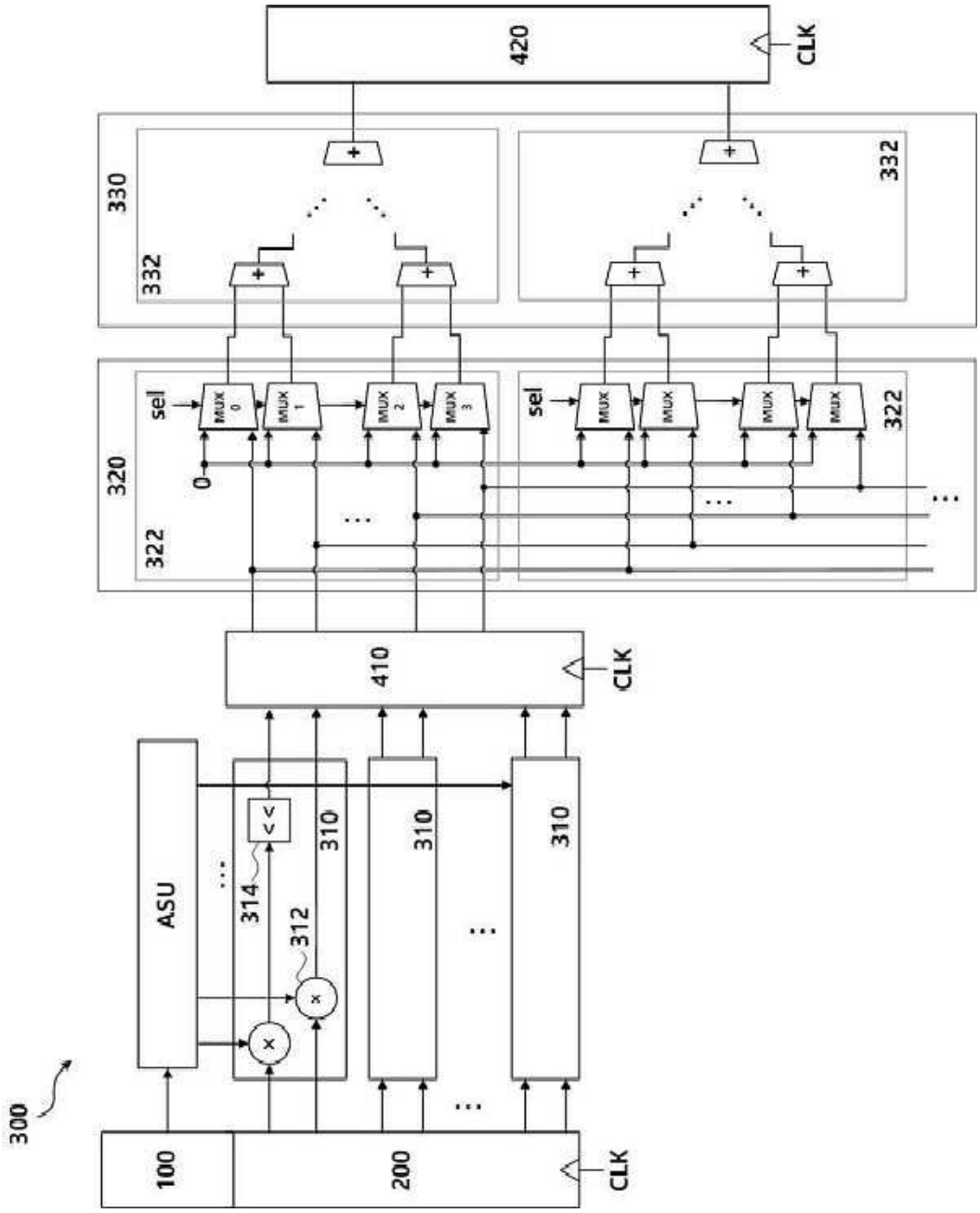
도면5


Wnol
Wol

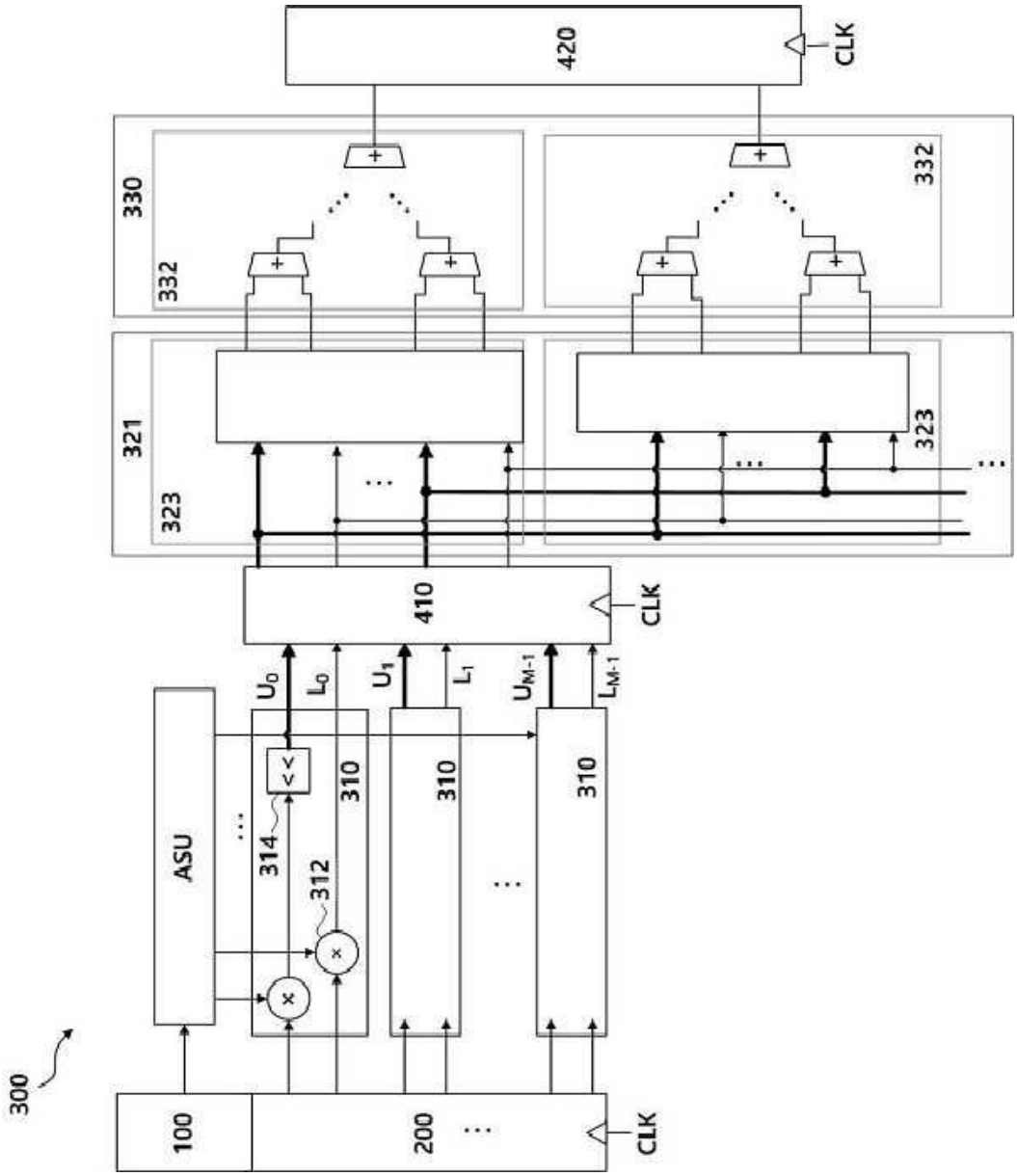
도면6


1
tile
Global buffer
Activation buffer
100
PE0
300
PE1
300
PEp-1
300
Weight buffer
200

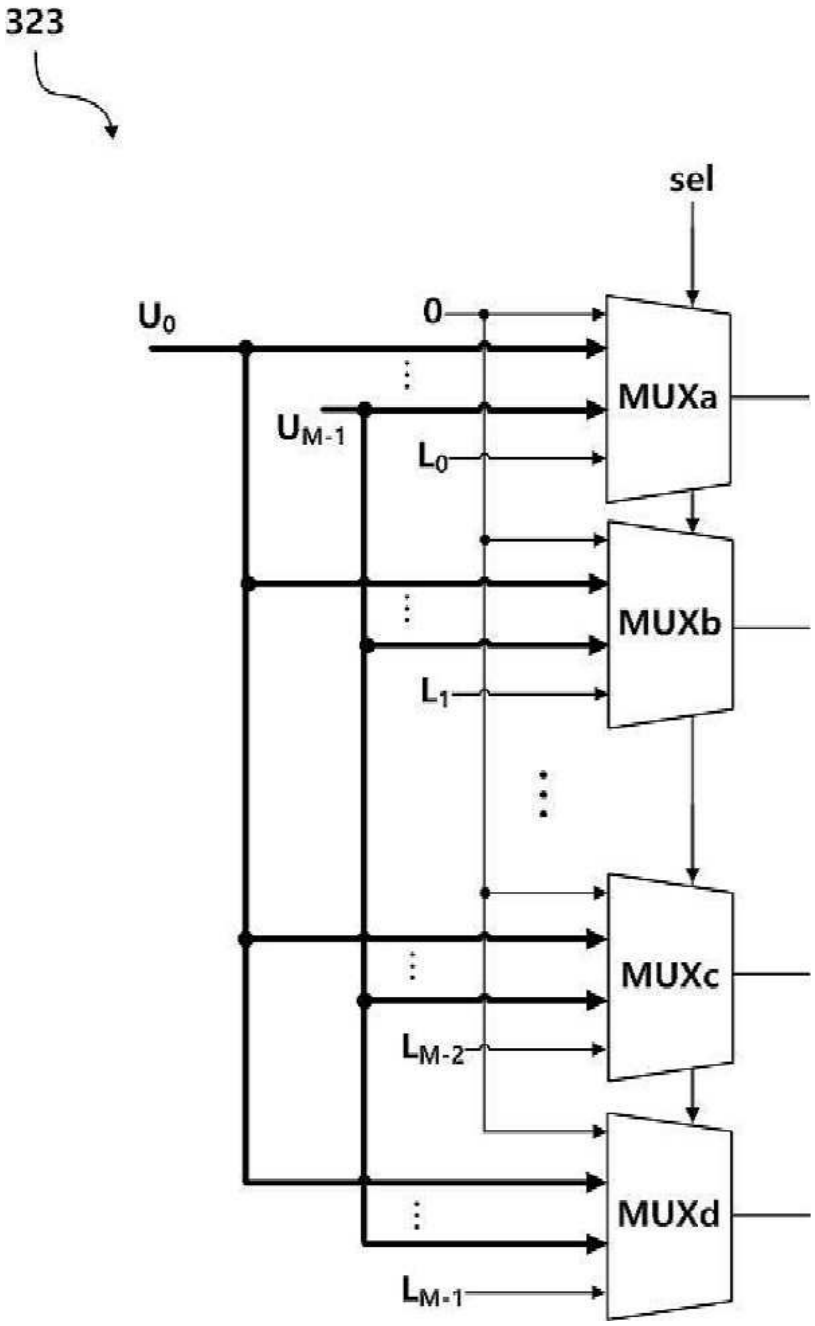
도면7


420
CLK
330
332
332
sel
MUX 0
MUX 1
MUX 2
MUX 3
sel
MUX
MUX
MUX
MUX
322
320
0
322
410
CLK
ASU
<<
314
312
310
310
310
300
100
200
CLK

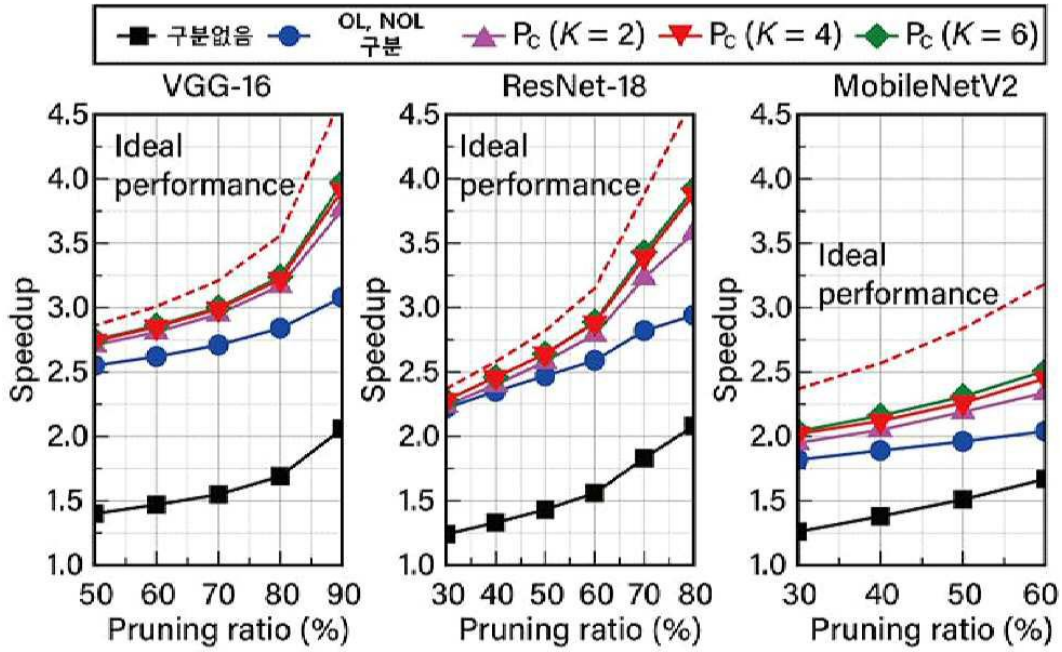
도면8


420
CLK
330
332
332
321
323
323
410
CLK
U_0
L_0
U_1
L_1
U_{M-1}
L_{M-1}
ASU
314
312
310
310
310
300
100
200
CLK

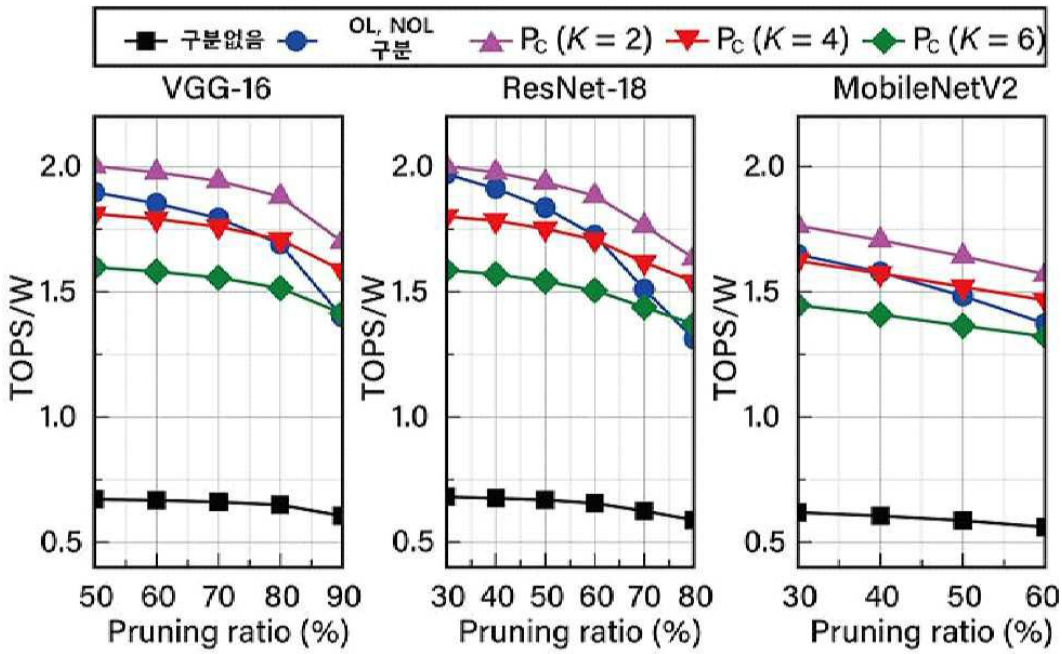
도면9



도면10


구분없음
OL, NOL 구분
Pc (K = 2)
Pc (K = 4)
Pc (K = 6)
VGG-16
ResNet-18
MobileNetV2
Ideal performance
Speedup
Pruning ratio (%)

도면11


구분없음
OL, NOL 구분
Pc (K = 2)
Pc (K = 4)
Pc (K = 6)
VGG-16
ResNet-18
MobileNetV2
TOPS/W
Pruning ratio (%)