



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년06월17일
(11) 등록번호 10-1989793
(24) 등록일자 2019년06월11일

- (51) 국제특허분류(Int. Cl.)
G06N 3/063 (2006.01) G06N 3/08 (2006.01)
- (52) CPC특허분류
G06N 3/063 (2013.01)
G06N 3/08 (2013.01)
- (21) 출원번호 10-2018-0126055
- (22) 출원일자 2018년10월22일
심사청구일자 2018년10월22일
- (56) 선행기술조사문헌
Han, Song, et al. EIE: efficient inference engine on compressed deep neural network. Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on. IEEE, 2016.*
Wen, Wei, et al. "Learning structured sparsity in deep neural networks." Advances in Neural Information Processing Systems. 2016.*
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
한국기술교육대학교 산학협력단
충청남도 천안시 동남구 병천면 충절로 1600 (한국기술교육대학교내)
- (72) 발명자
강형주
대전광역시 서구 둔산동 한마루아파트 4동 1204호
- (74) 대리인
특허법인(유한) 대아

전체 청구항 수 : 총 5 항

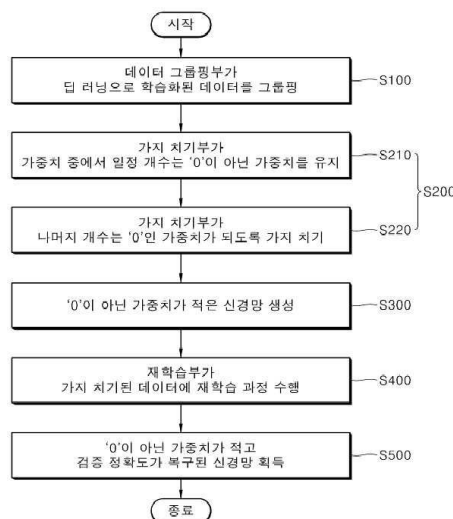
심사관 : 서광훈

(54) 발명의 명칭 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법 및 기록 매체

(57) 요약

본 발명은 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법 및 기록 매체를 제공한다. 상기 방법은 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 그룹핑 (S100) 가지 치기부가 가중치 중에서 일정 개수는 '0'이 아닌 가중치를 유지 (S210) 가지 치기부가 나머지 개수는 '0'인 가중치가 되도록 가지 치기 (S220) '0'이 아닌 가중치가 적은 신경망 생성 (S300) 재학습부가 가지 치기된 데이터에 재학습 과정 수행 (S400) '0'이 아닌 가중치가 적고 검증 정확도가 복구된 신경망 획득 (S500) 종료

대표도 - 도12



이 발명을 지원한 국가연구개발사업

과제고유번호 2015R1D1A1A01058768

부처명 교육부

연구관리전문기관 한국연구재단

연구사업명 이공분야기초연구사업

연구과제명 범용 행렬 처리기의 개발과 이를 활용한 임베디드 시스템에서의 딥러닝 가속화

기 여 율 1/1

주관기관 한국기술교육대학교 산학협력단

연구기간 2015.11.01 ~ 2018.10.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

(a) 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 입력받아 동시에 폐치하는 형태가 다른 가속기 회로에 맞춤형으로 소정의 축을 따라 일정 크기로 그룹핑하는 단계; 및

(b) 가지 치기부가 상기 그룹핑된 데이터를 입력받아, 가중치 중에서 일정 개수는 '0'이 아닌 가중치를 유지하고, 나머지 개수는 '0'인 가중치가 되도록 가지 치기하는 단계;

를 포함하고,

상기 (b) 단계는

상기 가지 치기부가 모든 가중치 폐칭 그룹을 m 개의 가지 치기 그룹으로 나누는 단계;

상기 가지 치기 그룹의 크기(g)가 활성화 개수(N_{par})/ 상기 가지 치기 그룹 개수(m)로 되는 단계; 및

상기 가지 치기 그룹이 상기 일정 개수의 '0'이 아닌 가중치를 가지도록 상기 가지 치기를 수행하는 단계; 를 포함하며,

상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나이며,

상기 가속기 회로는 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기인 것을 특징으로 하는,

컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법.

청구항 2

(a) 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 입력받아, 동시에 폐치하는 형태가 다른 가속기 회로에 맞추어 소정의 축을 따라 일정 크기로 그룹핑하는 단계;

(b) 가지 치기부가 상기 그룹핑된 데이터를 입력받아, 가중치 중에서 일정 개수는 '0'이 아닌 가중치를 유지하고, 나머지 개수는 '0'인 가중치가 되도록 가지 치기하는 단계;

(c) 상기 (a) 단계보다 상기 '0'이 아닌 가중치가 적은 신경망이 생성되는 단계; 및

(d) 재학습부에 의해 상기 '0'이 된 가중치는 '0'을 유지하면서 상기 가지 치기된 데이터에 재학습 과정이 수행되어, 상기 (a) 단계보다 상기 '0'이 아닌 가중치가 적고 검증 정확도가 복구된 신경망이 획득되는 단계;

를 포함하고,

상기 (b) 단계는

상기 가지 치기부가 모든 가중치 폐칭 그룹을 m 개의 가지 치기 그룹으로 나누는 단계;

상기 가지 치기 그룹의 크기(g)가 활성화 개수(N_{par})/ 상기 가지 치기 그룹 개수(m)로 되는 단계; 및

상기 가지 치기 그룹이 상기 일정 개수의 '0'이 아닌 가중치를 가지도록 상기 가지 치기를 수행하는 단계; 를 포함하며,

상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나이며,

상기 가속기 회로는 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기인 것을 특징으로 하는,

컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법.

청구항 3

(a) 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 입력받아 동시에 폐치하는 형태가 다른 가속기 회로에 맞춤형으로 소정의 축을 따라 일정 크기로 그룹핑하는 단계; 및

(b) 가지 치기부가 상기 그룹핑된 데이터를 입력받아, 가중치 중에서 일정 개수는 '0'이 아닌 가중치를 유지하고, 나머지 개수는 '0'인 가중치가 되도록 가지 치기하는 단계;

를 포함하고,

상기 (b) 단계는

상기 가지 치기부가 모든 가중치 폐칭 그룹을 m 개의 가지 치기 그룹으로 나누는 단계;

상기 가지 치기 그룹의 크기(g)가 활성화 개수(N_{par})/ 상기 가지 치기 그룹 개수(m)로 되는 단계; 및

상기 가지 치기 그룹이 상기 일정 개수의 '0'이 아닌 가중치를 가지도록 상기 가지 치기를 수행하는 단계; 를 포함하며,

상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나이며,

상기 가속기 회로는 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기이고,

상기 '0'이 아닌 가중치만 가중치 메모리에 저장하는 것을 특징으로 하는,

컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법.

청구항 4

(a) 멀티플렉서, 곱셈기, 가산기 및 누적기를 포함하는 가속기 인식용 컨볼루션 신경망을 위한 가지 치기 방법에 있어서,

데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 입력받아 동시에 폐치하는 형태가 다른 가속기 회로에 맞춤형으로 소정의 축을 따라 일정 크기로 그룹핑하는 단계; 및

(b) 가지 치기부가 상기 그룹핑된 데이터를 입력받아, 가중치 중에서 일정 개수는 '0'이 아닌 가중치를 유지하고, 나머지 개수는 '0'인 가중치가 되도록 가지 치기하는 단계;

를 포함하고,

상기 (b) 단계는

상기 가지 치기부가 모든 가중치 폐칭 그룹을 m 개의 가지 치기 그룹으로 나누는 단계;

상기 가지 치기 그룹의 크기(g)가 활성화 개수(N_{par})/ 상기 가지 치기 그룹 개수(m)로 되는 단계; 및

상기 가지 치기 그룹이 상기 일정 개수의 '0'이 아닌 가중치를 가지도록 상기 가지 치기를 수행하는 단계; 를 포함하며,

상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나이며,

상기 가속기 회로는 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기이고,

상기 곱셈기는 상기 '0'이 아닌 가중치와 상기 '0'이 아닌 가중치와 관련된 활성화 데이터를 입력받는 것을 특징으로 하는,

컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법.

청구항 5

(a) 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 입력받아, 동시에 폐치하는 형태가 다른 가속기 회로에 맞추어 소정의 축을 따라 일정 크기로 그룹핑하는 단계;

(b) 가지 치기부가 상기 그룹핑된 데이터를 입력받아, 가중치 중에서 일정 개수는 '0'이 아닌 가중치를 유지하고, 나머지 개수는 '0'인 가중치가 되도록 가지 치기하는 단계;

(c) 상기 (a) 단계보다 상기 '0'이 아닌 가중치가 적은 신경망이 생성되는 단계; 및

(d) 재학습부에 의해 상기 '0'이 된 가중치는 '0'을 유지하면서 상기 가지 치기된 데이터에 재학습 과정이 수행되어, 상기 (a) 단계보다 상기 '0'이 아닌 가중치가 적고 검증 정확도가 복구된 신경망이 획득되는 단계;

를 포함하고,

상기 (b) 단계는

상기 가지 치기부가 모든 가중치 폐칭 그룹을 m 개의 가지 치기 그룹으로 나누는 단계;

상기 가지 치기 그룹의 크기(g)가 활성화 개수(Npar)/ 상기 가지 치기 그룹 개수(m)로 되는 단계; 및

상기 가지 치기 그룹이 상기 일정 개수의 '0'이 아닌 가중치를 가지도록 상기 가지 치기를 수행하는 단계; 를 포함하며,

상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나이며,

상기 가속기 회로는 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기인 것을 특징으로 하는,

컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 실행하기 위한 프로그램이 기록되어 컴퓨터로 판독가능한 기록매체.

발명의 설명

기술 분야

[0001] 본 발명은 가속기 인식 가지 치기 방법 및 기록 매체에 관한 것으로, 특히 딥 러닝으로 학습화된 데이터를 일정 크기로 그룹핑한 후에, 병렬 처리 데이터 그룹에 해당하는 각 가중치 그룹에 대해 동일한 수의 가중치가 유지되도록 가지 치기를 수행할 수 있는 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법 및 기록 매체에 관한 것이다.

배경 기술

[0002] 최근에 컨볼루션 신경망(convolutional neural network)는 이미지 인식, 물체 탐지, 이미지 분할 분야에서 관심을 끌고 있다.

[0003] 컨볼루션 신경망 기법은 컴퓨터 비전 작업에 뛰어난 성능을 제공하지만, 임베디드(embedded) 환경에서 채택하기 전에 몇 가지 장애물이 존재한다.

[0004] 즉, 컨볼루션 신경망 기법에는 대개 많은 컨볼루션 계층(convolutional layer)과 완전히 연결된 몇 개의 계층으로 구성되는데, 컨볼루션 계층은 일반적으로 산술 연산의 90 % 이상을 차지하는 등 과도한 데이터 량의 저장 및 산술 연산이 필요하다.

[0005] 각 계층 간에는 정류된 선형 유닛(rectified linear units, ReLU), 풀링 계층 및 일괄 정규화 계층과 같은 활성화 계층이 있다.

[0006] 완전 연결 계층은 많은 저장 공간을 필요로 하기 때문에 최근의 컨볼루션 신경망은 단 하나의 완전 연결 계층을 갖는 경향이 있다.

[0007] 컨볼루션 계층에서의 연산은 다음의 수학식 1과 같다.

[0008] [수학식 1]

$$fo(m, h, w) = \sum_{c=0}^{C-1} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} w(m, c, i, j) \times fi(c, S \times h + i, S \times w + j)$$

[0009]

[0010] 여기에서, fo(m, h, w)는 출력 데이터 m에서 행 h와 열 w의 활성화 데이터이고, w(m, c, i, j)는 컨볼루션 가중치이고, fi(c, y, x)는 입력 데이터 c의 행 y와 열 x를 활성화한 것이다.

[0011] 또한, C는 입력 데이터 또는 채널의 수이고, K는 커널의 공간 크기이며, S는 스트라이드 크기이다.

[0012] 도 1은 컨볼루션 층에서 각 축을 따라 일정한 공간 크기, 깊이, 필터 수로 구성된 가중치 구조를 나타낸 도면이다.

[0013] 즉, 도 1에서 공간 크기는 K x K, 깊이는 C, 필터 수는 M 개로서, 이해의 편의를 위해 일부 축이 정의되어 있다.

[0014] 입력 활성화 데이터의 채널 축은 y와 x가 같은 fi(c, y, x)이다.

[0015] 마찬가지로, 가중치의 채널 축은 동일한 m, i 및 j를 갖는 w(m, c, i, j)이다.

[0016] 공간 축은 동일한 c를 가지는 fi(c, y, x) 또는 동일한 m 및 c를 가지는 w(m, c, i, j)이다.

[0017] 필터 축은 동일한 c, i 및 j를 가지는 w(m, c, i, j)이다.

[0018] 한편, 신경망 네트워크에서 가지 치기(pruning)란 중요하지 않은 가중치 또는 노드를 제거하여 저장량 및 작업량을 줄이는 작업을 말한다.

[0019] 기존의 가지 치기 기법은 많은 가중치를 제거할 수 있지만, 제거한 패턴에 불규칙성을 표시하고, 해당 가속기 아키텍처는 완전히 연결된 계층만 처리할 수 있다.

[0020] 특히, 컨볼루션 계층은 더 많은 에너지를 소비하기 때문에 종래에 제안된 에너지 인식 가지 치기 기법은 컨볼루션 계층에 중점을 두지만, 대부분 가지 치기 패턴의 규칙성은 고려되지 않았다.

[0021] 최근에, 가지 치기의 규칙성을 고려한 연구들이 발표되었는데, 이에 대한 구조는 다음과 같다.

[0022] 도 2는 가지 치기의 규칙성을 고려한 컨볼루션 신경망 네트워크의 종래 가지 치기 기법의 구조에 대한 표이다.

[0023] 도 2에서 보는 바와 같이, 이러한 종래의 가지 치기 기법은 채널 단위(channel-wise), 필터 단위(filter-wise) 및 모양 단위(shape-wise)의 가지 치기로 분류할 수 있다.

[0024] 예를 들어, 채널 단위 가지 치기에서는 채널 w(:, n, :, :)의 모든 가중치가 제거되거나 전혀 사용되지 않는다.

[0025] 이러한 가지 치기 방식을 구조화된 가지 치기(structured pruning)라 부르는데, 일부 구조화된 가지 치기 기법은 범용 그래픽 처리 장치(General-Purpose Graphic Processing Unit, GPGPU)를 대상으로 한다.

[0026] 범용 그래픽 처리 장치 구현에서, 컨볼루션은 대개 행렬 곱셈으로 변환되고, 변환된 가중치 행렬의 열을 제거하면 작업량을 줄일 수 있다.

[0027] 가중치 행렬의 열은 가중치 w(:, n, l, j)에 해당하므로 이 체계는 모양 단위의 가지 치기로 간주할 수 있다.

[0028] 이와 같이, 컨볼루션 신경망 네트워크의 규칙성을 고려한 종래의 가지 치기 기법은 범용 그래픽 처리 장치 또는 중앙 처리 장치(Central processing unit, CPU)를 대상으로 하고, 주문형 반도체(application specific integrated circuit, ASIC) 가속기 또는 필드 프로그래머블 게이트 어레이(field programmable gate array, FPGA) 가속기에의 적용 가능성을 고려하지 않고 개발되었고, 주로 가지 치기될 수 있는 가중치의 양에 포커싱되어 있다.

[0029] 가지 치기된 네트워크는 가중치 희박성(weight sparsity)을 이용할 수 있는 일부 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기에서 실행될 수 있다.

[0030] 그런데, 다양한 가지 치기 방식과 가속기에도 불구하고, 가지 치기된 네트워크를 가지고 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기의 성능을 효율적으로 활용하기는 쉽지 않다.

[0031] 즉, 제한이 없는 가지 치기는 '0'이 아니도록 남은 가중치에 불규칙한 패턴을 만들고, 불규칙한 패턴은 가지

치기된 네트워크가 스파스 가속기(sparse accelerator)에서 수행될 때 비효율적이다.

- [0032] 활성화 데이터 페칭(fetching)과 가중치 페칭의 오정렬은 '0' 삽입의 패딩(padding)이 요구된다.
- [0033] 여기에서, 패딩(padding)은 레코드나 블록의 맨 나중에 공백이나 의미가 없는 기호를 부가하여 고정 길이로 하는 것을 말한다.
- [0034] 처리 요소(processing elements, PE)는 서로 다른 가중치 수를 처리하기 때문에, 일부 처리 요소는 다른 처리 요소가 완료될 때까지 기다려야 한다.
- [0035] 이 문제를 완화하기 위해 일부 가속기는 복잡한 구조를 사용한다.
- [0036] 예를 들어, 종래의 스파스 가속기 중 하나인 캄브리콘-X(Cambricon-X) 기법은 매우 넓은(256 x 16 비트 너비) 메모리와 매우 넓은(256 대 1) 멀티플렉서(40)를 사용한다.
- [0037] 가속기는 보통 복수개의 처리 요소(Processing Element, PE)로 구성되며, 각 처리 요소에는 단일 또는 복수개의 곱셈기(multiplier)가 있다.
- [0038] 복수개의 곱셈기는 다중 가중치 및 다중 활성화(multiple weights and multiple activations, MWMA), 다중 가중치 및 단일 활성화(multiple weights and single activations, MWSA) 또는 단일 가중치 및 다중 활성화(single weights and multiple activations, SWMA)에 대하여 데이터를 곱할 수 있다.
- [0039] 도 3은 종래의 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기용 다중 가중치 및 다중 활성화(MWMA) 구조의 전형적인 회로도이다.
- [0040] 도 4는 종래의 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기용 다중 가중치 및 단일 활성화(MWSA) 구조의 전형적인 회로도이다.
- [0041] 도 3 및 도 4에서 보는 바와 같이, 처리 요소가 복수개의 곱셈기를 갖는 경우, 곱셈기의 피 연산자를 동시에 페칭하는 것이 중요하다.
- [0042] 페칭 그룹은 하나의 처리 요소에서 동시에 페치(fetch)되고 처리되는 활성화 데이터 또는 가중치를 의미한다.
- [0043] 활성화 데이터 페칭 그룹의 크기가 N_{par} 이고, 하나의 처리 요소에서 곱셈기의 수가 N_{mul} 이면, 페칭 그룹의 크기(N_{par})는 보통 곱셈기의 수(N_{mul})와 동일하다.
- [0044] 처리 요소의 구조는 가중치 및 활성화 페칭 그룹이 따르는 축으로 더 자세히 분류할 수 있다.
- [0045] 도 5는 채널 축을 따라 페치되는 종래의 다중 가중치 및 다중 활성화(MWMA) 구조의 컨볼루션 계층 신경망 네트워크 가속기의 프로세스를 나타낸 구성도이다.
- [0046] 도 5에서 보는 바와 같이, 입력 데이터의 일부가 채널 축을 따라 커널 가중치의 일부와 함께 페치되어 곱셈기(10)에서 곱해진다.
- [0047] 곱셈 결과는 가산기(20)에서 합산되고 누적기(30)에서 누적되어 하나의 출력 데이터가 된다.
- [0048] 그 밖에, 종래의 컨볼루션 신경망용 가속기의 프로세스에는 다중 가중치가 필터 축을 따라 페치되는 경우도 있고, 다중 가중치 및 다중 활성화 데이터가 공간 축을 따라 페치되는 경우도 있다.
- [0049] 그런데, 종래의 구조화되지 않은 가지 치기 기법은 높은 가지 치기 비율에 도달할 수 있지만, 스파스 가속기 구조에 잘 맞는 가지 치기 네트워크를 형성하기는 좀처럼 쉽지 않다.
- [0050] 그 주된 이유는 가지 치기 후에 남아있는 '0'이 아닌 가중치의 불균일 분포, 특히 각 가중치 페치 그룹에 대한 '0'이 아닌 가중치의 개수 때문이다.
- [0051] 이러한 가중치의 불균일 분배는 곱셈에 필요한 활성화 데이터 및 가중치 간의 오정렬을 초래할 수 있는 문제가 있다.
- [0052] 즉, 다중 가중치 및 다중 활성화(MWMA) 처리 요소가 있는 가속기에서 곱셈기(10)의 피연산자는 동시에 페치해야 한다.
- [0053] 복수개(N_{par})의 활성화 데이터 및 복수개(N_{mul})의 non-zero 가중치를 내부 버퍼에서 가져올 때, 가져온 모든 가중치가 페치된 활성화 데이터에서 해당 활성화 데이터를 찾을 수 있는 것은 아니므로, 가중치 프로세스를 위해

다른 활성화 데이터 패치 그룹에서 가져와야 할 수도 있게 된다.

- [0054] 이러한 오정렬 문제를 해결하기 위해 종래의 캠브리콘-X 기법에는 패딩- '0' 스킵이 도입되는데, 패딩- '0'을 삽입하여 각 활성화 데이터 패치 그룹에 대한 '0'이 아닌 가중치와 패딩- '0'의 수가 항상 곱셈기 개수(Nmul)의 배수가 되도록 해야 한다.
- [0055] 그러나, 패딩- '0'은 내부 가중치 버퍼를 낭비할 뿐 아니라, 또 다른 비효율을 초래하는 한계가 있다.
- [0056] 즉, 패딩- '0'의 총 개수는 활성화 개수(Npar)가 클수록 작아지는데, 이는 캠브리콘-X 기법이 곱셈기 개수(Nmul)와 비교하여 매우 큰 활성화 데이터 패치 그룹 크기 (Npar)를 사용하는 이유 중 하나이다.
- [0057] 컨볼루션 계층에서 75 %의 일반적인 가지 치기 비율의 경우, 곱셈기 개수(Nmul) = 16 일 때 활성화 데이터 패치 그룹 크기 (Npar) = 64로 충분하지만, 캠브리콘-X는 활성화 데이터 패치 그룹 크기 (Npar) = 256을 사용한다.
- [0058] 활성화 데이터 패치 그룹 크기 (Npar)가 크기 때문에 활성화 선택 부분인 인덱싱 모듈(Indexing Module, IM) 블록에는 매우 넓은 256 대 1 멀티플렉서(40)가 필요하다.
- [0059] 또한, 멀티플렉서(40)의 수가 처리 요소 당 $16 \times w$ (w 는 활성화 데이터 한 개의 비트 너비) 비트 너비로 많기 때문에, 이러한 넓은 멀티플렉서(40)는 전체 칩 영역의 30 % 이상을 차지하는 큰 인덱싱 모듈 블록 영역을 유발한다.
- [0060] 대형 활성화 개수(Npar)에는 매우 넓은($256 \times w$ 비트 너비) 내부 활성화 버퍼가 필요하여, 이러한 넓은 메모리는 일반적으로 정사각형 형상의 메모리보다 큰 영역을 초래한다.
- [0061] 또한, 처리 요소 간의 로드 밸런싱도 문제가 된다.
- [0062] 즉, 처리 요소는 '0'이 아닌 가중치의 개수(Nnon-zero)/곱셈기 개수(Nmul) 주기 동안 활성화 패치 그룹을 처리하므로, '0'이 아닌 가중치의 개수(Nnon-zero)의 다양성 때문에 주기의 헛수도 변화될 것이다.
- [0063] 이것은 캠브리콘-X와 같은 종래의 아키텍처에서 처리 요소 사이로부터 폐치된 활성화를 공유하여 내부 버퍼와 외부 메모리 사이에서 요구되는 대역폭을 줄이는 경우에 문제를 일으킬 수 있다.
- [0064] 일부 처리 요소가 폐치된 활성화 데이터의 처리를 일찍 완료하면, 처리 요소는 다른 처리 요소들이 완료 될 때까지 기다려야 하는 부하 불균형 문제가 존재한다.
- [0065] 이러한 부하 불균형 문제는 특히 제로 가중치를 건너 뛰는 단일 가중치 및 다중 활성화(SWMA) 및 단일 가중치 및 단일 활성화(SWSA) 구조와 같은 가중치 직렬 구조를 갖는 가속기에서 주요 이슈이기도 하다.
- [0066] 그런데, 종래의 가지 치기에 대한 연구에는 남아있는 '0'이 아닌 가중치의 분포(distribution)를 고려한 연구가 거의 없었다.
- [0067] 이에 본 발명자들은 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기에서 각 가중치 그룹에 대해 동일한 수의 가중치가 유지되도록 가지 치기를 수행함으로써, 연산량을 감소시키고 곱셈기의 낭비를 제거하여 회로를 단순화시키며, 내부 버퍼의 오정렬 및 로드 불균형 문제를 해결할 수 있는 멀티대역 네트워크 선택이 가능한 네트워크 선택 시스템을 고안하기에 이르렀다.

선행기술문헌

특허문헌

- [0068] (특허문헌 0001) JP 2018-129033 A

발명의 내용

해결하려는 과제

- [0069] 본 발명의 목적은 병렬 처리 데이터 그룹에 해당하는 각 가중치 그룹에 대해 동일한 수의 가중치가 유지되도록 가지 치기가 수행되어, 종래의 스파스 가속기의 효율성을 향상시키고 논리 복잡성을 줄일 수 있는 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 제공하는 것이다.

[0070] 또한, 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기에서 의미 없는 덧셈이나 곱셈을 제거하여 연산량을 감소시키고 곱셈기의 낭비를 제거하여 회로를 단순화시킬 수 있는 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 제공하는 것이다.

[0071] 본 발명의 다른 목적은 상기 목적을 달성하기 위하여 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체를 제공하는 것이다.

과제의 해결 수단

[0072] 상기 목적을 달성하기 위한 본 발명의 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법은 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 인가받아 가속기 회로에 맞춤형으로 소정의 축을 따라 일정 크기로 그룹핑하는 단계; 및 가지 치기부가 상기 그룹핑된 데이터를 인가받아, 가중치 중에서 일정 개수는 '0' 이 아닌 가중치를 유지하고, 나머지 개수는 '0' 인 가중치가 되도록 가지 치기하는 단계; 를 포함하고, 상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나인 것을 특징으로 한다.

[0073] 상기 목적을 달성하기 위한 본 발명의 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법은 (a) 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 인가받아, 동시에 폐지하는 형태가 다른 가속기 회로에 맞추어 소정의 축을 따라 일정 크기로 그룹핑하는 단계; (b) 가지 치기부가 상기 그룹핑된 데이터를 인가받아, 가중치 중에서 일정 개수는 '0' 이 아닌 가중치를 유지하고, 나머지 개수는 '0' 인 가중치가 되도록 가지 치기하는 단계; (c) 상기 (a) 단계보다 상기 '0' 이 아닌 가중치가 적은 신경망이 생성되는 단계; 및 (d) 재학습부에 의해 상기 '0'이 된 가중치는 '0'을 유지하면서 상기 가지 치기된 데이터에 재학습 과정이 수행되어, 상기 (a) 단계보다 상기 '0' 이 아닌 가중치가 적고 검증 정확도가 복구된 신경망이 획득되는 단계; 를 포함하고, 상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나인 것을 특징으로 한다.

[0074] 상기 목적을 달성하기 위한 본 발명의 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법은 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 인가받아 가속기 회로에 맞춤형으로 소정의 축을 따라 일정 크기로 그룹핑하는 단계; 및 가지 치기부가 상기 그룹핑된 데이터를 인가받아, 가중치 중에서 일정 개수는 '0' 이 아닌 가중치를 유지하고, 나머지 개수는 '0' 인 가중치가 되도록 가지 치기하는 단계; 를 포함하고, 상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나이며, 상기 '0' 이 아닌 가중치만 가중치 메모리에 저장하는 것을 특징으로 한다.

[0075] 상기 목적을 달성하기 위한 본 발명의 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법은 멀티플렉서, 곱셈기, 가산기 및 누적기를 포함하는 가속기 인식용 컨볼루션 신경망을 위한 가지 치기 방법에 있어서, 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 인가받아 가속기 회로에 맞춤형으로 소정의 축을 따라 일정 크기로 그룹핑하는 단계; 및 가지 치기부가 상기 그룹핑된 데이터를 인가받아, 가중치 중에서 일정 개수는 '0' 이 아닌 가중치를 유지하고, 나머지 개수는 '0' 인 가중치가 되도록 가지 치기하는 단계; 를 포함하고, 상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나이며, 상기 곱셈기는 상기 '0'이 아닌 가중치와 해당 활성화 데이터를 인가받는 것을 특징으로 한다.

[0076] 상기 다른 목적을 달성하기 위한 본 발명의 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법의 기록 매체는 (a) 데이터 그룹핑부가 딥 러닝으로 학습화된 데이터를 인가받아, 동시에 폐지하는 형태가 다른 가속기 회로에 맞추어 소정의 축을 따라 일정 크기로 그룹핑하는 단계; (b) 가지 치기부가 상기 그룹핑된 데이터를 인가받아, 가중치 중에서 일정 개수는 '0' 이 아닌 가중치를 유지하고, 나머지 개수는 '0' 인 가중치가 되도록 가지 치기하는 단계; (c) 상기 (a) 단계보다 상기 '0' 이 아닌 가중치가 적은 신경망이 생성되는 단계; 및 (d) 재학습부에 의해 상기 '0'이 된 가중치는 '0'을 유지하면서 상기 가지 치기된 데이터에 재학습 과정이 수행되어, 상기 (a) 단계보다 상기 '0' 이 아닌 가중치가 적고 검증 정확도가 복구된 신경망이 획득되는 단계; 를 포함하고, 상기 소정의 축은 채널 축, 필터 축, 공간 축 중 어느 하나인 것을 특징으로 한다.

[0077] 기타 실시예의 구체적인 사항은 "발명을 실시하기 위한 구체적인 내용" 및 첨부 "도면"에 포함되어 있다.

[0078] 본 발명의 이점 및/또는 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 각종 실시예를 참조하면 명확해질 것이다.

[0079] 그러나, 본 발명은 이하에서 개시되는 각 실시예의 구성만으로 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수도 있으며, 단지 본 명세서에서 개시한 각각의 실시예는 본 발명의 개시가 완전하도록 하며, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 본 발명의 범주를 완전하게 알려주기 위해 제공되는 것

이며, 본 발명은 청구범위의 각 청구항의 범주에 의해 정의될 뿐임을 알아야 한다.

발명의 효과

- [0080] 본 발명에 의한 경우, 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기에서 내부 버퍼의 고정렬이 해결되어 곱셈기의 낭비가 방지되고 회로가 단순화된다.
- [0081] 또한, 가속기의 처리 요소의 활성화 과정에서 모든 처리 요소가 완료될 때까지 기다릴 필요가 없어, 로드 불균형 문제가 해결되고 성능이 현저하게 향상된다.
- [0082] 또한, 가속기의 컨볼루션 계층의 가지 치기 및 재학습을 통해, 열화된 검증 정확도를 기본 정확도로 복구할 수 있게 된다.

도면의 간단한 설명

- [0083] 도 1은 컨볼루션 층에서 각 축을 따라 일정한 공간 크기, 깊이, 필터 수로 구성된 가중치 구조를 나타낸 도면이다.
- 도 2는 가지 치기의 규칙성을 고려한 컨볼루션 신경망 네트워크의 종래 가지 치기 기법의 구조에 대한 표이다.
- 도 3은 종래의 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기용 다중 가중치 및 다중 활성화(MWMA) 구조의 전형적인 회로도이다.
- 도 4는 종래의 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기용 다중 가중치 및 단일 활성화(MWSA) 구조의 전형적인 회로도이다.
- 도 5는 채널 축을 따라 폐지되는 종래의 다중 가중치 및 다중 활성화(MWMA) 구조의 컨볼루션 계층 신경망 네트워크 가속기의 프로세스를 나타낸 구성도이다.
- 도 6은 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 구현하기 위한 다중 가중치 및 다중 활성화(MWMA) 구조의 스파스 아키텍처 처리 요소의 구성도이다.
- 도 7은 도 6에 도시된 다중 가중치 및 다중 활성화(MWMA) 구조의 스파스 아키텍처 처리 요소에서 '0' 이 아닌 가중치의 수가 2인 가지 치기 기법의 구성도이다.
- 도 8은 도 6에 도시된 다중 가중치 및 다중 활성화(MWMA) 구조의 스파스 아키텍처 처리 요소에서 폐칭 그룹의 크기(Npar)가 8이고, 가지 치기 그룹의 크기 (g)가 4인 처리 요소의 구성도이다.
- 도 9는 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 구현한 직후에 측정된 이미지 넷 검증 정확도에 대한 표이다.
- 도 10은 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 및 재학습을 구현한 직후에 측정된 이미지 넷 검증 정확도에 대한 표이다.
- 도 11은 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 이용하고 재학습하여 성능이 복구된 신경망을 획득하는 방법에 대한 순서도이다.
- 도 12는 도 11에 도시된 방법에 이용된 가속기 인식 가지 치기 방법의 순서도이다.

발명을 실시하기 위한 구체적인 내용

- [0084] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예에 대하여 상세히 설명하면 다음과 같다.
- [0085] 본 발명을 상세히 설명하기 전에, 본 명세서에서 사용된 용어나 단어는 통상적이거나 사전적인 의미로 무조건 한정하여 해석되어서는 아니되며, 본 발명의 발명자가 자신의 발명을 가장 최선의 방법으로 설명하기 위해서 각종 용어의 개념을 적절하게 정의하여 사용할 수 있고, 더 나아가 이들 용어나 단어는 본 발명의 기술적 사상에 부합하는 의미와 개념으로 해석되어야 함을 알아야 한다.
- [0086] 즉, 본 명세서에서 사용된 용어는 본 발명의 바람직한 실시예를 설명하기 위해서 사용되는 것일 뿐이고, 본 발명의 내용을 구체적으로 한정하려는 의도로 사용된 것이 아니며, 이들 용어는 본 발명의 여러 가지 가능성을 고려하여 정의된 용어임을 알아야 한다.

- [0087] 또한, 본 명세서에 있어서, 단수의 표현은 문맥상 명확하게 다른 의미로 지시하지 않는 이상, 복수의 표현을 포함할 수 있으며, 유사하게 복수로 표현되어 있다고 하더라도 단수의 의미를 포함할 수 있음을 알아야 한다.
- [0088] 본 명세서의 전체에 걸쳐서 어떤 구성 요소가 다른 구성 요소를 "포함"한다고 기재하는 경우에는, 특별히 반대되는 의미의 기재가 없는 한 임의의 다른 구성 요소를 제외하는 것이 아니라 임의의 다른 구성 요소를 더 포함할 수도 있다는 것을 의미할 수 있다.
- [0089] 더 나아가서, 어떤 구성 요소가 다른 구성 요소의 "내부에 존재하거나, 연결되어 설치된다"고 기재한 경우에는, 이 구성 요소가 다른 구성 요소와 직접적으로 연결되어 있거나 접촉하여 설치되어 있을 수 있고, 일정한 거리를 두고 이격되어 설치되어 있을 수도 있으며, 일정한 거리를 두고 이격되어 설치되어 있는 경우에 대해서는 해당 구성 요소를 다른 구성 요소에 고정 내지 연결시키기 위한 제 3의 구성 요소 또는 수단이 존재할 수 있으며, 이 제 3의 구성 요소 또는 수단에 대한 설명은 생략될 수도 있음을 알아야 한다.
- [0090] 반면에, 어떤 구성 요소가 다른 구성 요소에 "직접 연결"되어 있다거나, 또는 "직접 접속"되어 있다고 기재되는 경우에는, 제 3의 구성 요소 또는 수단이 존재하지 않는 것으로 이해하여야 한다.
- [0091] 마찬가지로, 각 구성 요소 간의 관계를 설명하는 다른 표현들, 즉 " ~ 사이에"와 "바로 ~ 사이에", 또는 " ~ 에 이웃하는"과 " ~ 에 직접 이웃하는" 등도 마찬가지로 취지를 가지고 있는 것으로 해석되어야 한다.
- [0092] 또한, 본 명세서에 있어서 "일면", "타면", "일측", "타측", "제 1", "제 2" 등의 용어는, 사용된다면, 하나의 구성 요소에 대해서 이 하나의 구성 요소가 다른 구성 요소로부터 명확하게 구별될 수 있도록 하기 위해서 사용되며, 이와 같은 용어에 의해서 해당 구성 요소의 의미가 제한적으로 사용되는 것은 아님을 알아야 한다.
- [0093] 또한, 본 명세서에서 "상", "하", "좌", "우" 등의 위치와 관련된 용어는, 사용된다면, 해당 구성 요소에 대해서 해당 도면에서의 상대적인 위치를 나타내고 있는 것으로 이해하여야 하며, 이들의 위치에 대해서 절대적인 위치를 특정하지 않는 이상은, 이들 위치 관련 용어가 절대적인 위치를 언급하고 있는 것으로 이해하여서는 아니된다.
- [0094] 더욱이, 본 발명의 명세서에서는, "...부", "...기", "모듈", "장치" 등의 용어는, 사용된다면, 하나 이상의 기능이나 동작을 처리할 수 있는 단위를 의미하며, 이는 하드웨어 또는 소프트웨어, 또는 하드웨어와 소프트웨어의 결합으로 구현될 수 있음을 알아야 한다.
- [0095] 또한, 본 명세서에서는 각 도면의 각 구성 요소에 대해서 그 도면 부호를 명기함에 있어서, 동일한 구성 요소에 대해서는 이 구성 요소가 비록 다른 도면에 표시되더라도 동일한 도면 부호를 가지고 있도록, 즉 명세서 전체에 걸쳐 동일한 참조 부호는 동일한 구성 요소를 지시하고 있다.
- [0096] 본 명세서에 첨부된 도면에서 본 발명을 구성하는 각 구성 요소의 크기, 위치, 결합 관계 등은 본 발명의 사상을 충분히 명확하게 전달할 수 있도록 하기 위해서 또는 설명의 편의를 위해서 일부 과장 또는 축소되거나 생략되어 기술되어 있을 수 있고, 따라서 그 비례나 축척은 엄밀하지 않을 수 있다.
- [0097] 또한, 이하에서, 본 발명을 설명함에 있어서, 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 구성, 예를 들어, 종래 기술을 포함하는 공지 기술에 대한 상세한 설명은 생략될 수도 있다.
- [0098] 도 6은 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 구현하기 위한 다중 가중치 및 다중 활성화(MWMA) 구조의 스파스 아키텍처 처리 요소의 구성도이다.
- [0099] 도 7은 도 6에 도시된 다중 가중치 및 다중 활성화(MWMA) 구조의 스파스 아키텍처 처리 요소에서 '0' 이 아닌 가중치의 수가 2인 가지 치기 기법의 구성도이다.
- [0100] 도 8은 도 6에 도시된 다중 가중치 및 다중 활성화(MWMA) 구조의 스파스 아키텍처 처리 요소에서 페칭 그룹의 크기(Npar)가 8이고, 가지 치기 그룹의 크기(g)가 4인 처리 요소의 구성도이다.
- [0101] 도 6 내지 도 8을 참조하여 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 구현하기 위한 하드웨어를 설명하면 다음과 같다.
- [0102] 스파스(sparse) 아키텍처는 가중치 저장을 줄이기 위해 '0'이 아닌 가중치만 가중치 메모리에 저장하는 가중치 희박성(sparsity)을 이용한다.
- [0103] 페칭 그룹의 크기(Npar)는 보통 곱셈기(10)의 수(Nmul)보다 크므로 일부 가중치는 '0' 이고, 제로 가중치의 곱셈은 의미가 없으므로, 곱셈기(10)는 '0'이 아닌 가중치와 해당 활성화 데이터를 인가받는다.

- [0104] 페칭 그룹의 크기(Npar) 개수만큼의 활성화 데이터로부터 곱셈기(10)의 수(Nmul)만큼의 활성화 데이터를 선택하는 것은 곱셈기(10)의 수(Nmul) x w x 활성화 개수(Npar)-to-1 멀티플렉서(MUX, 40)를 필요로 하는데, 여기에서 w는 활성화의 비트 너비이다.
- [0105] 적절한 활성화를 선택하기 위해, 도 6에서 회색 직사각형으로 도시된 인덱스와 '0'이 아닌 가중치가 함께 저장된다.
- [0106] 본 발명에서 제안하는 가속기 인식 가지 치기 알고리즘은 가속기 아키텍처에 적합한 보다 일반적인 '0'이 아닌 가중치 패턴을 생성하는데, 가속기 아키텍처와 밀접한 관련이 있는 2가지 매개 변수인 활성화 및 가중치 페칭 그룹과 가중치 그룹을 위해 남겨진 '0'이 아닌 가중치의 수에 중점을 둔다.
- [0107] 즉, 활성화 및 가중치 페칭 그룹의 크기에 따라 내부 버퍼 너비가 결정되고, '0'이 아닌 가중치 수는 요구되는 곱셈기(10)의 개수 및 처리 주기와 관련된다.
- [0108] **가속기 인식 가지 치기 기법 비교**
- [0109] 본 발명의 가지 치기 기법은 가중치 페칭 그룹 내에서 가중치가 제거되어 남아 있는 '0'이 아닌 가중치의 수(Nnon-zero)가 모든 가중치 페칭 그룹에 대해 균일하다.
- [0110] 종래의 가지 치기 기법에서는 '0'이 아닌 가중치의 수(Nnon-zero)에 일정한 제한이 없어, '0' 또는 가지 치기 그룹의 크기(g) 만큼의 개수도 가능하지만, 본 발명에서의 가지 치기 기법은 도 7에 도시된 바와 같이 모든 가중치 추출 그룹에 대해 동일한 수의 '0'이 아닌 가중치를 남긴다.
- [0111] 도 7에서 모든 가중치 페칭 그룹이 가지 치기된 6 개의 가중치(백색 표시)와 두 개의 남아 있는 '0'이 아닌 가중치(회색 표시)를 가진다.
- [0112] 가중치 페칭 그룹 당 남아있는 '0'이 아닌 가중치의 수를 곱셈기 개수(Nmul)의 배수로 설정하면 '0'이 아닌 가중치의 수(Nnon-zero)가 모든 가중치 페칭 그룹에 대해 균일하므로, 스파스 다중 가중치 및 다중 활성화(MWMA) 및 다중 가중치 및 단일 활성화(MWSA) 구조의 오정렬 문제를 해결할 수 있다.
- [0113] 또한, 본 발명은 모든 처리 요소가 같은 수의 가중치 페칭 그룹을 처리할 경우, 처리할 처리 요소의 가중치 수는 자연스럽게 균형을 이루게 되어, 가중치 직렬 구조의 주요 이슈 중 하나인 로드 불균형 문제를 해결할 수 있다.
- [0114] 또한, 본 발명은 가속기의 복잡성, 특히 인덱싱 및 활성화 선택 로직을 줄이기 위해 다음과 같은 알고리즘을 가진다.
- [0115] 먼저, 모든 가중치 페칭 그룹을 m 개의 하위 그룹(가지 치기 그룹)으로 균등하게 나눈다.
- [0116] 가지 치기 그룹의 크기(g)는 곧 활성화 개수(Npar)/가지 치기 그룹 개수(m)가 되고, 각 가지 치기 그룹이 일정한 수(본 실시예에서는 2)의 '0'이 아닌 가중치를 가지도록 가지 치기가 수행된다.
- [0117] 가중치는 가지 치기 그룹의 크기(g) 개의 활성화 중 하나에 대응하기 때문에, 활성화 선택을 위한 멀티플렉서(40)의 폭은 가지 치기 그룹의 크기(g)로 감소될 수 있다.
- [0118] 가지 치기 그룹의 크기(g)가 작을수록, 도 6의 처리 요소 구조는 도 8의 구조로 단순화될 수 있어, 활성화 선택 논리는 더 좁은 멀티플렉서(50)로 단순화된다.
- [0119] 이때, 가지 치기 그룹의 크기(g)가 작아지면 통상적으로 동일한 가지 치기 비율로 네트워크 성능이 저하될 수 있다.
- [0120] 그러나, 본 발명에 의할 경우, 75 %의 가지 치기 비율을 가진 가지 치기 그룹의 크기(g) = 16은 컨볼루션 신경망 성능을 저하시키지 않는다.
- [0121] 즉, 종래 캠브리콘-X의 활성화 개수(Npar) = 256은 곱셈기(10) 개수(Nmul) = 16과 비교하여 크다.
- [0122] 컨볼루션 계층에서 75 %의 일반적인 잘라 내기 비율을 가질 경우, 활성화 개수(Npar) = 64이면 충분하다.
- [0123] 패딩-0의 양을 줄이기 위해 큰 활성화 개수(Npar)를 선택할 수 있지만, 본 발명은 패딩 제로가 불필요하므로 활성화 개수(Npar) = 64가 사용될 수 있다.
- [0124] 그러면 활성화 버퍼의 폭을 64로 줄일 수 있어, 보다 정사각형 같은 메모리를 사용할 수 있으므로 넓은 메모리

보다 면적 면에서 효율적이다.

- [0125] 또한, 색인 논리를 단순화할 수 있다.
- [0126] 불규칙한 '0' 이 아닌 가중치의 색인을 위해 종래의 가지 치기 기법은 두 개의 인접하지 않은 '0' 이 아닌 가중치 사이의 제로 가중치의 수가 저장되는 상대적 색인화를 사용한다.
- [0127] 간격은 4 비트로 인코딩되며, 인코딩 범위보다 큰 간격에는 필러 '0' 삽입이 필요하다.
- [0128] 이와 유사한 색인화가 종래의 캠브리콘-X에서 단계 인덱싱으로 사용되는데, 본 발명에서는 이러한 색인화가 가지 치기 기법으로 간소화된다.
- [0129] 본 발명의 가지 치기 기법에서, 가지 치기는 가지 치기 그룹의 크기(g)의 가지 치기 그룹 내에서 수행된다.
- [0130] 작은 가지 치기 그룹의 크기(g)는 직접 색인화를 가능케 하는데, 색인은 가지 치기 그룹 내에서 '0' 이 아닌 가중치의 위치를 나타낸다.
- [0131] 가지 치기 그룹의 크기(g)가 작기 때문에 캠브리콘-X의 단계 인덱싱보다 훨씬 간단하며 가중치 저장 장치의 낭비를 제거하는 직접 색인화를 사용하는 경우에도 인덱싱 비트 폭 w_i 는 최대값이 $\log_2(g)$ 비트로 작다.
- [0132] 종래의 가지 치기 기법은 임계 값보다 작은 크기의 가중치가 제거되고, 가지 치기 비율이 가지 치기 후에 결정되므로 가지 치기 비율을 쉽게 예측할 수 없다.
- [0133] 따라서, 가지 치기 비율이 목표 비율에 도달하지 않은 경우, 더 작은 임계 값으로 다시 가지 치기가 시도되고, 이러한 프로세스는 가지 치기 목표 비율에 도달할 때까지 반복된다.
- [0134] 그러나, 본 발명의 가지 치기 기법의 목표는 가지 치기 그룹당 가지 치기 가중치의 수로서, 가지 치기 비율은 직접적으로 포커싱된다.
- [0135] 가지 치기 그룹에서 가지 치기 가중치의 수가 p 개이면, 가지 치기 비율은 가지 치기 가중치 수(p) / 가지 치기 그룹의 크기(g)가 된다.
- [0136] 본 발명의 가지 치기 기법은 여러 가지 기준으로 가지 치기할 가중치를 선택할 수 있으나, 실험 결과에서는 가중치의 크기로 가지 치기할 가중치를 선택하므로, 가장 작은 크기의 가중치는 가지 치기 그룹에서 처음 가지 치기된다.
- [0137] 또한, 본 발명의 가지 치기 기법은 다양한 가속기 구조에 적용될 수 있다.
- [0138] 즉, 가속기의 구조에 따라 채널 축, 필터 축, 공간 축을 따라 활성화와 가중치를 가져와 처리한다.
- [0139] 또한, 본 발명의 가지 치기 기법은 완전 연결 계층(fully connected layer)에도 적용될 수 있다.
- [0140] 즉, 완전 연결 계층에서 가중치는 행 축을 따라 가중치에 다른 활성화가 곱해지고, 열 축을 따라 가중치에 하나의 활성화가 곱해진다.
- [0141] 다중 가중치 및 다중 활성화(MWMA) 구조의 처리 요소에서는 가중치가 행 축을 따라 그룹화되고, 각 그룹은 고정된 수의 '0' 이 아닌 가중치를 갖도록 가지 치기되어 행 축을 따라 가중치가 처리된다.
- [0142] 반면, 다중 가중치 및 단일 활성화(MWSA) 구조의 처리 요소에서는 가중치가 열 축을 따라 그룹화되고, 각 그룹은 고정된 수의 '0' 이 아닌 가중치를 갖도록 가지 치기되어 열 축을 따라 가중치가 처리된다.
- [0143] **실험 결과**
- [0144] 도 9는 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 구현한 직후에 측정된 이미지 넷 검증 정확도에 대한 표이다.
- [0145] 도 10은 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 및 재학습을 구현한 직후에 측정된 이미지 넷 검증 정확도에 대한 표이다.
- [0146] 도 6 내지 도 10을 참조하여 본 발명에 따른 가속기 인식 가지 치기 방법을 구현한 실험 결과를 설명하면 다음과 같다.
- [0147] 본 발명의 가지 치기 기법이 제약 조건에도 불구하고 컨볼루션 신경망의 성능을 잘 보존할 수 있음을 보여주기 위해 이미지 넷(ImageNet) 2012 유효성 확인 데이터 집합에 대하여 상위 5 가지 정확도가 측정되었다.

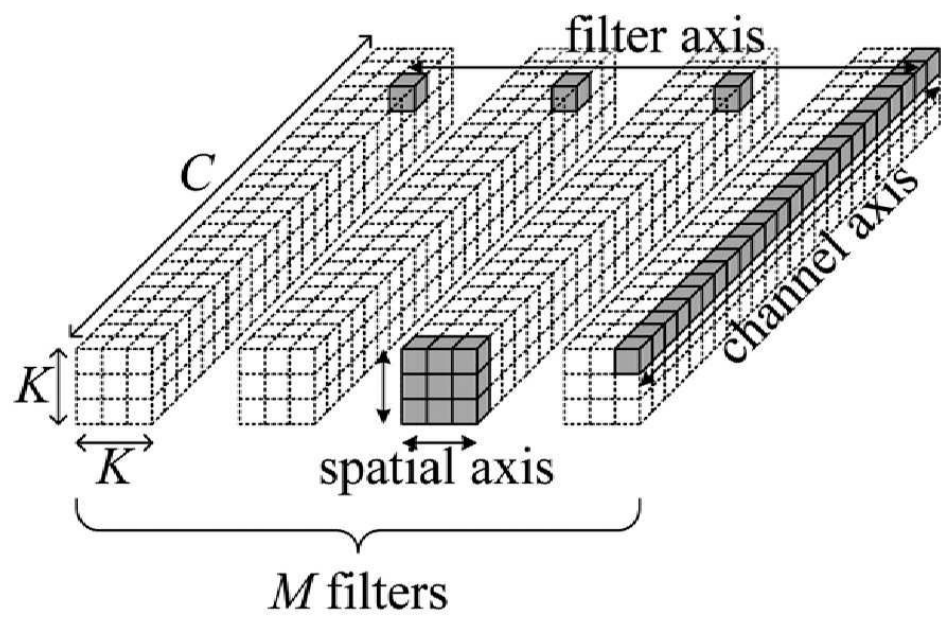
- [0148] 1 회 가지 치기 후 재학습은 12 회(epoch) 동안 5×10^{-4} 의 학습 속도로 수행되었다.
- [0149] 원래의 정확도가 회복되지 않으면, 8 회 재학습이 10^{-4} 의 학습 속도로 부가적으로 수행되었다.
- [0150] 재학습 1이 충분하지 않은 경우 재학습 2가 적용되었으며, 학습 속도는 5×10^{-4} 로 시작하여 유효성 검사 정확도가 포화된 경우 10^{-4} , 10^{-5} 및 10^{-6} 으로 감소한다.
- [0151] 재학습 3에서는 재학습 2가 증분 가지 치기(incremental pruning)와 함께 적용된다.
- [0152] 학습 정확도가 10^{-6} 의 학습 속도로 포화되면 p는 증가하고, 학습은 5×10^{-4} 의 학습 속도로 재개된다.
- [0153] 도 9에서 가중치는 채널 축을 따라 그룹화되며, 첫 번째 컨볼루션 계층은 다른 계층보다 훨씬 적은 가중치 및 연산 수를 가지므로 가지 치기되지 않는다.
- [0154] 도 9는 30 % (3/8 또는 6/16) 가지 치기가 이미 이미지 넷 검증 정확도를 떨어뜨리기 시작했음을 보여준다.
- [0155] 그러나, 이미지 넷 검증 정확도의 이러한 열화는 도 10에 나타난 바와 같이 재학습으로 회복될 수 있다.
- [0156] 도 10에서, 재학습 1 및 재학습 3이 적용되었으며, 재학습 3의 결과는 별표로 표시된다.
- [0157] 도 10에서 보는 바와 같이, VGG16, ResNet-50 및 ResNet-152를 포함한 매우 깊은 네트워크에서 재학습 1을 사용하면 최대 75 % (6/8 또는 12/16)의 가지 치기로 검증 정확도를 기본 정확도로 복구할 수 있었다.
- [0158] VGG16의 결과는 VGG16에서 컨볼루션 계층의 가지 치기 비율이 75 % 정도인 종래의 비 구조화 가지 치기 알고리즘의 결과와 일치한다.
- [0159] 상기 실험은 가속기 제약을 고려하여 제안된 가지 치기 기법으로 유사한 가지 치기 비율에 도달할 수 있음을 보여준다.
- [0160] 또한, 75 % 가지 치기 비율이 나머지 네트워크인 ResNet-50 및 ResNet-152에서의 정확도를 낮추지 않음을 보여준다.
- [0161] 또한, AlexNet에서 재학습 3은 (p; 가지 치기 그룹의 크기(g)) = (5; 8) 또는 (10; 16)으로 시작하고, p는 1씩 증가한다.
- [0162] AlexNet 컨볼루션 계층의 가지 치기 비율은 종래의 비 구조화 가지 치기 알고리즘의 경우, 약 65 %이지만 제안된 방식은 81.25 %의 가지 치기 비율에 도달할 수 있음을 알 수 있다.
- [0163] **가속기 인식 가지 치기 알고리즘**
- [0164] 도 11은 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 구현하기 위한 장치의 개략적인 블록도로서, 데이터 그룹핑부(100), 가지 치기부(200) 및 재학습부(300)를 구비한다.
- [0165] 도 12는 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 이용하고 재학습하여 성능이 복구된 신경망을 획득하는 방법에 대한 순서도이다.
- [0166] 도 6 내지 도 12를 참조하여 본 발명에 따른 가속기 인식 가지 치기 알고리즘을 설명하면 다음과 같다.
- [0167] 먼저, 가지 치기부(200)가 딥 러닝으로 학습된 신경망 데이터를 전달받아 가지 치기하면(S200), '0'이 아닌 가중치가 적은 신경망이 생성되고(S300), 재학습부(300)가 이 데이터에 재학습 과정을 수행하면(S400), '0'이 아닌 가중치가 적고 성능, 즉 검증 정확도가 복구된 신경망이 획득된다(S500).
- [0168] 이때, 본 발명에 따른 가속기 인식 가지 치기 방법은 가지 치기 동작 전에 데이터 그룹핑부(100)가 딥 러닝으로 학습화된 데이터를 인가받아, 가속기 회로에 맞춤형으로, 즉 데이터를 동시에 폐치하는 형태가 다른 가속기 회로에 맞추어 3개의 축(채널 축, 필터 축, 공간 축) 중 하나의 축을 따라 일정 크기로 그룹핑한다(S100).
- [0169] 이때, 본 발명에 따른 가속기 인식 가지 치기 방법의 가지 치기 동작은 가지 치기부(200)가 가중치 중에서 일정 개수(도 7의 실시예에서 2개)는 '0'이 아닌 가중치를 유지하고(S210), 가중치 중에서 나머지 개수(도 7의 실시예에서 6개)는 '0'인 가중치가 되도록 가지 치기한다(S220).
- [0170] 한편, 본 발명에 따른 기록매체는 전술한 본 발명에 따른 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법

을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 매체이다.

- [0171] 이러한 기록매체에는 (a) 데이터 그룹핑부(100)가 딥 러닝으로 학습화된 데이터를 인가받아, 동시에 폐치하는 형태가 다른 가속기 회로에 맞추어 소정의 축을 따라 일정 크기로 그룹핑하는 단계; (b) 가지 치기부(200)가 상기 그룹핑된 데이터를 인가받아, 가중치 중에서 일정 개수는 '0' 이 아닌 가중치를 유지하고, 나머지 개수는 '0' 인 가중치가 되도록 가지 치기하는 단계; (c) 상기 (a) 단계보다 상기 '0' 이 아닌 가중치가 적은 신경망이 생성되는 단계; 및 (d) 재학습부(300)에 의해 상기 가지 치기된 데이터에 재학습 과정이 수행되어, 상기 (a) 단계보다 상기 '0' 이 아닌 가중치가 적고 검증 정확도가 복구된 신경망이 획득되는 단계; 가 포함된다.
- [0172] 여기에서, 기록매체란 컴퓨터 시스템이 읽어들이 수 있는 데이터를 저장할 수 있는 모든 종류의 기록장치를 포함한다. 일 예로써, ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, USB 메모리 또는 광 데이터 장치 등이 있다.
- [0173] 이와 같이 본 발명은 병렬 처리 데이터 그룹에 해당하는 각 가중치 그룹에 대해 동일한 수의 가중치가 유지되도록 가지 치기가 수행되어, 종래의 스파스 가속기의 효율성을 향상시키고 논리 복잡성을 줄일 수 있는 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 제공한다.
- [0174] 또한, 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기에서 의미 없는 덧셈이나 곱셈을 제거하여 연산량을 감소시키고 곱셈기의 낭비를 제거하여 회로를 단순화시킬 수 있는 컨볼루션 신경망을 위한 가속기 인식 가지 치기 방법을 제공한다.
- [0175] 또한, 컨볼루션 신경망을 위한 상기 가속기 인식 가지 치기 방법을 컴퓨터에서 실행시키기 위한 프로그램이 탑재된 기록 매체를 제공한다.
- [0176] 이를 통하여, 주문형 반도체 가속기 또는 필드 프로그래머블 게이트 어레이 가속기에서 내부 버퍼의 오정렬이 해결되어 곱셈기의 낭비가 방지되고 회로가 단순화된다.
- [0177] 또한, 가속기의 처리 요소의 활성화 과정에서 모든 처리 요소가 완료될 때까지 기다릴 필요가 없어, 로드 불균형 문제가 해결되고 성능이 현저하게 향상된다.
- [0178] 또한, 가속기의 컨볼루션 계층의 가지 치기 및 재학습을 통해, 열화된 검증 정확도를 기본 정확도로 복구할 수 있게 된다.
- [0179] 이상, 일부 예를 들어서 본 발명의 바람직한 여러 가지 실시예에 대해서 설명하였지만, 본 "발명을 실시하기 위한 구체적인 내용" 항목에 기재된 여러 가지 다양한 실시예에 관한 설명은 예시적인 것에 불과한 것이며, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 이상의 설명으로부터 본 발명을 다양하게 변형하여 실시하거나 본 발명과 균등한 실시를 행할 수 있다는 점을 잘 이해하고 있을 것이다.
- [0180] 또한, 본 발명은 다른 다양한 형태로 구현될 수 있기 때문에 본 발명은 상술한 설명에 의해서 한정되는 것이 아니며, 이상의 설명은 본 발명의 개시 내용이 완전해지도록 하기 위한 것으로 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 본 발명의 범주를 완전하게 알려주기 위해 제공되는 것일 뿐이며, 본 발명은 청구범위의 각 청구항에 의해서 정의될 뿐임을 알아야 한다.

도면

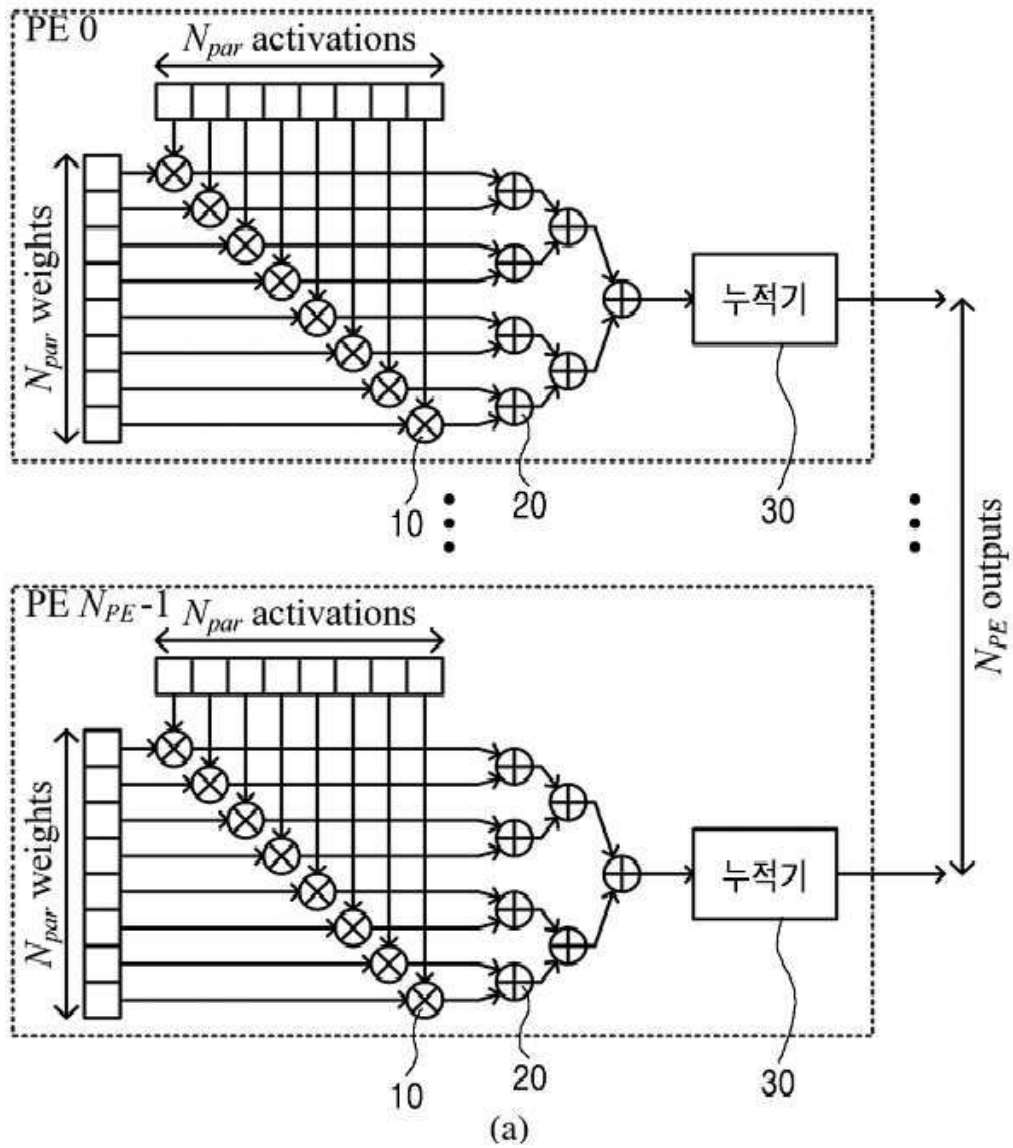
도면1



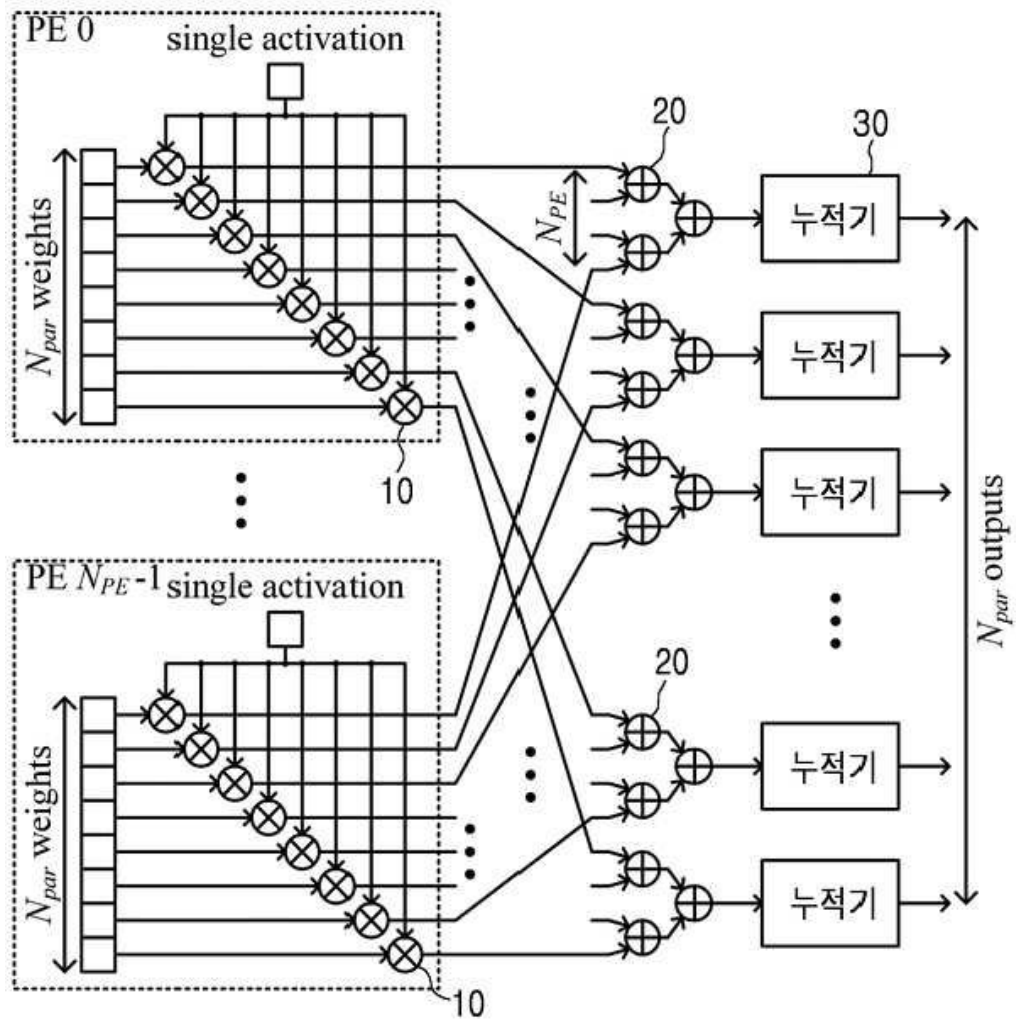
도면2

granularity	pruned weights
channel-wise or filter-wise	$w(:, n, :, :)$ or $w(m, :, :, :)$
shape-wise or GPU-aware	$w(:, n, i, j)$
2D/1D granularity	$w(m, n, :, :)$ $w(m, n, i, :)$ $w(m, n, :, j)$

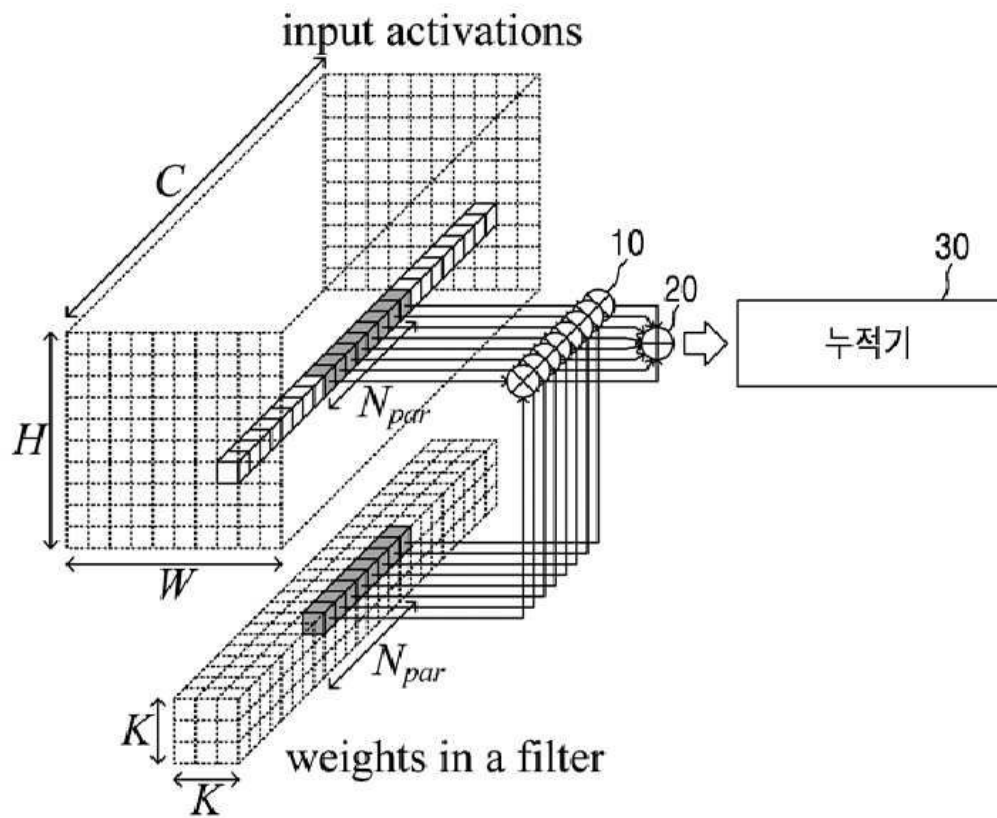
도면3



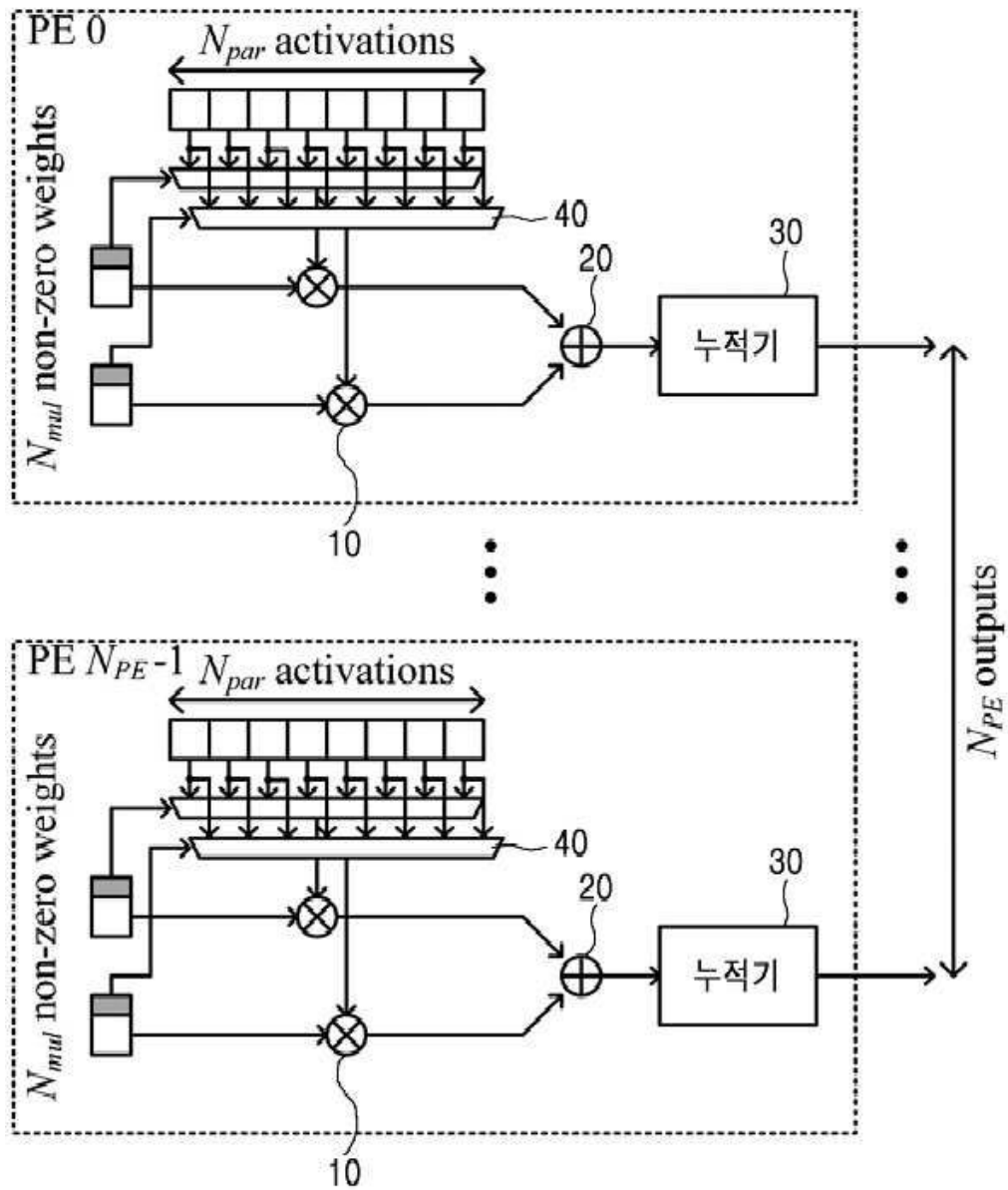
도면4



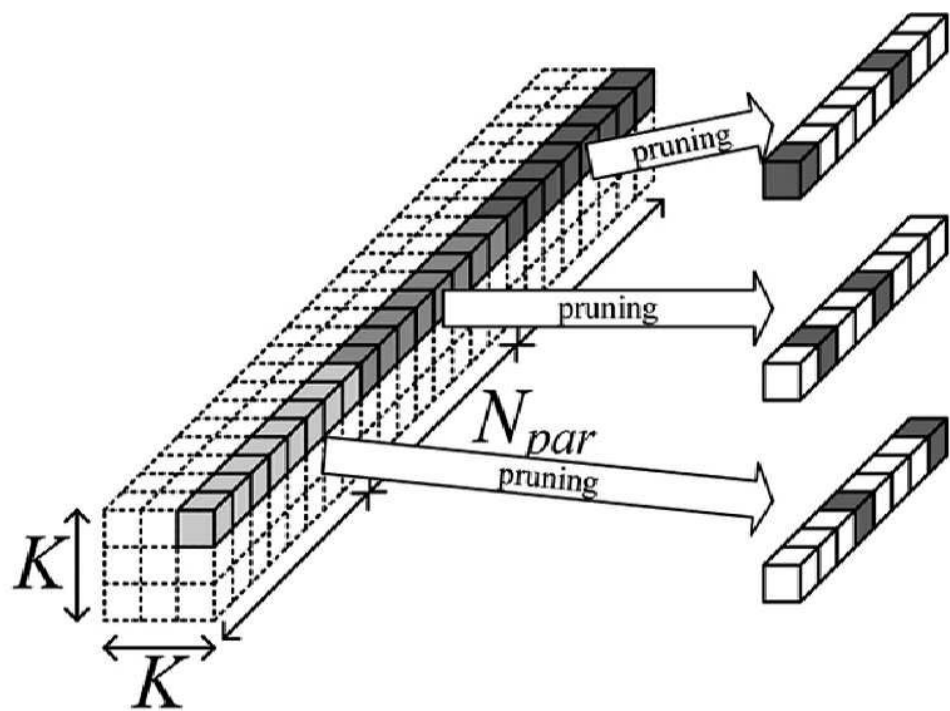
도면5



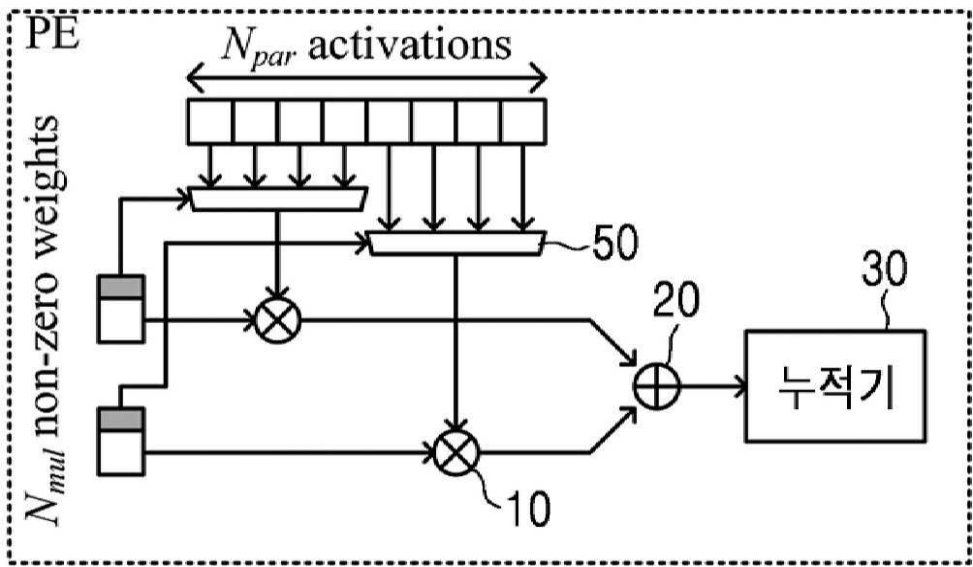
도면6



도면7



도면8



도면9

g	p	AlexNet	SqueezeNet v1.0	VGG16	ResNet-50	ResNet-152
-	-	0.79812	0.80390	0.88444	0.91136	0.92201
8	1	0.79676	0.79566	0.88308	0.90942	0.92123
8	2	0.79166	0.76610	0.88020	0.90296	0.91691
8	3	0.76888	0.68436	0.85524	0.88272	0.90716
8	4	0.65020	0.35740	0.70204	0.78792	0.86372
8	5	0.32194	0.03494	0.12234	0.17206	0.46580
16	1	0.79826	0.80240	0.88466	0.91158	0.92217
16	2	0.79762	0.79938	0.88356	0.91082	0.92177
16	4	0.79352	0.77876	0.88190	0.90800	0.92073
16	6	0.77996	0.71332	0.86642	0.88880	0.91363
16	8	0.69992	0.43120	0.75344	0.80744	0.87900
16	9	0.60840	0.23148	0.55150	0.59390	0.80360
16	10	0.38612	0.07712	0.20118	0.32418	0.65674

도면10

g	p	AlexNet	SqueezeNet v1.0	VGG16	ResNet-50	ResNet-152
-	-	0.79812	0.80390	0.88444	0.91136	0.92201
8	4	0.80446	*0.80514	0.90706	0.91950	0.93027
8	5	*0.80580	*0.79932	0.90454	0.91678	0.92787
8	6	*0.80422	*0.77586	0.89914	0.91142	0.92329
8	7	*0.79472	*0.68064	0.88480	0.88862	0.90997
16	8	0.80456	*0.80928	0.90734	0.91956	0.93021
16	9	0.80384	*0.80650	0.90798	0.91800	0.92885
16	10	*0.80620	*0.80294	0.90544	0.91860	0.92745
16	11	*0.80744	*0.79860	0.90378	0.91550	0.92548
16	12	*0.80498	*0.78800	0.90222	0.91350	0.92477
16	13	*0.80216	*0.76348	0.89652	0.90876	0.92204

도면11



도면12

