(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2024 년 11 월 21 일 (21.11.2024)

WIPO | PCT

(10) 국제공개번호
WO 2024/237652 A1

(51) 국제특허분류:
G06N 3/0464 (2023.01) *G06N 20/10* (2019.01)
G06N 3/0495 (2023.01)

(21) 국제출원번호: PCT/KR2024/006514

(22) 국제출원일: 2024 년 5 월 13 일 (13.05.2024)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
10-2023-0061814 2023 년 5 월 12 일 (12.05.2023) KR

(71) 출원인: 한양대학교 산학협력단 (**IUCF-HYU (INDUSTRY-UNIVERSITY COOPERATION FOUNDATION HANYANG UNIVERSITY)**) [KR/KR]; 04763 서울특별시 성동구 왕십리로 222, 한양대학교내, Seoul (KR).

(72) 발명자: 최정욱 (**CHOI, Jung Wook**); 04763 서울특별시 성동구 왕십리로 222, 한양대학교 공업센터 702-1호, Seoul (KR). 윤민용 (**YOON, Min Yong**); 04763 서울특별시 성동구 왕십리로 222, 한양대학교 공업센터 702-1호, Seoul (KR). 이민재 (**LEE, Min Jae**); 04763 서울특별시 성동구 왕십리로 222, 한양대학교 공업센터 702-1호, Seoul (KR).

(74) 대리인: 해움특허법인 (**HAEUM PATENT & LAW FIRM**); 06536 서울특별시 서초구 강남대로 525, 8층, Seoul (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

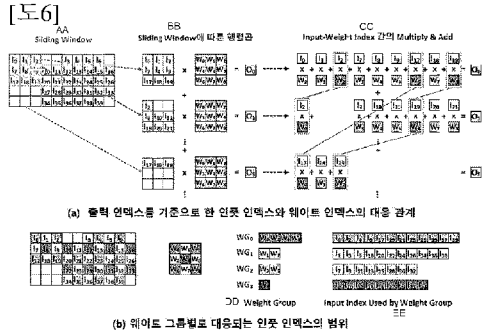
(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

공개:
— 국제조사보고서와 함께 (조약 제21조(3))

(54) **Title:** ARTIFICIAL NEURAL NETWORK-BASED CONVOLUTION OPERATION METHOD AND APPARATUS USING DATA FLOW OPTIMIZATION METHOD

(54) **발명의 명칭:** 데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법 및 장치

[도6]



(a) ... Correspondence relat on between nput index and weight ndex on basis of output index
(b) ... Range of input index corresponding to each we ght group
AA ... Sliding Window
BB ... Matrix product according to sliding window
CC ... Multiply & add input-weight indexes
DD ... Weight Group
EE ... Input Index Used by Weight Group

(57) **Abstract:** An artificial neural network-based convolution operation method using a data flow optimization method according to an embodiment may include: an index data extraction step of extracting input index data for sparse input data included in input data; a kernel weight group generation step of classifying weights of a kernel according to a preset criterion on the basis of the size of the kernel and the input index data and then generating a kernel weight group on the basis of classified information; a data flow rule generation step of generating a data flow rule by mapping corresponding input index data to the kernel weight group; and a convolution operation step of performing a convolution operation on the basis of the data flow rule.

(57) **요약서:** 일 실시예에 따른 데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법은, 입력 데이터에 포함되어 있는 희소 입력 데이터에 대한 입력 인덱스 데이터를 추출하는 인덱스 데이터 추출 단계, 커널의 크기 및 상기 입력 인덱스 데이터에 기초하여 상기 커널의 웨이트들을 미리 설정된 기준에 따라 분류한 후, 분류된 정보에 기초하여 커널 웨이트 그룹을 생성하는 커널 웨이트 그룹 생성 단계, 상기 커널 웨이트 그룹에 대응되는 입력 인덱스 데이터를 매핑하여 데이터 플로우 규칙을 생성하는 데이터 플로우 규칙 생성 단계 및 상기 데이터 플로우 규칙에 기초하여 컨볼루션 연산을 수행하는 컨볼루션 연산 단계를 포함할 수 있다.

WO 2024/237652 A1

발명의 설명

발명의 명칭: 데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법 및 장치

기술분야

[1] 본 발명은 데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법 및 장치에 관한 발명으로서, 보다 상세하게는 인공신경망을 이용하여 컨볼루션과 연산을 수행할 때, 데이터를 재사용할 수 있도록 데이터 플로우를 최적화하여 컨볼루션 연산을 보다 빠르고 효과적으로 수행하는 기술에 관한 발명이다.

배경기술

[2] 인공 지능(Artificial Intelligence, AI) 기술은, 인간의 학습능력과 추론능력, 지각능력, 자연언어의 이해 능력 등을 컴퓨터 프로그램으로 실현한 기술을 의미하며, 종래의 룰(Rule) 기반 스마트 시스템과 달리 기계가 스스로 학습하고 판단하며 똑똑해지는 시스템을 의미한다.

[3] 인공지능 기술은 기계학습(딥 러닝) 및 기계 학습을 활용한 요소 기술들로 구성된다. 기계 학습은 입력 데이터들의 특징을 스스로 분류/학습하는 알고리즘 기술이며, 요소 기술은 딥 러닝 등의 기계학습 알고리즘을 활용하여 인간 두뇌의 인지, 판단 등의 기능을 모사하는 기술로서, 언어적 이해, 시각적 이해, 추론/예측, 지식 표현, 동작 제어 등의 기술 분야로 구성된다.

[4] 인공지능의 기술의 발달에 따라, 자율주행 분야에서도 주행중인 차량 주변에 존재하는 객체를 인식하는 분야에도 인공지능 기술이 적용되고 있다. 구체적으로, Lidar/RGB-D 센서 기반의 객체 인식 방법이 주를 이루고 있는데, Lidar/RGB-D 센서 기반의 객체 인식 방법은 점 구름(Point Cloud) 형태의 데이터를 활용해 차량 주변에 존재하는 객체의 위치와 종류를 분별하며, 점 구름 데이터는 컨볼루션(Convolution, 합성곱) 연산을 여러 레이어에 걸쳐 대해 반복적으로 수행함으로써, 해당 데이터의 객체를 분류할 수 있는 특징을 추출한다.

[5] 그러나, 객체 인식 방법의 특성상, 점 구름 데이터가 공간에 희박하게(Sparse) 존재하기 때문에 컨볼루션 연산 또한 희박하게 존재하는 데이터들에 대해 이루어진다. 따라서, 레이어 별로 추출하는 특징(Feature)이 불규칙적으로 메모리에 저장되고, 이로 인해 컨볼루션 연산에 사용되는 특징 데이터를 메모리로부터 불러오기 위해 불규칙적인 접근을 할 수 밖에 없다. 따라서, 종래 기술에 따라 컨볼루션 연산을 수행하는 경우, 전 과정을 마치기 위해 필요한 시간이 크게 증가하는 문제점이 존재한다.

[6] 또한, 컨볼루션 연산에 있어서, 컨볼루션 연산을 주어진 하드웨어의 특성과 버퍼 크기에 맞도록 타일링(연산을 시공간적으로 분할)하여 최적의 데이터 플로우(맵핑)를 생성하는 과정은 매우 중요한 부분을 차지한다.

[7] 종래의 기술들은 타일 크기에 따라 입력-커널-출력 크기와 연산이 규칙적인 밀집 데이터의 합성곱만을 타겟으로 하여, 입력-커널-출력의 크기와 연산이 불규칙한 희소 데이터의 희소 합성곱에 대한 최적화에 어려움이 있다

[8] 구체적으로, 종래 기술에 따라 Gather-GEMM-Scatter의 과정으로 수행되는 희소 데이터 기반의 컨볼루션 연산 방법은, 입력 데이터에서 0이 아닌 입력 데이터만 버퍼에 Gather하여 해당하는 커널과 연산을 수행하여 출력된 출력 데이터를 메모리에 Scatter 하므로, 입력 데이터의 배치에 따라 인덱스 별 재사용 횟수 및 번짐 효과로 인한 출력 데이터의 개수가 불규칙하게 나타나는 문제점이 있다. 따라서, 입력-커널-출력 데이터의 크기와 연산 횟수의 불규칙성으로 인하여, 타일링에 따른 입력/출력 버퍼의 사용률과, 각 버퍼에서의 데이터 재사용을 평가하는 것이 불가능한 단점이 존재한다.

발명의 내용

기술적 과제

[9] 일 실시예에 따른 데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법 및 장치는 상기 설명한 문제점을 설명하기 위한 고안된 발명으로서, 희소 입력 데이터의 특성을 기초하여 컨볼루션 연산을 효율적으로 할 수 있는 방법 및 장치를 제공하는데 그 목적이 있다.

[10] 보다 상세하게는 인공신경망을 이용하여 컨볼루션 연산을 수행함에 있어서, 희소 입력 데이터의 위치 및 크기 정보를 이용하여 버퍼의 크기를 적절하게 가변함으로써, 불필요한 데이터의 수집을 최소화 하는 방법으로 컨볼루션 연산의 전체 속도를 효과적으로 높이는데 목적이 존재한다.

과제 해결 수단

[11] 일 실시예 따른 데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법은, 입력 데이터에 포함되어 있는 희소 입력 데이터에 대한 입력 인덱스 데이터를 추출하는 인덱스 데이터 추출 단계, 커널의 크기 및 상기 입력 인덱스 데이터에 기초하여 상기 커널의 웨이트들을 미리 설정된 기준에 따라 분류한 후, 분류된 정보에 기초하여 커널 웨이트 그룹을 생성하는 커널 웨이트 그룹 생성 단계, 상기 커널 웨이트 그룹에 대응되는 입력 인덱스 데이터를 매핑하여 데이터 플로우 규칙을 생성하는 데이터 플로우 규칙 생성 단계 및 상기 데이터 플로우 규칙에 기초하여 컨볼루션 연산을 수행하는 컨볼루션 연산 단계를 포함할 수 있다.

[12] 상기 커널은, 3X3 커널을 포함하고, 상기 컨볼루션 연산은 스탠다드 컨볼루션(standard convolution) 또는 스트라이드 컨볼루션(stride convolution) 연산을 포함할 수 있다.

[13] 상기 커널 웨이트 그룹 생성 단계는, 상기 웨이트 그룹은 3X3 커널에 기초한 컨볼루션 연산의 특성을 고려하여, 대칭적으로 복수 개의 웨이트 그룹을 생성하는 단계를 포함할 수 있다.

[14] 상기 커널 웨이트 그룹 생성 단계는, 제0웨이트 인덱스(W0), 제2웨이트 인덱스(W2), 제6웨이트 인덱스(W6) 및 제8웨이트 인덱스(W8)를 제0웨이트 그룹으로 생성하는 단계를 포함할 수 있다.

[15] 상기 커널 웨이트 그룹 생성 단계는, 제1웨이트 인덱스(W1) 및 제7웨이트 인덱스(W7)를 제1웨이트 그룹으로 생성하는 단계를 포함할 수 있다.

[16] 상기 커널 웨이트 그룹 생성 단계는, 제3웨이트 인덱스(W3) 및 제5웨이트 인덱스(W5)를 제2웨이트 그룹으로 생성하는 단계를 포함할 수 있다.

[17] 상기 커널 웨이트 그룹 생성 단계는, 제4웨이트 인덱스(W4)를 제3웨이트 그룹으로 생성하는 단계를 포함할 수 있다.

[18] 상기 데이터 플로우 규칙 생성 단계는, 상기 웨이트 그룹에 대응되는 입력 인덱스를 수집한 후, 수집된 입력 인덱스를 상기 인풋 버퍼에 상기 웨이트 그룹별로 순차적으로 저장하는 단계를 포함할 수 있다.

[19] 상기 컨볼루션 연산은 디컨볼루션(decovolution)을 포함하며, 상기 커널 웨이트 그룹 생성 단계는, 입력 버퍼 및 출력 버퍼의 사이즈에 기초하여 웨이트 그룹의 크기를 결정할 수 있다.

[20] 상기 데이터 플로우 규칙 생성 단계는, 상기 입력 인덱스와 상기 입력 인덱스에 대응되는 출력 인덱스의 비율에 기초하여 상기 입력 데이터에 대한 타일링(tiling) 사이즈를 조절하는 타일링 사이즈 조절 단계를 포함할 수 있다.

[21] 상기 타일링 사이즈 조절 단계는, 스탠다드 컨볼루션(standard convolution) 또는 스트라이드 컨볼루션(stride convolution)의 경우 입력 채널에 대한 타일링 사이즈를 증가시키고, 디컨볼루션(decovolution)의 경우 출력 채널에 대한 타일링 사이즈를 감소시키는 단계를 포함할 수 있다.

[22] 일 실시예 따른 데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 장치는, 입력 데이터에 포함되어 있는 희소 입력 데이터에 대한 입력 인덱스 데이터를 수집하는 인덱스 데이터 수집 모듈 ,커널의 크기 및 상기 입력 인덱스 데이터에 기초하여 상기 커널의 웨이트들을 미리 설정된 기준에 따라 분류한 후, 분류된 정보에 기초하여 커널 웨이트 그룹을 생성하고, 상기 커널 웨이트 그룹에 대응되는 입력 인덱스 데이터를 매핑하여 데이터 플로우 규칙을 생성하는 데이터 플로우 규칙 생성 모듈 및 상기 데이터 플로우 규칙에 기초하여 컨볼루션 연산을 수행하는 컨볼루션 연산 모듈을 포함할 수 있다.

[23] 상기 커널은, 3X3 커널을 포함하고, 상기 컨볼루션 연산은 스탠다드 컨볼루션(standard convolution) 또는 스트라이드 컨볼루션(stride convolution) 연산을 포함할 수 있다.

[24] 상기 데이터 플로우 규칙 생성 모듈은, 상기 웨이트 그룹은 3X3 커널에 기초한 컨볼루션 연산의 특성을 고려하여, 대칭적으로 복수 개의 웨이트 그룹을 생성할 수 있다.

[25] 상기 데이터 플로우 규칙 생성 모듈은, 상기 웨이트 그룹에 대응되는 입력 인덱스를 수집한 후, 수집된 입력 인덱스를 인풋 버퍼에 상기 웨이트 그룹 별로 순차적으로 저장할 수 있다.

[26] 상기 컨볼루션 연산은 디컨볼루션(decovolution)을 포함하며, 상기 데이터 플로우 규칙 생성 모듈은, 입력 버퍼 및 출력 버퍼의 사이즈에 기초하여 웨이트 그룹의 크기를 결정할 수 있다.

[27] 일 실시예 따른 데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 장치는, 입력 데이터에 포함되어 있는 희소 입력 데이터에 대한 입력 인덱스 데이터를 수집하는 인덱스 데이터 수집 모듈, 커널의 크기 및 상기 입력 인덱스 데이터에 기초하여 상기 커널의 웨이트들을 미리 설정된 기준에 따라 분류한 후, 분류된 정보에 기초하여 커널 웨이트 그룹을 생성하고, 상기 커널 웨이트 그룹에 대응되는 입력 인덱스 데이터를 매핑하여 데이터 플로우 규칙을 생성하는 데이터 플로우 규칙 생성 모듈 및 상기 데이터 플로우 규칙에 기초하여 컨볼루션 연산을 수행하는 컨볼루션 연산 모듈을 포함하며,상기 데이터 플로우 규칙 생성 모듈은, 스탠다드 컨볼루션(standard convolution) 또는 스트라이드 컨볼루션(stride convolution)의 경우 입력 채널에 대한 타일링 사이즈를 증가시키고, 디컨볼루션(decovolution)의 경우 출력 채널에 대한 타일링 사이즈를 감소시킬 수 있다.

발명의 효과

[28] 일 실시예에 따른 희소 입력 데이터의 위치 정보에 기초한 컨볼루션 연산 방법 및 장치는 희소 데이터 기반의 컨볼루션 연산을 수행함에 있어서, 입력 데이터와 출력 데이터의 특징을 고려하여 데이터를 수집하고 재활용하므로, 종래 기술보다 빠르게 컨볼루션 연산을 수행할 수 있는 장점이 존재한다.

[29] 또한, 컨불루션 연산의 경우 그 특성상 중첩되는 계산 과정이 많은데, 본 발명에 따른 컨볼루션 연산 방법은 종래에 수집하였던 입력 데이터의 정보를 다음 연산에서 재사용하므로, 컨볼루션 연산 자체의 연산 속도를 빠르게 증가시킬 수 있는 장점이 존재한다.

[30] 또한, 이러한 특징으로 인해 3차원 공간에 존재하는 객체 인식의 속도를 증가시킬 수 있는바, 고차원 자율 주행에 필수적인 고속 전방 장애물 인식을 효율적으로 수행할 수 있고, 빠르고 정확한 로봇 네비게이션을 위한 RGB-D 기반의 위치 추정도 효율적으로 수행할 수 있는 장점이 존재한다.

[31] 본 발명의 효과들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재들로부터 당업자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[32] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 간단한 설명이 제공된다.

[33] 도 1은 본 발명의 일 실시예에 따라 컨볼루션 연산 장치에 입력되는 입력 데이터의 일 예와 컨볼루션 연산을 수행하는 커널의 일 예를 도시한 도면이다.

[34] 도 2는 종래 기술에 따라 희소 데이터 기반의 컨볼루션 연산을 수행하는 과정을 설명하기 위한 도면이다.

[35] 도 3은 종래 기술에 따라 희소 데이터 기반의 컨볼루션 연산을 수행하는 경우 발생하는 문제점을 설명하기 위한 도면이다.

[36] 도 4는 본 발명의 일 실시예에 따른 인공신경망을 이용한 희소 데이터 기반의 컨볼루션 연산 장치의 일부 구성 요소를 도시한 블록도이다.

[37] 도 5는 본 발명의 일 실시예에 따른 데이터 플로우 최적화 모듈이 타일의 크기를 조정하는 방법을 설명하기 위한 도면이다.

[38] 도 6및 도 7은 본 발명의 일 실시예에 따른 데이터 플로우 최적화 모듈이 웨이트 그룹 기반의 입력 데이터를 수집하는 방법을 설명하기 위한 도면이다.

[39] 도 8과 도 9는 본 발명에 따른 컨볼루션 연산 방법과 종래 기술에 따른 컨볼루션 연산 방법을 비교 설명하기 위한 도면이다.

[40] 도 10은 본 발명에 따른 컨볼루션 연산 방법과 종래 기술에 따른 컨볼루션 연산 방법의 실험 결과를 비교 도시한 도면이다.

[41] 도 11은 본 발명의 일 실시예에 따른 데이터 플로우 최적화 모듈이 웨이트 그룹 기반의 분산(scatter) 연산을 수행하는 방법을 설명하기 위한 도면이다.

[42] 도 12 및 도 13은 본 발명에 따른 컨볼루션 연산과 종래 기술에 따른 컨볼루션 연산 속도의 실험 결과를 비교 도시한 도면이다.

발명의 실시를 위한 최선의 형태

[43] 본 명세서에 기재된 실시 예와 도면에 도시된 구성은 개시된 발명의 바람직한 일 예이며, 본 출원의 출원 시점에 있어서 본 명세서의 실시 예와 도면을 대체할 수 있는 다양한 변형 예들이 있을 수 있다.

[44] 또한, 본 명세서에서 사용한 용어는 실시 예를 설명하기 위해 사용된 것으로, 개시된 발명을 제한 및/또는 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다.

[45] 본 명세서에서, "포함하다", "구비하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 작동, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 작동, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는다.

[46] 또한, 본 명세서에서 사용한 "제 1", "제 2" 등과 같이 서수를 포함하는 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되지는 않는다.

[47] 아래에서는 첨부한 도면을 참고하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략한다.

[48] 한편, 본 명세서에서 발명의 명칭은 '데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법 및 장치'로 기재하였으나, 이하 설명의 편의를 위해 '데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법'은 '컨볼루션 연산 방법'으로 축약하고, '데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 장치'는 '컨볼루션 연산 장치'로 축약하여 설명하도록 한다.

[49]

[50] 도 1은 본 발명의 일 실시예에 따라 컨볼루션 연산 장치에 입력되는 입력 데이터의 일 예와 컨볼루션 연산을 수행하는 커널의 일 예를 도시한 도면이다.

[51] 컨볼루션 연산은, 입력 데이터(10)를 분석하여, 입력 인덱스 정보를 수집하는 과정에서 시작된다. 본 발명에서의 (index) 정보는 입력 데이터 내에 서 실제 데이터가 존재하는 유효 데이터가 위치하고 있는 포인트에 대한 위치 정보를 의미한다.

[52] 일 예로, 도 1에 도시된 바와 같은 5 X 5행렬의 데이터가 입력 데이터로 입력이 된다면(행렬에서 빗금친 부분이 데이터가 존재하는 영역이고, 민무늬 부분이 데이터가 존재하지 않는 영역을 의미하며, 데이터가 존재하는 위치의 데이터를 유효 데이터라 지칭하기로 한다) 인덱스 정보 추출 모듈(210)은 입력 데이터 내에서 유효 데이터가 위치하고 있는 위치 정보를 입력 인덱스 정보로 추출할 수 있다.

[53] 인덱스 정보를 표현하는 방법은 행렬 데이터를 표현하는 공지되어 있는 방법이면 그 어느 것이든 차용될 수 있는데, 대표적으로 CSR(Compressed Spare Row) 포맷이 사용될 수 있다. 이하 설명의 편의를 위해 본 발명에서의 인덱스 정보는 CSR 형식을 기준으로 설명하기로 한다.

[54] 구체적으로, CSR 포맷 형식은 순차적으로 각각의 행에 유효 데이터가 몇 개가 존재하는지에 대한 정보인 CSR_row 정보와, 순차적으로 각각의 행에 유효 데이터가 몇 열에 위치하는지에 대한 정보인 CSR_col에 대한 정보를 포함하고 있으며, CSR 형식 좌표 표현 방식은 일반적으로 행렬 표현 방법과 동일하나, 시작이 1이 아니라 0에서 시작하는 것에 차이점이 존재한다.

[55] 따라서, 데이터에서 가장 왼쪽 및 가장 위쪽에 있는 좌표는 (1,1)로 표현하지 않고 (0,0)으로 표현된다. 따라서, 입력 데이터는 0행 0열을 기준으로 행과 열이 시작된다고 볼 수 있으며, 커널 또한 3X3 행렬의 커널인 경우 커널의 오프셋은

행 기준으로 (0,1,2), 열 기준으로 (0,1,2)가 된다. 즉, 입력 데이터와 커널 모두 0행을 기준으로 표현이 된다.

[56] 도 1에 표시된 입력 데이터에 대해 인덱스 정보가 생성되는 과정을 순차적으로 설명하면, CSR_row 값은 데이터와 무관하게 처음은 0으로 시작하며, 2번째 값부터 각 행별로 있는 유효 데이터의 개수 만큼 누적된 값을 나타낸다. 따라서, 도면에 표시된 바와 같이 CSR_row는 0으로 시작한다.

[57] 제0행에서는 유효 데이터가 1개 (I0)존재하므로 제0행까지 고려한 에 CSR_row의 값은 [0,1]로 표현된다.

[58] 제1행에서는 유효 데이터가 1개 (I1)존재하므로 제1행까지 고려한 CSR_row 값은 [0,1,2]로 표현이 된다.

[59] 제2행에서는 유효 데이터가 2개(I2, I3) 존재하므로 제2행까지 고려한 CSR_row 값은 [0,1,2,4]로 표현이 된다.

[60] 제3행에서는 유효 데이터가 1개(I4) 존재하므로 제3행까지 고려한 CSR_row값은 [0,1,2,4,5]로 표현이 된다.

[61] 제4행에서는 유효 데이터가 1개(I5) 존재하므로 제4행까지 고려한 CSR_row값은 [0,1,2,4,5,6]로 표현이 된다.

[62] CSR_col은 순차적으로 각각의 행에서 몇 번째 열(column)에 유효 데이터가 위치하고 있는지에 대한 정보를 알려주는 정보이다. 따라서, 도 1을 기준으로 제0행에서는 4번째 열의 위치에 데이터가 존재하므로 CSR_col은 [4]가 되고, 제1행에는 1 열에 유효 데이터가 존재하므로 CSR_col은 [4,1]이 된다. 제2행에는 유효 데이터가 2번째 열과 3번째 열에 존재하므로 CSR_col은 [4,1,2,3]이 되고, 제3행에는 유효 데이터가 3 열에 위치하므로 CSR_col은 [4,1,2,3.3]이 된다. 이러한 방식으로 데이터를 생성하면 최종CSR_col는 [4,1,2,3,3,2] 가 된다.

[63]

[64] 도 2는 종래 기술에 따라 희소 데이터 기반의 컨볼루션 연산을 수행하는 과정을 설명하기 위한 도면이고, 도 3은 종래 기술에 따라 희소 데이터 기반의 컨볼루션 연산을 수행하는 경우 발생하는 문제점을 설명하기 위한 도면이다.

[65] 도 2를 참고하면, LiDAR를 통해 얻는 3차원 공간에 존재해는 객체(object)에 대한 포인트 클라우드(point cloud, 20)는 도면에 도시된 바와 같이 밀도가 희박한(Sparse)한 정보를 가지고 있다. 따라서, 이러한 데이터들에서는 바로 컨볼루션 연산을 수행하지 않고 입력된 포인트 클라우드 데이터에 대해 희소 유사 이미지(pseudo image)로 변환과정을 거친다. (S10) 즉, S10 과정은 2차원 희소 컨볼루션(spare convolution) 입력 데이터를 생성하는 것이다.

[66] 다만, 희소 이미지 데이터에 대해 컨볼루션 연산을 수행하면 연산의 비효율 문제가 발생하게 되므로, 희소 이미지를 밀집된 형태의 데이터로 변환한 후, 변환된 데이터를 기초로 밀집(dense) 컨볼루션 연산을 수행하게 된다. 그러나, 이러한 과정을 거쳐서 컨볼루션 연산을 수행하더라도 희소 데이터의 희소 정도

(sparsity)는 일반적으로 95%를 초과하기 때문에, 밀집 컨볼루션 연산을 위한 치환 연산은 매우 비효율적인 문제를 가지고 있다.

[67] 구체적으로, 이를 공간적 아키텍처 가속기와 연관 지어 설명하면, 공간적 아키텍처 가속기는 신경망 연산의 대부분을 차지하는 텐서 연산을 효율적으로 처리 할 수 있어, 신경망 가속에 일반적으로 많이 사용된다.

[68] 가속기의 텐서 연산은, 연산을 수행하기 위해 필요한 데이터의 파티셔닝(공간적 분할)과 스케줄링(시간적 분할)을 나타내는 데이터 플로우로 나뉘어 질 수 있다.

[69] 공간적 아키텍처 가속기는 다양한 데이터 플로우를 가질 수 있는데, 데이터 플로우를 어떻게 설정하는지에 따라 가속기 내에서 데이터 재사용과 하드웨어 사용률의 변화가 발생하므로, 이는 성능과 에너지 효율에 큰 영향을 미친다. 따라서, 주어진 가속기의 구성과 텐서 연산에 대해 성능과 에너지 효율을 최적화하는 방법들이 제안되고 있으며, 이를 데이터 플로우 최적화라고 한다.

[70] 기존의 데이터 플로우 최적화 방법은 밀집된 데이터 기반의 컨볼루션 연산에 적용되는 규칙적인 텐서 연산만을 고려하기 때문에, 이는 희소 데이터 기반의 희소 컨볼루션 연산에 적용되는 불규칙적인 텐서 연산의 최적화에 한계가 존재한다.

[71] 구체적으로, 밀집 데이터 기반의 컨볼루션 연산은 커널의 특성(높이*넓이, Stride, Dilation)에 기초하여 규칙적인 텐서 연산을 수행하므로, 특정 크기(높이*넓이)의 입력 데이터, 커널의 연산에 따른 출력 데이터의 크기(높이*넓이), 메모리 계층에서의 데이터 재사용 및 인덱스 별 연산 횟수가 커널에 의해서 결정되는 특징을 가지고 있다.

[72] 종래의 데이터 플로우 최적화 방법은 이러한 규칙성을 활용하여 최적화를 진행하며, 구체적으로 가속기의 입력 버퍼와 출력 버퍼 크기에 맞도록 연산을 데이터의 여러 Dimension에 대하여 타일 단위로 분할하고, 타일의 처리 순서에 따른 데이터 플로우의 성능을 평가하여, 최적의 타일 크기와 처리 순서를 결정한다.

[73] 그러나 이러한 방법은 Gather-GEMM-Scatter의 과정으로 수행되는 희소 데이터의 희소 합성곱의 불규칙한 연산에 대한 최적화에는 많은 어려움이 존재한다.

[74] 구체적으로, 종래 기술에 따라 Gather-GEMM-Scatter의 과정으로 수행되는 희소 데이터 기반의 컨볼루션 연산 방법은, 입력 데이터에서 0이 아닌 입력 데이터만 버퍼에 Gather하여 해당하는 커널과 연산을 수행하여 출력된 출력 데이터를 메모리에 Scatter 하므로, 입력 데이터의 배치에 따라 인덱스 별 재사용 횟수 및 번짐 효과로 인한 출력 데이터의 개수가 불규칙하게 나타나는 문제점이 있다. 따라서, 입력-커널-출력 데이터의 크기와 연산 횟수의 불규칙성으로 인하여, 타일링에 따른 입력/출력 버퍼의 사용률과, 각 버퍼에서의 데이터 재사용을 평가하는 것이 불가능한 단점이 존재한다.

[75] 즉, 희소 입력 데이터에 기반한 컨볼루션 연산의 경우, 입력 특성맵의 Non-zero 데이터의 배치, 희소성에 따른 번짐 효과로 인해 입력-출력 인덱스의 비율과 데이터의 재사용이 불규칙적인 문제가 발생한다.

[76] 구체적으로, 이를 도 3을 통해 알아보면, 도 3의 위쪽에 도시된 바와 같이 동일한 입력 인덱스 크기(Ta=3)를 기준으로 제1타일(Tile1) 영역에 대해 3x3 커널을 적용해 컨볼루션 연산을 하게 되면(스트라이딩 2 기준) 입력 인덱스와 아웃풋 인덱스의 비율은 25/3으로 8.33이 되나, 제2타일(Tile2) 영역을 기준으로 3x3 커널을 적용해 컨볼루션 연산을 하면(스트라이딩 2 기준) 입력 인덱스와 아웃풋 인덱스의 비율은 15/3으로 5가 된다.

[77] 즉, 불규칙적인 입력-출력 인덱스의 비율로 인해 입력 버퍼와 출력 버퍼의 사용량 예측 어렵게 되고, 불규칙적인 커널 인덱스의 재사용으로 인해 Mapping의 메모리 액서스 예측 어려운 단점이 존재한다. 이러한 현상은 도 3의 아래쪽 그래프에 도시된 바와 같이 밀집 데이터에 기반한 컨볼루션 연산보다 희소 데이터에 기반한 컨볼루션 연산에 더 많이 발생하게 된다.

[78] 따라서, 본 발명의 일 실시예에 따른 컨볼루션 연산의 데이터 플로우 최적화 방법 및 장치는 상기 설명한 문제점을 해결하기 위해 고안된 발명으로서, 컨볼루션 연산을 수행함에 있어서, 컨볼루션 연산의 특성(Conv, Strided Conv, Deconv), 입력 특성맵의 희소성/크기에 따른 번짐 효과의 영향을 고려한 데이터 플로우 최적화 방법을 제공하는데 목적이 존재한다.

[79] 보다 구체적으로는, Gather-GEMM-Scatter의 과정으로 수행되는 희소 입력 데이터에 대한 컨볼루션 연산에서, 데이터의 재사용이 극대화되도록 위이트 그룹(Weight Group)을 기반으로 Gather-Scatter을 수행하며, 고정된 버퍼 사이즈 하에서 컨볼루션 연산의 입력 버퍼와 출력 버퍼 사용량의 불균형을 개선하기 위해 계층의 연산 특성 및 희소성에 따른 입력 채널 사이즈와 출력 채널 타일 사이즈의 최적화를 수행하므로서, 컨볼루션 연산의 전반적인 속도를 높이는데 목적이 있다. 이하 도면을 통해 구체적으로 알아보도록 한다.

[80]

[81] 도 4는 본 발명의 일 실시예에 따른 인공신경망을 이용한 희소 데이터 기반의 컨볼루션 연산 장치의 일부 구성 요소를 도시한 블록도이다.

[82] 도 4를 참조하면, 일 실시예에 따른 희소 입력 데이터의 위치 정보에 기초한 컨볼루션 연산 장치는 프로세서(200)와 메모리 모듈(300)을 포함할 수 있고, 프로세서(200)는 입력 데이터 수집 모듈(210), 데이터 플로우 규칙 생성 모듈(220) 및 컨볼루션 연산 모듈(230)을 포함할 수 있으며, 메모리 모듈(300)은 입력 메모리(310), 입력 버퍼(320), 출력 버퍼(330) 및 출력 메모리(340)를 포함할 수 있다.

[83] 한편, 본 발명에서 컨볼루션 연산을 수행하기 이동되는 커널은, 3X3커널 및 스트라이드는 2를 기준으로 하여 설명한다. 그러나, 이는 본 발명의 일 실시예에 불과할 뿐, 본 발명의 원리는 2X2, 4X4, 5X5 등의 크기를 가지는 커널에 대해서도

적용될 수 있으며, 2차원 커널이 아닌 3차원 커널에서도 본 발명의 원리가 그대로 적용될 수 있다.

[84] 또한, 도 4에서는 설명의 편의를 위해 프로세서(200)와 메모리 모듈(300)을 분리하여 설명하였고, 프로세서(200) 내에서도 입력 데이터 수집 모듈(210), 데이터 플로우 규칙 생성 모듈(220) 및 컨볼루션 연산 모듈(230)을 각각의 별도의 구성 요소로 도시하여 설명하였지만, 이는 설명의 편의를 위해 구분한 것이고 각각의 모듈이 하는 역할을 하나의 프로세서(200)가 수행할 수도 있다.

[85] 입력 데이터 수집 모듈(210)은 입력 메모리(310)에 있는 입력 데이터(10)를 분석하여, 입력 데이터(10) 내에 있는 희소 입력 데이터의 입력 인덱스 정보를 수집하고, 입력 데이터(10) 내에 존재하는 희소 입력 데이터를 입력 버퍼(310)의 크기에 맞춰 수집한다.

[86] 데이터 플로우 규칙 생성 모듈(220)은 수집된 희소 입력 데이터와 입력 인덱스 정보를 기초로 입력/출력 채널에 대한 타일(tile) 사이즈를 조정하거나, 웨이트 그룹(weight group)기반의 인풋 가더(input gather) 및 아웃풋 스카터(output scatter)를 조정하여 컨볼루션 연산 수행에 대한 규칙을 생성한다. 이에 대한 자세한 설명은 후술하도록 한다.

[87] 컨볼루션 연산 모듈(230)은 입력 데이터 수집 모듈(210)이 수집한 희소 입력 데이터 및 데이터 플로우 규칙 생성 모듈(220)이 생성한 규칙을 기초로 컨볼루션 연산을 수행하고, 컨볼루션 연산에 의해 생성된 출력 데이터를 출력 버퍼(330)로 송신한다.

[88] 메모리 모듈(300)은 입력 데이터와 출력 데이터가 저장되는 모듈로서, 그 특성에 따라 출력 메모리(340)는 DRAM으로 구현될 수 있으며, 입력 버퍼(310), 웨이트 버퍼(320) 및 출력 버퍼(330)는 SRAM으로 구현될 수 있다.

[89] 입력 버퍼(310)에는 컨볼루션 연산에 수행될 희소 입력 데이터가 포함되어 있는 입력 데이터(10)가 임시적으로 저장되어 있을 수 있다.

[90] 입력 버퍼(310)에는 입력 데이터 수집 모듈(210)이 수집한 희소 입력 데이터가 임시적으로 저장될 수 있다.

[91] 출력 버퍼(330)에는 컨볼루션 연산 모듈(230)에 의해 수행된 출력 데이터가 임시적으로 저장될 수 있으며, 출력 메모리(340)에는 최종 출력 데이터가 저장될 수 있다.

[92]

[93] 도 5는 본 발명의 일 실시예에 따른 데이터 플로우 최적화 모듈이 타일의 크기를 조정하는 방법을 설명하기 위한 도면으로서, 구체적으로는 컨볼루션 연산 모듈(230)이 스탠다드 컨볼루션(Standard Convolution)을 수행할 때, 데이터 플로우 규칙 생성 모듈(220)이 생성하는 데이터 플로우 규칙을 설명하기 위한 도면이다.

[94] 도 5의 (a)를 참조하면, 입력 데이터(10) 내에서 희소 입력 데이터의 희소성이 높거나 입력 특성맵의 크기가 상대적으로 큰 경우에는 컨볼루션 연산을

수행함에 있어서 번짐 효과가 크게 발생하여 출력 버퍼와 입력 버퍼 크기의 불균형이 발생하게 된다.

[95] 구체적으로 도 5의 (a)에 도시된 바와 같이 아웃풋 인덱스와 입력 인덱스의 비율이 36/6=6이 되어 상당히 높은 비율로 불균형이 발생하게 된다. 따라서, 이러한 경우에는 입력 인덱스의 사용량이 적은 대신 아웃풋 인덱스의 사용량이 많아 인풋 버퍼는 상대적으로 적게 사용되나, 아웃풋 버퍼는 상대적으로 많이 사용되어, 버퍼 사용의 불균형 문제가 발생하게 된다.

[96] 그러나, 도 5의 (b)에 도시된 바와 같이 입력 데이터(10) 내에서 희소 입력 데이터의 희소성이 낮거나 입력 특성맵의 크기가 상대적으로 작은 경우에는 컨볼루션 연산을 수행함에 있어서 번짐 효과가 상대적으로 작게 발생하여 출력 버퍼와 입력 버퍼 크기의 불균형이 상대적으로 작게 발생하게 된다. 구체적으로 도 5의 (b)에 도시된 바와 같이 아웃풋 인덱스와 입력 인덱스의 비율이 75/36=2.08이 되어 도 5의 (a)의 경우와 다르게 불균형이 작아진다.

[97] 따라서, 본 발명에 따른 데이터 플로우 규칙 생성 모듈(220)은 입력 인덱스와 아웃풋 인덱스의 비를 계산하여, 미리 설정된 기준보다 계산된 비가 높은 경우에는 입력 채널에 대한 타일(tile)의 사이즈를 증가시켜, 입력 인덱스가 많이 사용되는 방법으로 입력 버퍼와 출력 버퍼 사이의 크기 불균형 문제를 해소시킬 수 있다. 일 예로 미리 설정된 기준은 다양한 숫자로 설정될 수 있으나 3 이상인 경우 타일의 사이즈를 증가시키도록 할 수 있으며, 계산된 입력 인덱스와 아웃풋 인덱스의 비가 커지면 커질수록 증가되는 타일의 사이즈 또한 비례하여 커지도록 설정할 수 있다.

[98]

[99] 도 6및 도 7은 본 발명의 일 실시예에 따른 데이터 플로우 최적화 모듈이 웨이트 그룹 기반의 입력 데이터를 수집하는 방법을 설명하기 위한 도면으로서, 컨볼루션 연산 모듈(230)이 스트라이드 컨볼루션(Strid Convolution)을 수행할 때, 데이터 플로우 규칙 생성 모듈(220)이 생성하는 데이터 플로우 규칙을 설명하기 위한 도면이다.

[100] 도 6 (a)는 입력 인덱스와 커널의 웨이트 인덱스가 컨볼루션 연산이 되어 출력 데이터가 생성되는 과정을 도시한 도면이고, 도 6의 (b)는 출력 데이터가 생성됨에 있어서 커널의 웨이트 그룹별로 대응되는 입력 인덱스를 그룹화 해놓은 도면이다.

[101] 도 6의 (a)를 참조하면, 입력 데이터(10)에 3x3 커널이 슬라이딩하면서 컨볼루션 연산을 수행하는 경우 제0입력 인덱스(O0)와 제1입력 인덱스(O1) 및 제5입력 인덱스(O5)는 도면에 도시된 바와 같이 입력 인덱스와 이에 대응되는 커널의 웨이트 인덱스의 곱에 의해 생성된다.

[102] 일반적으로 스트라이드 컨볼루션(Strided Convolution)은 입력되는 입력 인덱스에 대하여 웨이트 인덱스 간에 그룹을 지어 데이터를 재사용 하는 특성을 가지고 있는데, 3X3 커널에 의해 컨볼루션 연산이 이루어지면 도면에

도시된 바와 같이 커널의 웨이트들은 배치되어 있는 위치에 따라 대응되는 입력 인덱스의 범위가 그룹화되어 질 수 있다.

[103] 기본적인 데이터 플로우에서는 커널의 웨이트의 모든 인덱스(W0부터 W8)에 해당하는 Ta 개의 인풋 인덱스를 입력 버퍼에 수집(gather)하게 되는데, 이런 경우에는 입력 버퍼의 입력 인덱스 Ta 개 중 특정 웨이트 인덱스와 곱해지는 입력 인덱스는 1/4 로 감소하게 된다. 따라서, 입력 버퍼에서는 데이터 재사용이 감소하여 컨볼루션 연산의 효율적으로 이루어지지 못하는 단점이 존재한다.

[104] 그러나 본 발명의 경우 이를 개선하기 위해 커널의 특성 및 컨볼루션 연산의 특성을 고려하여 웨이트 그룹 기반으로 입력 데이터를 수집하므로 데이터의 재사용을 극대화를 수행할 수 있게 된다.

[105] 구체적으로, 특정 웨이트 그룹에 대응되는 입력 인덱스만을 입력 버퍼에 수집하게 되면, 특정 웨이트 그룹의 모든 웨이트에 대해 수집된 입력 인덱스가 모두 재사용될 수 있으므로 컨볼루션 연산을 효율성을 크게 향상시킬 수 있게 된다.

[106] 이를 도면을 통해 구체적으로 살펴보면, 제0웨이트 그룹(WG0)의 경우 제0웨이트 그룹에 포함되어 있는 제0,2,6,8웨이트 인덱스에 대응되는 입력 인덱스만을 입력 버퍼에 수집하게 되면, 제0웨이트 그룹(WG0)의 모든 웨이트 인덱스에 대하여 모든 입력 인덱스가 항상 재사용 되므로, 연산의 효율성을 높일 수 있다.

[107] 이를 웨이트 그룹별로 구체적으로 살펴보면, 제0웨이트 그룹(WG0)에 해당하는 제0웨이트 인덱스(W0), 제2웨이트 인덱스(W2), 제6웨이트 인덱스(W6) 및 제8웨이트 인덱스(W8)들은 제0입력 인덱스(I0), 제2입력 인덱스(I2), 제4입력 인덱스(I4), 제6입력 인덱스(I6), 제17입력 인덱스(I17), 제19입력 인덱스(I19), 제21입력 인덱스(I21), 제23입력 인덱스(I23), 제25입력 인덱스(I25), 제35입력 인덱스(I35), 제37입력 인덱스(I37) 및 제39입력 인덱스(I39)들 과만 컨볼루션 연산을 수행하게 된다.

[108] 또한, 제1웨이트 그룹(WG1)에 해당하는 제1웨이트 인덱스(W1) 및, 제7웨이트 인덱스(W7)들은 제1입력 인덱스(I1), 제3입력 인덱스(I3), 제5입력 인덱스(I5), 제18입력 인덱스(I18), 제20입력 인덱스(I20), 제22입력 인덱스(I22), 제24입력 인덱스(I24), 제26입력 인덱스(I26), 제34입력 인덱스(I34), 제36입력 인덱스(I36) 및 제38입력 인덱스(I38)들 과만 컨볼루션 연산을 수행하게 된다.

[109] 또한, 제2웨이트 그룹(WG2)에 해당하는 제3웨이트 인덱스(W3) 및 제5웨이트 인덱스(W5)들은 제7입력 인덱스(I7), 제9입력 인덱스(I9), 제11입력 인덱스(I11), 제13입력 인덱스(I13), 제15입력 인덱스(I15), 제28입력 인덱스(I28), 제30입력 인덱스(I30) 및 제32입력 인덱스(I32)들 과만 컨볼루션 연산을 수행하게 된다.

[110] 또한, 제3웨이트 그룹(WG3)에 해당하는 제4웨이트 인덱스(W4)는 제8입력 인덱스(I8), 제10입력 인덱스(I10), 제12입력 인덱스(I12), 제14입력 인덱스(I14),

제16입력 인덱스(I16), 제27입력 인덱스(I27), 제29입력 인덱스(I29), 제31입력 인덱스(I31) 및 제33입력 인덱스(I33)들 과만 컨볼루션 연산을 수행하게 된다.

[111] 이러한 방식으로 입력 데이터를 수집하고 컨볼루션 연산을 수행하게 되면, 모든 웨이트 인덱스가 출력 인덱스에 대한 데이터 재사용(Reduction)의 기회를 가지고 된다. 따라서, 도 7에 도시된 바와 같이 입력 버퍼(310)에 수집된 입력 인덱스에 대한 모든 웨이트 그룹에 대한 Gather-GEMM 연산을 완료한 후, Reduction이 완료된 출력 인덱스를 출력 버퍼(330)에서 출력 메모리(340)로 Scatter를 할 수 있게 되므로, 이를 통하여 인풋-아웃풋에 대하여 100% 데이터 재사용을 달성 할 수 있다.

[112]

[113] 도 8과 도 9는 본 발명에 따른 컨볼루션 연산 방법과 종래 기술에 따른 컨볼루션 연산 방법을 비교 설명하기 위한 도면으로서, 구체적으로 도 8은 종래 기술에 따라 웨이트 그룹 기반의 컨볼루션 연산 방법이 적용되지 않았을 때의 연산 과정을 설명하기 위한 도면이고, 도 9는 본 발명에 따라 웨이트 그룹 기반의 컨볼루션 연산 방법이 적용되었을 때의 연산 과정을 설명하기 위한 도면이다.

[114] 도 8을 참조하면, 종래 기술에 따라 컨볼루션 연산을 수행하는 경우 입력 버퍼의 제한된 크기로 인하여 입력 인덱스의 수집을 전체 입력 데이터 대해 2번에 나누어 수집하게 된다(제1수집(Ta=27), 제2수집(Ta=13)).

[115] 그리고 이러한 수집 데이터에 기초하여 컨볼루션 연산을 수행하는 경우 도면에 도시된 바와 같이 입력 버퍼에 수집된 입력 인덱스의 개수(Ta)대비 실제로 어레이(array)에서 연산에 사용되는 인덱스(Ta,eff)의 개수가 감소하게 되므로, 출력 버퍼가 입력 인덱스를 효율적으로 재사용하지 못하게 된다.

[116] 즉, 이에 따라 출력 버퍼의 용량을 효율적으로 사용하지 못하게 되므로, 입력 인덱스의 수집 횟수가 증가하게 되고, 이는 웨이트 데이터에 대한 출력 메모리(DRAM)로의 접근(access) 횟수가 증가하게 되어 컨볼루션 연산 속도가 저해되는 문제점이 발생하게 된다.

[117] 그러나, 본 발명에 따라 컨볼루션 연산을 수행하는 경우 도 9에 도시된 바와 같이 웨이트 그룹 단위로 입력 데이터에서 입력 인덱스를 수집하게 되므로, 더 넓은 인덱스 범위에 대해 데이터를 수집할 수 있게 되며, 이는 입력 인덱스의 수집 횟수 자체를 감소시킬 수 있다.

[118] 또한, 이에 따라 입력 버퍼에 수집된 입력 인덱스의 개수(Ta)대비 실제로 어레이(array)에서 연산에 사용되는 인덱스(Ta,eff)의 개수가 거의 동일하게 되므로, 출력 버퍼가 입력 인덱스를 대부분 재사용 할 수 있어, 컨볼루션 연산의 전체 속도를 높일 수 있는 장점이 존재한다.

[119] 도 10은 본 발명에 따른 컨볼루션 연산 방법과 종래 기술에 따른 컨볼루션 연산 방법의 실험 결과를 비교 도시한 도면으로서, 도 10의 (1)는 종래 기술에 따른 컨볼루션 연산에 따른 실험 결과를 도시한 그래프이고, 도 10의 (2)는 본 발명에 따라 웨이트 그룹 기반의 컨볼루션 연산에 따른 실험 결과를 도시한 그래프이며,

도 10의 (3)은 본 발명에 따라 웨이트 그룹 기반 및 타일링 최적화 방법을 적용한 컨볼루션 연산에 따른 실험 결과를 도시한 도면이다.

[120] 도 10의 (1)을 참고하면, 종래 기술에 따라 컨볼루션 연산을 하는 경우, 입력 데이터의 재사용 비율이 약 1/4(255/1023) 밖에 되지 않으나, 본 발명에 따라 웨이트 그룹 기반의 컨볼루션 연산을 수행 하는 경우 도 10의 (2)에 도시된 바와 같이 입력 데이터의 재사용 비율이 100%(358/358)임을 알 수 가 있다.

[121] 또한, 본 발명에 따른 타일링 기법까지 더 적용하여 컨볼루션 연산을 수행하는 경우, 도 10의 (3)에 도시된 바와 같이 입력 데이터의 재사용 비율이 100%(716/716)일 뿐만 아니라, 입력 인덱스의 수집 범위도 넓어져 보다 빠르게 연산을 수행할 있는 장점이 존재한다.

[122]

[123] 도 11은 본 발명의 일 실시예에 따른 데이터 플로우 최적화 모듈이 웨이트 그룹 기반의 분산(scatter) 연산을 수행하는 방법을 설명하기 위한 도면으로서, 컨볼루션 연산 모듈(230)이 디컨볼루션(Deconvolution)을 수행할 때, 데이터 플로우 규칙 생성 모듈(220)이 생성하는 데이터 플로우 규칙을 설명하기 위한 도면이다.

[124] 디컨볼루션의 경우에는 앞서 설명한 스트라이드 컨볼루션과는 다르게 컨볼루션 연산을 수행함에 있어서, 모든 웨이트 인덱스가 입력 인덱스를 공유하나, 그 특성상 웨이트 인덱스 간에 출력 인덱스에 대한 재사용을 하기가 어렵다. 따라서, 디컨볼루션의 경우에는 입력 데이터를 우선 입력 버퍼(210)에 수집 및 저장을 일시적으로 한 후, 웨이트를 도면에 도시된 바와 같이 일정한 기준 하에 그룹 지어 수집을 하고, 웨이트 그룹에 대한 연산이 끝나면 분산(Scatter) 연산을 하는 방법을 통해 데이터 플로우 규칙을 생성하면, 입력 데이터의 재사용을 최대화 할 수 있다.

[125] 웨이트 그룹의 크기는 출력 버퍼(330)의 크기에 따라 16x1, 8x2, 4x4, 2x8, 1x16의 패턴으로 조정을 할 수 있으며, 이에 따라 입력 버퍼와 출력 버퍼의 크기 불균형의 영향을 최소화 할 수 있다.

[126]

[127] 한편, 본 발명에 따른 데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 장치는, 연산에 있어서 데이터 재사용을 최대화하도록 데이터 플로우를 설정한 이후에, 입력 버퍼와 출력 버퍼의 사용량에 따라 도 5에서 설명하였던 입력 채널과 출력 채널에 대한 타일링 최적화를 수행할 수 도 있다.

[128] 컨볼루션 연산의 경우 입력 데이터 내의 희소 입력데이터의 위치 및 분포에 따라 번짐 효과가 달라지므로, 이에 따라 입력 인덱스와 출력 인덱스의 비율이 결정된다. 일 예로, 일반적인 컨볼루션은 그 비율이 1.5 배 이상, 스트라이드 컨볼루션의 경우 0.5 배, 디컨볼루션의 경우 웨이트의 크기에 따라 1배, 4배 또는 16배로 결정된다.

[129] 컨볼루션 연산을 함에 있어서, 특정 버퍼의 사용량이 100%가 되는 경우, 컨볼루션 연산의 속도를 저해하게 되므로, 출력 버퍼의 사용량이 100%가 되는 경우(즉, 입력 버퍼 사용량이 출력 버퍼에 비해 떨어지는 경우)에는 입력 채널에 대해 타일 사이즈를 증가시켜 입력 버퍼의 사용량을 증가시키는 방법으로, 버퍼의 사용량으로 인해 연산의 속도가 저해되는 문제점을 방지할 수 있다.

[130] 이와 반대로, 입력 버퍼의 사용량이 100%가 되는 경우(즉, 출력 버퍼의 사용량이 입력 버퍼에 비해 떨어지는 경우)에는, 출력 채널에 대하여 타일 사이즈를 증가시켜 출력 버퍼의 사용량을 증가시키는 방법으로, 버퍼의 사용량으로 인해 연산의 속도가 저해되는 문제점을 방지할 수 있다.

[131]

[132] 도 12 및 도 13은 본 발명에 따른 컨볼루션 연산과 종래 기술에 따른 컨볼루션 연산 속도의 실험 결과를 비교 도시한 도면이다.

[133] 도 12를 참고하면, 본 발명에 따른 스트라이드 컨볼루션(Strided Convolution)을 수행하는 경우, 웨이트 그룹 기반의 입력 수집 데이터 플로우를 통하여 입력 데이터 의 재사용을 50%에서 최대 100%로 향상시킬 수 있는 것을 알 수 있다. 이를 통해 데이터 재사용 감소에 따른 연산의 오버해드(Overhead, 전체 연산 대비 Non-Execution Cycle)가 62.9% 감소(12.4% -> 6%)되는 것을 알 수 있다.

[134] 본 발명에 따른 디컨볼루션(Deconvolution)을 수행하는 경우, 웨이트 그룹 기반의 분산(Scatter) 연산을 통해 데이터 플로우를 실행하는 경우, 출력 데이터 재사용과 버퍼 사용량을 최적화할 수 있기 때문에 연산의 오버해드가 종래 기술에 비해 56.9% 감소(29.9% -> 2.9%)되는 것을 알 수 있다.

[135] 또한, 본 발명에 따라 타일링 기법을 적용하여 컨볼루션 연산을 수행하는 경우, 고정된 버퍼 사이즈를 갖는 합성곱 처리 장치에서, 계층 별로 연산 특성과 희소성을 고려한 타일링을 통해 번짐 효과에 따른 입력 버퍼와 출력 버퍼의 사용량 불균형을 개선할 수 있다. 이러한 방법을 통해 연산의 오버해드가 20% 감소(4.5% -> 3.6%)되는 것을 알 수 있다.

[136] 이를 종합적으로 적용하여 컨볼루션 연산의 전체 속도를 살펴보면, 종래 기술에 따른 데이터 플로우를 적용한 컨볼루션 연산과 본 발명에 따란 데이터 플로우를 적용한 컨볼루션 연산의 경우, 도 13에 도시된 바와 같이 컨볼루션 연산의 수행 시간을 14.5%의 감소시키는 것을 알 수 있다.

[137] 지금까지 도면을 통해 본 발명의 구성 및 프로세스에 대해 자세히 알아보았다.

[138] 일 실시예에 따른 희소 입력 데이터의 위치 정보에 기초한 컨볼루션 연산 방법 및 장치는 희소 데이터 기반의 컨볼루션 연산을 수행함에 있어서, 입력 데이터와 출력 데이터의 특징을 고려하여 데이터를 수집하고 재활용하므로, 종래 기술보다 빠르게 컨볼루션 연산을 수행할 수 있는 장점이 존재한다.

[139] 또한, 컨불루션 연산의 경우 그 특성상 중첩되는 계산 과정이 많은데, 본 발명에 따른 컨볼루션 연산 방법은 종래에 수집하였던 출력 데이터의 정보를 다음

연산에서 재사용하므로, 컨볼루션 연산 자체의 연산 속도를 빠르게 증가시킬 수 있는 장점이 존재한다.

[140] 또한, 이러한 특징으로 인해 3차원 공간에 존재하는 객체 인식의 속도를 증가시킬 수 있는바, 고차원 자율 주행에 필수적인 고속 전방 장애물 인식을 효율적으로 수행할 수 있고, 빠르고 정확한 로봇 네비게이션을 위한 RGB-D 기반의 위치 추정도 효율적으로 수행할 수 있는 장점이 존재한다.

[141] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 컨트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 운영 체제 상에서 수행되는 하나 이상의 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.

[142] 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 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[143] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 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.

[144] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 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. 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

청구범위

[청구항 1] 프로세서, 입력 버퍼 및 출력 버퍼를 이용하여 컨볼루션 연산을 수행하는 방법에 있어서,
입력 데이터에 포함되어 있는 희소 입력 데이터에 대한 입력 인덱스 데이터를 수집하는 인덱스 데이터 수집 단계;
커널의 크기 및 상기 입력 인덱스 데이터에 기초하여 상기 커널의 웨이트들을 미리 설정된 기준에 따라 분류한 후, 분류된 정보에 기초하여 커널 웨이트 그룹을 생성하는 커널 웨이트 그룹 생성 단계;
상기 커널 웨이트 그룹에 대응되는 입력 인덱스 데이터를 매핑하여 데이터 플로우 규칙을 생성하는 데이터 플로우 규칙 생성 단계; 및
상기 데이터 플로우 규칙에 기초하여 컨볼루션 연산을 수행하는 컨볼루션 연산 단계;를 포함하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 2] 제 1항에 있어서,
상기 커널은, 3X3 커널을 포함하고,
상기 컨볼루션 연산은 스탠다드 컨볼루션(standard convolution) 또는 스트라이드 컨볼루션(stride convolution) 연산을 포함하며,
상기 커널 웨이트 그룹 생성 단계는,
상기 웨이트 그룹은 3X3 커널에 기초한 컨볼루션 연산의 특성을 고려하여, 대칭적으로 복수 개의 웨이트 그룹을 생성하는 단계를 포함하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 3] 제2항에 있어서,
상기 커널 웨이트 그룹 생성 단계는,
제0웨이트 인덱스(W0), 제2웨이트 인덱스(W2), 제6웨이트 인덱스(W6) 및 제8웨이트 인덱스(W8)를 제0웨이트 그룹으로 생성하는 단계를 포함하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 4] 제2항에 있어서,
상기 커널 웨이트 그룹 생성 단계는,
제1웨이트 인덱스(W1) 및 제7웨이트 인덱스(W7)를 제1웨이트 그룹으로 생성하는 단계를 포함하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 5] 제2항에 있어서,

상기 커널 웨이트 그룹 생성 단계는,
제3웨이트 인덱스(W3) 및 제5웨이트 인덱스(W5)를 제2웨이트 그룹으로 생성하는 단계를 포함하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 6] 제2항에 있어서,
상기 커널 웨이트 그룹 생성 단계는,
제4웨이트 인덱스(W4)를 제3웨이트 그룹으로 생성하는 단계를 포함하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 7] 제2항에 있어서,
상기 데이터 플로우 규칙 생성 단계는,
상기 웨이트 그룹에 대응되는 입력 인덱스를 수집한 후, 수집된 입력 인덱스를 상기 인풋 버퍼에 상기 웨이트 그룹 별로 순차적으로 저장하는 단계;를 포함하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 8] 제 1항에 있어서,
상기 컨볼루션 연산은 디컨볼루션(decovolution)을 포함하며,
상기 커널 웨이트 그룹 생성 단계는,
입력 버퍼 및 출력 버퍼의 사이즈에 기초하여 웨이트 그룹의 크기를 결정하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 9] 제 1항에 있어서,
상기 데이터 플로우 규칙 생성 단계는,
상기 입력 인덱스와 상기 입력 인덱스에 대응되는 출력 인덱스의 비율에 기초하여 상기 입력 데이터에 대한 타일링(tiling) 사이즈를 조절하는 타일링 사이즈 조절 단계를 포함하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 10] 제9항에 있어서,
상기 타일링 사이즈 조절 단계는,
스탠다드 컨볼루션(standard convolution) 또는 스트라이드 컨볼루션(stride convolution)의 경우 입력 채널에 대한 타일링 사이즈를 증가시키고,
디컨볼루션(decovolution)의 경우 출력 채널에 대한 타일링 사이즈를 감소시키는 단계를 포함하는,

데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 방법.

[청구항 11] 입력 데이터에 포함되어 있는 희소 입력 데이터에 대한 입력 인덱스 데이터를 수집하는 인덱스 데이터 수집 모듈;
커널의 크기 및 상기 입력 인덱스 데이터에 기초하여 상기 커널의 웨이트들을 미리 설정된 기준에 따라 분류한 후, 분류된 정보에 기초하여 커널 웨이트 그룹을 생성하고, 상기 커널 웨이트 그룹에 대응되는 입력 인덱스 데이터를 매핑하여 데이터 플로우 규칙을 생성하는 데이터 플로우 규칙 생성 모듈;
상기 데이터 플로우 규칙에 기초하여 컨볼루션 연산을 수행하는 컨볼루션 연산 모듈;을 포함하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 장치.

[청구항 12] 제 11항에 있어서,
상기 커널은, 3X3 커널을 포함하고,
상기 컨볼루션 연산은 스탠다드 컨볼루션(standard convolution) 또는 스트라이드 컨볼루션(stride convolution) 연산을 포함하며,
상기 데이터 플로우 규칙 생성 모듈은,
상기 웨이트 그룹은 3X3 커널에 기초한 컨볼루션 연산의 특성을 고려하여, 대칭적으로 복수 개의 웨이트 그룹을 생성하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 장치.

[청구항 13] 제12항에 있어서,
상기 데이터 플로우 규칙 생성 모듈은,
상기 웨이트 그룹에 대응되는 입력 인덱스를 수집한 후, 수집된 입력 인덱스를 인풋 버퍼에 상기 웨이트 그룹 별로 순차적으로 저장하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 장치.

[청구항 14] 제11항에 있어서,
상기 컨볼루션 연산은 디컨볼루션(decovolution)을 포함하며,
상기 데이터 플로우 규칙 생성 모듈은,
입력 버퍼 및 출력 버퍼의 사이즈에 기초하여 웨이트 그룹의 크기를 결정하는,
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 장치.

[청구항 15] 입력 데이터에 포함되어 있는 희소 입력 데이터에 대한 입력 인덱스 데이터를 수집하는 인덱스 데이터 수집 모듈;

커널의 크기 및 상기 입력 인덱스 데이터에 기초하여 상기 커널의 웨이트들을 미리 설정된 기준에 따라 분류한 후, 분류된 정보에 기초하여 커널 웨이트 그룹을 생성하고, 상기 커널 웨이트 그룹에 대응되는 입력 인덱스 데이터를 매핑하여 데이터 플로우 규칙을 생성하는 데이터 플로우 규칙 생성 모듈;

상기 데이터 플로우 규칙에 기초하여 컨볼루션 연산을 수행하는 컨볼루션 연산 모듈;을 포함하며,

상기 데이터 플로우 규칙 생성 모듈은,

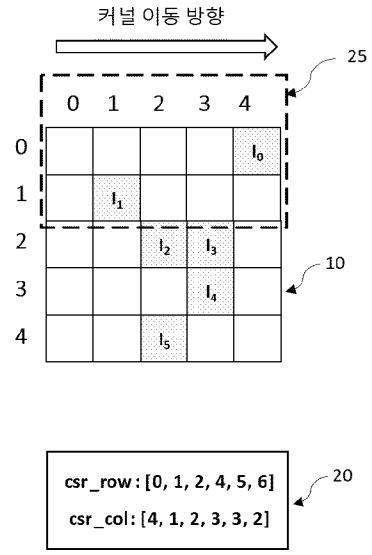
스탠다드 컨볼루션(standard convolution) 또는 스트라이드 컨볼루션(stride convolution)의 경우 입력 채널에 대한 타일링 사이즈를 증가시키고,

디컨볼루션(decovolution)의 경우 출력 채널에 대한 타일링 사이즈를 감소시키는,

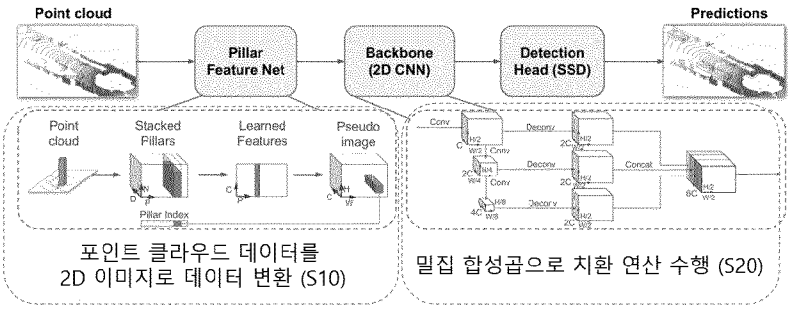
데이터 플로우 최적화 방법을 이용한 인공신경망 기반의 컨볼루션 연산 장치.

WO 2024/237652 PCT/KR2024/006514

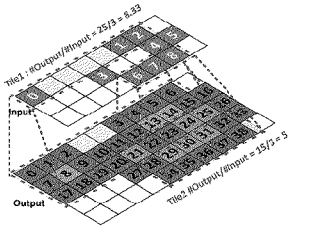
[도1]



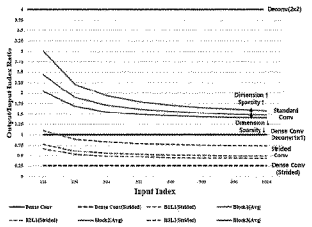
[도2]



[도3]



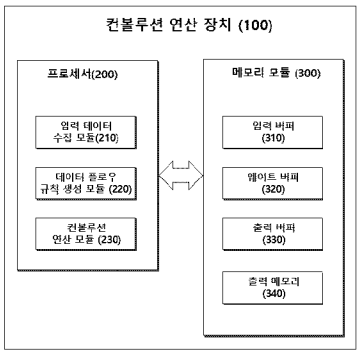
(a) 희소 합성곱에서의 입력-출력 인덱스의 불규칙성



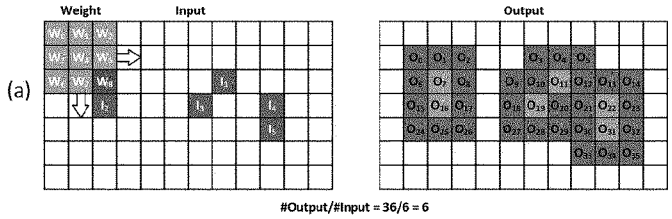
(b) 합성곱의 연산 특성, 입력 인덱스에 따른 출력-입력 인덱스의 비율

규칙 91,
18.07.2024

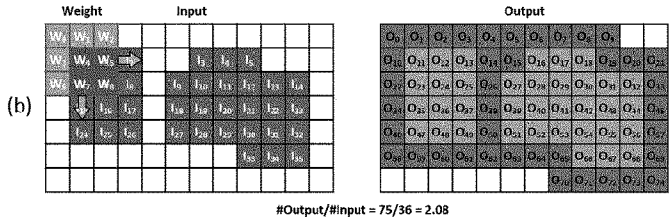
[도4]



[도5]



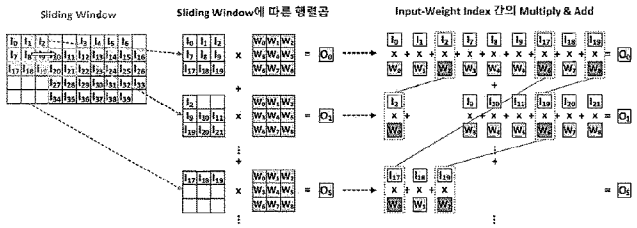
(a) 입력 특성맵 크기가 크거나 데이터의 희소성이 높은 경우



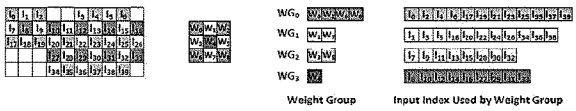
(b) 입력 특성맵 크기가 작거나 입력 데이터의 희소성이 낮은 경우

규칙 91, 18.07.2024

[도6]



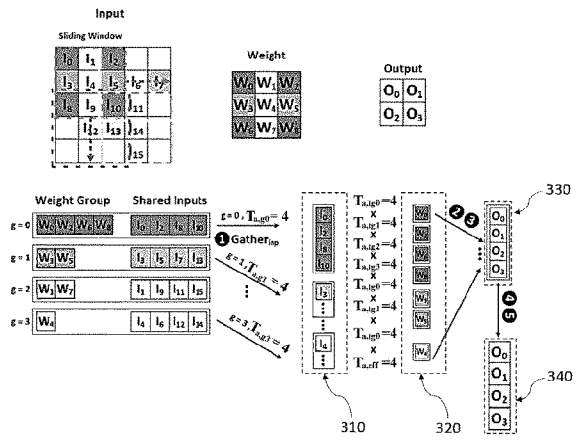
(a) 출력 인덱스를 기준으로 한 인풋 인덱스와 웨이트 인덱스의 대응 관계



(b) 웨이트 그룹별로 대응되는 인풋 인덱스의 범위

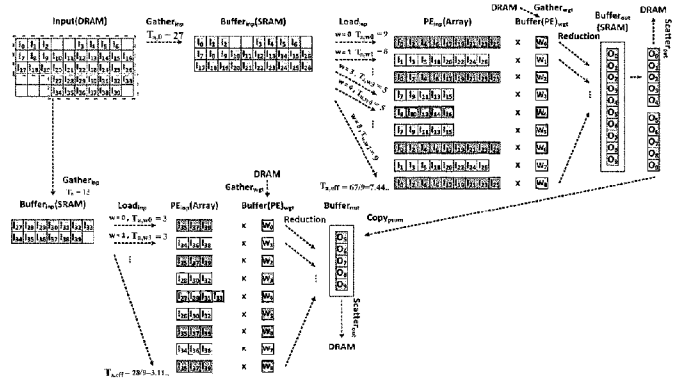
규칙 91, 18.07.2024

[도7]



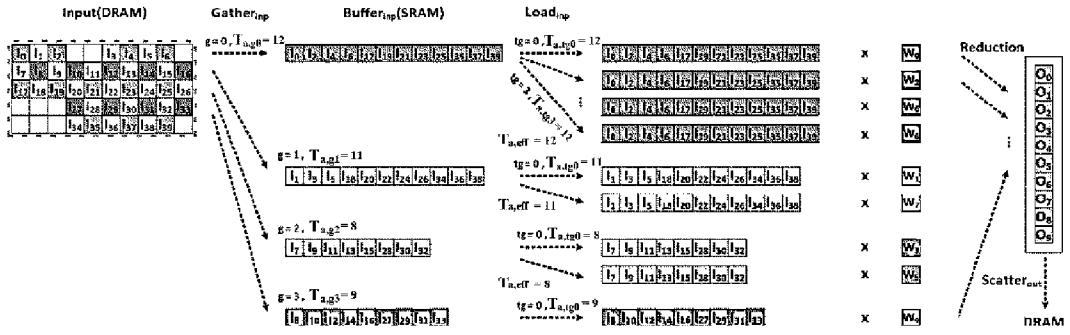
규칙 91,
18.07.2024

[도8]



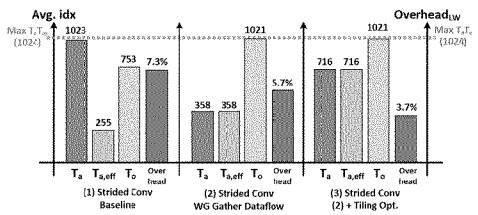
규칙 91,
18.07.2024

[도9]

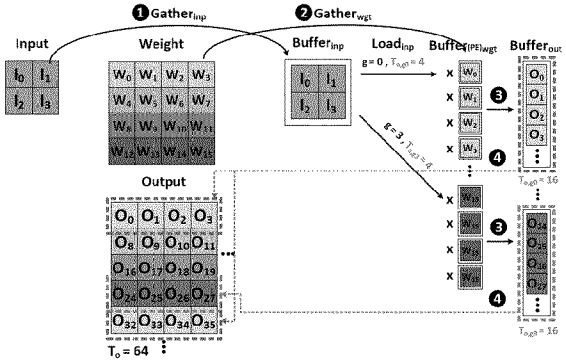


규칙 91,
18.07.2024

[도10]



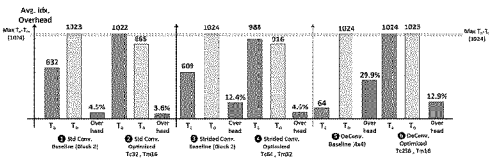
[도11]



[도12]

Operation	Layer Characteristics				Optimization Technique	
	Input Reuse Across Weight Index	PSUM Reduction Across Weight Index	O/I Ratio	Memory Bottleneck	Weight Group Dataflow	Tiling Optimization
(1) Standard Conv.	All Indices	All Indices	> 1.5	Output Buffer	N/A	Increase T_c
(2) Strided Conv.	Among Weight Group	All Indices	< 1	Input Buffer	WG Based Gather	Increase T_m
(3) Deconvolution	All Indices	N/A	1, 4, 16	Output Buffer	WG Based Scatter	Increase T_c

(a) Layer Characteristics and its Optimization Technique



(b) Average Index , Overhead by Tiling Optimization

[도13]

Point Pillar - Layer Profile							Sparsity			Baseline Dataflow	Optimized Dataflow	
Block	Layer	Layer Type	Feature	Kernel	Stride	Input Ch.	Output Ch.	#Point	Sparsity	Ops	Cycle	T_m/T_{ch}	Cycle
1	1	Downsample	496x432	2x3	2	64	64	5,318	97.52%	49010688	50451	64/64	**45749**
1	2	Convolution	248x216	3x3	1	64	64	3,709	93.38%	136228576	76059	64/64	**75625**
1	3	Convolution	248x216	3x3	1	64	64	6,753	87.39%	248942592	105612	64/64	**105184**
1	4	Convolution	248x216	3x3	1	64	64	9,151	82.32%	337342464	132000	64/64	**131542**
1	5	Upsample	248x216	1x1	1	64	128	11,741	78.38%	96182272	232763	64/64	**232763**
2	1	Downsample	248x216	3x3	2	64	128	4,355	91.37%	80271360	63122	64/64	**50262**
2	2	Convolution	124x108	3x3	1	128	128	1,338	90.01%	197296128	69879	128/64	**69192**
2	3	Convolution	124x108	3x3	1	128	128	2,586	80.39%	381321216	106853	128/64	**105610**
2	4	Convolution	124x108	3x3	1	128	128	3,273	75.58%	482623488	132414	128/64	**130507**
2	5	Convolution	124x108	3x3	1	128	128	2,679	80.20%	395034624	119443	128/64	**118404**
2	6	Convolution	124x108	3x3	1	128	128	3,080	77.20%	454:64480	132181	128/64	**130433**
2	7	Upsample	124x108	2x2	1	128	128	4,721	64.75%	309395456	377129	128/64	**375469**
3	1	Downsample	124x108	3x3	2	128	256	2,777	79.10%	204374016	113310	128/64	**78213**
3	2	Convolution	62x54	3x3	1	256	256	670	79.99%	395:82080	109596	128/64	**107352**
3	3	Convolution	62x54	3x3	1	256	256	1,002	70.37%	591003648	158606	128/64	**155721**
3	4	Convolution	62x54	3x3	1	256	256	1,039	68.97%	612427136	163196	128/64	**160321**
3	5	Convolution	62x54	3x3	1	256	256	1,039	68.97%	612427136	162919	128/64	**159775**
3	6	Convolution	62x54	3x3	1	256	256	1,037	69.03%	611047488	162235	128/64	**159402**
3	7	Upsample	62x54	4x4	1	256	128	1,523	54.51%	798490624	1676971	128/64	**782842**
4	1	Detection Head	248x216	3x3	1	384	80	25,770	51.59%	7124889600	2936493	128/64	**2886167**
								Total Ops/Cycle		14119555072	7091174		**6960333**
								Execution Time(ms)			7.08		**6.95**
												Speed Up	**14.9%**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2024/006514

A. CLASSIFICATION OF SUBJECT MATTER

G06N 3/0464(2023.01)i; **G06N 3/0495**(2023.01)i; **G06N 20/10**(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06N 3/0464(2023.01); G06N 20/10(2019.01); G06N 3/04(2006.01); G06N 3/063(2006.01); G06N 3/08(2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models: IPC as above
Japanese utility models and applications for utility models: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS (KIPO internal) & keywords: 컨볼루션 연산(convolution operation), 희소 데이터(sparsity data), 커널(kernel), 가중치(weight), 플로우(flow), 타일(tile), 사이즈(size)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	KR 10-2390379 B1 (SAMSUNG ELECTRONICS CO., LTD. et al.) 26 April 2022 (2022-04-26) See paragraphs [0063] and [0073]; and claims 1, 7 and 13.	1-8,11-14
Y		9-10,15
Y	KR 10-2021-0074173 A (INTEL CORPORATION) 21 June 2021 (2021-06-21) See claims 1 and 3.	9-10,15
A	KR 10-2021-0055582 A (INTEL CORPORATION) 17 May 2021 (2021-05-17) See paragraphs [0331]-[0357].	1-15
A	US 2021-0303980 A1 (POSTECH RESEARCH AND BUSINESS DEVELOPMENT FOUNDATION) 30 September 2021 (2021-09-30) See claims 1-10.	1-15
A	US 2020-0218978 A1 (NEURAL MAGIC INC.) 09 July 2020 (2020-07-09) See paragraphs [0144]-[0150]; and figure 7.	1-15

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 August 2024	**21 August 2024**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office **Government Complex-Daejeon Building 4, 189 Cheongsa-ro, Seo-gu, Daejeon 35208**	
Facsimile No. **+82-42-481-8578**	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/KR2024/006514

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
KR 10-2390379 B1	26 April 2022	KR 10-2018-0101978 A	14 September 2018
KR 10-2021-0074173 A	21 June 2021	CN 112950448 A	11 June 2021
		DE 102020130184 A1	17 June 2021
		JP 2021-093131 A	17 June 2021
		TW 202122994 A	16 June 2021
		US 11127108 B2	21 September 2021
		US 11443407 B2	13 September 2022
		US 2021-0183002 A1	17 June 2021
		US 2022-0092723 A1	24 March 2022
KR 10-2021-0055582 A	17 May 2021	CN 112785481 A	11 May 2021
		DE 102020129409 A1	12 May 2021
		JP 2021-077342 A	20 May 2021
		US 2021-0142179 A1	13 May 2021
US 2021-0303980 A1	30 September 2021	CN 113469323 A	01 October 2021
		KR 10-2021-0121533 A	08 October 2021
		KR 10-2474787 B1	06 December 2022
		US 11928578 B2	12 March 2024
US 2020-0218978 A1	09 July 2020	US 11544559 B2	03 January 2023

국 제 조 사 보 고 서

국제출원번호
PCT/KR2024/006514

A. 발명이 속하는 기술분류(국제특허분류(IPC))

G06N 3/0464(2023.01)i; **G06N 3/0495**(2023.01)i; **G06N 20/10**(2019.01)i

B. 조사된 분야

조사된 최소문헌(국제특허분류를 기재)

G06N 3/0464(2023.01); G06N 20/10(2019.01); G06N 3/04(2006.01); G06N 3/063(2006.01); G06N 3/08(2006.01)

조사된 기술분야에 속하는 최소문헌 이외의 문헌

한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC
일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우))

eKOMPASS(특허청 내부 검색시스템) & 키워드: 컨볼루션 연산(convolution operation), 희소 데이터(sparsity data), 커널(kernel), 가중치(weight), 플로우(flow), 타일(tile), 사이즈(size)

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
X	KR 10-2390379 B1 (삼성전자주식회사 등) 2022.04.26 단락 [0063], [0073]; 및 청구항 1, 7, 13	1-8,11-14
Y		9-10,15
Y	KR 10-2021-0074173 A (인텔 코포레이션) 2021.06.21 청구항 1, 3	9-10,15
A	KR 10-2021-0055582 A (인텔 코포레이션) 2021.05.17 단락 [0331]-[0357]	1-15
A	US 2021-0303980 A1 (POSTECH RESEARCH AND BUSINESS DEVELOPMENT FOUNDATION) 2021.09.30 청구항 1-10	1-15
A	US 2020-0218978 A1 (NEURALMAGIC INC.) 2020.07.09 단락 [0144]-[0150]; 및 도면 7	1-15

☐ 추가 문헌이 C(계속)에 기재되어 있습니다.

☑ 대응특허에 관한 별지를 참조하십시오.

* 인용된 문헌의 특별 카테고리:

"A" 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌
"D" 본 국제출원에서 출원인이 인용한 문헌
"E" 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌
"L" 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌
"O" 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌
"P" 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌
"T" 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌
"X" 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.
"Y" 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.
"&" 동일한 대응특허문헌에 속하는 문헌

국제조사의 실제 완료일	국제조사보고서 발송일
2024년08월21일(21.08.2024)	**2024년08월21일(21.08.2024)**

ISA/KR의 명칭 및 우편주소

대한민국 특허청
(35208) 대전광역시 서구 청사로 189, 4동 (둔산동, 정부대전청사)

팩스 번호 **+82-42-481-8578**

심사관

양정록

전화번호 **+82-42-481-5709**

국 제 조 사 보 고 서
대응특허에 관한 정보

국제출원번호
PCT/KR2024/006514

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
KR 10-2390379 B1	2022/04/26	KR 10-2018-0101978 A	2018/09/14
KR 10-2021-0074173 A	2021/06/21	CN 112950448 A	2021/06/11
		DE 102020130184 A1	2021/06/17
		JP 2021-093131 A	2021/06/17
		TW 202122994 A	2021/06/16
		US 11127108 B2	2021/09/21
		US 11443407 B2	2022/09/13
		US 2021-0183002 A1	2021/06/17
		US 2022-0092723 A1	2022/03/24
KR 10-2021-0055582 A	2021/05/17	CN 112785481 A	2021/05/11
		DE 102020129409 A1	2021/05/12
		JP 2021-077342 A	2021/05/20
		US 2021-0142179 A1	2021/05/13
US 2021-0303980 A1	2021/09/30	CN 113469323 A	2021/10/01
		KR 10-2021-0121533 A	2021/10/08
		KR 10-2474787 B1	2022/12/06
		US 11928578 B2	2024/03/12
US 2020-0218978 A1	2020/07/09	US 11544559 B2	2023/01/03