(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0133545
(43) 공개일자 2021년11월08일

(51) 국제특허분류(Int. Cl.)
G06N 3/02 (2019.01) *G06F 7/46* (2006.01)
G06N 5/04 (2006.01)
(52) CPC특허분류
G06N 3/02 (2019.01)
G06F 7/462 (2013.01)
(21) 출원번호 10-2020-0052268
(22) 출원일자 2020년04월29일
심사청구일자 2020년04월29일

(71) 출원인
한국항공대학교산학협력단
경기도 고양시 덕양구 항공대학로 76 (화전동, 한국항공대학교)
(72) 발명자
김지호
경기도 고양시 덕양구 향기로 77 (향동동, DMC호반베르디움 더포레 4단지)
김태환
경기도 고양시 덕양구 항공대학로 76 (화전동, 한국항공대학교)
(74) 대리인
안병규

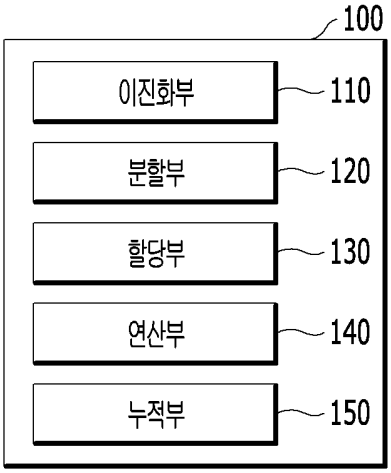
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 **신경망 모델의 추론 속도 향상 장치 및 방법**

(57) 요 약

신경망 모델의 추론 속도 향상 장치 및 방법이 개시되며, 본원의 일 실시예에 따른 신경망 모델의 추론 속도 향상 방법은, (a) 상기 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자 및 제2피연산자를 이진화하는 단계, (b) 상기 이진화된 제1피연산자를 복수의 제1비트군으로 분할하고, 상기 이진화된 제2피연산자를 복수의 제2비트군으로 분할하는 단계, (c) 상기 복수의 제1비트군의 값 및 상기 복수의 제2비트군의 값에 기초하여 상기 복수의 제1비트군 중 어느 하나와 상기 복수의 제2비트군 중 어느 하나 사이의 개별 곱연산을 위한 곱셈기를 선택적으로 할당하는 과정을 상기 복수의 제1비트군 각각과 상기 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 반복 수행하는 단계, (d) 상기 할당된 곱셈기에 기초하여 상기 개별 곱연산을 수행하는 단계 및 (e) 상기 개별 곱연산 결과에 시프트 연산을 수행하여 곱연산 결과를 누적하는 단계를 포함할 수 있다.

대 표 도 - 도1



(52) CPC특허분류
G06N 5/04 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 GRRC항공2017-B06
과제번호 GRRC항공2017-B06
부처명 경기도
과제관리(전문)기관명 경기도
연구사업명 2017년 경기도지역협력연구센터 (GRRC)사업
연구과제명 지능형 이동보조 수단을 위한 센서 데이터 처리 시스템 연구
기 여 율 1/1
과제수행기관명 한국항공대학교 산학협력단
연구기간 2019.07.01 ~ 2020.06.30

명 세 서

청구범위

청구항 1

신경망 모델의 추론 속도 향상 방법에 있어서,

(a) 상기 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자 및 제2피연산자를 이진화하는 단계;

(b) 상기 이진화된 제1피연산자를 복수의 제1비트군으로 분할하고, 상기 이진화된 제2피연산자를 복수의 제2비트군으로 분할하는 단계;

(c) 상기 복수의 제1비트군의 값 및 상기 복수의 제2비트군의 값에 기초하여 상기 복수의 제1비트군 중 어느 하나와 상기 복수의 제2비트군 중 어느 하나 사이의 개별 곱연산을 위한 곱셈기를 선택적으로 할당하는 과정을 상기 복수의 제1비트군 각각과 상기 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 반복 수행하는 단계;

(d) 상기 할당된 곱셈기에 기초하여 상기 개별 곱연산을 수행하는 단계; 및

(e) 상기 개별 곱연산 결과에 시프트 연산을 수행하여 곱연산 결과를 누적하는 단계,

를 포함하는, 추론 속도 향상 방법.

청구항 2

제1항에 있어서,

상기 (c) 단계는,

(c1) 상기 복수의 제1비트군 및 상기 복수의 제2비트군 중 값이 0인 비트군이 존재하는지 판정하는 단계; 및

(c2) 상기 값이 0인 비트군으로 판정된 비트군과 연계된 개별 곱연산에 대한 곱셈기는 미할당하는 단계,

를 포함하는 것인, 추론 속도 향상 방법.

청구항 3

제2항에 있어서,

상기 신경망 모델은 LSTM 신경망 모델이고,

상기 제1피연산자는 가중치이고 상기 제2피연산자는 단기 기억 메모리인 것인, 추론 속도 향상 방법.

청구항 4

제2항에 있어서,

상기 (a) 단계는,

상기 제1피연산자 및 상기 제2피연산자를 부호를 나타내는 비트 및 데이터를 나타내는 복수의 비트로 이진화하는 것인, 추론 속도 향상 방법.

청구항 5

제4항에 있어서,

상기 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 상기 복수의 제1비트군 및 상기 복수의 제2비트군이 각각 2개의 비트군이면,

상기 (b) 단계는,

상기 제1피연산자의 상기 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제1상위 비트군으로, 하위 (N-

A)개의 비트를 제1하위 비트군으로 분할하고,

상기 제2피연산자의 상기 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제2상위 비트군으로, 하위 (N-A)개의 비트를 제2하위 비트군으로 분할하는 것인, 추론 속도 향상 방법.

청구항 6

제5항에 있어서,

상기 (e) 단계는,

(e1) 상기 제1상위 비트군 및 상기 제2상위 비트군의 개별 곱연산 결과에 $2^{2(N-A)}$을 곱하는 제1시프트 연산을 수행하는 단계;

(e2) 상기 제1상위 비트군 및 상기 제2하위 비트군의 개별 곱연산 결과와 상기 제1하위 비트군 및 상기 제2상위 비트군의 개별 곱연산 결과에 $2^{(N-A)}$를 곱하는 제2시프트 연산을 수행하는 단계; 및

(e3) 상기 제1시프트 연산의 결과, 상기 제2시프트 연산의 결과 및 상기 제1하위 비트군 및 상기 제2하위 비트군의 개별 곱연산 결과를 합산하는 단계,

를 포함하는 것인, 추론 속도 향상 방법.

청구항 7

제1항에 있어서,

상기 (a) 내지 (e) 단계는, 엣지 디바이스에서 수행되는 것인, 추론 속도 향상 방법.

청구항 8

제1항에 있어서,

상기 (a) 단계의 수행에 앞서 상기 제1피연산자 및 상기 제2피연산자 중 Zero-data인 피연산자가 존재하는 것으로 판단되면, 상기 제1피연산자 및 상기 제2피연산자의 곱연산에 관하여 상기 (a) 단계 내지 상기 (e) 단계의 수행이 생략(skip)되는 것인, 추론 속도 향상 방법.

청구항 9

신경망 모델의 추론 속도 향상 장치에 있어서,

상기 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자 및 제2피연산자를 이진화하는 이진화부;

상기 이진화된 제1피연산자를 복수의 제1비트군으로 분할하고, 상기 이진화된 제2피연산자를 복수의 제2비트군으로 분할하는 분할부; 및

상기 복수의 제1비트군의 값 및 상기 복수의 제2비트군의 값에 기초하여 상기 복수의 제1비트군 중 어느 하나와 상기 복수의 제2비트군 중 어느 하나 사이의 개별 곱연산을 위한 곱셈기를 선택적으로 할당하는 과정을 상기 복수의 제1비트군 각각과 상기 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 반복 수행하는 할당부,

를 포함하는, 추론 속도 향상 장치.

청구항 10

제9항에 있어서,

상기 할당된 곱셈기에 기초하여 상기 개별 곱연산을 수행하는 연산부; 및

상기 개별 곱연산 결과에 시프트 연산을 수행하여 곱연산 결과를 누적하는 누적부,

를 더 포함하는, 추론 속도 향상 장치.

청구항 11

제10항에 있어서,

상기 할당부는,

상기 복수의 제1비트군 및 상기 복수의 제2비트군 중 값이 0인 비트군이 존재하는지 판정하고, 상기 값이 0인 비트군으로 판정된 비트군과 연계된 개별 곱연산에 대한 곱셈기는 미할당하는 것인, 추론 속도 향상 장치.

청구항 12

제11항에 있어서,

상기 신경망 모델은 LSTM 신경망 모델이고,

상기 제1피연산자는 가중치이고 상기 제2피연산자는 단기 기억 메모리인 것인, 추론 속도 향상 장치.

청구항 13

제11항에 있어서,

상기 이진화부는,

상기 제1피연산자 및 상기 제2피연산자를 부호를 나타내는 비트 및 데이터를 나타내는 복수의 비트로 이진화하는 것인, 추론 속도 향상 장치.

청구항 14

제13항에 있어서,

상기 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 상기 복수의 제1비트군 및 상기 복수의 제2비트군이 각각 2개의 비트군이면,

상기 분할부는,

상기 제1피연산자의 상기 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제1상위 비트군으로, 하위 (N-A)개의 비트를 제1하위 비트군으로 분할하고,

상기 제2피연산자의 상기 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제2상위 비트군으로, 하위 (N-A)개의 비트를 제2하위 비트군으로 분할하는 것인, 추론 속도 향상 장치.

청구항 15

제14항에 있어서,

상기 누적부는,

상기 제1상위 비트군 및 상기 제2상위 비트군의 개별 곱연산 결과에 $2^{2(N-A)}$을 곱하는 제1시프트 연산을 수행하고,

상기 제1상위 비트군 및 상기 제2하위 비트군의 개별 곱연산 결과와 상기 제1하위 비트군 및 상기 제2상위 비트군의 개별 곱연산 결과에 $2^{(N-A)}$를 곱하는 제2시프트 연산을 수행하고,

상기 제1시프트 연산의 결과, 상기 제2시프트 연산의 결과 및 상기 제1하위 비트군 및 상기 제2하위 비트군의 개별 곱연산 결과를 합산하는 것인, 추론 속도 향상 장치.

청구항 16

제1항 내지 제8항 중 어느 한 항의 방법을 컴퓨터에서 실행하기 위한 프로그램을 기록한 컴퓨터에서 판독 가능한 기록매체.

발명의 설명

기 술 분 야

[0001] 본원은 신경망 모델의 추론 속도 향상 장치 및 방법에 관한 것이다. 특히, 본원은 신경망 모델(예를 들면 LSTM 신경망 모델) 기반의 추론 과정에서 분할-정복 방식을 사용하여 곱셈 연산량을 줄이는 방법을 적용하여 신경망 모델의 추론 속도를 향상시키는 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 학습(Training)과 추론(Inference)으로 구분돼 있는 인공지능(AI) 영역에서 추론은 서비스 단과 직결된다. 다수의 데이터를 기반으로 연구개발이나 성능 고도화에 적용되는 학습과 달리 추론은 빠르고, 유연하며 지연 없이 처리되어야 한다. 이러한 인공지능 영역에서의 추론 속도는 최종 소비자와 연결되는 서비스 단에서 서비스 경쟁력을 좌우하는 요소라고 할 수 있다.

[0003] 일반적으로 신경망을 FFNets(Feed-Forward Neural Networks)라고 하는데, 이는 데이터를 트레이닝 셋(Set)과 테스트 셋으로 나누어 트레이닝 셋을 통해서 신경망의 가중치를 학습시키고, 이 결과를 테스트 셋을 통해서 확인하는 방식을 의미한다. 이러한 FFNets에서는 데이터를 입력하면 입력층에서 은닉층까지 연산이 차례로 진행되어 출력이 나오게 되며, 이 과정에서 입력 데이터는 모든 노드를 한 번씩만 지나가게 되는데, 이는 데이터의 순서를 무시하고 현재 주어진 데이터를 통해서 이전의 결과와 독립적으로 학습을 하는 구조를 갖는 것으로 이해될 수 있다.

[0004] 상술한 FFNets과 달리 RNN(Recurrent Neural Network)은 은닉층의 결과가 다시 같은 은닉층의 입력으로 들어가도록 연결되어, 은닉층의 결과가 다시 은닉층으로 들어가는 특징을 갖는 신경망 모델이다. 이러한 특성은 RNN이 일반적인 신경망 모델과는 다르게 시간적인 측면을 고려할 수 있도록 한다. 즉, RNN은 인간의 '기억력'의 개념을 주목하여, 인간이 과거의 대화 내용을 기억해서 현재의 대화 문맥을 이해하는 것처럼 이전에 들어왔던 데이터를 기억하고, 그 속에서 정보를 파악할 수 있는 것이다.

[0005] 다만, RNN은 가까운 과거의 결과만을 판단에 반영하기 때문에 입력이 길어지면 비교적 오래 전에 입력된 데이터가 정상적으로 반영되지 못하는 현상이 발생한다. 즉, RNN은 장기적인 데이터를 처리하는 데에 어려움이 있다. 이러한 문제점을 보안하기 위해 새로운 신경망 모델이 제안되었고, 대표적인 모델로는 LSTM(Long-Short Term Memory)이 있다. RNN의 경우에 다음 셀로 전달되는 정보는 직전 셀의 정보뿐이기 때문에 단기 기억만 가능한 반면, LSTM은 장기 기억 및 단기 기억이 모두 기억되는 구조를 갖는다. 따라서 입력의 길이가 길어져도 이전의 정보를 더 오래 기억하므로 비교적 긴 길이의 데이터를 처리하는데 있어서 RNN보다 성능이 우수하다.

[0006] 이러한 LSTM 신경망 모델의 추론 과정은 행렬과 벡터의 곱연산을 통해 이루어지는데, 이 때 곱연산되는 피연산자의 크기가 매우 크기 때문에 추론 과정에서 많은 양의 곱셈을 수행하게 되고 이로써 높은 하드웨어 복잡도를 가지게 된다.

[0007] 이를 해결하기 위하여 도입된 종래의 추론 가속 기법은 곱셈 연산량을 줄이기 위해 값이 0인 데이터(달리 말해, Zero-data)의 경우 곱연산을 생략하는 Zero-skipping 기법을 사용하였다. 다만, 이러한 Zero-skipping 기법에 의하면, 피연산자의 값이 0이면 곱연산을 건너뛸 수 있다고는 하나, 피연산자의 값 전체가 0인 경우에만 곱연산이 생략되어, 추론 속도의 향상이 제한적이라는 한계가 있었다.

[0008] 본원의 배경이 되는 기술은 한국공개특허공보 제10-2019-0066473호에 개시되어 있다.

발명의 내용

해결하려는 과제

[0009] 본원은 전술한 종래 기술의 문제점을 해결하기 위한 것으로서, 신경망 모델을 이용한 추론 과정에서 연산되는 피연산자를 분할-정복 형태로 연산을 수행함으로써 연산량을 감소시킬 수 있는 신경망 모델의 추론 속도 향상 장치 및 방법을 제공하려는 것을 목적으로 한다.

[0010] 다만, 본원의 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제들로 한정되지 않으며, 또 다른 기술적 과제들이 존재할 수 있다.

과제의 해결 수단

[0011] 상기한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본원의 일 실시예에 따른 신경망 모델의 추론 속도 향상 방법은, (a) 상기 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자 및 제2피연산자를 이진화하

는 단계, (b) 상기 이진화된 제1피연산자를 복수의 제1비트군으로 분할하고, 상기 이진화된 제2피연산자를 복수의 제2비트군으로 분할하는 단계, (c) 상기 복수의 제1비트군의 값 및 상기 복수의 제2비트군의 값에 기초하여 상기 복수의 제1비트군 중 어느 하나와 상기 복수의 제2비트군 중 어느 하나 사이의 개별 곱연산을 위한 곱셈기를 선택적으로 할당하는 과정을 상기 복수의 제1비트군 각각과 상기 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 반복 수행하는 단계, (d) 상기 할당된 곱셈기에 기초하여 상기 개별 곱연산을 수행하는 단계 및 (e) 상기 개별 곱연산 결과에 시프트 연산을 수행하여 곱연산 결과를 누적하는 단계를 포함할 수 있다.

[0012] 또한, 상기 (c) 단계는, (c1) 상기 복수의 제1비트군 및 상기 복수의 제2비트군 중 값이 0인 비트군이 존재하는지 판정하는 단계 및 (c2) 상기 값이 0인 비트군으로 판정된 비트군과 연계된 개별 곱연산에 대한 곱셈기는 미할당하는 단계를 포함할 수 있다.

[0013] 또한, 상기 신경망 모델은 LSTM 신경망 모델일 수 있다.

[0014] 또한, 상기 제1피연산자는 가중치이고 상기 제2피연산자는 단기 기억 메모리일 수 있다.

[0015] 또한, 상기 (a) 단계는, 상기 제1피연산자 및 상기 제2피연산자를 부호를 나타내는 비트 및 데이터를 나타내는 복수의 비트로 이진화할 수 있다.

[0016] 또한, 상기 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 상기 복수의 제1비트군 및 상기 복수의 제2비트군이 각각 2개의 비트군이면, 상기 (b) 단계는, 상기 제1피연산자의 상기 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제1상위 비트군으로, 하위 (N-A)개의 비트를 제1하위 비트군으로 분할할 수 있다.

[0017] 또한, 상기 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 상기 복수의 제1비트군 및 상기 복수의 제2비트군이 각각 2개의 비트군이면, 상기 (b) 단계는, 상기 제2피연산자의 상기 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제2상위 비트군으로, 하위 (N-A)개의 비트를 제2하위 비트군으로 분할할 수 있다.

[0018] 또한, 상기 (e) 단계는, (e1) 상기 제1상위 비트군 및 상기 제2상위 비트군의 개별 곱연산 결과에 $2^{2(N-A)}$을 곱하는 제1시프트 연산을 수행하는 단계, (e2) 상기 제1상위 비트군 및 상기 제2하위 비트군의 개별 곱연산 결과와 상기 제1하위 비트군 및 상기 제2상위 비트군의 개별 곱연산 결과에 $2^{(N-A)}$를 곱하는 제2시프트 연산을 수행하는 단계 및 (e3) 상기 제1시프트 연산의 결과, 상기 제2시프트 연산의 결과 및 상기 제1하위 비트군 및 상기 제2하위 비트군의 개별 곱연산 결과를 합산하는 단계를 포함할 수 있다.

[0019] 또한, 상기 (a) 내지 (e) 단계는, 엣지 디바이스에서 수행될 수 있다.

[0020] 또한, 상기 (a) 단계의 수행에 앞서 상기 제1피연산자 및 상기 제2피연산자 중 Zero-data인 피연산자가 존재하는 것으로 판단되면, 상기 제1피연산자 및 상기 제2피연산자의 곱연산에 관하여 상기 (a) 단계 내지 상기 (e) 단계의 수행이 생략(skip)될 수 있다.

[0021] 한편, 본원의 일 실시예에 따른 신경망 모델의 추론 속도 향상 장치는, 상기 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자 및 제2피연산자를 이진화하는 이진화부, 상기 이진화된 제1피연산자를 복수의 제1비트군으로 분할하고, 상기 이진화된 제2피연산자를 복수의 제2비트군으로 분할하는 분할부 및 상기 복수의 제1비트군의 값 및 상기 복수의 제2비트군의 값에 기초하여 상기 복수의 제1비트군 중 어느 하나와 상기 복수의 제2비트군 중 어느 하나 사이의 개별 곱연산을 위한 곱셈기를 선택적으로 할당하는 과정을 상기 복수의 제1비트군 각각과 상기 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 반복 수행하는 할당부를 포함할 수 있다.

[0022] 또한, 본원의 일 실시예에 따른 신경망 모델의 추론 속도 향상 장치는, 상기 할당된 곱셈기에 기초하여 상기 개별 곱연산을 수행하는 연산부 및 상기 개별 곱연산 결과에 시프트 연산을 수행하여 곱연산 결과를 누적하는 누적부를 포함할 수 있다.

[0023] 또한, 상기 할당부는, 상기 복수의 제1비트군 및 상기 복수의 제2비트군 중 값이 0인 비트군이 존재하는지 판정할 수 있다.

[0024] 또한, 상기 할당부는, 상기 값이 0인 비트군으로 판정된 비트군과 연계된 개별 곱연산에 대한 곱셈기는 미할당할 수 있다.

[0025] 또한, 상기 이진화부는, 상기 제1피연산자 및 상기 제2피연산자를 부호를 나타내는 비트 및 데이터를 나타내는

복수의 비트로 이진화할 수 있다.

[0026] 또한, 상기 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 상기 복수의 제1비트군 및 상기 복수의 제2비트군이 각각 2개의 비트군이면, 상기 분할부는, 상기 제1피연산자의 상기 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제1상위 비트군으로, 하위 (N-A)개의 비트를 제1하위 비트군으로 분할할 수 있다.

[0027] 또한, 상기 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 상기 복수의 제1비트군 및 상기 복수의 제2비트군이 각각 2개의 비트군이면, 상기 분할부는, 상기 제2피연산자의 상기 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제2상위 비트군으로, 하위 (N-A)개의 비트를 제2하위 비트군으로 분할할 수 있다.

[0028] 또한, 상기 누적부는, 상기 제1상위 비트군 및 상기 제2상위 비트군의 개별 곱연산 결과에 $2^{2(N-A)}$을 곱하는 제1시프트 연산을 수행할 수 있다.

[0029] 또한, 상기 누적부는, 상기 제1상위 비트군 및 상기 제2하위 비트군의 개별 곱연산 결과와 상기 제1하위 비트군 및 상기 제2상위 비트군의 개별 곱연산 결과에 $2^{(N-A)}$를 곱하는 제2시프트 연산을 수행할 수 있다.

[0030] 또한, 상기 누적부는, 상기 제1시프트 연산의 결과, 상기 제2시프트 연산의 결과 및 상기 제1하위 비트군 및 상기 제2하위 비트군의 개별 곱연산 결과를 합산할 수 있다.

[0031] 상술한 과제 해결 수단은 단지 예시적인 것으로서, 본원을 제한하려는 의도로 해석되지 않아야 한다. 상술한 예시적인 실시예 외에도, 도면 및 발명의 상세한 설명에 추가적인 실시예가 존재할 수 있다.

발명의 효과

[0032] 전술한 본원의 과제 해결 수단에 의하면, 신경망 모델을 이용한 추론 과정에서 연산되는 피연산자를 분할-정복 형태로 연산을 수행함으로써 연산량을 감소시킬 수 있는 신경망 모델의 추론 속도 향상 장치 및 방법을 제공할 수 있다.

[0033] 전술한 본원의 과제 해결 수단에 의하면, 종래의 Zero-skipping 기반의 추론 속도 향상 기법에 비하여 곱연산을 보다 많이 생략하여 연산량을 효과적으로 감소시켜 신경망 모델의 추론 속도를 비약적으로 향상시킬 수 있다.

[0034] 전술한 본원의 과제 해결 수단에 의하면, 메모리와 전력이 제한적인 엣지 디바이스에서도 신경망 모델을 이용한 전체 추론 과정이 진행되도록 하여 별도의 엣지 서버나 클라우드 서버로 데이터를 넘겨서 연산을 처리할 필요가 없어, 서비스 제공 주체의 입장에서는 서버 부하를 낮춰 운영비를 절감할 수 있도록 하고, 서비스 사용 주체의 입장에서는 사생활이 보장되고, 인터넷 연결 없이도 추론 서비스를 제공받을 수 있게 된다.

[0035] 다만, 본원에서 얻을 수 있는 효과는 상기된 바와 같은 효과들로 한정되지 않으며, 또 다른 효과들이 존재할 수 있다.

도면의 간단한 설명

[0036] 도 1은 본원의 일 실시예에 따른 신경망 모델의 추론 속도 향상 장치의 개략적인 구성도이다.

도 2는 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자 및 제2피연산자의 분포를 예시적으로 나타낸 그래프이다.

도 3은 이진화된 제1피연산자 및 이진화된 제2피연산자가 복수의 제1비트군 및 복수의 제2비트군으로 각각 분할되는 것을 설명하기 위한 개념도이다.

도 4는 제1피연산자인 가중치의 값을 가로축으로 하고 빈도를 세로축으로 하여 나타낸 가중치 분포도이다.

도 5는 개별 곱연산 결과에 시프트 연산을 수행하여 곱연산 결과를 누적하는 것을 설명하기 위한 개념도이다.

도 6은 본원의 일 실시예에 따른 신경망 추론 속도 향상 장치 또는 방법 적용시의 연산량을 어느 기법도 적용하지 않은 상태에서의 기존 연산량 및 종래의 Zero-skipping 기법 적용시의 연산량과 비교하여 나타낸 도표이다.

도 7은 본원의 일 실시예에 따른 신경망 모델의 추론 속도 향상 방법에 대한 동작 흐름도이다.

도 8은 복수의 제1비트군 각각과 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 선택적으로 곱셈기를 할당하는 과정에 대한 세부 동작 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0037] 아래에서는 첨부한 도면을 참조하여 본원이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본원의 실시예를 상세히 설명한다. 그러나 본원은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본원을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0038] 본원 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결" 또는 "간접적으로 연결"되어 있는 경우도 포함한다.

[0039] 본원 명세서 전체에서, 어떤 부재가 다른 부재 "상에", "상부에", "상단에", "하에", "하부에", "하단에" 위치하고 있다고 할 때, 이는 어떤 부재가 다른 부재에 접해 있는 경우뿐 아니라 두 부재 사이에 또 다른 부재가 존재하는 경우도 포함한다.

[0040] 본원 명세서 전체에서, 어떤 부분이 어떤 구성 요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성 요소를 제외하는 것이 아니라 다른 구성 요소를 더 포함할 수 있는 것을 의미한다.

[0041] 본원은 신경망 모델의 추론 속도 향상 장치 및 방법에 관한 것이다. 특히, 본원은 신경망 모델 기반의 추론 과정에서 곱셈 연산량을 줄이는 기법에 관한 것이다.

[0042] 도 1은 본원의 일 실시예에 따른 신경망 모델의 추론 속도 향상 장치의 개략적인 구성도이다.

[0043] 도 1을 참조하면, 본원의 일 실시예에 따른 신경망 모델의 추론 속도 향상 장치(100)(이하, '추론 속도 향상 장치(100)'라 한다.)는, 이진화부(110), 분할부(120), 할당부(130), 연산부(140) 및 누적부(150)를 포함할 수 있다.

[0044] 이하에서는, 먼저 본원에서 개시하는 추론 속도 향상 장치(100)를 적용할 수 있는 신경망 모델에 관해 먼저 설명하도록 한다. 본원의 실시예에 관한 설명에서, 신경망 모델은 RNN 모델, LSTM 신경망 모델, GRU 신경망 모델, Peephole-LSTM 등을 포함할 수 있으나, 이에만 한정되는 것은 아니고, 종래에 이미 공지되었거나 향후 개발되는 다양한 인공지능 모델을 포함할 수 있다.

[0045] 먼저, RNN(Recurrent Neural Network, 순환 신경망)은 은닉층의 결과가 다시 같은 은닉층의 입력으로 들어가도록 연결되어 있으며, RNN이라는 명칭에서 알 수 있듯이 은닉층의 결과가 다시 은닉층으로 들어가는 특징을 가진 신경망 모델이다. 이러한 특성은 RNN이 이전의 일반적인 신경망과는 다르게 시간적인 측면을 고려할 수 있는 모델이 될 수 있게 한다. 또한 RNN에서 주목하는 개념은 인간의 능력 중 '기억력'이다. 인간은 과거의 대화 내용을 기억해서 현재의 대화 문맥을 이해하는 것처럼 RNN은 이전에 들어왔던 데이터를 기억하고, 그 속에서 정보를 파악하도록 동작할 수 있다.

[0046] RNN에서 은닉층의 뉴런에 자기 자신을 가리키는 화살표를 순환 가중치라고 부른다. 순환 가중치는 순환과정에서 현재 학습에 반영하는 개념으로, 과거의 데이터 정보를 기억하고, 이를 통해 새로운 데이터를 처리할 때 과거의 기억을 사용한다.

[0047] 이해를 돕기 위해 순환 신경망을 사용하는 대표적인 예시인 '다음 단어 예측'을 생각해보면, "The tree is green."이라는 문장과 관련하여, "The tree is"라는 문장 뒤에 올 단어가 "green"일 확률을 예측하려는 경우, 필요한 정보를 가진 데이터와 입력하려는 위치가 가까워서 RNN으로 예측을 하는 데에 문제가 없다. 그러나, "I grew up in France. I was born there. So I speak fluent French"라는 문장과 관련하여, "I grew up in France. I was born there. So I speak fluent"뒤에 올 단어를 예측하려는 경우, 참고해야 할 정보(France)와 입력의 위치 차이가 크기 때문에 두 정보의 문맥을 연결하기 힘들어지고 이에 따라 성능이 저하되며, 이러한 성능 저하 문제를 'Vanishing Gradient Problem'라 일반적으로 지칭한다.

[0048] 이러한 문제점을 보안하기 위해 제시된 대표적인 모델은 LSTM(Long-Short Term Memory) 신경망 모델이 있다. 나아가, LSTM에서 변형된 구조를 가진 신경망 모델 또한 활발하게 제시되었는데, 대표적으로 GRU(Gated Recurrent Unit), LSTM-Peephole connections 등이 있다.

[0049] RNN의 경우에 다음 셀로 전달되는 정보는 직전 셀의 정보뿐이기 때문에 단기 기억만이 가능한 반면, LSTM은 그 명칭에서 볼 수 있듯이 장기 기억, 단기 기억이 모두 기억되는 구조로 입력의 길이가 길어져도 이전의 정보를 더 오래 기억할 수 있어 비교적 긴 길이의 데이터를 처리하는 데에 RNN보다 성능이 우수하다. LSTM은 3개의 게

이트가 있는 셀(Cell)로 이루어 지며 각 연산 단위를 통해 셀의 정보를 저장하거나, 이전의 정보를 불러올 수 있는 기능이 있다. 구체적으로, LSTM 신경망 모델은 장기 기억을 위한 데이터인 Cell State와 단기 기억을 위한 데이터(단기 기억 메모리)인 Hidden State를 사용하며 각 게이트의 출력값은 언제 신호를 불러올지, 내보낼지, 유지할지 등과 연관된다. 입력 게이트(Input Gate)는 새로 들어온 데이터의 입력이 셀로 유입되는 범위를 제어하고, 망각 게이트(Forget Gate)는 셀에 과거의 정보가 유입되는 범위를 제어하며 출력 게이트(Output Gate)는 어떤 출력값을 출력할지 결정한다. 이처럼 LSTM은 입력의 값을 무조건 State에 반영하지 않고, 정보의 전달량을 조절하는 게이트를 구비함으로써 장거리 의존성 반영이 가능하도록 개선된 RNN의 구조를 갖는다.

[0050] 또한, LSTM은 비교적 연산에 필요한 수식이 복잡하기 때문에 보다 단순화된 변형 구조가 제시되었는데, 일예로 GRU는 LSTM의 수식과 변수가 단순화된 구조로, 망각 게이트 와 입력 게이트를 하나의 갱신 게이트(Update Gate)로 통합하여, 장기 기억과 단기 기억을 합쳐 사용하는 구조를 갖는다. 다른 예로, Peephole connections는 각 게이트가 장기 기억 및 단기 기억을 모두 입력으로 사용하는 구조를 갖는다.

[0051] 이하에서는, 설명의 편의를 위하여 추론 속도 향상 장치(100)가 LSTM 신경망 모델에 적용되는 실시예를 중심으로 설명하나, 상술한 바와 같이 본원이 적용되는 신경망 모델의 유형은 LSTM 신경망 모델로 제한되는 것은 아니다.

[0052] 먼저, 이진화부(110)는, 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자 및 제2피연산자를 이진화할 수 있다. 본원의 일 실시예에 따르면, 제1피연산자는 가중치일 수 있다. 또한, 제2피연산자는 단기 기억 메모리(달리 말해, Hidden State)일 수 있다. 다만, 이에만 한정되는 것은 아니며, 본원이 적용되는 신경망 모델의 피연산자(예를 들면, 제1피연산자 또는 제2피연산자)는 상술한 가중치 또는 단기 기억 메모리로 제한되는 것은 아니고, 본원에서의 피연산자는 신경망 모델을 이용한 추론 과정에서 상호 연산될 수 있는 다양한 유형의 변수들을 폭넓게 포괄하는 개념일 수 있다. 참고로, 본원의 구현예에 따라 단기 기억 메모리는 활성 데이터 등으로 달리 지칭될 수 있으며, 가중치는 파라미터 등으로 달리 지칭될 수 있다.

[0053] 구체적으로, 이진화부(110)는, 제1피연산자 및 제2피연산자를 부호를 나타내는 비트 및 데이터를 나타내는 복수의 비트로 이진화할 수 있다. 참고로, 이진화부(110)가 제1피연산자 및 제2피연산자를 부호를 나타내는 단일 비트(1 bit)를 최상위 비트로 하고, 데이터를 나타내는 복수의 비트를 상기 최상위 비트의 하위에 오도록 이진화하는 방식을 부호화된 크기(Signed Magnitude) 방식의 인코딩으로 지칭할 수 있다.

[0054] 또한, 전술한 바와 같이 LSTM 신경망 모델의 추론 과정은 행렬과 벡터의 곱연산을 통해 이루어지며, 이와 관련하여 이진화부(110)는 행렬 또는 벡터 형태로 이루어진 제1피연산자 및 제2피연산자를 이진화하는 것으로 이해될 수 있다.

[0055] 도 2는 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자 및 제2피연산자의 분포를 예시적으로 나타낸 그래프이다. 특히, 도 2의 (a)는 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자인 가중치(Weight)의 분포를 나타낸 것이고, 도 2의 (b)는 신경망 모델을 이용한 추론 과정에서 곱연산되는 제2피연산자인 단기 기억 메모리(Hidden State)의 분포를 나타낸 것일 수 있다. 참고로, 도 2에 도시된 제1피연산자와 제2피연산자는 본원의 일 실시예와 연계된 일 실험예에서 활용된 영화 감상평에 대한 감정 분석과 연계된 IMDB 데이터를 학습시킨 LSTM 신경망 모델에서의 가중치와 단기 기억 메모리일 수 있다.

[0056] 도 2를 참조하면, 제1피연산자와 제2피연산자는 대부분이 0근처에 주로 분포하는 것을 확인할 수 있다. 달리 말해, 본원에서의 제1피연산자 및 제2피연산자를 각각이 0또는 1의 값을 갖는 복수의 비트로 이진화하면, 제1피연산자와 제2피연산자에서의 상위 비트 부분은 0에 해당하는 빈도가 높을 것으로 합리적으로 예측할 수 있다.

[0057] 이와 관련하여, 본원에서 개시하는 추론 속도 향상 장치(100)는 제1피연산자 및 제2피연산자를 이진화하여 복수의 비트군으로 분할하는 분할-정복 기법을 도입하여, 분할된 비트군 중 값이 0인 비트군에 대하여는 곱연산을 생략하도록 구현하고 피연산자의 전체 값이 0이 아닌 경우에도 곱연산을 생략하도록 하여 연산량을 획기적으로 감소시키고자 한 것이다.

[0058] 분할부(120)는 이진화된 제1피연산자를 복수의 제1비트군으로 분할할 수 있다. 또한, 분할부(120)는 이진화된 제2피연산자를 복수의 제2비트군으로 분할할 수 있다.

[0059] 본원의 일 실시예에 따르면, 분할부(120)는 제1피연산자를 2개의 비트군인 제1상위 비트군 및 제1하위 비트군으로 분할할 수 있다. 마찬가지로, 분할부(120)는 제2피연산자를 2개의 비트군인 제2상위 비트군 및 제2하위 비트군으로 분할할 수 있다. 다만, 이에만 한정되는 것은 아니며, 분할부(120)는 필요에 따라 비트군을 분할하지 않거나 2개 이상의 비트군으로 피연산자를 분할할 수 있다. 이와 관련하여, 본원의 일 실시예에 따르면, 추론 속

도 향상 장치(100)는 제1피연산자 및 제2피연산자를 이진화하기에 앞서 제1피연산자 및 제2피연산자 중 Zero-data인 피연산자가 존재하는 것으로 판단되면, 본원에서 개시하는 분할-정복 방식의 추론 속도 향상 기법을 미적용할 수 있다. 즉, 추론 속도 향상 장치(100)는 제1피연산자 및 제2피연산자 중 Zero-data인 피연산자가 존재하는 것으로 판단되면, 제1피연산자와 제2피연산자에 대한 이진화, 비트군 분할, 곱셈기 할당 등의 프로세스를 별도로 수행하지 않고 곧바로 제1피연산자와 제2피연산자에 대한 곱연산 결과가 0이 되도록 처리(달리 말해, Zero-data로 처리)할 수 있다.

[0060] 본원의 일 실시예에 따르면, 분할부(120)는 제1피연산자 및 제2피연산자의 분포 정보에 기초하여 제1피연산자 및 제2피연산자가 분할되는 비트군의 수를 결정하도록 동작할 수 있다. 여기서, 비트군의 수를 결정하는 기준이 되는 피연산자의 분포 정보는 예시적으로 도 2를 참조하여 이해될 수 있는 피연산자의 분포도에서의 분포 형상 정보를 포함할 수 있다. 본원의 일 실시예에 따르면, 도 2와 같이 제1피연산자와 제2피연산자의 분포 형상이 상이한 경우, 제1피연산자에 대한 제1비트군의 분할 수와 제2피연산자에 대한 제2비트군의 분할 수가 상이할 수 있으나, 이에만 한정되는 것은 아니다.

[0061] 구체적으로, 이진화된 제1피연산자 및 이진화된 제2피연산자의 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 복수의 제1비트군 및 복수의 제2비트군이 각각 2개의 비트군이면(달리 말해, 2개의 비트군으로 피연산자를 분할하는 경우), 분할부(120)는 제1피연산자의 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제1상위 비트군으로, 하위 (N-A)개의 비트를 제1하위 비트군으로 분할할 수 있다. 마찬가지로, 분할부(120)는 제2피연산자의 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제2상위 비트군으로, 하위 (N-A)개의 비트를 제2하위 비트군으로 분할할 수 있다.

[0062] 이해를 돕기 위해 예시하면, 이진화된 제1피연산자와 이진화된 제2피연산자가 각각 16개의 데이터를 나타내는 비트를 포함(달리 말해, N=16)하고, 분할부(120)가 2개의 비트군으로 피연산자를 분할하는 경우, 이진화된 제1피연산자의 상위 8개의 비트를 제1상위 비트군으로, 하위 8개의 비트를 제1하위 비트군으로 분할할 수 있다(A=8). 마찬가지로, 분할부(120)는 이진화된 제2피연산자의 상위 8개의 비트를 제2상위 비트군으로, 하위 8개의 비트를 제2하위 비트군으로 분할할 수 있다.

[0063] 본원의 일 실시예에 따르면, 분할부(120)는 제1피연산자 및 제2피연산자의 분포 정보에 기초하여 제1피연산자에 대한 복수의 제1비트군 각각에 포함된 비트의 수 및 제2피연산자에 대한 복수의 제2비트군 각각에 포함된 비트의 수를 결정하도록 동작할 수 있다. 달리 말해, 분할부(120)는 제1피연산자 및 제2피연산자의 분포 정보에 기초하여 상기 N의 값 및/또는 상기 A의 값을 결정하도록 동작할 수 있다.

[0064] 도 3은 이진화된 제1피연산자 및 이진화된 제2피연산자가 복수의 제1비트군 및 복수의 제2비트군으로 각각 분할되는 것을 설명하기 위한 개념도이다.

[0065] 도 3을 참조하면, 분할부(120)가 2개의 비트군으로 피연산자를 분할하는 경우, 제1피연산자(A)에 대한 제1상위 비트군을 a_H로, 제1하위 비트군을 a_L로, 제2피연산자(B)에 대한 제2상위 비트군을 b_H로, 제2하위 비트군을 b_L로 각각 지칭할 수 있다. 달리 말해, 제1피연산자는 $a_H*2^{(N-A)}+a_L$로, 제2피연산자는 $b_H*2^{(N-A)}+b_L$로 각각 표현될 수 있다.

[0066] 도 4는 제1피연산자인 가중치의 값을 가로축으로 하고 빈도를 세로축으로 하여 나타낸 가중치 분포도이다.

[0067] 도 4를 참조하면, 제1피연산자인 가중치의 값과 그 빈도를 나타낸 가중치 분포도의 가로축을 기준으로 하여, 구간 B와 구간 D에서는 a_H가 0이되, a_L은 0이 아닐 수 있다. 또한, 구간 C(달리 말해, 원점 C)에서는 a_H와 a_L 모두 0일 수 있다. 또한, 구간 A와 구간 E에서는 a_H가 0이 아닐 수 있다. 즉, 제1피연산자의 분포도는 $a_H=0$ & $a_L=0$을 만족하는 구간, $a_H=0$ & $a_L\neq0$을 만족하는 구간, $a_H\neq0$ & $a_L=0$을 만족하는 구간 및 $a_H\neq0$ & $a_L\neq0$을 만족하는 구간의 4구간으로 분류될 수 있다. 이 때 분류된 4구간 각각의 빈도(세로축)를 살펴보면, $a_H=0$ & $a_L\neq0$을 만족하는 구간(예를 들면, 도 3의 구간 B 및 구간 D)에 속하는 제1피연산자가 가장 많은 것을 확인할 수 있고, 이에 착안하여 상술한 Zero-Skipping 기법처럼 전체 비트가 0일 때 뿐만 아니라, 분할된 비트군 중 어느 하나가 0일 때도 제2피연산자와의 곱연산에서의 연산량을 획기적으로 줄일 수 있음을 추론할 수 있다.

[0068] 즉, a_H=0 & a_L≠0을 만족하는 구간에 속하는 제1피연산자(가중치)와 제2피연산자(단기 기억 메모리)의 곱연산에서는 a_H가 0이므로, a_H는 제2피연산자와 곱해지더라도 0이 되어 a_H에 대한 곱연산이 생략되더라도 전체 곱연산 결과에 실질적인 영향을 미치지 않을 수 있다.

[0069] 이와 관련하여, 할당부(130)는 복수의 제1비트군의 값 및 복수의 제2비트군의 값에 기초하여 복수의 제1비트군 중 어느 하나와 복수의 제2비트군 중 어느 하나 사이의 개별 곱연산을 위한 곱셈기를 선택적으로 할당하는 과정을 복수의 제1비트군 각각과 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 반복 수행할 수 있다.

[0070] 구체적으로, 할당부(130)는 복수의 제1비트군 및 복수의 제2비트군 중 값이 0인 비트군이 존재하는지 판정할 수 있다. 달리 말해, 할당부(130)는 분할된 비트군으로부터 0인 비트군을 검출하는 로직을 보유할 수 있다.

[0071] 또한, 할당부(130)는 값이 0인 비트군으로 판정된 비트군과 연계된 개별 곱연산에 대한 곱셈기는 미할당하도록 동작할 수 있다. 이와 관련하여, 예시적으로 제1피연산자 및 제2피연산자를 각각 M개의 비트군으로 분할한 경우, M개의 제1비트군 및 M개의 제2비트군이 모두 0이 아닌 경우에는 M^2 번의 개별 곱연산이 수행되어야 하는 반면, 제1비트군과 제2비트군 중 하나라도 0인 비트군이 존재하면, 해당 비트군에 대한 곱셈기가 미할당됨으로써 곱셈기의 개수 및 연산 횟수가 감소할 수 있어, 추론 과정에 필요한 하드웨어 복잡도를 크게 낮출 수 있다.

[0072] 본원의 일 실시예에 따르면, 제1피연산자 및 제2피연산자가 부호를 나타내는 최상위 비트 및 데이터를 포함한 8개의 비트를 포함하여 9개의 비트를 포함하고, 2개의 비트군으로 각각 분할되는 경우, 제1상위 비트군, 제1하위 비트군, 제2상위 비트군 및 제2하위 비트군은 각각 4비트를 포함하고, 제1피연산자와 제2피연산자가 0에 가까운 값을 가질수록 제1상위 비트군 및 제2상위 비트군은 0이 되고, 제1하위 비트군 및 제2하위 비트군에 0이 아닌 값이 존재할 것이다. 이 때, 최상위 비트인 부호 비트를 제외하고 분할된 비트군 간의 개별 곱연산을 수행하면, 피연산자 간의 전체 곱연산은 기존 8비트 간의 곱연산에서, 4비트 간의 곱연산으로 축소될 뿐만 아니라, 분할된 비트군이 0이면, 해당 비트군과 연계된 곱셈기의 할당이 생략되어 연산량이 획기적으로 감소할 수 있는 것이다.

[0073] 연산부(140)는, 상술한 과정을 통해 할당부(130)에 의해 할당된 곱셈기에 기초하여 개별 곱연산을 수행할 수 있다. 달리 말해, 연산부(140)는 곱셈기가 제1피연산자에 대한 복수의 비트군 중 0이 아닌 비트군과 제2피연산자에 대한 복수의 비트군 중 0이 아닌 비트군에 대하여 할당된 곱셈기에 기초하여 각각의 개별 곱연산을 수행할 수 있다.

[0074] 누적부(150)는, 개별 곱연산 결과에 시프트 연산을 수행하여 곱연산 결과를 누적할 수 있다.

[0075] 구체적으로, 누적부(150)는 제1상위 비트군 및 제2상위 비트군의 개별 곱연산 결과에 제1시프트 연산을 수행할 수 있다. 여기서, 제1시프트 연산은 이진화된 제1피연산자와 이진화된 제2피연산자의 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 상위 비트군이 각각 A개의 비트를, 하위 비트군이 각각 (N-A)개의 비트를 포함하는 경우 제1상위 비트군 및 제2상위 비트군의 개별 곱연산 결과에 $2^{2(N-A)}$을 곱해 2(N-A) 비트만큼의 시프트를 수행하는 것일 수 있다.

[0076] 또한, 누적부(150)는 제1상위 비트군 및 제2하위 비트군의 개별 곱연산 결과와 제1하위 비트군 및 제2상위 비트군의 개별 곱연산 결과에 제2시프트 연산을 수행할 수 있다. 여기서, 제2시프트 연산은 이진화된 제1피연산자와 이진화된 제2피연산자의 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 상위 비트군이 각각 A개의 비트를, 하위 비트군이 각각 (N-A)개의 비트를 포함하는 경우 제1상위 비트군 및 제2하위 비트군의 개별 곱연산 결과와 제1하위 비트군 및 제2상위 비트군의 개별 곱연산 결과에 $2^{(N-A)}$를 곱해 (N-A) 비트만큼의 시프트를 수행하는 것일 수 있다.

[0077] 또한, 누적부(150)는 상기의 제1시프트 연산의 결과, 상기의 제2시프트 연산의 결과 및 제1하위 비트군 및 제2하위 비트군의 개별 곱연산 결과를 합산하여 최종적인 곱연산 결과를 얻을 수 있다.

[0078] 도 5는 개별 곱연산 결과에 시프트 연산을 수행하여 곱연산 결과를 누적하는 것을 설명하기 위한 개념도이다.

[0079] 특히, 도 5의 (a)는 N개의 데이터를 나타내는 비트를 포함하는 제1피연산자 및 제2피연산자를 A개의 비트를 포함하는 상위 비트군 및 (N-A)개의 비트를 포함하는 하위 비트군으로 분할한 경우의 전체 곱연산 결과를 나타낸 것이고, 도 5의 (b)는 예시적으로 N=16, A=8인 경우의 전체 곱연산 결과를 나타낸 것이다.

[0080] 구체적으로 도 5의 (a)를 참조하면, 제1피연산자(A)와 제2피연산자(B)의 곱연산 결과(A X B)는, 제1상위 비트군

및 제2상위 비트군의 개별 곱연산에 대한 제1시프트 연산의 결과인 $a_H b_H \times 2^{2(N-A)}$와 제1상위 비트군 및 제2하위 비트군의 개별 곱연산과 제1하위 비트군 및 제2상위 비트군의 개별 곱연산에 대한 제2시프트 연산의 결과인 $(a_H b_L + a_L b_H) \times 2^{(N-A)}$와 제1하위 비트군 및 제2하위 비트군의 개별 곱연산 결과인 $a_L b_L$를 합산한 것일 수 있다.

[0081] 종합하면, 누적부(150)는 연산부(140)의 개별 곱연산 결과에 해당 개별 곱연산에서의 피연산자인 제1비트군의 전체 제1피연산자에서의 상대적 위치(자리) 및 피연산자인 제2비트군의 전체 제2피연산자에서의 상대적 위치(자리)에 기초하여 시프트 연산을 적용할 수 있다.

[0082] 예시적으로, 누적부(150)는 마지막 비트가 이진화된 제1피연산자의 하위 (X+1)번째 비트인 제1비트군과 마지막 비트가 이진화된 제2피연산자의 하위 (Y+1)번째 비트인 제2비트군의 개별 곱연산(여기서, 제1비트군 및 제2비트군은 모두 값이 0이 아니어서 곱셈기가 할당된 것을 가정한다.) 결과에 $2^{(X+Y)}$를 곱하는 시프트 연산을 수행할 수 있다.

[0083] 도 6은 본원의 일 실시예에 따른 신경망 추론 속도 향상 장치 또는 방법 적용시의 연산량을 어느 기법도 적용하지 않은 상태에서의 기존 연산량 및 종래의 Zero-skipping 기법 적용시의 연산량과 비교하여 나타낸 도표이다.

[0084] 도 6을 참조하면, 종래의 Zero-skipping 기법은 제1피연산자 및 제2피연산자 중 적어도 하나가 0인 경우에만 곱셈을 생략하기 때문에 피연산자가 zero-data일 경우에만 연산량을 감소시킬 수 있다. 이에 반해, 본원의 추론 속도 향상 장치(100)를 사용하면, 제1피연산자 및 제2피연산자 중 적어도 하나가 0인 경우뿐만 아니라, 제1피연산자를 분할한 복수의 제1비트군 및 제2피연산자를 분할한 복수의 제2비트군 중 0인 비트군에 대해서도 곱셈연산을 생략(skip)하여 선택적으로 연산을 수행할 수 있기 때문에 기존의 zero skipping 보다 더 좋은 효과를 얻을 수 있다.

[0085] 종래의 Zero-Skipping 기법은 제1피연산자 및 제2피연산자 각각을 복수의 비트군으로 분할하여, 복수의 비트군 중 0인 비트군에 대한 곱셈기를 미할당하는 프로세스를 포함하지 않아, 분할된 복수의 비트군 각각의 값과 무관하게 동등한 연산 횟수를 필요로 하는 반면(연산량 4), 본원의 추론 속도 향상 장치(100)를사용할 경우, 복수의 제1비트군 및 복수의 제2비트군이 모두 0이 아닌 경우(도6도표의 최하단 행)를 제외하면, 연산 횟수를 적어도 절반 이상(50%이상) 감소시킬 수 있는 것을 도6을 통해 확인할 수 있다(본원의 방법 적용 연산량 = 0,1,2 또는 4).

[0086] 보다 구체적으로, 본원의 일 실시예에 따른 추론 속도 향상 장치(100)와 연계된 일 실험예로서, IMDB 데이터셋을 활용하여 영화 감상평의 감정 분석을 수행하는 LSTM 신경망 모델에 대한 연산량을 도6 및 피연산자의 구간에 따른 빈도인 도2를 활용하여 계산 및 평가하였다. 해당 실험 결과를 종합하면 종래의 Zero-Skipping 기법의 경우 어느 기법도 적용하지 않은 상태에서의 기존 연산량 대비 5.7%의 연산량 감소 효과를 보인 반면, 본원에서 개시하는 분할-정복 기법에 기초하여 제1피연산자인 가중치와 제2피연산자인 단기 기억 메모리를 모두 8비트로 양자화(Quantization)하여 곱연산을 수행하였을 때, 52%의 연산량 감소 효과를 보였다. 이처럼, 본원에서 개시하는 추론 속도 향상 기법에 의하면, 8비트 양자화(Quantization)에 의한 정확도(Accuracy) 감소는 극히 적은(0.01%) 반면, 연산량에 있어서는 기존 Zero-Skipping 기법 대비 46.3%p의 획기적인 연산량 감소 효과를 보이는 것을 확인하였다.

[0087] 또한, 종래의 문헌 [Arash Ardakani, Zhengyun Ji, Warren J. Gross - "Learning to Skip Ineffectual Recurrent Computations in LSTMs" 2019]에 개시된 추론 속도 향상 기법은 입력을 배치 사이즈로 나누어 입력으로 인가된(들어온) 전체 배치 사이즈가 0인 경우에만 해당 연산을 건너뛰도록 구현한바 있으나, 이러한 종래 기법에 의하면, 대부분 값이 0인 입력이 인가되는 경우라도, 중간에 0이 아닌 값의 입력이 인가되면, 실질적인 연산량 저하가 이루어지지 않는 반면, 본원에 의할 때 모든 개별 연산에서 피연산자를 복수의 비트군으로 분할하여 0인 비트군을 검출하고, 검출된 비트군에 대한 연산을 생략할 수 있으므로, 배치 사이즈와 무관하게 연산량을 감소시킬 수 있는 효과가 있다.

[0088] 또한, 본원의 일 실시예에 따르면, 추론 속도 향상 장치(100)는 소정의 신경망 모델이 탑재되어 탑재된 신경망 모델을 이용한 추론 프로세스가 수행되는 엣지 디바이스(미도시)에 구비되는 것일 수 있다. 달리 말해, 추론 속도 향상 장치(100)의 동작은 소정의 엣지 디바이스(미도시)에서 수행되는 것일 수 있다.

[0089] 이와 관련하여, 신경망 모델을 이용한 추론(Inference) 과정은 실제 사용자에게 제공되는 서비스 단과 직결되므

로, 다수의 데이터를 기반으로 연구개발이나 성능 고도화에 적용되는 학습(Training) 과정과 달리 빠르고, 유연하며, 지연성 없이 처리되어야 하며, 많은 양의 데이터를 가지고 실시간 시스템의 목적 달성을 이루기 위해서는 엣지 디바이스 위에서 전체 추론 과정을 진행하는 것은 엣지 디바이스의 메모리 및 전력의 제한성에 따라 어려운 일이었으나, 본원에서 개시하는 추론 속도 향상 장치(100)에 의하면, 음성 인식, 실시간 번역 등의 서비스와 연계된 추론 연산을 엣지 디바이스 내에 직접 이루어지도록 구현하여, 별도의 엣지 서버나 클라우드 서버로 데이터를 넘기고 연산을 처리한 후 결과만 받아오는 방식을 탈피할 수 있다.

[0090] 즉, 본원에 의할 때 신경망 모델을 이용한 인공지능 서비스를 제공하는 주체의 입장에서는 서버에 요구되는 부하를 크게 낮춰 운영비를 절감할 수 있으며, 서비스를 영위하는 사용자 입장에서는 사용자가 인가한 입력이 별도의 서버를 통해 공유되지 않기 때문에 사생활이 보장되고 어플리케이션을 실행할 때 인터넷 등의 별도의 네트워크 연결 없이도 소정의 추론 서비스를 제공받을 수 있게 될 수 있는 이점이 있다.

[0091] 이하에서는 상기에 자세히 설명된 내용을 기반으로, 본원의 동작 흐름을 간단히 살펴보기로 한다.

[0092] 도 7은 본원의 일 실시예에 따른 신경망 모델의 추론 속도 향상 방법에 대한 동작 흐름도이다.

[0093] 도 7에 도시된 신경망 모델의 추론 속도 향상 방법은 앞서 설명된 추론 속도 향상 장치(100)에 의하여 수행될 수 있다. 따라서, 이하 생략된 내용이라고 하더라도 추론 속도 향상 장치(100)에 대하여 설명된 내용은 신경망 모델의 추론 속도 향상 방법에 대한 설명에도 동일하게 적용될 수 있다.

[0094] 도 7을 참조하면, 단계 S11에서 이진화부(110)는, (a) 신경망 모델을 이용한 추론 과정에서 곱연산되는 제1피연산자 및 제2피연산자를 이진화할 수 있다.

[0095] 또한, 단계 S11에서 이진화부(110)는, 제1피연산자 및 제2피연산자를 부호를 나타내는 비트 및 데이터를 나타내는 복수의 비트로 이진화할 수 있다.

[0096] 다음으로, 단계 S12에서 분할부(120)는, (b) 이진화된 제1피연산자를 복수의 제1비트군으로 분할하고, 이진화된 제2피연산자를 복수의 제2비트군으로 분할할 수 있다.

[0097] 또한, 단계 S12에서 분할부(120)는, 데이터를 나타내는 복수의 비트가 N개의 비트를 포함하고, 복수의 제1비트군 및 복수의 제2비트군이 각각 2개의 비트군이면, 제1피연산자의 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제1상위 비트군으로, 하위 (N-A)개의 비트를 제1하위 비트군으로 분할하고, 제2피연산자의 데이터를 나타내는 복수의 비트의 상위 A개의 비트를 제2상위 비트군으로, 하위 (N-A)개의 비트를 제2하위 비트군으로 분할할 수 있다.

[0098] 다음으로, 단계 S13에서 할당부(130)는, (c) 복수의 제1비트군의 값 및 복수의 제2비트군의 값에 기초하여 복수의 제1비트군 중 어느 하나와 복수의 제2비트군 중 어느 하나 사이의 개별 곱연산을 위한 곱셈기를 선택적으로 할당할 수 있다.

[0099] 또한, 할당부(130)는 단계 S13의 곱셈기를 선택적으로 할당하는 과정을 복수의 제1비트군 각각과 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 반복 수행할 수 있다.

[0100] 다음으로, 단계 S14에서 연산부(140)는, (d) 할당된 곱셈기에 기초하여 상기 개별 곱연산을 수행할 수 있다.

[0101] 다음으로, 단계 S15에서 누적부(150)는, (e) 단계 S14(달리 말해, (d) 단계)의 개별 곱연산 결과에 시프트 연산을 수행하여 곱연산 결과를 누적할 수 있다.

[0102] 구체적으로, 단계 S15에서 누적부(150)는, (e1) 제1상위 비트군 및 제2상위 비트군의 개별 곱연산 결과에 $2^{2(N-A)}$을 곱하는 제1시프트 연산을 수행할 수 있다.

[0103] 또한, 단계 S15에서 누적부(150)는, (e2) 제1상위 비트군 및 제2하위 비트군의 개별 곱연산 결과와 제1하위 비트군 및 제2상위 비트군의 개별 곱연산 결과에 $2^{(N-A)}$를 곱하는 제2시프트 연산을 수행할 수 있다.

[0104] 또한, 단계 S15에서 누적부(150)는, (e3) 제1시프트 연산의 결과, 제2시프트 연산의 결과 및 제1하위 비트군 및 상기 제2하위 비트군의 개별 곱연산 결과를 합산할 수 있다.

[0105] 또한, 본원의 일 실시예에 따르면, 단계 S11 내지 단계 S15(달리 말해, (a) 내지 (e) 단계)는 엣지 디바이스에서 수행될 수 있다.

[0106] 또한, 본원의 일 실시예에 따르면, 단계 S11의 수행에 앞서(달리 말해, (a) 단계의 수행에 앞서) 제1피연산자

및 제2피연산자 중 Zero-data인 피연산자가 존재하는 것으로 판단되면, 제1피연산자 및 제2피연산자의 곱연산에 관하여 단계 S11 내지 단계 S15(달리 말해, (a) 단계 내지 (e) 단계)의 수행이 생략(skip)될 수 있다.

[0107] 상술한 설명에서, 단계 S11 내지 S15는 본원의 구현예에 따라서, 추가적인 단계들로 더 분할되거나, 더 적은 단계들로 조합될 수 있다. 또한, 일부 단계는 필요에 따라 생략될 수도 있고, 단계 간의 순서가 변경될 수도 있다.

[0108] 도 8은 복수의 제1비트군 각각과 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 선택적으로 곱셈기를 할당하는 과정에 대한 세부 동작 흐름도이다.

[0109] 도 8에 도시된 선택적으로 곱셈기를 할당하는 과정은 앞서 설명된 추론 속도 향상 장치(100)에 의하여 수행될 수 있다. 따라서, 이하 생략된 내용이라고 하더라도 추론 속도 향상 장치(100)에 대하여 설명된 내용은 도 8에 대한 설명에도 동일하게 적용될 수 있다.

[0110] 도 8을 참조하면, 단계 S131에서 할당부(130)는, 복수의 제1비트군 중 어느 하나 및 복수의 제2비트군 중 어느 하나를 선택할 수 있다.

[0111] 다음으로, 단계 S132에서 할당부(130)는, 선택된 제1비트군 및 선택된 제2비트군 중 값이 0인 비트군이 존재하는지 판단할 수 있다.

[0112] 만약, 단계 S132의 판단 결과, 선택된 제1비트군 및 선택된 제2비트군 중 값이 0인 비트군이 존재하면(YES), 할당부(130)는 단계 S133에서 해당 제1비트군 및 제2비트군에 대한 곱셈기를 미할당(생략)할 수 있다.

[0113] 이와 달리, 단계 S132의 판단 결과, 선택된 제1비트군 및 선택된 제2비트군 중 값이 0인 비트군이 존재하지 않으면(NO), 할당부(130)는 단계 S134에서 해당 제1비트군 및 제2비트군에 대한 곱셈기를 할당할 수 있다.

[0114] 다음으로, 단계 S135에서 할당부(130)는, 단계 S131로 되돌아가 복수의 제1비트군 각각과 복수의 제2비트군 각각 사이의 개별 곱연산이 가능한 모든 조합에 대하여 단계 S131 내지 단계 S134가 반복 수행되도록 할 수 있다.

[0115] 상술한 설명에서, 단계 S131 내지 S135는 본원의 구현예에 따라서, 추가적인 단계들로 더 분할되거나, 더 적은 단계들로 조합될 수 있다. 또한, 일부 단계는 필요에 따라 생략될 수도 있고, 단계 간의 순서가 변경될 수도 있다

[0116] 본원의 일 실시 예에 따른 신경망 모델의 추론 속도 향상 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 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. 상기된 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0117] 또한, 전술한 신경망 모델의 추론 속도 향상 방법은 기록 매체에 저장되는 컴퓨터에 의해 실행되는 컴퓨터 프로그램 또는 애플리케이션의 형태로도 구현될 수 있다.

[0118] 전술한 본원의 설명은 예시를 위한 것이며, 본원이 속하는 기술분야의 통상의 지식을 가진 자는 본원의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0119] 본원의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본원의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0120] 100: 신경망 모델의 추론 속도 향상 장치

110: 이진화부

120: 분할부

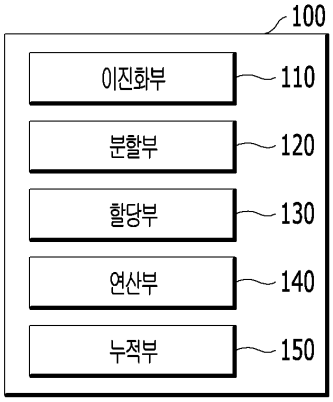
130: 할당부

140: 연산부

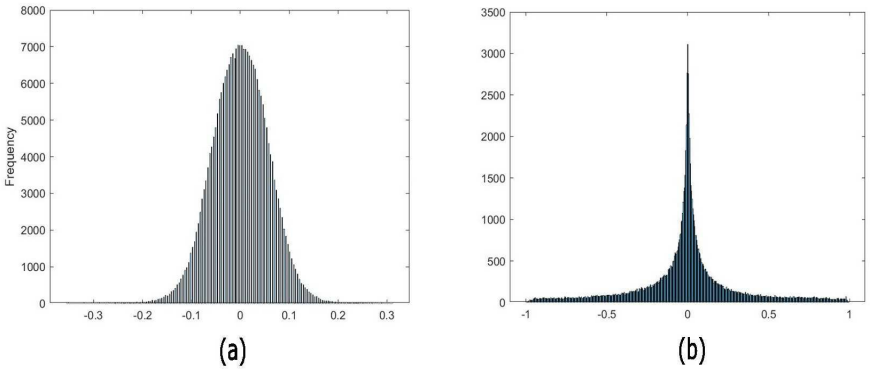
150: 누적부

도면

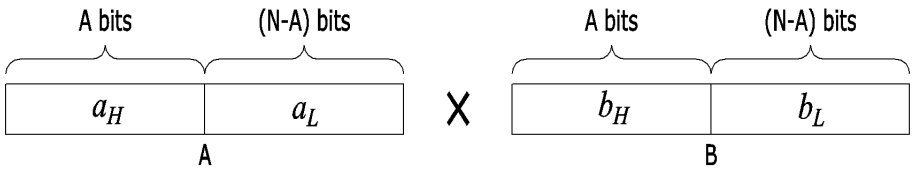
도면1



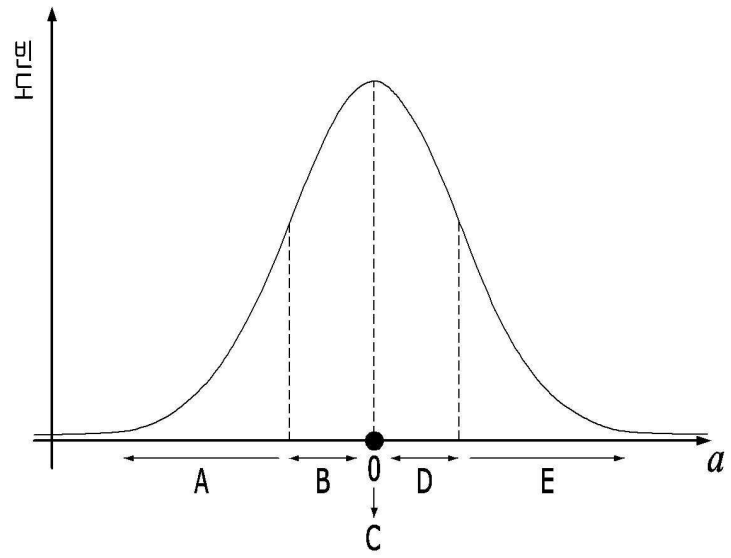
도면2



도면3



도면4



도면5

- $A = a_H \times 2^{(N-A)} + a_L$
- $B = b_H \times 2^{(N-A)} + b_L$

$$A \times B = a_H b_H \times 2^{2(N-A)} + (a_H b_L + a_L b_H) \times 2^{(N-A)} + a_L b_L$$

(a)

- $A = a_H \times 2^8 + a_L$
- $B = b_H \times 2^8 + b_L$

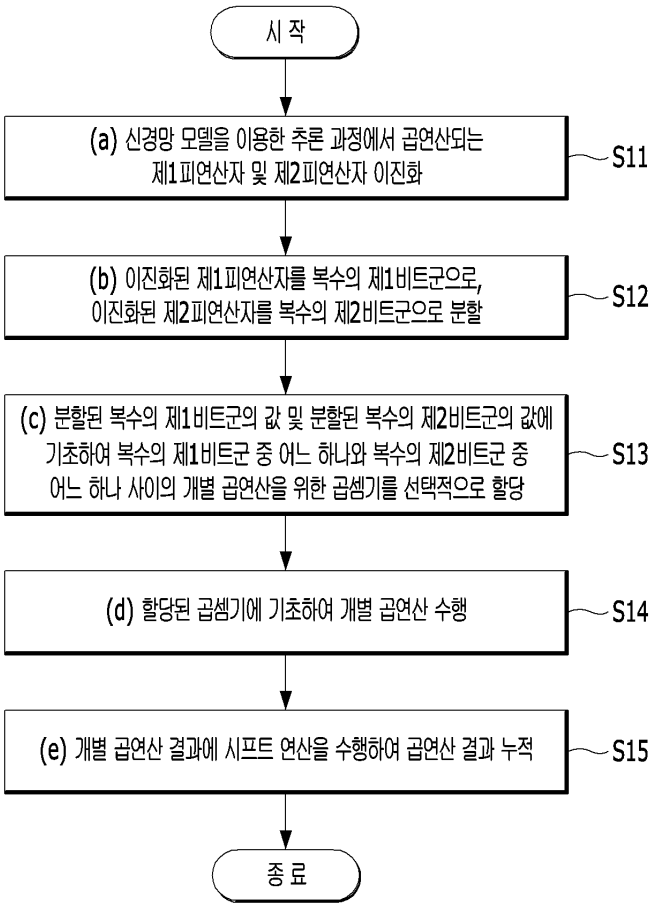
$$A \times B = a_H b_H \times 2^{16} + (a_H b_L + a_L b_H) \times 2^8 + a_L b_L$$

(b)

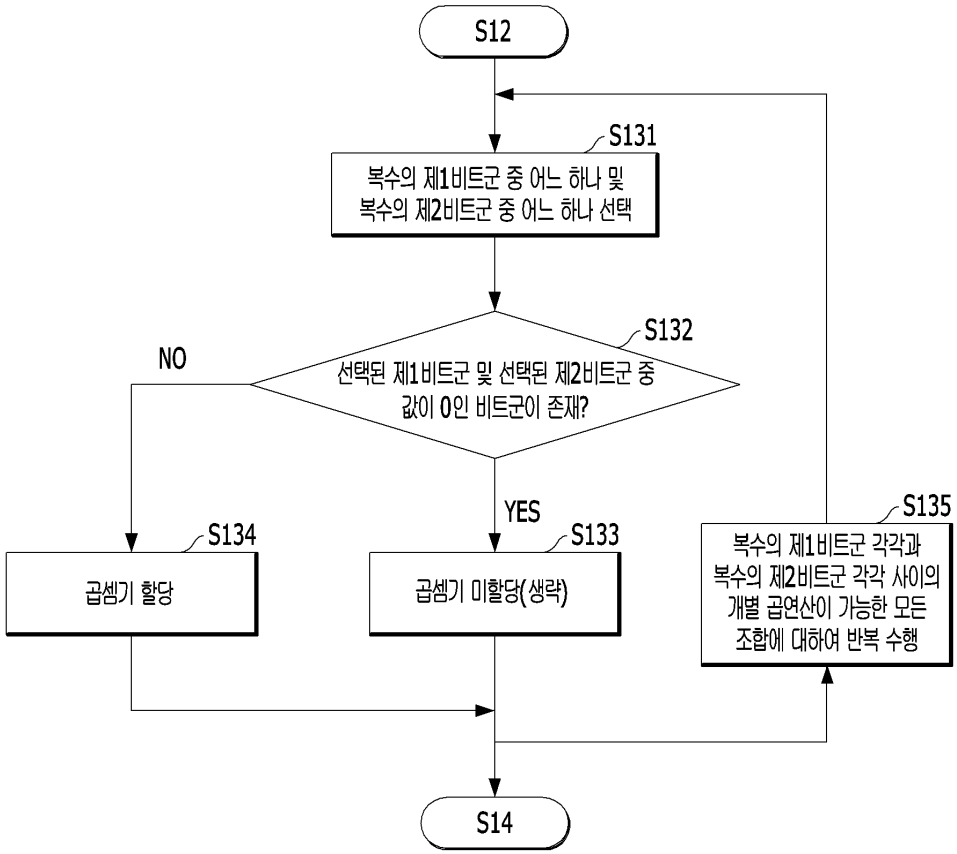
도면6

a_H	a_L	b_H	b_L	$A \times B$	기존 연산량	Zero-skipping 적용 연산량	본원의 방법 적용 연산량
$=0$	$=0$	$=0$	$=0$	0	4	**0**	**0**
$=0$	$=0$	$=0$	$\neq 0$	0	4	**0**	**0**
$=0$	$=0$	$\neq 0$	$=0$	0	4	**0**	**0**
$=0$	$=0$	$\neq 0$	$\neq 0$	0	4	**0**	**0**
$=0$	$\neq 0$	$=0$	$=0$	0	4	**0**	**0**
$=0$	$\neq 0$	$=0$	$\neq 0$	$a_L b_L$	4	4	**1**
$=0$	$\neq 0$	$\neq 0$	$=0$	$a_L b_H \times 2^8$	4	4	**1**
$=0$	$\neq 0$	$\neq 0$	$\neq 0$	$a_L b_H \times 2^8 + a_L b_L$	4	4	**2**
$\neq 0$	$=0$	$=0$	$=0$	0	4	**0**	**0**
$\neq 0$	$=0$	$=0$	$\neq 0$	$a_H b_L \times 2^8$	4	4	**1**
$\neq 0$	$=0$	$\neq 0$	$=0$	$a_H b_H \times 2^{16}$	4	4	**1**
$\neq 0$	$=0$	$\neq 0$	$\neq 0$	$a_H b_H \times 2^{16} + a_H b_L \times 2^8$	4	4	**2**
$\neq 0$	$\neq 0$	$=0$	$=0$	0	4	**0**	**0**
$\neq 0$	$\neq 0$	$=0$	$\neq 0$	$a_H b_L \times 2^8 + a_L b_L$	4	4	**2**
$\neq 0$	$\neq 0$	$\neq 0$	$=0$	$a_H b_H \times 2^{16} + a_L b_H \times 2^8$	4	4	**2**
$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$	$a_H b_H \times 2^{16} + (a_H b_L + a_L b_H) \times 2^8 + a_L b_L$	4	4	**4**

도면7



도면8


S12
S131
복수의 제1비트군 중 어느 하나 및
복수의 제2비트군 중 어느 하나 선택
S132
선택된 제1비트군 및 선택된 제2비트군 중
값이 0인 비트군이 존재?
NO
YES
S134
곱셈기 할당
S133
곱셈기 미할당(생략)
S135
복수의 제1비트군 각각과
복수의 제2비트군 각각 사이의
개별 곱연산이 가능한 모든
조합에 대하여 반복 수행
S14