



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0094350
(43) 공개일자 2025년06월25일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2023.01) G06F 7/483 (2006.01)
G06N 3/08 (2023.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06F 7/483 (2021.08)
(21) 출원번호 10-2023-0185069
(22) 출원일자 2023년12월18일
심사청구일자 2023년12월18일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
궁재하
서울특별시 성북구 안암로 145 (안암동5가) 신공학관 501호
이승현
서울특별시 성북구 안암로 145 (안암동5가) 신공학관 501호
(74) 대리인
리앤목특허법인

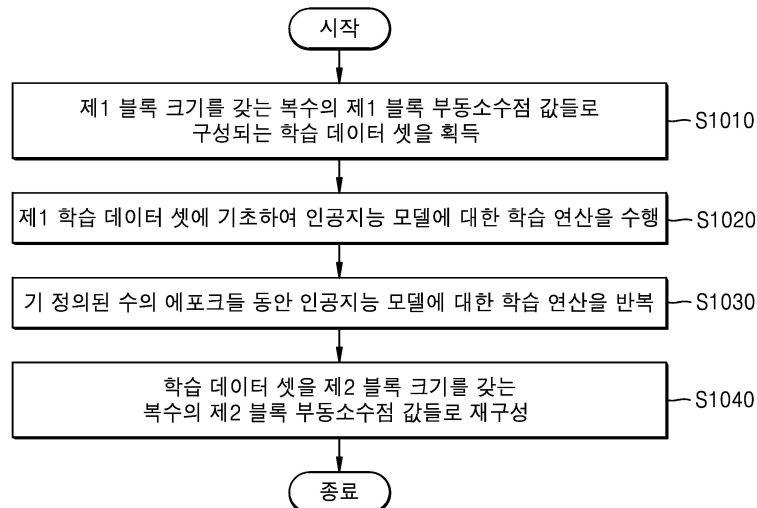
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법 및 이를 수행하는 하드웨어 가속기 및 전자 장치

(57) 요약

본 개시의 일 실시예에 따른 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법은, 제1 블록 크기를 갖는 복수의 제1 블록 부동소수점 값들로 구성되는 학습 데이터 셋을 획득하는 단계, 기 학습 데이터 셋에 기초하여 인공지능 모델에 대한 학습 연산을 수행하는 단계, 기 정의된 수의 에포크들 동안 인공지능 모델에 대한 학습 연산을 반복하는 단계, 및 학습 데이터 셋을 제2 블록 크기를 갖는 복수의 제2 블록 부동소수점 값들로 재구성하는 단계를 포함한다.

대표도 - 도10



(52) CPC특허분류
G06N 3/08 (2023.01)

명세서

청구범위

청구항 1

하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법에 있어서,
제1 블록 크기를 갖는 복수의 제1 블록 부동소수점 값들로 구성되는 학습 데이터 셋을 획득하는 단계;
상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계;
기 정의된 수의 에포크(epoch)들 동안 상기 인공지능 모델에 대한 학습 연산을 반복하는 단계; 및
상기 학습 데이터 셋을 제2 블록 크기를 갖는 복수의 제2 블록 부동소수점 값들로 재구성하는 단계를 포함하는,
방법.

청구항 2

제1항에 있어서,
상기 방법은:
상기 재구성된 학습 데이터 셋에 기초하여 상기 학습 연산을 수행하는 단계; 및
상기 기 정의된 수의 에포크들 동안 상기 인공지능 모델에 대한 학습 연산을 반복하는 단계를 포함하는, 방법.

청구항 3

제1항에 있어서,
상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계는:
상기 기 정의된 수의 에포크들 중 제1 에포크에서, 상기 학습 데이터 셋 중 기 정의된 수의 미니-배치들을 포함하는 제1 배치(batch)에 기초하여 상기 인공지능 모델의 제1 레이어(layer)에 대한 학습 연산을 수행하되, 상기 제1 배치는 제1 정밀도를 갖는, 단계;
부동소수점 형식의 상기 제1 배치에 대한 학습 연산 결과를 블록 부동소수점 형식으로 변환하는 단계;
상기 변환된 학습 연산 결과에 기초하여, 상기 학습 데이터 셋 중 상기 기 정의된 수의 미니-배치들을 포함하는 제2 배치의 제2 정밀도를 결정하는 단계; 및
상기 제1 에포크에서, 상기 제2 배치에 기초하여 상기 제1 레이어에 대한 학습 연산을 수행하는 단계를 포함하는, 방법.

청구항 4

제3항에 있어서,
상기 변환된 학습 연산 결과에 기초하여, 상기 학습 데이터 셋 중 상기 기 정의된 수의 미니-배치들을 포함하는 제2 배치의 제2 정밀도를 결정하는 단계는:
상기 변환된 학습 연산 결과에 대한 언더플로우(underflow) 발생 개수를 결정하는 단계; 및
상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 제2 정밀도를 결정하는 단계를 포함하는, 방법.

청구항 5

제4항에 있어서,

상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 제2 정밀도를 결정하는 단계는:

상기 결정된 언더플로우의 발생 개수가 제1 임계 개수보다 많은지를 결정하는 단계; 및

상기 결정된 언더플로우의 발생 개수가 상기 제1 임계 개수보다 많은 것으로 결정한 것에 기초하여, 상기 제2 정밀도를 상기 제1 정밀도보다 높은 정밀도로 결정하는 단계를 포함하는, 방법.

청구항 6

제4항에 있어서,

상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 제2 정밀도를 결정하는 단계는:

상기 결정된 언더플로우의 발생 개수가 제2 임계 개수보다 적은지를 결정하는 단계; 및

상기 결정된 언더플로우의 발생 개수가 상기 제2 임계 개수보다 적은 것으로 결정한 것에 기초하여, 상기 제2 정밀도를 상기 제1 정밀도보다 낮은 정밀도로 결정하는 단계를 포함하는, 방법.

청구항 7

제1항에 있어서,

상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계는:

상기 학습 데이터 셋에 대응하는 제1 텐서와 상기 인공지능 모델의 가중치 값에 대응하는 제2 텐서 간의 곱셈 연산을 수행하는 단계를 포함하는, 방법.

청구항 8

제7항에 있어서,

상기 곱셈 연산을 수행하는 단계는:

상기 제1 텐서의 공유 지수와 상기 제2 텐서의 공유 지수 간의 지수 곱셈 연산을 수행하는 단계;

상기 제1 텐서의 부호 및 맨티서와 상기 제2 텐서의 부호 및 맨티서 간의 MAC(multiply and accumulation) 연산을 수행하는 단계; 및

상기 지수 곱셈 연산 및 상기 MAC 연산에 기초하여 부동소수점 형식의 상기 곱셈 연산의 결과 값을 획득하는 단계를 포함하는, 방법.

청구항 9

제7항에 있어서,

상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계는:

상기 곱셈 연산의 결과 값을 블록 부동소수점 형식으로 변환하는 단계를 포함하는, 방법.

청구항 10

제7항에 있어서,

상기 제1 텐서의 정밀도와 상기 제2 텐서의 정밀도는 상이한, 방법.

청구항 11

적어도 하나의 인스트럭션을 저장하는 메모리; 및

상기 적어도 하나의 인스트럭션을 실행하는 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는:

제1 블록 크기를 갖는 복수의 제1 블록 부동소수점 값들로 구성되는 학습 데이터 셋을 획득하고,

상기 학습 데이터 셋에 기초하여 인공지능 모델에 대한 학습 연산을 수행하고,

기 정의된 수의 에포크(epoch)들 동안 상기 인공지능 모델에 대한 학습 연산을 반복하고,

상기 학습 데이터 셋을 제2 블록 크기를 갖는 복수의 제2 블록 부동소수점 값들로 재구성하는, 전자 장치.

청구항 12

제11항에 있어서,

상기 적어도 하나의 프로세서는:

상기 재구성된 학습 데이터 셋에 기초하여 상기 학습 연산을 수행하고,

상기 기 정의된 수의 에포크들 동안 상기 인공지능 모델에 대한 학습 연산을 반복하는, 전자 장치.

청구항 13

제11항에 있어서,

상기 적어도 하나의 프로세서는:

상기 기 정의된 수의 에포크들 중 제1 에포크에서, 상기 학습 데이터 셋 중 기 정의된 수의 미니-배치들을 포함하는 제1 배치(batch)에 기초하여 상기 인공지능 모델의 제1 레이어(layer)에 대한 학습 연산을 수행하되, 상기 제1 배치는 제1 정밀도를 갖고,

부동소수점 형식의 상기 제1 배치에 대한 학습 연산 결과를 블록 부동소수점 형식으로 변환하고,

상기 변환된 학습 연산 결과에 기초하여, 상기 학습 데이터 셋 중 상기 기 정의된 수의 미니-배치들을 포함하는 제2 배치의 제2 정밀도를 결정하고,

상기 제1 에포크에서, 상기 제2 배치에 기초하여 상기 제1 레이어에 대한 학습 연산을 수행하는, 전자 장치.

청구항 14

제13항에 있어서,

상기 적어도 하나의 프로세서는:

상기 변환된 학습 연산 결과에 대한 언더플로우(underflow) 발생 개수를 결정하고,

상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 제2 정밀도를 결정하는, 전자 장치.

청구항 15

제14항에 있어서,

상기 적어도 하나의 프로세서는:

상기 결정된 언더플로우의 발생 개수가 제1 임계 개수보다 많은지를 결정하고,

상기 결정된 언더플로우의 발생 개수가 상기 제1 임계 개수보다 많은 것으로 결정한 것에 기초하여, 상기 제2 정밀도를 상기 제1 정밀도보다 높은 정밀도로 결정하는, 전자 장치.

청구항 16

제14항에 있어서,

상기 적어도 하나의 프로세서는:

상기 결정된 언더플로우의 발생 개수가 제2 임계 개수보다 적은지를 결정하고,

상기 결정된 언더플로우의 발생 개수가 상기 제2 임계 개수보다 적은 것으로 결정한 것에 기초하여, 상기 제2 정밀도를 상기 제1 정밀도보다 낮은 정밀도로 결정하는, 전자 장치.

청구항 17

제11항에 있어서,

상기 적어도 하나의 프로세서는:

상기 학습 데이터 셋에 대응하는 제1 텐서와 상기 인공지능 모델의 가중치 값에 대응하는 제2 텐서 간의 곱셈 연산을 수행하는, 전자 장치.

청구항 18

제17항에 있어서,

상기 적어도 하나의 프로세서는:

상기 제1 텐서의 공유 지수와 상기 제2 텐서의 공유 지수 간의 지수 곱셈 연산을 수행하고,

상기 제1 텐서의 부호 및 맨티서와 상기 제2 텐서의 부호 및 맨티서 간의 MAC(multiply and accumulation) 연산을 수행하고,

상기 지수 곱셈 연산 및 상기 MAC 연산에 기초하여 부동소수점 형식의 상기 곱셈 연산의 결과 값을 획득하는, 전자 장치.

청구항 19

제17항에 있어서,

상기 적어도 하나의 프로세서는:

상기 곱셈 연산의 결과 값을 블록 부동소수점 형식으로 변환하는, 전자 장치.

청구항 20

제1항 내지 제10항 중 어느 한 항의 방법을 컴퓨터에서 수행하기 위한 프로그램이 기록된 컴퓨터로 읽을 수 있는 기록매체.

발명의 설명

기술 분야

- [0001] 본 개시는 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법 및 이를 수행하는 하드웨어 가속기 및 전자 장치에 관한 것으로, 보다 구체적으로는 암시적 비트를 활용하는 블록 부동 소수점 형식과, 이를 이용하여 다양한 정밀도 및 다양한 블록 크기로 학습 및 추론 동작을 지원하는 하드웨어 가속기 및 전자 장치에 관한 것이다.
- [0002] 본 연구는 삼성미래기술육성사업의 지원을 받아 수행되었음 (과제번호: SRFC-IT1902-03)

배경 기술

- [0004] 고성능 컴퓨팅 시스템 및 끊임없이 성장한 오픈 소스 데이터 세트의 영향으로 인공지능 기술은 매우 빠르게 발전하였다. 그리고 정확도 향상에 따라 컴퓨터 비전 분야, 언어 모델링, 자율 주행 등 많은 응용 분야에서 인공지능 기술이 활용되고 있다.
- [0005] 인공지능을 응용 분야에서 사용하기 위해선, 학습이라는 과정이 필요하다. 인공지능 기술에서 학습은 특정 데이터 셋을 통해 인공지능 모델(예컨대, 심층 신경망(Deep neural networks, DNNs))의 가중치를 갱신하는 과정을 말한다. 가중치가 잘 갱신될수록, 인공지능 모델은 주어진 태스크를 잘 수행할 수 있다.
- [0006] 그러나 학습 과정은 요구되는 계산량이 매우 많아 CPU 상에서 훈련을 진행하는 경우, 매우 많은 시간이 소요된다. GPU의 경우, 병렬 처리에 용이하여 CPU 보다 소모되는 시간이 적지만, 구조 특성상 낮은 이용률을 보인다. 최근에는 CPU 및 GPU의 단점들을 극복하고자, 심층 신경망의 연산을 수행하는 전용 하드웨어 가속기들이 많이 제안되고 있다.

발명의 내용

과제의 해결 수단

- [0008] 본 개시의 일 실시예에 있어서, 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법이 제공될 수 있다. 상기 방법은, 제1 블록 크기를 갖는 복수의 제1 블록 부동소수점 값들로 구성되는 학습 데이터 셋을 획득하는 단계를 포함할 수 있다. 상기 방법은, 상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계를 포함할 수 있다. 상기 방법은, 기 정의된 수의 에포크(epoch)들 동안 상기 인공지능 모델에 대한 학습 연산을 반복하는 단계를 포함할 수 있다. 상기 방법은, 상기 학습 데이터 셋을 제2 블록 크기를 갖는 복수의 제2 블록 부동소수점 값들로 재구성하는 단계를 포함할 수 있다.
- [0009] 본 개시의 일 실시예에 있어서, 전자 장치가 제공될 수 있다. 상기 전자 장치는 적어도 하나의 인스트럭션을 저장하는 메모리를 포함할 수 있다. 상기 전자 장치는 상기 적어도 하나의 인스트럭션을 실행하는 적어도 하나의 프로세서를 포함할 수 있다. 상기 적어도 하나의 프로세서는, 제1 블록 크기를 갖는 복수의 제1 블록 부동소수점 값들로 구성되는 학습 데이터 셋을 획득할 수 있다. 상기 적어도 하나의 프로세서는, 상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행할 수 있다. 상기 적어도 하나의 프로세서는, 기 정의된 수의 에포크(epoch)들 동안 상기 인공지능 모델에 대한 학습 연산을 반복할 수 있다. 상기 적어도 하나의 프로세서는, 상기 학습 데이터 셋을 제2 블록 크기를 갖는 복수의 제2 블록 부동소수점 값들로 재구성할 수 있다.
- [0010] 본 개시의 일 실시예에 있어서, 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법을 컴퓨터에서 수행하기 위한 프로그램이 기록된 컴퓨터로 읽을 수 있는 기록매체가 제공될 수 있다.

도면의 간단한 설명

- [0012] 본 개시는, 다음의 자세한 설명과 그에 수반되는 도면들의 결합으로 쉽게 이해될 수 있으며, 참조 번호(reference numerals)들은 구조적 구성요소(structural elements)를 의미한다.

도 1은 본 개시의 일 실시예에 따른 인공지능 모델의 연산 환경을 설명하기 위한 개념도이다.

도 2는 본 개시의 일 실시예에 따른 블록 부동소수점을 설명하기 위한 개념도이다.

도 3은 본 개시의 일 실시예에 따른 전자 장치(1000)의 구성을 설명하기 위한 블록도이다.

도 4는 본 개시의 일 실시예에 따른 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법을 설명하기 위한 개념도이다.

도 5는 본 개시의 일 실시예에 따른 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법을 보여주는 흐름도이다.

도 6은 본 개시의 일 실시예에 따른 비트 폭 또는 정밀도를 변경하는 방법을 보여주는 흐름도이다.

도 7은 도 6의 단계 S630의 세부 단계들을 보여주는 흐름도이다.

도 8은 본 개시의 일 실시예에 따른 블록 크기를 변경하는 방법을 보여주는 흐름도이다.

도 9는 본 개시의 일 실시예에 따른 블록 부동소수점 값을 처리하는 방법을 보여주는 흐름도이다.

도 10은 본 개시의 일 실시예에 따른 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법을 보여주는 흐름도이다.

도 11은 본 개시의 일 실시예에 따른 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법을 보여주는 흐름도이다.

도 12는 도 10의 단계 S1020의 세부 단계들을 보여주는 흐름도이다.

도 13은 도 12의 단계 S1230의 세부 단계들을 보여주는 흐름도이다.

도 14는 도 13의 단계 S1320의 세부 단계들을 보여주는 흐름도이다.

도 15는 본 개시의 일 실시예에 따른 하드웨어 가속기의 구성을 설명하기 위한 블록도이다.

도 16은 본 개시의 일 실시예에 따른 서브 코어의 구성 및 동작을 예시적으로 설명하기 위한 블록도이다.

도 17은 본 개시의 일 실시예에 따른 서브 코어의 구성 및 동작을 예시적으로 설명하기 위한 블록도이다.

도 18은 본 개시의 일 실시예에 따른 서브 코어의 구성 및 동작을 예시적으로 설명하기 위한 블록도이다.

도 19a 및 19b는 본 개시의 일 실시예에 따른 선택적 가산기 트리(1412)의 구성 및 동작을 예시적으로 설명하기 위한 블록도들이다.

도 20은 본 개시의 일 실시예에 따른 전자 장치(1000)의 동작을 설명하기 위한 개념도이다.

발명을 실시하기 위한 구체적인 내용

[0013] 본 명세서의 실시예들에서 사용되는 용어는 본 개시의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어들을 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도 또는 관례, 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한, 특정한 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 실시예의 설명 부분에서 상세히 그 의미를 기재할 것이다. 따라서 본 명세서에서 사용되는 용어는 단순한 용어의 명칭이 아닌, 그 용어가 가지는 의미와 본 개시의 전반에 걸친 내용을 토대로 정의되어야 한다.

[0014] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함할 수 있다. 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 용어들은 본 명세서에 기재된 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가질 수 있다.

[0015] 본 개시 전체에서 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있음을 의미한다. 또한, 본 명세서에 기재된 "...부", "...모듈" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어 또는 소프트웨어로 구현되거나 하드웨어와 소프트웨어의 결합으로 구현될 수 있다.

[0016] 본 개시에서 사용된 표현 "~하도록 구성된(또는 설정된)(configured to)"은 상황에 따라, 예를 들면, "~에 적합한(suitable for)", "~하는 능력을 가지는(having the capacity to)", "~하도록 설계된(designed to)", "~하도

록 변경된(adapted to)", "~하도록 만들어진(made to)", 또는 "~를 할 수 있는(capable of)"과 바꾸어 사용될 수 있다. 용어 "~하도록 구성된(또는 설정된)"은 하드웨어적으로 "특별히 설계된(specifically designed to)" 것만을 반드시 의미하지 않을 수 있다. 대신, 어떤 상황에서는, "~하도록 구성된 시스템"이라는 표현은, 그 시스템이 다른 장치 또는 부품들과 함께 "~할 수 있는" 것을 의미할 수 있다. 예를 들면, 문구 "A, B, 및 C를 수행하도록 구성된(또는 설정된) 프로세서"는 해당 동작을 수행하기 위한 전용 프로세서(예: 임베디드 프로세서), 또는 메모리에 저장된 하나 이상의 소프트웨어 프로그램들을 실행함으로써, 해당 동작들을 수행할 수 있는 범용 프로세서(generic-purpose processor)(예: CPU 또는 application processor)를 의미할 수 있다.

[0017] 또한, 본 개시에서 일 구성요소가 다른 구성요소와 "연결된다" 거나 "접속된다" 등으로 언급된 때에는, 상기 일 구성요소가 상기 다른 구성요소와 직접 연결되거나 또는 직접 접속될 수도 있지만, 특별히 반대되는 기재가 존재하지 않는 이상, 중간에 또 다른 구성요소를 매개하여 연결되거나 또는 접속될 수도 있다고 이해되어야 할 것이다.

[0018] 본 개시에서, '인공지능(Artificial Intelligence)'과 관련된 기능은 프로세서와 메모리를 통해 동작된다. 프로세서는 하나 또는 복수의 프로세서로 구성될 수 있다. 이때, 하나 또는 복수의 프로세서는 CPU, AP, DSP(Digital Signal Processor) 등과 같은 범용 프로세서, GPU, VPU(Vision Processing Unit)와 같은 그래픽 전용 프로세서 또는 NPU와 같은 인공지능 전용 프로세서일 수 있다. 하나 또는 복수의 프로세서는, 메모리에 저장된 기 정의된 동작 규칙 또는 인공지능 모델에 따라, 입력 데이터를 처리하도록 제어한다. 또는, 하나 또는 복수의 프로세서가 인공지능 전용 프로세서인 경우, 인공지능 전용 프로세서는, 특정 인공지능 모델의 처리에 특화된 하드웨어 구조로 설계될 수 있다.

[0019] 기 정의된 동작 규칙 또는 인공지능 모델은 학습을 통해 만들어진 것을 특징으로 한다. 여기서, 학습을 통해 만들어진다는 것은, 기본 인공지능 모델이 학습 알고리즘에 의하여 다수의 학습 데이터들을 이용하여 학습됨으로써, 원하는 특성(또는, 목적)을 수행하도록 설정된 기 정의된 동작 규칙 또는 인공지능 모델이 만들어짐을 의미한다. 이러한 학습은 본 개시에 따른 인공지능이 수행되는 기기 자체에서 이루어질 수도 있고, 별도의 서버 및/또는 시스템을 통해 이루어 질 수도 있다. 학습 알고리즘의 예로는, 지도형 학습(supervised learning), 비지도형 학습(unsupervised learning), 준지도형 학습(semi-supervised learning) 또는 강화 학습(reinforcement learning)이 있으나, 전술한 예에 한정되지 않는다.

[0020] 본 개시의 일 실시예에 있어서, '인공지능 모델'은 신경망 모델을 포함할 수 있다. 신경망 모델은 복수의 신경망 레이어들로 구성될 수 있다. 복수의 신경망 레이어들 각각은 복수의 가중치들(weight values)을 갖고 있으며, 이전(previous) 레이어의 연산 결과와 복수의 가중치들 간의 연산을 통해 신경망 연산을 수행한다. 복수의 신경망 레이어들이 갖고 있는 복수의 가중치들은 인공지능 모델의 학습 결과에 의해 최적화될 수 있다. 예를 들어, 학습 과정 동안 인공지능 모델에서 획득한 손실(loss) 값 또는 비용(cost) 값이 감소 또는 최소화되도록 복수의 가중치들이 갱신될 수 있다. 인공 신경망 모델은 심층 신경망(DNN: Deep Neural Network)를 포함할 수 있으며, 예를 들어, CNN(Convolutional Neural Network), RNN(Recurrent Neural Network), RBM(Restricted Boltzmann Machine), DBN(Deep Belief Network), BRDNN(Bidirectional Recurrent Deep Neural Network) 또는 심층 Q-네트워크(Deep Q-Networks) 등이 있으나, 전술한 예에 한정되지 않는다.

[0021] 본 개시에서, 텐서(tensor)는 n-차원의 데이터의 배열을 나타낼 수 있다. 텐서의 각 차원은 하나의 축(axis)을 형성할 수 있다. 예를 들어, 0차원의 텐서는 스칼라 값을 나타내고, 1차원의 텐서는 벡터 값을 나타내고, 2차원의 텐서는 행렬을 나타낼 수 있다. 예를 들어, 3 이상의 차원의 텐서는 복수의 행렬들로 구성되고, 3 개 이상의 축들을 형성할 수 있다.

[0022] 본 개시에서, 배치(batch)는 인공지능 모델의 학습을 위한 전체 데이터 셋을 그룹핑하여 파라미터를 업데이트하는 하나의 단위를 나타낼 수 있다. 예를 들어, 전체 데이터 셋은 복수의 배치들로 그룹핑할 수 있다. 인공지능 모델의 파라미터들은 하나의 배치 마다 업데이트될 수 있다. 하나의 배치는 기 정의된 수의 미니-배치들을 포함할 수 있다.

[0023] 이하에서 설명하는 구체적인 수학식은 가능한 여러 대안 중에서 예시적으로 설명되는 것이며, 본 개시의 권리 범위가 본 개시에 언급된 수학식에 제한되는 것으로 해석되어서는 아니된다.

[0024] 아래에서는 첨부한 도면을 참고하여 본 개시의 실시예들에 대하여 본 개시가 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 개시는 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시 예에 한정되지 않는다.

- [0026] 도 1은 본 개시의 일 실시예에 따른 인공지능 모델의 연산 환경을 설명하기 위한 개념도이다.
- [0027] 본 개시의 일 실시예에 있어서, 프로세서(200)는 인공지능 모델의 학습을 위해 입력 값을 수신할 수 있다. 인공지능 모델은 적어도 하나의 레이어를 포함할 수 있다. 적어도 하나의 레이어는 복수의 가중치 값들 및/또는 복수의 편향 값들로 구성될 수 있다. 입력 값, 가중치 값, 및 편향 값 중 적어도 하나는 기 정의된 데이터 형식(format)으로 구성될 수 있다. 예를 들어, 기 정의된 데이터 형식은 FP32일 수 있으나, 본 개시는 이에 한정되지 않으며, 기 정의된 데이터 형식은 다양한 데이터 형식으로 구성될 수 있다.
- [0028] 다만 모든 데이터에 FP32를 이용하는 경우, 큰 용량의 메모리(120)가 요구되며, 해당 데이터의 송수신 과정에서의 높은 통신 비용이 발생하게 된다. 본 개시의 일 실시예에 있어서, 기 정의된 데이터 형식의 값들은 더 효율적인 데이터 형식으로 양자화될 수 있다. 본 개시의 일 실시예에 따르면, 프로세서(200)에 입력 값, 가중치 값, 및/또는 편향 값이 입력되기 전에, 실시간으로 변하는 데이터 요구 사항에 더욱 효과적으로 대응하고, 전력 소모 및 처리 속도에 있어서 최적의 결과를 얻을 수 있다.
- [0029] 본 개시의 일 실시예에 있어서, 기 정의된 데이터 형식은, bfloat32, FP16, 및/또는 FP8 등으로 양자화될 수 있다. 프로세서(200)는 양자화된 값들을 수신하여, 입력 값과 가중치 값 간의 곱셈 연산을 수행할 수 있다. 프로세서(200)는 곱셈 연산 결과를 활용하여 인공지능 모델의 파라미터(예컨대, 가중치 값)를 갱신할 수 있다. 프로세서(200)는 갱신된 파라미터를 메모리(120)에 저장할 수 있다.
- [0030] 그러나, 도 1에 도시된 부동소수점 형식(예컨대, FP32, FP16, FP8, bfloat32 등)으로 인공지능 모델의 학습을 수행하는 경우, INT 형식에 비해 큰 메모리 공간이 요구되며, 전력 소비가 크다는 문제가 있다.
- [0031] 본 개시의 일 실시예에 따른 하드웨어 가속기는, 부동소수점 형식의 값들을 블록 부동소수점(block floating point) 형식으로 변환할 수 있다. 본 개시의 일 실시예에 따른 하드웨어 가속기는 블록 부동소수점 형식의 값들의 부호 및 맨티사에 대해 INT 형식의 MAC(multiply and accumulation) 연산을 수행할 수 있다. 본 개시의 일 실시예에 따른 하드웨어 가속기의 구체적인 구성, 기능 및 동작과, 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법에 대해서는 이하에서 설명한다.
- [0033] 도 2는 본 개시의 일 실시예에 따른 블록 부동소수점을 설명하기 위한 개념도이다.
- [0034] 본 개시에서는 블록 부동소수점 형식으로 연산을 수행한다. 블록 부동소수점 형식을 설명하게 앞서 부동소수점 값에 대한 표현 방식에 대해서 설명한다.
- [0035] 부동소수점(floating point) 값은 정밀도에 따라 다양한 형태가 있을 수 있는데 가장 많이 활용되는 것이 IEEE에서 규정한 형식인 FP32(20)이다. FP32(20)는 실수를 부호(sign), 지수(exponent), 맨티사(mantissa)로 구분하여 표시한다. 예를 들어, 부동소수점 값은 다음과 같은 수학적 식 1과 같이 표현될 수 있다.

수학적 식 1

$$x_i = (-1)^{s_i} \cdot m_i \cdot 2^{e_i}$$

- [0036]
- [0037] 여기서, x_i 는 실수 값, s_i 는 부호, m_i 는 맨티사, e_i 는 실수에 대한 지수 값으로 정의된다.
- [0038] 한편, FP32(20)에서는 부호를 1 비트로 표시하고, 맨티사를 23 비트로 표시하고, 지수 값을 8 비트로 표시하여, 총 32 비트로 부동소수점 값을 표시한다.
- [0039] 그러나 모든 값들에 대해서 32 비트로 데이터를 처리하는 경우, 처리 과정에서 많은 리소스가 낭비될 수 있다. 본 개시에서는 블록 부동소수점(Block Floating Point)(또는 BFP로도 지칭될 수 있음)이라는 개념을 도입하여, 여러 데이터를 동일한 지수로 표현한다.
- [0040] 구체적으로, BFP(30)는 하나의 블록 내의 N개의 값들의 지수들 중 가장 큰 지수(32)를 공유한다. 즉, 도 2의 우측과 같이 여러 값이 동일한 지수 값(shared exponents)(32)을 갖도록 하고, 각 값(31)들은 부호와 해당 공유 지수 값에 대한 맨티사 값만을 갖도록 하였다. 예를 들어, 복수의 실수 값은 아래와 같은 수학적 식 2와 같이 표현

될 수 있다.

수학식 2

[0041]
$$\vec{x} = [x_1, x_2, \dots, x_N] = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N] \cdot 2^{e_s} = \tilde{\vec{x}} \cdot 2^{e_s}$$

[0042] 여기서, e_s 는 블록의 공유 지수 값으로, $e_s = \lfloor \log_2(\max(|x_1|, \dots, |x_N|)) \rfloor$ 이다. 본 개시의 일 실시예에 있어서, 각각의 값들은 아래의 수학식 3과 같이 부호 및 맨티서만으로 구성된다.

수학식 3

[0043]
$$\hat{x}_i = \{s_i, m_i\}$$

[0044] 이와 같이 블록 부동소수점을 사용함으로써, 데이터 저장 공간을 크게 줄일 수 있을 뿐만 아니라, 실수 연산을 정수 산술만으로 연산을 수행할 수 있다.

[0045] 또한, 정밀도에 따라서, 부호 및 맨티서가 다양한 크기를 지원하도록 하였다. 예를 들어, 부호와 맨티서의 크기는 4 비트, 8 비트, 16 비트를 가질 수 있다. 본 개시의 일 실시예에 있어서, 부호 및 맨티서의 크기는 학습 과정에서의 단계별로 다르게 적용될 수 있다. 예를 들어, 피쳐 맵 계산시에는 4 비트의 부호 및 맨티서를 이용하고, 로컬 경사 계산 또는 가중치 업데이트시에는 8 비트 또는 16 비트의 부호 및 맨티서를 이용할 수 있다.

[0046] 이와 같이 본 개시에서는 다양한 크기의 맨티서(또는 부호 및 맨티서)를 지원하기 때문에, 본 개시에 따라 생성된 모델은 동일한 지수 비트를 사용하는 정밀도(예를 들어, bfloat 16과 같은) CPU/GPU를 지원하는 가속기에서도 실행하는 것이 가능하다. 또한, 지수를 관리하는 핸들러를 제어함으로써 4 비트, 8비트, 16비트 크기의 정수 연산도 수행하는 것이 가능하다.

[0048] 도 3은 본 개시의 일 실시예에 따른 전자 장치(1000)의 구성을 설명하기 위한 블록도이다.

[0049] 도 3을 참조하면, 전자 장치(1000)는 입출력 인터페이스(1100), 메모리(1200), 프로세서(1300), 및 하드웨어 가속기(1400)를 포함할 수 있다. 이러한 전자 장치는 PC, 노트북, 서버와 같은 장치뿐만 아니라, 스마트폰과 같은 모바일 장치일 수도 있다.

[0050] 입출력 인터페이스(1100)는 입력 인터페이스와 출력 인터페이스를 포함할 수 있다.

[0051] 본 개시의 일 실시예에 있어서, 입력 인터페이스는, 사용자로부터의 입력(이하에서, 사용자 입력)을 수신하기 위한 것이다. 입력 인터페이스는 키 패드(key pad), 돔 스위치 (dome switch), 터치 패드(접촉식 정전 용량 방식, 압력식 저항막 방식, 적외선 감지 방식, 표면 조음과 전도 방식, 적분식 장력 측정 방식, 피에조 효과 방식 등), 조그 휠, 조그 스위치, 마이크 중 적어도 하나일 수 있으나, 이에 한정되는 것은 아니다.

[0052] 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 입력 인터페이스를 통해 인공지능 모델의 하이퍼파라미터(hyperparameter)에 대응하는 사용자 입력을 수신할 수 있다. 예를 들어, 하이퍼파라미터는 학습 정밀도, 블록 부동소수점의 블록 크기, 정밀도 변경의 빈도, 블록 크기 변경의 빈도, 학습률, 에포크 수, 배치 크기, 활성화 함수, 드롭아웃 비율, 정규화 파라미터 등을 포함할 수 있으나, 본 개시는 이에 한정되지 않고, 인공지능 모델의 학습 또는 추론에 필요한 정보를 포함할 수 있다. 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 입력 인터페이스를 통해 인공지능 모델의 입력 값에 대응하는 사용자 입력을 수신할 수 있다. 입력 인터페이스를 통해 수신된 사용자 입력은 프로세서(1300)에 전달될 수 있다. 프로세서(1300)는 사용자 입력에 기초하여 인공지능 모델의 학습 및/또는 추론을 수행할 수 있다.

[0053] 본 개시의 일 실시예에 있어서, 출력 인터페이스는, 오디오 신호 또는 비디오 신호의 출력을 위한 것으로, 예컨대 디스플레이 또는 스피커 등을 포함할 수 있다. 디스플레이는 전자 장치(1000)가 지원하는 기능을 선택받기 위한 사용자 인터페이스 창을 표시할 수 있다. 구체적으로, 디스플레이는 전자 장치(100)가 제공하는 각종 기능

을 선택받기 위한 사용자 인터페이스 창을 표시할 수 있다. 이러한 디스플레이(130)는 LCD(liquid crystal display), OLED(Organic Light Emitting Diodes) 등과 같은 모니터일 수 있으며, 터치스크린으로 구현될 수도 있다.

[0054] 디스플레이는 학습 과정에 적용될 각종 파라미터의 입력을 요청하는 메시지를 표시할 수 있다. 구현시에 해당 파라미터는 사용자가 직접 입력할 수도 있으며, 인공지능 모델의 특성 및 데이터 셋의 특성에 따라 자동적으로 선택될 수도 있다.

[0055] 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 디스플레이를 통해 인공지능 모델의 하이퍼파라미터를 입력하는 사용자 인터페이스 창을 표시할 수 있다. 전자 장치(1000)는 디스플레이를 통해 사용자에게 인공지능 모델의 학습 경과를 표시할 수 있다.

[0056] 메모리(1200)는 프로세서(1300) 및/또는 하드웨어 가속기(1400)의 처리 및 제어를 위한 프로그램을 저장할 수도 있고, 입/출력되는 데이터들을 저장할 수도 있다. 메모리(1200)는 전자 장치(1000)에 관한 적어도 하나의 인스트럭션을 저장할 수 있다. 메모리(1200)는 적어도 하나의 인공지능 모델을 저장할 수도 있다. 예를 들어, 메모리(1200)는 머신 러닝 또는 딥 러닝에 필요한 심층 신경망을 저장할 수 있다. 여기서 심층 신경망은 딥 러닝 네트워크(Deep learning networks)일 수 있으나, 해당 용어에 한정되지 않고, 데이터 세트를 통하여 내부 가중치의 업데이트가 수행될 수 있는 모델이라면 다양한 모델이 적용될 수 있다.

[0057] 본 개시의 일 실시예에 있어서, 메모리(1200)에는 프로세서(1300)에 의하여 처리되거나 처리될 예정인 데이터, 펌웨어, 소프트웨어, 및 프로세스 코드 등이 저장될 수 있다. 본 개시의 일 실시예에 있어서, 메모리(1200)에는 인공지능 모델을 학습시키기 위한 학습 데이터 셋이 저장될 수 있다. 본 개시의 일 실시예에 있어서, 메모리(1200)에는 인공지능 모델의 학습 및/또는 추론에 필요한 데이터 및 프로그램 코드들이 저장될 수 있다. 본 개시의 일 실시예에 있어서, 메모리(1200)에는 인공지능 모델의 적어도 하나의 레이어의 가중치 값 및/또는 편향 값이 저장될 수 있다.

[0058] 메모리(1200)는 플래시 메모리 타입(flash memory type), 하드디스크 타입(hard disk type), 멀티미디어 카드 마이크로 타입(multimedia card micro type), 카드 타입의 메모리(예를 들어 SD 또는 XD 메모리 등), 램(RAM, Random Access Memory) SRAM(Static Random Access Memory), 롬(ROM, Read-Only Memory), EEPROM(Electrically Erasable Programmable Read-Only Memory), PROM(Programmable Read-Only Memory), 자기 메모리, 자기 디스크, 광디스크 중 적어도 하나의 타입의 저장매체를 포함할 수 있다. 본 개시의 일 실시예에 있어서, 메모리(1200)의 적어도 일부의 기능은 인터넷(internet)상에서 저장 기능을 수행하는 웹 스토리지(web storage) 또는 클라우드 서버에서 수행될 수 있다.

[0059] 프로세서(1300)는 전자 장치(1000)의 전반적인 동작을 제어할 수 있다. 프로세서(1300)는 메모리(1200)에 저장된 적어도 하나의 인스트럭션을 실행함으로써 전자 장치(1000)의 동작을 전반적으로 제어할 수 있다. 예를 들어, 프로세서(1300)는 하나 또는 복수의 프로세서로 구성될 수 있다. 이때, 하나 또는 복수의 프로세서는 CPU, AP, DSP(Digital Signal Processor) 등과 같은 범용 프로세서 또는 GPU, VPU(Vision Processing Unit)와 같은 그래픽 전용 프로세서로 구성될 수 있다.

[0060] 프로세서(1300)는 하드웨어 가속기(1400)의 동작을 제어할 수 있다. 예를 들어, 프로세서(1300)는 하드웨어 가속기(1400)의 제어를 위한 제어 신호를 생성하고, 생성한 제어 신호를 하드웨어 가속기에 제공할 수 있다. 여기서 제어 신호는 네트워크 유형, 레이어 수, 데이터의 차원, ReLU, 풀링 옵션, 블록 크기, 정밀도 등의 정보를 포함할 수 있다. 프로세서(1300)는 하드웨어 가속기(1400)에 특정 인공지능 모델의 학습 및/또는 추론을 지시할 수 있다. 프로세서(1300)는 하드웨어 가속기(1400)에 필요한 데이터를 제공하거나 제공받을 수 있다. 예를 들어, 프로세서(1300)는 메모리(1200)에 저장된 특정 인공지능 모델의 데이터(예컨대, 가중치 및/또는 편향)를 로드할 수 있다. 프로세서(1300)는 로드된 데이터를 하드웨어 가속기(1400)의 내부 메모리(미도시)에 전달할 수 있다. 프로세서(1300)는 메모리(1200)에 저장된 학습 데이터 셋(예컨대, 입력 값)을 하드웨어 가속기(1400)의 내부 메모리(미도시)에 전달할 수 있다.

[0061] 하드웨어 가속기(1400)는 인공지능 모델에 관련된 연산을 수행할 수 있다. 하드웨어 가속기(1400)는 블록 부동소수점(Block Floating Point) 형식으로 연산을 수행하며, 다양한 정밀도 및/또는 다양한 블록 크기로 연산 동작을 수행할 수 있다. 본 개시의 일 실시예에 있어서, 하드웨어 가속기(1400)는 지수를 공유하거나 지수가 없는 다양한 형태의 자료형을 지원할 수 있다. 예를 들어, 하드웨어 가속기(1400)는 고정 소수점 형태의 제1 자료형, 정수만 갖는 제2 자료형, 부호와 정수를 갖는 제3 자료형, 지수를 공유하는 실수 형태(즉, 블록 부동소수점)의

제4 자료형을 지원할 수 있다.

- [0062] 하드웨어 가속기(1400)는 부동소수점 형식의 제1 텐서(예컨대, 입력 값) 및 제2 텐서(예컨대, 가중치 값)를 수신할 수 있다. 하드웨어 가속기(1400)는 부동소수점 형식의 제1 텐서(예컨대, 입력 값) 및 제2 텐서(예컨대, 가중치 값)를 블록 부동소수점 형식으로 변환할 수 있다. 하드웨어 가속기(1400)는 블록 부동소수점 형식의 제1 텐서(예컨대, 입력 값) 및 제2 텐서(예컨대, 가중치 값) 간의 곱셈 연산을 수행할 수 있다. 설명의 편의를 위해, 제1 텐서가 입력 값으로 구성되는 텐서, 제2 텐서가 가중치 값으로 구성되는 텐서인 것을 가정하여 이하 설명되나, 본 개시는 이에 제한되지 않는다.
- [0063] 하드웨어 가속기(1400)는 프로세싱 코어(1410) 및 FP2BFP 변환기(1420)를 포함할 수 있다. 하드웨어 가속기(1400)는 프로세서(1300)로부터 제어 신호를 수신하고, 프로세싱 코어(1410)의 동작 상태에 따라 제어 신호를 최적화할 수 있다. 하드웨어 가속기(1400)는 제어 신호를 프로세싱 코어(1410)의 각 구성에 분배할 수 있다. 프로세싱 코어(1410)는 컨볼루션 연산 및/또는 GEMM(General matrix multiply) 연산 또는 이를 수행하기 위한 MAC(multiply and accumulation) 연산을 수행할 수 있다.
- [0064] 프로세싱 코어(1410)는 복수의 곱셈기들이 계층적으로 구성될 수 있다. 본 개시의 일 실시예에 있어서, 프로세싱 코어(1410)는 정수 곱셈기와 가산기(adder)로 구성될 수 있다. 예를 들어, 곱셈기(multiplier) → 프로세싱 엘리먼트(processing element, PE)(또는 프로세싱 엔진) → 서브 코어(subcore) → 프로세싱 코어(processing core)와 같이 계층적으로 구성될 수 있다.
- [0065] 프로세싱 코어(1410)는 복수의 서브 코어들을 포함할 수 있으며, 서브 코어들 각각은 복수의 프로세싱 엘리먼트들을 포함할 수 있으며, 프로세싱 엘리먼트들 각각은 복수의 곱셈기들을 포함할 수 있다. 후술하는 도 15 내지 20에서, 프로세싱 코어(1410)는 4개의 서브 코어들을 포함하며, 서브 코어들 각각은 16개의 프로세싱 엘리먼트들(또는 프로세싱 엔진들)을 포함하며, 프로세싱 엘리먼트들 각각은 8개의 멀티플라이어들을 포함하는 것으로 도시되었으나, 이는 일 예시이며, 본 개시는 이에 한정되지 않는다.
- [0066] 프로세싱 코어(1410)는 블록 부동소수점 값의 유효 숫자(즉, 부호 및 맨티사)만을 이용하여 계산하고, 지수는 공유 지수 핸들러(shared Exponent Handler)(미도시)에서 별도로 처리될 수 있다. 예를 들어, 공유 지수 핸들러(미도시)는 제1 텐서의 공유 지수와 제2 텐서의 공유 지수 간의 연산을 처리할 수 있다. 공유 지수 핸들러(미도시)는 하드웨어 가속기(1400)에 포함될 수 있으나, 프로세서(1300)에 포함될 수도 있다.
- [0067] FP2BFP 변환기(1420)는 부동소수점 값을 블록 부동소수점 값으로 변환할 수 있다. 프로세싱 코어(1410)의 출력은 부동소수점 값일 수 있다. FP2BFP 변환기(1420)는 복수의 부동소수점 값들 중 블록 크기 내의 최대 지수를 갖는 값의 지수를 공유 지수로 결정할 수 있다. 예를 들어, 블록 크기는 블록 부동소수점 형식 (예컨대, FB12, FB16, FB24), 학습 경과, 및/또는 레이어의 종류(예컨대, CONV1/FC, CONV3, CONV5, CONV7)에 따라 상이할 수 있으며, 사용자 또는 제조사의 설정에 따라 상이할 수 있다. FP2BFP 변환기(1420)는 최대 지수를 갖는 값의 인덱스를 메모리(1200) 또는 하드웨어 가속기(1400)의 내부 메모리(미도시)에 저장할 수 있다. FP2BFP 변환기(1420)는 공유 지수에서 입력 값들 각각의 지수를 감산할 수 있다. 즉, FP2BFP 변환기(1420)는 공유 지수와 동일한 블록의 입력 값들의 지수를 입력받아, 블록 부동소수점 형식에서의 지수 값을 계산할 수 있다. FP2BFP 변환기(1420)는 계산된 지수 값에 기초하여 정규화를 수행할 수 있다. 여기서 정규화란 맨티사를 '1.xxx' 형태로 바꾸는 것을 의미한다. FP2BFP 변환기(1420)는 블록 부동소수점 형식으로 변환시에 맨티사 값을 계산하기 위하여, 배럴 시프터(barrel shifter)를 이용하여 암시적 비트(implicit bit)와 명시적 비트들(explicit bits)의 유효숫자를 이동하는 동작을 수행함으로써, 앞서 계산된 지수 값에 대응하여 유효값(즉, 맨티사)를 조정할 수 있다. 배럴 시프터는 기 정의된 정밀도 또는 비트 폭으로 맨티사를 조정할 수 있다.
- [0068] FP2BFP 변환기(1420)는 맨티사를 조정함으로써 발생하는 언더플로우(underflow)의 발생 개수를 결정 또는 카운팅(counting)할 수 있다. 본 개시에서, 언더플로우는 비트 시프팅에 의해 모든 맨티사의 비트들이 '0'을 갖게 되는 현상을 나타낼 수 있다. FP2BFP 변환기(1420)는 결정된 언더플로우의 발생 개수를 프로세서(1300) 및/또는 프로세싱 코어(1410)에 전달할 수 있다. 프로세서(1300) 및/또는 프로세싱 코어(1410)는 결정된 언더플로우의 발생 개수에 기초하여 기 정의된 정밀도 또는 비트 폭을 상이한 정밀도 또는 비트 폭으로 변경할 수 있다. FP2BFP 변환기(1420)는 변경된 정밀도 또는 비트 폭으로 맨티사를 조정할 수 있다.
- [0069] 도시되지 않았지만, 하드웨어 가속기(1400)는 산술 변환기(미도시)를 포함할 수 있다. 산술 변환기(미도시)는 블록 부동소수점 값을 부동소수점 값으로 변환할 수 있다. 산술 변환기(미도시)는 데이터 타입 변환기, 리딩 제로 카운터(leading zero counter), 배럴 시프터(barrel shifter), 및 노말라이저(normalizer) 중 적어도 하나

를 포함할 수 있다. 본 개시의 일 실시예에 따르면, 정밀도와 데이터 포맷에 민감한 배치 정규화를, 부동소수점 값으로 변환된 값으로 수행함으로써, 학습 정확도가 보존될 수 있다. 산술 변환기(미도시)는 공유 지수 핸들러(미도시)가 출력한 지수 연산 결과 및 프로세싱 코어(1410)가 출력한 부호 및 맨티서 연산 결과에 기초하여 부동소수점 값을 출력할 수 있다.

[0070] 본 개시의 일 실시예에 있어서, 하드웨어 가속기(1400)는 인공지능 모델을 추론 및/또는 학습하기 위한 '전용 프로세서'이므로, 범용 프로세서인 프로세서(1300)와 함께 '프로세서'로도 지칭될 수 있다.

[0071] 본 개시의 일 실시예에 있어서, 하드웨어 가속기(1400)의 적어도 일부의 구성 및/또는 기능은 프로세서(1300)에 의해 수행될 수도 있다. 예를 들어, 프로세서(1300)는 FP2BFP 변환기(1420)의 기능을 수행할 수도 있다.

[0072] 본 개시의 일 실시예에 있어서, 하드웨어 가속기(1400)는 인공지능 모델의 여러 연산 과정 중에 다른 정밀도로 연산 동작을 수행하는 것이 가능하다. 구체적으로, 프로세서(150)는 포워드 패스 및 백워드 패스 중 적어도 하나의 과정에서는 제1 크기(예를 들어, 4 비트, 8 비트)로 데이터를 변환하며, 가중치 업데이트 과정에서는 제1 크기보다 큰 제2 크기(예를 들어, 16 비트)를 갖는 맨티서로 데이터를 변환하여 학습 동작을 수행할 수 있다. 또한, 가중치 업데이트 과정에서는 손실 경사 맵을 기설정된 크기로 분할하고, 분할된 손실 경사 맵 단위로 연산 동작을 수행할 수 있다.

[0073] 이상과 같이 본 개시에 따른 전자 장치(1000)는 인공지능 모델(예컨대, 심층 신경망)의 학습 또는 추론에 사용되는 전용의 하드웨어 가속기(1400)를 이용하여 학습 및/또는 추론을 수행하는바, 빠르게 학습 또는 추론 동작을 수행할 수 있다. 또한, 학습 과정에서 고정된 블록 크기 및/또는 정밀도를 이용하지 않고, 학습과정 내의 각 단계에 적합한 블록 크기 및/또는 정밀도를 이용하여 학습 과정을 수행하는 바, 에너지 소모를 효율적으로 관리할 수 있다. 또한, 블록 부동소수점 값들 중 최대 지수를 갖는 값들의 인덱스를 저장함으로써 메모리를 효율적으로 사용할 수 있다.

[0074] 한편, 이상에서는 하드웨어 가속기(1400)와 프로세서(1300)가 하나의 전자 장치(1000) 내에서 구성되는 것을 전제하여 설명하였지만, 하드웨어 가속기(1400)와 프로세서(1300) 각각은 서로 다른 전자 장치로 구성될 수도 있다. 또한, 도 3을 도시하고 설명함에 있어서, 상술한 동작이 인공지능 모델의 학습 과정에만 적용되는 것으로 설명하였지만, 학습된 인공지능 모델을 이용한 추론 과정에서도 적용될 수 있다.

[0076] 도 4는 본 개시의 일 실시예에 따른 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법을 설명하기 위한 개념도이다. FP2BFP 변환기(1420) 및 메모리(1200)의 구성, 기능, 및 동작은 도 3의 FP2BFP 변환기(1420) 및 메모리(1200)의 구성, 기능, 및 동작에 대응할 수 있다. 설명의 편의를 위해, 도 1 내지 3에서 설명한 내용은 생략한다.

[0077] 도 4와 함께, 도 3을 참조하면, FP2BFP 변환기(1420)는 복수의 부동소수점 값들(40)을 획득할 수 있다. 예를 들어, FP2BFP 변환기(1420)는 메모리(1200) 또는 하드웨어 가속기(1400)의 내부 메모리로부터 복수의 부동소수점 값들(40)을 획득할 수 있다. 복수의 부동소수점 값들(40) 각각은 부호, 지수, 및 맨티서로 구성될 수 있다. 예시적으로, 복수의 부동소수점 값들(40)이 부호는 1 비트, 지수는 8 비트들, 맨티서는 7 비트들(암시적 비트까지 8 비트들)로 구성되는 bfloat16 형식을 도시하였으나, 본 개시는 이에 한정되지 않으며, 부호, 지수, 및 맨티서 각각의 비트 폭은 다양할 수 있다.

[0078] 맨티서는 1 비트의 암시적 비트와 복수의 비트들의 명시적 비트들로 구성될 수 있다. 암시적 비트는 소수점 앞의 비트를 나타내고, 명시적 비트는 소수점 이하의 비트(또는 정밀도)를 나타낼 수 있다. 암시적 비트는 메모리에 저장되지 않는 비트를 나타내고, 명시적 비트는 메모리에 저장되는 비트를 나타낼 수 있다. 암시적 비트는 항상 특정 값(예컨대, '1')으로 설정되어 있어서 메모리 공간을 차지하지 않을 수 있다. 예시적으로, 맨티서는 '1.xxx' 형태로 구현될 수 있으며, 소수점 앞의 '1'이 암시적 비트로 표현될 수 있고, 소수점 뒤의 'xxx'가 명시적 비트로 표현될 수 있다.

[0079] 암시적 비트는 부동소수점 형식의 맨티서의 가장 왼쪽에 위치한 비트를 나타낼 수 있다. 암시적 비트는 맨티서의 최상위 비트(most significant bit)를 나타낼 수 있다. 예를 들어, 맨티서가 8 비트들로 구성되는 경우, 최상위 1 비트는 암시적 비트이고, 나머지 7 비트들은 명시적 비트들로 구성될 수 있다.

[0080] FP2BFP 변환기(1420)는 복수의 부동소수점 값들(40)을 복수의 블록 부동소수점 값들(50)로 변환할 수 있다. 복수의 블록 부동소수점 값들(50)은 하나의 블록에 대응할 수 있다. FP2BFP 변환기(1420)는 복수의 부동소수점 값

들 중 최대 지수를 갖는 적어도 하나의 제1 부동소수점 값의 지수를 공유 지수로 결정할 수 있다. 도 4에서, 복수의 부동소수점 값들(40)의 지수들 중에서, 최대 지수가 일점쇄선으로 표현된다. FP2BFP 변환기(1420)는 최대 지수를 갖는 제1 부동소수점 값의 지수를 메모리(1200) 또는 하드웨어 가속기(1400)의 내부 메모리(미도시)에 저장할 수 있다. FP2BFP 변환기(1420)는 최대 지수를 갖는 제1 부동소수점 값의 인덱스를 메모리(1200) 또는 하드웨어 가속기(1400)의 내부 메모리(미도시)에 저장할 수 있다. 예를 들어, 제1 부동소수점 값의 인덱스는 최대 지수 인덱스로도 지칭될 수 있다. 최대 지수 인덱스는 제1 부동소수점 값의 부호 및 맨티서에 대응하는 비트들에 매핑될 수 있다. 최대 지수 인덱스를 갖는 경우, 맨티서의 암시적 비트는 '1'인 것으로 해석될 수 있다.

[0081] FP2BFP 변환기(1420)는 복수의 부동소수점 값들 중 최대 지수를 갖지 않는 제2 부동소수점 값에서, 공유 지수와 제2 부동소수점 값의 지수의 차이를 계산할 수 있다. 예시적으로, 공유 지수가 '128'을 표현하고, 제2 부동소수점 값이 '124'를 표현하는 경우, 지수의 차이는 '4'일 수 있다. FP2BFP 변환기(1420)는 계산된 지수의 차이만큼 제2 부동소수점 값의 맨티서의 비트들을 오른쪽 시프팅할 수 있다. 맨티서의 비트들은 암시적 비트와 명시적 비트들을 포함할 수 있다. 예시적으로, 제2 부동소수점 값의 맨티서의 암시적 비트가 '1'이고, 명시적 비트들이 '1000101'이며, 지수의 차이가 '4'인 경우, FP2BFP 변환기(1420)는 제2 부동소수점 값의 맨티서의 암시적 비트를 '0'으로, 명시적 비트들을 '0001100'으로 변환할 수 있다.

[0082] 본 개시의 일 실시예에 있어서, FP2BFP 변환기(1420)는 반올림기를 포함할 수 있다. 반올림기는 시프팅 동작에 의해 버려지는 비트들을 식별할 수 있다. 반올림기는 식별된 비트들에 기초하여 기 정의된 비트 폭을 갖는 맨티서의 비트들을 반올림할 수 있다. 예시적으로, 제2 부동소수점 값의 맨티서의 암시적 비트가 '1'이고, 명시적 비트들이 '1001001'이며, 지수의 차이가 '4'인 경우, FP2BFP 변환기(1420)는 제2 부동소수점 값의 맨티서의 암시적 비트를 '0'으로, 명시적 비트들을 '0001101'으로 변환할 수 있다. 그러나 본 개시는 이에 한정되지 않으며, FP2BFP 변환기(1420)는 반올림기 대신 올림 또는 내림 동작을 수행하는 로직 회로를 포함할 수도 있다.

[0083] FP2BFP 변환기(1420)는 부호와, 변환된 맨티서의 명시적 비트들을 메모리(1200) 또는 하드웨어 가속기(1400)의 내부 메모리(미도시)에 저장할 수 있다.

[0084] 본 개시의 일 실시예에 따르면, 부동소수점 형식의 맨티서의 암시적 비트를 블록 부동소수점 형식에서도 활용함에 따라, 맨티서를 표현하기 위한 비트 수를 절약할 수 있다.

[0085] 암시적 비트를 활용하지 않는 블록 부동소수점 값을 저장하기 위해 필요한 평균 비트 수는 수학적 4를 따를 수 있다.

수학적 4

$$BFP_{bit} = \text{len}(s + m) + \frac{\text{len}(e_s)}{N}$$

[0086]

[0087] 수학적 4를 참조하면, BFP_{bit} 는 블록 부동소수점 형식의 특정 블록에 포함되는 값들 각각의 비트 수로 정의되고, $\text{len}(s+m)$ 은 부호 및 맨티서의 비트 폭으로 정의되고, N 은 해당 블록의 블록 크기로 정의되고, $\text{len}(e_s)$ 은 공유 지수의 비트 폭으로 정의된다. 예시적으로, 부호 및 맨티서의 비트 폭이 8이고, 해당 블록의 블록 크기가 64이고, 공유 지수의 비트 폭이 8인 경우, 해당 블록의 하나의 값을 저장하기 위해 필요한 평균 비트 수는 8.125이 된다.

[0088] 본 개시의 일 실시예에 따른 암시적 비트를 활용하는 블록 부동소수점 형식의 메모리 공간은 수학적 5를 따를 수 있다.

수학적 5

$$BFP_{bit} = \text{len}(s + m) + \frac{\text{len}(e_s)}{N} + \frac{[\log_2 N] \times N_{max}}{N}$$

[0089]

$$[\log_2 N] \times N_{max}$$

[0090] 수학식 4 및 수학식 5를 참조하면, N_{max} 는 최대 지수 인덱스를 갖는 값의 개수로 정의된다. 는 최대 지수 인덱스를 저장하기 위해 필요한 블록 당 비트 수로 정의된다. 예시적으로, 부호 및 맨티서의 비트 폭이 8이고, 해당 블록의 블록 크기가 64이고, 공유 지수의 비트 폭이 8이고, 최대 지수 인덱스를 갖는 값의 개수가 5인 경우, 해당 블록의 하나의 값을 저장하기 위해 필요한 평균 비트 수는 8.594가 된다.

[0091] 본 개시의 일 실시예에 따르면, 수학식 5에 따르는 경우 수학식 4에 따르는 경우보다 암시적 비트의 1 비트를 더 표현할 수 있음에도 불구하고, 수학식 4에 따른 블록 부동소수점 형식보다 약 0.469 비트 정도의 추가 메모리 공간만을 사용한다. 본 개시의 일 실시예에 따르면, 더 높은 정밀도를 표현함에도 메모리 공간을 효율적으로 활용하거나, 같은 정밀도를 표현함에도 더 적은 메모리 공간만을 필요로 할 수 있다.

표 1

Block Size	32	64	128	256
BFP4	11.7	15.0	26.0	48.2
BFP8	11.0	15.4	29.4	66.9
BFP16	16.3	26.2	49.8	94.7

[0092]

[0093] 표 1을 참조하면, 표 1은 최대 지수 인덱스를 저장하기 위해 필요한 블록 당 비트 수를 나타낸다. 최대 지수를 갖는 값들의 개수는 데이터마다 다르기 때문에, 표 1에 표시된 수치들은 평균적인 수치들을 나타낸다. BFPn에서, n은 부호 및 맨티서의 비트 폭을 나타낸다. 맨티서의 경우, 명시적 비트의 비트 폭만을 포함한다. 예를 들어, BFP4는 부호 및 맨티서의 비트 폭이 4 비트인 경우를 나타내고, BFP8은 부호 및 맨티서의 비트 폭이 8 비트인 경우를 나타내고, BFP16은 부호 및 맨티서의 비트 폭이 16 비트인 경우를 나타낼 수 있다. 예시적으로, 블록 크기가 32이고, BFP4 형식인 경우, 최대 지수 인덱스를 저장하기 위해 필요한 블록 당 평균 비트 수는 약 11.7 비트일 수 있다. 본 개시의 일 실시예에 따르면, 블록 크기가 32인 경우 맨티서의 암시적 비트로 총 32 비트를 확보할 수 있음에도, 32 비트가 아닌 약 11.7 비트만큼의 메모리 공간만을 사용할 수 있다.

[0095] 도 5는 본 개시의 일 실시예에 따른 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법을 보여주는 흐름도이다. 도 1 내지 4에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3을 참조하여, 도 5를 설명한다.

[0096] 도 5를 참조하면, 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법은 단계 S510 내지 S550을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S510 내지 S550은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S510 내지 S550은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법은 도 5에 도시된 바에 한정되지 않으며, 도 5에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 5에 도시되지 않은 단계들 더 포함할 수도 있다.

[0097] 단계 S510에서, 전자 장치(1000)는 복수의 부동소수점 값들을 획득할 수 있다. 예를 들어, 전자 장치(1000)는 학습 데이터 셋, 추론을 위한 입력 데이터 및/또는 인공지능 모델의 파라미터(예컨대, 가중치 값 및/또는 편향 값)를 획득할 수 있다. 학습 데이터 셋, 입력 데이터, 및/또는 파라미터는 부동소수점 형식으로 표현될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 학습 데이터 셋, 입력 데이터, 및/또는 파라미터는 다양한 형식으로 표현될 수 있으며, 전자 장치(1000)는 학습 데이터 셋, 입력 데이터, 및/또는 파라미터를 부동소수점 형식으로 변환할 수 있다. 전자 장치(1000)는 메모리(1200) 또는 하드웨어 가속기(1400)의 내부 메모리로부터 복수의 부동소수점 값들을 획득할 수 있다. 예를 들어, 전자 장치(1000)는 프로세싱 코어(1410)의 출력 값인 복수의 부동소수점 값들을 획득할 수 있다. 본 개시의 일 실시예에 있어서, 복수의 부동소수점 값들의 개수를 미리 정의될 수 있다. 복수의 부동소수점 값들은 하나의 블록에 대응할 수 있다. 하나의 블록은 기 정의된 개수의 부동소수점 값들을 포함할 수 있다. 하나의 블록에 포함되는 부동소수점 값들의 개수는 블록 크기로 지칭될 수도 있

다. 예시적으로, 블록 크기가 256인 경우, 하나의 블록은 256 개의 부동소수점 값들을 포함할 수 있다.

- [0098] 단계 S520에서, 전자 장치(1000)는 복수의 부동소수점 값들 중 최대 지수를 갖는 적어도 하나의 제1 부동소수점 값의 지수를 공유 지수로 결정할 수 있다. 전자 장치(1000)는 하나의 블록에 대응하는 복수의 부동소수점 값들 중 최대 지수를 갖는 값(예컨대, 제1 부동소수점 값)을 식별할 수 있다. 전자 장치(1000)는 식별된 값의 지수를 식별할 수 있다. 전자 장치(1000)는 식별된 지수를 해당 블록의 공유 지수로 결정할 수 있다.
- [0099] 단계 S530에서, 전자 장치(1000)는 적어도 하나의 제1 부동소수점 값의 인덱스를 메모리(1200)에 저장할 수 있다. 예를 들어, 복수의 부동소수점 값들은 대응하는 인덱스를 가질 수 있다. 전자 장치(1000)는 복수의 부동소수점 값들 중, 공유 지수와 동일한 지수를 갖는 적어도 하나의 제1 부동소수점 값에 대응하는 인덱스를 메모리에 저장할 수 있다.
- [0100] 단계 S540에서, 전자 장치(1000)는 복수의 부동소수점 값들 중 최대 지수를 갖지 않는 적어도 하나의 제2 부동소수점 값의 맨티서의 암시적 비트와 명시적 비트들을, 공유 지수와 적어도 하나의 제2 부동소수점 값의 지수의 차이만큼 오른쪽 시프팅할 수 있다. 본 개시의 일 실시예에 있어서, 시프팅되기 전 암시적 비트는 시프팅된 후에 명시적 비트로 표현될 수 있다. 본 개시의 일 실시예에 있어서, 시프팅된 제2 부동소수점 값의 맨티서의 암시적 비트는 '0'으로 해석될 수 있다.
- [0101] 단계 S550에서, 전자 장치(1000)는 적어도 하나의 제1 부동소수점 값의 부호 및 맨티서, 적어도 하나의 제2 부동소수점 값의 부호 및 맨티서, 그리고 공유 지수로 구성되는 복수의 블록 부동소수점 값들을 메모리(1200)에 저장할 수 있다. 본 개시의 일 실시예에 있어서, 맨티서 중 명시적 비트는 메모리(1200)에 저장되고, 암시적 비트는 메모리에 저장되지 않는다. 본 개시의 일 실시예에 있어서, 제1 부동소수점 값의 맨티서의 제1 암시적 비트는 제1 값(예컨대, '1')일 수 있다. 본 개시의 일 실시예에 있어서, 제2 부동소수점 값의 맨티서의 제2 암시적 비트는 제2 값(예컨대, '0')일 수 있다. 본 개시의 일 실시예에 있어서, 제1 부동소수점 값의 맨티서의 암시적 비트는 '1'로 해석되고, 제2 부동소수점 값의 맨티서의 암시적 비트는 '0'으로 해석될 수 있다.
- [0103] 도 6은 본 개시의 일 실시예에 따른 비트 폭 또는 정밀도를 변경하는 방법을 보여주는 흐름도이다. 도 1 내지 5에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3 및 5를 참조하여, 도 6을 설명한다.
- [0104] 도 6을 참조하면, 비트 폭 또는 정밀도를 변경하는 방법은 단계 S610 내지 S630을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S610 내지 S630은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S610 내지 S630은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 비트 폭 또는 정밀도를 변경하는 방법은 도 6에 도시된 바에 한정되지 않으며, 도 6에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 6에 도시되지 않은 단계들 더 포함할 수도 있다.
- [0105] 본 개시의 일 실시예에 있어서, 도 5의 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법은 비트 폭 또는 정밀도를 변경하는 방법에 대응하는 단계 S610 내지 S630을 포함할 수 있다. 단계 S540 이후, 절차는 단계 S610로 이동할 수 있다.
- [0106] 단계 S610에서, 전자 장치(1000)는 적어도 하나의 제1 부동소수점 값의 부호 및 맨티서와 적어도 하나의 제2 부동소수점 값의 부호 및 맨티서 각각이, 기 정의된 비트 폭을 갖도록 조정할 수 있다. 기 정의된 비트 폭은 사용자 또는 제조사의 설정에 따라 상이할 수 있다. 예시적으로, 제1 부동소수점과 제2 부동소수점의 부호는 1 비트의 비트 폭을 가지고 맨티서는 7 비트의 비트 폭을 갖되, 기 정의된 비트 폭은 4 비트인 경우를 가정한다. 기 정의된 비트 폭에서 1 비트는 부호에 대응하고, 3 비트는 맨티서에 대응할 수 있다. 제1 부동소수점의 맨티서의 오른쪽 4비트와 오른쪽 시프팅된 제2 부동소수점에서 오른쪽 4 비트는 버려질 수 있다.
- [0107] 단계 S620에서, 전자 장치(1000)는 적어도 하나의 제2 부동소수점 값에 대한 언더플로우 발생 개수를 결정할 수 있다. 적어도 하나의 제2 부동소수점 값이 오른쪽 시프팅되거나, 및/또는 기 정의된 비트 폭을 갖도록 조정됨으로써, 맨티서의 모든 비트들이 '0' 값을 갖는 언더플로우가 발생할 수 있다. 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 인공지능 모델의 학습 연산 과정에서 언더플로우 발생 개수를 결정할 수 있다. 복수의 부동소수점 값들은 학습 데이터 셋(예컨대, 입력 값) 및/또는 적어도 하나의 레이어의 파라미터(예컨대, 가중치 값 및/또는 편향 값)에 대응할 수 있다. 전자 장치(1000)는 기 정의된 학습 단위 동안의 적어도 하나의 제2 부동소수점 값에 대한 언더플로우 발생 개수를 결정할 수 있다. 예를 들어, 기 정의된 학습 단위는 배치(batch) 크기(또는 미니-배치의 개수로도 지칭될 수 있음), 에포크(epoch), 및 이터레이션(iteration) 중 적어도 하나일 수 있다.

다. 예시적으로, 기 정의된 학습 단위가 배치 크기이고, 배치 크기가 100 인 경우를 가정한다. 전자 장치(100)는 100 개의 미니-배치들에 대한 학습 연산을 수행할 동안 언더플로우 발생 개수를 카운팅할 수 있다. 전자 장치(1000)는 카운팅 결과에 기초하여 언더플로우 발생 개수를 결정할 수 있다.

[0108] 단계 S630에서, 전자 장치(1000)는 결정된 언더플로우의 발생 개수에 기초하여, 기 정의된 비트 폭을 변경할 수 있다. 기 정의된 비트 폭은 블록 부동소수점 값의 부호 및 맨티사의 비트 폭을 나타낼 수 있다. 예를 들어, 기 정의된 비트 폭이 4 비트인 경우, 제1 블록 부동소수점 값 및/또는 제2 블록 부동소수점 값의 부호 및 맨티사의 비트 폭은 4 비트일 수 있다. 여기서, 기 정의된 비트 폭은 맨티사의 암시적 비트는 제외한 표현일 수 있다. 전자 장치(1000)는 결정된 언더플로우의 발생 개수가 임계 범위를 초과한 것인지 결정할 수 있다. 전자 장치(1000)는 결정된 언더 플로우의 발생 개수가 임계 범위를 초과한 것으로 결정한 것에 기초하여, 기 정의된 비트 폭을 더 큰 비트 폭으로 변경하거나 더 작은 비트 폭으로 변경할 수 있다. 단계 S630 이후, 절차는 단계 S550으로 이동한다.

[0109] 본 개시의 일 실시예에 있어서, 단계 S550 이후, 절차는 단계 S510으로 이동하며, 전자 장치(1000)는 다음 학습 단위에 대한 부동소수점 값을 블록 부동소수점 값으로 변환하는 동작을 반복할 수 있다. 단계 S610에서는, 변경된 비트 폭을 갖도록 제1 부동소수점 값과 제2 부동소수점 값을 조정할 수 있다. 예시적으로, 기 정의된 학습 단위가 배치 크기이고, 배치 크기가 100 인 경우를 가정한다. 전자 장치(1000)는 다음 100 개의 미니-배치들에 대한 학습을 수행할 때, 단계 S510 내지 S550, 및 S610 내지 S630에 대응하는 동작들을 수행할 수 있다.

[0111] 도 7은 도 6의 단계 S630의 세부 단계들을 보여주는 흐름도이다. 도 1 내지 6에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3, 5, 및 6을 참조하여, 도 7을 설명한다.

[0112] 도 7을 참조하면, 도 6의 단계 S630은 단계 S710 내지 S740을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S710 내지 S740은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S710 내지 S740은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 단계 S630의 세부 단계들은 도 7에 도시된 바에 한정되지 않으며, 도 7에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 7에 도시되지 않은 단계를 더 포함할 수도 있다.

[0113] 단계 S710에서, 전자 장치(1000)는 언더플로우의 발생 개수가 제1 임계 개수보다 많은지를 결정할 수 있다. 제1 임계 개수는 사용자 또는 제조사의 설정에 따라 미리 정의될 수 있다. 언더플로우의 발생 개수가 제1 임계 개수보다 많은 것으로 결정한 것에 기초하여(예), 절차는 단계 S720으로 이동한다. 언더플로우의 발생 개수가 제1 임계 개수보다 많지 않은 것으로 결정한 것에 기초하여(아니오), 절차는 단계 S730으로 이동한다.

[0114] 단계 S720에서, 전자 장치(1000)는 기 정의된 비트 폭을 더 큰 비트 폭으로 변경할 수 있다. 예를 들어, 현재 비트 폭이 4 비트인 경우, 전자 장치(1000)는 다음 비트 폭을 8 비트를 갖도록 변경할 수 있으나, 본 개시는 이에 한정되지 않으며, 전자 장치(1000)는 기 정의된 비트 폭을, 4 비트보다 더 큰 임의의 정수 값의 비트 폭을 갖도록 변경할 수 있다. 단계 S720 이후, 절차는 단계 S550으로 이동한다.

[0115] 단계 S730에서, 전자 장치(1000)는 언더플로우의 발생 개수가 제2 임계 개수보다 적은지를 결정할 수 있다. 제2 임계 개수는 사용자 또는 제조사의 설정에 따라 미리 정의될 수 있다. 언더플로우의 발생 개수가 제2 임계 개수보다 적은 것으로 결정한 것에 기초하여(예), 절차는 단계 S740으로 이동한다. 언더플로우의 발생 개수가 제1 임계 개수보다 많지 않은 것으로 결정한 것에 기초하여(아니오), 절차는 단계 S550으로 이동한다.

[0116] 단계 S740에서, 전자 장치(1000)는 기 정의된 비트 폭을 더 작은 비트 폭으로 변경할 수 있다. 예를 들어, 현재 비트 폭이 16 비트인 경우, 전자 장치(1000)는 다음 비트 폭을 8 비트를 갖도록 변경할 수 있으나, 본 개시는 이에 한정되지 않으며, 전자 장치(1000)는 기 정의된 비트 폭을, 16 비트보다 더 작은 임의의 정수 값의 비트 폭을 갖도록 변경할 수 있다. 단계 S740 이후, 절차는 단계 S550으로 이동한다.

[0117] 본 개시의 일 실시예에 따르면, 기 정의된 학습 단위마다 부호 및 맨티사의 비트 폭을 변경함으로써, 데이터의 특성에 따라 적응적으로 학습 속도를 높이거나 학습 정확성을 높일 수 있다.

[0118] 본 개시의 일 실시예에 있어서, 도 7에 도시된 바와 달리, 단계 S730이 단계 S710보다 먼저 수행될 수 있다. 예를 들어, 언더플로우 발생 개수가 제2 임계 개수보다 적은지를 먼저 결정하고, 언더플로우 발생 개수가 제1 임계 개수보다 많은지를 그 다음에 결정할 수 있다.

- [0120] 도 8은 본 개시의 일 실시예에 따른 블록 크기를 변경하는 방법을 보여주는 흐름도이다. 도 1 내지 7에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3 및 5를 참조하여, 도 8을 설명한다.
- [0121] 도 8을 참조하면, 블록 크기를 변경하는 방법은 단계 S810 내지 S830을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S810 내지 S830은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S810 내지 S830은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 블록 크기를 변경하는 방법은 도 8에 도시된 바에 한정되지 않으며, 도 8에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 8에 도시되지 않은 단계들 더 포함할 수도 있다.
- [0122] 본 개시의 일 실시예에 있어서, 도 5의 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법은 블록 크기를 변경하는 방법에 대응하는 단계 S810 내지 S830을 포함할 수 있다. 단계 S550 이후, 절차는 단계 S810로 이동할 수 있다.
- [0123] 단계 S810에서, 전자 장치(1000)는 복수의 블록 부동소수점 값들에 기초하여 인공지능 모델에 대한 학습 연산을 수행할 수 있다. 예를 들어, 복수의 블록 부동소수점 값들은 입력 값 및 가중치 값 중 적어도 하나를 포함할 수 있다. 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 입력 값과 가중치 값 간의 MAC 연산을 수행할 수 있다.
- [0124] 단계 S820에서, 전자 장치(1000)는 기 정의된 수의 에포크(epoch)만큼 학습 연산을 수행했는지를 결정할 수 있다. 본 개시에서, 에포크는 전체 학습 데이터 셋이 인공지능 모델을 통과함으로써, 인공지능 모델의 파라미터가 갱신되는 하나의 주기를 나타낼 수 있다. 기 정의된 수의 에포크만큼 학습 연산이 수행되지 않은 경우, 블록 크기를 변경하지 않으며, 절차는 단계 S510으로 이동한다.
- [0125] 단계 S830에서, 전자 장치(1000)는 기 정의된 수의 에포크만큼 학습 연산을 수행한 것으로 결정한 것에 기초하여, 복수의 부동소수점 값들에 대응하는 블록 크기를 변경할 수 있다. 예를 들어, 복수의 부동소수점 값들은 제 1 블록 크기를 갖는 복수의 블록 부동소수점 값들로 변환될 수 있다. 기 정의된 수의 에포크 만큼 학습 연산이 수행된 경우, 다음 에포크에서 획득될 복수의 부동소수점 값들이 제2 블록 크기를 갖는 복수의 블록 부동소수점 값들로 변환할 수 있다. 제2 블록 크기는 제1 블록 크기보다 클 수 있다. 본 개시의 일 실시예에 있어서, 에포크의 기 정의된 수는 사용자 또는 제조사의 설정에 따라 상이할 수 있다.
- [0126] 본 개시의 일 실시예에 있어서, 인공지능 모델의 학습 과정에서 학습률 감소(learning rate decay) 기법을 적용하는 기 정의된 수의 에포크가 진행된 경우, 복수의 부동소수점 값들에 대응하는 블록 크기를 변경할 수 있다.
- [0127] 본 개시의 일 실시예에 따르면, 블록 크기를 적응적으로 증가시킴으로써 학습에 필요한 데이터 크기 또는 메모리 공간을 획기적으로 감소시킬 수 있다. 본 개시의 일 실시예에 따르면, 블록 크기를 적응적으로 증가시킴으로써 학습에 필요한 하드웨어 가속기의 에너지 소모를 획기적으로 감소시킬 수 있다.
- [0129] 도 9는 본 개시의 일 실시예에 따른 블록 부동소수점 값을 처리하는 방법을 보여주는 흐름도이다. 도 1 내지 8에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3을 참조하여, 도 9를 설명한다.
- [0130] 도 9를 참조하면, 블록 부동소수점 값을 처리하는 방법은 단계 S910 내지 S940을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S910 내지 S940은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S910 내지 S940은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 블록 부동소수점 값을 처리하는 방법은 도 9에 도시된 바에 한정되지 않으며, 도 9에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 9에 도시되지 않은 단계들 더 포함할 수도 있다.
- [0131] 단계 S910에서, 전자 장치(1000)는 공유 지수를 갖는 복수의 블록 부동소수점 값들 및 적어도 하나의 최대 지수 인덱스를 획득할 수 있다. 본 개시의 일 실시예에 있어서, 복수의 블록 부동소수점 값들은 적어도 하나의 제1 블록 부동소수점 값과 적어도 하나의 제2 블록 부동소수점 값을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 메모리(1200)에 저장된 블록의 공유 지수, 복수의 블록 부동소수점 값들 각각의 부호 및 맨티사, 및 최대 지수 인덱스 리스트를 로드할 수 있다. 전자 장치(1000)는 최대 지수 인덱스 리스트로부터

터 적어도 하나의 최대 지수 인덱스를 획득할 수 있다.

- [0132] 단계 S920에서, 전자 장치(1000)는 복수의 블록 부동소수점 값들 각각의 인덱스가 적어도 하나의 최대 지수 인덱스에 대응하는지를 결정할 수 있다. 본 개시의 일 실시예에 있어서, 전자 장치(100)는 적어도 하나의 최대 지수 인덱스에 매핑된 블록 부동소수점 값을 식별할 수 있다. 블록 부동소수점 값의 인덱스가 최대 지수 인덱스에 대응하는 경우, 절차는 단계 S830으로 이동한다. 복수의 블록 부동소수점 값의 인덱스가 최대 지수 인덱스에 대응하지 않는 경우, 절차는 단계 S840으로 이동한다.
- [0133] 단계 S930에서, 전자 장치(1000)는 인덱스에 대응하는 제1 블록 부동소수점 값의 제1 암시적 비트를 제1 값으로 결정할 수 있다. 예를 들어, 제1 값은 '1'일 수 있다. 예를 들어, 제1 블록 부동소수점 값의 지수는 공유 지수와 동일할 수 있다.
- [0134] 단계 S940에서, 전자 장치(1000)는 인덱스에 대응하는 제2 블록 부동소수점 값의 제2 암시적 비트를 제2 값으로 결정할 수 있다. 예를 들어, 제2 값은 '0'일 수 있다. 예를 들어, 제2 블록 부동소수점 값의 지수는 공유 지수보다 작을 수 있다.
- [0135]
- [0136] 도 10은 본 개시의 일 실시예에 따른 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법을 보여주는 흐름도이다. 도 1 내지 9에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3을 참조하여, 도 10을 설명한다.
- [0137] 도 10을 참조하면, 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법은 단계 S1010 내지 S1040을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S1010 내지 S1040은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S1010 내지 S1040은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법은 도 10에 도시된 바에 한정되지 않으며, 도 10에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 10에 도시되지 않은 단계를 더 포함할 수도 있다.
- [0138] 단계 S1010에서, 전자 장치(1000)는 제1 블록 크기를 갖는 복수의 제1 블록 부동소수점 값들로 구성되는 학습 데이터 셋을 획득할 수 있다.
- [0139] 단계 S1020에서, 전자 장치(1000)는 학습 데이터 셋에 기초하여 인공지능 모델에 대한 학습 연산을 수행할 수 있다. 본 개시의 일 실시예에 있어서, 인공지능 모델은 적어도 하나의 레이어를 포함할 수 있다. 적어도 하나의 레이어는 적어도 하나의 가중치 값을 포함할 수 있다. 학습 데이터 셋은 적어도 하나의 입력 값을 포함할 수 있다. 본 개시에서, 학습 연산은 입력 값과 가중치 값 간의 곱셈 연산을 통해 수행되는 포워드 패스(forward pass) 단계, 백워드 패스(backward pass) 단계, 및 가중치 갱신 단계로 구성될 수 있다. 예를 들어, 포워드 패스 단계는 학습 과정의 손실을 계산하는 과정이고, 백워드 패스 단계는 손실함수에 대한 경사를 계산하는 과정이다. 경사는 보통 연쇄법칙(Chain rule)을 통해 구해지며, 포워드 패스 단계와 반대 방향으로 인공지능 모델을 구성하는 모든 레이어에 대해 전파된다. 가중치 갱신 단계는 인공지능 모델의 가중치 값들을 갱신하는 과정으로, 가중치 값에 대한 손실함수의 경사를 학습률과 곱하여 기존의 가중치 값에서 해당 값을 뺌으로써 갱신이 이루어진다.
- [0140] 단계 S1030에서, 전자 장치(1000)는 기 정의된 수의 에포크들 동안 인공지능 모델에 대한 학습 연산을 반복할 수 있다. 기 정의된 수의 에포크들 동안 학습 연산이 반복된 이후, 절차는 단계 S1040으로 이동한다.
- [0141] 단계 S1040에서, 전자 장치(1000)는 학습 데이터 셋을 제2 블록 크기를 갖는 복수의 제2 블록 부동소수점 값들로 재구성할 수 있다. 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 학습 데이터 셋의 적어도 일부에 대응하는 복수의 부동소수점 값들을 획득할 수 있다. 전자 장치(1000)는 복수의 부동소수점 값들을 제2 블록 크기의 블록 부동소수점 값들로 변환할 수 있다. 본 개시의 일 실시예에 있어서, 제2 블록 크기는 제1 블록 크기보다 클 수 있다.
- [0143] 도 11은 본 개시의 일 실시예에 따른 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법을 보여주는 흐름도이다. 도 1 내지 10에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3 및 10을 참

조하여, 도 11을 설명한다.

- [0144] 도 11을 참조하면, 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법은 단계 S1010 내지 S1040과 함께, 단계 S1110 내지 S1120을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S1110 내지 S1120은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S1110 내지 S1120은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법은 도 11에 도시된 바에 한정되지 않으며, 도 11에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 11에 도시되지 않은 단계들을 더 포함할 수도 있다.
- [0145] 단계 S1110에서, 전자 장치(1000)는 채구성된 학습 데이터 셋에 기초하여 학습 연산을 수행할 수 있다. 단계 S1110은 도 10의 단계 S1020에 대응할 수 있다.
- [0146] 단계 S1120에서, 전자 장치(1000)는 기 정의된 수의 에포크들 동안 인공지능 모델에 대한 학습 연산을 반복할 수 있다. 단계 S1120은 도 10의 단계 S1030에 대응할 수 있다. 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 학습 데이터 셋을 제3 블록 크기를 갖는 복수의 제2 블록 부동소수점 값들로 채구성할 수 있다. 본 개시의 일 실시예에 있어서, 제3 블록 크기는 제2 블록 크기보다 클 수 있다.
- [0148] 도 12는 도 10의 단계 S1020의 세부 단계들을 보여주는 흐름도이다. 도 1 내지 11에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3, 및 10을 참조하여, 도 12를 설명한다.
- [0149] 도 12를 참조하면, 도 10의 단계 S1020은 단계 S1210 내지 S1240을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S1210 내지 S1240은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S1210 내지 S1240은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 단계 S1020의 세부 단계들은 도 12에 도시된 바에 한정되지 않으며, 도 12에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 12에 도시되지 않은 단계들을 더 포함할 수도 있다.
- [0150] 단계 S1210에서, 전자 장치(1000)는 기 정의된 수의 에포크들 중 제1 에포크에서, 학습 데이터 셋 중 기 정의된 수의 미니-배치들을 포함하는 제1 배치에 기초하여 인공지능 모델의 제1 레이어에 대한 학습 연산을 수행할 수 있다. 본 개시의 일 실시예에 있어서, 제1 배치는 제1 정밀도를 가질 수 있다. 예를 들어, 제1 배치는 제1 정밀도를 갖는 블록 부동소수점 값들로 구성될 수 있다. 전자 장치(1000)는 제1 정밀도 환경에서 학습 연산을 수행할 수 있다. 본 개시의 일 실시예에 있어서, 인공지능 모델은 복수의 레이어들을 포함할 수 있다. 예를 들어, 제1 레이어는 복수의 레이어들 중 하나의 레이어일 수 있다.
- [0151] 단계 S1220에서, 전자 장치(1000)는 제1 배치에 대한 학습 연산 결과를 블록 부동소수점 형식으로 변환할 수 있다. 본 개시의 일 실시예에 있어서, 제1 배치에 대한 학습 연산 결과는 부동소수점 형식으로 표현될 수 있다.
- [0152] 단계 S1230에서, 전자 장치(1000)는 변환된 학습 연산 결과에 기초하여, 학습 데이터 셋 중 기 정의된 수의 미니-배치들을 포함하는 제2 배치의 제2 정밀도를 결정할 수 있다. 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 제1 배치에 대한 언더플로우 발생 개수에 기초하여 제2 배치의 제2 정밀도를 결정할 수 있다.
- [0153] 단계 S1240에서, 전자 장치(1000)는 제1 에포크에서, 제2 그룹에 기초하여 제1 레이어에 대한 학습 연산을 수행할 수 있다. 전자 장치(1000)는 제2 정밀도 환경에서 학습 연산을 수행할 수 있다. 본 개시의 일 실시예에 있어서, 제1 배치와 제2 배치는 하나의 에포크에서 연산되는 서로 다른 데이터 그룹일 수 있다. 예를 들어, 제2 배치는 제1 배치의 미니-배치들 다음으로 인공지능 모델에 입력되는 미니-배치들로 구성될 수 있다.
- [0155] 도 13은 도 12의 단계 S1230의 세부 단계들을 보여주는 흐름도이다. 도 1 내지 12에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3, 10, 및 12을 참조하여, 도 13을 설명한다.
- [0156] 도 13을 참조하면, 도 12의 단계 S1230은 단계 S1310 및 S1320을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S1310 및 S1320은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S1310 및 S1320은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 단계 S1230의 세부 단계들

은 도 13에 도시된 바에 한정되지 않으며, 도 13에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 13에 도시되지 않은 단계를 더 포함할 수도 있다.

[0157] 단계 S1310에서, 전자 장치(1000)는 변환된 학습 연산 결과에 대한 언더플로우 발생 개수를 결정할 수 있다. 전자 장치(1000)는 기 정의된 수의 미니-배치들을 포함하는 하나의 배치(예컨대, 제1 배치)에 기초하여 하나의 레이어(예컨대, 제2 레이어)에 대한 학습 연산을 수행하는 동안 발생하는 언더플로우 발생 개수를 카운팅할 수 있다.

[0158] 단계 S1320에서, 전자 장치(1000)는 결정된 언더플로우 발생 개수에 기초하여 다음에 입력될 기 정의된 수의 미니-배치들을 포함하는 하나의 배치(예컨대, 제2 배치) 및/또는 하나의 레이어(예컨대, 제1 레이어)의 파라미터들에 대응하는 값들의 제2 정밀도를 결정할 수 있다. 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 결정된 언더플로우 발생 개수에 기초하여 제2 정밀도를 제1 정밀도보다 큰 정밀도 또는 작은 정밀도를 갖도록 결정할 수 있다.

[0160] 도 14는 도 13의 단계 S1320의 세부 단계들을 보여주는 흐름도이다. 도 1 내지 13에서 설명한 내용과 중복되는 내용은 생략한다. 설명의 편의를 위해, 도 3, 10, 12 및 13을 참조하여, 도 14를 설명한다.

[0161] 도 14를 참조하면, 도 13의 단계 S1320은 단계 S1410 내지 S1440을 포함할 수 있다. 본 개시의 일 실시예에 있어서, 단계 S1410 내지 S1440은 전자 장치(1000), 전자 장치(1000)의 프로세서(1300) 및 전자 장치(1000)의 하드웨어 가속기(1400) 중 적어도 하나에 의해 수행될 수 있다. 그러나 본 개시는 이에 한정되지 않으며, 단계 S1410 내지 S1440은 임의의 전자 장치에 의해 수행될 수도 있다. 본 개시의 일 실시예에 따른 단계 S1320의 세부 단계들은 도 14에 도시된 바에 한정되지 않으며, 도 14에 도시된 단계들 중 적어도 하나를 생략할 수도 있고, 도 14에 도시되지 않은 단계를 더 포함할 수도 있다.

[0162] 단계 S1410에서, 전자 장치(1000)는 언더플로우의 발생 개수가 제1 임계 개수보다 많은지를 결정할 수 있다. 제1 임계 개수는 사용자 또는 제조사의 설정에 따라 미리 정의될 수 있다. 언더플로우의 발생 개수가 제1 임계 개수보다 많은 것으로 결정한 것에 기초하여(예), 절차는 단계 S1420으로 이동한다. 언더플로우의 발생 개수가 제1 임계 개수보다 많지 않은 것으로 결정한 것에 기초하여(아니오), 절차는 단계 S1430으로 이동한다.

[0163] 단계 S1420에서, 전자 장치(1000)는 제2 정밀도를 제1 정밀도보다 높은 정밀도로 결정할 수 있다. 예를 들어, 제1 정밀도가 4인 경우, 전자 장치(1000)는 제2 정밀도를 8로 결정할 수 있으나, 본 개시는 이에 한정되지 않으며, 전자 장치(1000)는 제2 정밀도를, 4 보다 더 큰 임의의 정수 값으로 결정할 수 있다. 단계 S1420 이후, 절차는 단계 S1240으로 이동한다.

[0164] 단계 S1430에서, 전자 장치(1000)는 언더플로우의 발생 개수가 제2 임계 개수보다 적은지를 결정할 수 있다. 제2 임계 개수는 사용자 또는 제조사의 설정에 따라 미리 정의될 수 있다. 언더플로우의 발생 개수가 제2 임계 개수보다 적은 것으로 결정한 것에 기초하여(예), 절차는 단계 S1440으로 이동한다. 언더플로우의 발생 개수가 제1 임계 개수보다 많지 않은 것으로 결정한 것에 기초하여(아니오), 절차는 단계 S1240으로 이동한다.

[0165] 단계 S1440에서, 전자 장치(1000)는 제2 정밀도를 제1 정밀도보다 낮은 정밀도로 결정할 수 있다. 예를 들어, 제1 정밀도가 16 인 경우, 전자 장치(1000)는 제2 정밀도를 8로 결정할 수 있으나, 본 개시는 이에 한정되지 않으며, 전자 장치(1000)는 제2 정밀도를, 16 보다 더 작은 임의의 정수 값으로 결정할 수 있다. 단계 S1440 이후, 절차는 단계 S1240으로 이동한다.

[0166] 본 개시의 일 실시예에 있어서, 도 14에 도시된 바와 달리, 단계 S1430이 단계 S1410보다 먼저 수행될 수 있다. 예를 들어, 언더플로우 발생 개수가 제2 임계 개수보다 적은지를 먼저 결정하고, 언더플로우 발생 개수가 제1 임계 개수보다 많은지를 그 다음에 결정할 수 있다.

[0168] 도 15는 본 개시의 일 실시예에 따른 하드웨어 가속기의 구성을 설명하기 위한 블록도이다. 도 15의 하드웨어 가속기(1400)의 구성, 기능, 및 동작은 도 3의 하드웨어 가속기(1400)의 구성, 기능, 및 동작에 대응할 수 있다. 도 1 내지 14에서 설명한 내용과 중복되는 내용은 생략한다.

[0169] 도 15를 참조하면, 하드웨어 가속기(1400)는 프로세싱 코어(1410), FP2BFP 변환기(1420), 출력 버퍼(1430), 배치 정규화기(1440), 및 활성화 함수 및 풀링 모듈(1450)을 포함할 수 있다.

- [0170] 프로세싱 코어(1410)는 복수의 서브 코어들(1411) 및 선택적 가산기 트리(1412)를 포함할 수 있다. 도 15에서는 프로세싱 코어(1410)가 4 개의 서브 코어들(1411_1, 1411_2, 1411_3, 1411_4)을 포함하는 것으로 도시되나, 본 개시는 이에 한정되지 않는다. 본 개시의 일 실시예에 있어서, 프로세싱 코어(1410)는 제1 서브 코어(1411_1)(Sub-Core 0로도 지칭될 수 있음), 제2 서브 코어(1411_2)(Sub-Core 1로도 지칭될 수 있음), 제3 서브 코어(1411_3)(Sub-Core 2로도 지칭될 수 있음), 및 제4 서브 코어(1411_4)(Sub-Core 3으로도 지칭될 수 있음), 및 선택적 가산기 트리(1412)를 포함할 수 있다.
- [0171] 제1 내지 제4 서브 코어들(1411_1, 1411_2, 1411_3, 1411_4) 각각은 복수의 프로세싱 엘리먼트들을 포함할 수 있다. 제1 내지 제4 서브 코어들(1411_1, 1411_2, 1411_3, 1411_4) 각각은 복수의 출력 포트들을 통해 연산 결과 값들을 선택적 가산기 트리(1412)로 전달할 수 있다. 제1 내지 제4 서브 코어들(1411_1, 1411_2, 1411_3, 1411_4)의 연산 결과 값들은 선택적 가산기 트리(1412)에 의해 합산될 수 있다. 도시되지 않았지만, 선택적 가산기 트리(1412)는 공유 지수 간의 지수 곱셈 결과 값을 전달받을 수 있다. 선택적 가산기 트리(1412)는 지수 곱셈 결과 값 및 제1 내지 제4 서브 코어들(1411_1, 1411_2, 1411_3, 1411_4)의 연산 결과 값들에 기초하여 부동소수점 값을 출력할 수 있다. 선택적 가산기 트리(1412)는 부동소수점 값을 출력 버퍼(1430)로 전달할 수 있다.
- [0172] 제1 내지 제4 서브 코어들(1411_1, 1411_2, 1411_3, 1411_4) 각각은 한 사이클에 기 정의된 수의 기 정의된 정밀도를 갖는 MAC 연산을 수행할 수 있다. 예를 들어, 제1 내지 제4 서브 코어들(1411_1, 1411_2, 1411_3, 1411_4) 각각은 한 사이클에 128 개의 INT4 MAC 연산을 수행할 수 있으나, 본 개시는 이에 한정되지 않는다.
- [0173] 출력 버퍼(1430)는 프로세싱 코어(1410)로부터 출력되는 데이터를 수신할 수 있다. 출력 버퍼(1430)는 프로세싱 코어(1410)로부터 출력되는 데이터를 저장할 수 있다. 출력 버퍼(1430)는 프로세싱 코어(1410)로부터 출력되는 데이터를 배치 정규화기(1440)에 전달할 수 있다. 본 개시의 일 실시예에 있어서, 출력 버퍼(1430)는 프로세싱 코어(1410)로부터 출력되는 데이터를 큐잉(queueing)할 수 있다. 출력 버퍼(1430)의 기 정의된 크기의 큐가 가득 차면, 출력 버퍼(1430)는 큐잉된 데이터를 배치 정규화기(1440)에 전달할 수 있다. 본 개시의 일 실시예에 있어서, 출력 버퍼(1430)는 하드웨어 가속기(1400)의 내부 메모리의 적어도 일부로 구성될 수 있다.
- [0174] 배치 정규화기(1440)는 배치 정규화(batch normalization, BN)를 수행할 수 있다. 배치 정규화기(1440)는 출력 버퍼(1430)의 출력에 기초하여 배치 정규화를 수행할 수 있다. 본 개시의 일 실시예에 따르면, 배치 정규화는 더 빠른 수렴으로 가중치 파라미터를 찾기 위하여 사용되는 처리 방식으로, 내부 공변량(internal covariant shift)을 줄임으로써 훈련 과정을 보다 안정적으로 수행하게 한다.
- [0175] 활성화 함수 및 풀링(pooling) 모듈(1450)은 배치 정규화기(1440)의 출력에 기초하여 활성화 함수를 적용하고, 풀링 동작을 수행할 수 있다. 본 개시의 일 실시예에 있어서, 활성화 함수 및 풀링 모듈(1450)은 활성화 함수를 구현하는 제1 로직 회로 및 풀링 동작을 구현하는 제2 로직 회로로 구성될 수 있다.
- [0176] 활성화 함수는 ReLU, 시그모이드, tanh, leaky ReLU, PReLU, ELU, SELU 등과 같은 임의의 활성화 함수일 수 있으나 본 개시는 이에 한정되지 않으며, 사용자 또는 제조사의 설정에 따라 상이할 수 있다.
- [0177] 본 개시에서, 풀링은 입력 데이터를 기 정의된 기준으로 선별(또는 추출)하여 입력 데이터의 공간 차원을 줄이는 동작을 의미할 수 있다. 풀링은 서브샘플링(subsampling) 또는 다운샘플링(downsampling)으로 지칭될 수도 있다. 예를 들어, 풀링은 맥스 풀링(max pooling) 및/또는 평균 풀링(average pooling)과 같은 임의의 풀링일 수 있으나, 본 개시는 이에 한정되지 않으며, 사용자 또는 제조사의 설정에 따라 상이할 수 있다. 본 개시에서, 맥스 풀링은 입력 데이터에서 최대값을 추출하는 동작을 의미하고, 평균 풀링은 입력 데이터에서 평균값을 추출하는 동작을 의미한다.
- [0178] FP2BFP 변환기(1420)는 활성화 함수 및 풀링(pooling) 모듈(1450)의 출력에 기초하여 부동소수점 형식의 값들을 블록 부동소수점 형식의 값들로 변환할 수 있다. 프로세싱 코어(1410)의 출력, 출력 버퍼(1430)의 출력, 배치 정규화기(1440)의 출력 및, 활성화 함수 및 풀링(pooling) 모듈(1450) 각각의 출력은 부동소수점 형식의 값들로 표현될 수 있다. FP2BFP 변환기(1420)는 변환된 블록 부동소수점 형식의 값들을 메모리(1200)에 저장할 수 있다. FP2BFP 변환기(1420)는 변환 과정에서 발생하는 언더플로우 개수를 카운팅할 수 있다.
- [0180] 도 16은 본 개시의 일 실시예에 따른 서브 코어의 구성 및 동작을 예시적으로 설명하기 위한 블록도이다. 도 16의 프로세싱 코어(1410)의 구성, 기능, 및 동작은 도 3 및 15의 프로세싱 코어(1410)의 구성, 기능, 및 동작에

대응할 수 있다. 도 1 내지 15에서 설명한 내용과 중복되는 내용은 생략한다.

- [0181] 도 16을 참조하면, 서브 코어(1411)는 프로세싱 엘리먼트(processing element; PE) 어레이(PEA), 입력 버퍼(IB), 가중치 버퍼(WB), 복수의 시프터들, 및 복수의 가산기들을 포함할 수 있다.
- [0182] PE 어레이(PEA)는 복수의 프로세싱 엘리먼트들(PE)을 포함할 수 있다. 도 15에서, PE 어레이(PEA)에 16 개의 프로세싱 엘리먼트들(PE)을 포함하는 것으로 도시되었으나, PE 어레이(PEA)에 포함되는 프로세싱 엘리먼트들(PE)의 개수는 이에 한정되지 않는다. 프로세싱 엘리먼트들(PE) 각각은 적어도 하나의 곱셈기를 포함할 수 있다. 곱셈기는 부호가 있는 연산 및/또는 부호가 없는 연산을 지원할 수 있다. 본 개시의 일 실시예에 있어서, 곱셈기는 4 비트 곱셈기일 수 있으나, 본 개시는 이에 한정되지 않으며, 곱셈기는 다양한 정밀도의 부호 및 맨티서를 처리할 수 있는 임의의 곱셈기로 구현될 수 있다. 예를 들어, 곱셈기는 4 비트의 입력 값과 4 비트의 가중치 값을 피연산자로 하여 곱셈을 수행할 수 있다.
- [0183] 본 개시의 일 실시예에 있어서, 복수의 프로세싱 엘리먼트들(PE)은 다양한 정밀도를 지원하기 위해 다양한 개수(예컨대, 2, 4, 6, 8, 16 등)와 다양한 방식(예컨대, 1X1, 2X2, 2X1, 4X4, 4X2 등)으로 묶일 수 있다.
- [0184] 입력 버퍼(IB)는 입력 값(또는 입력 데이터, 입력 피쳐 맵(feature map), 입력 텐서 등으로도 지칭될 수도 있음)을 수신할 수 있다. 입력 버퍼(IB)는 입력 값을 PE 어레이(PEA)로 전송할 수 있다. 가중치 버퍼(WB)는 가중치 값(또는 가중치 데이터, 가중치 커널, 가중치 텐서 등으로도 지칭될 수도 있음)을 수신할 수 있다. 가중치 버퍼(WB)는 가중치 값을 PE 어레이(PEA)로 전송할 수 있다. 본 개시의 일 실시예에 있어서, 입력 버퍼(IB) 및 가중치 버퍼(WB)는 하드웨어 가속기(1400)의 내부 메모리의 적어도 일부로 구성될 수 있다.
- [0185] 복수의 시프터들 각각은 입력되는 비트들에 대한 시프팅을 수행할 수 있다. 예를 들어, 복수의 시프터들 각각은 왼쪽 시프팅을 수행하는 것을 가정하여 이하 설명하나, 본 개시는 이에 한정되지 않으며, 오른쪽 시프팅을 수행할 수도 있다. 복수의 시프터들은 임의의 비트 수 만큼 왼쪽 시프팅을 수행할 수 있으며, 예를 들어, 4 비트 내지 12 비트 만큼 왼쪽 시프팅을 수행할 수 있다. 복수의 시프터들은 프로세서(1300) 및/또는 하드웨어 가속기(1400)가 전달한 제어 신호에 응답하여 활성화되거나 비활성화될 수 있다.
- [0186] 복수의 가산기들 각각은 입력되는 비트들에 대한 합산 연산을 수행할 수 있다. 복수의 가산기들 각각은 정수 가산기로 구현될 수 있으나, 본 개시는 이에 한정되지 않는다. 복수의 가산기들은 프로세서(1300) 및/또는 하드웨어 가속기(1400)가 전달한 제어 신호에 응답하여 활성화되거나 비활성화될 수 있다.
- [0187] 본 개시의 일 실시예에 있어서, 4 개의 프로세싱 엘리먼트들(PE)은 4 개의 시프터들, 3 개의 가산기들에 연결될 수 있다. 여기서, 4 개의 프로세싱 엘리먼트들(PE)은 1 X 4의 형태, 즉 1 개 행 및 4 개 열을 갖도록 배치될 수 있다. 예를 들어, 제1 프로세싱 엘리먼트는 제1 시프터(S1)와 연결될 수 있다. 제1 시프터(S1)가 활성화된 경우, 제1 프로세싱 엘리먼트의 출력 값은 활성화된 제1 시프터(S1)를 통과할 수 있다. 제1 시프터(S1)는 기 정의된 비트 수만큼 출력 값을 시프팅할 수 있다. 제1 시프터(S1)가 활성화되지 않은 경우, 제1 프로세싱 엘리먼트의 출력 값은 제1 가산기(A1) 또는 선택적 가산기 트리(1412)로 전달될 수 있다. 제1 가산기(A1)가 활성화된 경우, 제1 시프터(S1)를 통과한 출력 값은 제1 가산기(A1)를 통과할 수 있다. 제1 가산기(A1)가 비활성화된 경우, 제1 시프터(S1)를 통과한 출력 값은 선택적 가산기 트리(1412)로 전달될 수 있다.
- [0188] 제2 프로세싱 엘리먼트는 제2 시프터(S2)와 연결될 수 있다. 제2 시프터(S2)가 활성화된 경우, 제2 프로세싱 엘리먼트의 출력 값은 활성화된 제2 시프터(S2)를 통과할 수 있다. 제2 시프터(S2)는 기 정의된 비트 수만큼 출력 값을 시프팅할 수 있다. 제2 시프터(S2)가 활성화되지 않은 경우, 제2 프로세싱 엘리먼트의 출력 값은 제1 가산기(A1) 또는 선택적 가산기 트리(1412)로 전달될 수 있다. 제1 가산기(A1)가 활성화된 경우, 제2 시프터(S2)를 통과한 출력 값은 제1 가산기(A1)를 통과할 수 있다. 제1 가산기(A1)가 비활성화된 경우, 제1 시프터(S1)를 통과한 출력 값은 선택적 가산기 트리(1412)로 전달될 수 있다. 제1 가산기(A1)는 제1 프로세싱 엘리먼트의 출력 값(또는 시프팅된 출력 값)에 제2 프로세싱 엘리먼트의 출력 값(또는 시프팅된 출력 값)을 가산할 수 있다. 활성화된 제1 가산기(A1)를 통과한 출력 값은 제3 가산기(A3) 또는 선택적 가산기 트리(1412)로 전달될 수 있다.
- [0189] 제3 프로세싱 엘리먼트는 제3 시프터(S3)와 연결될 수 있다. 제3 시프터(S3)가 활성화된 경우, 제3 프로세싱 엘리먼트의 출력 값은 활성화된 제3 시프터(S3)를 통과할 수 있다. 제3 시프터(S3)는 기 정의된 비트 수만큼 출력 값을 시프팅할 수 있다. 제3 시프터(S3)가 활성화되지 않은 경우, 제3 프로세싱 엘리먼트의 출력 값은 제2 가산기(A2) 또는 선택적 가산기 트리(1412)로 전달될 수 있다. 제2 가산기(A2)가 활성화된 경우, 제3 시프터(S3)를 통과한 출력 값은 제2 가산기(A2)를 통과할 수 있다. 제2 가산기(A2)가 비활성화된 경우, 제3 시프터(S3)를 통과한 출력 값은 선택적 가산기 트리(1412)로 전달될 수 있다.

- [0190] 제4 프로세싱 엘리먼트는 제4 시프터(S4)와 연결될 수 있다. 제4 시프터(S4)가 활성화된 경우, 제4 프로세싱 엘리먼트의 출력 값은 활성화된 제4 시프터(S4)를 통과할 수 있다. 제4 시프터(S4)는 기 정의된 비트 수만큼 출력 값을 시프팅할 수 있다. 제4 시프터(S4)가 활성화되지 않은 경우, 제4 프로세싱 엘리먼트의 출력 값은 제2 가산기(A2) 또는 선택적 가산기 트리(1412)로 전달될 수 있다. 제2 가산기(A2)가 활성화된 경우, 제4 시프터(S4)를 통과한 출력 값은 제2 가산기(A2)를 통과할 수 있다. 제2 가산기(A2)가 비활성화된 경우, 제4 시프터(S4)를 통과한 출력 값은 선택적 가산기 트리(1412)로 전달될 수 있다. 제2 가산기(A2)는 제3 프로세싱 엘리먼트의 출력 값(또는 시프팅된 출력 값)에 제4 프로세싱 엘리먼트의 출력 값(또는 시프팅된 출력 값)을 가산할 수 있다. 활성화된 제2 가산기(A2)를 통과한 출력 값은 제3 가산기(A3) 또는 선택적 가산기 트리(1412)로 전달될 수 있다. 제3 가산기(A3)가 활성화된 경우, 제3 가산기(A3)는 제1 가산기의 출력 값에 제2 가산기의 출력 값을 가산할 수 있다. 활성화된 제3 가산기(A3)를 통과한 출력 값은 선택적 가산기 트리(1412)로 전달될 수 있다.
- [0191] 도 16 내지 18에서, 파선으로 표현된 시프터들 및/또는 가산기들은 비활성화된 것을 표현하는 것이고, 실선으로 표현된 시프터들 및/또는 가산기들은 활성화된 것을 표현한 것이다.
- [0192] 예시적으로, 도 16은 입력 값과 가중치 값이 모두 4 비트로 구성되는 경우의 서브 코어(1411)의 동작을 도시한다. 프로세싱 엘리먼트들(PE) 각각은 4 비트의 가중치 값(W[3:0])과 4 비트의 입력 값(X[3:0]) 간의 곱셈 연산을 수행할 수 있다. 프로세싱 엘리먼트들(PE) 각각은 곱셈 연산 결과를 선택적 가산기 트리(1412)로 전달할 수 있다. 본 개시의 일 실시예에 있어서, 복수의 시프터들 및 복수의 가산기들은 활성화되지 않을 수 있다. 프로세서(1300) 및/또는 하드웨어 가속기(1400)는 복수의 시프터들 및 복수의 가산기들이 활성화되지 않도록 제어할 수 있다.
- [0194] 도 17은 본 개시의 일 실시예에 따른 서브 코어의 구성 및 동작을 예시적으로 설명하기 위한 블록도이다. 도 17의 서브 코어(1411)의 구성, 기능, 및 동작은 도 15의 서브 코어(1411)의 구성, 기능, 및 동작에 대응할 수 있다. 도 1 내지 16에서 설명한 내용과 중복되는 내용은 생략한다.
- [0195] 도 17과 함께, 도 3을 참조하면, 도 17은 입력 값과 가중치 값이 모두 8 비트로 구성되는 경우의 서브 코어(1411)의 동작을 도시한다. PE 어레이(PEA)의 16 개의 프로세싱 엘리먼트들은 4 개씩 묶일 수 있다. 제1 내지 제4 그룹들(G1, G2, G3, G4) 각각은 4 개의 프로세싱 엘리먼트들을 포함할 수 있다. 예를 들어, 제1 내지 제4 그룹들(G1, G2, G3, G4) 각각은 2 X 2 형태로 배치된 4 개의 프로세싱 엘리먼트들을 포함할 수 있다. 제1 내지 제4 그룹들(G1, G2, G3, G4) 각각은 8 비트의 가중치 값(W[0:7])과 8 비트의 입력 값(X[0:7]) 간의 곱셈 연산을 수행할 수 있다. 제1 내지 제4 그룹들(G1, G2, G3, G4) 각각은 곱셈 연산 결과를 시프터 및/또는 가산기로 전달할 수 있다.
- [0196] 이하에서는, 제2 그룹(G2)의 동작을 예시로 설명한다. 제2 그룹(G2)은 8 비트의 가중치 값(W[7:0])과 8 비트의 입력 값(X[7:0]) 간의 곱셈 연산을 수행하도록 구성될 수 있다. 가중치 값(W[7:0])과 입력 값(X[7:0]) 각각은 2 개의 4 비트 서브-워드들로 구분될 수 있다. 예를 들어, 가중치 값(W[7:0])은 제1 가중치 서브-워드(W[7:4])와 제2 가중치 서브-워드(W[3:0])로 구분될 수 있다. 예를 들어, 입력 값(X[7:0])은 제1 입력 서브-워드(X[7:4])와 제2 입력 서브-워드(X[3:0])로 구분될 수 있다. 제2 그룹(G2)은 제1 내지 제4 프로세싱 엘리먼트들을 포함할 수 있다.
- [0197] 제1 프로세싱 엘리먼트(PE1)는 제1 가중치 서브-워드(W[7:4])와 제1 입력 서브-워드(X[7:4]) 간의 곱셈 연산을 수행할 수 있다. 제1 프로세싱 엘리먼트(PE1)에 연결된 시프터는 활성화되고, 8 비트 왼쪽 시프팅을 수행할 수 있다. 제1 프로세싱 엘리먼트(PE1)는 시프팅된 연산 결과 값(또는 부분 합으로도 지칭될 수도 있음)을 제3 프로세싱 엘리먼트(PE3)에 전달할 수 있다.
- [0198] 제2 프로세싱 엘리먼트(PE2)는 제1 가중치 서브-워드(W[7:4])와 제2 입력 서브-워드(X[3:0]) 간의 곱셈 연산을 수행할 수 있다. 제2 프로세싱 엘리먼트(PE2)에 연결된 시프터는 활성화되고, 4 비트 왼쪽 시프팅을 수행할 수 있다. 제2 프로세싱 엘리먼트(PE2)는 시프팅된 연산 결과 값(또는 부분 합으로도 지칭될 수도 있음)을 제4 프로세싱 엘리먼트(PE4)에 전달할 수 있다.
- [0199] 제3 프로세싱 엘리먼트(PE3)는 제2 가중치 서브-워드(W[3:0])와 제1 입력 서브-워드(X[7:4]) 간의 곱셈 연산을 수행할 수 있다. 제3 프로세싱 엘리먼트(PE3)에 연결된 시프터는 활성화되고, 시프터는 4 비트 왼쪽 시프팅을 수행할 수 있다. 제3 프로세싱 엘리먼트(PE3)에 연결된 가산기는 활성화되고, 가산기는 제3 프로세싱 엘리먼트(PE3)의 시프팅된 연산 결과 값에 제1 프로세싱 엘리먼트(PE1)의 시프팅된 연산 결과 값을 가산함으로써 제1 가

산 값을 출력할 수 있다.

- [0200] 제4 프로세싱 엘리먼트(PE4)는 제2 가중치 서브-워드(W[3:0])과 제2 입력 서브-워드(X[3:0]) 간의 곱셈 연산을 수행할 수 있다. 제4 프로세싱 엘리먼트(PE4)에 연결된 가산기는 활성화되고, 가산기는 제4 프로세싱 엘리먼트(PE4)의 연산 결과 값에 제2 프로세싱 엘리먼트(PE2)의 시프팅된 연산 결과 값을 가산함으로써 제2 가산 값을 출력할 수 있다.
- [0201] 제3 프로세싱 엘리먼트(PE3)와 제4 프로세싱 엘리먼트(PE4)에 연결된 가산기는 활성화되고, 가산기는 제1 가산 값에 제2 가산 값을 가산함으로써 제3 가산 값을 출력할 수 있다. 제3 가산 값은 선택적 가산기 트리(1412)에 전달될 수 있다.
- [0203] 도 18은 본 개시의 일 실시예에 따른 서브 코어의 구성 및 동작을 예시적으로 설명하기 위한 블록도이다. 도 18의 서브 코어(1411)의 구성, 기능, 및 동작은 도 15의 서브 코어(1411)의 구성, 기능, 및 동작에 대응할 수 있다. 도 1 내지 17에서 설명한 내용과 중복되는 내용은 생략한다.
- [0204] 도 18과 함께, 도 3을 참조하면, 도 18은 입력 값은 16 비트 및 가중치 값은 4 비트로 구성되는 경우의 서브 코어(1411)의 동작을 도시한다. PE 어레이(PEA)의 16 개의 프로세싱 엘리먼트들은 4 개씩 묶일 수 있다. 제5 내지 제8 그룹들(G5, G6, G7, G8) 각각은 4 개의 프로세싱 엘리먼트들을 포함할 수 있다. 예를 들어, 제5 내지 제8 그룹들(G5, G6, G7, G8) 각각은 4 X 1 형태로 배치된 4 개의 프로세싱 엘리먼트들을 포함할 수 있다. 제5 내지 제8 그룹들(G5, G6, G7, G8) 각각은 4 비트의 가중치 값(W[3:0])과 16 비트의 입력 값(X[15:0]) 간의 곱셈 연산을 수행할 수 있다. 제5 내지 제8 그룹들(G5, G6, G7, G8) 각각은 곱셈 연산 결과를 시프터 및/또는 가산기로 전달할 수 있다.
- [0205] 이하에서는, 제8 그룹(G8)의 동작을 예시로 설명한다. 제8 그룹(G8)은 4 비트의 가중치 값(W[3:0])과 16 비트의 입력 값(X[15:0]) 간의 곱셈 연산을 수행하도록 구성될 수 있다. 16 비트의 입력 값(X[15:0])은 4 개의 4 비트 서브-워드들로 구분될 수 있다. 예를 들어, 입력 값(X[15:0])은 제1 입력 서브-워드(X[15:12]), 제2 입력 서브-워드(X[11:8]), 제3 입력 서브-워드(X[7:4]), 및 제4 입력 서브-워드(X[3:0])로 구분될 수 있다. 제8 그룹(G8)은 제5 내지 제8 프로세싱 엘리먼트들(PE5, PE6, PE7, PE8)을 포함할 수 있다.
- [0206] 제5 프로세싱 엘리먼트(PE5)는 가중치 값(W[3:0])과 제1 입력 서브-워드(X[15:12]) 간의 곱셈 연산을 수행할 수 있다. 제5 프로세싱 엘리먼트(PE5)에 연결된 시프터는 활성화되고, 시프터 12 비트 왼쪽 시프팅을 수행할 수 있다. 제5 프로세싱 엘리먼트(PE5)는 시프팅된 연산 결과 값(또는 제1 부분 합으로도 지칭될 수도 있음)을 제6 프로세싱 엘리먼트(PE6)에 전달할 수 있다.
- [0207] 제6 프로세싱 엘리먼트(PE6)는 가중치 값(W[3:0])과 제2 입력 서브-워드(X[11:8]) 간의 곱셈 연산을 수행할 수 있다. 제6 프로세싱 엘리먼트(PE6)에 연결된 시프터는 활성화되고, 시프터는 8 비트 왼쪽 시프팅을 수행할 수 있다. 제6 프로세싱 엘리먼트(PE6)에 연결된 가산기는 활성화되고, 가산기는 제1 부분 합에 시프팅된 연산 결과 값을 가산함으로써 제2 부분 합을 출력할 수 있다. 제6 프로세싱 엘리먼트(PE6)는 제2 부분 합을 제7 프로세싱 엘리먼트(PE7)에 전달할 수 있다.
- [0208] 제7 프로세싱 엘리먼트(PE7)는 가중치 값(W[3:0])과 제3 입력 서브-워드(X[7:4]) 간의 곱셈 연산을 수행할 수 있다. 제7 프로세싱 엘리먼트(PE7)에 연결된 시프터는 활성화되고, 시프터는 4 비트 왼쪽 시프팅을 수행할 수 있다. 제7 프로세싱 엘리먼트(PE7)에 연결된 가산기는 활성화되고, 가산기는 제2 부분 합에 시프팅된 연산 결과 값을 가산함으로써 제3 부분 합을 출력할 수 있다. 제7 프로세싱 엘리먼트(PE7)는 제3 부분 합을 제8 프로세싱 엘리먼트(PE8)에 전달할 수 있다.
- [0209] 제8 프로세싱 엘리먼트(PE8)는 가중치 값(W[3:0])과 제4 입력 서브-워드(X[3:0]) 간의 곱셈 연산을 수행할 수 있다. 제8 프로세싱 엘리먼트(PE8)에 연결된 가산기는 활성화되고, 가산기는 제8 프로세싱 엘리먼트(PE8)의 연산 결과 값에 제3 부분 합을 가산할 수 있다. 가산 결과 값은 선택적 가산기 트리(1412)에 전달될 수 있다.
- [0210] 도 16 내지 18을 참조하면, 프로세싱 코어(1410)는 입력 버퍼(IB)를 통해 학습 데이터 셋(또는 입력 값)에 대응하는 제1 텐서를 수신할 수 있다. 프로세싱 코어(1410)는 가중치 버퍼(WB)를 통해 가중치 값에 대응하는 제2 텐서를 수신할 수 있다. 프로세싱 코어(1410)는 복수의 곱셈기들을 이용하여, 제1 텐서와 제2 텐서 간의 곱셈 연산을 수행할 수 있다. 본 개시의 일 실시예에 있어서, 제1 텐서의 정밀도와 제2 텐서의 정밀도는 다양할 수 있다. 본 개시의 일 실시예에 있어서, 제1 텐서의 정밀도와 제2 텐서의 정밀도는 동일하거나 상이할 수 있다.

- [0211] 본 개시의 일 실시예에 있어서, 프로세싱 코어(1410)는 공유 지수 핸들러(미도시)를 이용하여 제1 텐서의 공유 지수와 제2 텐서의 공유 지수 간의 연산을 처리할 수 있다. 프로세싱 코어(1410)는 제1 텐서의 부호 및 맨티서와 제2 텐서의 부호 맨티서 간의 MAC 연산을 수행할 수 있다. 여기서, 부호 및 맨티서는 맨티서의 암시적 비트를 포함할 수 있다. 예를 들어, MAC 연산의 결과 값은 서브 코어들(1411)의 출력 값들을 의미할 수 있다. 본 개시의 일 실시예에 따르면, 프로세싱 코어(1410)는 복수의 곱셈기들을 다양한 방식으로 묶음(grouping)으로써, 다양한 정밀도(또는 비트 폭)에 대한 곱셈 연산을 지원할 수 있다.
- [0213] 도 19a 및 19b는 본 개시의 일 실시예에 따른 선택적 가산기 트리(1412)의 구성 및 동작을 예시적으로 설명하기 위한 블록도들이다. 도 19a 및 19b의 선택적 가산기 트리(1412)의 구성, 기능, 및 동작은 도 15의 선택적 가산기 트리(1412)의 구성, 기능, 및 동작에 대응할 수 있다. 도 1 내지 18에서 설명한 내용과 중복되는 내용은 생략한다.
- [0214] 선택적 가산기 트리(1412)는 INT2FP 변환기(1413) 및 적어도 하나의 가산기를 포함할 수 있다. 선택적 가산기 트리(1412)는 다양한 블록 크기에 적응적으로 동작할 수 있다. 프로세싱 코어(1410)의 서브 코어들 각각이 동일한 블록의 값들에 대한 MAC 연산을 수행하는 경우와 프로세싱 코어(1410)의 서브 코어들이 서로 다른 블록의 값들에 대한 MAC 연산을 수행하는 경우, INT2FP 변환기(1413)는 다른 방식으로 동작할 수 있다. 예시적으로, 도 19a는 프로세싱 코어(1410)의 서브 코어들(1411_1~1411_4) 각각이 동일한 블록의 값들에 대한 MAC 연산을 수행하는 경우를 설명하기 위한 도면이다. 예시적으로, 도 19b는 프로세싱 코어(1410)의 서브 코어들(1411_1~1411_4)이 서로 다른 블록의 값들에 대한 MAC 연산을 수행하는 경우를 설명하기 위한 도면이다.
- [0215] 도 19a를 참조하면, 하나의 서브 코어가 1 번의 사이클에 128 개의 INT4 MAC 연산을 수행하고, 입력 값과 가중치 값 각각이 16 비트 정밀도를 갖고 블록 크기가 16인 경우를 가정한다. 이 경우, 서브 코어들(1411_1~1411_4) 각각이 서로 다른 공유 지수를 갖는 블록들 간의 연산을 수행할 수 있다. 따라서, 서브 코어들(1411_1~1411_4) 각각의 출력 값은 INT2FP 변환기(1413)로 전달될 수 있다. INT2FP 변환기(1413)는 정수 형식의 출력 값들을 부동소수점 형식의 값들로 변환할 수 있다. INT2FP 변환기(1413)는 공유 지수 핸들러(미도시)로부터 처리된 지수 값을 수신할 수 있다. INT2FP 변환기(1413)는 서브 코어들(1411_1~1411_4) 각각의 출력 값과 지수 값에 기초하여 부동소수점 값을 생성할 수 있다. 제1 부동소수점 가산기는 제1 서브 코어(1411_1)의 출력 값에 대응하는 부동소수점 값에 제2 서브 코어(1411_2)의 출력 값에 대응하는 부동소수점 값을 가산함으로써 제1 가산 값을 출력할 수 있다. 제2 부동소수점 가산기는 제3 서브 코어(1411_3)의 출력 값에 대응하는 부동소수점 값에 제4 서브 코어(1411_4)의 출력 값에 대응하는 부동소수점 값을 가산함으로써 제2 가산 값을 출력할 수 있다. 제3 부동소수점 가산기는 제1 가산 값에 제2 가산 값을 가산함으로써 최종 부동소수점 값을 출력할 수 있다.
- [0216] 도 19b를 참조하면, 하나의 서브 코어가 1 번의 사이클에 128 개의 INT4 MAC 연산을 수행하고, 입력 값과 가중치 값 각각이 16 비트 정밀도를 갖고 블록 크기가 64인 경우를 가정한다. 이 경우, 서브 코어들(1411_1~1411_4) 각각이 같은 공유 지수를 갖는 블록들 간의 연산을 수행할 수 있다. 따라서, 서브 코어들 각각의 출력 값은 정수 가산기로 전달될 수 있다. 제1 정수 가산기는 제1 서브 코어(1411_1)의 출력 값에 제2 서브 코어(1411_2)의 출력 값을 가산함으로써 제3 가산 값을 출력할 수 있다. 제2 정수 가산기는 제3 서브 코어(1411_3)의 출력 값에 제4 서브 코어(1411_4)의 출력 값을 가산함으로써 제4 가산 값을 출력할 수 있다. 제3 정수 가산기는 제3 가산 값에 제4 가산 값을 가산함으로써 최종 정수 값을 출력할 수 있다. INT2FP 변환기(1413)는 최종 정수 값을 부동소수점 형식의 값으로 변환할 수 있다. INT2FP 변환기(1413)는 공유 지수 핸들러(미도시)로부터 처리된 지수 값을 수신할 수 있다. INT2FP 변환기(1413)는 최종 정수 값과 지수 값에 기초하여 부동소수점 값을 생성할 수 있다. 본 개시의 일 실시예에 따르면, 부동소수점 가산 대신 정수 가산을 이용함으로써, 메모리 부하 및 에너지 소모를 획기적으로 줄일 수 있다.
- [0217] 본 개시의 일 실시예에 있어서, 프로세싱 코어(1410)의 선택적 가산기 트리(1412)는 지수 곱셈 연산 및 MAC 연산에 기초하여 부동소수점 형식의 곱셈 연산의 결과 값을 획득할 수 있다.
- [0219] 도 20은 본 개시의 일 실시예에 따른 전자 장치(1000)의 동작을 설명하기 위한 개념도이다. 도 20의 전자 장치(1000)의 구성, 기능, 및 동작은 도 3의 전자 장치(1000)의 구성, 기능, 및 동작에 대응할 수 있다. 도 1 내지 19에서 설명한 내용과 중복되는 내용은 생략한다.
- [0220] 도 20과 함께, 도 3을 참조하면, 전자 장치(1000)는 인공지능 모델에 대응하는 정보를 획득할 수 있다. 예를 들

어, 전자 장치(1000)는 메모리(1200)로부터 인공지능 모델에 대응하는 정보를 로드할 수 있다. 예를 들어, 전자 장치(1000)는 외부 서버로부터 인공지능 모델에 대응하는 정보를 수신할 수 있다. 예를 들어, 전자 장치(1000)는 입출력 인터페이스(1100)를 통해 학습시키고자 하는 대상 인공지능 모델에 대응하는 정보를 입력받을 수 있다. 예를 들어, 인공지능 모델에 대응하는 정보는, 인공지능 모델의 레이어의 개수, 레이어의 종류, 데이터 차원, 동작 모드, 정밀도, 활성화 함수 및 풀링 옵션 등에 대한 데이터일 수 있으나, 본 개시는 이에 한정되지 않으며, 인공지능 모델에 대응하는 정보는 상술한 다양한 하이퍼파라미터를 포함할 수 있다.

- [0221] 본 개시의 일 실시예에 있어서, 전자 장치(1000)는 프로세서(1300), FSM 블록(1500), 신호 제어 분배기(1600), 하드웨어 가속기(1400), 및 메모리(1200)를 포함할 수 있다.
- [0222] 프로세서(1300)는 인공지능 모델에 대한 정보에 기초하여 인공지능 모델의 학습을 위한 제어 신호를 생성할 수 있다.
- [0223] FSM(finite state machine) 블록(1500)은 하드웨어 가속기(1400)의 동작 상태를 식별할 수 있다. FSM 블록(1500)은 제어 신호를 수신할 수 있다. FSM 블록(1500)은 하드웨어 가속기(1400)의 동작 상태 기초하여 제어 신호를 최적화할 수 있다.
- [0224] 신호 제어 분배기(1600)는 FSM 블록(1500)에 의해 최적화된 제어 신호를 하드웨어 가속기(1400) 내의 각 구성에 제공할 수 있다.
- [0225] 하드웨어 가속기(1400)는 블록 부동소수점 형식으로 구성되는 텐서들 간의 곱셈 연산을 수행할 수 있다. 하드웨어 가속기(1400)는 곱셈 연산 결과 값을 블록 부동소수점 형식으로 메모리(1200)에 저장할 수 있다.
- [0227] 본 개시의 일 실시예에 따른 부동소수점 값을 블록 부동소수점 값으로 변환하는 방법이 제공될 수 있다. 상기 방법은, 복수의 부동소수점 값들을 획득하는 단계를 포함할 수 있다. 상기 방법은, 상기 복수의 부동소수점 값들 중 최대 지수를 갖는 적어도 하나의 제1 부동소수점 값의 지수를 공유 지수로 결정하는 단계를 포함할 수 있다. 상기 방법은, 상기 적어도 하나의 제1 부동소수점 값의 인덱스(index)를 메모리에 저장하는 단계를 포함할 수 있다. 상기 방법은, 상기 복수의 부동소수점 값들 중 상기 최대 지수를 갖지 않는 적어도 하나의 제2 부동소수점 값의 맨티사(mantissa)의 암시적 비트(implicit bit)와 명시적 비트들(explicit bits)을, 상기 공유 지수와 상기 적어도 하나의 제2 부동소수점 값의 지수의 차이만큼 오른쪽 시프팅(right shifting)하는 단계를 포함할 수 있다. 상기 방법은, 상기 적어도 하나의 제1 부동소수점 값의 부호 및 맨티사, 상기 적어도 하나의 제2 부동소수점 값의 부호 및 맨티사, 그리고 상기 공유 지수로 구성되는 복수의 블록 부동소수점 값들을 상기 메모리에 저장하는 단계를 포함할 수 있다.
- [0228] 본 개시의 일 실시예에 있어서, 상기 방법은, 상기 적어도 하나의 제1 부동소수점 값의 부호 및 맨티사와 상기 적어도 하나의 제2 부동소수점 값의 부호 및 맨티사 각각이, 기 정의된 비트 폭(bit width)을 갖도록 조정하는 단계를 포함할 수 있다.
- [0229] 본 개시의 일 실시예에 있어서, 상기 방법은, 상기 적어도 하나의 제2 부동소수점 값에 대한 언더플로우(underflow) 발생 개수를 결정하는 단계를 포함할 수 있다.
- [0230] 본 개시의 일 실시예에 있어서, 상기 방법은, 상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 기 정의된 비트 폭을 변경하는 단계를 포함할 수 있다.
- [0231] 본 개시의 일 실시예에 있어서, 상기 기 정의된 비트 폭을 결정하는 단계는, 상기 결정된 언더플로우의 발생 개수가 제1 임계 개수보다 많은지를 결정하는 단계를 포함할 수 있다. 상기 기 정의된 비트 폭을 결정하는 단계는, 상기 결정된 언더플로우의 발생 개수가 상기 제1 임계 개수보다 많은 것으로 결정한 것에 기초하여, 상기 기 정의된 비트 폭을 더 큰 비트 폭으로 변경하는 단계를 포함할 수 있다.
- [0232] 본 개시의 일 실시예에 있어서, 상기 기 정의된 비트 폭을 결정하는 단계는, 상기 결정된 언더플로우의 발생 개수가 제2 임계 개수보다 적은지를 결정하는 단계를 포함할 수 있다. 상기 기 정의된 비트 폭을 결정하는 단계는, 상기 결정된 언더플로우의 발생 개수가 상기 제2 임계 개수보다 적은 것으로 결정한 것에 기초하여, 상기 기 정의된 비트 폭을 더 작은 비트 폭으로 변경하는 단계를 포함할 수 있다.
- [0233] 본 개시의 일 실시예에 있어서, 상기 언더플로우의 발생 개수는, 기 정의된 수의 미니-배치(mini-batch)들을 포함하는 하나의 배치(batch)에 기초하여 인공지능 모델의 하나의 레이어(layer)에 대한 학습 연산을 수행하는 과

정에서 발생한 언더플로어가 카운팅된 개수일 수 있다.

- [0234] 본 개시의 일 실시예에 있어서, 상기 방법은, 상기 복수의 블록 부동소수점 값에 기초하여 인공지능 모델에 대한 학습 연산을 수행하는 단계를 포함할 수 있다. 상기 방법은, 기 정의된 수의 에포크(epoch) 만큼 상기 학습 연산이 수행되었는지를 결정하는 단계를 포함할 수 있다. 상기 방법은, 상기 기 정의된 수의 에포크 만큼 상기 학습 연산이 수행된 것으로 결정한 것에 기초하여, 상기 복수의 부동소수점 값들에 대응하는 블록 크기를 변경하는 단계를 포함할 수 있다.
- [0235] 본 개시의 일 실시예에 있어서, 상기 메모리에 저장된 상기 인덱스에 대응하는 값의 암시적 비트는 제1 값이고, 상기 메모리에 저장된 상기 인덱스에 대응하지 않는 값의 암시적 비트는 제2 값일 수 있다.
- [0236] 본 개시의 일 실시예에 있어서, 상기 적어도 하나의 제1 부동소수점 값의 맨티서의 제1 암시적 비트는 제1 값이고, 상기 적어도 하나의 제2 부동소수점 값의 맨티서의 제2 암시적 비트는 제2 값일 수 있다.
- [0237] 본 개시의 일 실시예에 있어서, 전자 장치가 제공될 수 있다. 상기 전자 장치는 적어도 하나의 인스트럭션을 저장하는 메모리를 포함할 수 있다. 상기 전자 장치는 상기 적어도 하나의 인스트럭션을 실행하는 적어도 하나의 프로세서를 포함할 수 있다. 상기 적어도 하나의 프로세서는, 복수의 부동소수점 값들을 획득할 수 있다. 상기 적어도 하나의 프로세서는, 상기 복수의 부동소수점 값들 중 최대 지수를 갖는 적어도 하나의 제1 부동소수점 값의 지수를 공유 지수로 결정할 수 있다. 상기 적어도 하나의 프로세서는, 상기 적어도 하나의 제1 부동소수점 값의 인덱스(index)를 상기 메모리에 저장할 수 있다. 상기 적어도 하나의 프로세서는, 상기 복수의 부동소수점 값들 중 상기 최대 지수를 갖지 않는 적어도 하나의 제2 부동소수점 값의 맨티서(mantissa)의 암시적 비트(implicit bit)와 명시적 비트들(explicit bits)을, 상기 공유 지수에서 상기 적어도 하나의 제2 부동소수점 값의 지수의 차이만큼 오른쪽 시프팅(right shifting)할 수 있다. 상기 적어도 하나의 프로세서는, 상기 적어도 하나의 제1 부동소수점 값의 부호 및 맨티서, 상기 적어도 하나의 제2 부동소수점 값의 부호 및 맨티서, 그리고 상기 공유 지수로 구성되는 복수의 블록 부동소수점 값들을 상기 메모리에 저장할 수 있다.
- [0238] 본 개시의 일 실시예에 있어서, 블록 부동소수점 값을 처리하는 방법이 제공될 수 있다. 상기 방법은, 공유 지수를 갖는 복수의 블록 부동소수점 값들 및 적어도 하나의 최대 지수 인덱스를 획득하는 단계를 포함할 수 있다. 상기 방법은, 상기 복수의 블록 부동소수점 값들 각각의 인덱스가 상기 적어도 하나의 최대 지수 인덱스에 대응하는지를 결정할 수 있다. 상기 방법은, 상기 인덱스가 상기 적어도 하나의 최대 지수 인덱스에 대응하는 것으로 결정하는 것에 기초하여, 상기 인덱스에 대응하는 제1 블록 부동소수점 값의 제1 암시적 비트를 제1 값으로 결정할 수 있다. 상기 방법은, 상기 인덱스가 상기 적어도 하나의 최대 지수 인덱스에 대응하지 않는 것으로 결정하는 것에 기초하여, 상기 인덱스에 대응하는 제2 블록 부동소수점 값의 제2 암시적 비트를 제2 값으로 결정할 수 있다.
- [0239] 본 개시의 일 실시예에 있어서, 하드웨어 가속기를 이용하여 인공지능 모델을 학습하는 방법이 제공될 수 있다. 상기 방법은, 제1 블록 크기를 갖는 복수의 제1 블록 부동소수점 값들로 구성되는 학습 데이터 셋을 획득하는 단계를 포함할 수 있다. 상기 방법은, 상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계를 포함할 수 있다. 상기 방법은, 기 정의된 수의 에포크(epoch)들 동안 상기 인공지능 모델에 대한 학습 연산을 반복하는 단계를 포함할 수 있다. 상기 방법은, 상기 학습 데이터 셋을 제2 블록 크기를 갖는 복수의 제2 블록 부동소수점 값들로 재구성하는 단계를 포함할 수 있다.
- [0240] 본 개시의 일 실시예에 있어서, 상기 방법은, 상기 재구성된 학습 데이터 셋에 기초하여 상기 학습 연산을 수행하는 단계를 포함할 수 있다. 상기 방법은, 상기 기 정의된 수의 에포크들 동안 상기 인공지능 모델에 대한 학습 연산을 반복하는 단계를 포함할 수 있다.
- [0241] 본 개시의 일 실시예에 있어서, 상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계는, 상기 기 정의된 수의 에포크들 중 제1 에포크에서, 상기 기 정의된 수의 에포크들 중 제1 에포크에서, 상기 학습 데이터 셋 중 기 정의된 수의 미니-배치들을 포함하는 제1 배치(batch)에 기초하여 상기 인공지능 모델의 제1 레이어(layer)에 대한 학습 연산을 수행하는 단계를 포함할 수 있다. 본 개시의 일 실시예에 있어서, 상기 제1 배치는 제1 정밀도를 가질 수 있다. 상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계는, 부동소수점 형식의 상기 제1 배치에 대한 학습 연산 결과를 블록 부동소수점 형식으로 변환하는 단계를 포함할 수 있다. 상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계는, 상기 변환된 학습 연산 결과에 기초하여, 상기 학습 데이터 셋 중 상기 기 정의된 수의 미니-배치들을 포함하는 제2 배치의 제2 정밀도를 결정하는 단계를 포함할 수 있다. 상기 학습 데이터 셋

에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계는, 상기 제1 에포크에서, 상기 제2 배치에 기초하여 상기 제1 레이어에 대한 학습 연산을 수행하는 단계를 포함할 수 있다.

[0242] 본 개시의 일 실시예에 있어서, 상기 변환된 학습 연산 결과에 기초하여, 상기 학습 데이터 셋 중 상기 기 정의된 수의 미니-배치들을 포함하는 제2 배치의 제2 정밀도를 결정하는 단계는, 상기 변환된 학습 연산 결과에 대한 언더플로우(underflow) 발생 개수를 결정하는 단계를 포함할 수 있다. 상기 변환된 학습 연산 결과에 기초하여, 상기 학습 데이터 셋 중 상기 기 정의된 수의 미니-배치들을 포함하는 제2 배치의 제2 정밀도를 결정하는 단계는, 상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 제2 정밀도를 결정하는 단계를 포함할 수 있다.

[0243] 본 개시의 일 실시예에 있어서, 상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 제2 정밀도를 결정하는 단계는, 상기 결정된 언더플로우의 발생 개수가 제1 임계 개수보다 많은지를 결정하는 단계를 포함할 수 있다. 상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 제2 정밀도를 결정하는 단계는, 상기 결정된 언더플로우의 발생 개수가 상기 제1 임계 개수보다 많은 것으로 결정한 것에 기초하여, 상기 제2 정밀도를 상기 제1 정밀도보다 높은 정밀도로 결정하는 단계를 포함할 수 있다.

[0244] 본 개시의 일 실시예에 있어서, 상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 제2 정밀도를 결정하는 단계는, 상기 결정된 언더플로우의 발생 개수가 제2 임계 개수보다 적은지를 결정하는 단계를 포함할 수 있다. 상기 결정된 언더플로우의 발생 개수에 기초하여, 상기 제2 정밀도를 결정하는 단계는, 상기 결정된 언더플로우의 발생 개수가 상기 제2 임계 개수보다 적은 것으로 결정한 것에 기초하여, 상기 제2 정밀도를 상기 제1 정밀도보다 낮은 정밀도로 결정하는 단계를 포함할 수 있다.

[0245] 본 개시의 일 실시예에 있어서, 상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계는, 상기 학습 데이터 셋에 대응하는 제1 텐서와 상기 인공지능 모델의 가중치 값에 대응하는 제2 텐서 간의 곱셈 연산을 수행하는 단계를 포함할 수 있다.

[0246] 본 개시의 일 실시예에 있어서, 상기 곱셈 연산을 수행하는 단계는, 상기 제1 텐서의 공유 지수와 상기 제2 텐서의 공유 지수 간의 지수 곱셈 연산을 수행하는 단계를 포함할 수 있다. 상기 곱셈 연산을 수행하는 단계는, 상기 제1 텐서의 부호 및 맨티서와 상기 제2 텐서의 부호 및 맨티서 간의 MAC(multiply and accumulation) 연산을 수행하는 단계를 포함할 수 있다. 상기 곱셈 연산을 수행하는 단계는, 상기 지수 곱셈 연산 및 상기 MAC 연산에 기초하여 부동소수점 형식의 상기 곱셈 연산의 결과 값을 획득하는 단계를 포함할 수 있다.

[0247] 본 개시의 일 실시예에 있어서, 상기 제1 텐서의 정밀도와 상기 제2 텐서의 정밀도는 상이할 수 있다.

[0248] 본 개시의 일 실시예에 있어서, 상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행하는 단계는, 상기 곱셈 연산의 결과 값을 블록 부동소수점 형식으로 변환하는 단계를 포함할 수 있다.

[0249] 본 개시의 일 실시예에 있어서, 전자 장치가 제공될 수 있다. 상기 전자 장치는 적어도 하나의 인스트럭션을 저장하는 메모리를 포함할 수 있다. 상기 전자 장치는 상기 적어도 하나의 인스트럭션을 실행하는 적어도 하나의 프로세서를 포함할 수 있다. 상기 적어도 하나의 프로세서는, 제1 블록 크기를 갖는 복수의 제1 블록 부동소수점 값들로 구성되는 학습 데이터 셋을 획득할 수 있다. 상기 적어도 하나의 프로세서는, 상기 학습 데이터 셋에 기초하여 상기 인공지능 모델에 대한 학습 연산을 수행할 수 있다. 상기 적어도 하나의 프로세서는, 기 정의된 수의 에포크(epoch)들 동안 상기 인공지능 모델에 대한 학습 연산을 반복할 수 있다. 상기 적어도 하나의 프로세서는, 상기 학습 데이터 셋을 제2 블록 크기를 갖는 복수의 제2 블록 부동소수점 값들로 재구성할 수 있다.

[0250] 본 개시의 일 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 매체에 기록되는 프로그램 명령은 본 개시를 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.

[0251] 본 개시의 일부 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의

의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전송 메커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다. 또한, 본 개시의 일부 실시예는 컴퓨터에 의해 실행되는 컴퓨터 프로그램과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 컴퓨터 프로그램 또는 컴퓨터 프로그램 제품(computer program product)으로도 구현될 수 있다.

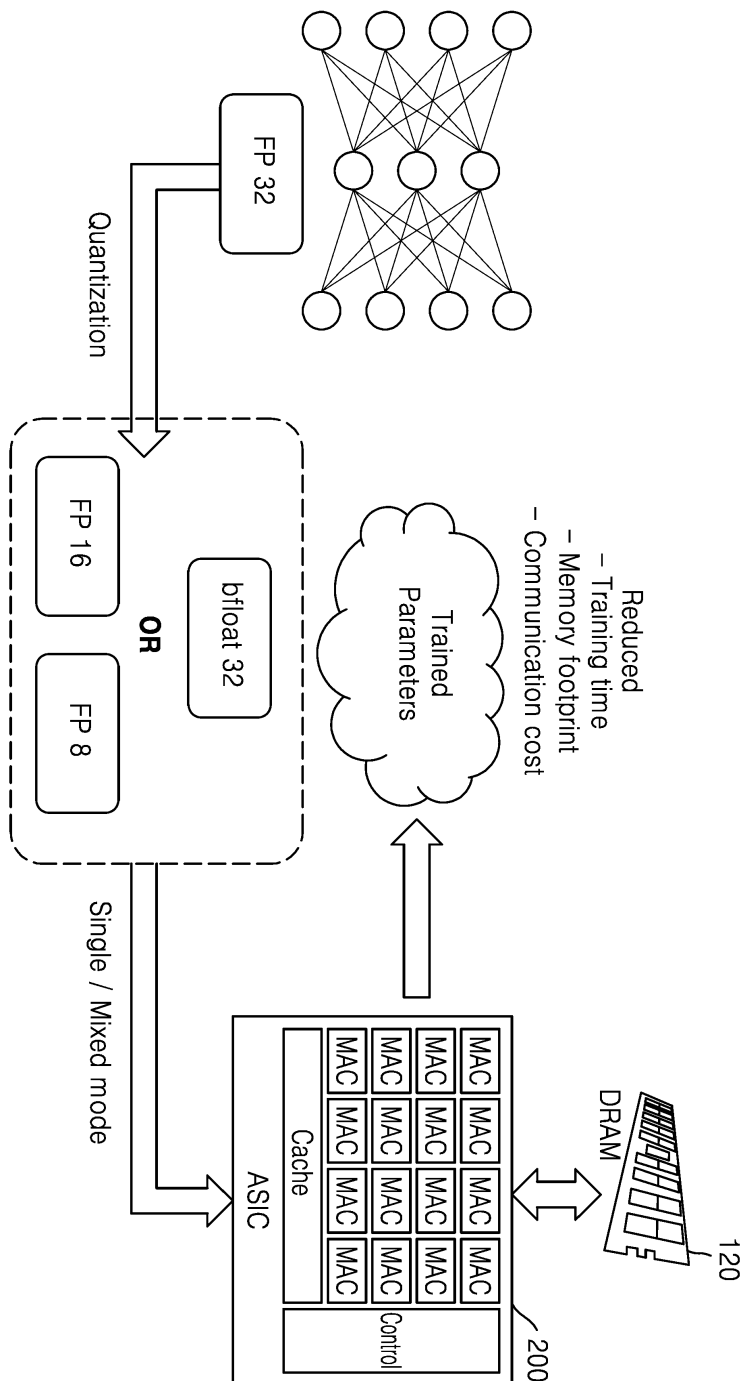
[0252] 본 개시의 일 실시예에 있어서, 기기로 읽을 수 있는 저장매체는, 비일시적(non-transitory) 저장매체의 형태로 제공될 수 있다. 여기서, '비일시적 저장매체'는 실재(tangible)하는 장치이고, 신호(signal)(예: 전자기파)를 포함하지 않는다는 것을 의미할 뿐이며, 이 용어는 데이터가 저장매체에 반영구적으로 저장되는 경우와 임시적으로 저장되는 경우를 구분하지 않는다. 예로, '비일시적 저장매체'는 데이터가 임시적으로 저장되는 버퍼를 포함할 수 있다.

[0253] 본 개시의 실시예에 따르면, 본 문서에 개시된 다양한 실시예들에 따른 방법은 컴퓨터 프로그램 제품(computer program product)에 포함되어 제공될 수 있다. 컴퓨터 프로그램 제품은 상품으로서 판매자 및 구매자 간에 거래될 수 있다. 컴퓨터 프로그램 제품은 기기로 읽을 수 있는 저장 매체(예: compact disc read only memory (CD-ROM))의 형태로 배포되거나, 또는 어플리케이션 스토어를 통해 또는 두개의 사용자 장치들(예: 스마트폰들) 간에 직접, 온라인으로 배포(예: 다운로드 또는 업로드)될 수 있다. 온라인 배포의 경우에, 컴퓨터 프로그램 제품(예: 다운로드할 앱(downloadable app))의 적어도 일부는 제조사의 서버, 어플리케이션 스토어의 서버, 또는 중계 서버의 메모리와 같은 기기로 읽을 수 있는 저장 매체에 적어도 일시 저장되거나, 임시적으로 생성될 수 있다.

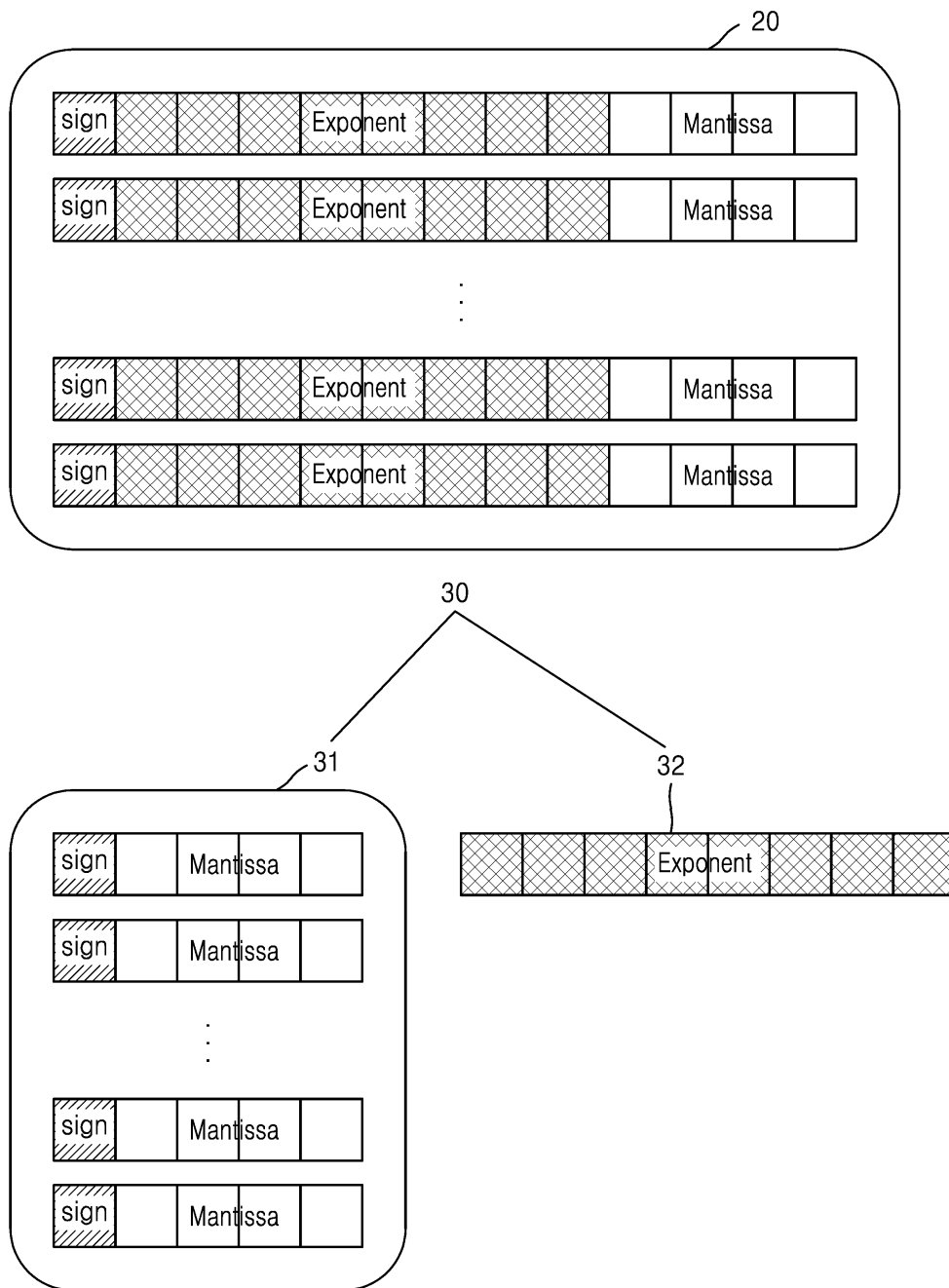
[0254] 이상에서 본 개시의 실시예들에 대하여 상세하게 설명하였지만 본 개시의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 개시의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 개시의 권리범위에 속한다.

도면

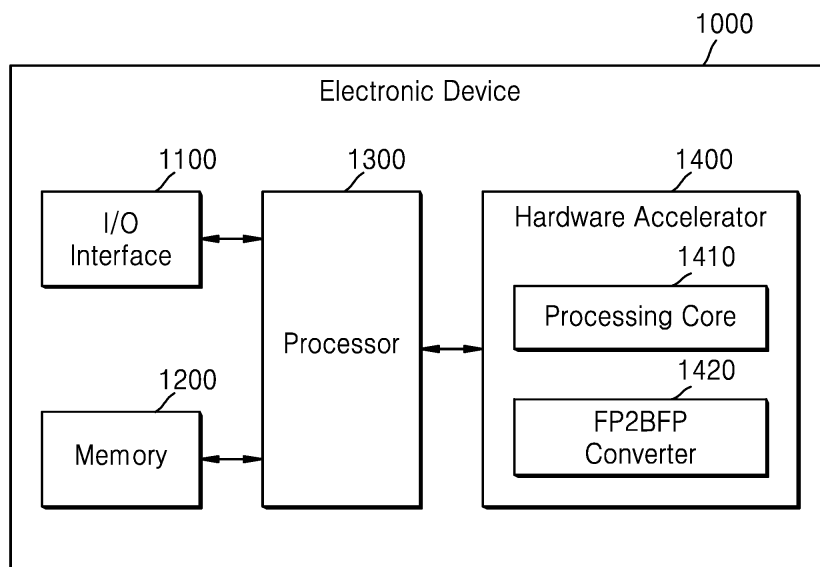
도면1



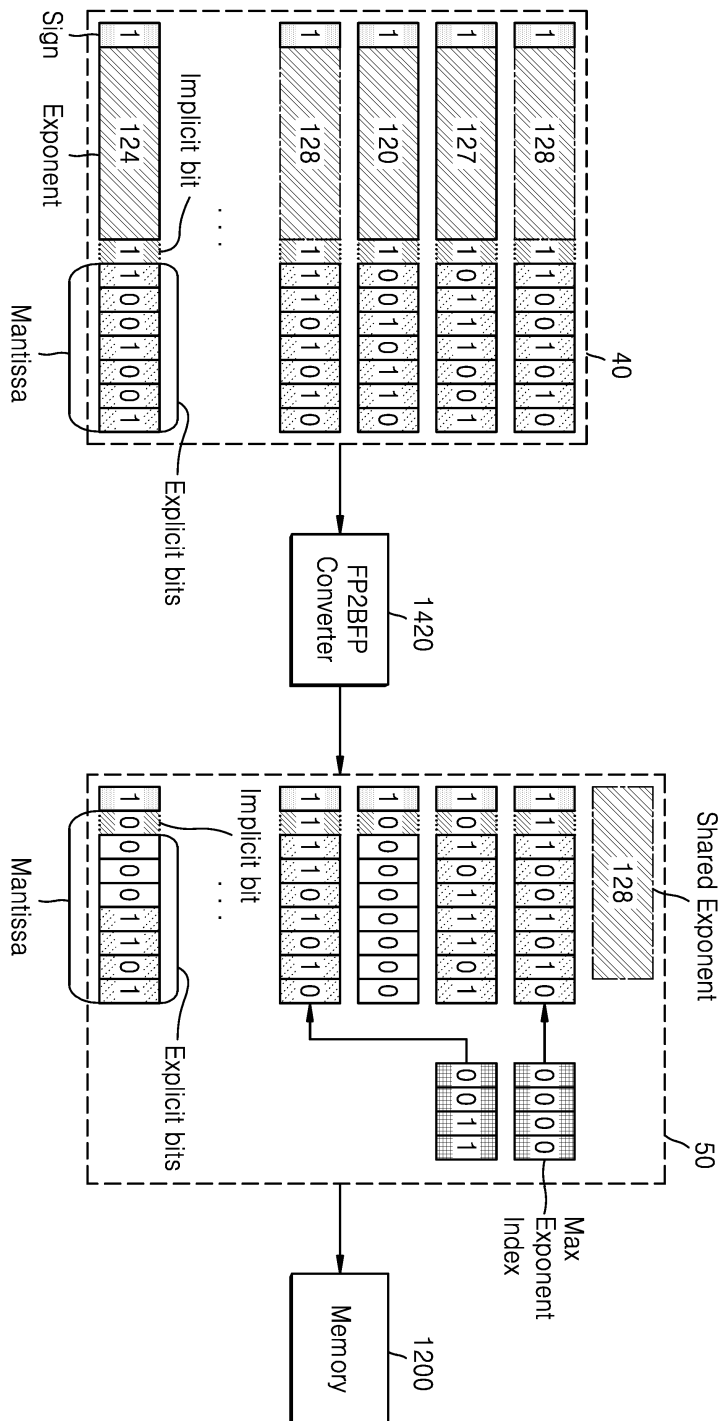
도면2



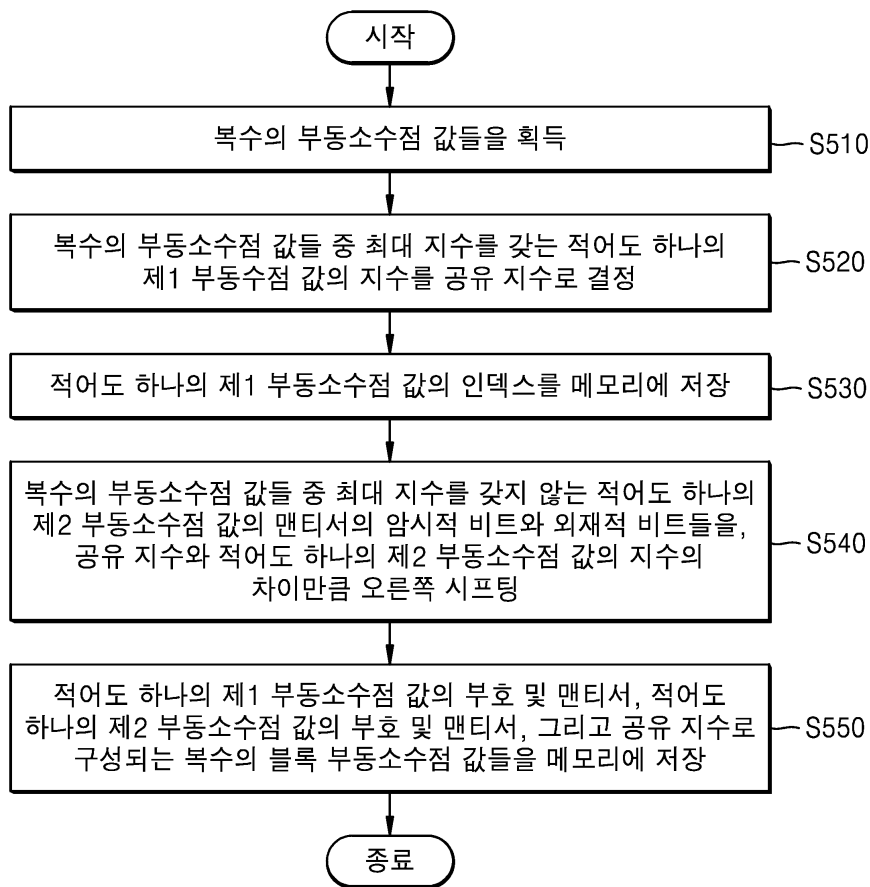
도면3



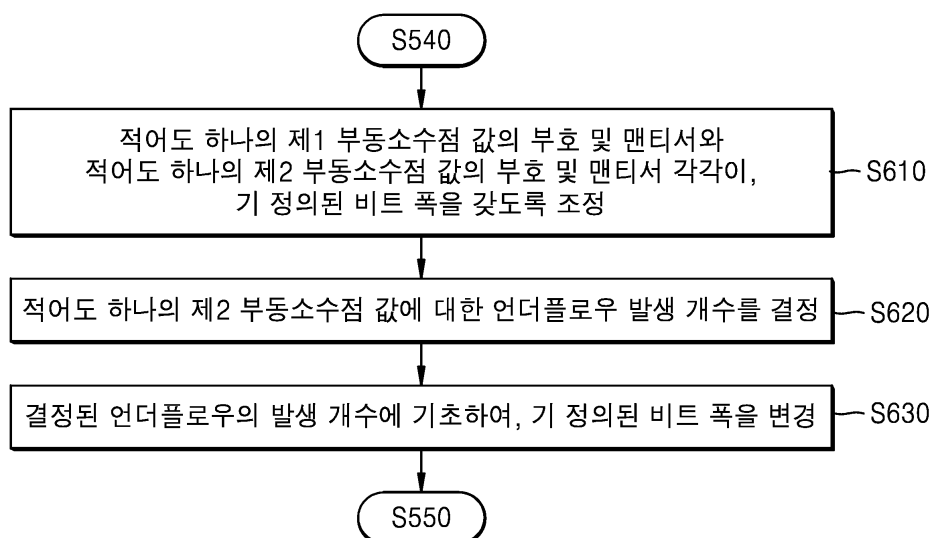
도면4



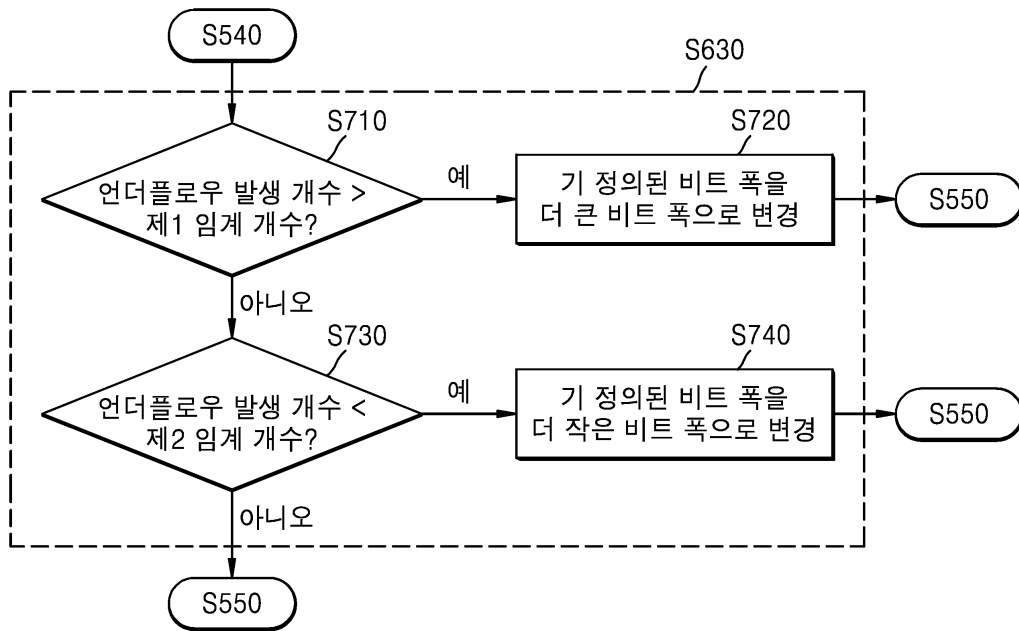
도면5



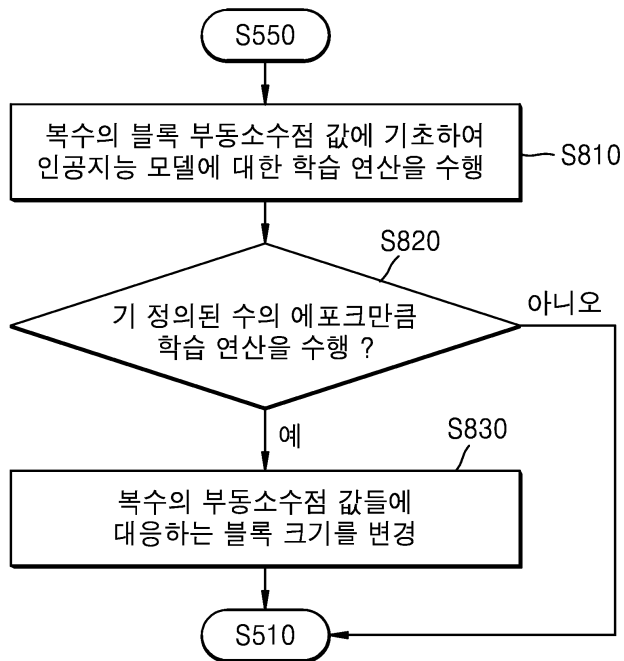
도면6



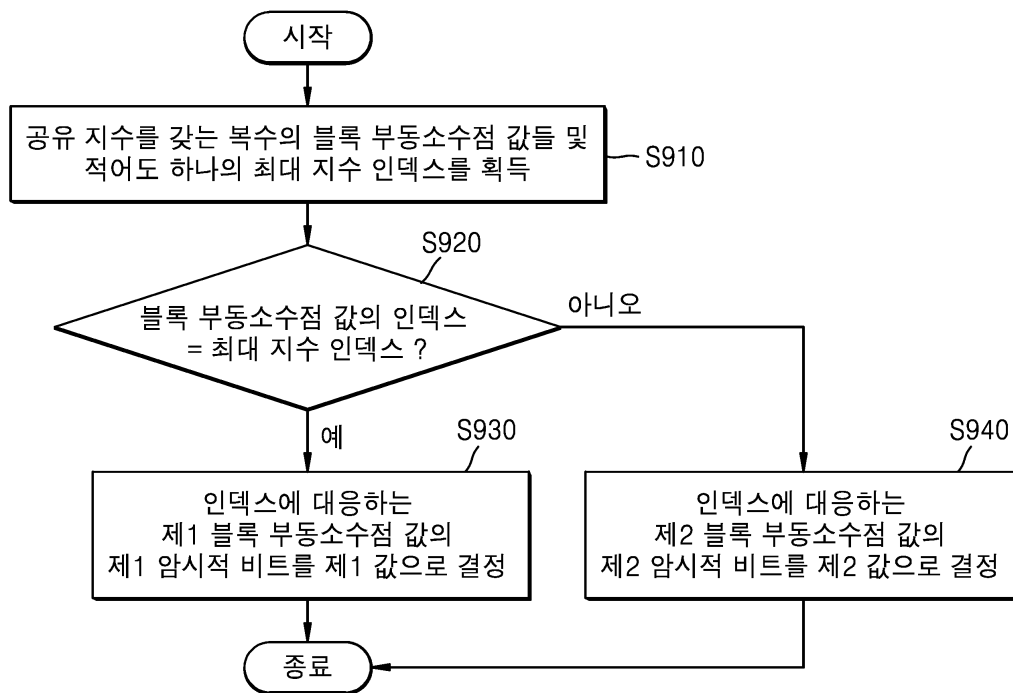
도면7



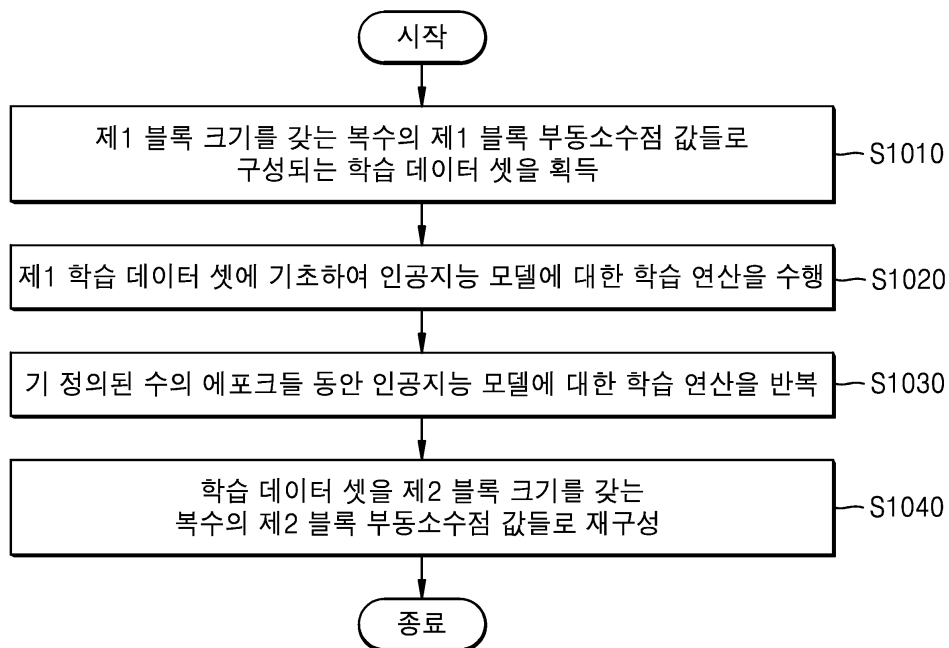
도면8



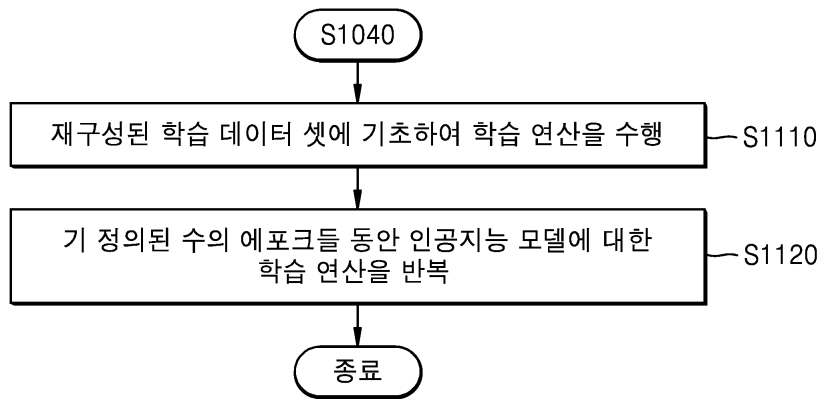
도면9



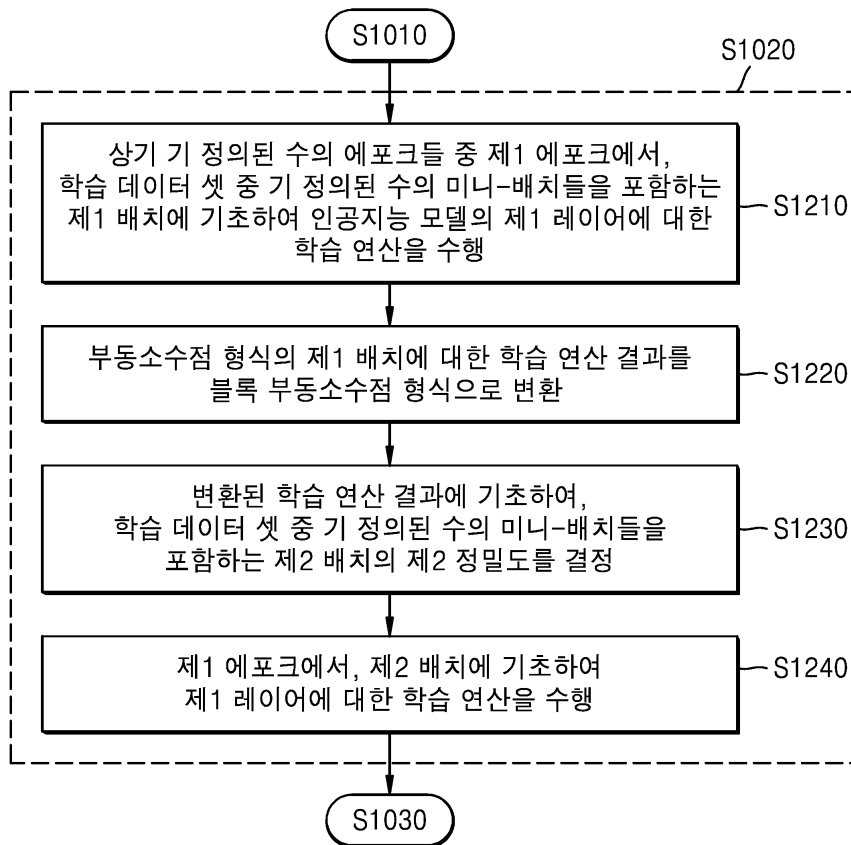
도면10



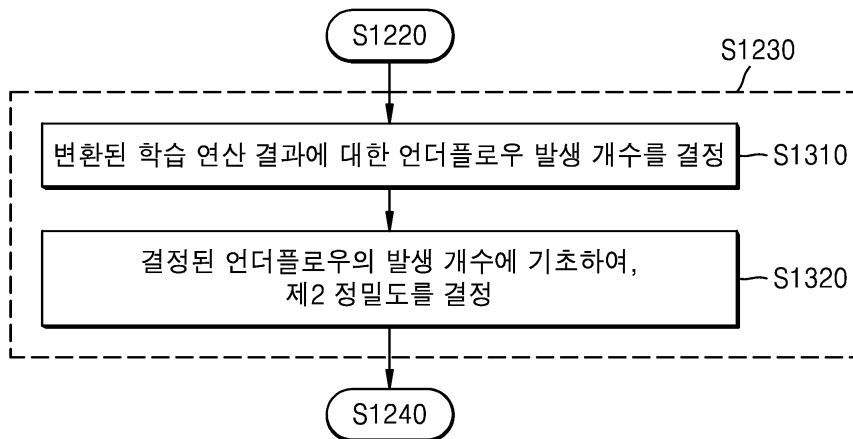
도면11



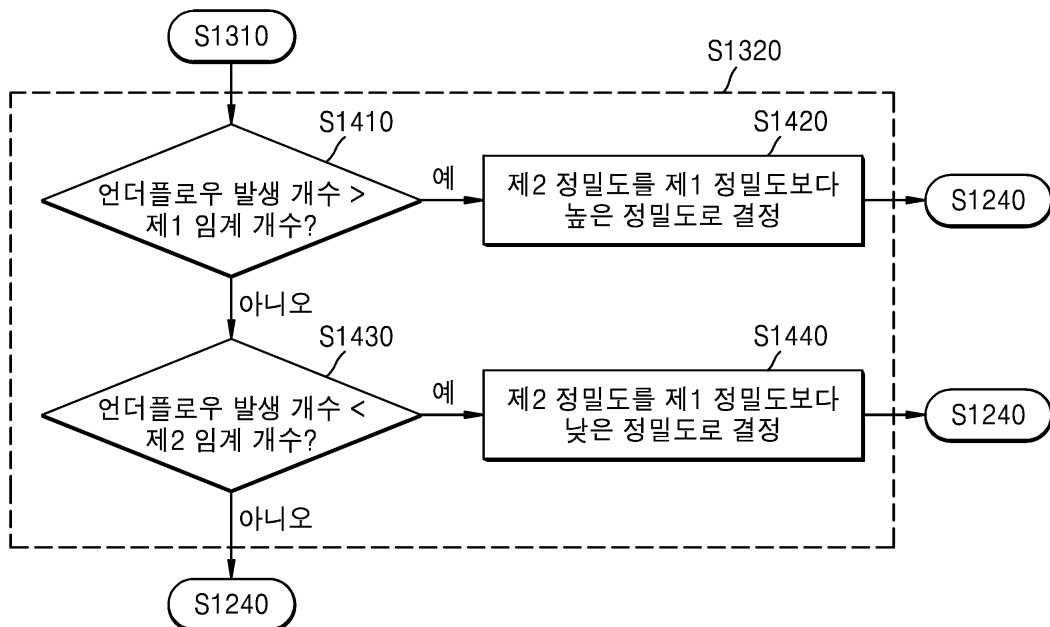
도면12



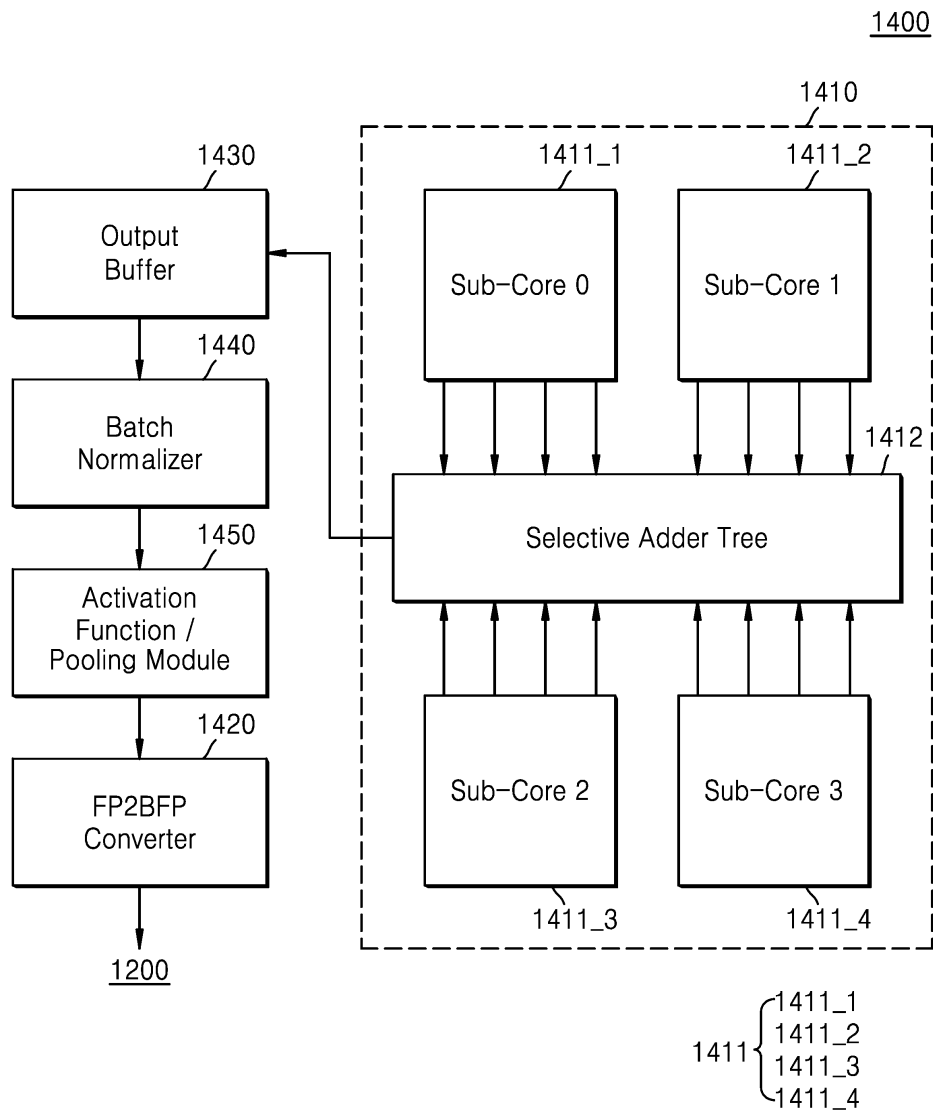
도면13



도면14

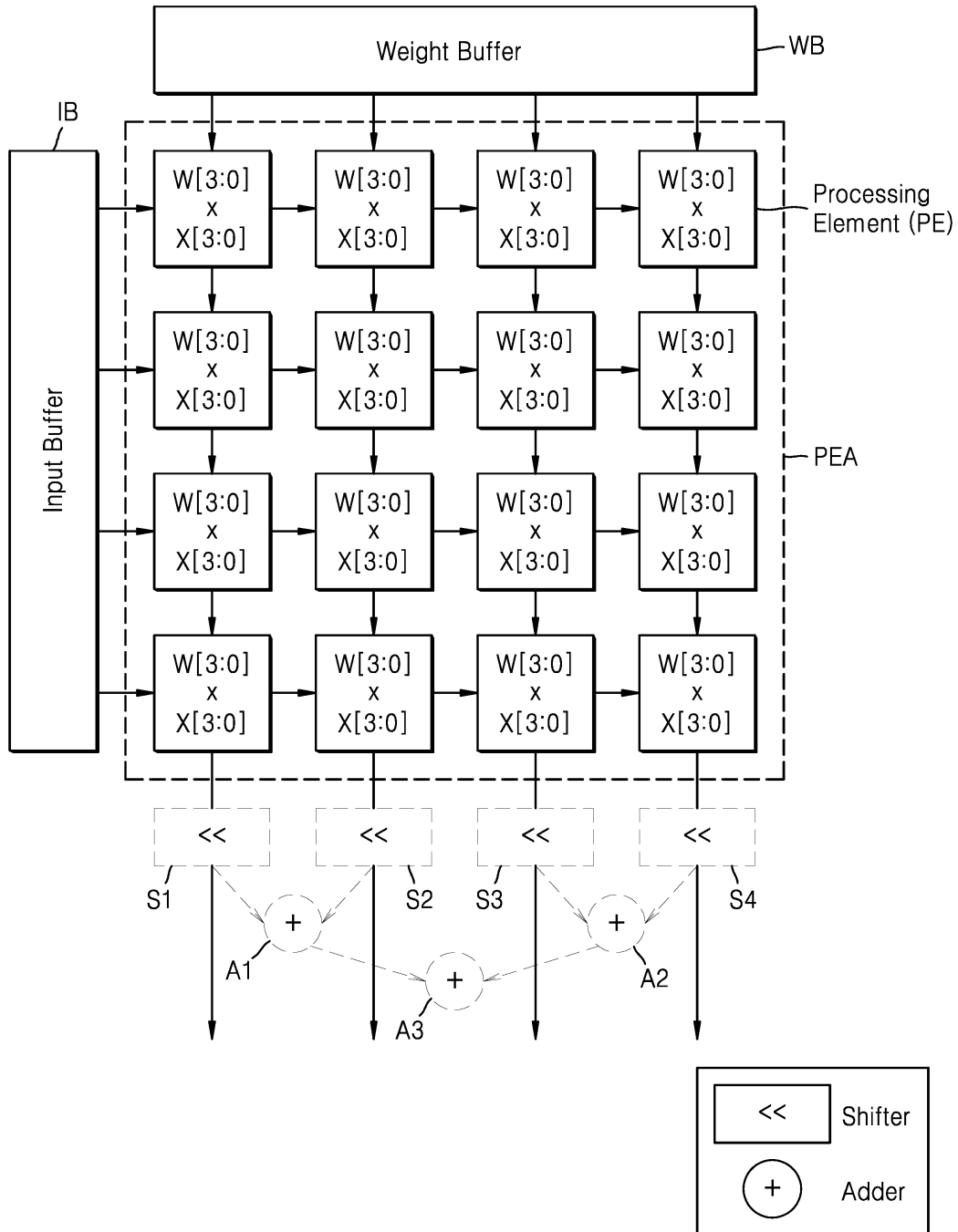


도면15

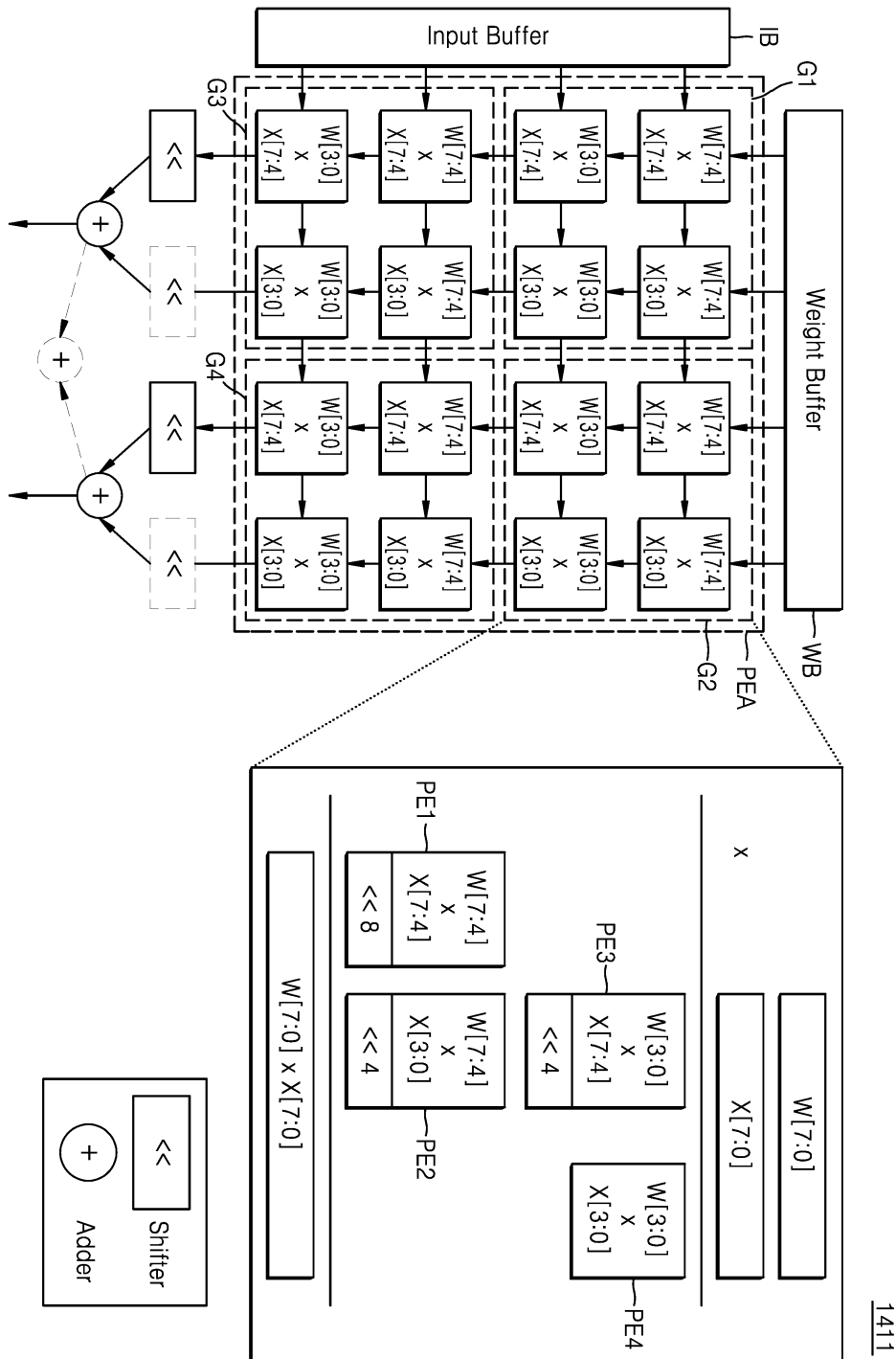


도면16

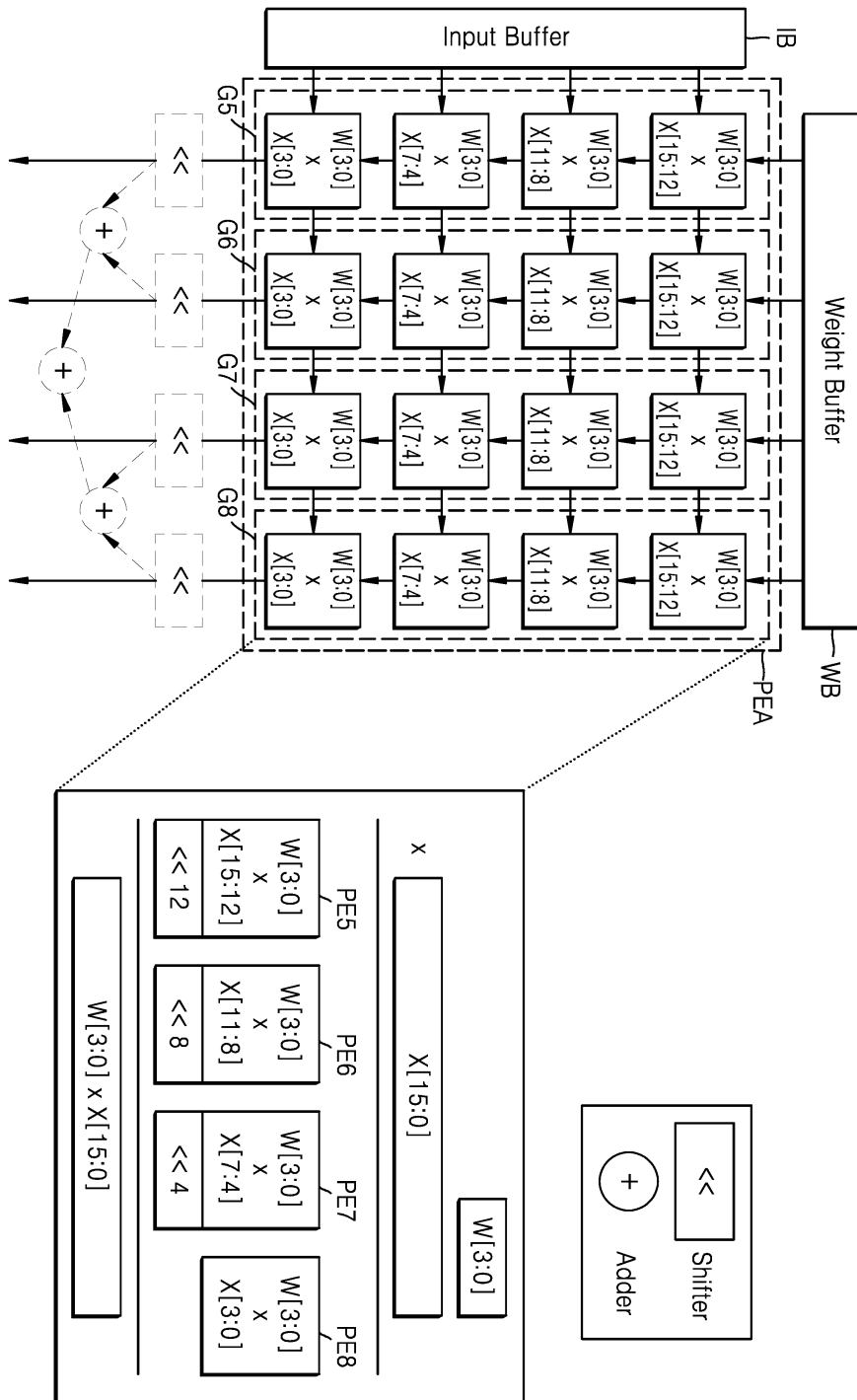
1411



도면17

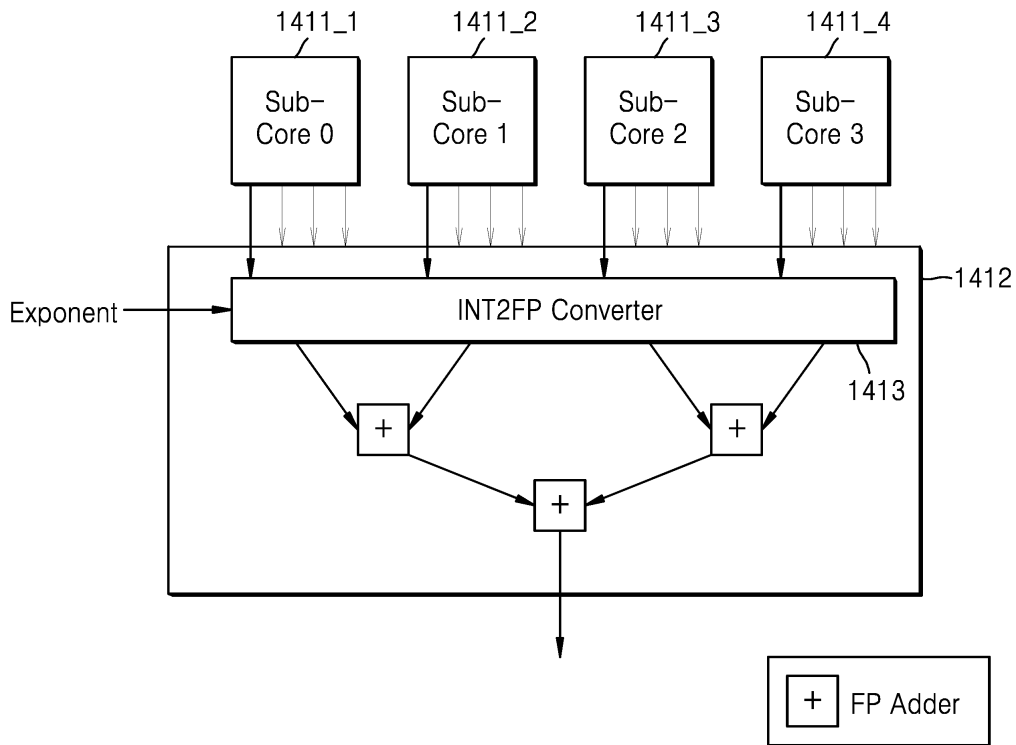


도면18

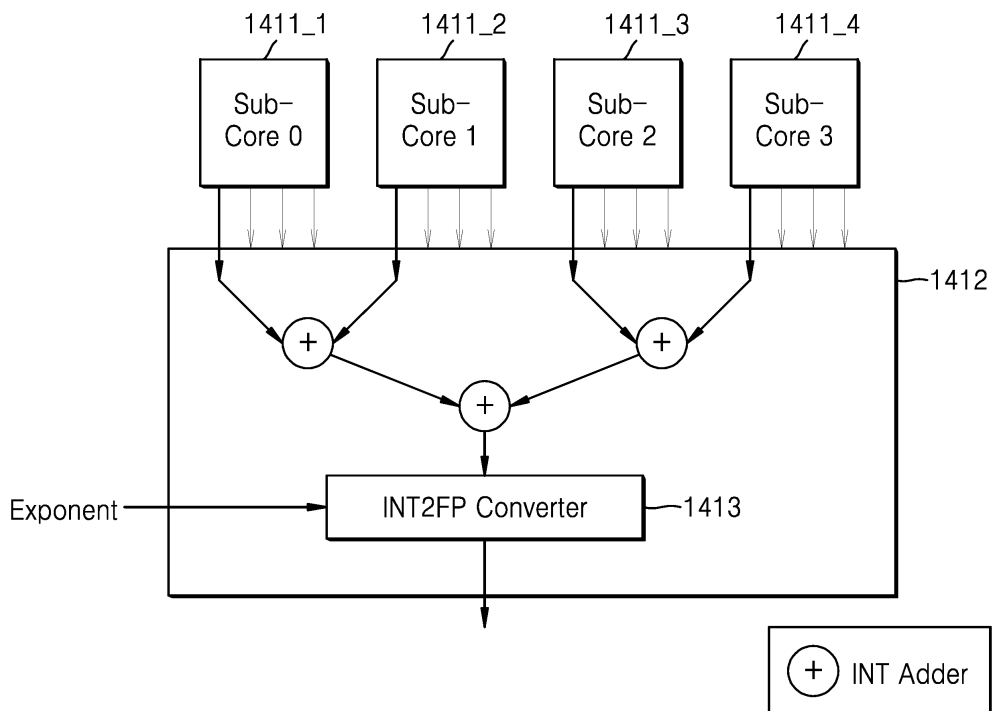


1411

도면19a



도면19b



도면20

