



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0163575
(43) 공개일자 2022년12월12일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2006.01) G06N 3/04 (2006.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06N 3/04 (2013.01)
(21) 출원번호 10-2021-0071874
(22) 출원일자 2021년06월03일
심사청구일자 2021년06월03일

(71) 출원인
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
송진호
서울특별시 용산구 원효로 216, 101동 1901호
김보길
서울특별시 서대문구 신촌로1안길 16, 301호
이성재
서울특별시 광진구 아차산로70길 62, 307동 304호
(74) 대리인
권성현, 유광철, 백두진, 강일신, 김정연

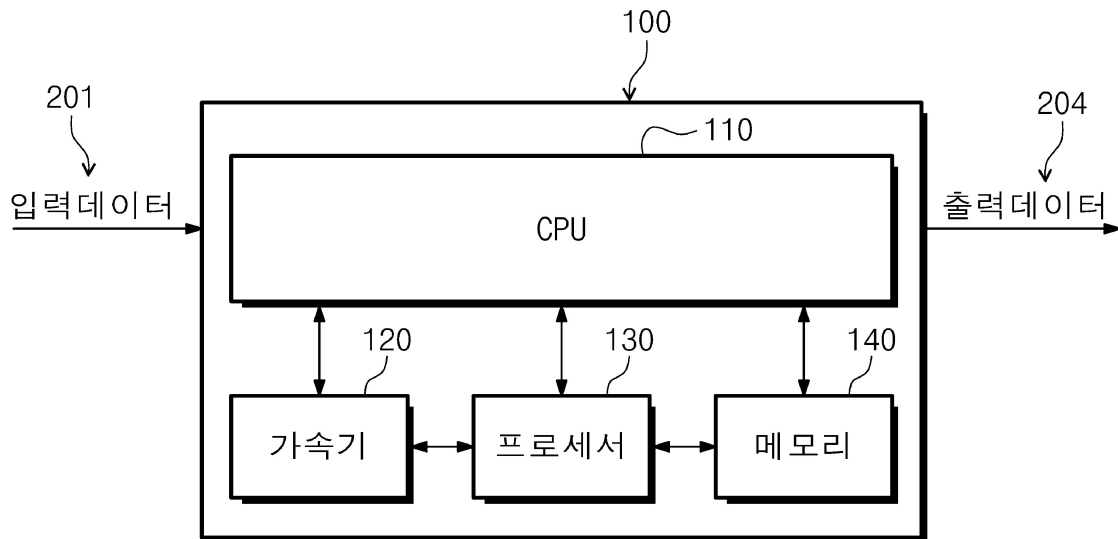
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 신경망 연산 장치, 신경망 연산 방법 및 신경망 연산 방법을 실행시키도록 기록매체에 저장된 컴퓨터 프로그램

(57) 요약

개시된 발명의 일 실시예에 따른 복수개의 계층으로 구성된 인공신경망 모델에 기초하여 입력 데이터에 대한 연산을 수행하는 신경망 연산 장치는, 상기 입력 데이터에 대하여 전처리 연산을 수행하여 전처리 데이터를 출력하도록 구성되는 CPU; 및 상기 전처리 데이터에 대하여 상기 인공신경망 모델의 제1 계층에 의한 연산을 수행하여 중간 데이터를 출력하도록 구성되는 가속기;를 포함하고, 상기 CPU는, 상기 중간 데이터에 대하여 상기 인공신경망 모델의 제2 계층에 의한 연산을 수행하여 출력 데이터를 출력하도록 구성될 수 있다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	1711116053
과제번호	10080674
부처명	과학기술정보통신부
과제관리(전문)기관명	한국산업기술평가관리원
연구사업명	전자정보디바이스산업원천기술개발(R&D)
연구과제명	재구성 가능한 인공신경망 가속기 구현 및 인스트럭션셋 기술개발
기 여 율	1/2
과제수행기관명	연세대학교 산학협력단
연구기간	2020.01.01 ~ 2020.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711120090
과제번호	2020-0-01847-001
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	비대면 · 인공지능 사회를 위한 반도체 시스템 융합혁신기술 개발
기 여 율	1/2
과제수행기관명	한국과학기술원
연구기간	2020.07.01 ~ 2027.12.31

명세서

청구범위

청구항 1

복수개의 계층으로 구성된 인공신경망 모델에 기초하여 입력 데이터에 대한 연산을 수행하는 신경망 연산 장치에 있어서,

상기 입력 데이터에 대하여 전처리 연산을 수행하여 전처리 데이터를 출력하도록 구성되는 CPU; 및

상기 전처리 데이터에 대하여 상기 인공신경망 모델의 제1 계층에 의한 연산을 수행하여 중간 데이터를 출력하도록 구성되는 가속기;를 포함하고,

상기 CPU는,

상기 중간 데이터에 대하여 상기 인공신경망 모델의 제2 계층에 의한 연산을 수행하여 출력 데이터를 출력하도록 구성되는 신경망 연산 장치.

청구항 2

제1항에 있어서,

상기 CPU는,

상기 입력 데이터에 포함된 복수개의 배치에 대한 전처리 연산을 수행하여 각 배치의 전처리 데이터를 출력하도록 구성되고,

상기 가속기는,

상기 인공신경망 모델에 기초하여 상기 각 배치의 전처리 데이터에 대한 연산을 수행하여 각 배치의 중간 데이터를 출력하도록 구성되는, 신경망 연산 장치.

청구항 3

제2항에 있어서,

상기 CPU는,

상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 의한 연산을 처리하는 동안, 제1 배치의 중간 데이터에 대하여 상기 제2 계층에 의한 연산을 중첩적으로 처리하도록 구성되는, 신경망 연산 장치.

청구항 4

제2항에 있어서,

상기 CPU는,

상기 가속기가 제1 배치의 전처리 데이터에 대하여 상기 제1 계층에 의한 연산을 처리하는 동안, 제2 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 상기 제2 배치의 전처리 데이터를 출력하도록 구성되는, 신경망 연산 장치.

청구항 5

제2항에 있어서,

상기 CPU는,

상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 의한 연산을 처리하는 동안, 제1 배치의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 상기 제1 배치의 출력 데이터를 출력하도록 구성되는, 신경망 연산 장치.

청구항 6

제2항에 있어서,

상기 CPU는,

상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 대한 연산을 처리하는 동안, 제1 배치의 중간 데이터에 대하여 상기 제2 계층에 의한 연산을 중첩적으로 처리하고, 상기 제1 배치의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 상기 제1 배치의 출력 데이터를 출력하고, 제3 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 상기 제3 배치의 전처리 데이터를 출력하도록 구성되는, 신경망 연산 장치.

청구항 7

제1항에 있어서,

상기 가속기의 성능 및 상기 CPU의 성능에 기초하여, 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하도록 구성되는 프로세서;를 더 포함하는 신경망 연산 장치.

청구항 8

제7항에 있어서,

상기 프로세서는,

상기 CPU의 연산속도 정보, 상기 가속기의 연산속도 정보 및 총 연산 시간에 대한 전처리 연산 시간과 후처리 연산 시간의 비율 정보에 기초하여 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하도록 구성되는, 신경망 연산 장치.

청구항 9

제7항에 있어서,

상기 프로세서는:

상기 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상인지 여부를 판단하고; 그리고

상기 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상이면, 상기 신경망 연산 장치에 공급되는 전압 및 주파수 중 적어도 하나를 감소시키도록 구성되는, 신경망 연산 장치.

청구항 10

제1항에 있어서,

상기 가속기의 성능 및 상기 CPU의 성능에 기초하여, 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하도록 구성되는 프로세서;를 더 포함하고,

상기 가속기는,

상기 인공신경망 모델에 기초하여 상기 입력 데이터에 포함된 복수개의 배치에 대한 연산을 수행하여 각 배치의 중간 데이터를 출력하도록 구성되고,

상기 CPU는:

상기 입력 데이터에 포함된 복수개의 배치에 대한 전처리 연산을 수행하여 각 배치의 전처리 데이터를 출력하도록 구성되고; 그리고

상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 의한 연산을 처리하는 동안, 제1 배치의 중간 데이터에 대하여 상기 제2 계층에 의한 연산을 중첩적으로 처리하고, 상기 제1 배치의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 상기 제1 배치의 출력 데이터를 출력하고, 제3 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 제3 배치에 대한 전처리 데이터를 출력하도록 구성되고,

상기 프로세서는:

상기 CPU의 연산속도 정보, 상기 가속기의 연산속도 정보 및 총 연산 시간에 대한 전처리 연산 시간과 후처리

연산 시간의 비율 정보에 기초하여 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하도록 구성되고;

상기 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상인지 여부를 판단하고; 그리고

상기 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상이면, 상기 신경망 연산 장치에 공급되는 전압 및 주파수 중 적어도 하나를 감소시키도록 구성되는, 신경망 연산 장치.

청구항 11

복수개의 계층으로 구성된 인공신경망 모델에 기초하여 입력 데이터에 대한 연산을 수행하는 신경망 연산 방법에 있어서,

CPU에 의해, 상기 입력 데이터에 대하여 전처리 연산을 수행하여 전처리 데이터를 출력하는 단계;

가속기에 의해, 상기 전처리 데이터에 대하여 상기 인공신경망 모델의 제1 계층에 의한 연산을 수행하여 중간 데이터를 출력하는 단계; 및

상기 CPU에 의해, 상기 중간 데이터에 대하여 상기 인공신경망 모델의 제2 계층에 의한 연산을 수행하여 출력 데이터를 출력하는 단계;를 포함하는 신경망 연산 방법.

청구항 12

제11항에 있어서,

상기 전처리 데이터를 출력하는 단계는,

상기 CPU에 의해, 상기 입력 데이터에 포함된 복수개의 배치에 대한 전처리 연산을 수행하여 각 배치의 전처리 데이터를 출력하는 단계;를 포함하고,

상기 중간 데이터를 출력하는 단계는,

상기 가속기에 의해, 상기 인공신경망 모델에 기초하여 상기 각 배치의 전처리 데이터에 대한 연산을 수행하여 각 배치의 중간 데이터를 출력하는 단계;를 포함하는, 신경망 연산 방법.

청구항 13

제12항에 있어서,

상기 출력 데이터를 출력하는 단계는,

상기 CPU에 의해, 상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 의한 연산을 처리하는 동안, 제1 배치의 중간 데이터에 대하여 상기 제2 계층에 의한 연산을 중첩적으로 처리하는 단계;를 포함하는 신경망 연산 방법.

청구항 14

제12항에 있어서,

상기 전처리 데이터를 출력하는 단계는,

상기 CPU에 의해, 상기 가속기가 제1 배치의 상기 제1 계층에 대한 연산을 처리하는 동안, 제2 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 상기 제2 배치의 전처리 데이터를 출력하는 단계;를 포함하는 신경망 연산 방법.

청구항 15

제12항에 있어서,

상기 출력 데이터를 출력하는 단계는,

상기 CPU에 의해, 상기 가속기가 제2 배치의 상기 제1 계층에 대한 연산을 처리하는 동안, 제1 배치의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 상기 제1 배치의 출력 데이터를 출력하는 단계;를 포함하는 신경망 연산 방법.

청구항 16

제12항에 있어서,

상기 출력 데이터를 출력하는 단계는,

상기 CPU에 의해, 상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 대한 연산을 처리하는 동안, 제1 배치의 중간 데이터에 대하여 상기 제2 계층에 의한 연산을 중첩적으로 처리하는 단계; 및

상기 CPU에 의해, 상기 가속기가 상기 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 대한 연산을 처리하는 동안, 상기 제1 배치의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 상기 제1 배치의 데이터의 출력 데이터를 출력하는 단계;를 포함하고,

상기 전처리 데이터를 출력하는 단계는,

상기 CPU에 의해, 상기 가속기가 상기 제2 배치의 상기 제1 계층에 대한 연산을 처리하는 동안, 제3 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 상기 제3 배치의 전처리 데이터를 출력하는 단계;를 포함하는 신경망 연산 방법.

청구항 17

제11항에 있어서,

프로세서에 의해, 상기 가속기의 성능 및 상기 CPU의 성능에 기초하여, 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하는 단계;를 더 포함하는 신경망 연산 방법.

청구항 18

제17항에 있어서,

상기 제1 계층 및 상기 제2 계층으로 분류하는 단계는,

상기 프로세서에 의해, 상기 CPU의 연산속도 정보, 상기 가속기의 연산속도 정보 및 총 연산 시간에 대한 전처리 연산 시간과 후처리 연산 시간의 비율 정보에 기초하여 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하는 단계;를 포함하는 신경망 연산 방법.

청구항 19

제17항에 있어서,

상기 프로세서에 의해, 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상인지 여부를 판단하는 단계; 및

상기 프로세서에 의해, 상기 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상이면, 상기 신경망 연산 장치에 공급되는 전압 및 주파수 중 적어도 하나를 감소시키는 단계;를 더 포함하는 신경망 연산 방법.

청구항 20

제11항 내지 제19항 중 어느 한 항의 신경망 연산 방법을 실행시키도록 컴퓨터로 판독 가능한 기록매체에 저장된 컴퓨터 프로그램.

발명의 설명

기술 분야

[0001] 본 발명은 인공신경망 모델을 구성하는 복수개의 계층 중 일부의 계층에 의한 연산을 가속기가 아닌 CPU가 수행하도록 하여 신경망 연산의 효율을 개선할 수 있는 신경망 연산 장치 및 신경망 연산 방법에 관한 것이다.

배경 기술

[0002] 인공신경망이 발전함에 따라 이를 신속하게 처리하기 위한 인공신경망 가속기가 장착된 하드웨어가 다수 개발되고 있다.

[0003] 인공신경망 가속 하드웨어는 먼저 인공신경망 가속기가 장착된 하드웨어에서 호스트인 CPU가 인공신경망이 들어

오면 전처리 과정을 수행한 후, 가속기에 처리 명령을 내리고 가속기는 이를 처리하는 역할을 수행한 후에 다시 호스트인 CPU에 처리된 데이터를 넘겨주고, CPU가 후처리 과정을 수행한다.

[0004] 전술한 동작 방법은 심층 신경망(Deep Neural Network; DNN)의 처리 과정에서 CPU와 가속기중 하나의 프로세서만 특정 시점에 활용한다는 특징을 가지게 되어 하드웨어가 낼 수 있는 최대의 성능을 내지 못하는 단점이 존재한다.

[0005] 또한, 두 개의 서로 다른 프로세서 중 하나만 사용함으로써 사용하지 않는 프로세서가 불필요한 파워를 소비하는 문제를 일으켜 하드웨어 전체의 에너지 효율을 떨어뜨린다는 문제가 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 CPU 및 가속기의 휴지기를 최소화하고, CPU 및 가속기 사이의 통신으로 인한 통신 오버헤드가 발생하지 않도록 하여 하드웨어의 최대 성능을 이끌어 내고 연산 속도를 향상시킬 수 있는 합성곱 연산 장치, 합성곱 연산 방법 및 컴퓨터 프로그램을 제공하기 위한 것이다.

[0007] 또한, 본 발명은 CPU 및 가속기의 휴지기를 최소화하여 불필요한 전력 소비를 줄이고 에너지 효율을 개선할 수 있는 합성곱 연산 장치, 합성곱 연산 방법 및 컴퓨터 프로그램을 제공하기 위한 것이다.

과제의 해결 수단

[0008] 개시된 발명의 일 측면에 따른 복수개의 계층으로 구성된 인공신경망 모델에 기초하여 입력 데이터에 대한 연산을 수행하는 신경망 연산 장치는, 상기 입력 데이터에 대하여 전처리 연산을 수행하여 전처리 데이터를 출력하도록 구성되는 CPU; 및 상기 전처리 데이터에 대하여 상기 인공신경망 모델의 제1 계층에 의한 연산을 수행하여 중간 데이터를 출력하도록 구성되는 가속기;를 포함하고, 상기 CPU는, 상기 중간 데이터에 대하여 상기 인공신경망 모델의 제2 계층에 의한 연산을 수행하여 출력 데이터를 출력하도록 구성될 수 있다.

[0009] 또한, 상기 CPU는, 상기 입력 데이터에 포함된 복수개의 배치에 대한 전처리 연산을 수행하여 각 배치의 전처리 데이터를 출력하도록 구성되고, 상기 가속기는, 상기 인공신경망 모델에 기초하여 상기 각 배치의 전처리 데이터에 대한 연산을 수행하여 각 배치의 중간 데이터를 출력하도록 구성될 수 있다.

[0010] 또한, 상기 CPU는, 상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 의한 연산을 처리하는 동안, 제1 배치의 중간 데이터에 대하여 상기 제2 계층에 의한 연산을 중첩적으로 처리하도록 구성될 수 있다.

[0011] 또한, 상기 CPU는, 상기 가속기가 제1 배치의 전처리 데이터에 대하여 상기 제1 계층에 의한 연산을 처리하는 동안, 제2 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 상기 제2 배치의 전처리 데이터를 출력하도록 구성될 수 있다.

[0012] 또한, 상기 CPU는, 상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 의한 연산을 처리하는 동안, 제1 배치의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 상기 제1 배치의 출력 데이터를 출력하도록 구성될 수 있다.

[0013] 또한, 상기 CPU는, 상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 대한 연산을 처리하는 동안, 제1 배치의 중간 데이터에 대하여 상기 제2 계층에 의한 연산을 중첩적으로 처리하고, 상기 제1 배치의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 상기 제1 배치의 출력 데이터를 출력하고, 제3 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 상기 제3 배치의 전처리 데이터를 출력하도록 구성될 수 있다.

[0014] 또한, 상기 가속기의 성능 및 상기 CPU의 성능에 기초하여, 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하도록 구성되는 프로세서;를 더 포함할 수 있다.

[0015] 또한, 상기 프로세서는, 상기 CPU의 연산속도 정보, 상기 가속기의 연산속도 정보 및 총 연산 시간에 대한 전처리 연산 시간과 후처리 연산 시간의 비율 정보에 기초하여 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하도록 구성될 수 있다.

[0016] 또한, 상기 프로세서는: 상기 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상인지 여부를 판단하고; 그리고 상기 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상이면, 상기 신경망 연산 장치에 공급되는 전압 및

주파수 중 적어도 하나를 감소시키도록 구성될 수 있다.

- [0017] 개시된 발명의 일 측면에 따른 복수개의 계층으로 구성된 인공신경망 모델에 기초하여 입력 데이터에 대한 연산을 수행하는 신경망 연산 방법은, CPU에 의해, 상기 입력 데이터에 대하여 전처리 연산을 수행하여 전처리 데이터를 출력하는 단계; 가속기에 의해, 상기 전처리 데이터에 대하여 상기 인공신경망 모델의 제1 계층에 의한 연산을 수행하여 중간 데이터를 출력하는 단계; 및 상기 CPU에 의해, 상기 중간 데이터에 대하여 상기 인공신경망 모델의 제2 계층에 의한 연산을 수행하여 출력 데이터를 출력하는 단계;를 포함할 수 있다.
- [0018] 또한, 상기 전처리 데이터를 출력하는 단계는, 상기 CPU에 의해, 상기 입력 데이터에 포함된 복수개의 배치에 대한 전처리 연산을 수행하여 각 배치의 전처리 데이터를 출력하는 단계;를 포함하고, 상기 중간 데이터를 출력하는 단계는, 상기 가속기에 의해, 상기 인공신경망 모델에 기초하여 상기 각 배치의 전처리 데이터에 대한 연산을 수행하여 각 배치의 중간 데이터를 출력하는 단계;를 포함할 수 있다.
- [0019] 또한, 상기 출력 데이터를 출력하는 단계는, 상기 CPU에 의해, 상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 의한 연산을 처리하는 동안, 제1 배치의 중간 데이터에 대하여 상기 제2 계층에 의한 연산을 중첩적으로 처리하는 단계;를 포함할 수 있다.
- [0020] 또한, 상기 전처리 데이터를 출력하는 단계는, 상기 CPU에 의해, 상기 가속기가 제1 배치의 상기 제1 계층에 대한 연산을 처리하는 동안, 제2 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 상기 제2 배치의 전처리 데이터를 출력하는 단계;를 포함할 수 있다.
- [0021] 또한, 상기 출력 데이터를 출력하는 단계는, 상기 CPU에 의해, 상기 가속기가 제2 배치의 상기 제1 계층에 대한 연산을 처리하는 동안, 제1 배치의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 상기 제1 배치의 출력 데이터를 출력하는 단계;를 포함할 수 있다.
- [0022] 또한, 상기 출력 데이터를 출력하는 단계는, 상기 CPU에 의해, 상기 가속기가 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 대한 연산을 처리하는 동안, 제1 배치의 중간 데이터에 대하여 상기 제2 계층에 의한 연산을 중첩적으로 처리하는 단계; 및 상기 CPU에 의해, 상기 가속기가 상기 제2 배치의 전처리 데이터에 대하여 상기 제1 계층에 대한 연산을 처리하는 동안, 상기 제1 배치의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 상기 제1 배치의 데이터의 출력 데이터를 출력하는 단계;를 포함하고, 상기 전처리 데이터를 출력하는 단계는, 상기 CPU에 의해, 상기 가속기가 상기 제2 배치의 상기 제1 계층에 대한 연산을 처리하는 동안, 제3 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 상기 제3 배치의 전처리 데이터를 출력하는 단계;를 포함할 수 있다.
- [0023] 또한, 프로세서에 의해, 상기 가속기의 성능 및 상기 CPU의 성능에 기초하여, 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하는 단계;를 더 포함할 수 있다.
- [0024] 또한, 상기 제1 계층 및 상기 제2 계층으로 분류하는 단계는, 상기 프로세서에 의해, 상기 CPU의 연산속도 정보, 상기 가속기의 연산속도 정보 및 총 연산 시간에 대한 전처리 연산 시간과 후처리 연산 시간의 비율 정보에 기초하여 상기 복수개의 계층을 상기 제1 계층 및 상기 제2 계층으로 분류하는 단계;를 포함할 수 있다.
- [0025] 또한, 상기 프로세서에 의해, 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상인지 여부를 판단하는 단계; 및 상기 프로세서에 의해, 상기 신경망 연산 장치의 연산 속도가 기준 처리 속도 이상이면, 상기 신경망 연산 장치에 공급되는 전압 및 주파수 중 적어도 하나를 감소시키는 단계;를 더 포함할 수 있다.
- [0026] 개시된 발명의 일 측면에 따른 컴퓨터 프로그램은, 상기 신경망 연산 방법을 실행시키도록 컴퓨터로 판독 가능한 기록매체에 저장될 수 있다.

발명의 효과

- [0027] 개시된 발명의 일 측면에 따르면, CPU 및 가속기의 휴지기를 최소화하고, CPU 및 가속기 사이의 통신으로 인한 통신 오버헤드가 발생하지 않도록 하여 하드웨어의 최대 성능을 이끌어 내고 신경망 연산의 속도를 향상시킬 수 있다.
- [0028] 또한, 본 발명의 실시예에 의하면, CPU 및 가속기의 휴지기를 최소화하고, 상황에 따라 전압 및 주파수의 조절을 통하여 불필요한 전력 소비를 줄이고 신경망 연산 장치의 에너지 효율을 개선할 수 있다.

도면의 간단한 설명

- [0029] 도 1은 일 실시예에 따른 신경망 연산 장치의 구성도이다.
- 도 2는 일 실시예에 따른 신경망 연산 방법과 종래의 신경망 연산 방법을 비교하기 위한 도면이다.
- 도 3은 일 실시예에 따른 가속기가 연산을 수행하는 동안, CPU가 전처리 연산 및 후처리 연산을 중첩적으로 수행하는 것을 설명하기 위한 도면이다.
- 도 4는 일 실시예에 따른 CPU 및 가속기가 연산을 수행하는 방식을 종래의 연산 방식과 비교하기 위한 도면이다.
- 도 5는 또다른 실시예에 따른 CPU 및 가속기가 연산을 수행하는 방식을 또다른 종래의 연산 방식과 비교하기 위한 도면이다.
- 도 6은 일 실시예에 따른 가속기가 제1 계층 연산을 수행하는 동안, CPU가 제2 계층 연산, 후처리 연산 및 전처리 연산을 중첩적으로 수행하는 것을 설명하기 위한 도면이다.
- 도 7은 일 실시예에 따른 가속기가 특정 배치의 제1 계층 연산을 수행하는 동안, CPU가 다른 배치의 제2 계층의 연산, 후처리 연산 및 전처리 연산을 중첩적으로 수행하는 것을 설명하기 위한 도면이다.
- 도 8은 일 실시예에 따라 가속기의 성능 및 CPU의 성능에 기초하여 복수개의 계층을 분류하는 방법을 설명하기 위한 도면이다.
- 도 9는 일 실시예에 따른 신경망 연산 방법의 순서도이다.
- 도 10은 일 실시예에 따른 신경망 연산 방법과 기존의 연산 방법의 성능을 비교하는 표이다.
- 도 11은 일 실시예에 따른 신경망 연산 방법이 기존의 연산 방법보다 개선되었음을 나타내는 그래프이다.
- 도 12는 일 실시예에 따른 절전 모드의 효과를 설명하기 위한 그래프이다.

발명을 실시하기 위한 구체적인 내용

- [0030] 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성요소를 지칭한다. 본 명세서가 실시예들의 모든 요소들을 설명하는 것은 아니며, 개시된 발명이 속하는 기술분야에서 일반적인 내용 또는 실시예들 간에 중복되는 내용은 생략한다.
- [0031] 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [0032] 제1, 제2 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위해 사용되는 것으로, 구성요소가 전술된 용어들에 의해 제한되는 것은 아니다.
- [0033] 단수의 표현은 문맥상 명백하게 예외가 있지 않는 한, 복수의 표현을 포함한다.
- [0034] 각 단계들에 있어 식별부호는 설명의 편의를 위하여 사용되는 것으로 식별부호는 각 단계들의 순서를 설명하는 것이 아니며, 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않는 이상 명기된 순서와 다르게 실시될 수 있다.
- [0035] 이하 첨부된 도면들을 참고하여 개시된 발명의 작용 원리 및 실시예들에 대해 설명한다.
- [0036] 도 1은 일 실시예에 따른 신경망 연산 장치의 구성도이다.
- [0037] 도 1을 참조하면, 본 발명의 실시예에 따른 신경망 연산 장치(100)는, CPU(Central Processing Unit)(110), 가속기(accelerator)(120), 프로세서(130), 메모리(140)를 포함할 수 있다.
- [0038] 신경망 연산 장치(100)는 복수개의 계층(300)으로 구성된 인공신경망 모델에 기초하여 입력 데이터(201)에 대한 연산을 수행할 수 있다.
- [0039] 본 발명의 신경망 연산은 CPU(110) 또는 가속기(120)에 의해 구동되는 딥 러닝(deep learning) 방식에 의해 수행될 수 있다.
- [0040] 즉, 본 발명의 신경망 연산은 심층 신경망(Deep Neural Network; DNN) 방식에 의해 수행될 수 있다. 심층 신경망 방식의 연산은 컴퓨터가 스스로 분류 레이블을 만들어 내고, 공간을 왜곡하고 데이터를 구분하는 과정을 반

복해서 최적의 구변선을 도출해내는 방식일 수 있다.

- [0041] 하지만, 본 발명의 신경망 연산의 방식이 반드시 심층 신경망 방식에 한정되는 것은 아니다. 즉, 본 발명의 연산은 다른 종류의 인공 신경망(artificial neural network; ANN) 방식, 합성곱 신경망(convolutional neural network; CNN) 방식 등과 같은 임의의 머신 러닝(machine learning) 방식에 의해서도 수행될 수 있다.
- [0042] 인공신경망 모델은 본 발명의 신경망 연산에 이용되기 전에 미리 학습용 입력 데이터로부터 추출된 특징을 입력 받아 학습될 수 있다. 또한, 미리 학습된 인공신경망 모델은 메모리(140)에 저장되어 있을 수 있다.
- [0043] 인공신경망 모델은 복수개의 계층(layer)(300)으로 구성될 수 있다.
- [0044] 예를 들어, 인공신경망 모델이 합성곱 신경망(CNN) 모델이라면, 인공신경망 모델은 다수의 합성곱층(convolution layer)과, 다수의 풀링층(pooling layer), 완전 연결층(fully connected layer) 등을 포함할 수 있다.
- [0045] 이때, 다수의 합성곱층은 입력 데이터(201)로부터 추출된 특징들을 기반으로 합성곱 필터를 이용하여 시간 축 및/또는 공간 축 합성곱(convolution) 처리를 수행해서 출력 데이터(204)를 생성할 수 있다.
- [0046] 한편, 전술한 바와 같이 인공신경망 모델은 합성곱 신경망 모델일 수 있으나, 이에 한정되는 것은 아니다. 즉, 복수개의 계층(300)으로 구성될 수 있다면 인공신경망 모델은 합성곱 신경망 모델이 아닌 다른 인공신경망 모델 또는 심층 신경망 모델이더라도 상관없다.
- [0047] CPU(110)는 입력 데이터(201)에 대하여 전처리 연산을 수행하여 전처리 데이터(202)를 출력하도록 구성될 수 있다.
- [0048] 전처리 연산은 입력된 데이터를 머신 러닝 알고리즘에 적합한 전처리 데이터(202)로 변환시키는 연산일 수 있다. 전처리 연산은 인공신경망 모델을 실제로 활용하는 단계뿐만 아니라 인공신경망 모델을 학습시키는 단계에서 수행될 수 있다.
- [0049] CPU(110)는 출력된 전처리 데이터(202)를 가속기(120)에 전달할 수 있다. 이때, CPU(110)는 가속기(120)에 전처리 데이터(202)에 대한 연산을 수행하라는 처리 명령을 전달할 수 있다.
- [0050] 가속기(120)는 전처리 데이터(202)에 대하여 인공신경망 모델의 제1 계층(301)에 의한 연산을 수행하여 중간 데이터(203)를 출력하도록 구성될 수 있다.
- [0051] 가속기(120)는 GPU(Graphics Processing Unit)를 포함할 수 있으나, 이에 한정되는 것은 아니며, 전처리 데이터(202)에 대하여 인공신경망 모델에 의한 연산을 수행할 수 있다면 어떠한 디바이스(device)라도 본 발명의 가속기(120)를 구성할 수 있다.
- [0052] 인공신경망 모델을 구성하는 복수개의 계층(300)들은 적어도 하나 이상의 제1 계층(301) 및 적어도 하나 이상의 제2 계층(302)으로 분류될 수 있다.
- [0053] 제1 계층(301)은 복수개의 계층(300) 중 먼저 연산에 이용되는 계층(300)들일 수 있다. 또한, 제2 계층(302)은 복수개의 계층(300) 중 제1 계층(301)에 의한 연산 직후의 연산에 이용되는 계층(300)들일 수 있다.
- [0054] 즉, 전처리 데이터(202)에 대하여 제1 계층(301)에 의한 연산이 수행되어 중간 데이터(203)가 출력 되면 비로소 해당 중간 데이터(203)에 대하여 제2 계층(302)에 의한 연산이 수행될 수 있다.
- [0055] 중간 데이터(203)는 복수개의 계층(300)에 따라 순서대로 연산이 수행되는 과정에서, 어느 특정 계층(300)에 의한 연산까지 수행되었을 때 출력되는 데이터일 수 있다.
- [0056] 즉, 가속기(120)는 전처리 데이터(202)에 대하여, 복수개의 계층(300)들 중 먼저 연산에 이용되는 계층(300)인 제1 계층(301)에 의한 연산을 수행하여 중간 데이터(203)를 출력할 수 있다.
- [0057] CPU(110)는 중간 데이터(203)에 대하여 인공신경망 모델의 제2 계층(302)에 의한 연산을 수행하여 출력 데이터(204)를 출력하도록 구성될 수 있다.
- [0058] CPU(110) 및 가속기(120)는 신경망 연산 장치(100)에 포함된 복수개의 프로세서(130) 중 어느 하나의 프로세서(130)를 포함할 수 있다. 또한, 지금까지 설명된 본 발명의 실시예 및 앞으로 설명할 실시예에 따른 신경망 연산 방법은 CPU(110), 가속기(120) 및 프로세서(130)에 의해 구동될 수 있는 프로그램의 형태로 구현될 수 있다.
- [0059] 여기서 프로그램은, 프로그램 명령, 데이터 파일 및 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.

프로그램은 기계어 코드나 고급 언어 코드를 이용하여 설계 및 제작된 것일 수 있다. 프로그램은 상술한 부호 수정을 위한 방법을 구현하기 위하여 특별히 설계된 것일 수도 있고, 컴퓨터 소프트웨어 분야에서 통상의 기술자에게 기 공지되어 사용 가능한 각종 함수나 정의를 이용하여 구현된 것일 수도 있다. 전술한 정보 표시 방법을 구현하기 위한 프로그램은 CPU(110), 가속기(120) 및 프로세서(130)에 의해 판독 가능한 기록매체에 기록될 수 있다. 이때, 기록매체는 메모리(140)일 수 있다.

- [0060] 메모리(140)는 전술한 동작 및 후술하는 동작을 수행하는 프로그램을 저장할 수 있으며, 메모리(140)는 저장된 프로그램을 실행시킬 수 있다. 프로세서(130)와 메모리(140)가 복수인 경우에, 이들이 하나의 칩에 집적되는 것도 가능하고, 물리적으로 분리된 위치에 마련되는 것도 가능하다. 메모리(140)는 데이터를 일시적으로 기억하기 위한 S램(Static Random Access Memory, S-RAM), D램(Dynamic Random Access Memory) 등의 휘발성 메모리를 포함할 수 있다. 또한, 메모리(140)는 제어 프로그램 및 제어 데이터를 장기간 저장하기 위한 롬(Read Only Memory), 이퍼롬(Erasable Programmable Read Only Memory: EPROM), 이이퍼롬(Electrically Erasable Programmable Read Only Memory: EEPROM) 등의 비휘발성 메모리를 포함할 수 있다.
- [0061] CPU(110), 가속기(120) 및 프로세서(130)는 각종 논리 회로와 연산 회로를 포함할 수 있으며, 메모리(140)로부터 제공된 프로그램에 따라 데이터를 처리하고, 처리 결과에 따라 제어 신호를 생성할 수 있다.
- [0062] 도 2는 일 실시예에 따른 신경망 연산 방법과 종래의 신경망 연산 방법을 비교하기 위한 도면이다.
- [0063] 도 2를 참조하면, 종래의 연산 방법 및 본 발명의 실시예에 따른 신경망 연산 방법은 입력 데이터(201)에 포함된 복수개의 배치(400)에 대한 연산을 수행할 수 있다.
- [0064] 배치(batch)(400)는 딥 러닝 연산 과정에서 인공신경망 모델의 가중치를 한번 업데이트할 때 이용되는 샘플들의 묶음을 일 수 있다. 예를 들어, 배치(400)의 크기가 10이라면 10개의 샘플 단위마다 인공신경망 모델의 가중치를 한 번씩 업데이트할 수 있다. 만약 총 200개의 학습용 입력 데이터가 있고, 배치(400)의 크기가 10이라면 20번 가중치가 업데이트 될 수 있다.
- [0065] 도 2의 (a)를 참조하면, 종래의 연산 방법은 CPU가 입력 데이터(201)에 대하여 전처리 과정을 수행하여 전처리 데이터(202)를 출력하고, 가속기가 전처리 데이터(202)에 대하여 연산을 수행하여 출력된 데이터를 다시 CPU에 전달하고, CPU가 전달받은 데이터에 대하여 후처리 연산을 수행한다.
- [0066] 예를 들어, 종래의 연산 방법은 CPU가 제1 배치(401)의 입력 데이터(201)에 대한 전처리 과정을 수행하여 제1 배치(401)의 전처리 데이터(202)를 출력하고, 가속기가 제1 배치(401)의 전처리 데이터(202)에 대하여 연산을 수행하여 출력된 데이터를 다시 CPU에 전달하고, CPU가 전달받은 데이터에 대하여 후처리 연산을 수행하여 제1 배치(401)의 출력 데이터(204)를 출력한다. 이때 CPU는 제1 배치(401)의 출력 데이터(204)를 출력하고 비로소 제2 배치(402)의 입력데이터에 대한 전처리 과정을 수행하고, 제2 배치(402)의 데이터에 대하여 전술한 과정을 반복한다.
- [0067] 전술한 연산 방법은 연산 과정에서 CPU와 가속기 중 하나의 프로세서만 특정 시점에 활용한다는 특징을 갖고 있다. 따라서 이러한 연산 방법은 하드웨어가 낼 수 있는 최대의 성능을 다 내지 못하는 단점이 존재한다.
- [0068] 또한, 두 개의 서로 다른 프로세서 중 하나만 사용함으로써 사용하지 않는 프로세서가 불필요한 파워를 소비하는 문제를 일으켜 하드웨어 전체의 에너지 효율을 떨어뜨린다는 단점도 있다.
- [0069] 결과적으로, 종래의 연산 방법은 에너지의 효율이 중요시되는 엣지 디바이스(edge device)와 같은 연산 장치에 치명적인 영향을 줄 수 있다. 따라서, 리소스의 양이 제한된 엣지 디바이스에서 주어진 리소스를 최대한 활용하여 인공신경망의 성능을 향상시키며 하드웨어 에너지 효율을 향상시키는 방법이 요구된다.
- [0070] 본 발명의 실시예에 따른 신경망 연산 장치(100)는 전술한 문제를 해결하기 위하여 엣지 디바이스에 들어오는 입력 데이터(201)를 처리하는 과정에서 두 개의 프로세서를 동시에 사용하여 각각의 하드웨어들이 불필요하게 쉬는 시간을 최소화할 수 있다.
- [0071] 도 2의 (b)를 참조하면, 본 발명의 실시예에 따른 CPU(110)는 입력 데이터(201)에 포함된 복수개의 배치(batch)(400)에 대한 전처리 연산을 수행하여 각 배치(400)의 전처리 데이터(202)를 출력하도록 구성될 수 있다.
- [0072] 가속기(120)는 인공신경망 모델에 기초하여 각 배치(400)의 전처리 데이터(202)에 대한 연산을 수행하여 각 배치(400)의 중간 데이터(203)를 출력하도록 구성될 수 있다.

- [0073] 본 발명의 신경망 연산 방법은 종래의 연산 방법과 달리, 가속기(120)가 어느 한 배치(400)의 전처리 데이터(202)에 대하여 연산을 수행하는 동안, CPU(110)가 중첩적으로 또 다른 배치(400)의 데이터에 대한 연산을 수행할 수 있다.
- [0074] 예를 들어, 본 발명의 신경망 연산 방법은 가속기(120)가 제2 배치(402)의 전처리 데이터(202)에 대하여 연산을 수행하는 동안, CPU(110)가 제1 배치(401)의 데이터에 대하여 연산을 수행하거나, 제3 배치(403)의 데이터에 대하여 연산을 수행할 수 있다.
- [0075] 즉, 본 발명은 CPU(110)에서 처리해야 하는 전처리 및 후처리 연산과 가속기(120)에서 수행해야 하는 연산 과정을 동시에 진행함으로써 각각의 하드웨어들의 휴지기를 없앤다.
- [0076] 이를 통해서 각각의 프로세서의 휴지기를 최소화하여 하드웨어의 최대 성능을 이끌어 낼 수 있을 뿐만 아니라 동시에 불필요한 전력 소비를 줄여 에너지 효율을 개선할 수 있다.
- [0077] 도 3은 일 실시예에 따른 가속기가 연산을 수행하는 동안, CPU가 전처리 연산 및 후처리 연산을 중첩적으로 수행하는 것을 설명하기 위한 도면이며, 도 4는 일 실시예에 따른 CPU 및 가속기가 연산을 수행하는 방식을 종래의 연산 방식과 비교하기 위한 도면이다.
- [0078] 도 3 및 도4를 참조하면, 종래의 연산 방식과 달리, 본 발명의 신경망 연산 방법은 가속기(120)가 제1 배치(401)의 전처리 데이터(202)에 대하여 제1 계층(301)에 의한 연산을 처리하는 동안, CPU(110)는 제2 배치(402)의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 제2 배치(402)의 전처리 데이터(202)를 출력하도록 구성될 수 있다.
- [0079] 즉, CPU(110)는 가속기(120)가 계층(300)에 의한 연산을 수행하는 동안 연산을 쉬고 있는 것이 아니라, 전처리 연산을 수행할 수 있다. 마찬가지로, 가속기(120)는 CPU(110)가 전처리 연산을 수행하는 동안 연산을 수행할 수 있다.
- [0080] 또한, 가속기(120)가 제2 배치(402)의 전처리 데이터(202)에 대하여 제1 계층(301)에 의한 연산을 처리하는 동안, CPU(110)는 제1 배치(401)의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 제1 배치(401)의 출력 데이터(204)를 출력할 수 있다.
- [0081] 즉, CPU(110)는 가속기(120)가 계층(300)에 의한 연산을 수행하는 동안 연산을 쉬고 있는 것이 아니라, 후처리 연산을 수행할 수 있다. 마찬가지로, 가속기(120)는 CPU(110)가 후처리 연산을 수행하는 동안 연산을 수행할 수 있다.
- [0082] 도 5는 또다른 실시예에 따른 CPU 및 가속기가 연산을 수행하는 방식을 또다른 종래의 연산 방식과 비교하기 위한 도면이다.
- [0083] 도 5의 (a)를 참조하면, 종래의 연산 방식은 가속기를 이용하여 처리되는 연산들을 프로세서의 성능에 따라 분배하여 CPU 및 가속기 각각이 각자 맡은 부분을 동시에 처리한다.
- [0084] 예를 들어, 종래의 연산 방법은 신경망 모델의 어느 한 계층(300)에 의한 연산을 수행할 때, 해당 계층(300)의 연산을 CPU 및 가속기의 성능에 따라 분배하여 CPU 및 가속기 각각이 각자 맡은 연산을 동시에 처리하도록 한다.
- [0085] 이후 해당 계층(300)에 대한 연산이 종료되면 다음 계층(300)에 대한 연산에 대해서도 CPU 및 가속기가 각자 맡은 연산을 동시에 처리한다.
- [0086] 종래의 연산 방법은 가속기가 혼자 처리하던 연산을 CPU와 나누어 처리하여 전체 연산 속도를 늘렸다는 장점이 있으나, CPU가 혼자 처리하던 전처리 및 후처리 연산은 여전히 CPU 만을 이용하여 처리를 하기 때문에 가속기의 휴지기는 없애지 못했다는 문제가 있다.
- [0087] 또한, 종래의 연산 방법은 인공신경망을 이루고 있는 모든 인공신경망 모델의 계층(300)의 연산을 프로세서의 연산 성능에 따라 배분하기 때문에 각자 맡은 부분의 계산 결과를 각각의 저장공간에 저장한다. 이때, 각각의 인공신경망 모델의 계층(300)에서 연산이 끝나고 다음 인공신경망 계층(300)으로 넘어가기 위해서는 앞쪽의 인공신경망 계층(300)의 연산 결과가 필요하다. 종래의 연산 방법은 CPU와 가속기에 연산 결과가 산재되어 있기 때문에 이를 통합해주어야 하고, 이 과정에서 CPU와 가속기의 통신이 필수적이며, 결국 통신 오버헤드가 발생하여 신경망 연산 성능이 저해되는 문제가 있다.

- [0088] 도 5의 (b)를 참조하면, 본 발명의 신경망 연산 방법은 종래의 연산 방법과 달리 특정 계층(300)에 대한 연산을 CPU(110) 및 가속기(120) 중 하나에 의해서만 이루어질 수 있다.
- [0089] 즉, 인공신경망 모델을 구성하는 복수개의 계층(300)은 미리 제1 계층(301)들과 제2 계층(302)들로 분류될 수 있고, 가속기(120)는 특정 배치(400)의 데이터에 대하여 제1 계층(301)에 의한 연산을 수행하여 해당 배치(400)의 중간 데이터(203)를 생성할 수 있다. 이후, CPU(110)는 해당 배치(400)의 중간 데이터(203)에 대하여 제2 계층(302)에 의한 연산을 수행할 수 있다.
- [0090] 본 발명의 CPU(110)는 가속기(120)가 제2 배치(402)의 전처리 데이터(202)에 대하여 제1 계층(301)에 의한 연산을 처리하는 동안, 제1 배치(401)의 중간 데이터(203)에 대하여 제2 계층(302)에 의한 연산을 중첩적으로 처리하도록 구성될 수 있다.
- [0091] 즉, 본 발명의 신경망 연산 방법은 단순히 인공신경망 모델을 구성하는 계층(300)들 중 일부의 계층(300)에 대하여 CPU(110)가 연산을 수행하도록 한 것이 아니라, 가속기(120)가 특정 배치(400)의 제1 계층(301)에 대한 연산을 수행하는 동안, CPU(110)가 또 다른 배치(400)의 제2 계층(302)에 대한 연산을 동시에 수행할 수 있다.
- [0092] 결과적으로, 본 발명은 CPU(110)의 연산과 가속기(120)의 연산을 중첩하였기 때문에 두 종류 프로세서의 휴지기를 없앨 수 있다.
- [0093] 또한, 본 발명의 신경망 연산 방법은 CPU(110)와 가속기(120)가 서로 독립적인 연산을 수행하기 때문에 종래의 기술에서 발생한 추가적인 통신 오버헤드가 발생하지 않을 수 있다.
- [0094] 도 6은 일 실시예에 따른 가속기가 제1 계층 연산을 수행하는 동안, CPU가 제2 계층 연산, 후처리 연산 및 전처리 연산을 중첩적으로 수행하는 것을 설명하기 위한 도면이며, 도 7은 일 실시예에 따른 가속기가 특정 배치(400)의 제1 계층 연산을 수행하는 동안, CPU가 다른 배치(400)의 제2 계층의 연산, 후처리 연산 및 전처리 연산을 중첩적으로 수행하는 것을 설명하기 위한 도면이다.
- [0095] 도 6을 참조하면, 본 발명의 신경망 연산 방법은 가속기(120)가 제2 배치(402)의 데이터에 대하여 제1 계층(301)에 의한 연산을 수행하는 동안, CPU(110)는 제1 배치(401)의 데이터 및 제3 배치(403)의 데이터에 대하여 연산을 수행할 수 있다.
- [0096] 이렇게 가속기(120)가 특정 배치(400)의 제1 계층(301)에 의한 연산을 수행하는 동안 CPU(110)가 또다른 배치(400)에 대하여 연산을 수행하는 과정은 반복적으로 이루어질 수 있다.
- [0097] 예를 들어, 가속기(120)가 제2 배치(402)의 데이터에 대하여 제1 계층(301)에 의한 연산을 수행하는 동안 CPU(110)가 제1 배치(401)의 데이터 및 제3 배치(403)의 데이터에 대하여 연산을 수행한 직후, CPU(110)는 가속기(120)가 제3 배치(403)의 데이터에 대하여 제1 계층(301)에 의한 연산을 수행하는 동안 CPU(110)가 제2 배치(402)의 데이터 및 제4 배치의 데이터에 대하여 연산을 수행할 수 있다.
- [0098] 도 7을 참조하면, 가속기(120)가 제2 배치(402)의 제1 계층(301)에 대한 연산을 수행할 때 CPU(110)가 어느 연산을 중첩적으로 수행하는지 확인할 수 있다.
- [0099] 가속기(120)가 제2 배치(402)의 전처리 데이터(202)에 대하여 제1 계층(301)에 대한 연산을 처리하는 동안, CPU(110)는 제1 배치(401)의 중간 데이터(203)에 대하여 제2 계층(302)에 의한 연산을 중첩적으로 처리하고, 제1 배치(401)의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 제1 배치(401)의 출력 데이터(204)를 출력할 수 있다.
- [0100] 또한, 가속기(120)가 제2 배치(402)의 전처리 데이터(202)에 대하여 제1 계층(301)에 대한 연산을 처리하는 동안, CPU(110)는 제3 배치(403)의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 제3 배치(403)의 전처리 데이터(202)를 출력할 수 있다.
- [0101] 이러한 과정은 또 다른 배치(400)에 대한 연산 과정에서 반복적으로 이루어질 수 있다.
- [0102] 예를 들어, 가속기(120)가 제3 배치(403)의 전처리 데이터(202)에 대하여 제1 계층(301)에 대한 연산을 처리하는 동안, CPU(110)는 제2 배치(402)의 중간 데이터(203)에 대하여 제2 계층(302)에 의한 연산을 중첩적으로 처리하고, 제2 배치(402)의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 제2 배치(402)의 출력 데이터(204)를 출력할 수 있다.
- [0103] 또한, 가속기(120)가 제3 배치(403)의 전처리 데이터(202)에 대하여 제1 계층(301)에 대한 연산을 처리하는 동

안, CPU(110)는 제4 배치의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 제4 배치의 전처리 데이터 (202)를 출력할 수 있다.

[0104] 이처럼 본 발명의 신경망 연산 방법은 CPU(110) 및 가속기(120) 각각이 이용되지 않는 시간 없이 동시에 연산을 수행할 수 있다. 한편, 도시된 것과 같이 CPU(110) 및 가속기(120) 중 어느 하나가 쉬는 시간이 없도록 하려면 제1 계층(301) 및 제2 계층(302)을 적절히 분류하는 것이 필요하다.

[0105] 도 8은 일 실시예에 따라 가속기의 성능 및 CPU의 성능에 기초하여 복수개의 계층을 분류하는 방법을 설명하기 위한 도면이다.

[0106] 도 8을 참조하면, 인공신경망 모델에 따라 계층(300)을 분류하는데 기준이 되는 파라미터가 다른 것을 확인할 수 있다.

[0107] 모델의 파라미터 중 α 는 어느 한 배치(400)의 신경망 연산에 걸리는 총 시간에 대한 해당 배치(400)의 전처리와 후처리 연산에 걸리는 시간의 비율일 수 있다.

[0108] 모델의 파라미터 중 γ 는 CPU(110) 및 가속기(120) 사이의 성능 차이를 나타내는 파라미터일 수 있다. 이때, γ 는 가속기(120)의 연산 속도가 CPU(110)의 연산 속도에 비해 몇 배 빠른지 나타내는 파라미터일 수 있다.

[0109] 종래의 연산 방법에서는 가속기(120)가 연산에 $1-\alpha$ 의 시간 비율을 사용하지만, 본 발명의 신경망 연산 방법은 $1-\alpha$ 의 전체 연산 시간에서 중첩된 전처리 및 후처리 작업시간 비율인 α 를 빼게 되므로, CPU(110) 및 가속기(120)로 분할할 수 있는 연산 시간 비율은 $1-2\alpha$ 일 수 있다.

[0110] CPU(110)와 가속기(120) 사이에서 $1-2\alpha$ 의 시간 비율이 걸리는 연산을 분할하기 위한 최적의 연산 시간 비율은 $(1-2\alpha) \times \frac{\gamma}{\gamma+1}$ 일 수 있다.

[0111] 결과적으로, 본 발명의 신경망 연산 방법에서 하나의 배치(400)에 대해 걸리는 연산 시간은 $\alpha + \frac{(1-2\alpha)\gamma}{\gamma+1}$ 일 수 있다.

[0112] 또한, 본 발명의 신경망 연산 방법에서 하나의 배치(400)에 대해 걸리는 연산 속도는 종래의 연산 속도보다 하기 [방정식 1]에 도시된 값만큼 증가할 수 있다.

[0113] [방정식 1]

$$\text{Speedup} = \frac{\gamma + 1}{1 - (\alpha - 1)(\gamma - 1)}$$

[0115] 한편, 인공신경망 모델에 따라 α 및 γ 의 값은 다를 수 있으며, 이에 따라 인공신경망 모델의 계층(300) 중 제1 계층(301)으로 분류되는 계층(300)의 수 및 제2 계층(302)으로 분류되는 계층(300)의 수 또한 인공신경망 모델의 종류에 따라 다를 수 있다.

[0116] 도 8의 (a)의 표는 동일한 연산 장치가 연산을 수행하더라도 인공신경망 모델에 따라 α 및 γ 의 값이 다를 수 있음을 나타낸다.

[0117] 예를 들어, MLP 모델은 다른 인공신경망 모델에 비하여 상대적으로 큰 α 값을 가지고, 상대적으로 작은 γ 값을 가질 수 있다. 또한, YOLO 모델은 반대로 상대적으로 작은 α 값을 가지고, 상대적으로 큰 γ 값을 가질 수 있다. 결국, 본 발명에서 이용되는 인공신경망 모델이 MLP 모델인지 또는 YOLO 모델인지에 따라 복수개의 계층(300) 중 제1 계층(301)으로 분류되는 계층(300)의 수 및 제2 계층(302)으로 분류되는 계층(300)의 수는 다를 수 있다.

[0118] 프로세서(130)는 가속기(120)의 성능 및 CPU(110)의 성능에 기초하여, 복수개의 계층(300)을 제1 계층(301) 및 제2 계층(302)으로 분류하도록 구성될 수 있다.

[0119] 프로세서(130)는 CPU(110)의 연산속도 정보, 가속기(120)의 연산속도 정보 및 총 연산 시간에 대한 전처리 연산 시간과 후처리 연산 시간의 비율 정보에 기초하여 복수개의 계층(300)을 제1 계층(301) 및 제2 계층(302)으로 분류하도록 구성될 수 있다.

[0120] 즉, 프로세서(130)는 어느 한 인공신경망 모델을 이용할 때의 CPU(110) 연산속도에 대한 가속기(120)의 연산속도 비율 정보(γ) 및 해당 인공신경망 모델을 이용할 때의 총 연산 시간에 대한 전처리와 후처리 연산 시간의

비율 정보(α)에 기초하여 복수개의 계층(300)을 제1 계층(301) 및 제2 계층(302)으로 분류할 수 있다.

- [0121] 도 8의 (b)를 참조하면, γ 값이 10일때의 α 에 따른 이상적인 연산 속도 증가 정도 및 γ 값이 100일때의 α 에 따른 이상적인 연산 속도 증가 정도를 확인할 수 있다. 또한, 실제 인공신경망 모델들의 α 파라미터와 그에 따른 연산 속도 증가 정도를 확인할 수 있다.
- [0122] 프로세서(130)는 신경망 연산 장치(100)의 연산 속도가 기준 처리 속도 이상인지 여부를 판단할 수 있다.
- [0123] 기준 처리 속도는 신경망 연산 장치(100)의 상태를 절전 모드로 변경할지 여부를 결정하는데 기준이 되는 연산 속도일 수 있다.
- [0124] 프로세서(130)는 신경망 연산 장치(100)의 연산 속도가 기준 처리 속도 이상이면, 신경망 연산 장치(100)에 공급되는 전압 및 주파수 중 적어도 하나를 감소시킬 수 있다. 이때, 신경망 연산 장치(100)의 상태는 절전 모드로 변경되고, 신경망 연산 장치(100)가 소모하는 전력 및 에너지가 감소할 수 있다.
- [0125] 프로세서(130)는 신경망 연산 장치(100)의 연산 속도가 기준 처리 속도 미만이면, 신경망 연산 장치(100)에 공급되는 전력을 증가시킬 수 있다. 이때, 신경망 연산 장치(100)에 공급되는 전압 및 주파수 중 적어도 하나가 증가될 수 있으며, 이에 따라 연산 속도는 기준 처리 속도 수준으로 증가할 수 있다.
- [0126] 이상에서 설명된 구성요소들의 성능에 대응하여 적어도 하나의 구성요소가 추가되거나 삭제될 수 있다. 또한, 구성요소들의 상호 위치는 시스템의 성능 또는 구조에 대응하여 변경될 수 있다는 것은 당해 기술 분야에서 통상의 지식을 가진 자에게 용이하게 이해될 것이다.
- [0127] 도 9는 일 실시예에 따른 신경망 연산 방법의 순서도이다. 이는 본 발명의 목적을 달성하기 위한 바람직한 실시예일 뿐이며, 필요에 따라 일부 구성이 추가되거나 삭제될 수 있음은 물론이다.
- [0128] 도 9를 참조하면, 프로세서(130)는 가속기(120)의 성능 및 CPU(110)의 성능에 기초하여, 복수개의 계층(300)을 제1 계층(301) 및 제2 계층(302)으로 분류할 수 있다(1001).
- [0129] CPU(110)는 입력 데이터(201)에 대하여 전처리 연산을 수행하여 전처리 데이터(202)를 출력할 수 있다(1002). 이때, CPU(110)는 입력 데이터(201)에 포함된 복수개의 배치(400)에 대한 전처리 연산을 수행하여 각 배치(400)의 전처리 데이터(202)를 출력할 수 있다.
- [0130] 가속기(120)는 전처리 데이터(202)에 대하여 인공신경망 모델의 제1 계층(301)에 의한 연산을 수행하여 중간 데이터(203)를 출력할 수 있다(1003). 이때, 가속기(120)는 인공신경망 모델에 기초하여 각 배치(400)의 전처리 데이터(202)에 대한 연산을 수행하여 각 배치(400)의 중간 데이터(203)를 출력할 수 있다.
- [0131] CPU(110)는 중간 데이터(203)에 대하여 인공신경망 모델의 제2 계층(302)에 의한 연산을 수행하여 출력 데이터(204)를 출력할 수 있다(1004).
- [0132] 이때, CPU(110)는 가속기(120)가 제2 배치(402)의 전처리 데이터(202)에 대하여 제1 계층(301)에 대한 연산을 처리하는 동안, 제1 배치(401)의 중간 데이터(203)에 대하여 제2 계층(302)에 의한 연산을 중첩적으로 처리하고, 제1 배치(401)의 데이터에 대하여 후처리 연산을 중첩적으로 수행하여 제1 배치(401)의 출력 데이터(204)를 출력하고, 제3 배치(403)의 데이터에 대하여 전처리 연산을 중첩적으로 수행하여 제3 배치(403)의 전처리 데이터(202)를 출력할 수 있다.
- [0133] 프로세서(130)는 신경망 연산 장치(100)의 연산 속도가 기준 처리 속도 이상인지 여부를 판단할 수 있다(1005).
- [0134] 신경망 연산 장치(100)의 연산 속도가 기준 처리 속도 이상이면(1005의 '예'), 프로세서(130)는 신경망 연산 장치(100)에 공급되는 전압 및 주파수 중 적어도 하나가 감소하도록 신경망 연산 장치(100)를 제어할 수 있다(1006).
- [0135] 신경망 연산 장치(100)의 연산 속도가 기준 처리 속도 미만이면(1005의 '아니오'), 프로세서(130)는 신경망 연산 장치(100)에 공급되는 전력이 증가하도록 신경망 연산 장치(100)를 제어할 수 있다(1007).
- [0136] 도 10은 일 실시예에 따른 신경망 연산 방법과 기존의 연산 방법의 성능을 비교하는 표이고, 도 11은 일 실시예에 따른 신경망 연산 방법이 기존의 연산 방법보다 개선되었음을 나타내는 그래프이며, 도 12는 일 실시예에 따른 절전 모드의 효과를 설명하기 위한 그래프이다.
- [0137] 본 발명의 실시예에 따른 신경망 연산 장치(100)의 성능을 검증하기 위하여, 실제 NVIDIA의 하드웨어를 이용한

신경망 연산 실험을 진행하였다.

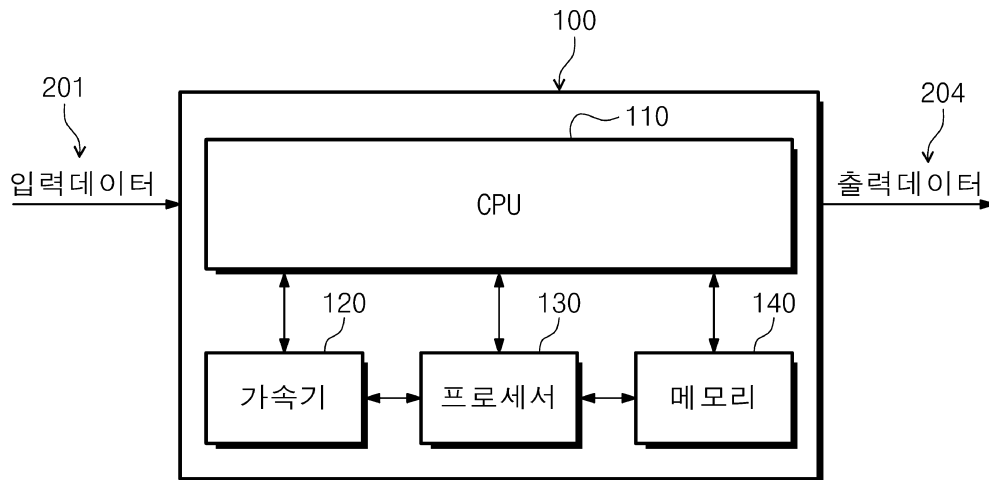
- [0138] 도 10의 (a)를 참조하면, 본 발명의 신경망 연산 방법(NeuroPipe)은 종래의 연산 방법(Baseline)보다 총 연산 시간이 적다는 결과의 실험 데이터를 확인할 수 있다.
- [0139] 한편, 본 실험은 DNN 모델로서 MobileNet을 사용하였으며, NVIDIA Jetson AGX Xavier 내 8개의 ARM CPU 코어 및 64개의 텐서 코어로 연산이 수행되었다.
- [0140] 본 발명의 신경망 연산 방법(NeuroPipe)은 도 5의 (a)에 도시된 종래의 연산 방법(Slicing)보다도 총 연산 시간이 적다는 결과가 나온 실험 데이터를 확인할 수 있다. 이는 본 발명의 신경망 연산 방법(NeuroPipe)은 엡지 디바이스 내부에서 CPU(110)와 가속기(120) 간의 통신으로 인한 통신 오버헤드가 발생하지 않기 때문이다.
- [0141] 또한, 본 발명의 신경망 연산 방법(NeuroPipe)은 도 5의 (a)에 도시된 종래의 연산 방법(Slicing)보다 필요한 메모리 용량이 적다는 결과가 나왔다.
- [0142] 도 10의 (b)를 참조하면, 본 발명의 신경망 연산 방법(NeuroPipe)에 따라 CPU(110) 및 가속기(120)의 성능을 고려하여 제1 계층(301) 및 제2 계층(302)을 분류한 연산 방법이 종래의 연산 방법(Baseline) 및 각 디바이스의 성능을 고려하지 않은 신경망 연산(PU-oblivious)보다 총 연산 시간이 적다는 결과의 실험 데이터를 확인할 수 있다.
- [0143] 이는 디바이스의 성능을 고려하지 않은 신경망 연산(PU-oblivious)의 경우, CPU(110) 및 가속기(120)의 성능 차이를 고려하지 않고 계층(300)을 할당했기 때문에 CPU(110) 입장에서는 기존의 전처리 및 후처리 연산과 과도한 수의 신경망 계층 연산을 수행하게 되어 총 연산 시간이 증가했기 때문이다.
- [0144] 도 11은 일 실시예에 따른 신경망 연산 방법이 기존의 연산 방법보다 개선되었음을 나타내는 그래프이다.
- [0145] 도 11의 (a)를 참조하면, 인공신경망 모델에 관계없이 본 발명의 신경망 연산 방법(NeuroPipe)은 종래의 연산 방법(Baseline)보다 연산 시간이 적다는 실험 결과를 확인할 수 있다.
- [0146] 평균적으로 본 발명의 신경망 연산 방법(NeuroPipe)은 종래의 연산 방법(Baseline)보다 연산 시간이 17.6% 감소했으며, 특히 MobileNet 모델의 경우, 연산 시간이 32% 감소하였다.
- [0147] 도 11의 (b)를 참조하면, 본 발명의 신경망 연산 방법(NP)이 종래의 연산 방법(Base)보다 전력은 더 소모한다는 실험 결과를 확인할 수 있다.
- [0148] 하지만, 도 11의 (c)를 참조하면, 본 발명의 신경망 연산 방법(NP)은 종래의 연산 방법(Base)보다 에너지를 덜 소모한다는 실험 결과를 확인할 수 있다.
- [0149] 평균적으로 본 발명의 신경망 연산 방법(NP)은 종래의 연산 방법(Base)보다 에너지를 8.4% 덜 소모했으며, 특히 MobileNet 모델의 경우, 에너지를 14.4% 덜 소모하였다.
- [0150] 또한, 도 11의 (d)를 참조하면, 본 발명의 신경망 연산 방법(NP)은 종래의 연산 방법(Base)보다 에너지 효율이 높다는 실험 결과를 확인할 수 있다. 구체적으로, 본 발명의 신경망 연산 방법(NP)은 종래의 연산 방법(Base)보다 EDP(Energy-delay product)가 낮다.
- [0151] 도 12는 일 실시예에 따른 절전 모드의 효과를 설명하기 위한 그래프이다.
- [0152] 도 12의 (a)를 참조하면, 본 발명의 신경망 연산이 절전 모드(NeuroPipe-P)에서 수행될 경우, 종래의 연산 방법(Baseline) 보다 연산 속도가 별 차이 없다는 실험 결과를 확인할 수 있다.
- [0153] 반면, 도 12의 (b)를 참조하면, 본 발명의 신경망 연산 방법이 절전 모드(NP-P)에서 수행될 경우, 종래의 연산 방법(Base)보다 평균적으로 전력 소모가 11.4% 감소한다는 실험 결과를 확인할 수 있다.
- [0154] 이는 본 발명의 신경망 연산 방법이 종래의 연산 방법보다 연산 속도가 빠르기 때문에, 신경망 연산 장치(100)에 공급되는 전압 및 주파수 중 적어도 하나를 감소시키는 절전 모드가 되면 종래의 연산 방법과 연산 속도는 동일할지라도 소모하는 에너지가 감소하기 때문이다.
- [0155] 이상에서와 같이 첨부된 도면을 참조하여 개시된 실시예들을 설명하였다. 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고도, 개시된 실시예들과 다른 형태로 본 발명이 실시될 수 있음을 이해할 것이다. 개시된 실시예들은 예시적인 것이며, 한정적으로 해석되어서는 안 된다.

부호의 설명

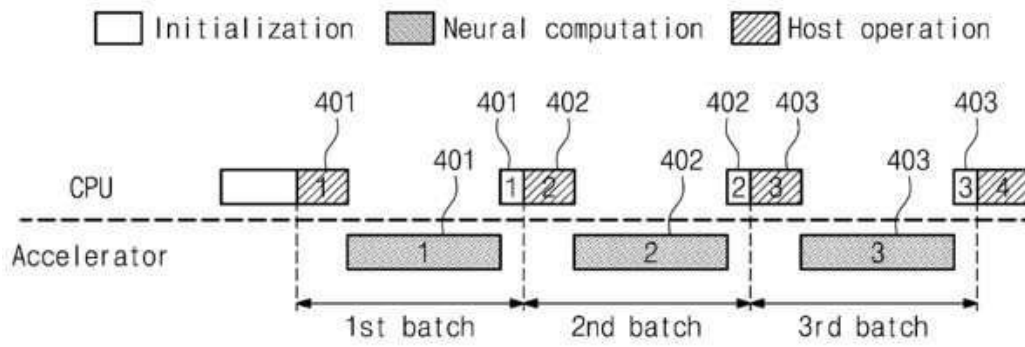
- [0156]
- 100: 신경망 연산 장치
 - 110: CPU
 - 120: 가속기
 - 130: 프로세서
 - 140: 메모리
 - 201: 입력 데이터
 - 202: 전처리 데이터
 - 203: 중간 데이터
 - 204: 출력 데이터
 - 300: 계층
 - 301: 제1 계층
 - 302: 제2 계층
 - 400: 배치
 - 401: 제1 배치
 - 402: 제2 배치
 - 403: 제3 배치

도면

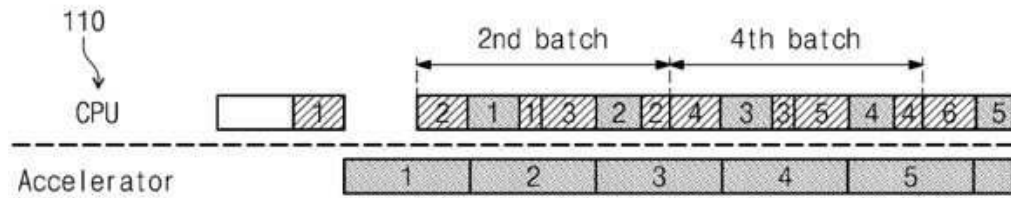
도면1



도면2



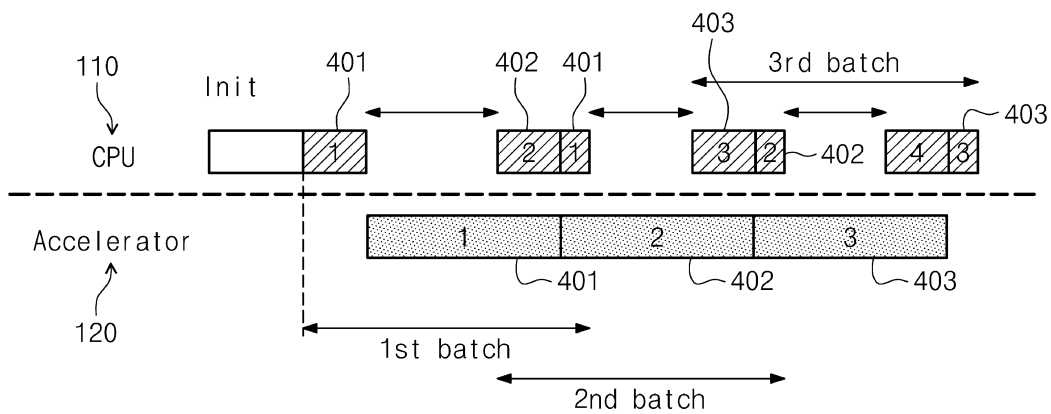
(a) Conventional host-device execution(baseline)



(b) Pipelined heterogeneous processing units (NeuroPipe)

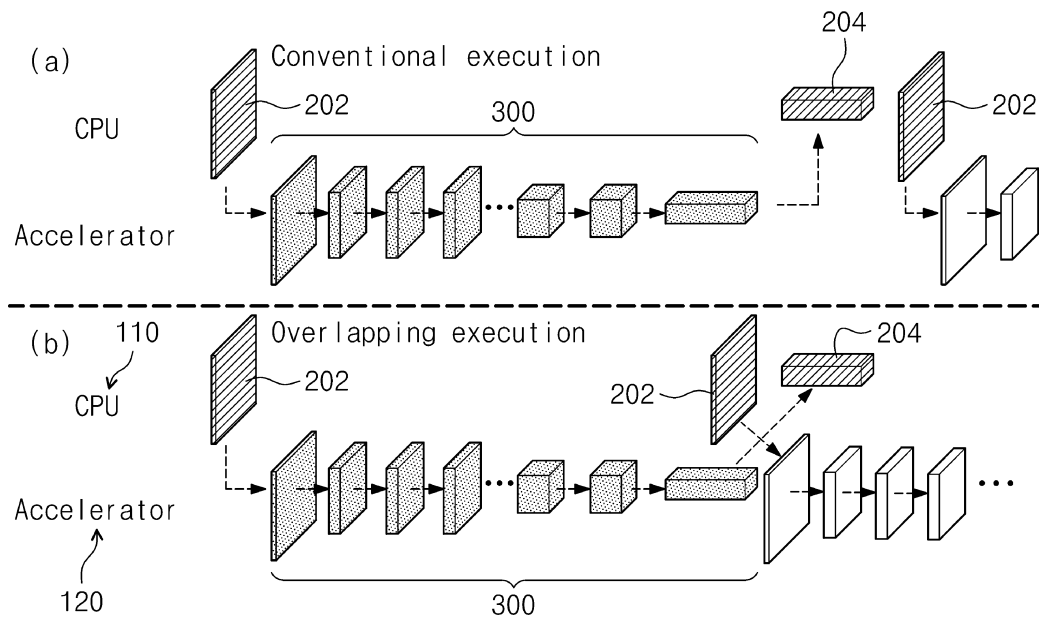
400: 401 ~ 403

도면3

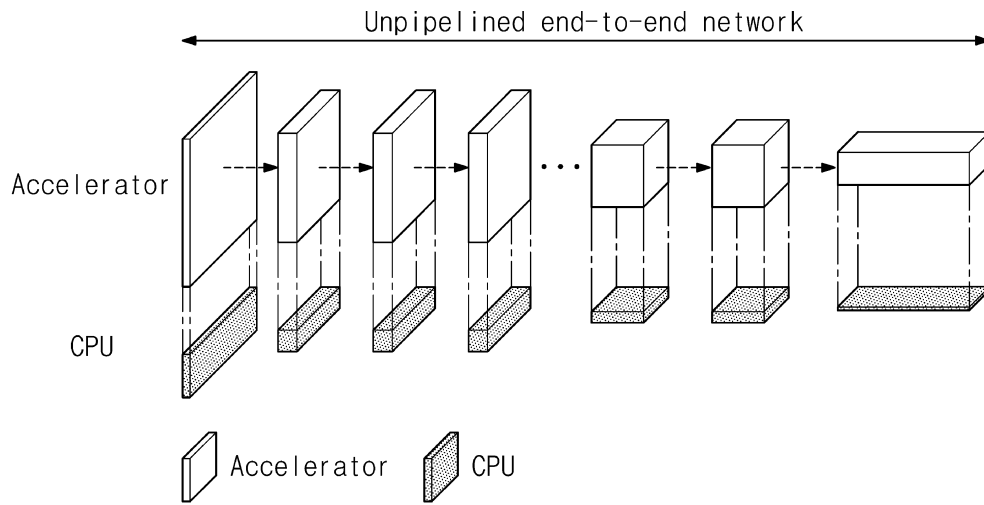


400: 401 ~ 403

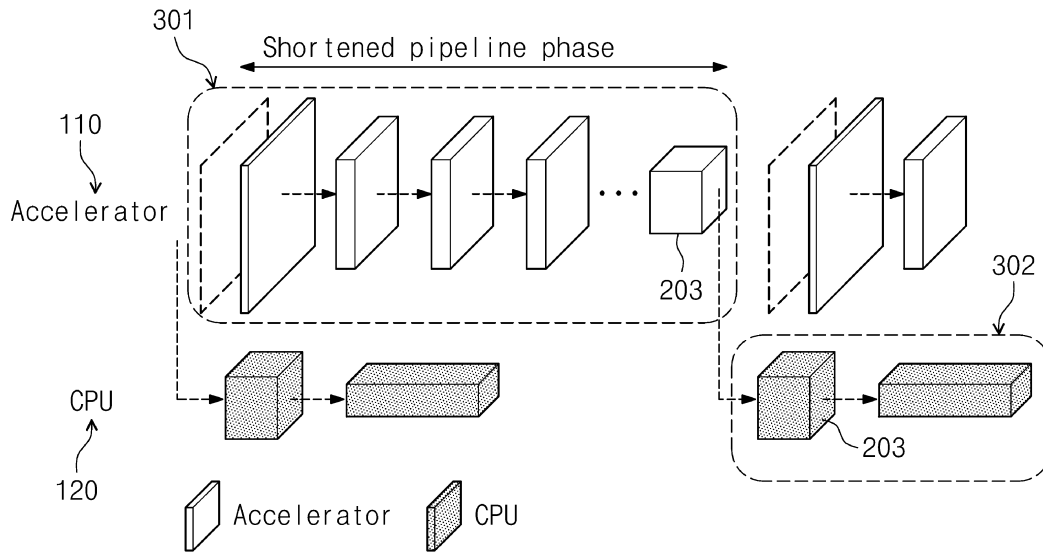
도면4



도면5



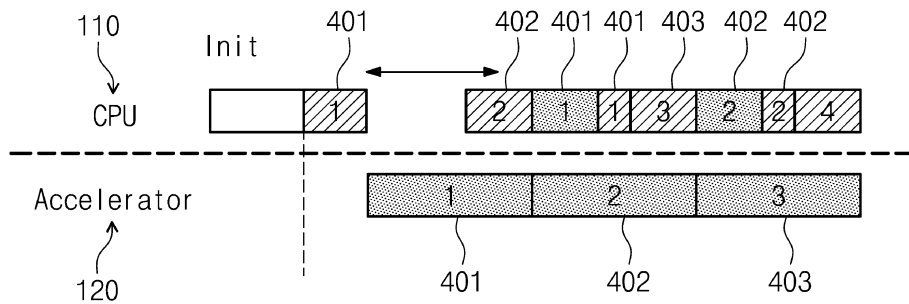
(a) inner-layer slicing and parallelization



(b) Network partitioning and pipelining

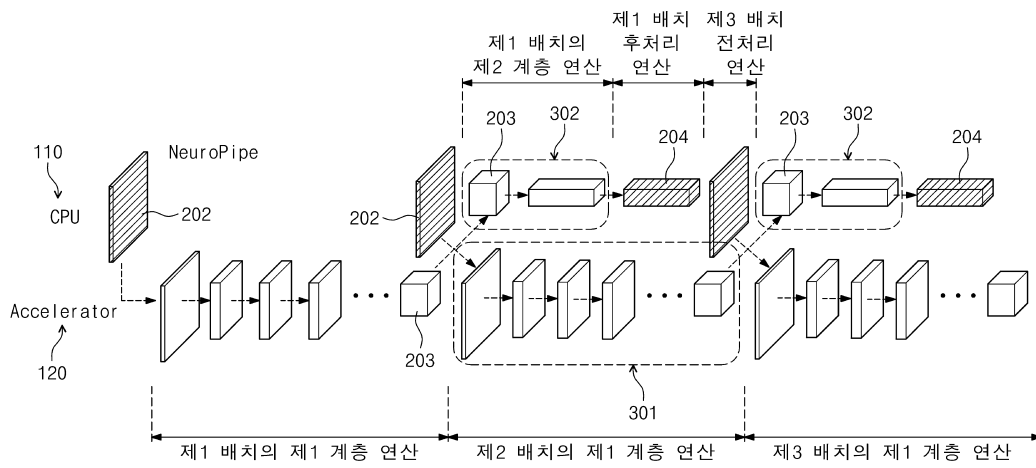
300: 301, 302

도면6



400: 401 ~ 403

도면7

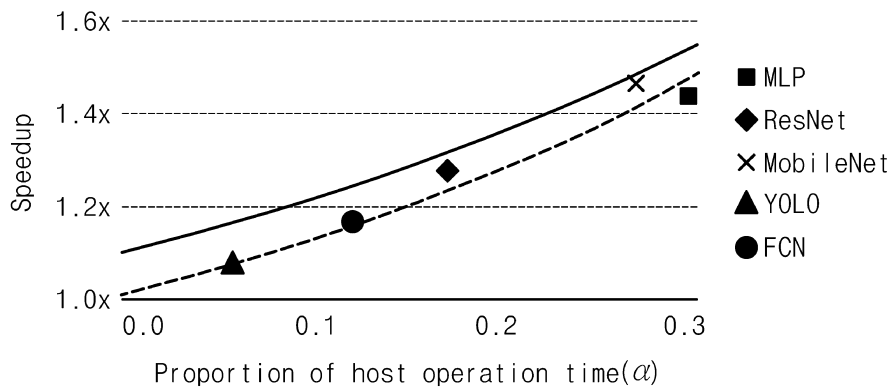


300: 301, 302

도면8

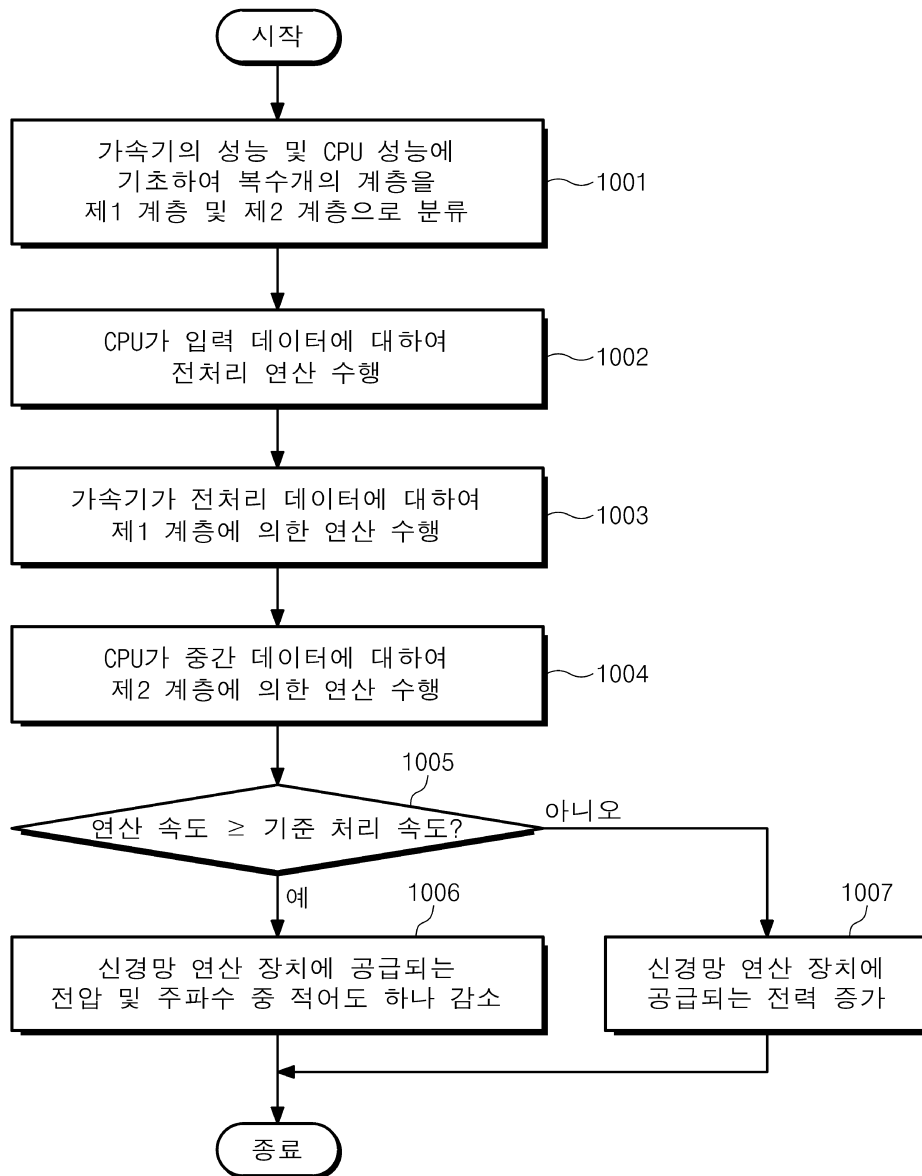
Network	Operation count (GFLOPS)	Memory usage (MB)	Model parameters	
			α	ν
MLP[2]	0.02	121.25	0.32	11.0
ResNet[8]	7.2	246.76	0.18	18.7
MobileNet[25]	0.61	161.53	0.29	17.9
YOLO[23]	44.9	771.25	0.07	37.4
FCN[15]	11.1	463.50	0.13	36.8

(a)



(b)

도면9



도면10

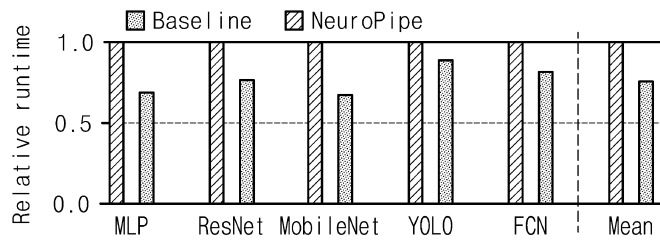
(a) Comparison of execution methods for MobileNet.

Execution method	Baseline	Slicing	Pipelining (NeuroPipe)
Inference interval	40.7ms	45.9ms	28.0ms
Neural computation	29.3ms	28.5ms	31.2ms
Communication overhead	11.4ms	17.4ms	12.1ms
Memory requirement	161.5MB	285.7MB	161.5MB

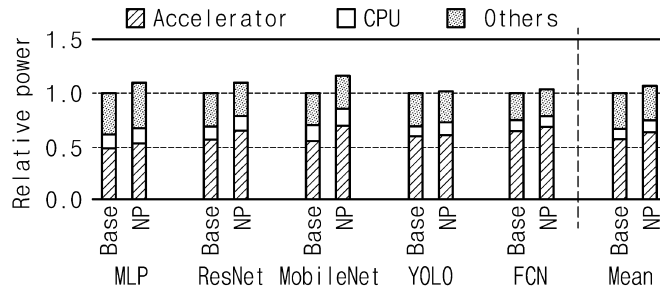
(b) Comparison of pipelining methods for ResNet.

Execution method		Baseline	Pu-oblivious	PU-aware (NeuroPipe)
Inference interval		67.9ms	61.8ms	53.1ms
Host operation(CPU)		11.9ms	14.2ms	13.4ms
Neural computation	CPU	–	47.6ms	30.8ms
	Accelerator	56.0ms	49.7ms	53.1ms

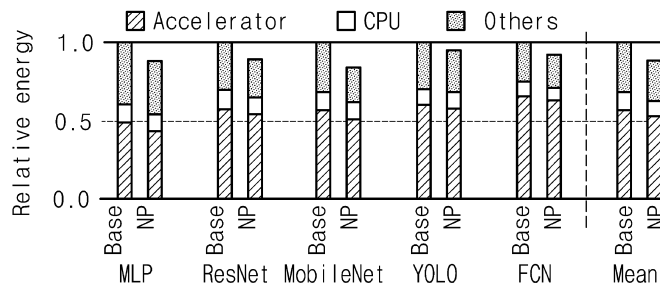
도면11



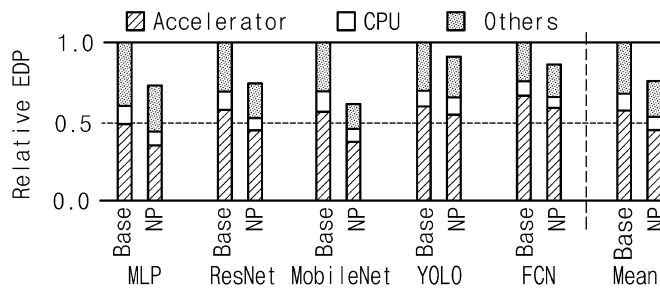
(a) Execution time



(b) Power dissipation

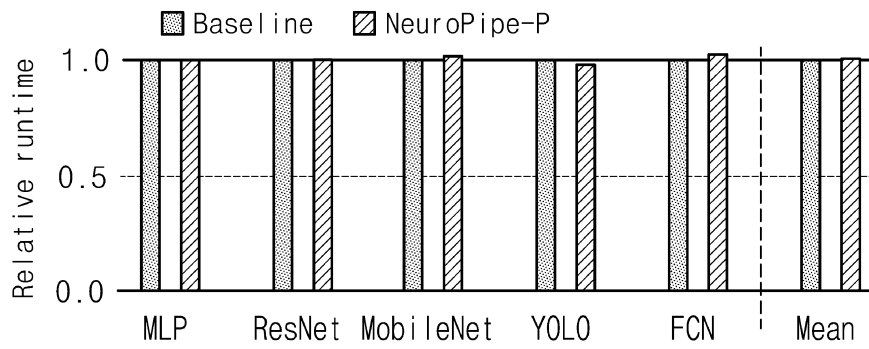


(c) Energy consumption

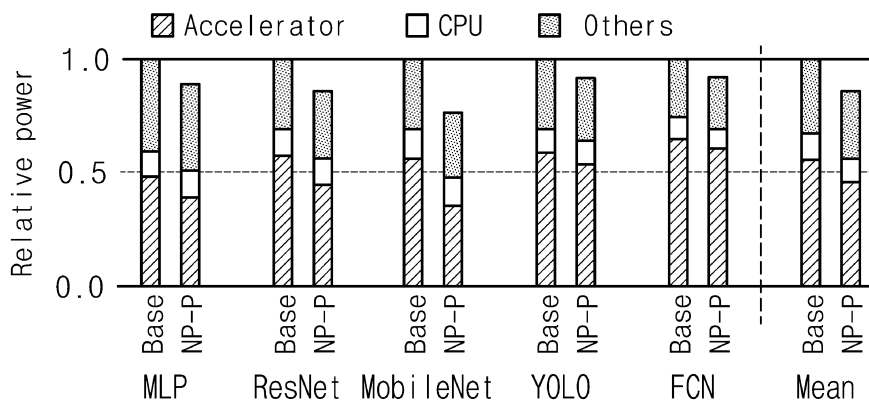


(d) Energy-delay product(EDP)

도면12



(a)Execution time



(b)Power dissipation