(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0120434
(43) 공개일자 2021년10월07일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2006.01) *G06F 17/15* (2006.01)
G06F 17/16 (2006.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06F 17/153 (2013.01)
(21) 출원번호 10-2020-0037129
(22) 출원일자 2020년03월26일
심사청구일자 없음

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
강석주
서울특별시 마포구 백범로 35 서강대학교 최양업관 419호
안세현
서울특별시 강동구 진황도로 61길 7, 102동 402호 (둔촌동, 현대아파트)
(74) 대리인
이철희

전체 청구항 수 : 총 16 항

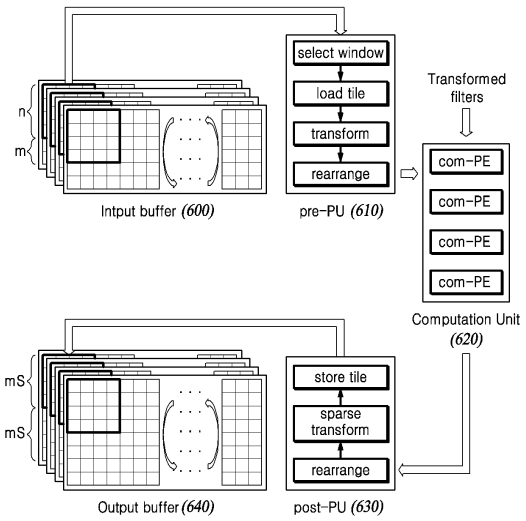
(54) 발명의 명칭 **전치 합성곱 신경망의 가속 장치 및 동작 방법**

(57) 요 약

전치 합성곱 신경망의 가속 장치 및 동작 방법을 개시한다.

본 발명의 일 측면에 의하면, 전치 합성곱 신경망의 가속 장치에 있어서, 상기 전치 합성곱 신경망으로부터 특징맵들을 입력 받는 입력 버퍼; 각 특징맵들 내에서 대응되는 입력 블록들을 선택하고 위노그래드 도메인으로 변환하며, 위노그래드 도메인 상에서 상기 입력 블록들의 요소와 제로 웨이트들이 포함된 커널들의 요소를 재배열하는 전처리부; 재배열된 입력 블록들과 재배열된 커널들에 대해 요소별 곱셈(element-wise multiply) 연산을 수행함으로써 출력 블록들을 생성하는 연산부; 상기 출력 블록들을 위노그래드 도메인에서 공간 도메인으로 변환하는 후처리부; 및 변환된 출력 블록들에 기초하여 출력 특징맵을 생성하여 출력하는 출력 버퍼를 포함하는 가속 장치 및 동작 방법를 제공한다.

대 표 도 - 도6



(52) CPC특허분류
G06F 17/16 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 20PQWO-B153369-02
부처명 행정안전부 (산업통상자원부, 과학기술정보통신부, 국토교통부)
과제관리(전문)기관명 국토교통과학기술진흥원
연구사업명 스마트 도로조명 플랫폼 개발 및 실증연구
연구과제명 스마트 도로조명 활용 도시재난안전관리 연계 기술 개발
기 여 율 1/2
과제수행기관명 서강대학교산학협력단
연구기간 2020.01.01 ~ 2020.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711116161
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 대학ICT연구센터지원사업
연구과제명 인공지능 서비스 실현을 위한 지능형 반도체 설계 핵심기술 개발
기 여 율 1/2
과제수행기관명 서강대학교 산학협력단
연구기간 2020.01.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

전치 합성곱 신경망의 가속 장치에 있어서,

상기 전치 합성곱 신경망으로부터 특징맵들을 입력 받는 입력 버퍼;

위노그래드 도메인 상에서 제로 웨이트들이 포함된 커널들의 요소를 재배열하는 재배열부;

각 특징맵들 내에서 대응되는 입력 블록들을 선택하고 위노그래드 도메인으로 변환하며, 위노그래드 도메인 상에서 상기 입력 블록들의 요소를 상기 커널들의 요소에 따라 재배열하는 전처리부;

재배열된 입력 블록들과 재배열된 커널들에 대해 요소별 곱셈(element-wise multiply) 연산을 수행함으로써 출력 블록들을 생성하는 연산부;

상기 출력 블록들을 위노그래드 도메인에서 공간 도메인으로 변환하는 후처리부; 및

변환된 출력 블록들에 기초하여 출력 특징맵을 생성하여 출력하는 출력 버퍼;

를 포함하는 가속 장치.

청구항 2

제1항에 있어서,

상기 재배열부는,

상기 커널들이 하나 이상의 제로 행벡터를 포함하도록 상기 커널들의 요소를 재배열하고,

상기 전처리부는,

상기 커널들의 재배열된 요소에 대응되도록 상기 입력 블록들의 요소를 재배열하는 가속 장치.

청구항 3

제2항에 있어서,

상기 연산부는 상기 재배열된 커널들에서 상기 제로 행벡터를 제외한 벡터들에 대해서만 요소별 곱셈을 수행하는 가속 장치.

청구항 4

제2항에 있어서,

상기 연산부는 복수의 프로세싱 엘리먼트(processing elements, Pes)를 포함하고, 상기 하나 이상의 제로 행벡터에 해당하는 프로세싱 엘리먼트들에 대해 연산을 스킵(skip)하는 가속 장치.

청구항 5

제1항에 있어서,

상기 재배열부는 상기 커널들을 각 행렬이 하나 이상의 제로 행벡터를 포함하는 복수의 커널 행렬들로 병합하고,

상기 전처리부는 상기 입력 블록들을 하나의 입력 행렬로 변환하며,

상기 연산부는 상기 복수의 커널 행렬들 내에서 상기 하나 이상의 제로 행벡터를 제외한 행벡터들에 대해 상기 입력 행렬과의 요소별 곱셈을 수행함으로써 출력 행렬을 생성하고,

상기 후처리부는 상기 출력 행렬을 상기 출력 블록으로 변환하는 가속 장치.

청구항 6

제1항에 있어서,

상기 커널들은,

상기 전치 합성곱 신경망 내 전치 합성곱 레이어의 디컨볼루션 커널들로부터 컨볼루션 커널로 변환된 것인 가속 장치.

청구항 7

제1항에 있어서,

상기 입력 버퍼와 상기 출력 버퍼는 상기 연산부와 병렬적으로 동작하는 가속 장치.

청구항 8

전치 합성곱 신경망에 대한 가속 장치의 동작 방법에 있어서,

상기 전치 합성곱 신경망으로부터 특징맵들을 입력 받는 과정;

각 특징맵들 내에서 대응되는 입력 블록들을 선택하고 위노그래드 도메인으로 변환하는 과정;

위노그래드 도메인 상에서 상기 입력 블록들의 요소와 제로 웨이트들이 포함된 커널들의 요소를 재배열하는 과정;

재배열된 입력 블록들과 재배열된 커널들에 대해 요소별 곱셈 연산을 수행함으로써 출력 블록들을 생성하는 과정;

상기 출력 블록들을 위노그래드 도메인에서 공간 도메인으로 변환하는 과정; 및

변환된 출력 블록들에 기초하여 출력 특징맵을 생성하여 출력하는 과정;

을 포함하는 동작 방법.

청구항 9

제1항에 있어서,

상기 재배열하는 과정은,

상기 커널들이 하나 이상의 제로 행벡터를 포함하도록 상기 커널들의 요소를 재배열하고, 상기 커널들의 재배열된 요소에 대응되도록 상기 입력 블록들의 요소를 재배열하는 과정을 포함하는 동작 방법.

청구항 10

제9항에 있어서,

상기 출력 블록들을 생성하는 과정은,

상기 재배열된 커널들에서 상기 제로 행벡터를 제외한 벡터들에 대해서만 요소별 곱셈을 수행하는 과정을 포함하는 동작 방법.

청구항 11

제9항에 있어서,

상기 출력 블록들을 생성하는 과정은,

상기 가속 장치가 복수의 프로세싱 엘리먼트를 포함할 때, 상기 하나 이상의 제로 행벡터에 해당하는 프로세싱 엘리먼트들에 대해 연산을 스킵하는 동작 방법.

청구항 12

제8항에 있어서,

상기 커널들을 각 행렬이 하나 이상의 제로 행벡터를 포함하는 복수의 커널 행렬들로 병합하고, 상기 입력 블록들을 하나의 입력 행렬로 변환하는 과정;

상기 복수의 커널 행렬들 내에서 상기 하나 이상의 제로 행벡터를 제외한 행벡터들에 대해 상기 입력 행렬과의 요소별 곱셈을 수행함으로써 출력 행렬을 생성하는 과정; 및

상기 출력 행렬을 상기 출력 블록으로 변환하는 과정을 더 포함하는 동작 방법.

청구항 13

제8항에 있어서,

상기 커널들은,

상기 전치 합성곱 신경망 내 전치 합성곱 레이어의 디컨볼루션 커널들로부터 컨볼루션 커널로 변환된 것인 동작 방법.

청구항 14

제8항에 있어서,

상기 입력 받는 과정은,

상기 입력 블록들로부터 상기 출력 블록들이 생성되는 도중에 상기 특징맵들 내에서 다른 입력 블록들을 입력 받는 과정을 포함하는 동작 방법.

청구항 15

전치 합성곱 신경망의 가속 장치에 있어서,

상기 전치 합성곱 신경망으로부터 특징맵들을 입력 받는 입력 버퍼;

위노그래드 도메인 상에서 제로 웨이트들을 포함하는 커널들로부터 제로 행벡터를 포함하도록 재배열된 커널들이 저장된 메모리부;

각 특징맵들 내에서 대응되는 입력 블록들을 선택하고 위노그래드 도메인으로 변환하며, 위노그래드 도메인 상에서 상기 입력 블록들의 요소를 상기 커널들의 요소에 따라 재배열하는 전처리부;

재배열된 입력 블록들과 재배열된 커널들에 대해 요소별 곱셈 연산을 수행함으로써 출력 블록들을 생성하는 연산부;

상기 출력 블록들을 위노그래드 도메인에서 공간 도메인으로 변환하는 후처리부; 및

변환된 출력 블록들에 기초하여 출력 특징맵을 생성하여 출력하는 출력 버퍼;

를 포함하는 가속 장치.

청구항 16

제8항 내지 제14항 중 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한, 컴퓨터로 읽을 수 있는 기록매체.

발명의 설명

기 술 분 야

[0001] 본 발명의 실시예들은 전치 합성곱 신경망의 디컨볼루션 연산 시 특징맵과 커널 간 연산량을 줄여 신경망 연산을 가속하는 장치 및 그의 동작 방법에 관한 것이다.

배 경 기 술

[0002] 이 부분에 기술된 내용은 단순히 본 발명에 대한 배경 정보를 제공할 뿐 종래기술을 구성하는 것은 아니다.

[0003] 최근 딥러닝 기반 컨볼루션 뉴럴 네트워크(convolution neural network, CNN)는 객체 인식, 분류, 그리고 음성

인식 분야 등 다양한 분야에서 뛰어난 성능을 보여주고 있다. 하지만, CNN의 트레이닝 데이터 셋을 만들기 위해 개발자는 주어진 데이터마다 라벨링(labeling)을 수행해야 한다.

[0004] 이러한 문제를 해결하기 위해, 생성적 적대 신경망 (Generative adversarial network)의 생성기(Generator)가 이용될 수 있다. 생성적 적대 신경망이란 소수의 데이터로부터 새로운 샘플 데이터를 생성하는 신경망이다. 구체적으로, 생성적 적대 신경망은 생성기와 구별기를 포함하며, 생성기는 새로운 데이터를 생성하고, 구별기는 생성된 데이터의 진위를 구별하도록 학습된다. 여기서, 생성기는 주로 디컨볼루션 레이어라 불리는 전치 합성곱 레이어들(Transposed convolutional layers)을 포함한다. 전치 합성곱 레이어들은 합성곱 레이어들이 원본 데이터에서 추출한 특징을 이용하여 원본 데이터와 유사한 데이터를 생성하며, 딥러닝 기반 해상도 증가(Super-resolution), 딥러닝 기반 HDR(High dynamic imaging) 등 영상을 출력하는 딥러닝 기반 알고리즘에 많이 이용된다.

[0005] 전치 합성곱 레이어들은 방대한 연산량을 처리해야 하기 때문에, 저전력으로 빠른 연산을 수행하기 위한 하드웨어 가속기가 이용된다. 하드웨어 가속기를 이용될 수 있는 전치 합성곱 층의 디컨볼루션 연산은 세 가지 종류가 있다.

[0006] 도 1a, 도 1b 및 도 1c는 가속기를 이용한 전치 합성곱 층의 디컨볼루션 연산 종류를 설명하기 위해 예시한 도면이다.

[0007] 도 1a, 도 1b 및 도 1c에는 디컨볼루션 커널(K_D)을 이용하여 입력 블록(100, 110, 120)으로부터 출력 블록(102, 112, 124)를 생성하는 과정이 도시되어 있고, 도 1c의 경우, 서브 출력 특징맵들(122)을 더 이용한다.

[0008] 도 1a는 하나의 입력 픽셀로부터 디컨볼루션 커널 크기의 출력 블록을 생성하는 방법이며, 도 1b는 제로 패딩된 입력 블록으로부터 디컨볼루션 커널 크기의 출력 블록을 생성하는 방법이며, 도 1c는 디컨볼루션 커널을 컨볼루션 커널로 변환한 후 컨볼루션 연산을 통해 입력 블록으로부터 출력 블록을 생성하는 방법이다.

[0009] 도 1a에 도시된 방법은 일반적인 전치 합성곱 층의 디컨볼루션 연산 방법이다. 이 방법은 입력 블록(100) 내 하나의 요소(element)로부터 출력 블록(102) 내 다수의 요소를 생성한다. 다만, 이 방법은 일반적으로 전치 합성곱 층은 입력 특징맵에서 인접한 요소들끼리 연산 결과가 겹치는 문제가 발생한다. 즉, 인접한 입력 요소로부터 생성되는 두 출력 블록이 겹치는 오버랩핑 섬(over-lapping sum) 문제가 발생한다. 오버랩핑된 영역은 입력 블록(100) 내 여러 요소들에 대해 오버랩핑된 뒤 모두 합산됨으로써, 최종적으로 출력 픽셀 또는 출력 영역이 완성될 수 있다. 신경망 가속기가 오버랩핑 영역의 합산을 위해서는 이전 입력 픽셀의 출력 픽셀 값을 불러와야 하고, 외부 메모리(Off-chip memory)에서 데이터를 빈번하게 읽어와야 한다. 따라서, 가속기의 외부 메모리 데이터 읽기 횟수가 증가하기 때문에 가속기의 연산 속도가 저하되고, 에너지 소비가 증가한다는 문제점이 있다.

[0010] 도 1b에 도시된 방법은 일반적인 전치 합성곱 연산 방법으로서 입력 특징 맵(Input feature map)의 픽셀들 사이에 영(zero) 값들을 삽입하여 연산하는 방법이다. 도 1b의 방법을 가속기로 구현하는 경우 오버랩핑 섬 문제를 해결할 수 있다. 하지만, 입력 픽셀 사이에 제로 픽셀을 삽입한 후에 디컨볼루션 연산을 수행하는 것은 가속기의 연산량을 증가시키기 때문에 비효율적이라는 문제점이 있다.

[0011] 도면 1c에 도시된 방법은 전치 합성곱 층을 합성곱 층으로 변환한 후 컨볼루션 연산을 수행하는 방법이다. 이를 디컨볼루션 레이어의 컨볼루션 레이어 변환(transform the deconvolution layer into the convolution layer, TDC)이라 한다. TDC 방법은 입력 특징맵(120)의 입력 블록으로부터 서브 출력 특징맵들(122)을 생성한 후 서브 출력 특징맵들(122)을 이용하여 출력 특징맵을 생성하는 방법이다. 이를 위해, 디컨볼루션 커널을 컨볼루션 커널로 변환한다. 입력 특징맵(120)에 복수의 컨볼루션 커널을 적용함으로써, 서브 출력 특징맵들(122)을 생성하며, 이를 공간적으로 합산하면 출력 블록(124)을 생성할 수 있다.

[0012] 이 방법은 전치 합성곱 층을 공간적으로 변환하는 것이므로 도 1a, 도 1b와 동일한 결과를 얻을 수 있다. 또한, 병렬적으로 연산할 수 있기 때문에 연산 속도(Throughput)가 증가하는 장점이 있다. 다시 말하면, TDC 방법은 디컨볼루션 연산과 같은 결과를 출력하되, 디컨볼루션 레이어를 병렬적으로 연산하여 시간당 연산 처리량을 증가시킬 수 있다.

[0013] 다만, 변환된 컨볼루션 커널에는 제로 값들이 많이 포함되어 있고 행렬 곱셈을 수행해야하므로, 하드웨어 가속기 상에서 제로 값들을 효율적으로 처리하는 방법에 대한 연구가 필요하다.

발명의 내용

해결하려는 과제

[0014] 본 발명의 실시예들은, 전치 합성곱 층의 디컨볼루션 커널로부터 변환된 컨볼루션 커널의 제로 웨이트들(zero weights)에 대한 연산을 줄이기 위한 신경망 가속 장치 및 동작 방법을 제공하는 데 주된 목적이 있다.

과제의 해결 수단

[0015] 본 발명의 일 측면에 의하면, 전치 합성곱 신경망의 가속 장치에 있어서, 상기 전치 합성곱 신경망으로부터 특징맵들을 입력 받는 입력 버퍼; 각 특징맵들 내에서 대응되는 입력 블록들을 선택하고 위노그래드 도메인으로 변환하며, 위노그래드 도메인 상에서 상기 입력 블록들의 요소와 제로 웨이트들이 포함된 커널들의 요소를 재배열하는 전처리부; 재배열된 입력 블록들과 재배열된 커널들에 대해 요소별 곱셈(element-wise multiply) 연산을 수행함으로써 출력 블록들을 생성하는 연산부; 상기 출력 블록들을 위노그래드 도메인에서 공간 도메인으로 변환하는 후처리부; 및 변환된 출력 블록들에 기초하여 출력 특징맵을 생성하여 출력하는 출력 버퍼를 포함하는 가속 장치를 제공한다.

[0016] 본 실시예의 다른 측면에 의하면, 전치 합성곱 신경망에 대한 가속 장치의 동작 방법에 있어서, 상기 전치 합성곱 신경망으로부터 특징맵들을 입력 받는 과정; 각 특징맵들 내에서 대응되는 입력 블록들을 선택하고 위노그래드 도메인으로 변환하는 과정; 위노그래드 도메인 상에서 상기 입력 블록들의 요소와 제로 웨이트들이 포함된 커널들의 요소를 재배열하는 과정; 재배열된 입력 블록들과 재배열된 커널들에 대해 요소별 곱셈 연산을 수행함으로써 출력 블록들을 생성하는 과정; 상기 출력 블록들을 위노그래드 도메인에서 공간 도메인으로 변환하는 과정; 및 변환된 출력 블록들에 기초하여 출력 특징맵을 생성하여 출력하는 과정을 포함하는 동작 방법을 제공한다.

발명의 효과

[0017] 이상에서 설명한 바와 같이 본 발명의 일 실시예에 의하면, 전치 합성곱 층의 디컨볼루션 커널에 위노그래드 변환을 적용하고, 변환된 컨볼루션 커널의 제로 웨이트들을 재배열한 후 재배열된 제로 웨이트들에 대한 연산을 하드웨어 상에서 생략함으로써, 하드웨어 가속 장치의 연산 속도를 높이고, 에너지 소모를 줄일 수 있다.

도면의 간단한 설명

[0018] 도 1a, 도 1b 및 도 1c는 가속기를 이용한 전치 합성곱 층의 디컨볼루션 연산 종류를 설명하기 위해 예시한 도면이다.

도 2는 TDC 방법을 이용하는 디컨볼루션 연산을 구체적으로 설명하기 위해 예시한 도면이다.

도 3은 디컨볼루션 커널 크기가 5일 때 TDC 기반 디컨볼루션 연산 방법을 설명하기 위해 예시한 도면이다.

도 4는 디컨볼루션 커널 크기가 4일 때 TDC 기반 디컨볼루션 연산 과정을 예시한 도면이다.

도 5는 본 발명의 일 실시예에 따른 재배열 과정과 연산 과정을 설명하기 위해 예시한 도면이다.

도 6은 본 발명의 일 실시예에 따른 가속 장치의 구성요소를 예시한 구성도다.

도 7은 본 발명의 일 실시예에 따른 프로세싱 엘리먼트들의 연산 스킵을 설명하기 위해 예시한 도면이다.

도 8은 본 발명의 일 실시예에 따른 전치 합성곱 신경망의 가속 방법을 설명하기 위해 예시한 순서도다.

발명을 실시하기 위한 구체적인 내용

[0019] 이하, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다. 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 발명을 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.

[0020] 또한, 본 발명의 구성 요소를 설명하는 데 있어서, 제 1, 제 2, A, B, (a), (b) 등의 용어를 사용할 수 있다. 이러한 용어는 그 구성 요소를 다른 구성 요소와 구별하기 위한 것일 뿐, 그 용어에 의해 해당 구성 요소의 본질이나 차례 또는 순서 등이 한정되지 않는다. 명세서 전체에서, 어떤 부분이 어떤 구성요소를 '포함', '구비'한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를

더 포함할 수 있는 것을 의미한다. 또한, 명세서에 기재된 '~부', '모듈' 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0021] 이하에서 특징맵의 크기는 2 내지 16으로 설명하나, 이는 하나의 실시예에 불과하므로, 다양한 크기의 특징맵에도 동일한 가속 방법이 적용될 수 있다. 또한, 커널(kernel)과 필터(filter) 용어는 혼용될 수 있다.

[0022] 도 2는 TDC 방법을 이용하는 디컨볼루션 연산을 구체적으로 설명하기 위해 예시한 도면이다.

[0023] 도 2를 참조하면, 입력 블록(200), 복수의 제1 커널들(210, 212, 214, 216), 출력 특징맵(220), 제1 출력 블록(222), 복수의 제2 커널들(230, 232, 234, 236), 및 제2 출력 블록(242)이 도시되어 있다.

[0024] 입력 블록(200)은 입력 특징맵 중 일부 영역 또는 전부 영역을 의미한다.

[0025] 복수의 제1 커널들(210, 212, 214, 216)은 전치 합성곱 층의 디컨볼루션 커널들로부터 TDC 방법에 의해 변환된 컨볼루션 커널들이며, 각 커널마다 하나의 출력 픽셀을 생성하도록 학습된 커널들이다. S가 전치 합성곱 층의 디컨볼루션 커널의 스트라이드 크기를 의미할 때, 하나의 출력 블록을 생성하기 위해서는 S^2개의 커널들이 필요하다.

[0026] 출력 특징맵(220)은 입력 특징맵으로부터 디컨볼루션 연산된 결과이며, 제1 출력 블록(222)은 입력 블록(200)과 복수의 제1 커널들(210, 212, 214, 216)의 요소별 곱셈(element-wise multiplication) 결과다.

[0027] 구체적으로, TDC 기반 디컨볼루션 연산 방법은 하나의 출력 픽셀을 생성하기 위해 복수의 제1 커널들(210, 212, 214, 216) 중 하나와 입력 블록(200)을 이용한다. 예를 들면, 입력 블록(200)과 제1-1 커널(210)에 요소별 곱셈 연산을 수행함으로써, 제1 출력 블록(222) 중 8 값을 가지는 하나의 출력 픽셀을 만들 수 있다.

[0028] 이러한 과정을 나머지 제1 커널들에 적용함으로써 2*2 크기의 제1 출력 블록(222)이 생성된다. 또한, 입력 블록(200)과 복수의 제2 커널들(230, 232, 234, 236)의 요소별 곱셈을 통해 제2 출력 블록(242)을 생성할 수 있다. 이때, TDC 기반 디컨볼루션 연산 방법은 오버랩핑 섬 문제를 수반하지 않는다.

[0029] 하지만, TDC 변환에 의해 복수의 제1 커널들(210, 212, 214, 216)과 복수의 제2 커널들(230, 232, 234, 236)에는 수 많은 제로 웨이트들(zero weights)이 포함되어 있다. TDC 기반 디컨볼루션 연산 방법을 하드웨어 가속 장치에 구현하는 경우, 가속 장치는 제로 웨이트들에 대해서도 요소별 곱셈을 수행해야 하기 때문에 연산 속도가 느리고 에너지 소모도 높다.

[0030] 도 3은 디컨볼루션 커널 크기가 5일 때 TDC 기반 디컨볼루션 연산 방법을 설명하기 위해 예시한 도면이다.

[0031] 도 3을 참조하면, 디컨볼루션 커널들(300), 컨볼루션 커널들(310), 위노그래드 커널들(320), 입력 특징맵들(330), 위노그래드 특징맵들(340), 3번 커널들(350), 서브 출력 특징맵(360) 및 출력 특징맵(370)이 도시되어 있다.

[0032] 디컨볼루션 커널들(300)의 크기는 5이고, 총 M*N 개의 커널들로 구성될 때, TDC 방법에 의해 S^2 개의 컨볼루션 커널들(310)로 변환된다. 여기서, N은 입력 특징맵들(330)의 수를 의미하고, M은 최종 출력 특징맵들의 수를 의미하며, S는 디컨볼루션 커널들(300)의 스트라이드 크기를 의미한다. 즉, 스트라이드 크기를 제곱하면 컨볼루션 커널들(310)의 개수가 된다. S^2 개의 컨볼루션 커널들(310)은 하나의 입력 블록에 대해 하나의 출력 블록을 생성하기 위한 커널 단위다. 컨볼루션 커널들(310) 내 요소의 값은 출력 픽셀 또는 출력 블록 내 요소의 인덱스를 예시한 것이다.

[0033] 위노그래드 커널들(320)은 컨볼루션 커널들(310)을 공간 도메인 상에서 위노그래드 도메인 상으로 변환한 커널들이다. 또한, 위노그래드 특징맵들(340)은 입력 특징맵들(330)으로부터 위노그래드 도메인 상으로 변환된 특징맵들을 의미한다.

[0034] 위노그래드 알고리즘(Winograd Algorithm)은 행렬 곱셈을 요소별 곱셈으로 변환하여 곱셈 횟수를 줄이고, 덧셈 횟수를 증가시키는 알고리즘이다. 위노그래드 변환은 수학식 1 내지 수학식 2에 의해 구현될 수 있다.

수학식 1

[0035] $$Y = A^T[(Gf) \odot (B^T Z)]$$

수학식 2

[0036] $$B^T = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \quad G^T = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 \end{bmatrix}$$

[0037] 수학식 1 및 수학식 2에서 Y는 위노그래드 출력, B^T, G^T, A^T 은 변환 매트릭스, f는 커널 또는 필터, Z는 입력을 의미한다. 연산자 $\odot$는 요소별 곱셈 연산을 의미한다. Gf 는 공간 도메인에서 위노그래드 도메인상으로 변환된 커널이며, $B^T Z$ 는 위노그래드 도메인상으로 변환된 입력 특징맵을 의미한다. 위노그래드 도메인 상에서는 행렬 곱셈이 아닌 요소별 곱셈을 수행하므로 곱셈 수가 줄어들고 덧셈 수가 증가한다.

[0038] 입력 특징맵들(330)은 전치 합성곱 신경망이 입력 이미지로부터 추출한 특징맵이다. 입력 특징맵들(330) 중에서 하나의 특징맵 내에 일부 또는 전부 영역을 입력 블록이라 한다. 이하에서는, 하나의 특징맵이 입력 블록을 의미하는 것으로 설명한다.

[0039] 가속 장치는 B 매트릭스를 이용하여 입력 특징맵들(330)을 공간 도메인에서 위노그래드 도메인으로 변환하여 위노그래드 특징맵들(340)을 획득한다. 또한, 가속 장치는 G 매트릭스에 의해 변환된 위노그래드 커널들(320) 중 3번 커널들(350)을 추출한다.

[0040] 이후, 가속 장치는 위노그래드 특징맵들(340)과 3번 커널들(350) 간 요소별 곱셈을 통해 서브 출력 특징맵(360)을 생성한다. 가속 장치는 3번 인덱스를 가진 커널들에 대해 요소별 곱셈을 수행함으로써, 서브 출력 특징맵들을 생성할 수 있다.

[0041] 가속 장치는 서브 출력 특징맵(360)과 다른 서브 출력 특징맵들 간 채널별 덧셈(channel-wise summation)을 수행하고, 합산된 서브 출력 특징맵을 위노그래드 도메인으로부터 공간 도메인으로 A 매트릭스를 이용하여 변환하며, 활성화(activation) 함수를 통해 출력 특징맵(370)을 생성한다. 이때, 각 필터는 요소별 곱셈을 통해 S*S 크기의 출력 특징맵을 생성하며, 위노그래드 알고리즘을 통해 m*m 크기의 출력 블록을 생성하므로, 최종적으로 mS*mS 크기의 출력 특징맵(370)이 생성된다.

[0042] 한편, 위노그래드 커널들(320)은 고정된 위치에 제로 웨이트들(zero weights)을 포함한다. 다만, 제로 웨이트들의 수와 위치가 S 인덱스마다 다를 수 있다. 예를 들어, 1 번 인덱스 커널들은 첫 번째 열벡터가 모두 제로 웨이트로 구성된 제로 열벡터이지만, 2번 인덱스 커널들은 첫 번째 행벡터가 모두 제로 웨이트로 구성된 제로 행벡터다. 즉, 위노그래드 커널들(320)은 S 인덱스별로 고정된 위치에 제로 웨이트를 포함한다.

[0043] 요소별 곱셈 과정에서, 본 발명의 일 실시예에 따른 가속 장치는 3번 커널들(350)에서 값이 0인 제로 웨이트를 재배열하고, 위노그래드 특징맵들(340)을 3번 커널들(350)의 요소에 대응되도록 재배열하고, 재배열된 제로 웨이트들에 대한 요소별 곱셈 연산을 수행하지 않을 수 있다. 또한, 본 발명의 일 실시예에 따른 가속 장치는 제로 웨이트들에 대해 채널별 덧셈, 공간 도메인으로의 변환 및 활성화를 스킵할 수 있다.

[0044] 도 4는 디컨볼루션 커널 크기가 4일 때 TDC 기반 디컨볼루션 연산 과정을 예시한 도면이다.

[0045] 도 4는 도 3과 비교할 때, 디컨볼루션 커널들(400)의 크기, 컨볼루션 커널들(410)의 크기, 및 위노그래드 커널들(420)의 크기만 다를 뿐, 출력 특징맵(470)을 생성하는 과정은 도 3과 같다.

[0046] 도 5는 본 발명의 일 실시예에 따른 재배열 과정과 연산 과정을 설명하기 위해 예시한 도면이다.

[0047] 도 5를 참조하면, 입력 블록들(500), 위노그래드 커널들(510), 3번 커널들(520), 입력 행렬(502), 복수의 커널 행렬들(522, 524, 526), 연산부(530), 서브 출력 특징맵들(540), 출력 블록들(550)이 도시되어 있다.

[0048] 입력 행렬(502)과 복수의 커널 행렬들(522, 524, 526)는 위노그래드 도메인으로 변환된 것으로서, 가속 장치가 입력 행렬(502)과 복수의 커널 행렬들(522, 524, 526)에 대해 행렬 곱셈이 아닌 요소별 곱셈을 수행할 수 있으므로 연산량을 줄일 수 있다.

[0049] 가속 장치는 메모리로부터 입력 블록들(500)과 위노그래드 커널들(510)을 페치(fetch)한다. 이때, 위노그래드 커널들(510)은 서로 다른 비율로 제로 웨이트들을 포함하고 있다.

[0050] 가속 장치는 N개의 입력 블록들(500)을 하나의 입력 행렬(502)로 재배열한다. 구체적으로, 가속 장치는 입력 블록들(500) 중 첫 번째 입력 블록을 첫 번째 열벡터로 재배열한다. 마찬가지로, 가속 장치는 나머지 입력 블록들을 각각 순서에 따라 열벡터로 재배열한다. 가속 장치는 모든 열벡터를 순서대로 배열하여 하나의 입력 행렬(502)을 생성할 수 있다. 이 외에도, 가속 장치는 입력 블록들(500) 내 동일한 인덱스를 가지는 요소들을 하나의 행벡터로 재배열함으로써, 입력 행렬(502)을 생성할 수 있다. 즉, 첫 번째 행벡터는 입력 블록들(500)의 첫 번째 요소들로 구성되는 것이다.

[0051] 가속 장치는 위노그래드 커널들(510)을 커널 행렬로 재배열한다. 재배열 방식은 입력 행렬(502)을 생성하는 과정과 같다. 이하에서는 3번 커널들(520)에 대해 예시적으로 설명하나, 다른 커널들에 대해 동일한 과정이 적용될 수 있다.

[0052] 가속 장치는 3번 커널들(520)을 복수의 커널 행렬들(522, 524, 526)으로 재배열한다. 3번 커널들(520)의 수가 M*N일 때, 복수의 커널 행렬들(522, 524, 526)의 수는 M개이며, 각 커널 행렬의 열벡터는 N개의 요소를 가지며 하나의 커널을 의미한다. 입력 블록들(500)과 3번 커널들(520)의 크기가 n일 때, 입력 행렬(502) 및 복수의 커널 행렬들(522, 524, 526)의 크기는 n의 제곱이 된다.

[0053] 본 발명의 일 실시예에 따른 가속 장치는 입력 행렬(502) 및 복수의 커널 행렬들(522, 524, 526)의 요소를 재배열함으로써, 값이 모두 0인 제로 행벡터를 포함하도록 할 수 있다. 예를 들어, 가속 장치는 입력 행렬(502)과 복수의 커널 행렬들(522, 524, 526)에서 제로 웨이트들을 첫 번째 행벡터에 위치시킬 수 있다.

[0054] 본 발명의 일 실시예에 따른 가속 장치는 복수의 커널 행렬들(522, 524, 526)의 요소를 직접 재배열하거나, 외부 장치에 의해 재배열된 커널 행렬들(522, 524, 526)을 입력 받아 메모리부에 저장한 후 요소별 곱셈에 이용할 수도 있다.

[0055] 연산부(530)는 입력 행렬(502)과 복수의 커널 행렬들(522, 524, 526)에서 제로 행벡터에 대해서는 연산을 수행하지 않고, 제로 행벡터가 아닌 부분들에 대해서 요소별 곱셈을 수행함으로써, 서브 출력 특징맵들(540)을 연산할 수 있다. 입력 블록의 크기가 T_n이고, 출력 블록의 크기가 T_m일 때, 연산부(530)는 중간 결과물로서 $T_n \times T_m$ 크기의 데이터를 병렬적으로 연산할 수 있다.

[0056] 가속 장치는 서브 출력 특징맵들(540)에 대한 채널별 덧셈, 위노그래드 역변환 및 활성화를 통해 출력 블록들(550)을 출력할 수 있다.

[0057] 도 6은 본 발명의 일 실시예에 따른 가속 장치의 구성요소를 예시한 구성도다.

[0058] 도 6을 참조하면, 본 발명의 일 실시예에 따른 가속 장치(미도시)는 입력 버퍼(600), 전처리부(610), 연산부(620), 후처리부(630) 및 출력 버퍼(640)를 포함한다. 본 발명의 일 실시예에 따른 가속 장치는 재배열부(미도시) 또는 메모리부(미도시)를 더 포함할 수 있다.

[0059] 본 발명의 일 실시예에 따라 가속 장치는 TDC 변환된 입력 블록과 커널들을 이용할 수 있다. 즉, 이하에서 언급되는 커널들은 전치 합성곱 신경망 내 전치 합성곱 레이어의 디컨볼루션 커널들로부터 TDC 방법에 의해 컨볼루션 커널로 변환된 커널일 수 있다.

[0060] 입력 버퍼(600)는 전치 합성곱 신경망으로부터 특징맵들을 입력 받는 구성요소다. 입력 버퍼(600)는 T_n 크기의 입력 특징맵을 입력 받기 위해 n+m개의 라인을 포함한다.

[0061] 재배열부는 위노그래드 도메인 상에서 제로 웨이트들이 포함된 커널들의 요소를 재배열하는 구성요소다. 반면, 메모리부는 외부 장치에 의해 제로 행벡터를 포함하도록 재배열된 커널들을 저장하는 구성요소다. 본 발명의 일

실시예에 따른 가속 장치가 재배열부를 포함하는 경우 커널들의 요소를 직접 재배열할 수 있다. 본 발명의 일 실시예에 따른 가속 장치가 메모리부를 포함하는 경우, 외부 장치에 의해 제로 행벡터를 포함하도록 재배열된 커널들을 수신하여 저장한다. 메모리부에 저장된 재배열된 커널들은 연산부(620)에 의해 호출된다. 이하에서, 메모리부에 저장되는 커널들은 재배열부가 여러 방법을 이용하여 재배열한 커널들과 같은 것으로 설명한다.

[0062] 전처리부(610)는 각 특징맵들 내에서 대응되는 입력 블록들을 선택하고 입력 블록들을 공간 도메인에서 위노그래드 도메인으로 변환하며, 위노그래드 도메인 상에서 재배열된 커널들의 요소에 따라 입력 블록들의 요소를 재배열하는 구성요소다. 여기서, 입력 블록들은 크기가 TDC 컨볼루션 커널의 크기와 같고, 특징맵들 내 동일한 위치의 블록들이며, 가속 장치의 연산 처리량 감소를 위해 위노그래드 도메인으로 변환된다.

[0063] 본 발명의 일 실시예에 따라 재배열부가 커널들이 하나 이상의 제로 행벡터를 포함하도록 커널들의 요소를 재배열하고, 전처리부(610)가 커널들의 재배열된 요소에 대응되도록 입력 블록들의 요소를 재배열할 수 있다. 예를 들어, 재배열부는 커널들의 제로 웨이트들을 상위 행벡터로 재배열함으로써, 커널 행렬들이 하나 이상의 행벡터를 포함하도록 한다. 이때, 전처리부(610)는 커널의 제로 웨이트에 대응되는 입력 블록의 요소들을 상위 행벡터로 재배열함으로써, 입력 행렬이 하나 이상의 행벡터를 포함하도록 한다.

[0064] 본 발명의 다른 실시예에 따라 재배열부가 도 5에서 설명한 바와 같이 커널들을 각 행렬이 하나 이상의 제로 행벡터를 포함하는 복수의 커널 행렬들로 병합하고, 전처리부(610)가 입력 블록들을 하나의 입력 행렬로 변환할 수 있다.

[0065] 연산부(620)는 재배열된 입력 블록들과 재배열된 커널들에 대해 요소별 곱셈 연산을 수행함으로써 출력 블록들을 생성하는 구성요소다. 여기서, 재배열된 커널들은 재배열부에 의해 재배열되거나 메모리부에 저장되어 있는 커널들이다.

[0066] 본 발명의 일 실시예에 따른 재배열부와 전처리부(610)가 제로 행벡터를 포함하도록 커널들의 요소와 입력 블록들의 요소를 재배열한 경우, 연산부(620)는 재배열된 커널들에서 제로 행벡터를 제외한 벡터들에 대해서만 입력 블록들과의 요소별 곱셈을 수행한다.

[0067] 또한, 본 발명의 다른 실시예에 따른 재배열부와 전처리부(610)가 제로 행벡터를 포함하는 복수의 커널 행렬들을 생성한 경우, 연산부(620)는 복수의 커널 행렬들 내에서 제로 행벡터를 제외한 행벡터들에 대해 입력 행렬과의 요소별 곱셈을 수행함으로써 출력 행렬을 생성할 수 있다.

[0068] 한편, 본 발명의 다른 실시예에 따른 연산부(620)는 복수의 프로세싱 엘리먼트(processing elements, Pes)를 포함하고, 하나 이상의 제로 행벡터에 해당하는 프로세싱 엘리먼트들에 대해 연산을 스킵(skip)할 수 있다.

[0069] 후처리부(630)는 출력 블록들을 위노그래드 도메인에서 공간 도메인으로 변환하는 구성요소다. 본 발명의 일 실시예에 따른 연산부(620)가 출력 행렬을 생성한 경우, 후처리부(630)는 도메인 변환 전에 출력 행렬을 출력 블록으로 먼저 변환한다.

[0070] 출력 버퍼(640)는 변환된 출력 블록들에 기초하여 출력 특징맵을 생성하여 출력하는 구성요소다. 출력 버퍼(640)는 T_m 크기의 출력 특징맵을 출력하기 위해 2*mS개의 라인을 포함한다.

[0071] 본 발명의 일 실시예에 따라 입력 버퍼(600)와 출력 버퍼(640)는 전처리부(610), 연산부(620) 및 후처리부(630)와 병렬적으로 동작할 수 있다. 즉, 가속 장치는 데이터 전송 시간과 데이터 연산 시간을 중첩시킬 수 있다. 이를 위해서 입력 블록의 크기와 출력 블록의 크기 등에 따른 데이터 전송 시간 및 데이터 연산 시간을 결정해야 한다. 이는 수학식 3 내지 수학식 5와 같이 표현될 수 있다.

수학식 3

[0072]
$$T_C = \left\lceil \frac{S^2 M}{T_m} \right\rceil \times \left\lceil \frac{N}{T_n} \right\rceil \times \left\lceil \frac{W_I}{m} \right\rceil \times \frac{C(K_C)}{m^2} \times \frac{1}{freq}$$

수학식 4

[0073] $$T_D = \frac{mS \times W_I \times S^2 M \times n^2}{Bandwidth}$$

수학식 5

[0074] $$Bandwidth = \frac{m^2}{C(K_C)} \times \left\lceil \frac{T_m \times T_n}{N} \right\rceil \times mS \times n^2 \times freq$$

[0075] 수학식 3 내지 수학식 5에서 T_C는 데이터 연산 시간, T_D는 데이터 전송 시간, S는 디컨볼루션 커널의 스트라이드 크기, N은 입력 블록의 수, M은 출력 블록의 수, T_n은 입력 행렬의 요소의 수, T_m은 커널 행렬의 요소의 수, m은 출력 행렬의 크기, W_I는 입력 블록의 가로 크기, K_C는 커널 크기다. bandwidth는 온 칩 버퍼(가속 장치)와 오프 칩 메모리 간 입출력 블록의 크기를 정하기 위한 대역폭을 의미한다.

[0076] 이때, 본 발명의 실시에에 따른 데이터 전송 시간은 데이터 연산 시간보다 같거나 작도록 설정된다.

[0077] 도 7은 본 발명의 일 실시예에 따른 프로세싱 엘리먼트들의 연산 스킵을 설명하기 위해 예시한 도면이다.

[0078] 도 7을 참조하면, 커널에 포함된 제로 웨이트 들의 수에 따라 가속 장치가 스킵할 수 있는 연산량이 도시되어 있다.

[0079] 케이스 1의 경우, 커널이 제로 웨이트를 포함하지 않기 때문에 가속 장치의 모든 프로세싱 엘리먼트에서 연산을 수행해야 한다.

[0080] 케이스 2의 경우, 커널이 제로 웨이트를 소수 포함하며, 본 발명의 실시예에 따라 제로 행벡터를 포함하도록 재배열 됨으로써, 가속 장치는 첫 번째 제로 행벡터에 대해 연산을 생략할 수 있다.

[0081] 케이스 3의 경우, 커널이 제로 웨이트를 다수 포함하며, 본 발명의 실시예에 따라 제로 행벡터를 포함하도록 재배열 됨으로써, 가속 장치는 첫 번째 행벡터 및 두 번째 행벡터에 대해 연산을 생략할 수 있다.

[0082] 프로세싱 엘리먼트들이 각 행벡터 순으로 연산을 수행하기 때문에, 가속 장치가 제로 웨이트를 재배열하지 않으면 행벡터에 제로 웨이트가 하나라도 포함된 경우 제로 웨이트에 대한 대기 작업을 수행해야 한다. 이때, 제로 웨이트에 대한 연산은 휴면 상태로 기 설정된 시간만큼 대기하는 것을 의미한다.

[0083] 하지만, 본 발명의 일 실시예에 따른 가속 장치는 커널이 제로 행벡터를 포함하도록 커널을 재배열하고, 행벡터 단위로 연산을 스킵하므로 연산 처리량을 줄이고 연산 속도를 증가시킬 수 있다.

[0084] 도 8은 본 발명의 일 실시예에 따른 전치 합성곱 신경망의 가속 방법을 설명하기 위해 예시한 순서도다.

[0085] 가속 장치는 전치 합성곱 신경망으로부터 특징맵들을 수신한다(S800).

[0086] 가속 장치는 각 특징맵들 내에서 대응되는 입력 블록들을 선택하고 위노그래드 도메인으로 변환한다(S802). 본 발명의 실시예에 따른 위노그래드 변환은 가속 장치의 곱셈 횟수를 줄이고 덧셈 횟수를 증가시킴으로써 가속 장치의 총 연산량을 감소시킨다.

[0087] 가속 장치는 위노그래드 도메인 상에서 입력 블록들의 요소와 제로 웨이트들이 포함된 커널들의 요소를 재배열한다(S804). 본 발명의 일 실시예에 따라 가속 장치는 커널들이 하나 이상의 제로 행벡터를 포함하도록 커널들의 요소를 재배열하고, 커널들의 재배열된 요소에 대응되도록 입력 블록들의 요소를 재배열할 수 있다.

[0088] 가속 장치는 재배열된 입력 블록들과 재배열된 커널들에 대해 요소별 곱셈 연산을 수행함으로써 출력 블록들을 생성한다(S806). 본 발명의 일 실시예에 따른 가속 장치는 재배열된 커널들에서 제로 행벡터를 제외한 벡터들에

대해서만 요소별 곱셈을 수행함으로써 출력 블록들을 생성할 수 있다.

[0089] 가속 장치는 출력 블록들을 위노그래드 도메인에서 공간 도메인으로 변환한다(S808).

[0090] 가속 장치는 변환된 출력 블록들에 기초하여 출력 특징맵을 생성하여 출력한다(S810). 한편, 본 발명의 일 실시예에 따른 가속 장치는 입력 블록들로부터 출력 블록들이 생성되는 도중에 특징맵들 내에서 다른 입력 블록들을 입력 받는 과정을 병렬적으로 수행할 수 있다.

[0091] 도 8에서는 과정 S800 내지 과정 S810을 순차적으로 실행하는 것으로 기재하고 있으나, 이는 본 발명의 일 실시예의 기술 사상을 예시적으로 설명한 것에 불과한 것이다. 다시 말해, 본 발명의 일 실시예가 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 일 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 도 8에 기재된 순서를 변경하여 실행하거나 과정 S800 내지 과정 S810 중 하나 이상의 과정을 병렬적으로 실행하는 것으로 다양하게 수정 및 변형하여 적용 가능할 것이므로, 도 8은 시계열적인 순서로 한정되는 것은 아니다.

[0092] 한편, 도 8에 도시된 과정들은 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 즉, 이러한 컴퓨터가 읽을 수 있는 기록매체는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등의 비일시적인(non-transitory) 매체일 수 있으며, 또한 캐리어 웨이브(예를 들어, 인터넷을 통한 전송) 및 데이터 전송 매체(data transmission medium)와 같은 일시적인(transitory) 매체를 더 포함할 수도 있다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

[0093] 이상의 설명은 본 실시예의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 실시예가 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 실시예들은 본 실시예의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시예에 의하여 본 실시예의 기술 사상의 범위가 한정되는 것은 아니다. 본 실시예의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 실시예의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

부호의 설명

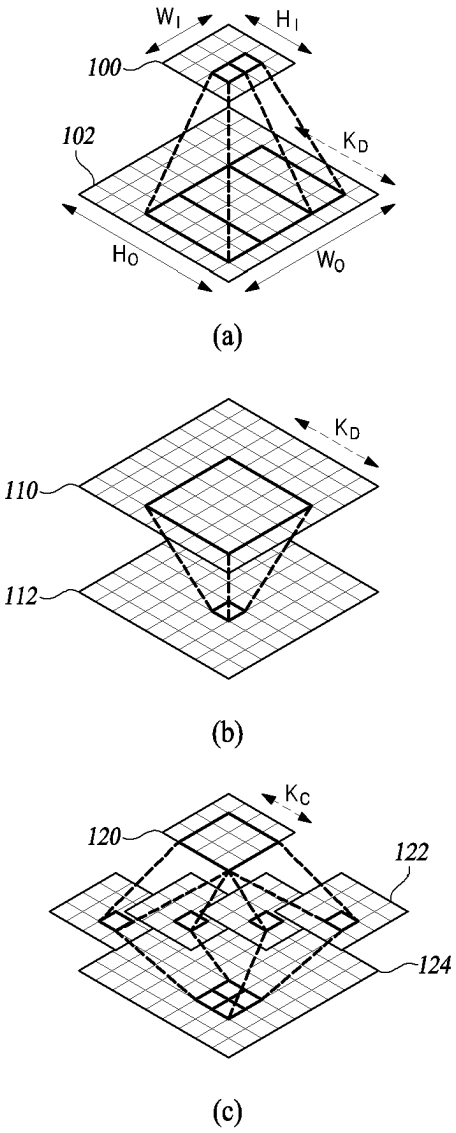
[0094] 600: 입력 버퍼 610: 전처리부

620: 연산부 630: 후처리부

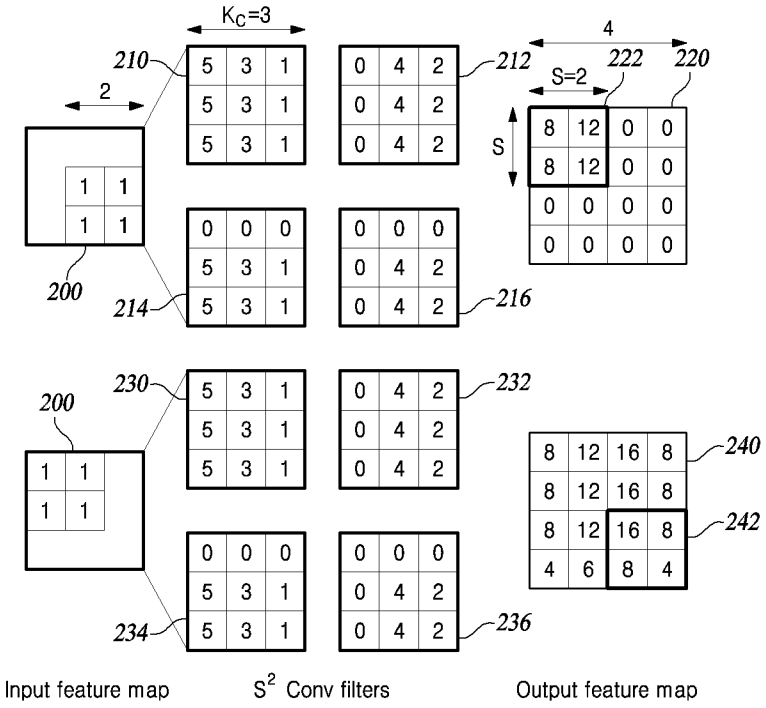
640: 출력 버퍼

도면

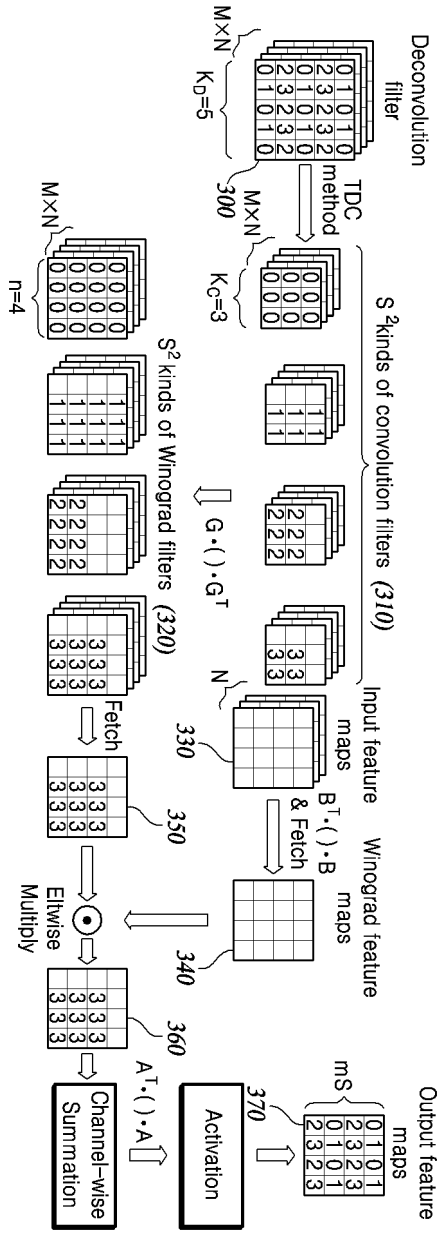
도면1



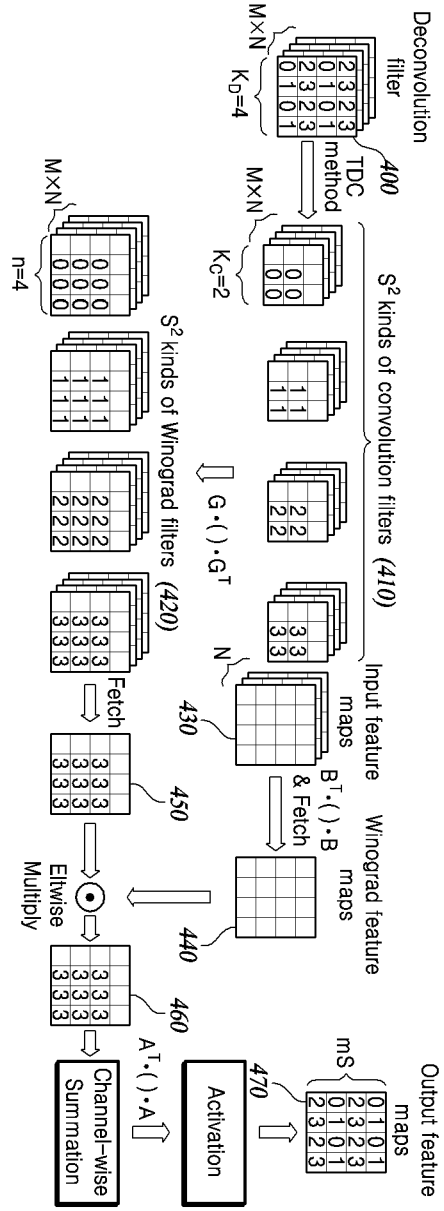
도면2


Kc=3
210
2
200
214
212
216
4
S=2
222
220
S
230
232
234
236
240
242
Input feature map
S2 Conv filters
Output feature map

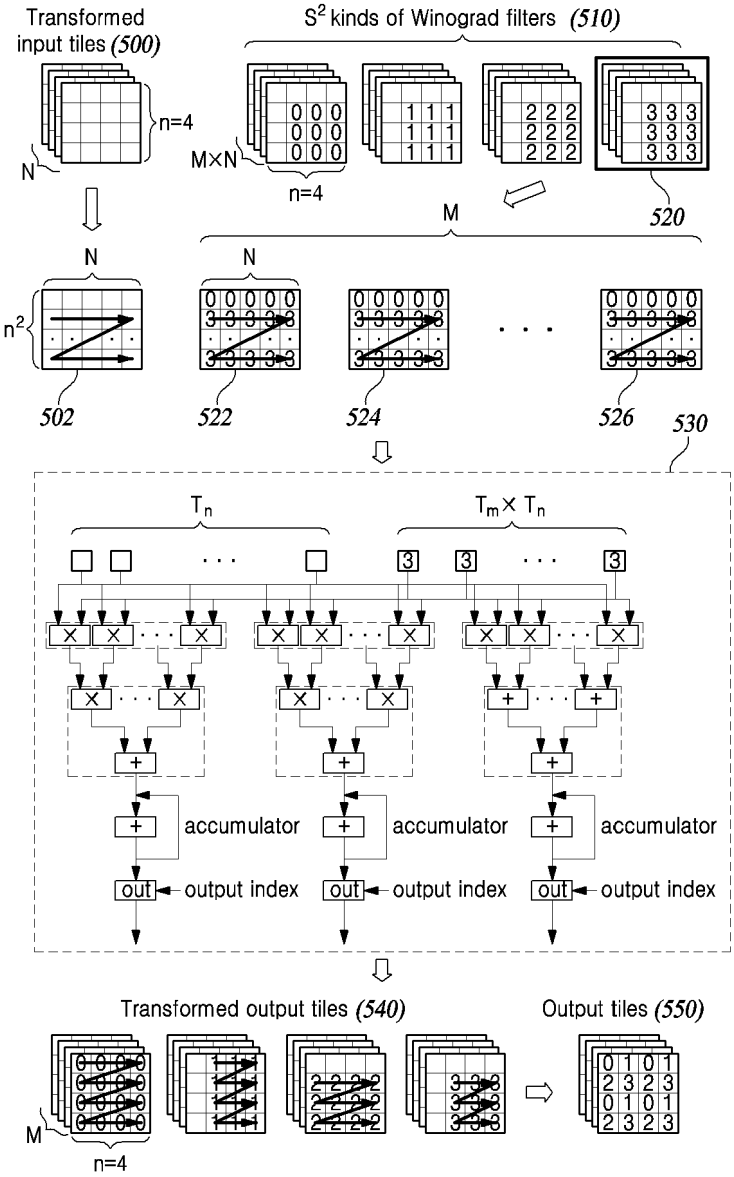
도면3


Deconvolution filter
M×N
K_D=5
TDC method
300
M×N
K_C=3
S^2kinds of convolution filters (310)
M×N
n=4
S^2 kinds of Winograd filters (320)
G•()•G^T
N
330
Input feature maps
B^T•()•B & Fetch
Winograd feature maps
Fetch
350
Eltwise Multiply
340
360
A^T•()•A
Channel-wise Summation
mS
Output feature maps
370
Activation

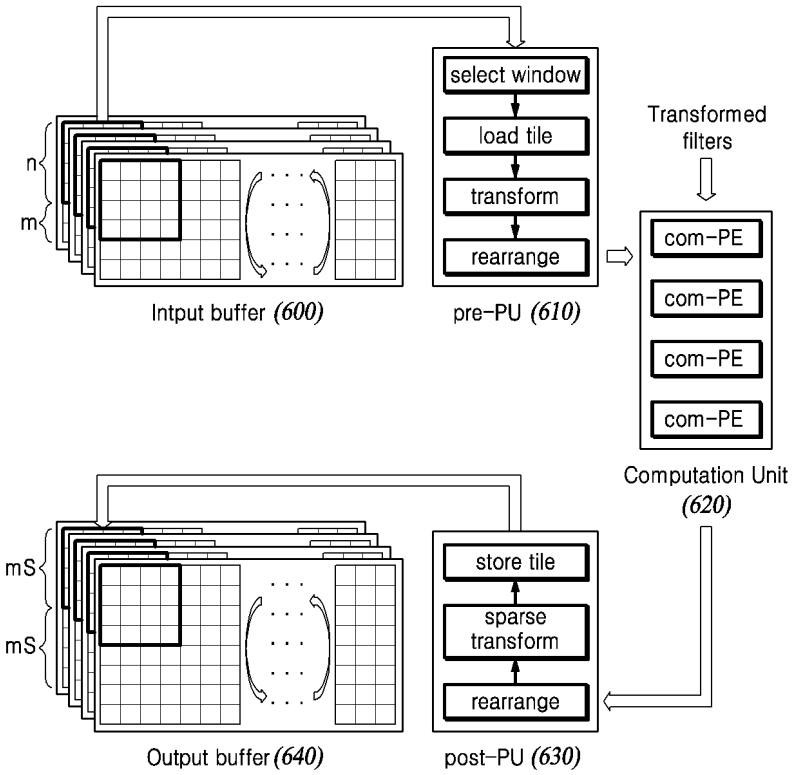
도면4


Deconvolution filter
M×N
K_D=4
TDC method
400
M×N
K_C=2
S² kinds of convolution filters (410)
G·( )·G^T
S² kinds of Winograd filters (420)
M×N
n=4
N
Input feature maps
430
B^T·( )·B & Fetch
Winograd feature maps
440
Fetch
450
Eltwise Multiply
460
A^T·( )·A
Channel-wise Summation
mS
Activation
470
Output feature maps

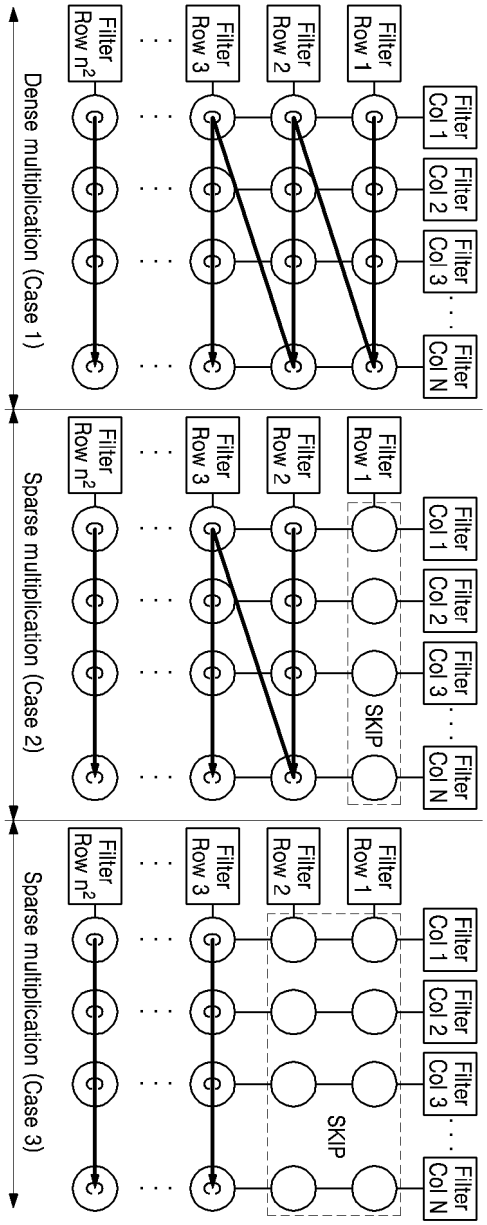
도면5


Transformed input tiles (500)
S2 kinds of Winograd filters (510)
n=4
N
M×N
n=4
M
520
N
N
n2
502
522
524
526
530
Tn
Tm× Tn
accumulator
accumulator
accumulator
out
output index
out
output index
out
output index
Transformed output tiles (540)
Output tiles (550)
M
n=4

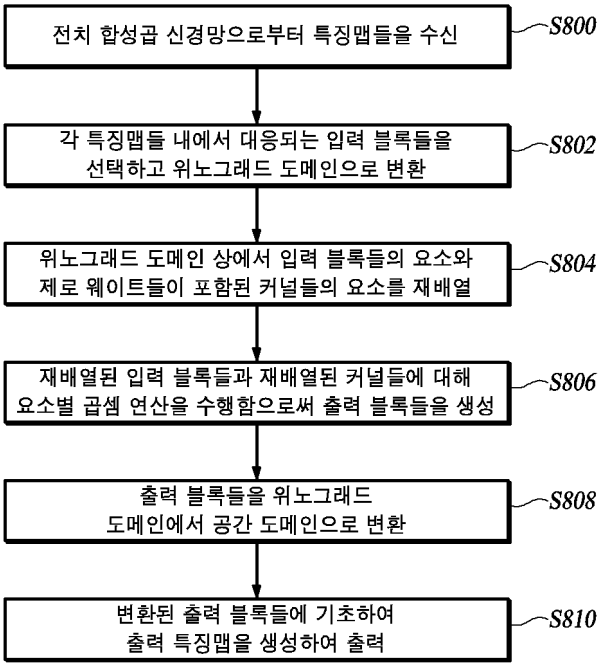
도면6


n
m
Intput buffer (600)
select window
load tile
transform
rearrange
pre-PU (610)
Transformed filters
com-PE
com-PE
com-PE
com-PE
Computation Unit (620)
mS
mS
Output buffer (640)
store tile
sparse transform
rearrange
post-PU (630)

도면7


Filter Row 1
Filter Row 2
Filter Row 3
Filter Row n²
Filter Col 1
Filter Col 2
Filter Col 3
Filter Col N
SKIP
Dense multiplication (Case 1)
Sparse multiplication (Case 2)
Sparse multiplication (Case 3)

도면8


전치 합성곱 신경망으로부터 특징맵들을 수신
S800
각 특징맵들 내에서 대응되는 입력 블록들을 선택하고 위노그래드 도메인으로 변환
S802
위노그래드 도메인 상에서 입력 블록들의 요소와 제로 웨이트들이 포함된 커널들의 요소를 재배열
S804
재배열된 입력 블록들과 재배열된 커널들에 대해 요소별 곱셈 연산을 수행함으로써 출력 블록들을 생성
S806
출력 블록들을 위노그래드 도메인에서 공간 도메인으로 변환
S808
변환된 출력 블록들에 기초하여 출력 특징맵을 생성하여 출력
S810