



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2024년10월31일

(11) 등록번호 10-2724725

(24) 등록일자 2024년10월28일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2023.01) G06N 3/04 (2023.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06N 3/04 (2023.01)
(21) 출원번호 10-2021-0156873
(22) 출원일자 2021년11월15일
심사청구일자 2021년11월15일
(65) 공개번호 10-2023-0070863
(43) 공개일자 2023년05월23일
(56) 선행기술조사문헌
KR1020190076916 A*
US20210240797 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)
(72) 발명자
강석형
경상북도 포항시 남구 지곡로 155, 5동 203호 (지곡동, 교수아파트)
강예성
경기도 용인시 기흥구 공세로 76, 101동 509호 (고매동, 세원아파트)
(74) 대리인
특허법인 무한
(뒷면에 계속)

전체 청구항 수 : 총 13 항

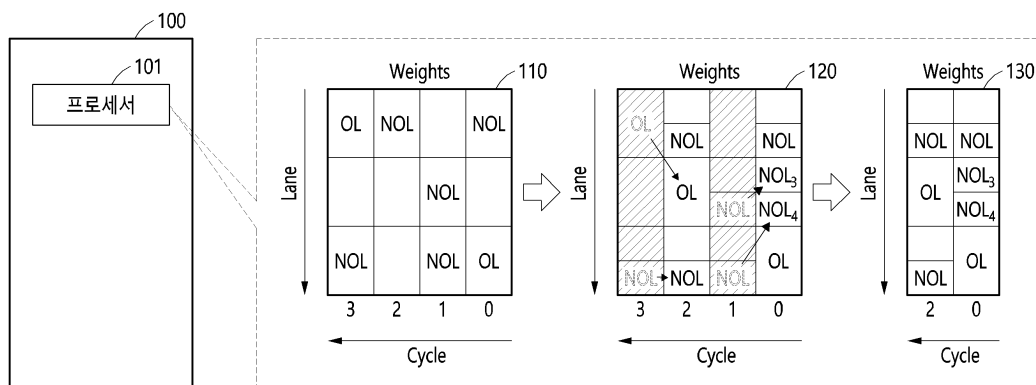
심사관 : 이준상

(54) 발명의 명칭 하드웨어 가속기 성능을 개선시키기 위한 합성곱 신경망의 연산 순서 조절을 위한 장치 및 방법

(57) 요약

일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 장치는, 복수의 레인들에 할당되어 미리 결정된 연산 순서에 따른 사이클에서 연산이 수행되는 복수의 가중치들 중에서 대상 가중치를 선택하고, 상기 대상 가중치에 관한 연산이 수행되는 대상 사이클을 기준으로 룩어헤드 윈도우(lookahead window)에 속하는 복수의 후보 가중치들 중 후보 가중치를 선택하며, 상기 대상 가중치가 제1 비트 정밀도로 표현되는 논-아웃라이어인 경우에 응답하여 상기 선택된 후보 가중치를 상기 대상 가중치와 결합함으로써 결합 가중치를 생성하고, 상기 대상 가중치를 상기 생성된 결합 가중치로 변경함으로써 연산 순서를 조정하는 프로세서를 포함할 수 있다.

대표도 - 도1



(72) 발명자

권은지

부산광역시 금정구 수림로19번길 26, 101동 401호
(부곡동, 동원로알듀크)

이승규

경상북도 포항시 남구 효자동길2번길 43, 302호 (효자동, 벤틀리)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711133968
과제번호	2019R1A5A1027055
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	집단연구지원(R&D)
연구과제명	확장형 양자컴퓨터 기술융합 플랫폼 센터
기 여 율	1/1
과제수행기관명	포항공과대학교
연구기간	2021.03.01 ~ 2022.02.28

공지예외적용 : 있음

명세서

청구범위

청구항 1

합성곱 신경망의 연산 순서를 조정하는 장치에 있어서,

복수의 레인들에 할당되어 미리 결정된 연산 순서에 따른 사이클에서 연산이 수행되는 복수의 가중치들 중에서 대상 가중치를 선택하고, 상기 대상 가중치에 관한 연산이 수행되는 대상 사이클을 기준으로 룩어헤드 윈도우(lookahead window)에 속하는 복수의 후보 가중치들 중 후보 가중치를 선택하며, 상기 대상 가중치가 제1 비트 정밀도로 표현되는 논-아웃라이어인 경우에 응답하여 상기 선택된 후보 가중치를 상기 대상 가중치와 결합함으로써 결합 가중치를 생성하고, 상기 대상 가중치를 상기 생성된 결합 가중치로 변경함으로써 연산 순서를 조정하는 프로세서

를 포함하고,

상기 프로세서는,

상기 복수의 후보 가중치들 각각에 대하여 상기 대상 사이클에서 해당 후보 가중치의 할당이 허용되는 레인의 개수를 산출하고,

상기 해당 후보 가중치의 할당이 허용되는 레인의 개수에 기초하여 상기 복수의 후보 가중치들 중 하나를 선택하는,

합성곱 신경망의 연산 순서를 조정하는 장치.

청구항 2

제1항에 있어서,

상기 프로세서는,

상기 대상 가중치가 제로 가중치인 경우에 응답하여, 상기 대상 가중치를 상기 선택된 후보 가중치로 변경함으로써 연산 순서를 조정하는,

합성곱 신경망의 연산 순서를 조정하는 장치.

청구항 3

제1항에 있어서,

상기 프로세서는,

대상 사이클을 선택하고,

상기 대상 사이클에서 연산이 수행되는 가중치들 각각에 대하여 해당 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치의 개수를 산출하며,

상기 산출된 후보 가중치의 개수에 기초하여 상기 대상 가중치를 선택하는,

합성곱 신경망의 연산 순서를 조정하는 장치.

청구항 4

제3항에 있어서,

상기 프로세서는,

상기 해당 가중치가 제로 가중치인 경우에 응답하여, 논-아웃라이어인 가중치 및 아웃라이어인 가중치를 후보 가중치로 결정하고,

상기 해당 가중치가 논-아웃라이어인 경우에 응답하여, 논-아웃라이어인 가중치를 후보 가중치로 결정하는,

합성곱 신경망의 연산 순서를 조정하는 장치.

청구항 5

삭제

청구항 6

제1항에 있어서,

상기 프로세서는,

상기 대상 사이클에 할당된 가중치가 모두 제로인 경우에 응답하여, 상기 대상 사이클에서의 연산을 스킵(skip)하는,

합성곱 신경망의 연산 순서를 조정하는 장치.

청구항 7

제1항에 있어서,

상기 프로세서는,

상기 복수의 가중치들 중에서 임계 범위에 포함되는 값을 갖는 상기 제1 비트 정밀도보다 높은 제2 비트 정밀도로 표현되는 아웃라이어인 가중치를 상기 제1 비트 정밀도로 표현되는 값으로 수정함으로써, 논-아웃라이어로 변경하는,

합성곱 신경망의 연산 순서를 조정하는 장치.

청구항 8

프로세서에 의해 수행되는 합성곱 신경망의 연산 순서를 조정하는 방법에 있어서,

복수의 레인들에 할당되어 미리 결정된 연산 순서에 따른 사이클에서 연산이 수행되는 복수의 가중치들 중에서 대상 가중치를 선택하는 단계;

상기 대상 가중치에 관한 연산이 수행되는 대상 사이클을 기준으로 룩어헤드 윈도우(lookahead window)에 속하는 복수의 후보 가중치들 중 후보 가중치를 선택하는 단계;

상기 대상 가중치가 제1 비트 정밀도로 표현되는 논-아웃라이어인 경우에 응답하여, 상기 선택된 후보 가중치를 상기 대상 가중치와 결합함으로써 결합 가중치를 생성하는 단계; 및

상기 대상 가중치를 상기 생성된 결합 가중치로 변경함으로써 연산 순서를 조정하는 단계

를 포함하고,

상기 후보 가중치를 선택하는 단계는,

상기 복수의 후보 가중치들 각각에 대하여 상기 대상 사이클에서 해당 후보 가중치의 할당이 허용되는 레인의 개수를 산출하는 단계; 및

상기 해당 후보 가중치의 할당이 허용되는 레인의 개수에 기초하여 상기 복수의 후보 가중치들 중 하나를 선택

하는 단계를 포함하는,
합성곱 신경망의 연산 순서를 조정하는 방법.

청구항 9

제8항에 있어서,
상기 대상 가중치가 제로 가중치인 경우에 응답하여, 상기 대상 가중치를 상기 선택된 후보 가중치로 변경함으로써 연산 순서를 조정하는 단계;
를 더 포함하는 합성곱 신경망의 연산 순서를 조정하는 방법.

청구항 10

제8항에 있어서,
상기 대상 가중치를 선택하는 단계는,
대상 사이클을 선택하는 단계;
상기 대상 사이클에서 연산이 수행되는 가중치들 각각에 대하여 해당 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치의 개수를 산출하는 단계; 및
상기 산출된 후보 가중치의 개수에 기초하여 상기 대상 가중치를 선택하는 단계를 포함하는,
합성곱 신경망의 연산 순서를 조정하는 방법.

청구항 11

제10항에 있어서,
상기 후보 가중치의 개수를 산출하는 단계는,
상기 해당 가중치가 제로 가중치인 경우에 응답하여, 논-아웃라이어인 가중치 및 아웃라이어인 가중치를 후보 가중치로 결정하는 단계; 및
상기 해당 가중치가 논-아웃라이어인 경우에 응답하여, 논-아웃라이어인 가중치를 후보 가중치로 결정하는 단계를 포함하는,
합성곱 신경망의 연산 순서를 조정하는 방법.

청구항 12

삭제

청구항 13

제8항에 있어서,
상기 대상 사이클에 할당된 가중치가 모두 제로인 경우에 응답하여, 상기 대상 사이클에서의 연산을 스킵(skip)하는 단계
를 더 포함하는 합성곱 신경망의 연산 순서를 조정하는 방법.

청구항 14

제8항에 있어서,

상기 복수의 가중치들 중에서 임계 범위에 포함되는 값을 갖는 상기 제1 비트 정밀도보다 높은 제2 비트 정밀도로 표현되는 아웃라이어인 가중치를 상기 제1 비트 정밀도로 표현되는 값으로 수정함으로써, 논-아웃라이어로 변경하는 단계

를 더 포함하는 합성곱 신경망의 연산 순서를 조정하는 방법.

청구항 15

하드웨어와 결합되어 제8항 내지 제11항, 제13항, 및 제14항 중 어느 하나의 항의 방법을 실행시키기 위하여 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램.

발명의 설명

기술 분야

[0001] 이하, 합성곱 신경망의 연산 순서 조정에 관련된 기술이 개시된다.

배경 기술

[0003] 합성곱 신경망은 이미지 분류, 물체 감지 및 분할 응용 분야에서 규칙 기반 학습 알고리즘보다 높은 정확도를 달성할 수 있다. 합성곱 신경망에서의 정확도 향상은 큰 연산량을 요구하기 때문에 가용 전력이 제한된 모바일 장치에 적용하는데 큰 어려움이 있을 수 있다.

[0004] 연산량을 줄이기 위한 방법은 합성곱 신경망의 가중치와 입력(예를 들어, 입력 값, 또는 활성화 값)의 정밀도를 감소시키는 것을 포함할 수 있다. 합성곱 신경망의 정확도(accuracy) 및 가중치 및 입력의 정밀도는 트레이드 오프 관계를 가질 수 있지만, 수용할 수 있는 정확도의 손실 내에서 정밀도를 8 비트까지 양자화(quantization)할 수 있다. VGG-16에서 8 비트 고정 소수점 정밀도를 양자화할 때 제로 가중치의 비율이 크다는 사실로부터 여전히 비효율적으로 계산이 수행될 수 있음을 알 수 있다. 우리는 8비트 정밀도로 표현할 수 있는 논-제로 가중치(Non-zero weight)들 중 4비트 정밀도만으로도 표현 가능한 가중치들을 논-아웃라이어 가중치로 정의하고, 그 외의 가중치들을 아웃라이어 가중치로 정의할 수 있다. 각 레이어마다 비율이 다를 수 있지만, 평균적으로 전체 가중치들 중 제로 가중치가 차지하는 비율은 34.2%일 수 있다. 제로 가중치의 곱셈은 연산 결과에 영향을 주지 않으므로 비효율적인 연산을 야기할 수 있다.

[0005] 비효율적인 연산으로부터 연산 자원의 낭비를 줄이기 위해 제로 가중치를 건너뛰 수 있는 제로 스킵 가속기가 제안되었다. 제로 가중치 스킵 가속기는 논-아웃라이어 가중치를 계산함에 있어서 연산 자원의 낭비를 발생시킬 수 있다. 논-아웃라이어 가중치가 59.7%의 비율을 차지할 수 있는 것을 고려하면, 논-아웃라이어 가중치의 계산에 의해 낭비되는 에너지는 무시할 수 없을 수 있다. 논-아웃라이어에 대한 연산에 대하여, 8비트-8비트 곱셈기 대신 4비트-8비트 곱셈기를 사용함으로써 논-아웃라이어 가중치를 계산하는 데 사용되는 에너지를 절반으로 줄일 수 있다. 논-아웃라이어 가중치에 대하여 4비트-8비트 곱셈기를 사용함으로써 이론적으로 에너지 소비를 29.9%만큼 줄일 수 있다. 소수의 아웃라이어 가중치를 처리할 수 있는 아웃라이어 가중치 인식 가속기가 제안되었다. 아웃라이어 가중치 인식 가속기는, 가중치를 인식하기 위한 추가적인 논리 연산 및 하드웨어 구조를 요구하고, 전력 소모 및 하드웨어 면적이 커질 수 있다.

발명의 내용

과제의 해결 수단

[0007] 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 장치는, 복수의 레인들에 할당되어 미리 결정된 연산 순서에 따른 사이클에서 연산이 수행되는 복수의 가중치들 중에서 대상 가중치를 선택하고, 상기 대상 가중치에 관한 연산이 수행되는 대상 사이클을 기준으로 룩어헤드 윈도우(lookahead window)에 속하는 복수의 후보 가중치들 중 후보 가중치를 선택하며, 상기 대상 가중치가 제1 비트 정밀도로 표현되는 논-아웃라이어인 경우에 응답하여 상기 선택된 후보 가중치를 상기 대상 가중치와 결합함으로써 결합 가중치를 생성하고, 상기 대상 가중

치를 상기 생성된 결합 가중치로 변경함으로써 연산 순서를 조정하는 프로세서를 포함할 수 있다.

- [0008] 상기 프로세서는 상기 대상 가중치가 제로 가중치인 경우에 응답하여, 상기 대상 가중치를 상기 선택된 후보 가중치로 변경함으로써 연산 순서를 조정할 수 있다.
- [0009] 상기 프로세서는, 대상 사이클을 선택하고, 상기 대상 사이클에서 연산이 수행되는 가중치들 각각에 대하여 해당 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치의 개수를 산출하며, 상기 산출된 후보 가중치의 개수에 기초하여 상기 대상 가중치를 선택할 수 있다.
- [0010] 상기 프로세서는, 상기 해당 가중치가 제로 가중치인 경우에 응답하여, 논-아웃라이어인 가중치 및 아웃라이어인 가중치를 후보 가중치로 결정하고, 상기 해당 가중치가 논-아웃라이어인 경우에 응답하여, 논-아웃라이어인 가중치를 후보 가중치로 결정할 수 있다.
- [0011] 상기 프로세서는, 상기 복수의 후보 가중치들 각각에 대하여 상기 대상 사이클에서 해당 후보 가중치의 할당이 허용되는 레인의 개수를 산출하고, 상기 해당 후보 가중치의 할당이 허용되는 레인의 개수에 기초하여 상기 복수의 후보 가중치들 중 하나를 선택할 수 있다.
- [0012] 상기 프로세서는, 상기 대상 사이클에 할당된 가중치가 모두 제로인 경우에 응답하여, 상기 대상 사이클에서의 연산을 스킵(skip)할 수 있다.
- [0013] 상기 프로세서는, 상기 복수의 가중치들 중에서 임계 범위에 포함되는 값을 갖는 상기 제1 비트 정밀도보다 높은 제2 비트 정밀도로 표현되는 아웃라이어인 가중치를 상기 제1 비트 정밀도로 표현되는 값으로 수정함으로써, 논-아웃라이어로 변경할 수 있다.
- [0014] 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 방법은, 복수의 레인들에 할당되어 미리 결정된 연산 순서에 따른 사이클에서 연산이 수행되는 복수의 가중치들 중에서 대상 가중치를 선택하는 단계, 상기 대상 가중치에 관한 연산이 수행되는 대상 사이클을 기준으로 룩아헤드 윈도우(lookahead window)에 속하는 복수의 후보 가중치들 중 후보 가중치를 선택하는 단계, 상기 대상 가중치가 제1 비트 정밀도로 표현되는 논-아웃라이어인 경우에 응답하여, 상기 선택된 후보 가중치를 상기 대상 가중치와 결합함으로써 결합 가중치를 생성하는 단계, 및 상기 대상 가중치를 상기 생성된 결합 가중치로 변경함으로써 연산 순서를 조정하는 단계를 포함할 수 있다.
- [0015] 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 방법은, 상기 대상 가중치가 제로 가중치인 경우에 응답하여, 상기 대상 가중치를 상기 선택된 후보 가중치로 변경함으로써 연산 순서를 조정하는 단계를 더 포함할 수 있다.
- [0016] 상기 대상 가중치를 선택하는 단계는, 대상 사이클을 선택하는 단계, 상기 대상 사이클에서 연산이 수행되는 가중치들 각각에 대하여 해당 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치의 개수를 산출하는 단계, 및 상기 산출된 후보 가중치의 개수에 기초하여 상기 대상 가중치를 선택하는 단계를 포함할 수 있다.
- [0017] 상기 후보 가중치의 개수를 산출하는 단계는, 상기 해당 가중치가 제로 가중치인 경우에 응답하여, 논-아웃라이어인 가중치 및 아웃라이어인 가중치를 후보 가중치로 결정하는 단계 및 상기 해당 가중치가 논-아웃라이어인 경우에 응답하여, 논-아웃라이어인 가중치를 후보 가중치로 결정하는 단계를 포함할 수 있다.
- [0018] 상기 후보 가중치를 선택하는 단계는, 상기 복수의 후보 가중치들 각각에 대하여 상기 대상 사이클에서 해당 후보 가중치의 할당이 허용되는 레인의 개수를 산출하는 단계, 및 상기 해당 후보 가중치의 할당이 허용되는 레인의 개수에 기초하여 상기 복수의 후보 가중치들 중 하나를 선택하는 단계를 포함할 수 있다.
- [0019] 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 방법은, 상기 대상 사이클에 할당된 가중치가 모두 제로인 경우에 응답하여, 상기 대상 사이클에서의 연산을 스킵(skip)하는 단계를 더 포함할 수 있다.
- [0020] 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 방법은, 상기 복수의 가중치들 중에서 임계 범위에 포함되는 값을 갖는 상기 제1 비트 정밀도보다 높은 제2 비트 정밀도로 표현되는 아웃라이어인 가중치를 상기 제1 비트 정밀도로 표현되는 값으로 수정함으로써, 논-아웃라이어로 변경하는 단계를 더 포함할 수 있다.
- [0021] 일 실시예에 따른 컴퓨터 프로그램은, 하드웨어와 결합되어 전술된 합성곱 신경망의 연산 순서를 조정하는 방법들 중 어느 하나의 방법을 실행시키기 위하여 컴퓨터 판독 가능한 기록매체에 저장될 수 있다.

도면의 간단한 설명

- [0023] 도 1은 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 과정을 나타낸다.
- 도 2는 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 방법의 흐름도를 나타낸다.
- 도 3은 일 실시예에 따른 대상 가중치에 따른 룩어헤드 윈도우 및 후보 가중치를 나타낸다.
- 도 4는 일 실시예에 따른 프로세서에 의한 대상 가중치의 선택을 나타낸다.
- 도 5는 일 실시예에 따른 후보 가중치들 중에서 대상 가중치에 대한 연산 순서로 조정될 후보 가중치를 선택하는 것을 나타낸다.
- 도 6는 일 실시예에 따른 선택된 후보 가중치를 대상 가중치에 대한 연산 순서로 할당하는 것을 나타낸다.
- 도 7는 일 실시예에 따른 대상 가중치가 논-아웃라이어인 경우 연산 순서 조정을 나타낸다.
- 도 8는 일 실시예에 따른 대상 가중치가 제로 가중치인 경우 연산 순서 조정을 나타낸다.
- 도 9는 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 것을 나타낸다.
- 도 10a 및 도 10b는 일 실시예에 따른 조정된 연산 순서를 위한 재구성가능한 곱셈기(reconfigurable Multiplier)를 나타낸다.
- 도 11은 일 실시예에 따른 속도 개선 및 가중치 희소성(sparsity)의 관계를 나타낸다.
- 도 12는 비교 실시예 및 일 실시예에 따른 레이어마다 제로 가중치, 논-아웃라이어 가중치의 비율, 및 속도 개선을 나타낸다.
- 도 13는 비교 실시예 및 일 실시예에 따른 4개의 합성곱 신경망들 및 4개의 멀티플렉서들의 크기에 따른 성능 개선을 나타낸다.
- 도 14는 일 실시예에 따른 클리핑 기법에 따른 정확도 및 성능 향상의 관계를 나타낸다.
- 도 15는 일 실시예에 따른 활성화 값의 희소성을 이용하여 실시간으로 연산 순서를 조정(예를 들어, 온라인 스케줄링)하는 것을 나타낸다.
- 도 16는 일 실시예에 따른 연산 순서 조정된 가중치들에 대한 연산을 수행하는 합성곱 신경망 가속기의 마이크로 아키텍처(micro-architecture)를 나타낸다.
- 도 17는 비교 실시예들에 따른 장치 및 일 실시예에 따른 장치의 에너지 소모를 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0024] 실시예들에 대한 특정한 구조적 또는 기능적 설명들은 단지 예시를 위한 목적으로 개시된 것으로서, 다양한 형태로 변경되어 구현될 수 있다. 따라서, 실제 구현되는 형태는 개시된 특정 실시예로만 한정되는 것이 아니며, 본 명세서의 범위는 실시예들로 설명한 기술적 사상에 포함되는 변경, 균등물, 또는 대체물을 포함한다.
- [0025] 제1 또는 제2 등의 용어를 다양한 구성요소들을 설명하는데 사용될 수 있지만, 이런 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 해석되어야 한다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소는 제1 구성요소로도 명명될 수 있다.
- [0026] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다.
- [0027] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 설명된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함으로써 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0028] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 해당 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

- [0029] 이하, 실시예들을 첨부된 도면들을 참조하여 상세하게 설명한다. 첨부 도면을 참조하여 설명함에 있어, 도면 부호에 관계없이 동일한 구성 요소는 동일한 참조 부호를 부여하고, 이에 대한 중복되는 설명은 생략하기로 한다.
- [0030] 도 1은 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 과정을 나타낸다.
- [0031] 일 실시예에 따른 합성곱 신경망(Convolutional Neural Network; CNN)의 연산 순서 조정 장치는 합성곱 신경망의 적용에 따른 복수의 연산들의 순서를 조정할 수 있다. 합성곱 신경망은 합성곱 레이어를 포함하는 인공 신경망으로서, 다층의 피드-포워드적인 인공 신경망일 수 있고, 데이터(예를 들어, 시각적 영상)를 분석하는 데 사용될 수 있다. 합성곱 신경망의 연산은 복수의 가중치들 및 각 가중치에 대응하는 활성화 값들의 곱셈 및 누적(예를 들어, 복수의 곱셈 누적 연산(multiply and accumulation; MAC))을 포함할 수 있다. 연산 순서 조정 장치는 전술한 복수의 곱셈 누적 연산들의 순서를 조정할 수 있다. 연산 순서를 조정하는 것은 스케줄링으로 표현될 수 있다. 후술하겠으나, 연산 순서를 조정하는 것은 가중치에, 예시적으로 가중치의 비트 레벨 희소성에 기초하는 가중치 스케줄링(가중치 오프라인 스케줄링 또는 오프라인 스케줄링으로도 표현됨)을 포함할 수 있다. 또한, 연산 순서를 조정하는 것은, 활성화 값에 기초하여, 예시적으로 제로 활성화 값의 분포에 기초하는 활성화 값 스케줄링(활성화 값 온라인 스케줄링 또는 온라인 스케줄링으로도 표현됨)을 포함할 수도 있다.
- [0032] 연산 순서 조정 장치는 프로세서를 포함할 수 있다. 예를 들어, 프로세서는 합성곱 신경망의 복수의 가중치들에 대한 연산 순서를 획득할 수 있다. 복수의 가중치들에 대한 연산들은 복수의 사이클들에 걸쳐 수행될 수 있다. 연산 순서는 복수의 레인들의 각각에 배치된 복수의 가중치들 및 해당하는 활성화 값(activation value)(활성화(activation)로도 표현됨)(예를 들어, 이전 레이어로부터 타겟 레이어로 입력되는 활성화 값 또는 입력 레이어로 입력되는 입력 값(input value)) 간의 곱셈 연산이 수행되는 순서를 나타낼 수 있다. 복수의 레인들에 할당된 가중치들 중 같은 사이클에 할당된 가중치들에 대한 MAC 연산은 해당 사이클 동안 병렬적으로 함께 수행될 수 있다.
- [0033] 일 실시예에 따른 가중치들(110)은 복수의 입력 라인들의 각각 및 복수의 출력 라인들의 각각에 대해 설정될 수 있다. 본 명세서에서는 설명의 편의상 합성곱 신경망에 포함된 복수의 레이어들 중 한 대상 레이어의 한 노드에서 수행되는 신경망 연산을 주로 설명하며, 복수의 입력 라인들은 대상 레이어의 해당 노드 및 이전 레이어의 노드들 간의 연결들을 나타내고, 한 출력 라인은 대상 레이어의 해당 노드로 가중합 결과가 전파되는 라인을 나타낼 수 있다. 다시 말해, 복수의 입력 라인을 통해 이전 레이어의 노드로부터 출력된 활성화 값이 대상 레이어의 해당 노드로 전파되고, 노드들 간 연결에 매핑된 가중치가 해당 대응하는 활성화 값에 곱해질 수 있다. 가중합 결과는 전술한 가중치 및 활성화 값 간의 곱들이 누적된 결과일 수 있다. 전술한 복수의 입력 라인들의 개수만큼 복수의 레인들이 배치될 수 있다. 초기 연산 순서는 연산 순서를 조정하기 전의 연산 순서로서, 복수의 레인들의 각각에서 가중치 및 해당하는 활성화 값 간의 곱셈이 수행되는 순서를 나타낼 수 있다.
- [0034] 참고로, 본 명세서에서 초기 연산 순서에서는 복수의 레인들의 각각이 입력 라인에 대응하지만, 연산 순서가 조정된 이후에는 각 레인에서 여러 입력 라인들을 통한 활성화 값들이 수신될 수도 있다. 후술하겠으나, 연산 순서가 조정되면 같은 레인에서 서로 다른 입력 라인들에 대응하는 가중치들이 배치될 수도 있기 때문이다.
- [0035] 가중치들(110)은 각각 제로 가중치, 논-아웃라이어(non-outlier), 및 아웃라이어(outlier) 중 하나로 분류될 수 있다. 제로 가중치는 0의 값을 가지는 가중치를 나타낼 수 있다. 논-아웃라이어는 제2 비트 범위 내의 비트 값들이 모두 0이면서 전술한 제2 비트 범위보다 낮은 제1 비트 범위 내의 비트 값들 중 적어도 하나가 1인 가중치를 나타낼 수 있다. 예시적으로 제1 비트 범위(예를 들어, LSB로부터 4번째까지의 범위)는 임계 비트 위치 이하의 자리 범위를 나타낼 수 있고, 제2 비트 범위(예를 들어, LSB로부터 5번째부터 MSB까지의 범위)는 임계 비트 위치를 초과하는 자리 범위를 나타낼 수 있다. 임계 비트 위치는 예시적으로 가중치의 수치 범위(numerical range) 중 최대값을 표현하기 위해 요구되는 총 비트 개수 중 절반에 대응하는 비트 위치일 수 있으나, 이로 한정하는 것은 아니다. 논-아웃라이어는 제1 비트 정밀도로 표현가능하고, 아웃라이어는 제1 비트 정밀도보다 높은 제2 비트 정밀도로 표현가능할 수 있다. 도 1에서 제로 가중치(zero weight)는 빈칸으로, 논-아웃라이어는 NOL로, 아웃라이어는 OL로 표현된다.
- [0036] 참고로, 비트 정밀도(bit precision)는 가중치의 값을 표현하기 위해 요구되는 정밀도의 척도(measure)로서, 예시적으로 가중치의 값을 표현하기 위해 요구되는 비트들의 개수를 나타낼 수 있다. 예를 들어, 가중치의 값을 표현하기 위하여 1개 내지 4개의 비트를 요구하는 가중치는, 4 비트 정밀도로 표현될 수 있다. 다른 예를 들어, 가중치의 값을 표현하기 위하여 5개 내지 8개의 비트를 요구하는 가중치는 8 비트 정밀도로 표현될 수 있다. 4 비트 정밀도로 표현가능한 가중치는, 예를 들어 상위 비트들을 0으로 채움으로써, 8 비트 정밀도로 표현

될 수도 있다. 이와 달리, 8 비트 정밀도로 표현되는 값(예를 들어, 6개의 비트들을 요구하는 값)은 4 비트 정밀도로 표현되기 어렵다.

- [0037] 제1 비트 정밀도에서 값을 표현하는 데 사용하는 비트 개수의 2배가 제2 비트 정밀도에서 값을 표현하는 데 사용하는 비트 개수 이하인 경우, 2개의 논-아웃라이어가 결합됨으로써 생성된 결합 가중치는 제2 비트 정밀도로 표현가능할 수 있다. 예를 들어, 프로세서는 4 비트 정밀도로 표현가능한 제1 논-아웃라이어 및 제2 논-아웃라이어를 결합함으로써 8 비트 정밀도로 표현가능한 결합 가중치를 생성할 수 있다. 결합 가중치의 상위 비트들(upper bits)은 제1 논-아웃라이어에 대응하고, 결합 가중치의 하위 비트들(lower bits)은 제2 논-아웃라이어에 대응할 수 있다.
- [0038] 참고로, 제로 가중치에 관하여, 제로 가중치 및 활성화 값의 곱셈 결과는 활성화 값과 무관하게 제로이므로, 사이클에 할당된 가중치들에 대한 곱셈 결과들을 모두 덧셈하는 경우에는 제로 가중치에 대한 곱셈 결과가 최종 결과에 영향을 주지 않을 수 있다. 따라서 한 사이클 내에서 모든 레인들의 가중치가 제로 가중치인 경우, 해당 사이클에 대한 연산이 스킵(skip)될 수 있다.
- [0039] 가중치들(110) 중 적어도 일부의 연산 순서는 조정될 수 있다. 연산 순서는 가중치들의 이동, 변경, 결합 및 대체를 통해 조정될 수 있다. 예를 들어, 가중치들(120)에서, 프로세서는 사이클 1의 2개의 논-아웃라이어 가중치들을 결합하여 사이클 0에 할당함으로써 연산 순서를 조정할 수 있다. 다른 예를 들어, 가중치들(120)에서, 프로세서는 사이클 2의 제로 가중치(도 1에서 빈칸으로 표시됨)를 사이클 3의 아웃라이어 가중치로 변경함으로써 연산 순서를 조정할 수 있다. 연산 순서 조정을 위한 가중치들의 변경은 도 6 내지 도 8에서 자세히 후술한다.
- [0040] 연산 순서를 조정함으로써 가중치들(120)에서 사이클 3 및 사이클 1에는 모두 제로 가중치들이 할당될 수 있다. 프로세서는 사이클에 할당된 가중치가 모두 제로인 경우 해당 사이클에서의 연산을 스킵(skip)할 수 있다. 예를 들어, 가중치들(130)에서, 프로세서는 사이클 1 및 사이클 3을 스킵함으로써 사이클 0 및 사이클 2만을 수행함으로써 사이클 0 내지 사이클 3에 대한 연산을 모두 수행하는 것보다 연산의 속도를 향상시킬 수 있다.
- [0041] 도 2는 일 실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 방법의 흐름도를 나타낸다.
- [0042] 단계(210)에서, 프로세서는 복수의 가중치들 중에서 대상 가중치(target weight)를 선택한다. 대상 가중치는 다른 가중치로 변경될 수 있는 가중치를 나타낼 수 있다. 대상 가중치는 제로 가중치 및 논-아웃라이어 가중치 중 하나인 가중치일 수 있다. 대상 가중치의 선택은 도 4에서 자세히 설명한다.
- [0043] 단계(220)에서, 프로세서는 복수의 후보 가중치들 중에서 후보 가중치(candidate weight)를 선택할 수 있다. 후보 가중치는 대상 가중치가 할당된 대상 사이클과 다른 사이클에서 연산이 수행되는 가중치를 포함할 수 있다. 예를 들어, 후보 가중치는 대상 사이클을 기준으로 룩어헤드 윈도우(lookahead window)에 속하는 가중치들 중 적어도 일부를 포함할 수 있다. 선택된 후보 가중치는 대상 가중치와 결합되거나 대상 가중치를 변경함으로써 대상 가중치를 변경하는 데 이용될 수 있다. 후보 가중치의 선택은 도 5에서 자세히 설명한다.
- [0044] 단계(230)에서, 프로세서는 선택된 후보 가중치의 연산 순서를 대상 사이클로 조정할 수 있다. 대상 사이클은 대상 가중치가 속하는 사이클을 나타낼 수 있다. 연산 순서를 조정함으로써, 다른 사이클에 할당되었던 후보 가중치에 대한 연산을 대상 사이클에서 미리 수행할 수 있다. 선택된 후보 가중치에 대한 연산 순서 조정은 도 6 내지 도 8에서 자세히 설명한다.
- [0045] 도 3은 일 실시예에 따른 대상 가중치에 따른 룩어헤드 윈도우 및 후보 가중치를 나타낸다.
- [0046] 룩어헤드 윈도우는 가중치의 이동(예를 들어, 순서 조정 또는 할당)이 가능한 범위로서, 가중치의 이동을 제한하기 위하여 설정될 수 있다. 멀리 떨어진 사이클에 할당된 가중치의 결합 및/또는 변경을 통해 연산 순서가 조정되는 경우, 가중치 및 가중치에 대응하는 활성화 값을 멀리 조정하는 동작에 의하여 파워 및 에너지 오버헤드가 크게 발생할 수 있다. 가중치의 이동을 일정 범위 내로 제한하는 것이 바람직할 수 있다. 대상 사이클을 기준으로 룩어헤드 윈도우에 포함된 가중치들 중 적어도 일부를 대상 가중치에 대한 후보 가중치로 획득할 수 있다.
- [0047] 가중치의 결합 및/또는 변경 가능성을 개선하기 위하여, 룩어헤드 윈도우가 몇 개의 사이클을 포함하는지는 설계에 따라, 예를 들어 멀티플렉서(multiplexer; MUX)의 크기에 따라, 결정될 수 있다. 예를 들어, 가중치들(310)에서, 대상 가중치(311)의 대상 사이클을 기준으로 하는 룩어헤드 윈도우(h1)는 1개의 사이클을 포함할 수 있다. 후보 가중치들(312a, 312b, 312c)은 대상 사이클을 기준으로 룩어헤드 윈도우(h1)에 포함되는 가중치들

중 일부를 포함한다. 다른 예를 들어, 가중치들(330)에서, 대상 가중치(331)의 대상 사이클을 기준으로 하는 룩어헤드 윈도우(h3)는 2개의 사이클들을 포함할 수 있다.

[0048] 룩어헤드 윈도우와 함께 룩어사이드(lookaside)는, 가중치의 이동이 가능한 범위로서 설정될 수 있다. 룩어사이드는 대상 가중치가 할당된 대상 레인을 기준으로 일정 범위 내의 레인들을 포함할 수 있다. 다만, 이에 한정하는 것은 아니고 룩어사이드는, 가중치들(330)에서와 같이, 삼지창 형태의 레인들을 포함할 수 있다. 룩어헤드 윈도우의 사이클들에서 룩어사이드의 레인들에 할당된 가중치들 중 적어도 일부가 후보 가중치로 획득될 수 있다. 대상 가중치가 가장 위 또는 가장 아래의 레인에 할당된 경우, 동일한 크기의 멀티플렉서를 사용할 수 있고 스케줄링 병목(bottleneck)을 방지할 수 있도록, 룩어사이드는 반대쪽 레인(예를 들어, 가장 아래 또는 가장 위의 레인)을 포함할 수 있다. 예를 들어, 가중치들(320)에서, 대상 가중치(321)가 가장 아래의 레인에 할당된 경우, 후보 가중치는 가장 위의 레인에 할당된 가중치(322a)를 포함할 수 있다. 룩어사이드는 설계에 따라, 예를 들어 멀티플렉서의 크기에 따라, 결정될 수 있다.

[0049] 도 4는 일 실시예에 따른 프로세서에 의한 대상 가중치의 선택을 나타낸다.

[0050] 단계(410)에서, 프로세서는 대상 사이클을 선택한다. 예를 들어, 가장 먼저 수행되는 사이클(예를 들어, 사이클 0)이 대상 사이클로 선택될 수 있다. 다른 예를 들어, 연산 순서 조정이 가능한 가중치(예를 들어, 결합 및/또는 변경 가능한 후보 가중치를 갖는 가중치)에 대한 연산이 할당된 사이클 중에서 가장 먼저 수행되는 사이클이 대상 사이클로 선택될 수 있다. 도 4에서, 사이클 0가 대상 사이클(411)로 선택되는 것을 나타낸다.

[0051] 단계(420)에서, 프로세서는 대상 사이클에 할당된 가중치들에 대하여 연산 순서 조정가능한 후보 가중치들의 개수를 산출한다. 해당 가중치에 대하여 연산 순서 조정가능한 후보 가중치들은, 해당 가중치를 변경하는 데 사용될 수 있는 가중치를 나타낼 수 있다. 전술된 후보 가중치들은, 대상 사이클(또는 대상 사이클의 대상 레인)에 할당됨으로써 연산 순서 조정가능한 가중치를 나타낼 수 있다.

[0052] 제로 가중치는 논-아웃라이어 및 아웃라이어 중 하나로 변경될 수 있다. 제로 가중치에 대한 연산은 스킵될 수 있으므로, 제로 가중치의 연산 순서에 다른 가중치에 대한 연산이 수행되도록 조정될 수 있다. 제1 비트 정밀도 및 제2 비트 정밀도를 요구하는 가중치들(예를 들어, 논-아웃라이어 및 아웃라이어)은, 제로 가중치의 연산 순서(예를 들어, 레인 및/또는 사이클)에 다시 할당될 수 있다.

[0053] 논-아웃라이어는 다른 논-아웃라이어와 결합 및 변경될 수 있다. 제1 비트 정밀도로 표현가능한 논-아웃라이어는 다른 논-아웃라이어와 결합됨으로써 제2 비트 정밀도로 표현가능한 결합 가중치가 생성될 수 있다. 논-아웃라이어인 가중치는 결합 가중치로 변경됨으로써 연산 순서가 조정될 수 있다. 논-아웃라이어에 대한 연산 및 다른 논-아웃라이어에 대한 연산이 개별적으로 수행되는 것 대신에, 결합 가중치에 대한 연산이 수행됨으로써 2개의 논-아웃라이어들에 대한 연산들이 한 번에(예를 들어, 한 사이클의 하나의 레인에서) 수행될 수 있다.

[0054] 아웃라이어는 결합 및/또는 변경되지 않을 수 있다. 제2 비트 정밀도로 표현가능한 아웃라이어가 할당된 연산 순서는, 다른 가중치(예를 들어, 논-아웃라이어 또는 다른 아웃라이어)를 표현하는 데 사용될 수 있는 여유 비트를 포함하지 않을 수 있다.

[0055] 전술한 바와 같이, 해당 가중치가 제로 가중치, 논-아웃라이어, 및 아웃라이어 중 해당하는 분류에 기초하여 전술된 후보 가중치의 개수를 산출할 수 있다.

[0056] 일 실시예에 따르면, 룩어헤드 윈도우가 1이고, 룩어사이드가 1로 설정될 수 있다. 대상 사이클이 사이클 0로 선택될 수 있다. 대상 사이클(411)에 속하는 가중치들에 대하여 해당 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치들의 개수를 산출할 수 있다.

[0057] 프로세서는 가중치가 제로 가중치인 경우에 응답하여, 후보 가중치들 중에서 논-아웃라이어 및 아웃라이어를 해당 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치로 결정할 수 있다. 예를 들어, 가중치(421)의 룩어헤드 윈도우 및 룩어사이드는 사이클 1의 레인 0, 레인 1, 및 레인 4일 수 있다. 가중치(421)의 후보 가중치는, 사이클 1의 레인 0, 레인 1, 및 레인 4에 할당된 가중치들($w_{0,1}$, $w_{1,1}$, $w_{4,1}$ (제로 가중치로 도 4에서 빈칸으로 표시됨))을 포함할 수 있다. 프로세서는, 사이클 1의 레인 0의 논-아웃라이어 가중치($w_{0,1}$) 및 레인 1의 논-아웃라이어 가중치($w_{1,1}$)를, 후보 가중치로 결정할 수 있다. 프로세서는 후보 가중치로 결정된 가중치들의 개수를 카운팅할 수 있고, 후보 가중치의 개수(C[0])는 2로 산출될 수 있다.

[0058] 프로세서는 가중치가 논-아웃라이어인 경우에 응답하여, 후보 가중치들 중에서 논-아웃라이어를 해당 가중치에

대한 연산 순서로 할당이 허용되는 후보 가중치로 결정할 수 있다. 예를 들어, 가중치(422)의 특어헤드 윈도우 및 특어사이드는 사이클 1의 레인 0, 레인 1, 및 레인 2일 수 있다. 가중치(422)의 후보 가중치는, 사이클 1의 레인 0, 레인 1, 및 레인 2에 할당된 가중치들($w_{0,1}$, $w_{1,1}$, $w_{2,1}$)일 수 있다. 프로세서는 가중치(422)가 논-아웃라이어 가중치인 경우에 응답하여, 후보 가중치들 중에서 사이클 1의 레인 0의 논-아웃라이어 가중치($w_{0,1}$), 레인 1의 논-아웃라이어 가중치($w_{1,1}$), 레인 1의 논-아웃라이어 가중치($w_{2,1}$)를 가중치(422)에 대한 연산 순서로 할당이 허용되는 후보 가중치로 결정할 수 있다. 프로세서는 후보 가중치로 결정된 가중치들의 개수를 카운팅할 수 있고, 후보 가중치의 개수($C[1]$)는 3으로 산출될 수 있다.

[0059] 프로세서는 가중치가 아웃라이어인 경우에 응답하여, 해당 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치가 존재하지 않는 것으로 결정할 수 있다. 예를 들어, 프로세서는 가중치(423)가 아웃라이어 가중치인 경우에 응답하여, 해당 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치의 개수($C[3]$)를 0으로 산출할 수 있다.

[0060] 단계(430)에서, 프로세서는 산출된 후보 가중치의 개수에 기초하여 대상 가중치를 선택할 수 있다. 예를 들어, 프로세서는 1 이상의 값을 가지는 산출된 후보 가중치의 개수들 중에서 최소 값을 갖는 가중치(421)를 대상 가중치로 선택할 수 있다. 도 4에서와 같이 최소 값을 갖는 가중치들이 여러 개인 경우에는 임의로 하나를 대상 가중치로 선택할 수 있다. 산출된 후보 가중치의 개수가 작을수록 해당 가중치가 결합 및/또는 변경될 수 있는 가능성이 적은 것으로 우선적으로 연산 순서를 조정할 수 있다. 참고로, 산출된 후보 가중치의 개수가 0인 경우, 해당 가중치와 결합 및/또는 변경됨으로써 해당 가중치의 연산 순서로 할당이 허용되는 후보 가중치가 0개 이므로 대상 가중치로 선택되지 않을 수 있다.

[0061] 도 5는 일 실시예에 따른 후보 가중치들 중에서 대상 가중치에 대한 연산 순서로 조정될 후보 가중치를 선택하는 것을 나타낸다. 프로세서는 대상 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치들이 여러 개인 경우, 후보 가중치가 결합 및/또는 변경될 수 있는 레인이 적은 후보 가중치를 선택할 수 있다.

[0062] 단계(510)에서, 프로세서는 하나 이상의 후보 가중치에 대하여 할당이 허용되는 레인의 개수를 산출할 수 있다. 프로세서는, 대상 가중치(511)에 대한 후보 가중치들($w_{0,1}$, $w_{1,1}$) 각각에 대하여 대상 사이클에서 할당이 허용되는 레인의 개수를 산출할 수 있다.

[0063] 사이클 1에서 레인 0에 할당된 논-아웃라이어인 후보 가중치($w_{0,1}$)는, 논-아웃라이어 및 제로 가중치와 결합 및/또는 변경됨으로써 연산 순서가 조정될 수 있다. 구체적으로, 후보 가중치($w_{0,1}$)는 사이클 0에서 레인 0에 할당된 제로 가중치($w_{0,0}$), 레인 1에 할당된 논-아웃라이어인 가중치($w_{1,0}$), 및 레인 4에 할당된 제로 가중치($w_{4,0}$) 중 하나와 결합 및/또는 변경될 수 있다. 후보 가중치($w_{0,1}$)는 사이클 0에서 레인 0, 레인 1, 및 레인 4에 할당이 허용된다. 따라서, 대상 사이클에서 후보 가중치($w_{0,1}$)의 할당이 허용되는 레인의 개수($D[0,1]$)는 3으로 산출될 수 있다. 후보 가중치($w_{1,1}$)는, 사이클 0에서 레인 0에 할당된 제로 가중치($w_{0,0}$) 및 레인 1에 할당된 논-아웃라이어인 가중치($w_{1,0}$)에 대한 연산 순서로 할당될 수 있다. 대상 사이클에서 후보 가중치($w_{1,1}$)의 할당이 허용되는 레인의 개수($D[1,1]$)는 2로 산출될 수 있다.

[0064] 단계(520)에서, 프로세서는 할당이 허용되는 레인의 개수에 기초하여 하나의 후보 가중치를 선택할 수 있다. 예를 들어, 하나 이상의 후보 가중치들($w_{0,1}$, $w_{1,1}$) 중에서 할당이 허용되는 레인의 최소 개수를 갖는 후보 가중치($w_{1,1}$)가 선택될 수 있다. 다만 이에 한정하는 것은 아니고, 프로세서는 가중치의 이동 거리, 가중치의 비트 정밀도 등에 기초하여 후보 가중치를 선택할 수도 있다.

[0065] 도 6는 일 실시예에 따른 선택된 후보 가중치를 대상 가중치에 대한 연산 순서로 할당하는 것을 나타낸다. 도 7는 일 실시예에 따른 대상 가중치가 논-아웃라이어인 경우 연산 순서 조정을 나타낸다. 도 8는 일 실시예에 따른 대상 가중치가 제로 가중치인 경우 연산 순서 조정을 나타낸다.

[0066] 단계(610)에서, 프로세서는 대상 가중치가 제로 가중치인지 또는 논-아웃라이어인지 판단할 수 있다.

[0067] 프로세서는 대상 가중치가 논-아웃라이어인 경우에 응답하여, 결합 가중치를 생성 및 변경함으로써 연산 순서를 조정할 수 있다. 대상 가중치가 논-아웃라이어인 경우, 선택된 후보 가중치는 논-아웃라이어일 수 있다. 단계(620)에서, 대상 가중치(711)와 선택된 후보 가중치(712)를 결합함으로써 결합 가중치(713)를 생성할 수 있다.

단계(630)에서, 프로세서는 대상 가중치를 결합 가중치(713)로 변경함으로써 연산 순서를 조정할 수 있다. 도 4에서 전술한 바와 같이, 프로세서는 결합 가중치(713)에 대한 연산을 수행함으로써, 대상 가중치(711)에 대한 연산 및 선택된 후보 가중치(712)에 대한 연산을 한 번에 수행할 수 있다.

[0068] 프로세서는, 연산 순서를 조정한 후에, 대상 사이클에 할당된 가중치들에 대한 연산 순서 조정 가능한 후보 가중치들의 개수를 업데이트할 수 있다. 예를 들어, 도 7에서 선택된 후보 가중치(712)의 연산 순서 조정으로 인하여, 연산 순서 조정 전에 후보 가중치(712)가 할당이 허용되는 대상 사이클의 라인 0, 라인 1, 및 라인 4에 대한 후보 가중치들의 개수들($C[0]$, $C[1]$, $C[4]$)이 업데이트될 수 있다.

[0069] 단계(640)에서, 프로세서는 대상 가중치가 제로 가중치인 경우에 응답하여, 대상 가중치를 선택된 후보 가중치로 변경함으로써 연산 순서를 조정할 수 있다.

[0070] 대상 가중치가 제로 가중치인 경우, 선택된 후보 가중치는 논-아웃라이어 및 아웃라이어 가중치 중 하나일 수 있다. 도 4에서 전술한 바와 같이, 제로 가중치에 대한 연산은 스킵될 수 있다. 선택된 후보 가중치는 제로 가중치인 대상 가중치를 변경함으로써 대상 가중치의 연산 순서로 할당될 수 있다. 예를 들어, 제로 가중치인 대상 가중치(811)는 논-아웃라이어인 선택된 후보 가중치(812)에 의하여 변경될 수 있다. 다른 예를 들어, 제로 가중치인 대상 가중치(821)는 아웃라이어인 선택된 후보 가중치(822)에 의하여 변경될 수 있다.

[0071] 도 9는 일실시예에 따른 합성곱 신경망의 연산 순서를 조정하는 것을 나타낸다. 알고리즘 1은 일실시예에 따른 합성곱 신경망의 연산 순서 조정을 나타낸다:

[0072] [알고리즘 1]

Algorithm 1 Weight Scheduling

```

1: Initialize  $AS[i, j, k]$ 
2: for  $t = 0$  to  $T - 1$  do
3:   Initialize  $C[0, \dots, L - 1]$  and  $M[0, \dots, L - 1]$ 
4:   if  $w_{x,t}$  for all  $x \in 0, \dots, L - 1 == 0$  then
5:      $B[t] \leftarrow 0$ 
6:     Continue
7:   for  $l = 0$  to  $L - 1$  do
8:      $C[l] \leftarrow$  the number of candidate weights which can be combined to  $w_{l,t}$ 
9:      $M[l] \leftarrow \bigcup$  indices of candidate weights which can be combined to  $w_{l,t}$ 
10:   $C_{min} \leftarrow$  Find the minimum positive number in  $C[0, \dots, L - 1]$ 
11:  while  $C_{min} > 0$  do
12:    for  $l = 0$  to  $L - 1$  do
13:      if  $C[l] == C_{min}$  then
14:        for  $w_{i,j}$  in  $M[l]$  which has priority on movement do
15:          if both  $w_{i,j}$  and  $w_{l,t}$  are non-outlier then
16:            Append  $w_{i,j}$  into  $w_{l,t}$ 
17:             $AS[l, t, 1] \leftarrow$  relative index for  $w_{i,j}$ 
18:            Update  $C[0, \dots, L - 1]$  and  $C_{min}$ 
19:            Break
20:          else if  $w_{l,t}$  is zero then
21:            Swap  $w_{i,j}$  and  $w_{l,t}$ 
22:             $AS[l, t, 0] \leftarrow$  relative index for  $w_{i,j}$ 
23:            if  $w_{i,j}$  is Outlier then
24:               $AS[l, t, 1] \leftarrow$  relative index for  $w_{i,j}$ 
25:            Update  $C[0, \dots, L - 1]$  and  $C_{min}$ 
26:            Break
27: return  $W, AS$ 

```

[0073]

[0074] 알고리즘 1 및 도 9를 함께 아래와 같이 설명한다.

[0075] 알고리즘 1은 가중치들의 스케줄링을 수행하고, 스케줄링된 가중치들에 대응하는 활성화 값들을 매핑하는 데 사용되는 활성화 선택(Activation Selection; AS)을 반환(return)하는 것을 나타낼 수 있다. 알고리즘 1에서, T는 사이클의 총 개수, L은 라인의 총 개수를 나타낼 수 있다.

[0076] 알고리즘 1의 라인 1에서, 활성화 선택($AS[i, j, k]$)를 초기화할 수 있다.

[0077] 알고리즘 1의 라인 2 내지 라인 3에서, 사이클 t에 대하여 각 라인에 할당이 허용되는 후보 가중치들의 개수

(C[1]) 및 할당이 허용되는 후보 가중치들의 인덱스의 집합(M[1])을 초기화할 수 있다.

- [0078] 알고리즘 1의 라인 4 내지 라인 6에서, 사이클 t에 할당된 가중치들이 모두 제로인 경우에 응답하여, 사이클 t에 대한 비트 벡터(B[t])가 0으로 결정될 수 있다. 합성곱 신경망의 연산을 수행하는 장치는, 비트 벡터가 0인 것에 응답하여, 사이클 t에 대한 연산을 스킵할 수 있다.
- [0079] 단계(910)에서, 프로세서는 해당 가중치($w_{i,t}$)는 아웃라이어, 논-아웃라이어, 및 제로 가중치 중 하나로 분류함으로써, 해당 가중치($w_{i,t}$)의 연산 순서로 할당이 허용되는 후보 가중치(w_c)의 타입(type)을 결정할 수 있다. 단계(911)에서, 프로세서는 해당 가중치($w_{i,t}$)가 아웃라이어인지 판단할 수 있다. 해당 가중치($w_{i,t}$)가 아웃라이어가 아닌 경우에 응답하여, 해당 가중치($w_{i,t}$)가 제로 가중치인지 판단될 수 있다. 단계(912)에서, 해당 가중치($w_{i,t}$)가 제로 가중치인 경우에 응답하여, 후보 가중치(w_c)의 타입은 논-제로(예를 들어, 논-아웃라이어 및 아웃라이어)로 결정될 수 있다. 단계(913)에서, 해당 가중치($w_{i,t}$)가 제로 가중치가 아닌 경우에 응답하여, 후보 가중치(w_c)의 타입은 논-아웃라이어로 결정될 수 있다. 단계(914)에서, 결정된 후보 가중치의 타입에 기초하여, 후보 가중치들 중에서 해당 가중치($w_{i,t}$)에 대한 연산 순서로 할당이 허용되는 후보 가중치의 개수(C[1]) 및 할당이 허용되는 후보 가중치들의 인덱스들의 집합(M[1])을 업데이트할 수 있다. 알고리즘 1의 라인 7 내지 라인 9에서, 예를 들어 전술된 단계(910)에 따라, 사이클 t에 할당된 가중치들에 중 라인 1에 할당이 허용되는 후보 가중치들의 개수(C[1]) 및 할당이 허용되는 후보 가중치들의 인덱스의 집합(M[1])은 결정될 수 있다.
- [0080] 단계(920)에서, 프로세서는 최소 후보 가중치의 개수(C[1])를 갖는 레인을 선택할 수 있다. 선택된 레인에 할당된 가중치($w_{i,t}$)는 대상 가중치로 선택될 수 있다. 알고리즘 1의 라인 10 내지 라인 13에서, 후보 가중치의 개수의 최소 값(C_{min})은 획득될 수 있다. 후보 가중치의 개수의 최소 값이 0보다 큰 경우에, 사이클 t에서의 연산 순서 조정이 수행될 수 있다. 라인 1의 후보 가중치의 개수(C[1])가 최소 값(C_{min})인 경우에 응답하여, 해당 레인에 할당된 대상 가중치에 대한 연산 순서가 조정될 수 있다.
- [0081] 단계(930)에서, 후보 가중치들 중에서 결합될 가능성(chance)이 가장 적은 한 후보 가중치는 탐색(find)될 수 있다. 알고리즘 1의 라인 14에서, 대상 가중치에 대한 연산 순서로 할당이 허용되는 후보 가중치들의 인덱스의 집합(M[1])에서 이동(movement)에 우선권(priority)을 갖는 한 후보 가중치($w_{i,j}$)를 선택할 수 있다.
- [0082] 단계(940)에서, 대상 가중치에 대한 연산 순서를 조정할 수 있다. 대상 가중치에 대한 연산 순서는, 대상 가중치가 제로 가중치인지에 따라 다음과 같이 수행될 수 있다.
- [0083] 단계(941)에서, 대상 가중치가 제로 가중치가 아닌 경우에 응답하여, 2개의 논-아웃라이어들인 후보 가중치($w_{i,t}$) 및 선택된 후보 가중치를 결합할 수 있다. 예를 들어, 알고리즘 1의 라인 15 내지 라인 16에서, 대상 가중치($w_{i,t}$) 및 후보 가중치($w_{i,j}$)가 모두 논-아웃라이어인 경우에 응답하여, 대상 가중치에 후보 가중치가 부가(append)될 수 있다. 알고리즘 1의 라인 17에서, 대상 가중치의 상위 비트를 위한 활성화 선택(AS[1, t, 1])는 선택된 후보 가중치($w_{i,j}$)에 대한 상대적인 인덱스로 결정될 수 있다. 알고리즘 1의 라인 18에서, 사이클 t에 대한 각 레인에 할당이 허용되는 후보 가중치의 개수(C[0, ..., L-1]) 및 후보 가중치의 개수의 최소 값(C_{min})은 업데이트될 수 있다.
- [0084] 단계(942)에서, 프로세서는 대상 가중치가 제로 가중치인 경우에 응답하여, 논-제로인 후보 가중치를 페칭(fetch)할 수 있다. 예를 들어, 알고리즘 1의 라인 20 내지 라인 22에서, 대상 가중치($w_{i,t}$) 및 선택된 후보 가중치($w_{i,j}$)를 스왑(swap)할 수 있다. 대상 가중치의 하위 비트를 위한 활성화 선택(AS[1, t, 0])는 선택된 후보 가중치($w_{i,j}$)에 대한 상대적인 인덱스로 결정될 수 있다. 알고리즘 1의 라인 23 내지 라인 25에서, 선택된 후보 가중치($w_{i,j}$)가 아웃라이어인 경우에 응답하여, 대상 가중치의 상위 비트를 위한 활성화 선택(AS[1, t, 1])은 선택된 후보 가중치($w_{i,j}$)에 대한 상대적인 인덱스로 결정될 수 있다. 알고리즘 1의 라인 25에서, 후보 가중치의 개수(C[0, ..., L-1]) 및 후보 가중치의 개수의 최소 값(C_{min})은 업데이트될 수 있다.
- [0085] 도 10a 및 도 10b는 일 실시예에 따른 조정된 연산 순서를 위한 재구성가능한 곱셈기(reconfigurable Multiplier)를 나타낸다. 합성곱 신경망에서, 하나의 사이클에 할당된 가중치들에 대한 곱셈들의 결과는 덧셈될 수 있다. 가중치들에 대한 곱셈은 곱셈기에 의하여 수행될 수 있다. 누산기(accumulator)는 복수의 곱셈기

들로부터의 곱셈 결과들을 덧셈하기 위하여 사용될 수 있다.

- [0086] 도 10a에서, 조정된 연산 순서에 따라 복수의 가중치들에 대한 연산들을 수행하기 위하여, 가중치에 대응하는 활성화 값이 곱셈될 수 있도록 재구성가능한 곱셈기를 설계할 수 있다. 예를 들어, 룩어헤드가 1이고 룩어사이드가 1인 경우, 대상 가중치와 결합 및/또는 변경될 수 있는 후보 가중치는 3개일 수 있다. 대상 가중치 및 3개의 후보 가중치에 대응하는 4개의 활성화 값들 중 해당 연산 순서에서 수행될 연산에 대응하는 활성화 값을 4:2 멀티플렉서를 사용하여 선택할 수 있다. 4:2 멀티플렉서는 활성화 값에 대응하는 활성화 선택(activation selection; AS)에 기초하여 활성화 값을 선택할 수 있다.
- [0087] 도 10b에서, 재구성가능한 곱셈기는 가중치 및 활성화 값을 획득할 수 있다. 가중치는 제2 비트 정밀도로 표현될 수 있다. 재구성가능한 곱셈기는 상위 비트에 대응하는 가중치(W_1)(예를 들어, 제1 비트 정밀도로 표현됨) 및 하위 비트에 대응하는 가중치(W_2)(예를 들어, 제1 비트 정밀도로 표현됨)를 획득함으로써 전체 가중치(W)를 획득할 수 있다. 활성화 값은 가중치(W_1)에 대응하는 활성화 값(A_1) 및 가중치(W_2)에 대응하는 활성화 값(A_2)으로 획득될 수 있다.
- [0088] 가중치(W)가 아웃라이어인 경우, 상위 비트에 대응하는 가중치(W_1) 및 하위 비트에 대응하는 가중치(W_2)의 조합은 하나의 가중치를 표현할 수 있다. 상위 비트에 대응하는 가중치(W_1) 및 대응하는 활성화 값(A_1)의 곱셈은 시프트 연산을 수행할 수 있다. 누산기는 가중치(W_1) 및 활성화 값(A_1)의 곱셈 결과에 시프트 연산이 수행된 결과, 및 가중치(W_2) 및 활성화 값(A_2)의 곱셈 결과를 획득할 수 있다.
- [0089] 가중치(W)가 2개의 논-아웃라이어가 결합된 결합 가중치인 경우, 상위 비트에 대응하는 가중치(W_1) 및 하위 비트에 대응하는 가중치(W_2)는 각각 별개의 논-아웃라이어 가중치를 표현할 수 있다. 각각 별개의 가중치에 대한 연산이므로, 시프트 연산은 가중치(W_1) 및 활성화 값(A_1)의 곱셈 결과에 대하여 수행되지 않을 수 있다. 누산기는 가중치(W_1) 및 활성화 값(A_1)의 곱셈 결과, 및 가중치(W_2) 및 활성화 값(A_2)의 곱셈 결과를 획득할 수 있다.
- [0090] 도 11은 일 실시예에 따른 속도 개선 및 가중치 희소성(sparsity)의 관계를 나타낸다.
- [0091] 제로 가중치의 비율 및 논-아웃라이어 가중치의 비율은 합성곱 신경망의 레이어들 간에 다를 수 있다. 레이어에서의 제로 가중치의 비율에 따라 비교 실시예(예를 들어, 제로 가중치의 스킵)의 연산 순서 조정의 성능이 다를 수 있다. 제로 가중치 스킵만을 적용하는 경우, 희소성이 높은 레이어(예를 들어, 제로 가중치의 비율이 큰 레이어)에 대한 성능 향상이 클 수 있다. 밀집도가 높은 레이어(예를 들어, 제로 가중치의 비율이 작은 레이어)에 대한 성능 향상이 작을 수 있다. 이와 달리, 일 실시예는 밀집도가 높은 레이어도 논-아웃라이어 가중치의 비율이 높다면 성능 향상이 클 수 있다.
- [0092] 도 12는 비교 실시예 및 일 실시예에 따른 레이어마다 제로 가중치, 논-아웃라이어 가중치의 비율, 및 속도 개선을 나타낸다.
- [0093] 비교 실시예는 제로 가중치를 스킵하는 방법에 따른 실시예일 수 있다. 성능 향상의 정도를 확인하기 위하여 VGG-16 네트워크에 대하여 실험이 진행될 수 있다. VGG-16 네트워크는 CIFAR-100 트레이닝 세트로 트레이닝될 수 있다. 가중치는 8 비트, 활성화 값은 16 비트 고정 소수점으로 양자화되고 미세 조정(fine tuning) 과정이 수행될 수 있다. 도 12에서, 일 실시예는 필터의 희소성(예를 들어, 제로 가중치의 비율)이 증가함에 따라 속도 개선이 나타날 수 있다. 일 실시예는 모든 레이어에서 비교 실시예보다 더 큰 속도 개선을 가질 수 있다. 논-아웃라이어 가중치의 비율이 큰 경우, 비교 실시예는 논-아웃라이어 가중치에 대한 속도 개선을 가지지 않을 수 있지만, 일 실시예는 논-아웃라이어의 결합을 통한 속도 개선을 가질 수 있다. 따라서, 논-아웃라이어 가중치의 비율이 큰 경우, 비교 실시예보다 일 실시예에서 속도 개선이 훨씬 더 클 수 있다.
- [0094] 도 13는 비교 실시예 및 일 실시예에 따른 4개의 합성곱 신경망들 및 4개의 멀티플렉서들의 크기에 따른 성능 개선을 나타낸다. 4개의 합성곱 신경망들(예를 들어, VGG-16, ResNet-18, SqueezeNext, 및 MobileNet) 및 4개의 멀티플렉서 크기(예를 들어, 4, 8, 12, 및 16)에서 일 실시예는 비교 실시예보다 더 큰 속도 개선을 가질 수 있다.
- [0095] 도 14는 일 실시예에 따른 클리핑 기법에 따른 정확도 및 성능 향상의 관계를 나타낸다.
- [0096] 아웃라이어인 가중치는 다른 가중치와 결합 및/또는 변경되지 않을 수 있다. 논-아웃라이어인 가중치는 다른

가중치와 결합 및/또는 변경될 가능성을 가질 수 있다. 아웃라이어인 가중치가 논-아웃라이어인 가중치로 전환될 수 있다면, 일 실시예는 추가적인 속도 개선을 가질 수 있다.

[0097] 클리핑 기법은 제1 비트 정밀도로 표현가능한 범위에서 임계 이내로 벗어난 아웃라이어 가중치를 논-아웃라이어 가중치로 전환(예를 들어, 편입 또는 변경)시키기 위해 수행될 수 있다. 예를 들어, 제1 비트 정밀도가 4 비트 정밀도인 경우, 다음의 수학적 식 1에 기초하여 클리핑 기법이 수행될 수 있다:

[0098] [수학적 식 1]

$$W_{clip} = \begin{cases} -2^3 & \text{if } (-2^3 - threshold \leq W < -2^3) \\ 2^3 - 1 & \text{if } (2^3 \leq W < 2^3 + threshold) \\ W, & \text{otherwise} \end{cases} \quad (1)$$

[0100] 제1 비트 정밀도로 표현가능한 범위가 -2^3 이상 및 $2^3 - 1$ 미만일 때, 가중치(W)가 제1 비트 정밀도로 표현 가능한 범위에서 벗어난 정도가 임계(threshold) 이내인 경우, 가중치(W)는 클리핑됨으로써(예를 들어, 가중치(W)에 대하여 클리핑 기법이 수행됨으로써) 생성된 가중치(W_{clip})는 논-아웃라이어로 편입될 수 있다. 다시 말해, 아웃라이어인 가중치(W)가 임계 범위에 포함되는 경우에 응답하여, 수학적 식 1에 기초하여 논-아웃라이어인 가중치(W_{clip})로 변경될 수 있다.

[0101] 일 실시예에 따르면, VGG-16 네트워크에서 클리핑 기법은 사용될 수 있다. 그래프(1410)에서, 클리핑 기법은 정확도와 성능 향상의 트레이드-오프 관계를 가질 수 있다. 정확도는 임계 값이 증가함에 따라 감소할 수 있다. 프로세서는 합성곱 신경망(예를 들어, VGG-16 네트워크) 클리핑된 가중치에 기초하여 다시 트레이닝(retraining)시킬 수 있다. 그래프(1410)에서, 합성곱 신경망은 다시 트레이닝됨으로써, 정확도를 회복할 수 있다. 그래프(1420)에서, 정확도 감소의 경계(Accuracy Loss Boundary)가 1%로 설정되는 경우, VGG-16 네트워크는 6%의 추가적인 속도 개선을 가질 수 있다.

[0102] 다음의 표 2는 4개의 합성곱 신경망에서 서로 다른 임계들에 대하여 클리핑 기법에 따른 정확도 감소(Accuracy Drop) 및 추가적인 속도 개선(Additional Speedup)을 나타낸다:

[0103] [표 2]

Fig. 8: (a) Accuracy reduction and (b) additional performance improvement by clipping outlier weights of VGG-16.

TABLE II: Accuracy reduction and speedup by clipping outliers.

Accuracy Drop							
Network	Lossless	Threshold					
		1	2	3	4	5	6
VGG-16	72.8	-0.3	-0.4	-0.7	-0.9	-0.9	-1.5
ResNet-18	70.6	-0.1	-0.3	-0.3	-0.6	-0.9	-0.9
SqueezeNext	71.2	0.0	-0.1	-0.2	-0.4	-0.6	-0.9
MobileNet	73.0	0.0	-0.6	-0.5	-1.0	-1.8	-2.2

Additional Speedup							
Network	Lossless	Threshold					
		1	2	3	4	5	6
VGG-16	2.11	2.13	2.15	2.16	2.17	2.17	2.18
ResNet-18	2.13	2.16	2.19	2.2	2.21	2.22	2.23
SqueezeNext	3.39	3.42	3.45	3.47	3.49	3.5	3.51
MobileNet	2.71	2.76	2.79	2.83	2.86	2.9	2.92

[0104] 4개의 합성곱 신경망들(예를 들어, VGG-16, ResNet-18, SqueezeNet, 및 MobileNet)은 클리핑 기법을 수행하지 않는 경우(표 2에서 Lossless로 표시됨)보다 클리핑 기법을 수행한 경우에 추가적인 속도 개선을 가질 수 있다.

- [0106] 도 15는 일 실시예에 따른 활성화 값의 희소성을 이용하여 실시간으로 연산 순서를 조정(예를 들어, 온라인 스케줄링)하는 것을 나타낸다. 활성화 값이 제로인 경우에는 가중치의 값과 무관하게 가중치 및 활성화 값의 곱셈 결과가 제로이므로, 연산 결과(예를 들어, 곱셈의 결과들에 대한 덧셈의 결과)에 영향을 미치지 않을 수 있다. 해당 사이클에서 수행되는 연산의 활성화 값들이 모두 제로인 경우, 해당 사이클은 스킵될 수 있다.
- [0107] 가중치는 연산 순서 조정에 따라 결합 및/또는 변경됨으로써 재할당될 수 있다. 이와 달리, 활성화 값은 활성화 선택에 따라 조절될 뿐이고 재할당되지 않을 수 있다. 해당 사이클에 할당된 가중치들은 해당 사이클이 아닌 다른 사이클에서의 활성화 값에 대응할 수 있다. 예를 들어, 기존에 사이클 1에 할당된 가중치가 연산 순서 조정을 통해 사이클 0의 가중치와 결합 및/또는 변경된 경우, 사이클 0에 할당된 연산은 기존 사이클 1에 할당되었던 가중치에 대응하는 사이클 1의 활성화 값을 사용해야 할 수 있다.
- [0108] 해당 사이클에 할당된 가중치들은 연산 조정에 의하여 해당 사이클을 기준으로 롱어헤드 윈도우에 속하는 가중치들이 결합 및/또는 변경될 수 있다. 해당 사이클의 활성화 값들이 모두 제로 가중치인 것만으로 바로 해당 사이클을 스킵할 수 없을 수 있다. 해당 사이클 및 해당 사이클을 기준으로 롱어헤드 윈도우의 활성화 값들이 모두 제로인 경우, 해당 사이클(예를 들어, 가장 선행하는 사이클)은 스킵될 수 있다.
- [0109] 가중치 비트 벡터(wight bit vector)는, 연산 조정에 의하여 해당 사이클에 할당된 가중치들이 모두 제로 가중치인 경우 0, 하나 이상의 논-제로 가중치(예를 들어, 논-아웃라이어 가중치 및 아웃라이어 가중치 중 하나)가 포함된 경우 1을 포함한다. 가중치 비트 벡터에 비트 별 NOT 연산을 수행함으로써 비효과적인 가중치 지시자(ineffectual weight indicator)가 획득될 수 있다. 비효과적인 가중치 지시자는 가중치에 기초하여 해당 사이클이 스킵될 수 있는지 여부를 나타낼 수 있다. 예를 들어, 해당 사이클에 대한 비효과적인 가중치 지시자의 원소가 0인 경우, 해당 사이클에 할당된 가중치들은 모두 제로 가중치이므로, 해당 사이클은 스킵될 수 있다. 해당 사이클에 대한 비효과적인 가중치 지시자의 원소가 1인 경우, 해당 사이클에 할당된 가중치들 중 적어도 하나는 논-제로 가중치 이므로, 해당 사이클은 스킵되기 어려울 수 있다.
- [0110] 활성화 비트 벡터(activation bit vector)는 해당 사이클에 할당된 활성화 값이 모두 제로인 경우 0, 하나 이상의 논-제로인 활성화 값이 포함된 경우 1을 포함한다. 해당 사이클 및 해당 사이클을 기준으로 롱어헤드 윈도우에 대응하는 활성화 비트 벡터의 원소 값이 모두 제로인 경우에 응답하여, 해당 사이클에 대응하는 원소가 1인 연속 비효과적인 활성화 지시자(consecutive ineffectual activation indicator)가 획득될 수 있다. 연속 비효과적인 활성화 지시자는 활성화 값에 기초하여 해당 사이클이 스킵될 수 있는지 여부를 나타낼 수 있다. 예를 들어, 해당 사이클에 대한 연속 비효과적인 활성화 지시자의 원소가 1인 경우, 해당 사이클 및 해당 사이클을 기준으로 롱어헤드 윈도우에 대응하는 활성화 값들이 모두 제로이므로, 해당 사이클은 스킵될 수 있다.
- [0111] 따라서, 해당 사이클에 대한 비효과적인 가중치 지시자의 원소 및 연속 비효과적인 활성화 지시자의 원소 중 적어도 하나가 1인 경우 해당 사이클은 스킵될 수 있다. 비효과적인 가중치 지시자 및 연속 비효과적인 활성화 지시자가 모두 0인 경우, 해당 사이클은 스킵되기 어려운(예를 들어, 스킵될 수 없는) 유효 사이클(effectual cycle)으로 결정될 수 있다. 일 실시예에 따른 스케줄러는 두 개의 지시자들의 NOR 연산을 통해 유효 사이클들을 획득할 수 있다.
- [0112] 가중치 및 활성화 값은 획득된 유효 사이클들에 기초하여 압축될 수 있다. 온라인 스케줄러는 압축된 가중치 및 활성화 값에 대한 입력 순서를 결정할 수 있다. 활성화 값의 희소성을 이용하여 온라인 스케줄링을 수행함으로써 4개의 합성곱 신경망들은 다음의 표 4와 같은 속도 개선을 가질 수 있다:

[표 3]

	VGG-16	ResNet-18	SqueezeNext	MobileNet
실시예 1	2.33	2.46	5.34	3.22
실시예 2	2.50	2.76	7.29	5.33

[0114]

- [0115] 실시예 1은, 가중치의 결합 및/또는 변경을 통한 연산 순서 조정(가중치 스케줄링, 가중치 오프라인 스케줄링, 또는 오프라인 스케줄링으로도 표현됨)을 수행할 수 있다. 실시예 2는 가중치의 결합 및/또는 변경을 통한 연산 순서 조정(예를 들어, 가중치 스케줄링)과 함께 제로 활성화 값에 기초한 스케줄링(활성화 값 스케줄링, 제로 활성화 값 스킵, 또는 온라인 스케줄링으로도 표현됨)을 수행할 수 있다.

[0116] 도 16는 일 실시예에 따른 연산 순서 조정된 가중치들에 대한 연산을 수행하는 합성곱 신경망 가속기의 마이크로 아키텍처(micro-architecture)를 나타낸다.

[0117] 합성곱 신경망 가속기는 하드웨어 연산 요소(Processing element) 구조일 수 있다. 하드웨어 연산 요소는 선택 버퍼(select buffer), 활성화 값 선택기(activation selector), 가중치 버퍼(wight buffer), 하나 이상의 누산기, 활성화 값 메모리(activation memory), 가중치 메모리(weight memory), 및 곱셈기 배열(Multiplier Array)를 포함할 수 있다.

[0118] 연산 순서 조정된 가중치들 및 가중치들에 대한 연산 순서는 가중치 메모리에 저장될 수 있다. 활성화 값은 활성화 값 메모리에 저장될 수 있다. 활성화 값 선택기는, 선택 버퍼로부터 획득된 활성화 선택에 기초하여, 하나 이상의 멀티플렉서들을 이용하여 가중치에 대응하는 활성화 값을 선택할 수 있다.

[0119] 각 곱셈기 배열(1611)은 하나 이상의 곱셈기들(예를 들어, 도 16에서, N개의 곱셈기들)을 포함할 수 있다. 2개의 비트 정밀도(예를 들어, 제1 비트 정밀도 및 제2 비트 정밀도)로 표현되는 가중치들(예를 들어, 논-아웃라이어 및 아웃라이어)에 대한 연산을 수행하기 위하여, 곱셈기(1621)는 2개의 서브-곱셈기들(1622) 및 시프트 연산기(1623)를 포함할 수 있다.

[0120] 온라인 스케줄러는, 도 15에서 전술된 바에 따른 활성화 값의 희소성을 이용하여 실시간 연산 순서 조정(예를 들어, 온라인 스케줄링)을 수행할 수 있다. 온라인 스케줄러는 온라인 스케줄링에 따라 해당 사이클의 스킵 여부를 결정할 수 있다. 가중치 및 활성화 값은, 온라인 스케줄러의 온라인 스케줄링에 따라, 제어될 수 있다.

[0121] 도 17는 비교 실시예들에 따른 장치 및 일 실시예에 따른 장치의 에너지 소모를 나타낸다.

[0122] 연산 순서 조정에 따른 합성곱 신경망 가속기는 베릴로그 HDL을 사용하여 표 4와 같은 설계 파라미터(design parameter)로 설계될 수 있다:

[0123] [표 4]

TABLE IV: Hardware configurations of the baseline accelerator, the zero-skip accelerator and the proposed accelerator.

Parameters			
Filters	8	Lanes	8
Lookahead Buffers	2	Size of multiplexer	8
Precision of Weight	8 bits	Precision of Activation	16 bits
Global Buffer			
Weight Memory	2MB	Activation Memory	4MB
Local Buffer			
Baseline	3KB	Zero-skip & Proposed	3KB

[0124]

[0125] RTL 코드는 1 GHz 주파수에서 TSMC 65-nm GP cell 라이브러리와 Synopsys Design Compiler를 이용하여 합성될 수 있다. 에너지 소모는 동적 전력(Dynamic power) 및 누설 전력(Leakage power)의 합계로 산출될 수 있다. 비교 실시예 1, 비교 실시예 2, 및 일 실시예 각각의 합성곱 신경망의 총 면적을 동일한 그룹(예를 들어, 로컬 버퍼(local buffer), MAC(Multiply and Accumulate), 및 주변 회로(peripheral circuit))으로 나타내면 다음의 표 5와 같다:

[0126] [표 5]

	Local Buffer	MAC	Peripheral Circuit	Total Area
비교 실시예 1	0.50	0.37	0.12	1.00
비교 실시예 2	1.01	0.37	0.14	1.52
일 실시예	1.16	0.39	0.22	1.77

[0127]

[0128] 비교 실시예 1은 기본 가속기 구조일 수 있다. 비교 실시예 2는 제로 가중치를 스킵하는 하드웨어 가속기인 제로 스킵 구조일 수 있다. 일 실시예는, 비교 실시예 2에 비하여 총 면적(Total Area)이 1.16배만큼 증가할 수

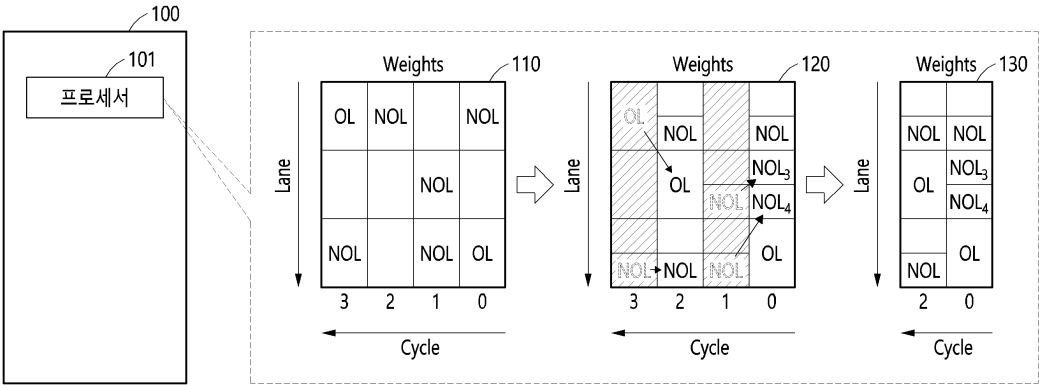
있다.

- [0129] 도 17에서, 활성화 값 선택기, 주변 회로, MAC, 및 로컬 버퍼에 의하여 각각 사용되는 에너지가 나타난다. 로컬 버퍼가 상당한 에너지(예를 들어, 전력)를 소모할 수 있다. 활성화 값 선택기 및 곱셈기(예를 들어, MAC)로 인하여 발생하는 에너지소모(예를 들어, 전력 소모)는, 로컬 버퍼가 소모하는 에너지에 비하여 무시될 수 있는 수준일 수 있다.
- [0130] 비교 실시예 2는 SqueezeNext와 MobileNet의 경우 기본 가속기 구조보다 더 작은 에너지 소모를 가지는 것과 달리, VGG-16과 ResNet-18의 경우 기본 가속기 구조보다 더 큰 에너지 소모를 가질 수 있다. 비교 실시예 2의 성능 향상이 적은 것에 비해 큰 크기의 로컬 버퍼(Local buffer)를 요구할 수 있다. 반면에 일 실시예는 더 큰 성능 향상을 가질 수 있으므로, 에너지 소모를 비교 실시예 2보다 더 감소시킬 수 있다. 일 실시예는 곱셈기의 비트 정밀도에 기초하여 연산 순서를 조정함으로써, 비교 실시예 1 및 비교 실시예 2보다 각각 평균 29.3% 및 30.2% 감소된 에너지 소모를 가질 수 있다.
- [0131] 이상에서 설명된 실시예들은 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치, 방법 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [0132] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [0133] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있으며 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.
- [0134] 위에서 설명한 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 또는 복수의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0135] 이상과 같이 실시예들이 비록 한정된 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 이를 기초로 다양한 기술적 수정 및 변형을 적용할 수 있다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

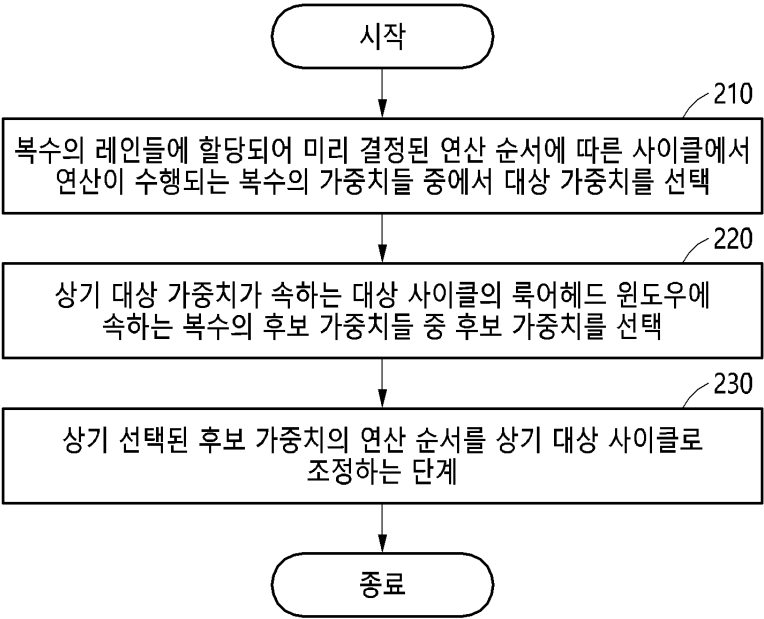
[0136] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

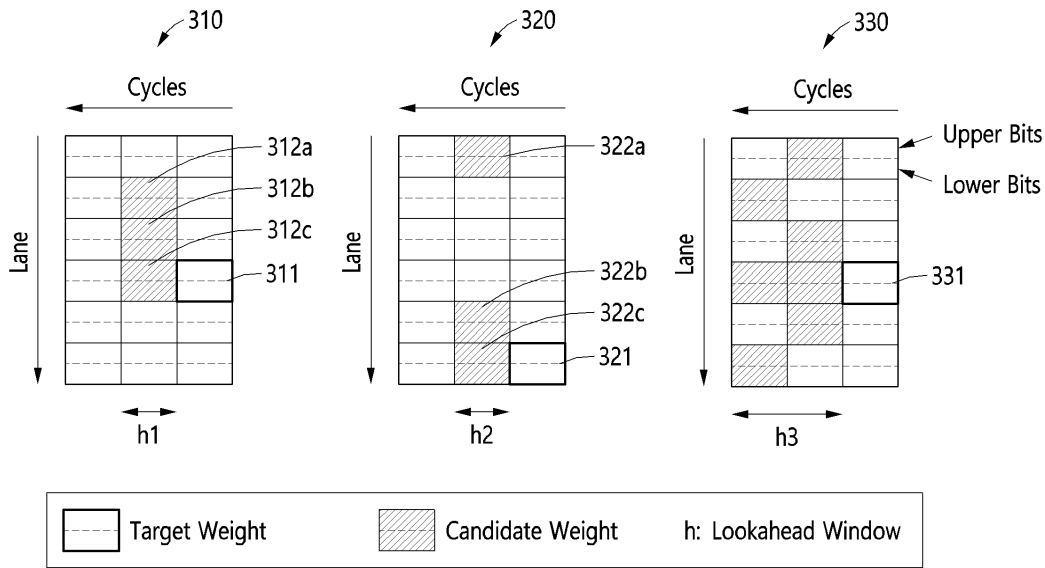
도면1



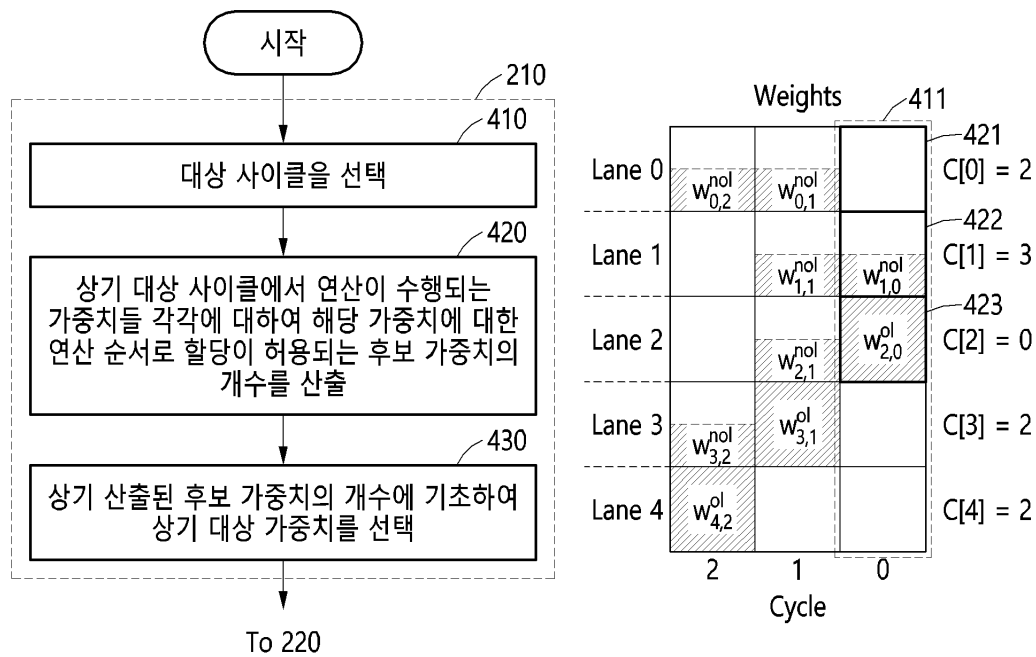
도면2



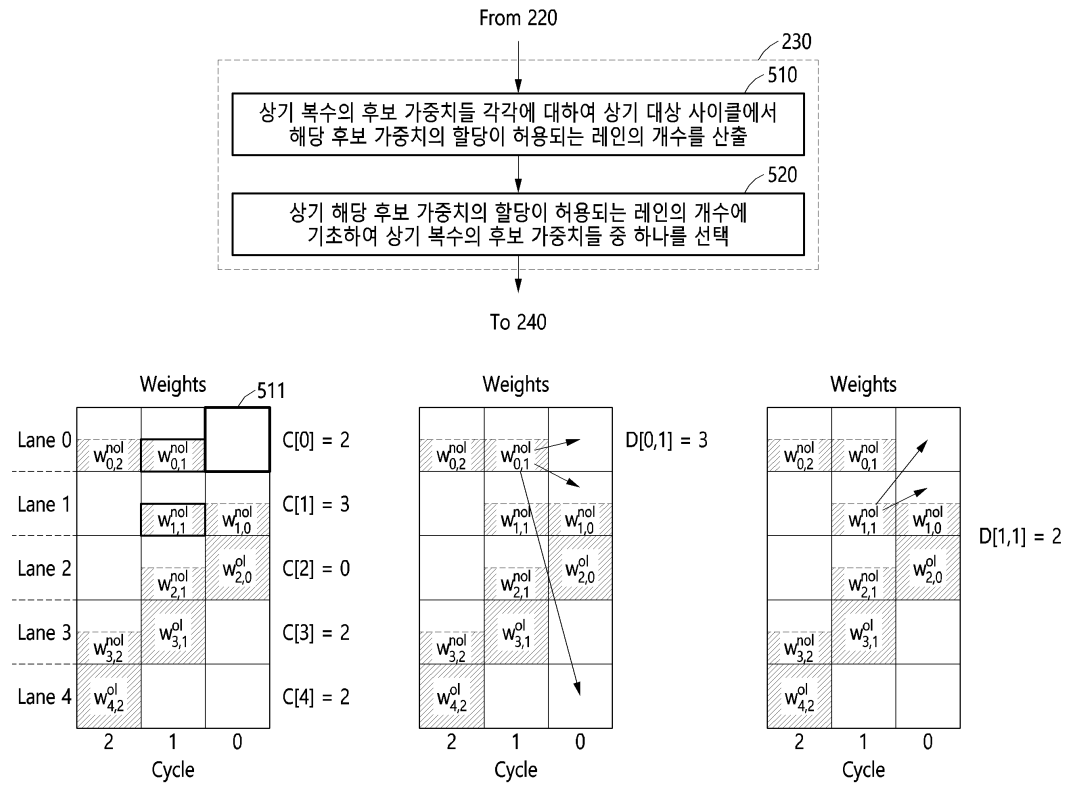
도면3



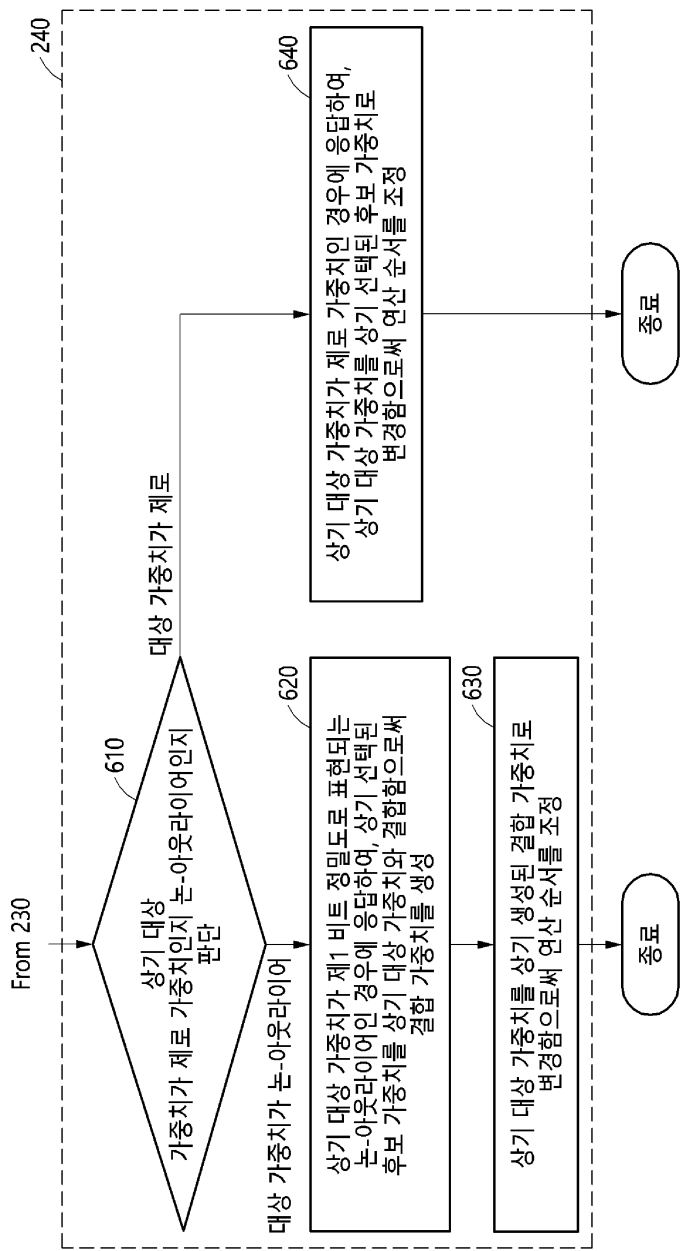
도면4



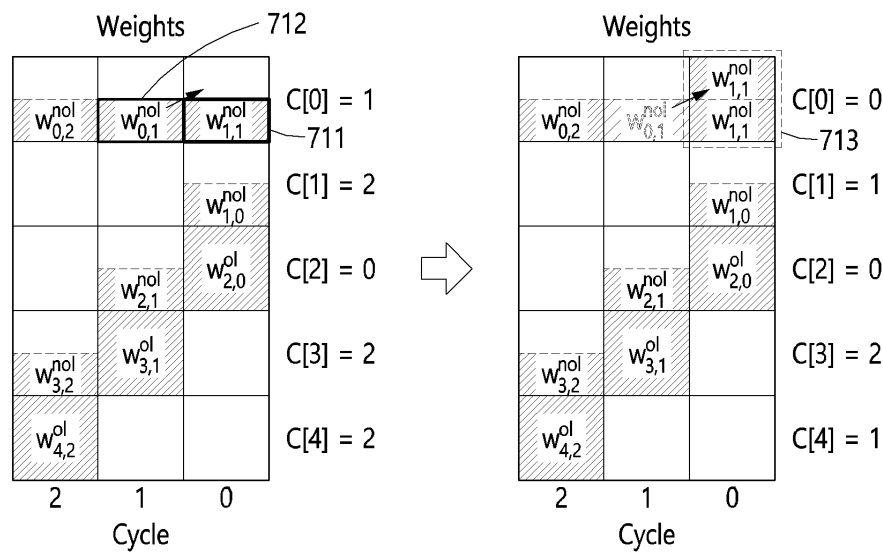
도면5



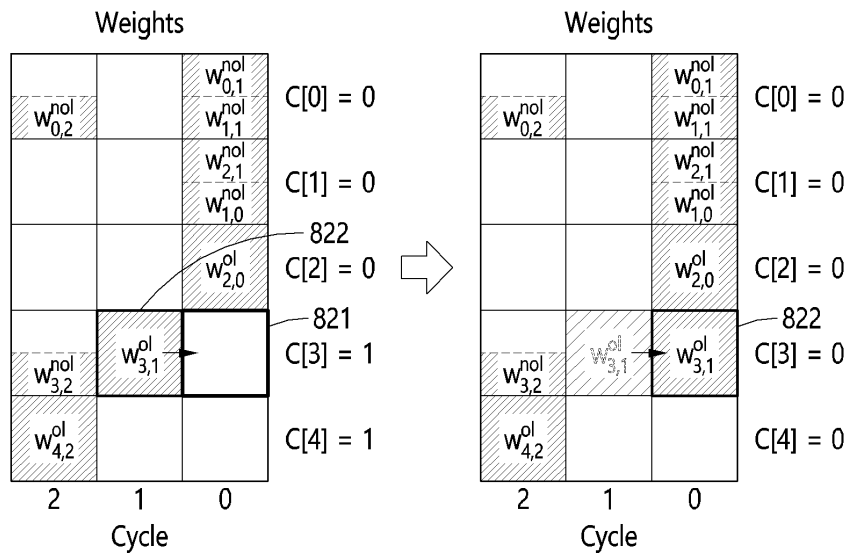
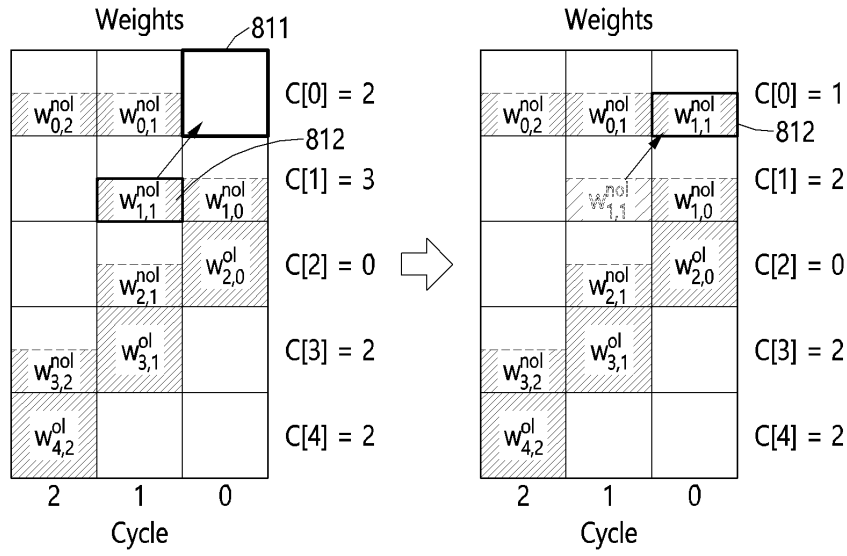
도면6



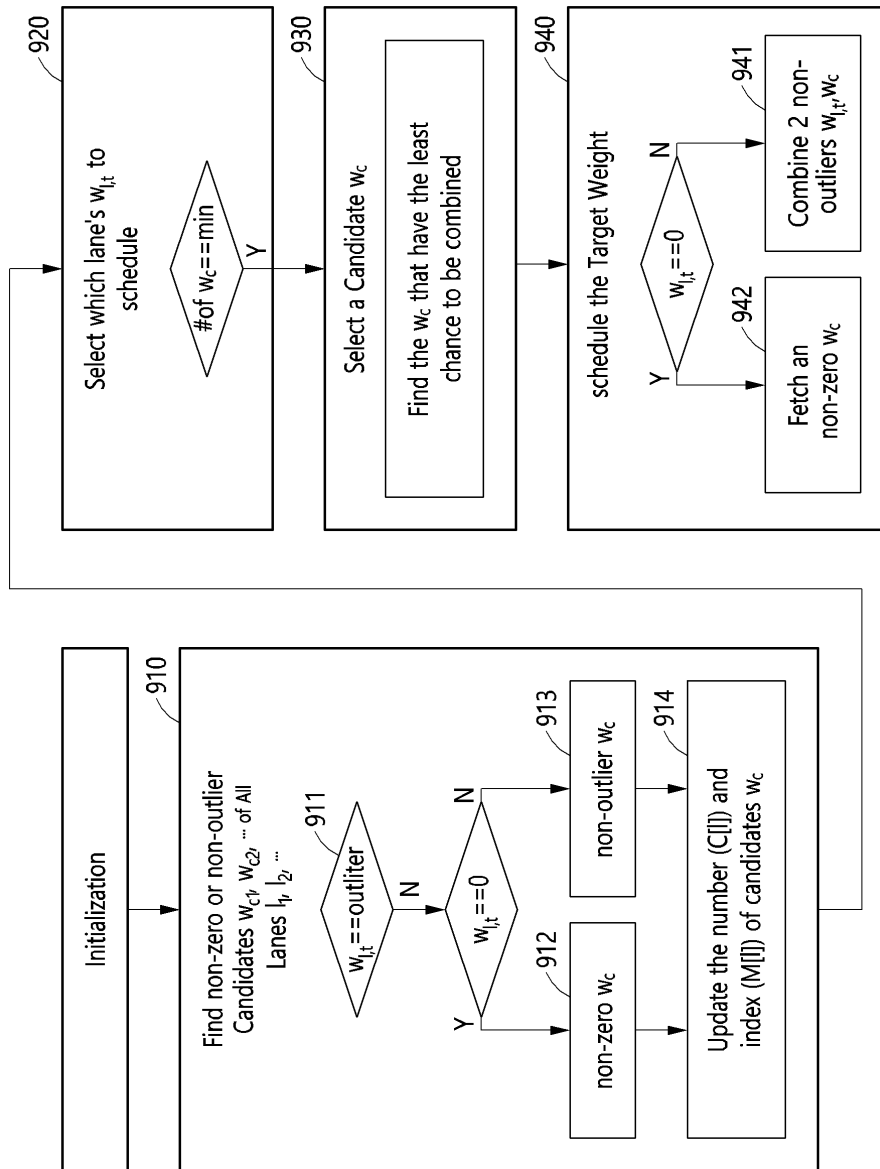
도면7



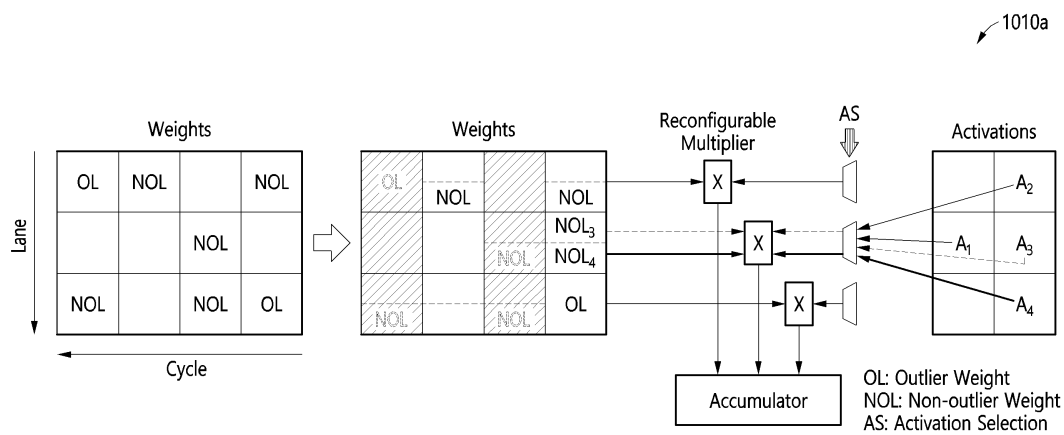
도면8



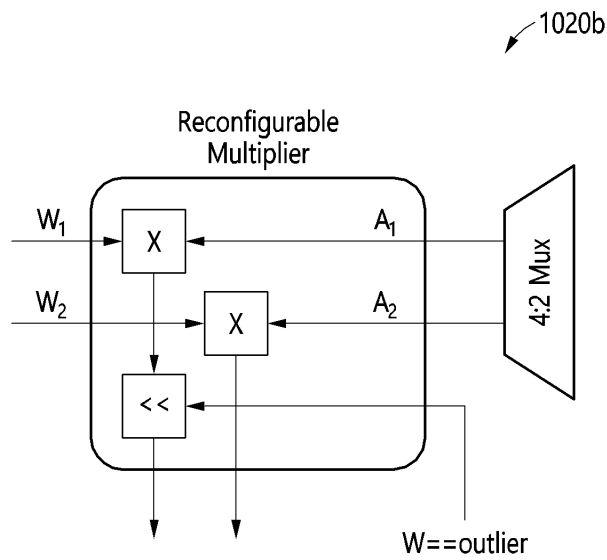
도면9



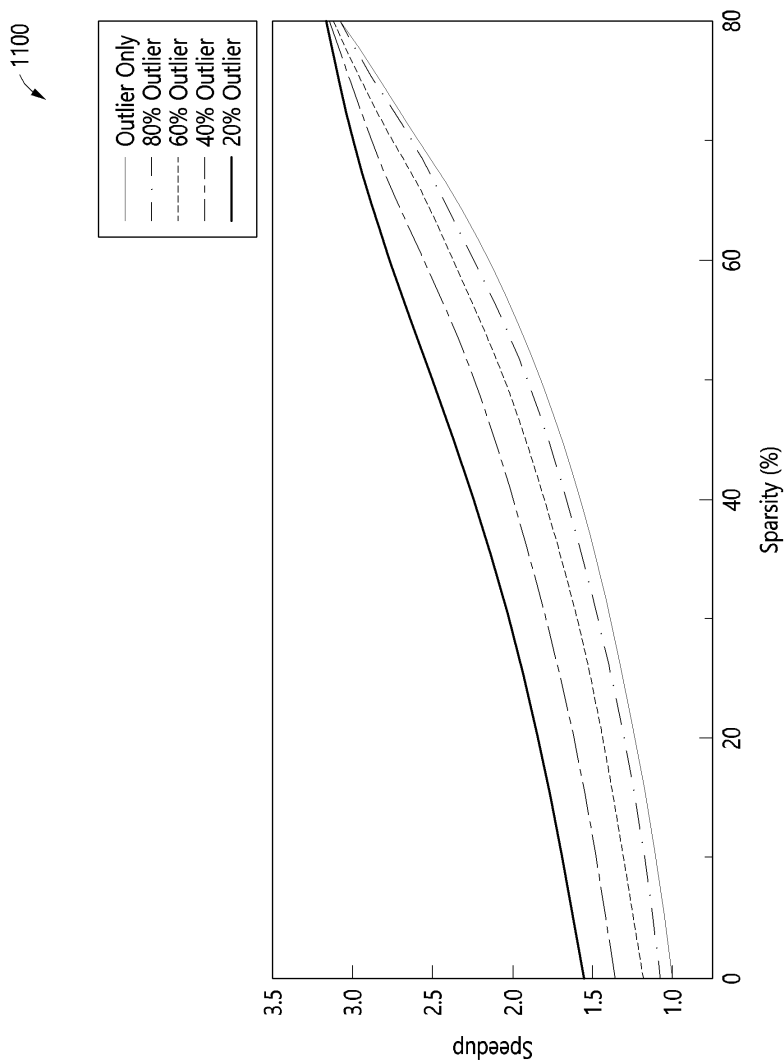
도면10a



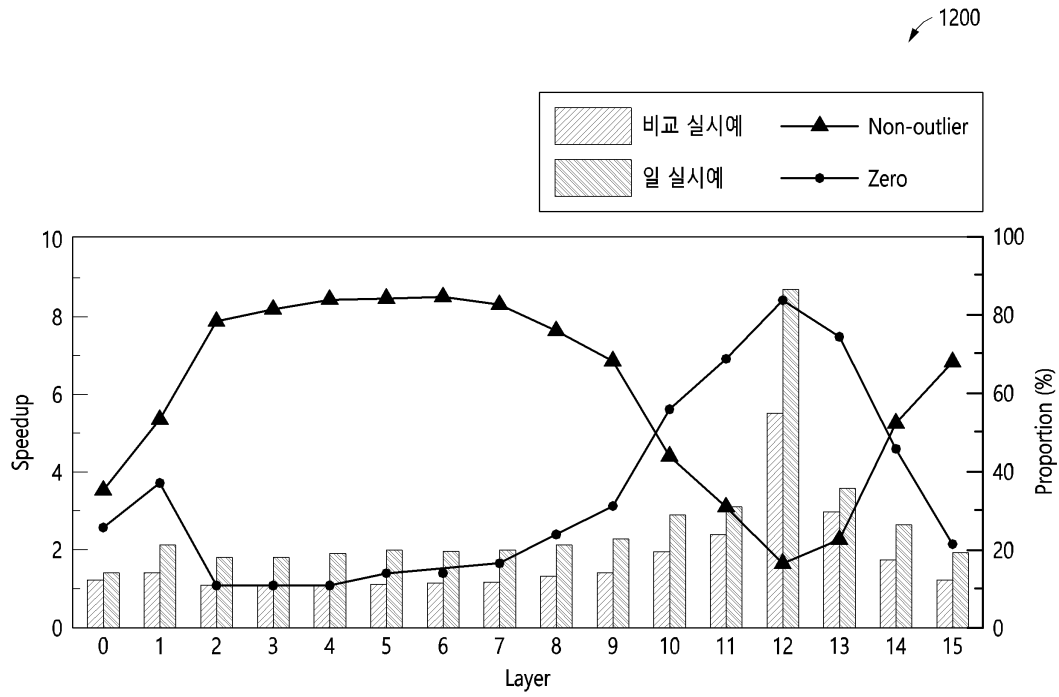
도면10b



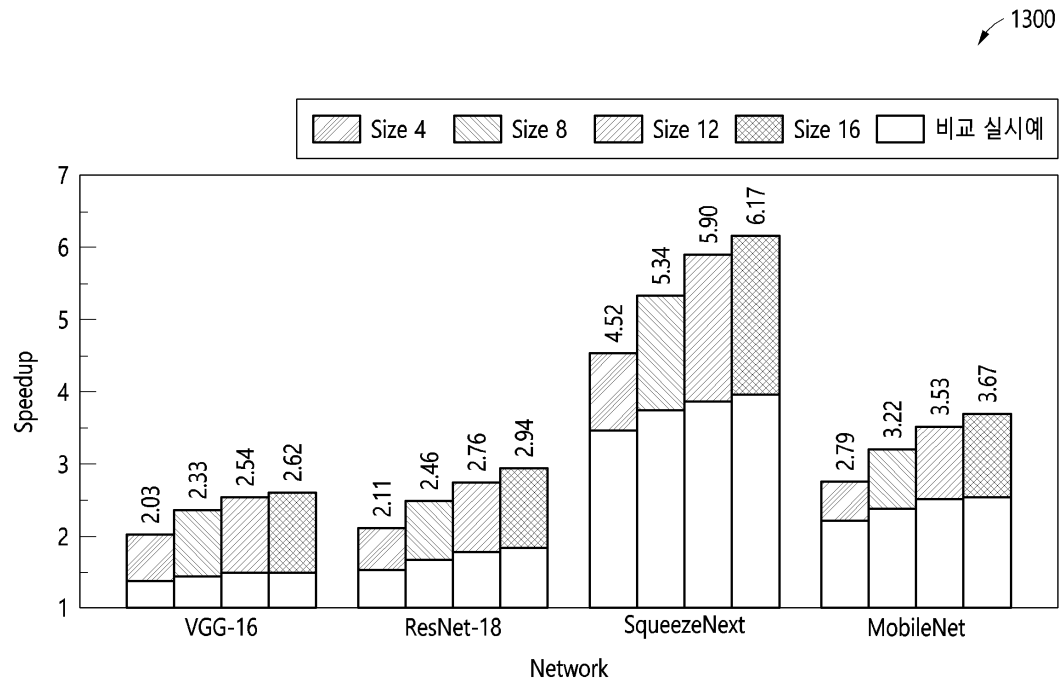
도면11



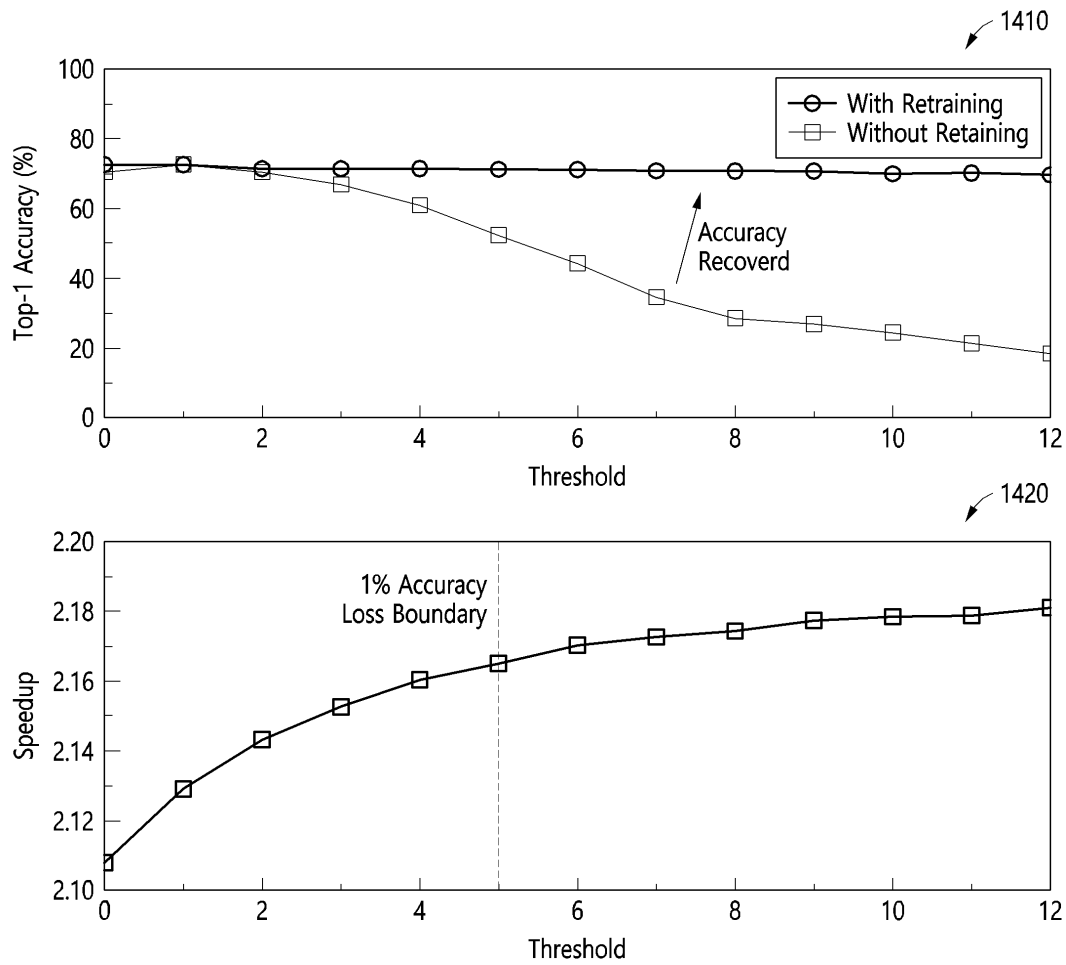
도면12



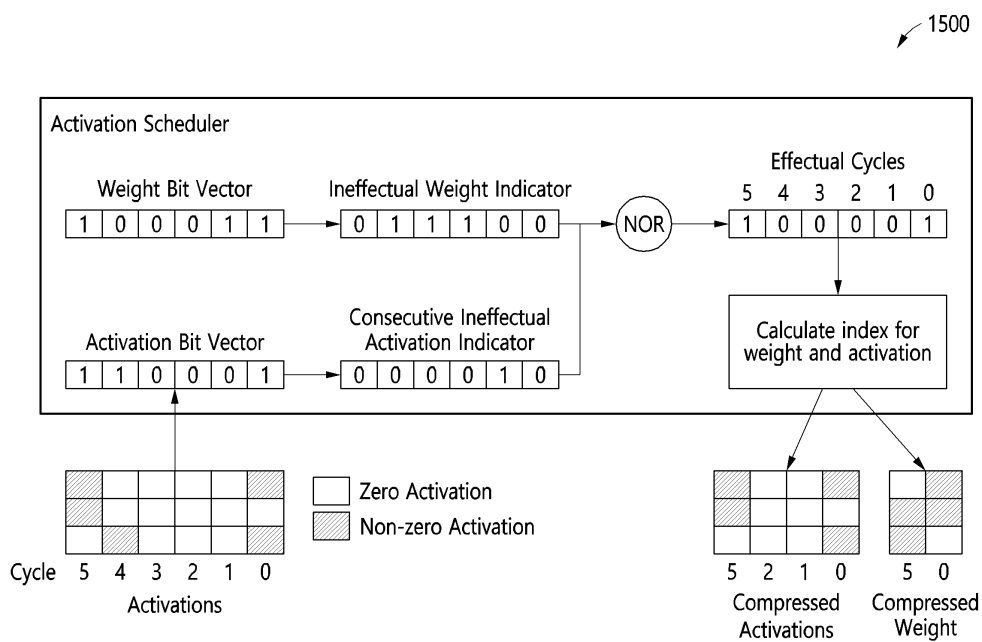
도면13



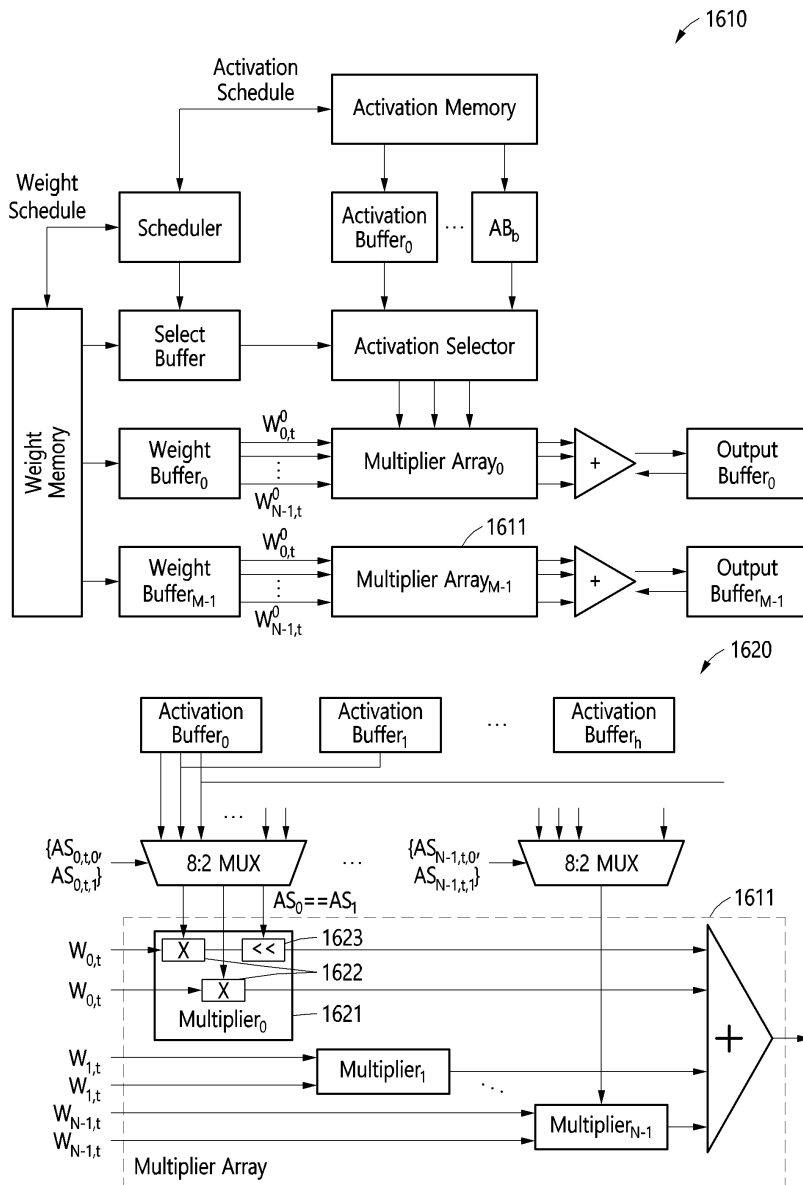
도면14



도면15



도면16



도면17

