(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0143042
(43) 공개일자 2023년10월11일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2023.01) *G06F 17/16* (2006.01)
G06F 7/523 (2006.01) *G06N 3/08* (2023.01)
G11C 11/401 (2006.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06F 17/16 (2013.01)
(21) 출원번호 10-2022-0041848
(22) 출원일자 2022년04월04일
심사청구일자 2022년04월04일

(71) 출원인
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
노원우
서울특별시 강남구 삼성로51길 35, 201동 1202호
갈홍주
서울특별시 서대문구 연희로10길 79-20, 208호
(뒷면에 계속)
(74) 대리인
특허법인(유한)아이시스

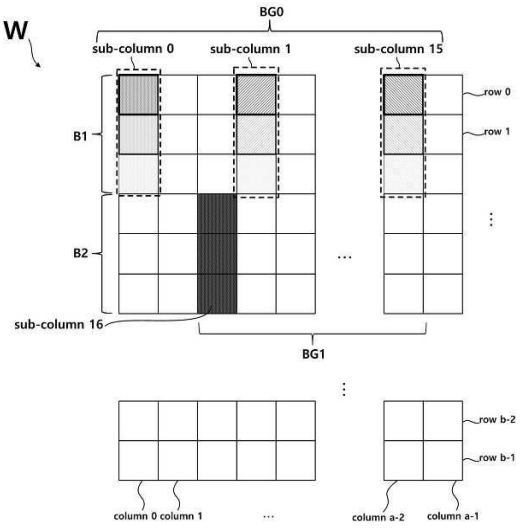
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **메모리 최적화 기술을 포함하는 신경망 처리 방법**

(57) 요 약

본 실시예에 의한 PIM(processing-in-memory) 연산기와, SRAM 버퍼 및 DRAM 어레이를 포함하는 DRAM(dynamic RAM) 모듈에서 수행되는 신경망 처리 방법은: 가중치 행렬을 블록 단위로 프루닝(pruning)하여 압축하는 압축 단계와, 입력 벡터를 상기 DRAM 모듈 내의 SRAM 버퍼에 타일링하는 타일링 단계 및 압축된 상기 가중치 행렬을 상기 DRAM 어레이의 동일한 행에 배치하는 배치 단계를 포함한다.

대 표 도 - 도2



(52) CPC특허분류
G06F 7/523 (2013.01)
G06N 3/082 (2023.01)
G11C 11/401 (2018.05)
(72) 발명자
박천준
서울특별시 서대문구 연세로5다길 22-8
이현욱
서울특별시 마포구 모래내로1길 17, 415호

정이품
서울특별시 마포구 백범로25길 82-17, 102호
이지원
서울특별시 서대문구 성산로20길 24

이 발명을 지원한 국가연구개발사업

과제고유번호	1711134555
과제번호	2021-0-00853
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원(한국연구재단부설)
연구사업명	신개념PIM반도체선도기술개발(R&D)
연구과제명	[통합이지바로] PIM 활용을 위한 SW 플랫폼 개발(1/2)
기 여 율	1/1
과제수행기관명	연세대학교 산학협력단
연구기간	2021.04.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

PIM(processing-in-memory) 연산기와, SRAM 버퍼 및 DRAM 어레이를 포함하는 DRAM(dynamic RAM) 모듈에서 수행되는 신경망 처리 방법으로, 상기 방법은:

가중치 행렬을 블록 단위로 프루닝(pruning)하여 압축하는 압축 단계와,

입력 벡터를 상기 DRAM 모듈 내의 SRAM 버퍼에 타일링하는 타일링 단계 및

압축된 상기 가중치 행렬을 상기 DRAM 어레이의 동일한 행에 배치하는 배치 단계를 포함하는 신경망 처리 방법.

청구항 2

제1항에 있어서,

상기 압축 단계는,

상기 가중치 행렬의 2^n-1(n: 자연수) 개의 행을 하나의 블록으로 하여 수행하는 신경망 처리 방법.

청구항 3

제2항에 있어서,

상기 압축 단계는,

상기 블록에서 0이 아닌 값을 가지는 서브 컬럼을 2^h(h: 자연수)개 단위로 하는 블록 그룹을 형성하는 단계 및 열 인덱스 및 값 어레이를 형성하는 단계로 수행되고, 상기 열 인덱스 및 값 어레이의 원소들 각각은 상기 블록 그룹에 포함된 서브 컬럼들의 열 인덱스(column)와 서브 컬럼에 포함된 원소의 값들을 포함하는 신경망 처리 방법.

청구항 4

제3항에 있어서,

상기 압축 단계는,

상기 열 인덱스 및 값 어레이를 지시하는 블록 그룹 포인터 어레이를 형성하는 단계를 더 포함하되,

상기 블록 그룹 포인터 어레이에 포함된 원소(element)들 각각은 상기 열 인덱스 및 값 어레이의 각 원소를 지시하는 신경망 처리 방법.

청구항 5

제3항에 있어서,

상기 열 인덱스 및 값 어레이의 원소들 각각은

상기 DRAM 어레이의 행의 데이터 사이즈에 부합하는 데이터 사이즈를 가지는 신경망 처리 방법.

청구항 6

제3항에 있어서,

상기 배치 단계는,

상기 열 인덱스 및 값 어레이의 원소들 각각을 상기 DRAM 어레이에 배치하여 수행하는 신경망 처리 방법.

청구항 7

제3항에 있어서,

상기 블록 그룹을 형성하는 단계는,

상기 0이 아닌 값을 가지는 서브 컬럼의 개수가 상기 2^h(h: 자연수)개 미만이면 0을 채워(zero filling) 상기 블록 그룹을 형성하는 신경망 처리 방법.

청구항 8

제1항에 있어서,

상기 SRAM 버퍼의 용량은,

상기 입력 벡터의 데이터 사이즈에 상응하는 신경망 처리 방법.

청구항 9

제1항에 있어서,

상기 신경망 처리 방법은,

상기 압축된 가중치 행렬과 타일링된 상기 입력 벡터의 곱을 누적하는 MAC(multiply and accumulate) 연산 단계를 더 포함하는 신경망 처리 방법.

청구항 10

제1항에 있어서,

상기 신경망 처리 방법은,

상기 PIM(processing-in-memory) 연산기가 수행하는 신경망 처리 방법.

발명의 설명

기 술 분 야

[0001] 본 기술은 신경망 처리 방법과 관련된다.

배 경 기 술

[0002] 심층 신경망(Deep Neural Network, DNN)은 컴퓨터 비전, 자연 언어 프로세싱 및 개인화된 추천 시스템을 포함하는 다양한 애플리케이션 도메인에 널리 적용되는 추세이다. 컨볼루션 신경망(convolution neural network, CNN), 다층 퍼셉트론(multi-layer perceptron; MLP), 장기 단기 메모리(long short-term memory; LSTM) 및 메모리 증강 신경망(memory augmented neural network; MANN)과 같은 다양한 DNN 알고리즘들은 주로 매트릭스 연산들을 포함한다. 매트릭스 연산들은 큰 메모리 풋프린트 및 낮은 데이터 재사용 레이트를 나타내기 때문에, 이들 DNN 층들의 성능은 메모리 대역폭에 의해 제한된다. 이러한 동작들을 위한 유망한 해결책으로서, DRAM 뱅크들 내부의 PIM(Processing-in-Memory)을 채용하여 뱅크 근처 PIM으로 분류된 뱅크들의 총 대역폭을 활용한다. 니어-뱅크 PIM 연산 환경에서 메모리-집약적 특성들을 나타내는 가중치 행렬 및 입력 벡터의 곱셈을 목표로 하였다.

발명의 내용

해결하려는 과제

[0003] 실제의 DNN 계층들은 가중치 행렬이 희소 매트릭스(sparse matrix)로 표현되어, 일반적 행렬-벡터 곱셈 연산기가 이들을 처리하기에 비효율적이므로, 이를 해소하기 위하여 종래 기술은 희소 행렬인 가중치 행렬(weight matrix)을 압축하여 벡터와 가중치 행렬 사이의 곱셈 및 누적 연산을 수행하였다.

[0004] 그러나, 가중치 행렬을 압축하는 종래 기술에 의하여도 인덱싱 데이터의 전송은 메모리 뱅크 내에서의 과도한

메모리 트래픽을 야기한다. 또한, 벡터와의 곱셈을 위하여 가중치 행렬의 형태에 따라 벡터 원소(vector element)에 대한 불규칙적인 접근에 의하여 메모리의 대역폭 소모가 크다. 나아가, 압축된 가중치 값들을 행 단위로 메모리에 매핑하면 DRAM 특성상 서로 다른 셀 어레이의 행에 저장이 되는데, 이러한 경우에는 읽기 및 쓰기시 별도의 기능이 수행되어 수행 시간에 따른 성능이 열화된다는 문제점이 있다.

[0005] 본 실시예로 해결하고자 하는 과제 중 하나는 상술한 종래 기술에 의한 난점을 해소하기 위한 것이다.

과제의 해결 수단

[0006] 본 실시예에 의한 PIM(processing-in-memory) 연산기와, SRAM 버퍼 및 DRAM 어레이를 포함하는 DRAM(dynamic RAM) 모듈에서 수행되는 신경망 처리 방법은: 가중치 행렬을 블록 단위로 프루닝(pruning)하여 압축하는 압축 단계와, 입력 벡터를 상기 DRAM 모듈 내의 SRAM 버퍼에 타일링하는 타일링 단계 및 압축된 상기 가중치 행렬을 상기 DRAM 어레이의 동일한 행에 배치하는 배치 단계를 포함한다.

[0007] 본 실시예의 어느 한 측면에 의하면, 상기 압축 단계는, 상기 가중치 행렬의 2^n-1(n: 자연수) 개의 행을 하나의 블록으로 하여 수행한다.

[0008] 본 실시예의 어느 한 측면에 의하면, 상기 압축 단계는, 상기 블록에서 0이 아닌 값을 가지는 서브 컬럼을 2^h(h: 자연수)개 단위로 하는 블록 그룹을 형성하는 단계 및 열 인덱스 및 값 어레이를 형성하는 단계로 수행되고, 상기 열 인덱스 및 값 어레이의 원소들 각각은 상기 블록 그룹에 포함된 서브 컬럼들의 열 인덱스(column)와 서브 컬럼에 포함된 원소의 값들을 포함한다.

[0009] 본 실시예의 어느 한 측면에 의하면, 상기 압축 단계는, 상기 열 인덱스 및 값 어레이를 지시하는 블록 그룹 포인터 어레이를 형성하는 단계를 더 포함하되, 상기 블록 그룹 포인터 어레이에 포함된 원소(element)들 각각은 상기 열 인덱스 및 값 어레이의 각 원소를 지시한다.

[0010] 본 실시예의 어느 한 측면에 의하면, 상기 열 인덱스 및 값 어레이의 원소들 각각은 상기 DRAM 어레이의 행의 데이터 사이즈에 부합하는 데이터 사이즈를 가진다.

[0011] 본 실시예의 어느 한 측면에 의하면, 상기 배치 단계는, 상기 열 인덱스 및 값 어레이의 원소들 각각을 상기 DRAM 어레이에 배치하여 수행한다.

[0012] 본 실시예의 어느 한 측면에 의하면, 상기 블록 그룹을 형성하는 단계는, 상기 0이 아닌 값을 가지는 서브 컬럼의 개수가 상기 2^h(h: 자연수)개 미만이면 0을 채워(zero filling) 상기 블록 그룹을 형성한다.

[0013] 본 실시예의 어느 한 측면에 의하면, 상기 SRAM 버퍼의 용량은, 상기 입력 벡터의 데이터 사이즈에 상응한다.

[0014] 본 실시예의 어느 한 측면에 의하면, 상기 신경망 처리 방법은, 상기 압축된 가중치 행렬과 타일링된 상기 입력 벡터의 곱을 누적하는 MAC(multiply and accumulate) 연산 단계를 더 포함한다.

[0015] 본 실시예의 어느 한 측면에 의하면, 상기 신경망 처리 방법은, 상기 PIM(processing-in-memory) 연산기가 수행한다.

발명의 효과

[0016] 본 실시예에 의하면 메모리 내부 대역폭을 사용하여 입력 벡터와 가중치 행렬과의 MAC 연산이 수행되므로 종래 기술에 비하여 신속하고 효율적인 연산이 가능하다는 장점이 제공된다.

도면의 간단한 설명

[0017] 도 1은 본 실시예의 신경망 처리 방법의 개요를 도시한 순서도이다.

도 2는 본 실시예에 의한 프루닝된 가중치 행렬을 도시한 도면이다.

도 3은 가중치 행렬의 압축을 위한 블록 그룹 포인터 어레이, 열 인덱스 및 값 어레이를 도시한 도면이다.

도 4는 도 2로 예시된 가중치 행렬과 입력 벡터의 MAC 연산 과정을 예시하는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0018] 이하에서는 첨부된 도면들을 참조하여 본 실시예를 설명한다. 도 1은 본 실시예의 신경망 처리 방법의 개요를 도시한 순서도이다. 도 1을 참조하면, 신경망 처리 방법은 PIM(processing-in-memory) 연산기와, SRAM 버퍼 및 DRAM 어레이를 포함하는 DRAM(dynamic RAM) 모듈에서 수행되는 신경망 처리 방법으로, 가중치 행렬을 블록 단위로 프루닝(pruning)하여 압축하는 압축 단계(S100)와, 입력 벡터를 상기 DRAM 모듈 내의 SRAM 버퍼에 타일링하는 타일링 단계(S200) 및 압축된 상기 가중치 행렬을 상기 DRAM 어레이의 동일한 행에 배치하는 배치 단계(S300)를 포함한다. 도시되지 않은 실시예에 의하면, 신경망 처리 방법은 압축된 가중치 행렬과 타일링된 상기 입력 벡터의 곱을 누적하는 MAC(multiply and accumulate) 연산 단계를 더 포함할 수 있다.

[0019] 도 2는 본 실시예에 의한 프루닝된 가중치 행렬을 도시한 도면이다. 도 2를 참조하면, 가중치 행렬(W)은 column 0, column 1, ..., column a-1의 a 개 열과 row 0, row 1, ..., row b-1의 b 개 행을 포함한다. 가중치 행렬(W)에서 0이 아닌 값을 가지는 원소는 원소의 내부를 음영으로 도시하였으며, 값으로 0을 가지는 원소는 백색으로 도시하였다. 가중치 행렬(W)은 도시된 바와 같이 값을 가지는 원소들의 개수에 비하여 값이 없는(즉, 값으로 0을 가지는) 원소들의 개수가 더 많은 희소 행렬(sparse matrix)의 형태를 가진다.

[0020] 희소 행렬을 압축하여 메모리에 저장하고, 입력 벡터와의 MAC(multiply and accumulate) 연산을 수행하면 메모리에 저장된 원소들과 더불어 수많은 행과 열들에 대한 인덱스 값에 접근(access)를 수행하여야 하므로 병목현상이 발생하여 연산 효율이 감소한다. 이러한 특징상 희소 행렬을 압축된 형태로 DRAM 등의 메모리에 저장하여 PIM(Processing in Memory) 연산장치로 MAC 연산을 수행하는 경우에도 메모리 내부의 트래픽을 증가시켜 연산 효율을 감소시킨다.

[0021] 본 실시예에서, 가중치 행렬(W)은 복수의 행들로 이루어진 블록들(B1, B2, ...)으로 분할된다. 일 실시예로, 블록들(B1, B2) 각각에 포함된 행들의 개수는 2^n-1(n: 자연수) 개일 수 있다. 이하 예시되는 실시예는 블록들(B1, B2)을 이루는 행들의 개수는 3을 상정하여 설명한다. 그러나, 이는 단지 용이한 설명을 위한 것으로 본 발명의 범위를 제한하고자 하는 것이 아니며, 하나의 블록은 7 개의 행, 15개의 행 등으로 이루어질 수 있다.

[0022] 가중치 행렬(W)은 각 원소별로 2 바이트로 표시될 수 있는 부동 소수점(floating point) 값을 가질 수 있고, 원소를 지시하는 인덱스 값은 정수(integer) 값을 가질 수 있다. 신경망에 대한 최초 학습이 완료되면 가중치 행렬(W)의 각 원소들에는 학습 결과에 상응하는 값들이 설정된다. 가중치 행렬(W)에서 0이 아닌 값을 가지는 원소를 포함하는 서브 컬럼들은 신경망 설계시에 결정된다. 가중치 행렬(W)의 각 원소에 설정된 값들 중 0에 근사한 값들은 0으로 설정되며, 이러한 단계를 프루닝(pruning) 단계라고 하며, 블록 단위로 프루닝이 수행될 수 있다.

[0023] 프루닝이 완료된 가중치 행렬(W)에 대한 압축을 수행한다. 일 실시예로, 제1 블록(B1)은 0이 아닌 값을 가지는 복수의 서브 컬럼들(sub-column 0, sub-column 1, ..., sub-column 15)을 포함한다. 도시된 실시예에서, 제1 블록(B1)은 16개의 0이 아닌 값을 가지는 서브 컬럼들(sub-column 0, sub-column 1, ..., sub-column 15)을 포함한다.

[0024] 동일한 블록 내에 포함되어 0이 아닌 값을 가지는 서브 컬럼들을 2^h(h: 0, 또는 양의 정수)개씩 하나의 블록 그룹으로 형성한다. 도시된 실시예에서, 각각의 블록 그룹은 16개의 0이 아닌 값을 가지는 서브 컬럼들을 포함한다. 즉, 제0 서브 컬럼(sub-column 0) 내지 제15 서브 컬럼(sub-column 15)은 블록 그룹(BG0)에 포함된다. 다만, 이것은 용이한 설명을 위한 것으로, 하나의 블록 그룹에 포함되는 0이 아닌 값을 가지는 서브 컬럼의 개수는 8, 32 등과 같이 달리 설정될 수 있다.

[0025] 도시되지 않은 실시예에서, 제1 블록(B1)은 0이 아닌 값을 가지는 서브 컬럼 16(sub-column 16)을 더 포함할 수 있다. 이러한 예에서, 서브 컬럼 16(sub-column 16) 및 제1 블록(B1)에서 0이 아닌 값을 가지는 서브 컬럼들은 제1 블록 그룹에 속할 수 있다.

[0026] 도시되지 않은 다른 실시예에서, 제1 블록(B1)에서 0이 아닌 값을 가지는 서브 컬럼이 하나 이상 2^h(h: 0, 또는 양의 정수)개 미만일 수 있다. 이러한 예에서, 0이 아닌 값을 가지는 서브 컬럼과 함께 0으로 채워진 값들로 블록 그룹을 형성할 수 있다.

[0027] 도 3은 가중치 행렬(W)의 압축을 위한 블록 그룹 포인터 어레이(BG_Ptr), 열 인덱스 및 값 어레이를 도시한 도면이다. 도 2 및 도 3을 참조하면, 블록 그룹 포인터 어레이(BG_Ptr)의 i 번째 원소의 값은 0에서 i 번째 블록 그룹까지 포함된 0이 아닌 값을 가지는 서브 컬럼들의 개수를 나타낸다.

[0028] 따라서, 도 3으로 도시된 블록 그룹 포인터 어레이(BG_Ptr)의 0번째 원소의 값은, 이전까지의 블록 그룹에 포함

된 0이 아닌 값을 가지는 서브 컬럼들의 개수가 0이므로 0의 값을 가진다. 또한, 블록 그룹 포인터 어레이(BG_Ptr) 1번째 원소의 값은 블록 그룹(BG0)에 포함된 0이 아닌 값을 가지는 서브 컬럼들의 개수인 16이다. 따라서, i 번째 블록 그룹(BGi)에 포함된 0 이 아닌 서브 컬럼들의 개수는 블록 그룹 포인터 어레이의 i 번째 원소값과 i+1 번째 원소 값의 차이를 연산하면 얻을 수 있다.

[0029] 도 2로 예시된 실시예에서, 블록 그룹(BG1)에 0이 아닌 값을 가지는 서브 컬럼 그룹으로 sub-column 16의 하나만 있다면 블록 그룹 포인터 어레이(BG_Ptr)의 2번째 원소의 값은 17이며, 블록 그룹(BG1)의 나머지는 0으로 채워진 값을 가진다.

[0030] 블록 그룹 포인터 어레이(BG_Ptr)의 원소들 각각은 해당 블록 그룹의 열 인덱스 및 값 어레이(BCV)를 지시한다. 블록 그룹 포인터 어레이(BG_Ptr)의 i 번째 원소는 i 번째 블록 그룹(BGi)에 상응하는 열 인덱스 및 값 어레이(BCV)를 지시한다.

[0031] 열 인덱스 및 값 어레이(BCV)는 해당 블록 그룹(BG)에 포함된 0이 아닌 값을 가지는 서브 컬럼들의 열 인덱스를 저장하는 서브 어레이(BG_ColIdx)와 BG_ColIdx에 상응하는 서브 컬럼들에 포함된 원소들의 값들을 저장하는 값 서브 어레이(Value)를 포함한다.

[0032] 도 2와 도 3으로 예시된 실시예에서, 블록 그룹(BG0)에 포함된 서브 컬럼 0(sub-column 0)의 열 인덱스는 0이고, 서브 컬럼 1(sub-column 1)의 열 인덱스는 3이며, 서브 컬럼 15(sub-column 15)의 열 인덱스가 a-2 라면, 열 인덱스 및 값 어레이(BCV)에 포함된 원소 BG_ColIdx는 0, 3, ..., a-2를 저장한다.

[0033] 또한 값을 저장하는 값 서브 어레이(Value)는 서브 컬럼 0(sub-column 0)에 포함된 원소들의 값, 서브 컬럼 1(sub-column 1)에 포함된 원소들의 값 내지 서브 컬럼 15(sub-column 15)에 포함된 원소들의 값들을 저장한다. 저장되는 순서는 각 서브 컬럼의 첫 번째 value 들을, 이어서 두 번째 value 들을 저장하는 방식으로 순차적으로 저장하는 데, 마지막에는 2^n-1(n: 자연수)번째 value 원소를 저장한다. 이와 같이 블록 그룹 포인터 어레이(BG_Ptr)와 열 인덱스 및 값 어레이(BCV)를 이용하여 프루닝된 희소 행렬인 가중치 행렬을 압축할 수 있다(S100).

[0034] 이어서, 입력 벡터를 타일링 한다(S200). 신경망을 연산하는 과정에서 빈번하게 입력 벡터(V)에 대한 접근이 이루어진다. 즉, 희소 행렬의 원소와 곱셈이 이루어지는 벡터의 원소에 대한 접근이 이루어지는 것으로, 가중치 행렬(W)에서 값이 0이 아닌 원소에 상응하는 벡터 데이터에 대한 불규칙한 액세스 패턴을 나타내므로 연산의 병목 현상을 발생시켜 결과적으로 연산 속도에 악영향을 미친다.

[0035] 이를 해소하기 위하여 DRAM 모듈에 포함된 SRAM 버퍼에 입력 벡터를 타일링하여 지속적으로 사용한다. 일 실시예로, SRAM 버퍼의 용량은 입력 벡터의 용량에 상응할 수 있다. 일 예로, 입력 벡터가 1024 개의 원소를 가지고, 원소 각각의 크기가 16 비트이면, SRAM 버퍼의 용량은 2kByte일 수 있으며, 입력 벡터는 2kByte로 타일링되어 활용된다. 또한, 가중치 행렬은 DRAM 어레이에 타일링되어 사용된다.

[0036] 배치 단계에서 압축된 상기 가중치 행렬을 상기 DRAM 어레이의 동일한 행에 배치한다(S300). 압축된 가중치 행렬에 포함된 열 인덱스 및 값 어레이(BCV)는 DRAM 어레이의 동일한 행에 배치된다. 일 실시예로, 블록 그룹 열 인덱스를 저장하는 BG_ColIdx 서브 어레이는 2 Byte 크기의 원소를 포함하며, 각 원소는 16개의 열 인덱스를 저장하므로, BG_ColIdx 서브 어레이는 32 Byte 크기를 가진다.

[0037] 또한, 값 서브 어레이(Value)는 16 개의 0이 아닌 값을 가지는 서브 컬럼의 값들을 저장한다. 서브 컬럼에 포함된 원소의 크기는 2Byte이므로, 각 서브 컬럼당 32Byte 이므로 블록에 포함된 행의 개수 i와의 곱인 i*32Byte의 이다. i는 2^n-1 개이므로 값 서브 어레이(Value)의 크기는 32B, 96B, 224B, 480B, 992B가 될 수 있다.

[0038] BG_ColIdx 서브 어레이와 값 서브 어레이(Value)로 이루어진 BCV의 값은 64B, 128B, 256B, 1024B일 수 있으며, 이 값은 DRAM 어레이의 행의 크기와 상응할 수 있다.

[0039] 따라서. 본 실시예에 의하면, 가중치 행렬에서 동일한 블록 그룹은 DRAM 어레이의 동일한 행에 배치될 수 있다. 이로부터 가중치 행렬에 속한 원소들이 DRAM 어레이의 서로 다른 행에 저장되어 발생하는 충돌에 의한 시간 지연에 의한 성능 저하를 막을 수 있다는 장점이 제공된다.

[0040] 도 4는 도 2로 예시된 가중치 행렬(W)과 입력 벡터(V)의 MAC 연산 과정을 예시하는 도면이다. 도 4를 참조하면, 블록 그룹 포인터 어레이(BG_Ptr)의 원소들로부터 어느 한 블록 그룹(BG)에 속한 0이 아닌 값을 가지는 서브 컬럼들의 개수를 파악한다. 상술한 바와 같이, 블록 그룹 포인터 어레이(BG_Ptr)에서 서로 인접한 원소들의 차이

를 연산하여 해당 블록(B)에 속한 0이 아닌 값을 가지는 서브 컬럼들의 개수를 파악할 수 있다.

[0041] 도 4로 예시된 것과 같이 열 인덱스 및 값 어레이(BCV)의 0번째 원소에 포함된 열 인덱스 서브 어레이(BG_ColIdx)는 0 이 아닌 값을 가지는 서브 컬럼들의 열 인덱스 값인 0, 3, ..., a-2 를 저장한다. 또한 값 서브 어레이(Value)는 각각 0 이 아닌 값을 가지는 서브 컬럼들의 값들을 행별로 저장한다.

[0042] 블록 그룹 포인터(BG_Ptr)가 지시하는 열 인덱스 및 값 어레이의 0번째 원소 BCV[0] 에서 0이 아닌 값을 가지는 서브 컬럼들의 열 인덱스인 0, 3, ..., a-2 값과 이에 대한 값들을 파악한다. 파악된 열 인덱스로부터 SRAM 버퍼에 타일링된 벡터(V)에서 0, 3, ... a-2 번째 원소값들(음영)을 취득하고, 벡터(V)의 원소값들과 값 서브 어레이(Value)에 저장된 값들과 곱셈 연산한 후, 곱셈 연산 결과를 누적한다. 이와 같은 방식으로 동일한 블록에 속한 서브 컬럼과 벡터와의 MAC 연산을 수행할 수 있다.

[0043] 도시된 예에서, 벡터(V) 원소는 상술한 바와 같이 DRAM 모듈 내의 SRAM 버퍼에 저장되고, 벡터(V) 원소와 연산되는 가중치 행렬(W)의 원소들은 DRAM에 저장되며, 이들에 대한 MAC 연산은 DRAM 모듈 내에 포함된 PIM 프로세서에 의하여 이루어진다. 도시된 예와 같이 입력 벡터(V)와 가중치 행렬(W)의 MAC 연산 수행시 압축된 가중치 행렬을 압축 해제하지 않고 MAC 연산을 수행할 수 있어 MAC 연산에 필요한 시간을 감축시킬 수 있다는 장점이 제공된다.

[0044] 즉, 종래 기술에서 메모리 내부에 저장된 벡터 값과 가중치 행렬의 값을 페치(fetch)하여 중앙 처리 장치(CPU)에서 연산을 수행하는데 필요한 시간에 비하여 적은 시간으로 연산을 수행할 수 있어 효율적이라는 장점이 제공된다. 나아가, 희소 행렬인 가중치 행렬을 압축하여 DRAM 행의 용량에 맞추어 배치할 수 있어 불규칙적 메모리 접근에 의한 연산의 비효율성을 제거할 수 있다는 장점이 제공된다. 또한 하나의 서브 컬럼이 동일한 인덱스를 공유하기 때문에, 인덱스를 읽기 위해 발생하는 메모리 트래픽을 줄일 수 있다.

[0046] 본 발명에 대한 이해를 돕기 위하여 도면에 도시된 실시 예를 참고로 설명되었으나, 이는 실시를 위한 실시예로, 예시적인 것에 불과하며, 당해 분야에서 통상적 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 특허청구범위에 의해 정해져야 할 것이다.

부호의 설명

[0047] B1, B2: 블록 BG1: 제1 블록 그룹

W: 가중치 행렬 BG_Ptr: 블록 그룹 포인터

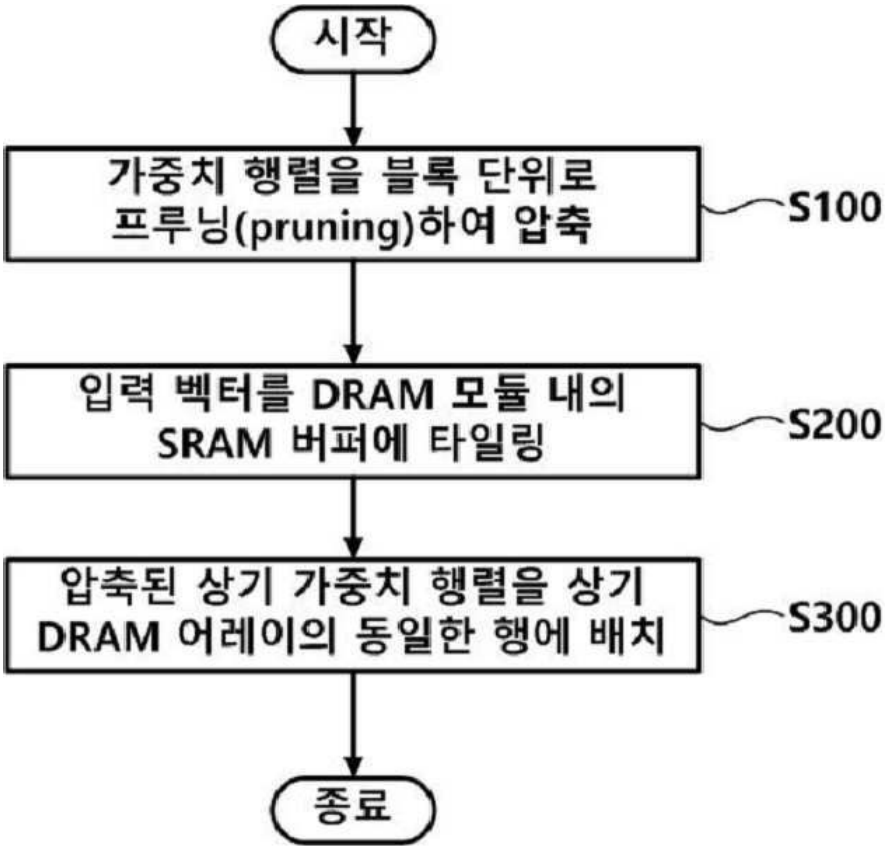
BCV: 블록 그룹의 열 인덱스 및 값 어레이

BG_ColIdx: 블록 그룹 컬럼 인덱스 서브 어레이

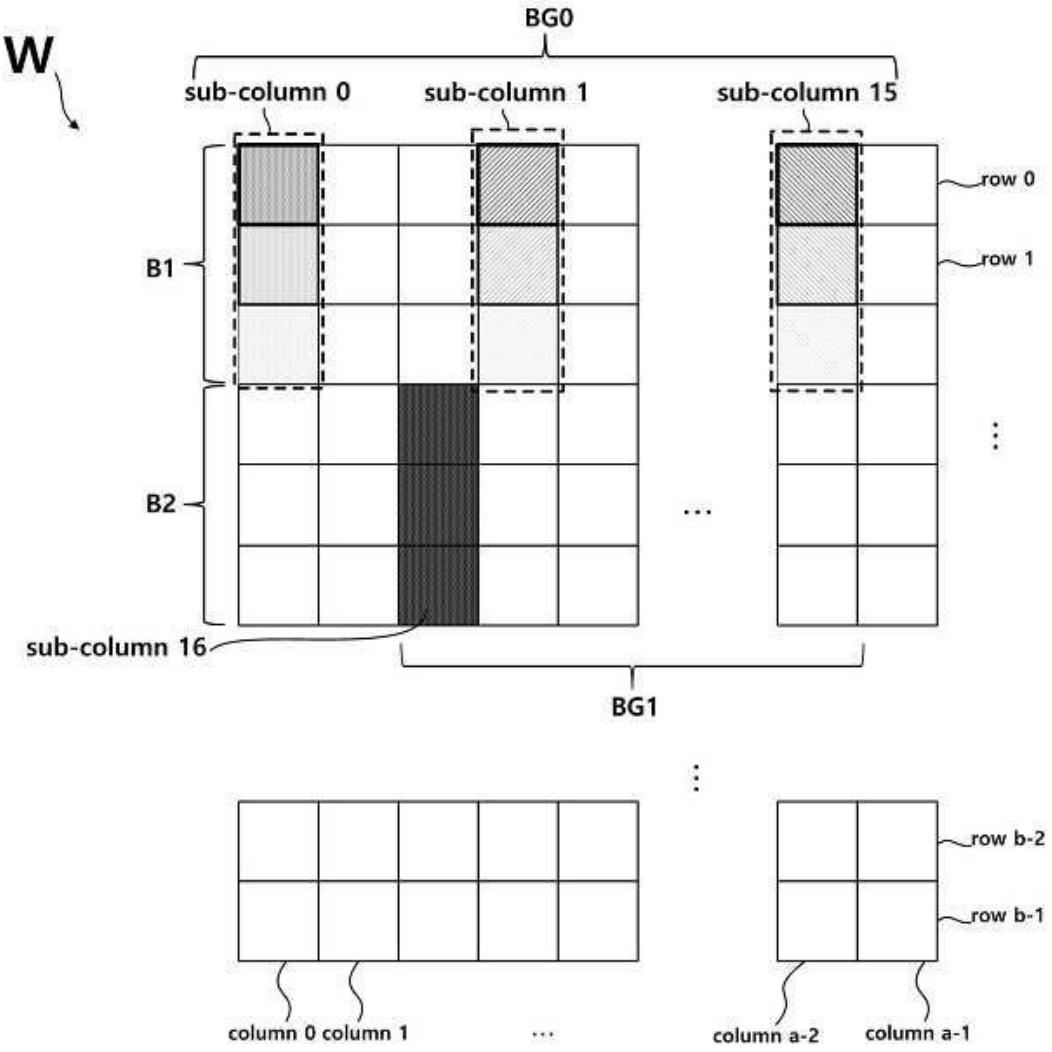
Value: 값 서브 어레이

도면

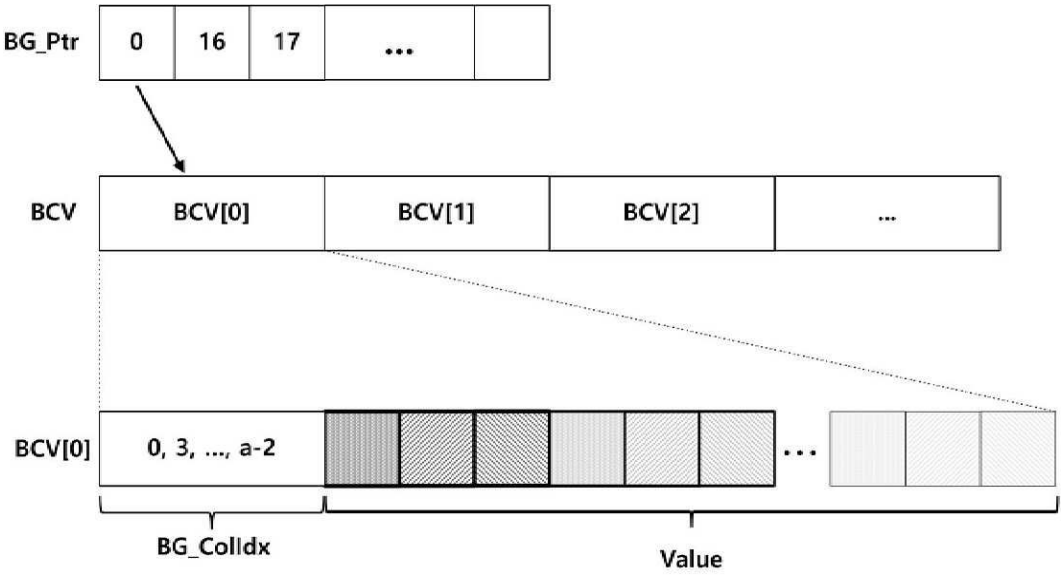
도면1


시작
가중치 행렬을 블록 단위로 프루닝(pruning)하여 압축
S100
입력 벡터를 DRAM 모듈 내의 SRAM 버퍼에 타일링
S200
압축된 상기 가중치 행렬을 상기 DRAM 어레이의 동일한 행에 배치
S300
종료

도면2


W
BG0
sub-column 0
sub-column 1
sub-column 15
row 0
row 1
B1
B2
sub-column 16
BG1
row b-2
row b-1
column 0
column 1
column a-2
column a-1

도면3


BG_Ptr
0
16
17
BCV
BCV[0]
BCV[1]
BCV[2]
BCV[0]
0, 3, ..., a-2
BG_ColIdx
Value

도면4