(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0045463
(43) 공개일자 2023년04월04일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2023.01) *G06F 7/49* (2006.01)
G06F 7/544 (2017.01) *G06N 3/08* (2023.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06F 7/49 (2013.01)
(21) 출원번호 10-2021-0128354
(22) 출원일자 2021년09월28일
심사청구일자 2021년09월28일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
김지훈
서울특별시 서초구 서초대로38길 12, 102동 603호 (서초동, 마제스타시티, 힐스테이트 서리풀)
김하나
경기도 의정부시 가능로 110(의정부동)
(74) 대리인
리앤목특허법인

전체 청구항 수 : 총 15 항

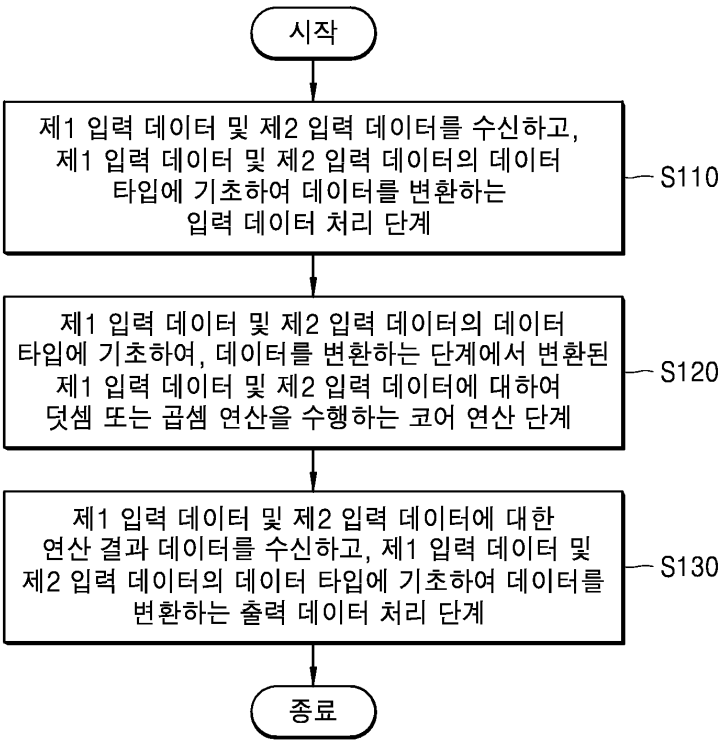
(54) 발명의 명칭 **신경망 가속기를 위한 연산 장치, 연산 방법 및 상기 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램**

(57) 요 약

신경망 가속기를 위한 연산 장치, 연산 방법 및 상기 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램을 위하여, 본 발명에 따른 연산 장치는, 신경망 가속기의 내부에 구현되는 적어도 하나의 프로세싱 엘리먼트(Processing element, PE)를 포함하는 연산 장치에 있어서, 상기 프로세싱 엘리먼트는 인티저(Integer) 타입

(뒷면에 계속)

대 표 도 - 도4



및 포짓(Posit) 타입의 두 가지 데이터 타입에 대한 연산을 제공하고, 상기 프로세싱 엘리먼트는, 제1 입력 데이터 및 제2 입력 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환하는 입력 데이터 처리부, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여, 상기 입력 데이터 처리부에서 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대하여 덧셈 또는 곱셈 연산을 수행하는 코어 연산부, 및 상기 코어 연산부로부터 상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대한 연산 결과 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여 데이터를 변환하는 출력 데이터 처리부를 포함한다.

(52) CPC특허분류
G06F 7/544 (2018.05)
G06N 3/08 (2023.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126247
과제번호	2020-0-01308-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	차세대지능형반도체기술개발(설계)(R&D)
연구과제명	딥러닝 초소형 코어 어레이 기반 지능형 모바일 프로세서
기 여 율	1/1
과제수행기관명	한국전자통신연구원
연구기간	2021.01.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

신경망 가속기의 내부에 구현되는 적어도 하나의 프로세싱 엘리먼트(Processing element, PE)를 포함하는 연산 장치에 있어서,

상기 프로세싱 엘리먼트는,

제1 입력 데이터 및 제2 입력 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환하는 입력 데이터 처리부;

상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여, 상기 입력 데이터 처리부에서 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대하여 덧셈 또는 곱셈 연산을 수행하는 코어 연산부; 및

상기 코어 연산부로부터 상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대한 연산 결과 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여 데이터를 변환하는 출력 데이터 처리부를 포함하고,

인티저(Integer) 타입 및 포짓(Posit) 타입의 두 가지 데이터 타입에 대한 연산을 제공하는, 연산 장치.

청구항 2

제1항에 있어서,

상기 프로세싱 엘리먼트는 상기 신경망 가속기의 적어도 하나의 레이어에 대하여 상기 레이어의 특성에 기초하여 인티저(Integer) 타입 또는 포짓(Posit) 타입 각각에 대한 연산을 제공하고,

상기 레이어의 특성은 데이터 타입을 포함하는, 연산 장치.

청구항 3

제1항에 있어서,

상기 입력 데이터 처리부는, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 제1 입력 데이터 및 상기 제2 입력 데이터로부터 포짓 데이터(Posit data)를 추출하고,

상기 포짓 데이터는 포짓 타입인 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 이진수 형식의 8 비트 숫자인, 연산 장치.

청구항 4

제1항에 있어서,

상기 코어 연산부는,

상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 미리 결정된 지수 비트를 이용하여 상기 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 연결부를 0으로 연결하고,

상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 비트수와 상기 미리 결정된 지수 비트를 이용하여 상기 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 가수부를 생성하는, 연산 장치.

청구항 5

제4항에 있어서,

상기 연결부의 비트수는 상기 미리 결정된 지수 비트에 숫자 2를 더한 비트수이고,

상기 가수부의 비트수는 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 비트수에서 상기 미리 결정된 지수 비트와 숫자 2를 차감한 비트수인, 연산 장치.

청구항 6

제4항에 있어서,

상기 코어 연산부는,

상기 가수부의 덧셈 또는 곱셈 연산을 수행하는 인티저 연산기;

를 더 포함하는, 연산 장치.

청구항 7

제1항에 있어서,

상기 출력 데이터 처리부는, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 연산 결과 데이터를 포짓 타입으로 변환하는, 연산 장치.

청구항 8

컴퓨팅 장치에 의해 수행되는 신경망 가속기를 위한 연산 방법에 있어서,

제1 입력 데이터 및 제2 입력 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환하는 입력 데이터 처리 단계;

상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여, 상기 데이터를 변환하는 단계에서 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대하여 덧셈 또는 곱셈 연산을 수행하는 코어 연산 단계; 및

상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대한 연산 결과 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여 데이터를 변환하는 출력 데이터 처리 단계를 포함하고,

인티저(Integer) 타입 및 포짓(Posit) 타입의 두 가지 데이터 타입에 대한 연산을 제공하는, 연산 방법.

청구항 9

제8항에 있어서,

상기 연산 방법은 상기 신경망 가속기의 적어도 하나의 레이어에 대하여 상기 레이어의 특성에 기초하여 인티저(Integer) 타입 또는 포짓(Posit) 타입 각각에 대한 연산을 제공하고,

상기 레이어의 특성은 데이터 타입을 포함하는, 연산 방법.

청구항 10

제8항에 있어서,

상기 입력 데이터 처리 단계는 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 제1 입력 데이터 및 상기 제2 입력 데이터로부터 포짓 데이터(Posit data)를 추출하는 단계이고,

상기 포짓 데이터는 포짓 타입인 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 이진수 형식의 8 비트 숫자인, 연산 방법.

청구항 11

제8항에 있어서,

상기 코어 연산 단계는,

상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 미리 결정된 지수 비트를 이용하여 상기 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 연결부를 0으로 연결하는 단계; 및

상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 제1 입력 데이터 및

상기 제2 입력 데이터의 비트수와 상기 미리 결정된 지수 비트를 이용하여 상기 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 가수부를 생성하는 단계를 포함하는, 연산 방법.

청구항 12

제11항에 있어서,

상기 연결부의 비트수는 상기 미리 결정된 지수 비트에 숫자 2를 더한 비트수이고,

상기 가수부의 비트수는 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 비트수에서 상기 미리 결정된 지수 비트와 숫자 2를 차감한 비트수인, 연산 방법.

청구항 13

제11항에 있어서,

상기 코어 연산 단계는,

상기 가수부의 덧셈 또는 곱셈 연산을 수행하는 단계를 더 포함하는, 연산 방법.

청구항 14

제8항에 있어서,

상기 출력 데이터 처리 단계는 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 연산 결과 데이터를 포짓 타입으로 변환하는 단계인, 연산 방법.

청구항 15

컴퓨터를 이용하여 제8항 내지 제14항 중 어느 한 항의 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램.

발명의 설명

기 술 분 야

[0001] 본 발명의 실시예들은, 신경망 가속기를 위한 연산 장치, 연산 방법 및 상기 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램에 관한 것이다.

배 경 기 술

[0002] 최근 많은 국가에서 과학, 산업, 국가 안보와 같은 다양한 분야를 강화하기 위해 고성능 컴퓨팅에 많은 관심을 가지고 있다. 고성능 컴퓨팅에는 기존의 과학 컴퓨팅뿐만 아니라 인공지능 학습 및 데이터 분석 방법의 응용 프로그램도 포함된다.

[0003] 이러한 고성능 컴퓨팅 디바이스에서 실시간 객체 탐지의 수요가 증가하면서 제한적인 하드웨어 자원에서 DNN(Deep Neural Network) 연산을 요구하고 있으며, 객체 탐지의 정확성을 높이기 위해 작은 물체에 대한 객체 탐지의 수요가 늘어나면서 연산 파라미터의 수와 연산량이 늘어나고 있다.

[0004] 따라서, 하드웨어 경량화의 한가지 방법으로 정확도를 최대한 떨어뜨리지 않으면서 낮은 정밀도로 구현하여 memory bandwidth, footprint, 및 연산의 latency를 줄이기 위한 많은 노력을 하고 있다.

[0005] 그러나, 기존에 널리쓰이는 양자화(quantization) 방식만으로는 16-bit 미만의 정밀도로 구현할 경우 정확도가 떨어지는 문제가 있어서 일정 수준 이하의 정밀도의 구현이 어려웠다.

발명의 내용

해결하려는 과제

[0006] 본 발명이 해결하고자 하는 과제는 전술한 문제점을 극복하기 위한 것으로서, 하드웨어 이용률을 향상시킬 수 있는 신경망 가속기를 위한 연산 장치, 연산 방법 및 상기 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램을 제공하는 것이다. 그러나 이러한 과제는 예시적인 것으로, 이에 의해 본 발명의 범위가 한정되는 것

은 아니다.

과제의 해결 수단

[0007] 상술한 기술적 과제들을 달성하기 위한 기술적 수단으로서, 본 발명의 일 측면에 따른 연산 장치는, 신경망 가속기의 내부에 구현되는 적어도 하나의 프로세싱 엘리먼트(Processing element, PE)를 포함하는 연산 장치에 있어서, 상기 프로세싱 엘리먼트는 인티저(Integer) 타입 및 포짓(Posit) 타입의 두 가지 데이터 타입에 대한 연산을 제공하고, 상기 프로세싱 엘리먼트는, 제1 입력 데이터 및 제2 입력 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환하는 입력 데이터 처리부, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여, 상기 입력 데이터 처리부에서 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대하여 덧셈 또는 곱셈 연산을 수행하는 코어 연산부, 및 상기 코어 연산부로부터 상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대한 연산 결과 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여 데이터를 변환하는 출력 데이터 처리부를 포함한다.

[0008] 일 예에 따르면, 상기 프로세싱 엘리먼트는 상기 신경망 가속기의 적어도 하나의 레이어에 대하여 상기 레이어의 특성에 기초하여 인티저(Integer) 타입 또는 포짓(Posit) 타입 각각에 대한 연산을 제공하고, 상기 레이어의 특성은 데이터 타입을 포함할 수 있다.

[0009] 다른 예에 따르면, 상기 입력 데이터 처리부는, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 제1 입력 데이터 및 상기 제2 입력 데이터로부터 포짓 데이터(Posit data)를 추출하고, 상기 포짓 데이터는 포짓 타입인 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 이진수 형식의 8 비트 숫자일 수 있다.

[0010] 또 다른 예에 따르면, 상기 코어 연산부는, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 미리 결정된 지수 비트를 이용하여 상기 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 연결부를 0으로 연결하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 비트수와 상기 미리 결정된 지수 비트를 이용하여 상기 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 가수부를 생성할 수 있다.

[0011] 또 다른 예에 따르면, 상기 연결부의 비트수는 상기 미리 결정된 지수 비트에 숫자 2를 더한 비트수이고, 상기 가수부의 비트수는 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 비트수에서 상기 미리 결정된 지수 비트와 숫자 2를 차감한 비트수일 수 있다.

[0012] 또 다른 예에 따르면, 상기 코어 연산부는, 상기 가수부의 덧셈 또는 곱셈 연산을 수행하는 인티저 연산기를 더 포함할 수 있다.

[0013] 또 다른 예에 따르면, 상기 출력 데이터 처리부는, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 연산 결과 데이터를 포짓 타입으로 변환할 수 있다.

[0014] 상술한 기술적 과제들을 달성하기 위한 기술적 수단으로서, 본 발명의 일 측면에 따른 연산 방법은, 컴퓨팅 장치에 의해 수행되는 신경망 가속기를 위한 연산 방법에 있어서, 상기 연산 방법은 인티저(Integer) 타입 및 포짓(Posit) 타입의 두 가지 데이터 타입에 대한 연산을 제공하고, 상기 연산 방법은, 제1 입력 데이터 및 제2 입력 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환하는 입력 데이터 처리 단계, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여, 상기 데이터를 변환하는 단계에서 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대하여 덧셈 또는 곱셈 연산을 수행하는 코어 연산 단계, 및 상기 제1 입력 데이터 및 상기 제2 입력 데이터에 대한 연산 결과 데이터를 수신하고, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 상기 데이터 타입에 기초하여 데이터를 변환하는 출력 데이터 처리 단계를 포함한다.

[0015] 일 예에 따르면, 상기 연산 방법은 상기 신경망 가속기의 적어도 하나의 레이어에 대하여 상기 레이어의 특성에 기초하여 인티저(Integer) 타입 또는 포짓(Posit) 타입 각각에 대한 연산을 제공하고, 상기 레이어의 특성은 데이터 타입을 포함할 수 있다.

[0016] 다른 예에 따르면, 상기 입력 데이터 처리 단계는 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 제1 입력 데이터 및 상기 제2 입력 데이터로부터 포짓 데이터(Posit data)를 추출하는 단계이고, 상기 포짓 데이터는 포짓 타입인 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 이진수 형식

의 8 비트 숫자일 수 있다.

[0017] 또 다른 예에 따르면, 상기 코어 연산 단계는, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 미리 결정된 지수 비트를 이용하여 상기 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 연결부를 0으로 연결하는 단계, 및 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 비트수와 상기 미리 결정된 지수 비트를 이용하여 상기 변환된 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 가수부를 생성하는 단계를 포함할 수 있다.

[0018] 또 다른 예에 따르면, 상기 연결부의 비트수는 상기 미리 결정된 지수 비트에 숫자 2를 더한 비트수이고, 상기 가수부의 비트수는 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 비트수에서 상기 미리 결정된 지수 비트와 숫자 2를 차감한 비트수일 수 있다.

[0019] 또 다른 예에 따르면, 상기 코어 연산 단계는, 상기 가수부의 덧셈 또는 곱셈 연산을 수행하는 단계를 더 포함할 수 있다.

[0020] 또 다른 예에 따르면, 상기 출력 데이터 처리 단계는 상기 제1 입력 데이터 및 상기 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 상기 연산 결과 데이터를 포짓 타입으로 변환하는 단계일 수 있다.

[0021] 상술한 기술적 과제들을 달성하기 위한 기술적 수단으로서, 컴퓨터를 이용하여 상술한 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램이 제공된다.

[0022] 전술한 것 외의 다른 측면, 특징, 이점은 이하의 발명을 실시하기 위한 구체적인 내용, 청구범위 및 도면으로부터 명확해질 것이다.

발명의 효과

[0023] 상기한 바와 같이 이루어진 본 발명의 일 실시예에 따르면, 하드웨어 이용률을 향상시킬 수 있는 신경망 가속기를 위한 연산 장치, 연산 방법 및 상기 방법을 실행시키기 위하여 기록매체에 저장된 컴퓨터 프로그램을 구현할 수 있다. 물론 이러한 효과에 의해 본 발명의 범위가 한정되는 것은 아니다.

도면의 간단한 설명

[0024] 도 1은 본 발명의 일 실시예에 따른 연산 장치가 포함하는 프로세싱 엘리먼트(PE)를 개략적으로 설명하는 도면이다.

도 2는 본 발명의 일 실시예에 따른 연산 장치에서 연산되는 포짓 타입의 데이터 타입을 설명하기 위한 도면이다.

도 3은 본 발명의 일 실시예에 따른 연산 장치가 포함하는 프로세싱 엘리먼트(PE)의 구성을 개략적으로 도시하는 도면이다.

도 4는 본 발명의 일 실시예에 따른 연산 방법을 설명하기 위한 순서도이다.

도 5는 본 발명의 일 실시예에 따른 인티저(Integer) 타입의 데이터 타입에 대한 연산 방법을 설명하기 위한 도면이다.

도 6은 본 발명의 일 실시예에 따른 포짓(Posit) 타입의 데이터 타입에 대한 연산 방법을 설명하기 위한 도면이다.

도 7 및 도 8은 본 발명의 일 실시예에 따른 연산 장치의 성능을 실험한 실험 결과의 예시들이다.

발명을 실시하기 위한 구체적인 내용

[0025] 아래에서는 첨부한 도면을 참조하여 본 개시가 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 다양한 실시예들을 상세히 설명한다. 그러나 본 개시의 기술적 사상은 다양한 형태로 변형되어 구현될 수 있으므로 본 명세서에서 설명하는 실시예들로 제한되지 않는다. 본 명세서에 개시된 실시예들을 설명함에 있어서 관련된 공지 기술을 구체적으로 설명하는 것이 본 개시의 기술적 사상의 요지를 흐릴 수 있다고 판단되는 경우 그 공지 기술에 대한 구체적인 설명을 생략한다. 동일하거나 유사한 구성요소는 동일한 참조 번호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다.

[0026] 본 명세서에서 어떤 요소가 다른 요소와 "연결"되어 있다고 기술될 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라 그 중간에 다른 요소를 사이에 두고 "간접적으로 연결"되어 있는 경우도 포함한다. 어떤 요소가 다른 요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 요소 외에 또 다른 요소를 배제하는 것이 아니라 또 다른 요소를 더 포함할 수 있는 것을 의미한다.

[0027] 일부 실시예들은 기능적인 블록 구성들 및 다양한 처리 단계들로 설명될 수 있다. 이러한 기능 블록들의 일부 또는 전부는 특정 기능을 실행하는 다양한 개수의 하드웨어 및/또는 소프트웨어 구성들로 구현될 수 있다. 예를 들어, 본 개시의 기능 블록들은 하나 이상의 마이크로프로세서들에 의해 구현되거나, 소정의 기능을 위한 회로 구성들에 의해 구현될 수 있다. 본 개시의 기능 블록들은 다양한 프로그래밍 또는 스크립팅 언어로 구현될 수 있다. 본 개시의 기능 블록들은 하나 이상의 프로세서들에서 실행되는 알고리즘으로 구현될 수 있다. 본 개시의 기능 블록이 수행하는 기능은 복수의 기능 블록에 의해 수행되거나, 본 개시에서 복수의 기능 블록이 수행하는 기능들은 하나의 기능 블록에 의해 수행될 수도 있다. 또한, 본 개시는 전자적인 환경 설정, 신호 처리, 및/또는 데이터 처리 등을 위하여 종래 기술을 채용할 수 있다.

[0028] 도 1은 본 발명의 일 실시예에 따른 연산 장치가 포함하는 프로세싱 엘리먼트(PE)를 개략적으로 설명하는 도면이고, 도 2는 본 발명의 일 실시예에 따른 연산 장치에서 연산되는 포짓 타입의 데이터 타입을 설명하기 위한 도면이다. 또한, 도 3은 본 발명의 일 실시예에 따른 연산 장치가 포함하는 프로세싱 엘리먼트(PE)의 구성을 개략적으로 도시하는 도면이다.

[0029] 먼저, 도 1을 참조하면, 본 발명의 일 실시예에 따른 연산 장치는 적어도 하나의 프로세싱 엘리먼트(100)를 포함한다. 예를 들어, 적어도 하나의 프로세싱 엘리먼트(100)는 신경망 가속기의 내부에 구현될 수 있다. 하지만 본 발명은 이에 한정되지 않으며, 연산 장치는 다른 구성요소를 더 포함할 수도 있고 일부 구성요소가 생략될 수도 있다. 연산 장치의 일부 구성요소는 복수의 장치로 분리될 수도 있고, 복수개의 구성요소들이 하나의 장치로 병합될 수도 있다.

[0030] 본 발명의 일 실시예에 따른 프로세싱 엘리먼트(100)는 프로세서 및 메모리를 포함할 수 있다. 하지만 본 발명은 이에 한정되지 않으며, 프로세서 및 메모리는 연산 장치에 포함될 수 있다.

[0031] 프로세서는 메모리에 저장된 데이터의 검색, 추출, 연산, 읽기, 쓰기 등의 동작을 수행할 수 있다. 프로세서는 마이크로프로세서(Microprocessor), 중앙처리장치(Central Processing Unit: CPU), 프로세서 코어(Processor Core), 멀티프로세서(Multiprocessor), ASIC(Application-Specific Integrated Circuit), FPGA(Field Programmable Gate Array) 등의 처리 장치를 포함할 수 있으나, 본 발명은 이에 한정되지 않는다.

[0032] 메모리는 컴퓨터에서 판독 가능한 기록 매체로서, RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 또한, 메모리에는 연산 장치를 제어하기 위한 프로그램 코드 및 데이터가 일시적 또는 영구적으로 저장될 수 있다.

[0033] 도 1에 도시된 바와 같이, 본 발명의 일 실시예에 따른 연산 장치는 적어도 하나의 프로세싱 엘리먼트(PE)를 포함할 수 있다. 본 발명의 일 실시예에 따른 프로세싱 엘리먼트(100)는 적어도 하나의 인티저 연산기(135)를 포함할 수 있다. 예를 들어, 도 1에 도시된 바와 같이, 인티저 연산기(135)는 프로세싱 엘리먼트(100)에 입력되는 데이터에 대하여 곱셈 또는 덧셈 연산을 수행할 수 있다. 예컨대, 프로세싱 엘리먼트(100)는 복수의 인티저 연산기(135)에 의하여 연산된 결과값을 출력할 수 있다.

[0034] 본 발명의 일 실시예에 따른 프로세싱 엘리먼트(100)는 인티저(Integer) 타입 및 포짓(Posit) 타입의 두 가지 데이터 타입에 대한 연산을 제공할 수 있다. 예를 들어, 도 1에 도시된 바와 같이, 본 발명의 일 실시예에 따른 프로세싱 엘리먼트(100)는 인티저(Integer) 타입 및 포짓(Posit) 타입의 두 가지 데이터 타입에 대한 연산을 제공하는 하이브리드 PE(Processing Element) 일 수 있다.

[0035] 두 가지 데이터 타입에 대하여, 인티저(Integer) 타입은 획일적(uniform) 범위의 수 표현에 적합하며 이를 지원하는 하드웨어가 단순한 반면, 동적(dynamic) 범위에서 수를 표현할 때는 낮은 정밀도의 표현이 불가능하다는 단점이 존재한다. 또한, 포짓(Posit) 타입은 동적(dynamic) 범위의 수를 표현할 때 효율이 좋으며 따라서 낮은 정밀도의 수 표현으로도 높은 정확도의 표현이 가능하다. 그러나, 포짓(Posit) 타입의 경우 정확도가 떨어지고 지원하는 하드웨어가 복잡하다는 단점이 존재한다.

[0036] 따라서, 본 발명에 따른 연산 장치는 두 데이터 타입의 장단점을 고려하여 단점을 보완할 수 있도록 두 가지 데이터 타입을 신경망 가속기의 레이어(layer) 특성에 따라 적용한 하이브리드(Hybrid) 데이터 타입을 제공하여

하드웨어 이용률(utility)을 높일 수 있는 효과가 있다.

[0037] 도 2를 참조하면, 본 발명의 일 실시예에 따른 연산 장치에서 연산되는 포짓 타입의 데이터 타입은 사인 비트(sign bit), 레짐 비트(regime bits), 익스포넌트 비트(exponent bits), 및 프랙션 비트(fraction bits)의 이진수 형식으로 표현되는 데이터일 수 있다. 예를 들어, 도 2에 도시된 바와 같이, 부동 소수점(floating point) 타입 형식의 데이터는 포짓 타입의 형식으로 표현될 수 있다.

[0038] 예를 들어, 본 발명의 일 실시예에 따른 포짓 타입의 데이터는 8비트의 데이터일 수 있다. 예컨대, 사인 비트(sign bit), 레짐 비트(regime bits), 익스포넌트 비트(exponent bits), 및 프랙션 비트(fraction bits)가 도 2에 도시된 바와 같이 연결되어 총 8비트의 데이터가 될 수 있다. 또한, 익스포넌트 비트(exponent bits)의 자리수(es)는 미리 설정된 특정값일 수 있다.

[0039] 도 3을 참조하면, 프로세싱 엘리먼트(100)는 입력 데이터 처리부(110), 코어 연산부(130), 및 출력 데이터 처리부(150)를 포함한다.

[0040] 프로세싱 엘리먼트(100)와 이에 포함된 모듈들 또는 부들은, 전적으로 하드웨어일 수 있고, 또는 부분적으로 하드웨어이고 부분적으로 소프트웨어인 측면을 가질 수 있다. 예컨대, 프로세싱 엘리먼트(100)와 이에 포함된 각 모듈 또는 부는 특정 형식 및 내용의 데이터를 처리하거나 전자통신 방식으로 주고받기 위한 하드웨어 및 이에 관련된 소프트웨어를 통칭할 수 있다. 본 명세서에서 "모듈" 또는 "부" 등의 용어는 하드웨어 및 해당 하드웨어에 의해 구동되는 소프트웨어의 조합을 지칭하는 것으로 해석될 수 있다. 예컨대, 하드웨어는 CPU 또는 다른 프로세서(processor)를 포함하는 데이터 처리 기기일 수 있다. 또한, 하드웨어에 의해 구동되는 소프트웨어는 실행중인 프로세스, 객체(object), 실행파일(executable file), 실행 스레드(thread of execution), 프로그램(program) 등을 지칭할 수 있다.

[0041] 또한, 프로세싱 엘리먼트(100)를 구성하는 각각의 요소는 반드시 서로 물리적으로 구분되는 별개의 장치를 지칭하는 것으로 한정되는 것은 아니다. 즉, 도 3의 입력 데이터 처리부(110), 코어 연산부(130), 및 출력 데이터 처리부(150) 등은 프로세싱 엘리먼트(100)에 의해 수행되는 동작에 따라 기능적으로 구분한 것일 뿐, 반드시 서로 독립적으로 구분되어야 하는 것이 아니다. 물론, 실시예에 따라서는 이들은 서로 물리적으로 구분되는 별개의 장치로 구현되는 것도 가능하다.

[0042] 이하에서는 상술한 것과 같은 구성을 취할 수 있는 프로세싱 엘리먼트(100)의 각 부들의 동작 등에 대해, 연산 방법과 함께 설명한다.

[0043] 도 4는 본 발명의 일 실시예에 따른 연산 방법을 설명하기 위한 순서도이고, 도 5는 본 발명의 일 실시예에 따른 인티저(Integer) 타입의 데이터 타입에 대한 연산 방법을 설명하기 위한 도면이다. 또한, 도 6은 본 발명의 일 실시예에 따른 포짓(Posit) 타입의 데이터 타입에 대한 연산 방법을 설명하기 위한 도면이다.

[0044] 먼저, 도 4를 참조하면, 단계 S110에서, 프로세싱 엘리먼트(100)는 제1 입력 데이터 및 제2 입력 데이터를 수신할 수 있다. 예를 들어, 프로세싱 엘리먼트(100)는 복수의 입력 데이터를 수신할 수 있다. 예컨대, 제1 입력 데이터 및 제2 입력 데이터는 인티저(Integer) 타입의 데이터일 수 있다. 이 경우, 인티저 타입의 제1 입력 데이터 및 제2 입력 데이터는 16비트 또는 8비트의 데이터일 수 있다. 또는, 제1 입력 데이터 및 제2 입력 데이터는 포짓(Posit) 타입의 데이터일 수 있다. 이 경우, 포짓 타입의 제1 입력 데이터 및 제2 입력 데이터는 8비트의 데이터일 수 있다.

[0045] 또한, 프로세싱 엘리먼트(100)는 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환할 수 있다. 예를 들어, 프로세싱 엘리먼트(100)는 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우 제1 입력 데이터 및 제2 입력 데이터를 변환할 수 있다. 예컨대, 프로세싱 엘리먼트(100)는 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우 제1 입력 데이터 및 제2 입력 데이터를 변환하여 제1 입력 데이터 및 제2 입력 데이터의 덧셈 또는 곱셈 연산을 수행할 수 있다.

[0046] 본 발명의 일 실시예에서 프로세싱 엘리먼트(100)는 신경망 가속기의 적어도 하나의 레이어에 대하여 레이어의 특성에 기초하여 인티저(Integer) 타입 또는 포짓(Posit) 타입 각각에 대한 연산을 제공할 수 있다. 예컨대, 레이어의 특성은 데이터 타입을 포함할 수 있다.

[0047] 예컨대, 신경망 가속기에 포함된 전체 레이어를 기본적으로 구성하고자하는 정밀도(예를 들어, 16비트)의 인티저(integer) 타입으로 설정하고 신경망의 각 레이어를 돌아가면서 하나씩 포짓(posit) 타입으로 바꾸어 신경망의 정확도를 측정할 수 있다. 이 경우, 전체 레이어의 결과 중에서 포짓(posit) 타입을 적용했을 때 정확도가

월등히 높아지는 레이어를 선정하여 해당 레이어는 포짓(posit) 타입으로 구성하고, 나머지 레이어는 인티저(integer) 타입으로 구성할 수 있다.

[0048] 본 발명의 일 실시예에 따른 연산 장치는 신경망 가속기에 포함된 레이어의 데이터 타입에 기초하여 인티저(Integer) 타입 또는 포짓(Posit) 타입에 대한 연산을 제공할 수 있다.

[0049] 본 발명의 일 실시예에서 프로세싱 엘리먼트(100)는, 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 제1 입력 데이터 및 제2 입력 데이터로부터 포짓 데이터(Posit data)를 추출할 수 있다. 예컨대, 포짓 데이터는 포짓 타입인 제1 입력 데이터 및 제2 입력 데이터의 이진수 형식의 8 비트 숫자일 수 있다.

[0050] 단계 S120에서, 프로세싱 엘리먼트(100)는 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입에 기초하여, 단계 S110에서 변환된 제1 입력 데이터 및 제2 입력 데이터에 대하여 덧셈 또는 곱셈 연산을 수행할 수 있다.

[0051] 도 5를 함께 참조하면, 본 발명의 일 실시예에서 프로세싱 엘리먼트(100)는 N비트의 인티저 타입인 제1 입력 데이터 및 제2 입력 데이터를 수신할 수 있다. 그리고, 프로세싱 엘리먼트(100)는 코어 연산부(130)의 인티저 연산기(135)를 이용하여 제1 입력 데이터 및 제2 입력 데이터의 덧셈 또는 곱셈 연산을 수행할 수 있다. 예컨대, N비트는 16비트 또는 8비트일 수 있다. 그리고, 프로세싱 엘리먼트(100)는 제1 입력 데이터 및 제2 입력 데이터의 연산 결과를 출력 할 수 있다.

[0052] 도 6을 함께 참조하면, 본 발명의 일 실시예에서 프로세싱 엘리먼트(100)는 N비트의 포짓 타입인 제1 입력 데이터 및 제2 입력 데이터를 수신할 수 있다.

[0053] 그리고, 입력 데이터 처리부(110)는 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환할 수 있다. 예를 들어, 포짓 타입인 제1 입력 데이터 및 제2 입력 데이터로부터 포짓 데이터(Posit data)를 추출할 수 있다. 예컨대, 도 2에 도시된 바와 같이, 포짓 데이터는 사인 비트(sign bit), 레짐 비트(regime bits), 익스포넌트 비트(exponent bits), 및 프랙션 비트(fraction bits)를 포함하는 이진수 형식의 8 비트 숫자일 수 있다.

[0054] 그리고, 코어 연산부(130)는 제1 입력 데이터 및 제2 입력 데이터의 포짓 데이터를 인티저 연산기(135)의 데이터 타입에 맞도록 데이터 형식을 변경할 수 있다. 예를 들어, 도 6에 도시된 바와 같이, 코어 연산부(130)는 제1 입력 데이터 및 제2 입력 데이터의 포짓 데이터에 대하여 미리 결정된 지수 비트(ES)를 이용하여 연결부(concatenate)를 0으로 연결할 수 있다. 예컨대, 미리 결정된 지수 비트(ES)는 도 2에 도시된 익스포넌트 비트(exponent bits)의 자리수(es)를 나타낼 수 있다. 또한, 본 발명의 일 실시예에서 연결부(c1, c2)의 비트수는 미리 결정된 지수 비트(ES)에 숫자 2를 더한 비트수일 수 있다.

[0055] <수학식1>

[0056] $$Cbit = ES + 2$$

[0057] 여기서, Cbit는 연결부(c1, c2)의 비트수이고, ES는 미리 결정된 지수 비트를 나타낸다.

[0058] 또한, 코어 연산부(130)는 제1 입력 데이터 및 제2 입력 데이터의 비트수(N)와 미리 결정된 지수 비트(ES)를 이용하여, 제1 입력 데이터 및 제2 입력 데이터의 가수부(Mantissa)를 생성할 수 있다. 예컨대, 코어 연산부(130)는 제1 입력 데이터 및 제2 입력 데이터의 포짓 데이터를 기초로 인티저 연산기(135)의 데이터 타입에 맞도록 데이터 형식을 변경할 수 있다. 또한, 코어 연산부(130)는 제1 입력 데이터 및 제2 입력 데이터의 포짓 데이터를 기초로 제1 입력 데이터 및 제2 입력 데이터의 비트수(N)와 미리 결정된 지수 비트(ES)를 이용하여, 제1 입력 데이터 및 제2 입력 데이터의 가수부(m1, m2)를 생성할 수 있다.

[0059] 또한, 본 발명의 일 실시예에서 가수부(m1, m2)의 비트수는 제1 입력 데이터 및 제2 입력 데이터의 비트수(N)에서 미리 결정된 지수 비트(ES)와 숫자 2를 차감한 비트수일 수 있다.

[0060] <수학식2>

[0061] $$Mbit = N - ES - 2$$

[0062] 여기서, Mbit는 가수부(m1, m2)의 비트수를 나타내고, N은 제1 입력 데이터 및 제2 입력 데이터의 비트수를 나타내고, ES는 미리 결정된 지수 비트를 나타낸다. 예컨대, N은 8비트 일 수 있다.

[0063] 예를 들어, 미리 결정된 지수 비트(ES)가 3인 경우, 연결부(c1, c2)의 비트수는 5비트이고, 가수부(m1, m2)의

비트수는 3비트일 수 있다.

[0064] 본 발명의 일 실시예에서 프로세싱 엘리먼트(100)는 가수부(m1, m2)의 덧셈 또는 곱셈 연산을 수행하는 인티저 연산기(135)를 더 포함할 수 있다. 예를 들어, 도 6에 도시된 바와 같이, 프로세싱 엘리먼트(100)는 인티저 연산기(135)를 이용하여 가수부(m1, m2)의 덧셈 또는 곱셈 연산을 수행할 수 있다.

[0065] 그리고, 출력 데이터 처리부(150)는 제1 입력 데이터 및 제2 입력 데이터에 대한 연산 결과 데이터를 포짓 타입으로 변환하여 출력할 수 있다.

[0066] 단계 S130에서, 프로세싱 엘리먼트(100)는 제1 입력 데이터 및 제2 입력 데이터에 대한 연산 결과 데이터를 수신할 수 있다. 또한, 프로세싱 엘리먼트(100)는 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환할 수 있다.

[0067] 예를 들어, 본 발명의 일 실시예에서 프로세싱 엘리먼트(100)는 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입이 포짓 타입인 경우, 연산 결과 데이터를 포짓 타입으로 변환할 수 있다.

[0068] 도 7 및 도 8은 본 발명의 일 실시예에 따른 연산 장치의 성능을 실험한 실험 결과의 예시들이다.

[0069] 도 7 및 도 8을 참조하면, 기존의 양자화 방식인 부동소수점 32비트(FLOAT32), 인티저 16비트(INT16), 및 인티저 8비트(INT8)로 신경망에 따라 약 10%에서 많게는 44%의 정확도 감소가 일어난 반면 기존 16비트 인티저(Integer) 타입 대비 본 발명에 따른 하이브리드 16비트 및 8비트(Hybrid[16,8])의 경우 특정 레이어에서 정밀도를 반으로(16비트에서 8비트로) 줄이면서 신경망 정확도는 기존의 인티저 8비트에 비하여 감소하지 않은 결과를 도시하고 있다.

[0070] 기존 포짓(Posit) 타입을 지원하는 하드웨어의 경우, 포짓 타입의 데이터를 처리하는 경우에만 사용하였으나, 본 발명에 따른 연산 장치는 포짓(Posit) 타입을 처리하는 프로세싱 엘리먼트(PE)에서 인티저(integer) 연산이 가능한 인티저 연산기를 추가함으로써 연산의 속도를 높이고 하드웨어 이용률을 향상시킬 수 있는 효과가 있다.

[0071] 본 발명에 따르면, 낮은 정밀도의 수 표현으로 memory bandwidth, footprint, 연산의 latency를 줄여 하드웨어를 경량화할 수 있고, 낮은 정밀도로 구현하였음에도 불구하고 기존의 양자화 방식 대비 정확도를 향상 시킬 수 있다.

[0072] 기존의 신경망 가속기가 16-bit 미만의 정밀도로 구성될 경우 정확도가 현저히 낮아지는 문제로 인하여 낮은 정밀도로 구현이 어려웠던 반면에 본 발명에 따른 “Hybrid” 데이터 타입의 도입으로 정확도가 떨어지지 않으면서 8-bit 정밀도의 하드웨어 구성이 가능해졌다.

[0073] 또한, Posit이 fixed-point보다 정확도가 높다고 해서 posit만으로 구현할 경우, dynamic range가 아닌 경우에 오히려 정확도는 떨어진다. 따라서 이를 극복하기 위해 본 발명에서는, dynamic range를 가지는 layer에서는 Posit 타입을 사용하여 정확도 하락을 최소화하면서도 uniform한 특성을 가지는 layer에는 fixed-point로 구현함으로써 posit 적용으로 인한 속도 저하를 최소화 하여, 정확도와 속도 측면에서 모두 이점이 있는 8-bit의 precision 구성이 가능해졌다.

[0074] 본 발명은 기존의 NPU 모델에서 사용하지 않았던 posit을 fixed-point(integer)타입과 함께 사용하여 Hybrid[m,n](m-bit integer, n-bit posit) 데이터 타입을 제공한다.

[0075] 본 발명에서 제공하는 Hybrid[m,n] 타입을 NPU에 적용할 경우, posit을 지원하는 PE(Processing Element)와 integer(fixed-point)를 지원하는 PE를 모두 사용할 수 있다. 기존 모델의 경우, posit을 지원하는 PE는 posit을 연산할 때만 사용하기 때문에 하드웨어 utility(이용률)이 현저히 떨어지는 문제가 발생했다. 그러나, 본 발명에서는 Posit 연산기 내부의 “mantissa” 연산을 위한 integer(fixed-point) 연산기를 이용하여 integer 타입의 연산도 가능하다. 이렇게 Posit 연산기 내부의 Integer 연산기를 활용하여 하드웨어 이용률과 연산 속도를 향상 시킬 수 있는 효과가 있다.

[0077] 이상 설명된 다양한 실시예들은 예시적이며, 서로 구별되어 독립적으로 실시되어야 하는 것은 아니다. 본 명세서에서 설명된 실시예들은 서로 조합된 형태로 실시될 수 있다.

[0078] 이상 설명된 다양한 실시예들은 컴퓨터 상에서 다양한 구성요소를 통하여 실행될 수 있는 컴퓨터 프로그램의 형태로 구현될 수 있으며, 이와 같은 컴퓨터 프로그램은 컴퓨터로 판독 가능한 매체에 기록될 수 있다. 이때, 매

체는 컴퓨터로 실행 가능한 프로그램을 계속 저장하거나, 실행 또는 다운로드를 위해 임시 저장하는 것일 수도 있다. 또한, 매체는 단일 또는 수개 하드웨어가 결합된 형태의 다양한 기록수단 또는 저장수단일 수 있는데, 어떤 컴퓨터 시스템에 직접 접속되는 매체에 한정되지 않고, 네트워크 상에 분산 존재하는 것일 수도 있다. 매체의 예시로는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM 및 DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical medium), 및 ROM, RAM, 플래시 메모리 등을 포함하여 프로그램 명령어가 저장되도록 구성된 것이 있을 수 있다. 또한, 다른 매체의 예시로, 애플리케이션을 유통하는 앱 스토어나 기타 다양한 소프트웨어를 공급 내지 유통하는 사이트, 서버 등에서 관리하는 기록매체 내지 저장매체도 들 수 있다.

[0079] 본 명세서에서, "부", "모듈" 등은 프로세서 또는 회로와 같은 하드웨어 구성(hardware component), 및/또는 프로세서와 같은 하드웨어 구성에 의해 실행되는 소프트웨어 구성(software component)일 수 있다. 예를 들면, "부", "모듈" 등은 소프트웨어 구성 요소들, 객체 지향 소프트웨어 구성 요소들, 클래스 구성 요소들 및 태스크 구성 요소들과 같은 구성 요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들에 의해 구현될 수 있다.

[0080] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0081] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0082] 100: 프로세싱 엘리먼트

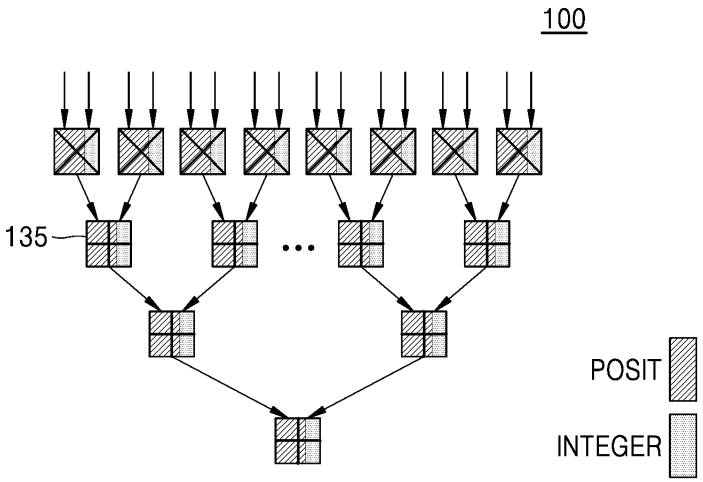
110: 입력 데이터 처리부

130: 코어 연산부

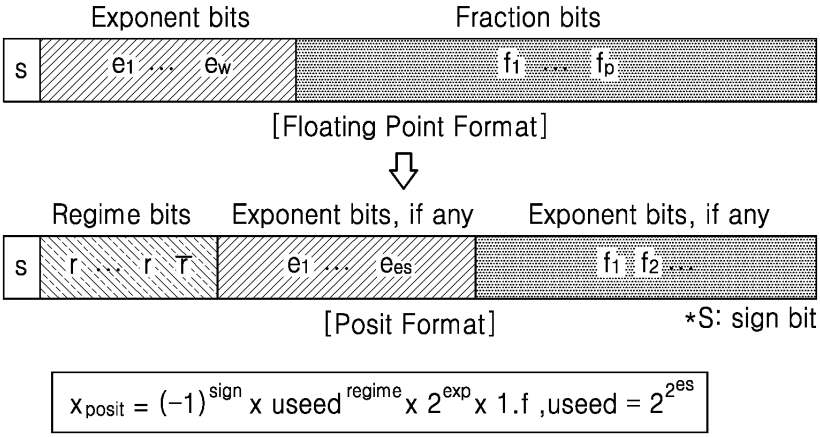
150: 출력 데이터 처리부

도면

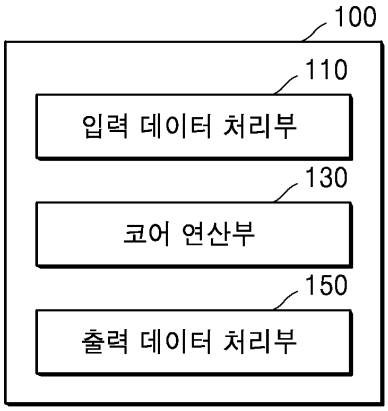
도면1



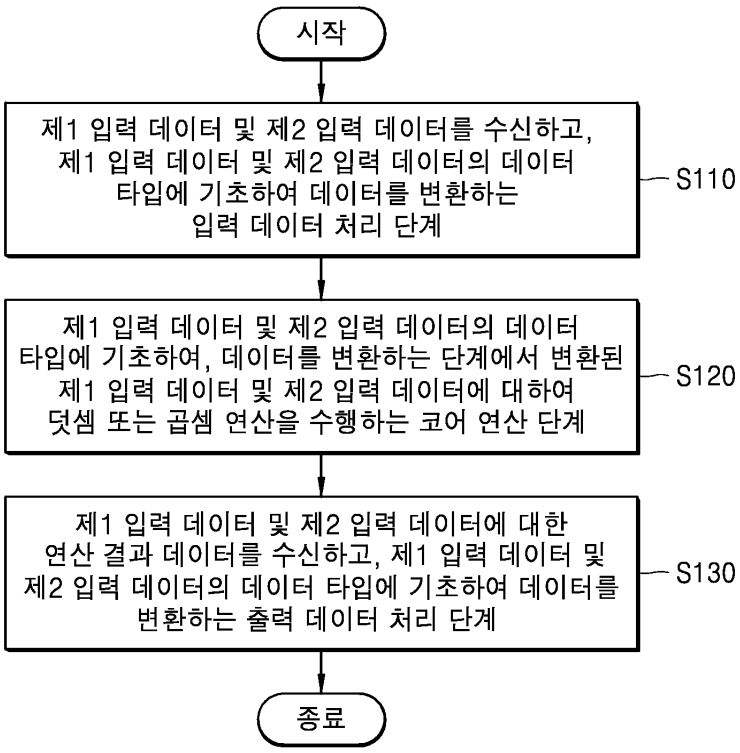
도면2


Exponent bits
Fraction bits
s
e1 ... ew
f1 ... fp
[Floating Point Format]
Regime bits
Exponent bits, if any
Exponent bits, if any
s
r ... r r̄
e1 ... ees
f1 f2 ...
[Posit Format]
*S: sign bit
xposit = (−1)sign x useed regime x 2exp x 1.f ,useed = 2^2es

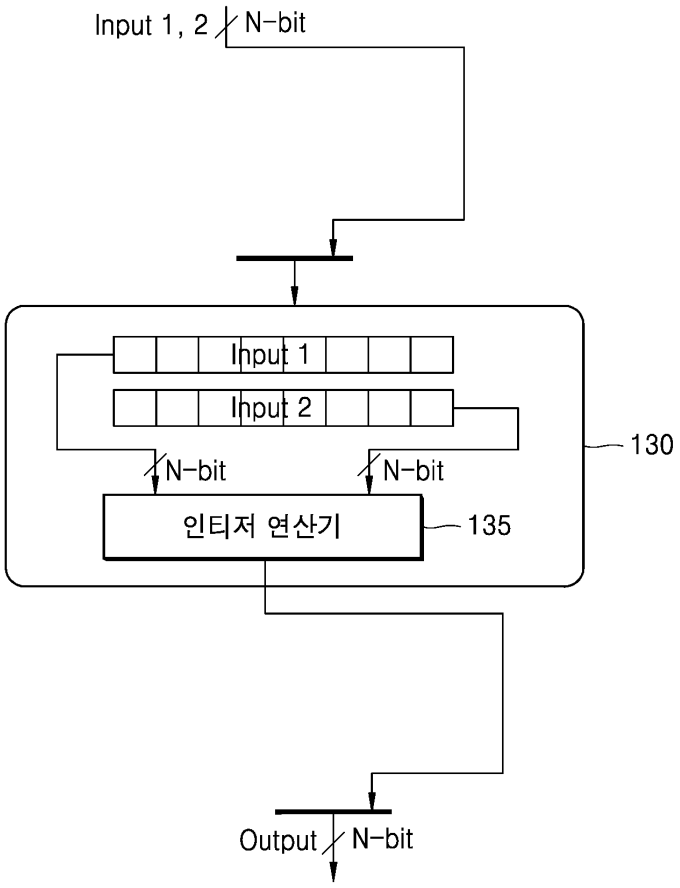
도면3


100
110
입력 데이터 처리부
130
코어 연산부
150
출력 데이터 처리부

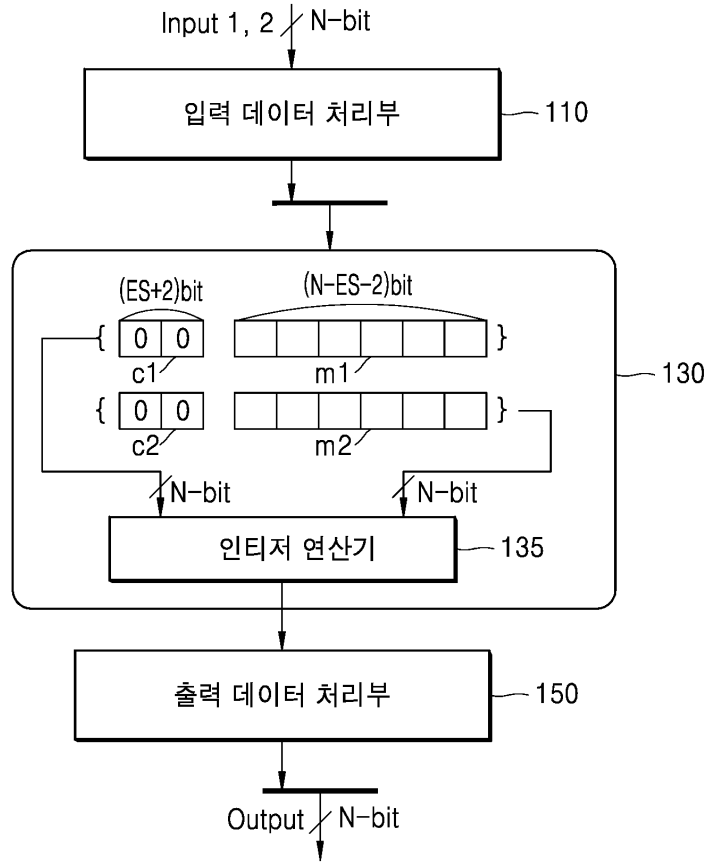
도면4


시작
제1 입력 데이터 및 제2 입력 데이터를 수신하고, 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환하는 입력 데이터 처리 단계
S110
제1 입력 데이터 및 제2 입력 데이터의 데이터 타입에 기초하여, 데이터를 변환하는 단계에서 변환된 제1 입력 데이터 및 제2 입력 데이터에 대하여 덧셈 또는 곱셈 연산을 수행하는 코어 연산 단계
S120
제1 입력 데이터 및 제2 입력 데이터에 대한 연산 결과 데이터를 수신하고, 제1 입력 데이터 및 제2 입력 데이터의 데이터 타입에 기초하여 데이터를 변환하는 출력 데이터 처리 단계
S130
종료

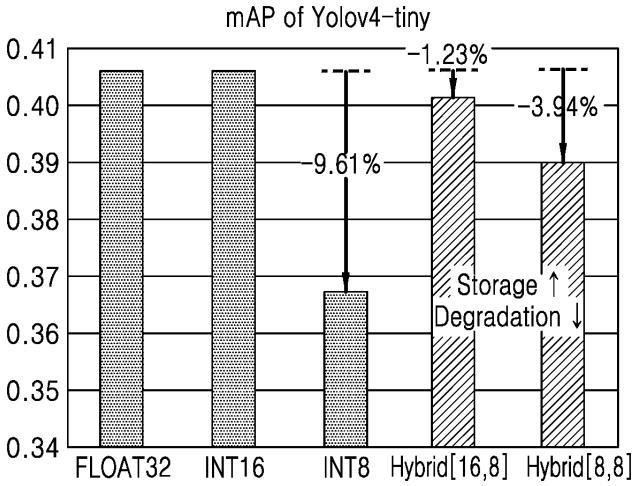
도면5


Input 1, 2
N-bit
Input 1
Input 2
N-bit
N-bit
130
인티저 연산기
135
Output
N-bit

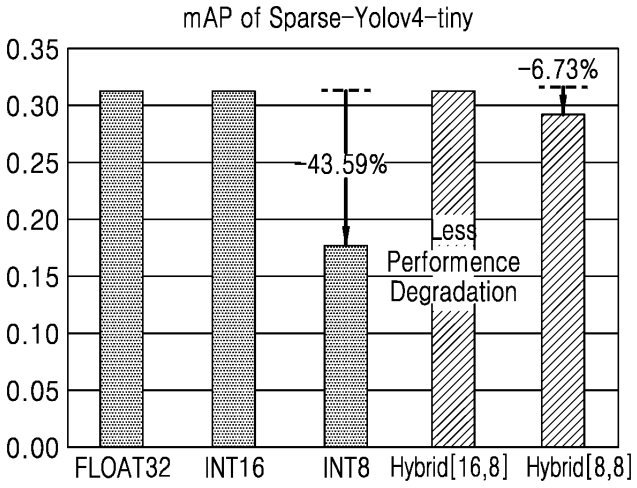
도면6


Input 1, 2
N-bit
입력 데이터 처리부
110
(ES+2)bit
(N-ES-2)bit
0 0
c1
m1
0 0
c2
m2
130
N-bit
N-bit
인티저 연산기
135
출력 데이터 처리부
150
Output
N-bit

도면7


mAP of Yolov4-tiny
0.41
0.40
0.39
0.38
0.37
0.36
0.35
0.34
-1.23%
-3.94%
-9.61%
Storage ↑
Degradation ↓
FLOAT32
INT16
INT8
Hybrid[16,8]
Hybrid[8,8]

도면8


mAP of Sparse-Yolov4-tiny
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-6.73%
-43.59%
Less
Performence
Degradation
FLOAT32
INT16
INT8
Hybrid[16,8]
Hybrid[8,8]