



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0050895
(43) 공개일자 2020년05월12일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2006.01) G06N 3/04 (2006.01)
G06N 3/08 (2006.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06N 3/04 (2013.01)
(21) 출원번호 10-2019-0138779
(22) 출원일자 2019년11월01일
심사청구일자 2019년11월01일
(30) 우선권주장
1020180133928 2018년11월02일 대한민국(KR)

(71) 출원인
울산과학기술원
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
이종은
울산광역시 울주군 언양읍 유니스트길 50
심현욱
울산광역시 울주군 언양읍 유니스트길 50
(74) 대리인
특허법인(유한)아이시스

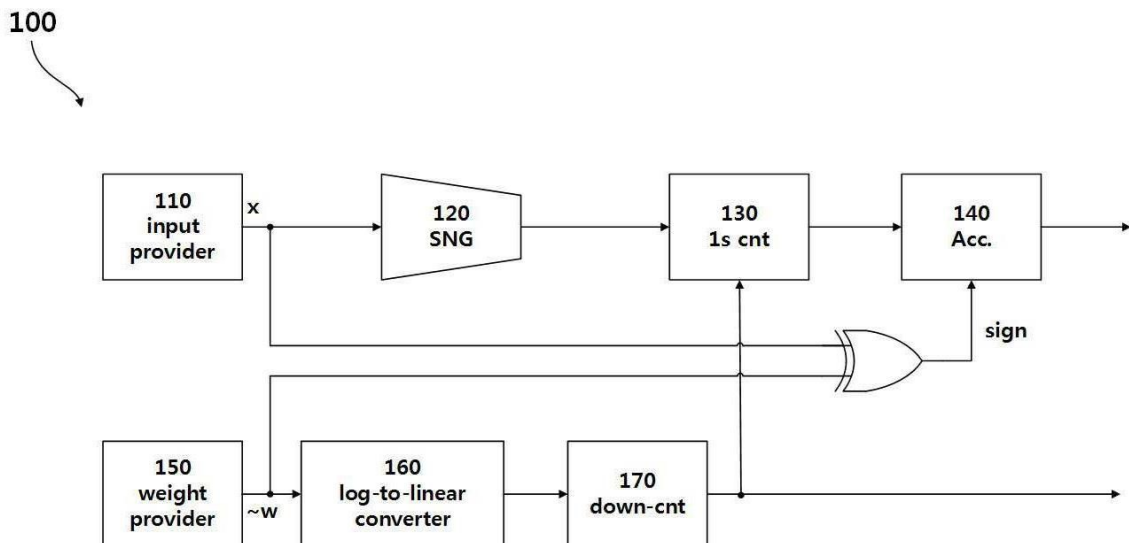
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 스토캐스틱 연산을 위한 로그-양자화된 곱셈 및 누적기와 이를 포함하는 가속기

(57) 요약

본 실시예에 의한 곱셈 누적기(multiply-accumulator)는 입력 데이터를 제공하는 입력 제공부와, 가중치 입력을 제공받아 로그로 양자화된 가중치를 출력하는 가중치 제공부와, 로그로 양자화된 가중치를 제공받고, 선형으로 변환된 가중치를 출력하는 로그 대 선형 변환기와, 선형으로 변환된 가중치로부터 윈도우 너비를 출력하는 역 계수기와, 입력 데이터에 상응하는 스토캐스틱 수에서 윈도우 너비 내의 1의 값을 계수하는 1 계수기 및 1 계수기가 출력한 값을 누적하여 출력하는 누적기를 포함한다.

대표도



(52) CPC특허분류

G06N 3/08 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 2016M3A7B4910576

부처명 과학기술정보통신부

연구관리전문기관 한국연구재단

연구사업명 나노·소재원천기술개발사업

연구과제명 채구성로직 아키텍처 기술개발

기 여 율 1/1

주관기관 울산과학기술원

연구기간 2018.03.01 ~ 2019.01.31

명세서

청구범위

청구항 1

입력 데이터를 제공하는 입력 제공부;

가중치 입력을 제공받아 로그로 양자화된 가중치를 출력하는 가중치 제공부;

상기 로그로 양자화된 가중치를 제공받고, 선형으로 변환된 가중치를 출력하는 로그 대 선형 변환기;

상기 선형으로 변환된 가중치로부터 윈도우 너비를 출력하는 역 계수기;

상기 입력 데이터에 상응하는 스토캐스틱 수에서 상기 윈도우 너비 내의 1의 값을 계수하는 1 계수기 및

상기 1 계수기가 출력한 값을 누적하여 출력하는 누적기를 포함하는 곱셈 누적기(multiply-accumulator).

청구항 2

제1항에 있어서,

상기 곱셈 누적기는,

상기 입력 데이터의 부호 비트와

상기 로그로 양자화된 가중치에서의 부호 비트로부터 상기 1 계수기가 출력한 값의 부호를 연산하는 배타적 논리 합(XOR, exclusive or) 게이트를 더 포함하는 곱셈 누적기.

청구항 3

제1항에 있어서,

상기 곱셈 누적기는,

상기 입력 데이터를 스토캐스틱 수로 변환하는 스토캐스틱 수 제너레이터(stochastic number generator)를 더 포함하는 곱셈 누적기.

청구항 4

제1항에 있어서,

상기 1 계수기는

상기 입력 데이터에 상응하는 스토캐스틱 수를 형성하는 곱셈 누적기.

청구항 5

제1항에 있어서,

상기 1 계수기는,

상기 스토캐스틱 수가 미리 정해진 비트 수에 따라 분할되어 형성된 복수개의 블록 단위로 계수를 수행하는 곱셈 누적기.

청구항 6

제5항에 있어서,

상기 윈도우의 너비가 상기 블록의 비트 너비보다 크거나 같을 때,

상기 1 계수기는,

수학식

$$1s_cnt = \{x_{N-1}, \dots, x_{N-n}\}_{bin} + mux\{x_{N-n-1}, \dots, x_0\}$$

을 연산하여 상기 블록 내의 1의 개수를 계수하는 곱셈 누적기.

청구항 7

제5항에 있어서,

상기 윈도우의 너비가 상기 블록의 비트 너비보다 작을 때,

상기 1 계수기는,

수학식

$$\sum_{i=1}^n x_{N-i} \times round\left(\frac{w}{2^i}\right)$$

를 연산하여 상기 윈도우 너비내 1의 개수를 계수하는 1 계수기를 포함하는 곱셈 누적기.

청구항 8

제5항에 있어서,

상기 윈도우의 너비가 상기 블록의 비트 너비보다 작을 때,

상기 1 계수기는,

수학식

$$\sum_{i=1}^n x_{N-i} \times (w \gg i) + \sum_{i=1}^n x_{N-i} \wedge w_{i-1}$$

을 연산하여 상기 윈도우 너비내 1의 개수를 계수하는 1 계수기를 포함하는 곱셈 누적기.

청구항 9

제5항에 있어서,

상기 윈도우의 너비가 상기 블록의 비트 너비보다 작을 때,

상기 1 계수기는,

$$\sum_{i=1}^n x_{N-i} \times (2^P \gg i) + x_{N-P-1}$$

를 연산하여 상기 윈도우 너비내 1의 개수를 계수하는 1 계수기를 포함하는 곱셈 누적기.

청구항 10

제5항에 있어서,

상기 1 계수기는,

블록의 최초 비트에서 최종 이전 비트 비트를 입력받아 우측 시프트 연산을 수행하는 시프터;

상기 블록의 최초 비트에서 최후 이전 비트 및 상기 블록의 최후 비트가 입력으로 제공되어 어느 하나를 선택하여 출력하는 다중화기;

상기 시프터와 상기 다중화기의 출력을 더하는 덧셈기; 및

상기 시프터가 시프트 연산하는 비트수와 상기 다중화기의 입력을 선택하는 제어부를 포함하는 1 계수기를 포함하는 곱셈 누적기.

청구항 11

복수의 입력 데이터 벡터들을 제공하는 입력 제공부;

가중치 입력을 제공받아 로그로 양자화된 가중치를 출력하는 가중치 제공부;

상기 로그로 양자화된 가중치를 제공받고, 선형으로 변환된 가중치를 출력하는 로그 대 선형 변환기;

상기 선형으로 변환된 가중치로부터 윈도우 너비를 출력하는 역 계수기;

상기 입력 데이터 벡터에 각각에 상응하는 스토캐스틱 수에서 상기 윈도우 너비 내의 1의 값을 계수하는 복수의 1 계수기들 및

상기 1 계수기가 출력한 값을 누적하여 출력하는 상기 복수의 누적기들을 포함하는 매트릭스 벡터 곱셈기(matrix-vector multiplier).

청구항 12

제11항에 있어서,

상기 매트릭스 벡터 곱셈기(matrix-vector multiplier)는,

상기 입력 데이터 벡터 각각의 부호 비트와

상기 로그로 양자화된 가중치에서의 부호 비트로부터 상기 1 계수기가 출력한 값의 부호들을 연산하는 상기 복수의 배타적 논리 합(XOR, exclusive or) 게이트들을 더 포함하는 매트릭스 벡터 곱셈기(matrix-vector multiplier).

청구항 13

제11항에 있어서,

상기 곱셈 누적기는,

상기 입력 데이터 벡터 각각을 상응하는 스토캐스틱 수로 변환하는 상기 복수의 스토캐스틱 수 제너레이터(stochastic number generator)들을 더 포함하는 매트릭스 벡터 곱셈기(matrix-vector multiplier).

청구항 14

제11항에 있어서,

상기 1 계수기는

상기 입력 데이터 벡터에 상응하는 스토캐스틱 수를 형성하는 매트릭스 벡터 곱셈기(matrix-vector multiplier).

청구항 15

제11항에 있어서,

상기 1 계수기는,

상기 스토캐스틱 수가 미리 정해진 비트 수에 따라 분할되어 형성된 복수개의 블록 단위로 계수를 수행하는 매트릭스 벡터 곱셈기(matrix-vector multiplier).

청구항 16

제15항에 있어서,

상기 윈도우의 너비가 상기 블록의 비트 너비보다 크거나 같을 때,

상기 1 계수기는,

수학식

$$1s_cnt = \{x_{N-1}, \dots, x_{N-n}\}_{bin} + mux\{x_{N-n-1}, \dots, x_0\}$$

을 연산하여 상기 블록 내의 1의 개수를 계수하는 매트릭스 벡터 곱셈기(matrix-vector multiplier).

청구항 17

제15항에 있어서,

상기 윈도우의 너비가 상기 블록의 비트 너비보다 작을 때,

상기 1 계수기는,

수학식

$$\sum_{i=1}^n x_{N-i} \times round\left(\frac{w}{2^i}\right)$$

를 연산하여 상기 윈도우 너비내 1의 개수를 계수하는 1 계수기를 포함하는 매트릭스 벡터 곱셈기(matrix-vector multiplier).

청구항 18

제15항에 있어서,

상기 윈도우의 너비가 상기 블록의 비트 너비보다 작을 때,

상기 1 계수기는,

수학식

$$\sum_{i=1}^n x_{N-i} \times (w \gg i) + \sum_{i=1}^n x_{N-i} \wedge w_{i-1}$$

을 연산하여 상기 윈도우 너비내 1의 개수를 계수하는 1 계수기를 포함하는 매트릭스 벡터 곱셈기(matrix-vector multiplier).

청구항 19

제15항에 있어서,

상기 윈도우의 너비가 상기 블록의 비트 너비보다 작을 때,

상기 1 계수기는,

$$\sum_{i=1}^n x_{N-i} \times (2^p \gg i) + x_{N-p-1}$$

를 연산하여 상기 윈도우 너비내 1의 개수를 계수하는 1 계수기를 포함하는 매트릭스 벡터 곱셈기(matrix-

vector multiplier).

청구항 20

제15항에 있어서,

상기 1 계수기는,

블록의 최초 비트에서 최후 이전 비트 비트를 입력받아 우측 시프트 연산을 수행하는 시프터;

상기 블록의 최초 비트에서 최후 이전 비트 및 상기 블록의 최후 비트가 입력으로 제공되어 어느 하나를 선택하여 출력하는 다중화기;

상기 시프터와 상기 다중화기의 출력을 더하는 덧셈기; 및

상기 시프터가 시프트 연산하는 비트수와 상기 다중화기의 입력을 선택하는 제어부를 포함하는 1 계수기를 포함하는 매트릭스 벡터 곱셈기(matrix-vector multiplier).

발명의 설명

기술 분야

[0001] 본 기술은 로그-양자화된 가중치 값을 이용하여 스토캐스틱 연산을 수행하는 MAC 및 이를 포함하는 가속기와 관련된다.

배경 기술

[0002] 스토캐스틱 컴퓨팅(stochastic computing)은 비트 스트림(bitstream)에서 1의 등장 빈도를 이용하여 숫자를 표현하는 방법을 이용하는 컴퓨팅 방법으로, 종래의 2의 보수(2's complement)에 기초한 디지털 구현예들에 비하여 저비용, 낮은 전력 소모, 높은 내오류 특성 및 점진적 정밀도를 갖는 등 많은 장점을 가진다. 최근 스토캐스틱 컴퓨팅은 저전력 이미지 처리와 같은 다른 응용분야 뿐만 아니라 딥 뉴럴 네트워크(DNN, deep neural network) 가속에 효과적인 것으로 나타났다.

[0003] 스토캐스틱 컴퓨팅은 근사 컴퓨팅(approximate computing)의 맥락에서 환경, 입력 또는 컴퓨팅 상황(예를 들어, DNN의 다른 계층)에 따라 처리 시간 및 에너지 소모와 결과 품질을 트레이드 오프(trade-off)할 수 있는 유용한 프레임워크를 제공한다. 이러한 런타임 조절은 매우 자연스러우며, 스토캐스틱 컴퓨팅에서 과부하(overhead)를 발생시키지 않는다.

[0004] 한편, 사람의 뇌에서는 뉴런들이 이어져 네트워크를 형성하는데, 그 과정에서 학습, 추론 및 사고한다. 뉴럴 네트워크는 그 학습 활동을 모방하여 형성된다. 뉴럴 네트워크의 구성 요소로는 뉴런과 시냅스가 있을 수 있으며, 시냅스는 뉴런과 뉴런이 연결되는 부분으로, 뉴런 사이의 시냅스 연결도의 강하고, 약한 정도를 가중치(weight)라고 한다.

[0005] 뉴럴 네트워크를 학습한다고 하는 것은 네트워크를 만들고, 학습을 시키고 싶은 데이터를 주었을 때, 목적하는 데이터에 상응하는 값이 입력되고, 가중치 값과 곱하여 목적하는 값을 얻을 때까지 가중치 값을 변화시켜 출력될 때까지 학습을 시키는 것이며, 결과적으로, 학습이라는 과정은 가중치를 얻어내는 과정이다.

[0006] 가속기(accelerator)를 이용하여 뉴럴 네트워크 학습 및 추론을 수행하는데, 그 이유는 뉴럴 네트워크라는 학습 프로그램(및 추론 프로그램)을 수행하는 것은 수많은 병렬적 연산을 수행하는 것이다. 즉, 뉴런과 뉴런 사이의 가중치 뉴런을 가중치 매트릭스로 표현하고, 입력 벡터와 가중치 매트릭스를 곱하여 출력 벡터를 얻는 과정으로 표현할 수 있으나, 수행하는 연산의 횟수가 많다.

발명의 내용

해결하려는 과제

[0007] 스토캐스틱 컴퓨팅의 가장 중요한 문제 중 하나는 정밀도와 관련하여 런 타임 및 에너지 소비가 기하급수적으로 증가한다는 것이다. 스토캐스틱 컴퓨팅은 N 비트 바이너리와 유사한 정밀도를 달성하기 위해 길이 2^N 의 비트 스트림을 필요로하기 때문이며, 이러한 점은 높은 정밀도를 달성하기 위한 SC의 지수적 복잡성은 계산 및 스토리

지 요구 사항 모두에 영향을 미치고, 대규모 SC 기반 DNN과 같은 큰 문제에 SC를 적용하는 데 심각한 문제가 된다. 비트 병렬화(bit-parallelism)는 지연 시간(latency)을 줄일 수 있는 방법 중 하나이나, 비용이 많이 소모된다.

과제의 해결 수단

[0008] 본 실시예에 의한 곱셈 누적기(multiply-accumulator)는 입력 데이터를 제공하는 입력 제공부와, 가중치 입력을 제공받아 로그로 양자화된 가중치를 출력하는 가중치 제공부와, 로그로 양자화된 가중치를 제공받고, 선형으로 변환된 가중치를 출력하는 로그 대 선형 변환기와, 선형으로 변환된 가중치로부터 윈도우 너비를 출력하는 역 계수기와, 입력 데이터에 상응하는 스토캐스틱 수에서 윈도우 너비 내의 1의 값을 계수하는 1 계수기 및 1 계수기가 출력한 값을 누적하여 출력하는 누적기를 포함한다.

[0009] 본 실시예에 의한 매트릭스 벡터 곱셈기(matrix-vector multiplier)는 복수의 입력 데이터 벡터들을 제공하는 입력 제공부와, 가중치 입력을 제공받아 로그로 양자화된 가중치를 출력하는 가중치 제공부와, 로그로 양자화된 가중치를 제공받고, 선형으로 변환된 가중치를 출력하는 로그 대 선형 변환기와, 선형으로 변환된 가중치로부터 윈도우 너비를 출력하는 역 계수기와, 입력 데이터 벡터에 각각에 상응하는 스토캐스틱 수에서 윈도우 너비 내의 1의 값을 계수하는 복수의 1 계수기들 및 1 계수기가 출력한 값을 누적하여 출력하는 복수의 누적기들을 포함한다.

발명의 효과

[0010] 본 실시예에 의하면 작은 면적으로 높은 에너지 효율을 가지는 곱셈 및 누적기와 이를 이용하는 가속기가 제공된다.

도면의 간단한 설명

[0011] 도 1은 본 실시예에 의한 로그 양자화를 이용하는 곱셈 및 누적기의 개요를 도시한 도면이다.

도 2(a)는 스토캐스틱 수 제너레이터의 동작을 설명하기 위한 개요적인 도면이고, 도 2(b)는 비트 병렬화에 따른 스토캐스틱 수 제너레이터의 동작 개요를 설명하기 위한 도면이며, 도 2(c)는 비트 병렬화에 따른 스토캐스틱 수 제너레이터(120) 동작의 다른 예를 설명하기 위한 도면이다.

도 3(a)는 비트 병렬화에 따라 동작하는 1 계수기에 제공된 윈도우 너비가 8인 경우를 예시한 도면이고, 도 3(b)는 비트 병렬화에 따라 동작하는 1 계수기에 제공된 윈도우 너비가 2 경우를 예시한 도면이다,

도 4는 본 실시예에 의한 1 계수기의 개요를 도시한 블록도이다.

도 5은 본 실시예에 의한 곱셈 및 누적기를 포함하는 가속기의 개요를 도시한 블록도이다.

도 6는 스토캐스틱 컴퓨팅 가속기의 평균 제곱 오차(mean square error)로 표시된 곱셈 오차를 도시한다.

도 7(a)는 가속기의 면적, 도 7(b)는 전력 소모를 비교 도시한 도면이다.

도 8은 5000회 반복하여 미세 조정된 대상 신경 네트워크에 인식 테스트 결과를 도시한 도면으로, 도 8(a)는 10-클래스 이미지 분류 결과이고, 도 8(b)는 1000 클래스 이미지 분류 결과이다.

발명을 실시하기 위한 구체적인 내용

[0012] 이하에서는 첨부된 도면들을 참조하여 본 실시예를 설명한다. 도 1은 본 실시예에 의한 로그 양자화를 이용하는 곱셈 및 누적기(MAC, multiply and accumulator, 100)의 개요를 도시한 도면이다. 도 1을 참조하면, 본 실시예에 의한 곱셈 및 누적기(100)는 입력(x)을 제공하는 입력 제공부(110)와, 입력(x)에 상응하는 스토캐스틱 수(stochastic number)를 출력하는 스토캐스틱 수 제너레이터(120, SNG, stochastic number generator) 및 스토캐스틱 수에 포함된 1을 계수하는 1 계수기(130, 1s cnt) 및 가중치(weight)에 로그(logarithm)를 취하여 출력하는 가중치 제공부(150, weight provider), 입력된 로그 양자화 된 수를 선형으로 변환하는 로그 대 선형 변환기(log-to-linear converter, 160)와, 미리 정해진 수에서 감산하는 역계수기(170, down-cnt) 및 가중치(~w)와 입력(x)으로부터 곱셈 결과의 부호를 연산하여 출력하는 XOR 게이트를 포함한다.

[0013] 입력 제공부(110)는 뉴럴 네트워크(미도시)에 제공되는 입력(x)을 제공한다. 일 예로, 입력(x)는 이진수(binary)로 뉴럴 네트워크에 제공되는 입력 벡터의 일 성분일 수 있다. 일 실시예로, 입력 제공부(110)는 본 실

시에에 의한 곱셈 및 누적기(100)의 외부에서 제공된 입력(x)을 출력한다.

- [0014] 도 2(a)는 스토캐스틱 수 제너레이터(120)의 동작을 설명하기 위한 개요적인 도면이다. 도 2를 참조하면 스토캐스틱 수 제너레이터(120)는 스토캐스틱 수 제너레이터(120)는 입력(x)의 확률에 상응하는 스토캐스틱 수를 형성하여 비트 스트림(bit-stream)의 형태로 출력한다. 일 실시예로, x가 십진수로 (0.5)dec이고, 16비트를 가정하면 스토캐스틱 수 제너레이터(120)는 x3를 1로, x2, x1, x0를 0으로 설정하고, 선택 FSM(selector FSM, Finite State Machine)을 통해 다중화기(MUX)를 제어하여 비트 스트림 (1010 1010 1010 1010)bin을 출력하도록 제어한다.
- [0015] 다른 실시예로, x의 값이 (0.625)dec인 경우에, $0.625 = 0.5 + 0.125$ 이므로, 스토캐스틱 수 제너레이터(120)는 $x3 = x1 = 1$ 로 설정하고, $x2 = x0 = 0$ 으로 설정한다. 선택 FSM을 통해 다중화기(MUX)를 제어하여 비트 스트림 (1011 1010 1011 1010)bin을 출력하도록 제어한다.
- [0016] 도 2(b)는 비트 병렬화에 따른 스토캐스틱 수 제너레이터(120)의 동작 개요를 설명하기 위한 도면이다. 도 2(b)를 참조하면, 비트 병렬화 방식에 의하면 1 계수기(130)는 $\frac{N}{2^n}$ 사이클로 수행된다. 도 2(b)로 예시된 실시예에서, N이 총 16 비트이고, n은 비트 병렬화 매개변수이다. 비트 병렬화가 2^2 (n=2) 만큼 이루어진 바, 총 4 사이클로 수행된다.
- [0017] 1 계수기(130)가 총 4 사이클로 수행되며, 스토캐스틱 수 제너레이터(120)는 x의 16비트를 각 블록은 4 비트를 가지는 b1 내지 b4의 4 블록으로 분할한다. 각 블록의 첫 세 비트는 점선으로 표시된 것과 같이 모두 x3, x2, x3 순으로 동일하다. 마지막 비트는 선택 FSM에 의하여 x1과 x0 중에서 선택된다. 따라서, b1, b2, b3 및 b4의 네 블록들의 세 비트에는 설정된 x3 및 x2 값이 그대로 기입될 수 있다.
- [0018] 일 예로, x가 십진수로 (0.5)dec일 때, 스토캐스틱 수 제너레이터(120)는 x3를 1로, x2, x1 및 x0를 0으로 설정하고, 선택 FSM(selector FSM, Finite State Machine)을 통해 다중화기(MUX)를 제어하여 첫 번째 블록(b1)은 (1010)bin, 두 번째 블록(b2)은 (1010)bin, 세 번째 블록(b3)은 (1010)bin, 네 번째 블록(b4)은 (1010)bin을 출력하도록 한다.
- [0019] 다른 예로, x가 십진수로 (0.625)dec일 때, 스토캐스틱 수 제너레이터(120)는 x3와 x1을 1로, x2 및 x0를 0으로 설정한다. 선택 FSM(selector FSM, Finite State Machine)을 통해 다중화기(MUX)를 제어하여 첫 번째 블록(b1)은 (1011)bin, 두 번째 블록(b2)은 (1010)bin, 세 번째 블록(b3)은 (1011)bin, 네 번째 블록(b4)은 (1010)bin을 출력하도록 한다.
- [0020] 도 2(c)는 비트 병렬화에 따른 스토캐스틱 수 제너레이터(120) 동작의 다른 예를 설명하기 위한 도면이다. 도 2(c)를 참조하면, x가 총 16 비트이고, 1 계수기(130)가 2^n (n=1) = 2 사이클로 수행되는 경우를 예시한다. 1 계수기(130)가 총 2 사이클로 수행되므로, 이에 부합하도록 스토캐스틱 수 제너레이터(120)는 x의 16비트를 b1과 b2의 2 블록으로 분할하며, 각 블록은 8 비트를 가진다.
- [0021] 각 블록의 첫 일곱 비트는 점선으로 표시된 것과 같이 모두 x3, x2, x3, x1, x3, x2, x3 순으로 동일하다. 마지막 비트는 선택 FSM에 의하여 x0와 0 중에서 선택된다. 따라서, b1 및 b2의 두 블록들의 일곱 비트에는 설정된 x3, x2 및 x1 값이 그대로 기입될 수 있다.
- [0022] 도 1로 예시된 실시예는 스토캐스틱 수 제너레이터(120)가 입력된 데이터(x) 값에 따라 상응하는 확률의 스토캐스틱 수를 형성하여 1 계수기(130)에 제공하는 것을 예시한다. 그러나, 도 2(b) 및 도 2(c)로 예시된 실시예와 같이 스토캐스틱 수가 분할되어 형성된 블록에는, 블록 내의 비트 위치에 따라 정해진 값이 위치한다.
- [0023] 총 16비트 중에서, 어느 한 비트는 0으로 설정될 수 있다. 즉, x3는 0.5의 확률로 비트 스트림내에 위치하고, x2는 0.25의 확률로 비트 스트림내에 위치하고, x1는 0.125의 확률로 비트 스트림내에 위치하고, x0는 0.0625의 확률로 비트 스트림내에 위치한다. 이들을 다 포함하여도 확률이 1이 되지 않으므로 총 16비트 중 한 비트를 0으로 설정한다.
- [0024] 가중치 제공부(150)는 뉴럴 네트워크(미도시)의 가중치를 제공받고, 이를 로그로 양자화(quantize)하여 출력한다. 가중치 제공부(150)에 제공된 가중치를 w라 하면, 가중치 제공부(150)는 실수인 가중치 w에 로그를 취하고 반올림하여 출력한다. 가중치 제공부(150)가 출력하는 값 $\lceil \tilde{w} \rceil$ 은 아래의 수학식 1로 표시될 수 있다. 일 예로, 로그는 밑(base)이 2인 로그일 수 있다.

수학식 1

$$|\tilde{w}| = \text{round}(-\log(|w|))$$

(round ()): 반올림 함수, w: 실수 가중치)

표 1은 십진수인 실수 가중치(weight)와 $|\tilde{w}|$ 의 관계를 데이터 표시를 예시한다.

표 1

w	$\tilde{w}$		
Real value	Sign bit	$ \tilde{w} $	Binary format
-0.5	1	1	1 0001
0	0 or 1	0	0 0000 or 1 0000
0.25	0	2	0 0010

일 예로, 실수 가중치(weight)가 0.25이면 부호 비트는 0이고, 밑이 2인 로그를 취하여 부호를 연산하면, 그 값은 십진수로 $(2)_{\text{dec}}$ 이며, 이진수로는 $(0010)_{\text{bin}}$ 이다.

실수 가중치(weight)는 십진수 $(-1,1)$ 의 범위에 있으므로 그 절대값은 1보다 작다. 따라서, 0을 제외한 가중치(weight)의 절대값에 로그를 취하여 반올림한 $|\tilde{w}|$ 는 0보다 크다. 부호 비트를 포함하는 16비트로 표현될 수 있는 최소한의 단위는 $\frac{1}{2^{15}}$ 이고, 그때 가중치 제공부(150)의 출력 $|\tilde{w}|$ 는 15이다. w가 0인 경우를 제외하고 $|\tilde{w}|$ 는 $[1,15]$ 의 범위 내에 있으므로, 4 비트의 언사인드(unsigned) 정수로 $|\tilde{w}|$ 를 표시할 수 있다. 따라서, 가중치 제공부(150)가 출력하는 로그 양자화된 가중치 $|\tilde{w}|$ 는 1 비트의 부호 비트와 크기(magnitude)를 표시하는 4 비트를 포함하는 총 5 비트로 표시된다.

예시된 실시예와 같이 가중치(weight)를 로그 스케일로 양자화하며, 딥 뉴럴 네트워크에 있어서 부동 소수점 레벨 정확도(floating point level accuracy)를 얻도록 로그의 밑(base) 2를 취할 수 있다. 다른 실시예로, $\sqrt{2}$, $\sqrt[3]{2}$ 등의 2의 거듭제곱으로 표시되며, 2보다 정밀한 밑(base)을 선택하여 가중치를 양자화 하면 뉴럴 네트워크 인식의 정밀도를 개선할 수 있다.

이하에서는 1 비트의 부호 비트 및 크기(magnitude)를 표시하는 15비트를 포함하는 16비트 데이터의 정밀도를 가정한다. 다만, 이는 설명의 편의를 위한 것이며, 본 발명의 범위를 제한하기 위한 것이 아니다. 로그 대 선형 변환기(160)는 로그로 양자화된 이진 가중치 $|\tilde{w}|$ 를 제공받고, 이를 선형 이진수로 변환하여 출력한다.

일 예로, $w=0.25$ 이고 로그로 양자화된 값이 0 0010 이면, 로그 대 선형변환기(160) 출력값은 부호 비트(sign bit) + 15 bit 소수부 절대값 형태로 $(0\ 010\ 0000\ 0000\ 0000)_{\text{bin}}$ 으로 표시된다. 또한, $w=0.0125(=1/8)$ 일 때 로그 양자값은 0 0011이고, 로그 대 선형 변환기(160) 출력값은 $(0\ 001\ 0000\ 0000\ 0000)_{\text{bin}}$ 이다. 다른 예로, $w=0.0625(=1/16)$ 일 때, 로그 양자값은 0 0100이며, 로그 대 선형 변환기(160) 출력값은 $(0\ 000\ 1000\ 0000\ 0000)_{\text{bin}}$ 이다. 위에서 알 수 있는 바와 같이 선형으로 변환된 가중치 값은 로그로 양자화된 값이 선형으로 변환된 것이므로 가중치를 표현하는 여러 비트들 중 어느 한 자리만 "1"을 가진다.

- [0034] 일 실시예로, 상술한 바와 같이 16비트의 실수 가중치는 밑이 2인 로그가 취하여져서 부호 비트를 포함하는 5비트로 양자화된다. 로그 대 선형 변환기(160)는 5 비트의 양자화된 가중치를 제공받고 부호 비트를 포함하는 15비트의 선형으로 변환된 가중치를 역계수기(170)에 출력한다.
- [0035] 역계수기(170)는 로그-대-선형 변환기(160)가 출력한 15비트의 선형으로 변환된 가중치를 입력 받고, 선형으로 변환된 가중치에 상응하는 윈도우 너비를 연산하여 1 계수기(130)에 출력한다. 일 실시예로, 윈도우 너비(w)는 비트수와 선형으로 변환된 가중치를 곱하여 얻어진 수 일 수 있다. 일 예로, 선형으로 변환된 가중치에 상응하는 값이 부호 비트를 제외하고 010 0000 0000 0000 이면 이는 소수인 .25이다. 즉, 선형으로 변환된 가중치 값에는 0.25의 확률로 1이 포함되어 있는 것을 의미한다. 따라서, 가중치를 나타내는 비트 수가 16 이라 하면, 가중치에는 4 개의 "1"을 포함하며, 이 때의 윈도우 너비(width)는 4에 해당한다. 다른 예로, 선형으로 변환된 가중치 값 $w = 0.0125 (= 1/8)$ 일 때 로그 양자값은 0 0011이고, 로그 대 선형 변환기(160) 출력값은 (0 001 0000 0000 0000)bin이다. 이 때 윈도우 너비(width)는 $0.125 \times 16 = 2$ 이다.
- [0036] 역 계수기(170)가 비트 병렬화의 형태로 동작할 때, 역 계수기(170)는 윈도우 너비(width)를 2^n 씩 감산하여 1 계수기(130)에 출력한다. 도 2(b)로 예시된 실시예와 같이 1 계수기(130)가 2^n ($n=2$)인 4 사이클에 걸쳐 동작하고, 역 계수기(170)가 연산하여 출력한 윈도우 너비(width)가 8이라면 첫 사이클에서 역 계수기(170)가 출력하는 윈도우 너비(width)는 8이고, 다음 사이클에서는 4를 출력한다.
- [0037] 1 계수기(130)는 역계수기(170)가 출력하는 윈도우 너비(width)와 스토캐스틱 수 제너레이터(120)가 제공하는 비트 스트림을 입력받고, 제공된 너비 만큼의 윈도우 내에서 1의 개수를 계수한다.
- [0038] 1 계수기(130)가 비트 병렬화 형태로 동작하는 실시예로, 1 계수기(130)는 멀티 비트 입력을 제공 받을 수 있으며, 복수의 블록들로 분할된 비트 스트림에서, 각 블록에 포함된 비트들을 한 사이클 동안 읽어 각 블록에 포함된 1의 개수를 계수할 수 있다(bit-parallelism). 일 실시예로, 스토캐스틱 수 제너레이터(120)가 제공하는 비트 스트림이 총 16비트이고, 윈도우 너비(width)가 4인 경우에, 1 계수기는 한 사이클에 비트 스트림을 4개의 비트씩 읽는다.
- [0039] 일 실시예로, 1 계수기(130)는 윈도우의 너비(width)만큼의 데이터(x)의 비트들을 읽어서 1의 개수를 계수하고, 나머지는 처리하지 않는다. 일 실시예로, 1 계수기(130)에 제공된 데이터(x)가 이진수(1010 1011 1010 1010)bin이고, 역계수기(170)가 출력한 윈도우의 너비가 4비트이면, 1 계수기(130)는 비트 스트림의 최초 4 비트를 읽고, 포함된 1의 개수를 계수하여 결과인 십진수 (2)dec를 누적기(140)에 출력한다.
- [0040] 다른 예로, 1 계수기(130)에 제공된 데이터(x)가 (1010 1011 1010 1010)bin이고, 역계수기(170)가 출력한 윈도우의 너비가 8이면 1 계수기(130)은 비트 스트림의 최초 8 비트를 읽고, 포함된 1의 개수를 계수하여 결과인 십진수 (5)dec를 누적기(140)에 출력한다. 누적기(140)는 1 계수기(130)가 제공한 수를 누적하여 출력한다.
- [0041] 누적기(140)가 누적하여 출력한 결과는 입력 제공부(110)가 제공한 입력(x)와 가중치 제공부(150)가 출력한 가중치(weight)의 곱인 $w \times x$ 에 상응한다. XOR 게이트는 곱셈 및 누적기(100)는 입력 제공부(110)가 제공하는 입력(x)의 부호 비트와 가중치 제공부(150)가 제공하는 양자화된 가중치의 부호 비트를 제공받아 곱셈 결과의 부호를 연산하여 부호 비트(sign)를 누적기(140)에 출력한다. 누적기(140)는 부호 비트(sign)에 따라 곱셈 결과를 양수 혹은 음수로 취급하여 누적한다.
- [0042] 도 3(a) 및 도 3(b)는 비트 병렬화로 동작하는 1 계수기(130)의 동작을 설명하기 위한 도면이다. 수학식 2는 1 계수기(130)의 동작을 설명하기 위한 수학적식이다. 도 3을 참조하면, 1 계수기(130)의 입력이 $\{x_{N-1}, \dots, x_0\}$ 인 N 비트에서 $N = 16$ 이고, 1 계수기(130)는 $16 / 2^n$ ($n=2$) = 4 사이클의 비트 병렬화에 따라 수행된다고 가정한다.

수학식 2

[0043]
$$1s_cnt = \{x_{N-1}, \dots, x_{N-n}\}_{bin} + mux\{x_{N-n-1}, \dots, x_0\} \cdots w \geq 2^n$$

- [0044] 도 3(a)로 예시된 것과 같이 윈도우의 너비(width)가 8인 경우를 가정한다. 윈도우의 너비(width)가 $2^2 = 4$ 보다 크거나 같으므로, 1 계수기(130)는 수학식 2에 따라 제1 블록(b1) 및 제2 블록(b2)에 포함된 "1"의 개수를

계수한다. 또한, 1 계수기(130)는 비트 병렬화로 동작하므로, 제1 블록(b1)내의 1의 개수를 계수한 후, 제2 블록(b2) 내의 1의 개수를 계수한다. 일 실시예로, 역계수기(170)이 최초사이클에서 제공한 윈도우 너비(width)은 8일 수 있고, 다음 사이클에서는 4일 수 있다. 비트 병렬화로 동작하는 1 계수기(130)는 최초 사이클에서 제1 블록(b1)에 포함된 1의 개수를 계수하고, 다음 사이클에서는 제2 블록(b2)에 포함된 1의 개수를 계수한다.

[0045] 제1 블록(b1)의 최초 3 비트는 상술한 바와 같이 x_3, x_2, x_3 로 고정되며, 마지막 비트가 x_1 혹은 x_0 로 변화할 수 있다. 이는 $(x_3x_2)_{bin} + \text{mux}(x_1, x_0)$ 로 나타낼 수 있다. 예를 들어, x_3 는 1이고 x_2 가 0이며, 제1 블록(b1)에서 x_1 은 1인 경우에, 제1 블록(b1)에 포함된 "1"의 개수는 $(10)_{bin} + (1)_{bin} = (11)_{bin} = (3)_{dec}$ 로 구할 수 있다. 상술한 바와 같이 다중화기(MUX)가 x_1, x_0 를 선택하는 신호는 선택 유한 상태 기계(Finite State Machine, FSM)으로부터 제공될 수 있다.

[0046] 두번째 사이클에서 1 계수기(130)는 제2 블록(b2)에 포함된 1의 개수를 계수한다. 제1 블록(b1)에서 1을 계수하는 방법과 유사하게 $(x_3x_2)_{bin} + \text{mux}(x_1, x_0)$ 를 연산할 수 있으며, 제2 블록(b2)에서는 다중화기를 통하여 x_0 를 선택하도록 한다. 위의 예와 같이, x_3 는 1이고 x_2 가 0이며, 제2 블록(b1)에서 x_0 은 0인 경우에, 제2 블록(b2)에 포함된 "1"의 개수는 $(10)_{bin} + (0)_{bin} = (10)_{bin} = (2)_{dec}$ 로 구할 수 있다.

[0047] 도 3(b)로 예시된 것과 같이 윈도우의 너비(width)가 2^n ($n=2$) = 4보다 작을 때 1의 개수를 계수하는 방법은 아래와 같다.

수학식 3

$$\sum_{i=1}^n x_{N-i} \times \text{round}\left(\frac{w}{2^i}\right)$$

[0048]

[0049] 즉, 도 3(b)에서, x_3 는 1이고, x_2 는 0인 경우를 보면, 제1 블록(b1)의 최초 3비트에서의 1의 개수는 수학식 3에

$x_3 \times \text{round}\left(\frac{2}{2}\right) + x_2 \times \text{round}\left(\frac{2}{4}\right)$ 로 연산된다. 즉, 윈도우 너비(width)에 상응하는 비트 수를 x_3 가 나타내는 확률인 1/2에 곱하여 반올림(round)하면 윈도우 너비 내 x_3 가 포함된 개수를 얻을 수 있다. 마찬가지로, 윈도우 너비(width)인 2에 x_2 가 나타내는 확률인 1/4을 곱하여 반올림(round)하면 윈도우 너비 내 x_2 가 포함된 개수를 얻을 수 있다. 따라서, 이들 각각에 x_3 값과 x_2 값을 곱하면 윈도우 너비(width) 내의 1의 개수를 구할 수 있다.

[0050] 1의 개수를 계수할 수 있는 다른 방법은 아래와 같다.

수학식 4

$$\sum_{i=1}^n x_{N-i} \times (w \gg i) + \sum_{i=1}^n x_{N-i} \wedge w_{i-1}$$

[0051]

[0052] ($\wedge$: 비트-와이즈 논리곱(bit-wise AND) 연산)

[0053] 수학식 3에서 $\frac{w}{2^i}$ 를 정수부와 소수부로 분할하여 고려한다. 정수부는 w 를 i 번 우측으로 시프트 연산한 것과 동일하다. 즉, 정수부를 a 라고 하면, $a = w \gg i$ 이다($\gg$: 시프트 연산). 나아가, 소수점 이하의 자리로 시프트되어 해당 수는 버려진다. 상기한 결과로부터 수학식 4의 첫 항이 도출된다.

[0054] 또한, 수학식 3에서는 소수부의 값을 반올림한다. 반올림 연산은 이진수 소수의 첫 번째 자리가 1인 경우에는

정수부에 1을 더하고 이진수 소수 첫 번째 자리가 0인 경우에는 이하의 값을 버린다. 따라서, 소수점 이하로 시프트 되기 전의 값과 비트 와이즈 연산을 수행하여 반올림 연산을 수행할 수 있다.

[0055] 이와 같이 수행하여 1의 개수를 구할 수 있다.

[0056] 일 예로, $x = \{1100\}_{bin}$ 이고, 윈도우 너비(width)가 $2^{dec} = 0010_{bin}$ 이라고 하자. 수학적 3을 연산하면 아래의 표 2와 같다.

표 2

	정수부	소수부
i=1	$x_{N-1} \times (w \gg 1) = 1 \times (0001)_{bin} = 1$	$x_{N-1} \wedge w_0 = 1 \wedge 0 = 0$
i=2	$x_{N-2} \times (w \gg 2) = 1 \times 0 = 0$	$x_{N-2} \wedge w_1 = 1 \wedge 1 = 1$
i=3	$x_{N-3} \times (w \gg 3) = 0$	$x_{N-3} \wedge w_2 = 0 \wedge 0 = 0$
i=4	$x_{N-4} \times (w \gg 4) = 0$	$x_{N-4} \wedge w_3 = 0 \wedge 0 = 0$
	...	...

[0057]

[0058] 따라서, 윈도우 너비(width)에 상응하는 i=1, i=2 경우에 정수부와 소수부의 연산 결과를 더하면 십진수 2로, x에 포함된 1의 개수와 일치한다. 비록 윈도우 너비가 2이나, 하드웨어적 구성에 따르면 i=1, i=2, i=3, i=4인 경우 모두에 대하여 상기한 연산을 수행한다.

[0059] 다만, 윈도우 너비(width)는 로그를 선형으로 변환한 값으로부터 얻어진 수로, 윈도우 너비(width)를 표현하는 이진수 값에는 오직 하나의 1 값만이 존재한다. 위의 예와 같이, $x = \{1100\}_{bin}$ 이고, 윈도우 너비가 $2^{dec} = 0010_{bin}$ 이라고 하면 윈도우 너비(width) 내에서 1의 위치를 p라 하면, p=1 이다. 이로부터 수학적 4는 아래의 수학적 5와 같이 간단하게 정리된다.

수학적 5

$$\sum_{i=1}^n x_{N-i} \times (2^p \gg i) + x_{N-p-1}$$

[0060]

[0061] 이러한 연산 결과를 보면 아래의 표 3과 같다.

표 3

	정수부	소수부
i=1	$x_{N-1} \times (2^p \gg 1) = 1 \times (0001)_{bin} = 1$	$x_{N-p-1} = x_2 = 1$
i=2	$x_{N-2} \times (2^p \gg 2) = 1 \times 0 = 0$	
i=3	$x_{N-3} \times (2^p \gg 3) = 0$	
i=4	$x_{N-4} \times (2^p \gg 4) = 0$	
	...	...

[0062]

[0063] 비록 윈도우 너비가 2이나, 하드웨어적 구성에 따르면 i=1, i=2, i=3, i=4인 경우 모두에 대하여 상기한 연산을 수행한다.

[0064] 정수부와 소수부의 연산 결과를 더하면 십진수 2로, x에 포함된 1의 개수와 일치한다.

[0065] 도 4는 1 계수기(130)의 개요를 도시한 도면이다. 도 4로 예시된 1 계수기(130)는 수학식 2와 수학식 5를 하드웨어로 구현한 예시이다. 도 4를 참조하면, 도 4에서 굵은 실선으로 도시된 경로는 수학식 1의 연산을 수행하는 경로이다. 윈도우 너비(width)가 2^n 보다 크면 시프터(132)는 입력된 신호를 시프트하지 않고 그대로 출력하고, 다중화기(MUX)는 입력된 $(x_{N-1}, \dots, x_0)$ 중에서 블록에 포함하는 비트를 선택하여 출력한다.

[0066] 일 실시예로, 도 3에서 제1 블록(b1)에 포함된 1의 개수를 계수하는 것을 예시된 실시예에서, $(x_3 \ x_2 \ x_3)_{bin}$ 은 시프터(132)를 거치나, 비트가 시프트되지 않고 덧셈기(134)에 제공된다. 제1 블록(b1)의 x1 비트는 다중화기(MUX)에서 선택되어 덧셈기(134)에 출력된다. 덧셈기(134)는 입력으로 제공된 값을 더하여 출력한다.

[0067] 도 4에서 점선으로 도시된 경로는 수학식 5을 연산하는 경로이다. $(x_{N-1}, \dots, x_{N-n})$ 은 시프터(132)로 입력되며, 윈도우 너비(width)에서 1의 위치(p)를 나타내는 신호(p)에 따라 순차적으로 시프트되어 덧셈기(134)에 제공된

다. 도시된 경로에 따라 덧셈기(134)에 제공된 신호는 $\sum_{i=1}^n x_{N-i} \times (2^p \gg i)$ 에 상응한다. $(x_{N-1}, \dots, x_{N-n})$ 에 포함된 비트들 중에서 다중화기(MUX)는 1의 위치(p)를 나타내는 신호(p)를 제공받고, x_{N-p-1} 에 상응하는 비트를 선택하여 덧셈기(134)에 출력한다. 덧셈기(134)는 두 경로를 통해 입력된 값들을 더하여 출력한다.

[0068] 도 4에서 도시되지는 않았지만, 시프터(132)는 도시되지 않은 제어부로부터 시프트 신호를 제공받는다. 시프터(132)는 시프트 신호에 따라 입력 신호가 시프트되는 비트 수가 제어된다. 다중화기(MUX)는 도시되지 않은 제어부로부터 선택 신호를 제공받는다. 다중화기(MUX)는 선택 신호에 따라 입력된 신호 중 어느 하나를 선택하여 출력한다.

[0069] 도 5은 본 실시예에 의한 곱셈 및 누적기(100)를 포함하는 가속기(10)의 개요를 도시한 도면이다. 도 5를 참조하면, 본 실시예에 의한 가속기(10)는 입력 벡터들을 제공하는 입력 제공부(210)를 포함하며, 로그로 양자화된 가중치들을 제공하는 가중치 제공부(250)를 포함한다.

[0070] 가중치 제공부(210)는 각 입력 벡터의 성분(x_a, x_b, x_c, x_d)을 스토캐스틱 수 제너레이터(120a, 120b, 120c, 120d)에 출력한다. 스토캐스틱 수 제너레이터들은 입력된 성분(x_a, x_b, x_c, x_d)에 상응하는 스토캐스틱 수의 비트 스트림을 형성하여 1 계수기(130a, 130b, 130c, 130d)에 제공한다.

[0071] 가중치 제공부(250)가 제공하는 로그 양자화된 가중치 값은 각 성분별로 공통되고, 로그 대 선형 변환기(160)에 제공되며, 역계수기(170)는 윈도우 너비를 형성하여 각각의 1 계수기(130a, 130b, 130c, 130d)에 출력한다. 각각의 1 계수기(130a, 130b, 130c, 130d)는 각 비트 스트림의 제공된 윈도우 내에서 1을 계수하고, 각 누적기

(140a, 140b, 140c, 140d)에 출력하며, 각 누적기(140a, 140b, 140c, 140d)들은 제공된 값들을 누적한다. 각 XOR 게이트는 가중치의 부호와 각 입력 벡터 성분(xa, xb, xc, xd)의 부호 비트로부터 곱셈 결과를 각 누적기(140a, 140b, 140c, 140d)에 출력하며, 각 누적기(140a, 140b, 140c, 140d)는 1 계수기(130a, 130b, 130c, 130d)가 제공한 결과에 부호를 부가하여 누적하여 결과를 출력한다.

[0072] **로그 양자화된 SC 곱셈 정확성 분석**

[0073] 도 6은 스토캐스틱 컴퓨팅 가속기의 평균 제곱 오차(mean square error)로 표시된 곱셈 오차를 도시한다. 부호 비트를 포함하는 균일하게 분포된 16비트의 소수와 Alexnet의 제1 레이어의 가중치에 곱하였다. 양자화 된 곱셈 오류의 경우, 실제 곱셈 출력은 부동 소수점 가중치를 곱하여 계산하였다. 이에 비하여 선형으로 양자화된 가중치 또는 선형 또는 로그 양자화된 가중치들은 곱셈 오차를 연산하기 위하여 실제 출력으로 곱해졌다.

[0074] 선형 양자화된 스토캐스틱 컴퓨팅 가속기에서, 양자화된 곱셈 에러는 곱셈 에러와 유사하며, 이것은 선형 양자화된 경우의 곱셈 오류는 주로 잘못된 곱셈 자체에 의한 것임을 의미한다. 그러나, 본 실시예의 경우, 오류가 주로 로그 양자화 오류에서 발생했음을 나타내며, 곱셈 에러는 선형 양자화 된 경우보다 낮으며, 로그로 표현된 곱해지는 수(multiplicand)에 대한 고정 소수점 곱셈인 시프터 기반의 곱셈의 정확도에 이른다. 로그로 양자화된 본 실시예 경우에 양자화된 곱셈 오류는 처음에는 감소하나, 높은 데이터 정밀도에 의하여 양자화 오류를 개선할 수 없기 때문에 포화되기 시작한다. 예를 들어, 데이터 정밀도가 증가하더라도 0.7은 항상 0.5로 양자화된다.

[0075] 작은 비트 폭(bitwidth) 양자화 방법은 일반적으로 원래 네트워크 정확도를 복원하기 위해 재교육 또는 미세 조정이 필요하다. 스토캐스틱 컴퓨팅 MAC을 포함하는 근사가속기는 미세 조정으로 더 높은 네트워크 정확도를 달성할 수 있다. 네트워크 정확도는 곱셈 오류보다 로그 양자화 오류에 대해 더 탄력적이므로, 최신 SC와 비교하여 대수 양자화 된 SC에 대한 정밀도 요구 사항이 거의 동일하게 유지될 수 있다

[0076] **면적 및 전력 소모의 효율성 평가**

[0077] 모든 비교 사례에 대해 256 개의 곱셈 및 누적기로 구성된 가속기를 평가하였다. 고정 소수점 이진 설계에 기반한 가속기(Fixed), 시프터 기반 가속기, 직렬 가속기, 비트 병렬 가속기들이 비교 대상 기술로, 로그 양자화를 이용하는 본 실시예에 의한 가속기와 비교하였다. 본 실시예에 의한 가속기는 로그 양자화 된 가중치를 사용하는 곱셈 및 누적기에 있어서 병목 현상이 경감된 1 계수기를 포함한다. 가속기는 Verilog RTL 코드로 구현하였으며, Synopsys Design Compiler를 사용하여 합성되었다. 대상 표준 셀 라이브러리는 TSMC 45nm이며 대상 주파수는 1GHz로 고정되었다. 일반적인 데이터 정밀도는 16 비트이다. 비트 병렬 설계가 있으며 이 평가를 위해 2^4 사이클로 병렬화 되었다(하드웨어 정밀도는 4).

[0078] 도 7(a)는 가속기의 면적, 도 5(b)는 전력 소모를 비교 도시한 표이다. 도 7(a)를 참조하면, 고정 소수점 이진 설계(Fixed)에는 전체 면적의 80 % 이상을 차지하는 곱셈기를 포함한다. 누적기의 면적(Cntr/Acc., 직렬 SC의 카운터)은 비교 대상 기술의 경우에 유사하다. 시프터 기반 가속기는 고정 소수점 이진 설계(Fixed)대비 여러 배 더 작다.

[0079] 직렬 가속기의 면적은 카운터가 90% 가량 차지한다. 비트 병렬 가속기의 면적은 큰 1s 카운터에 의하여 거의 두 배에 인접하나, 시프터 기반 가속기의 면적(Shift)보다 작다. 본 실시예에서 1 계수기의 면적은 약 1/5이며 총 면적은 누적기의 면적(Cnt/Acc)에 좌우된다. 결과적으로, 면적은 비트 병렬 가속기에 비해 40 % 감소하였다.

[0080] 전력 소모를 비교하여 도시한 도 5b를 참조하면, 전력 소모는 면적 소모와 유사한 경향을 보인다. 본 실시예의 전력 소모는 고정 소수점 이진 설계(Fixed) 기반 가속기 및 시프터 기반 가속기의 경우에 비하여 각각 84% 및 39% 낮다. 비트 병렬 가속기와 비교하면, 본 실시예의 전력 소모는 비교 기술에 비하여 24% 낮으나, 직렬 가속기에 비하여는 17% 높다.

[0081] **로그 양자화된 스토캐스틱 컴퓨팅의 DNN 적용의 효율성**

[0082] 로그 양자화는 원래 값에 비하여 자연적으로 오차가 형성되므로, 명시적으로 허용되는 수준의 뉴럴 네트워크 성능이 보장되어야 한다. 카페 프레임워크(Caffe framework)를 확장한 시뮬레이션으로 이미지 분류 테스트를 수행하였다. 목표 이미지 분류 대부분의 연산을 수행하는 컨벌루션 레이어만이 가속되었다. 대상 DNN은 10 클래스 데이터 세트인 MNIST 및 CIFAR10 용 LeNet 및 AlexNet이다. 대규모 AlexNet은 1000개 클래스들의 LSVRC-2010 이미지넷 데이터세트에 의하여 실험되었다. 부동 소수점 학습 가중치는 공통적으로 5,000회 반복 미세 조정되었다.

[0083] 도 8은 5000회 반복하여 미세 조정된 대상 신경 네트워크에 인식 테스트 결과를 도시한 도면으로, 도 8(a)는 10-클래스 이미지 분류 결과이고, 도 8(b)는 1000 클래스 이미지 분류 결과이다. 도 8(a)를 참조하면, 로그 양자화를 구현한 본 실시예는 비교 대상 기술들에 대하여 전반적으로 더 나은 인식률을 보인다(5 비트 및 9 비트는 엄격한 오류 제한 조건(12% 손실)에 대한 일반적 요구 사항임). 이것은 로그 양자화 가중치에 대해 양자화 오류가 더 크더라도 가중치를 미세 조정하여 인식률을 복구할 수 있음을 의미한다. 나아가, 미세 조정에서 곱셈 오류는 양자화 오류보다 더 큰 문제점이다. 이러한 현상은 도 8(b)로 도시된 더 크고 어려운 뉴럴 네트워크에서 더욱 향상되었다. 종래 기술의 SC는 10비트 정밀도에서 학습을 시작하고 로그 양자화의 본 실시예는 7비트에서 시작한다. 정확도 포화는 선형 양자화 된 이진 고정 소수점과 유사하게 8 비트에서 시작한다. 본 실시예는 더 작은 파라미터 크기 및 더 큰 양자화 에러에도 불구하고 선형 양자화 비교예보다 더 정확한 것으로 파악된다.

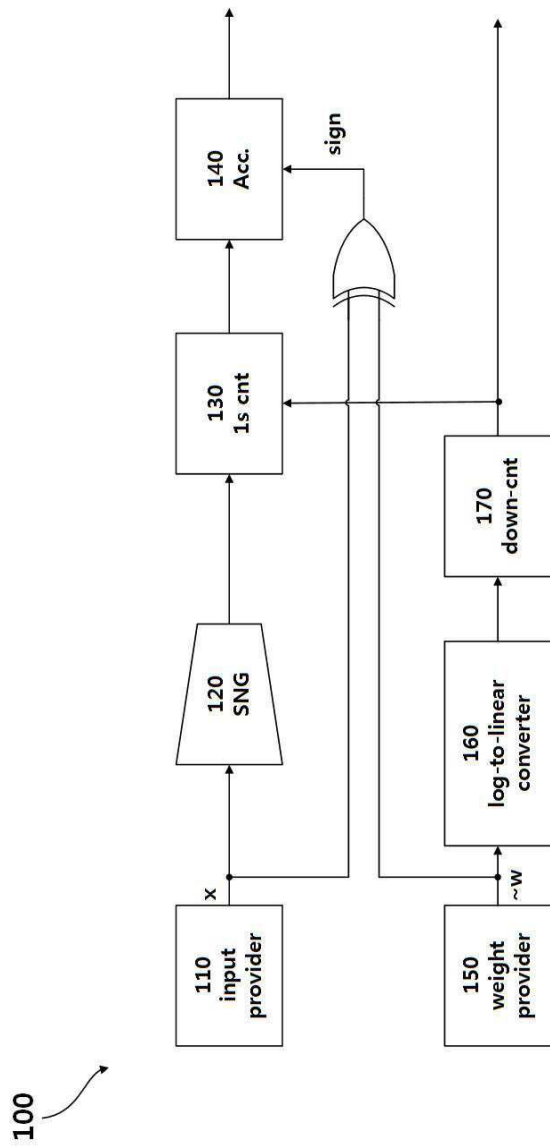
[0084] 그러나, 종래 기술의 스토캐스틱 컴퓨팅은 부호 비트 처리를 제외하고 동일한 입력에 대해 본 실시예와 동일하게 작동하므로 종래 기술의 스토캐스틱 컴퓨팅의 부호 크기 설계는 미세 조정된 가중치를 사용하여 본 실시예와 동일한 정확도를 달성할 수 있다. 따라서 본 실시예가 최종적으로 정밀 요구 사항 측면에서 종래 기술의 스토캐스틱 컴퓨팅을 능가한다고 말할 수는 없지만 본 실시예는 미세 조정 성능을 증가하며 종래 기술에 비해 인식률 저하가 없다.

부호의 설명

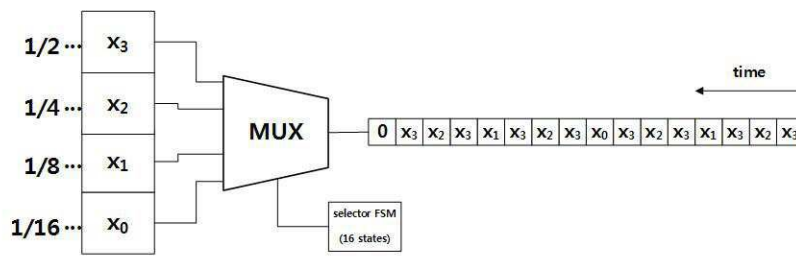
[0085] 10: 100: 곱셈 및 누적기
110: 입력 제공부 120: 스토캐스틱 수 제너레이터
130: 1 계수기 132: 시프터(shiftter)
134: 덧셈기 140: 누적기
150: 가중치 제공부 160: 로그 대 선형 변환기
170: 역계수기

도면

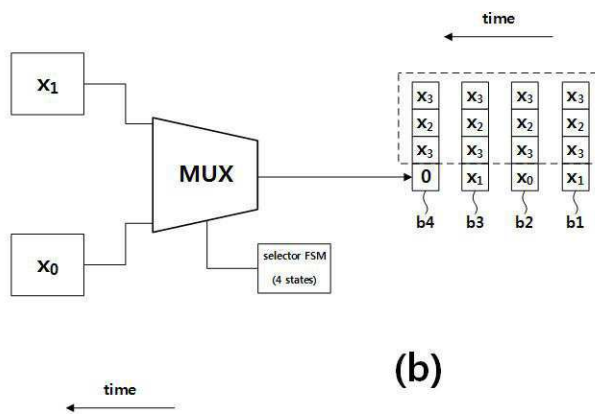
도면1



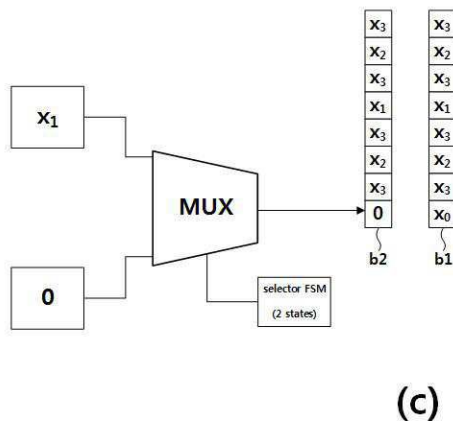
도면2



(a)

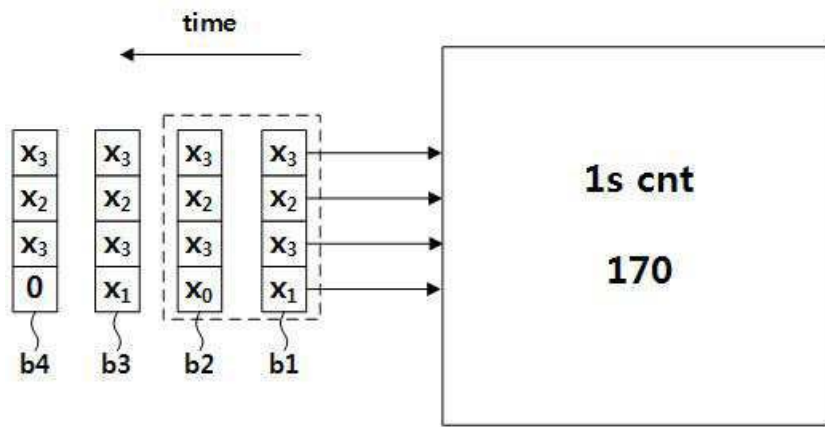


(b)

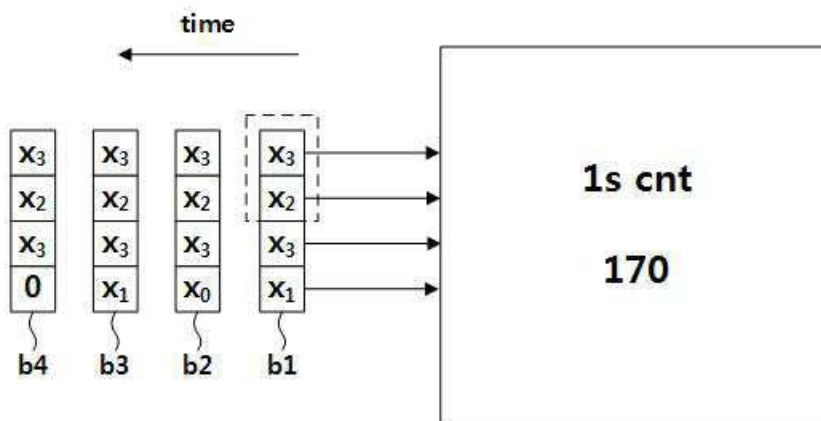


(c)

도면3



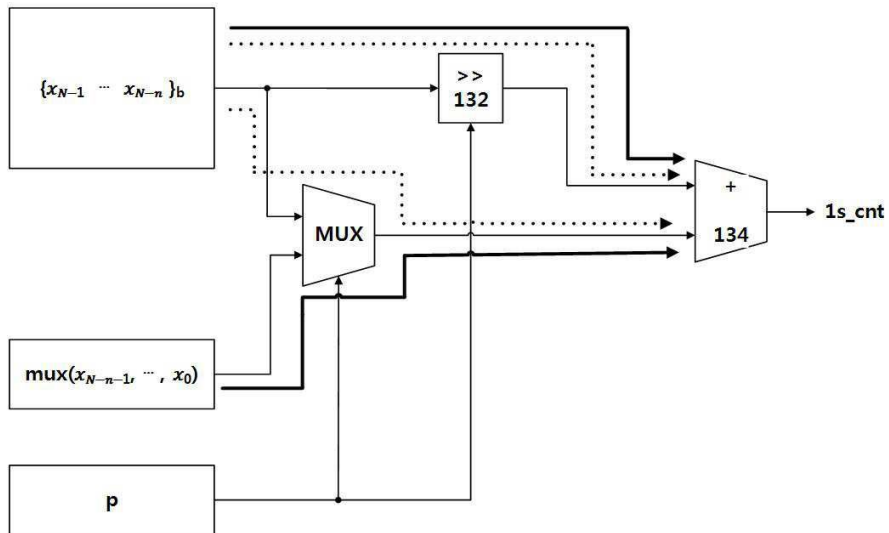
(a)



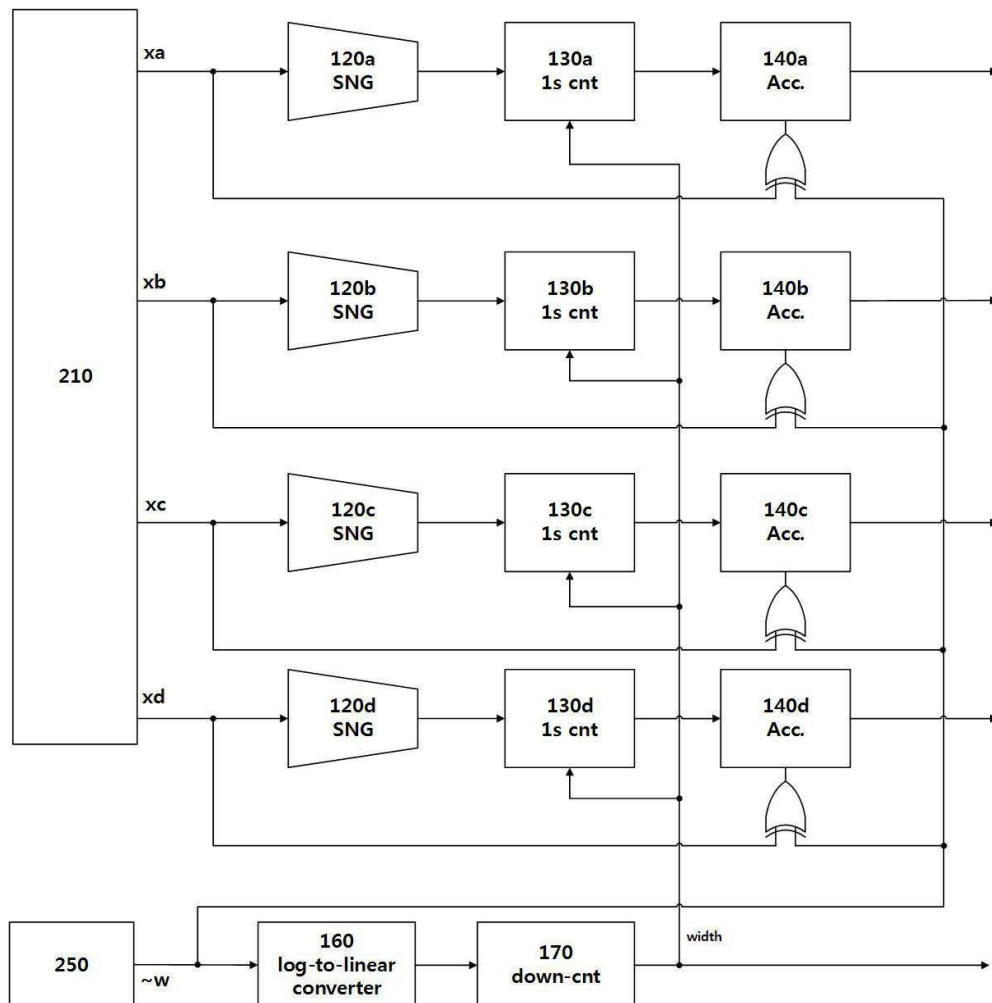
(b)

도면4

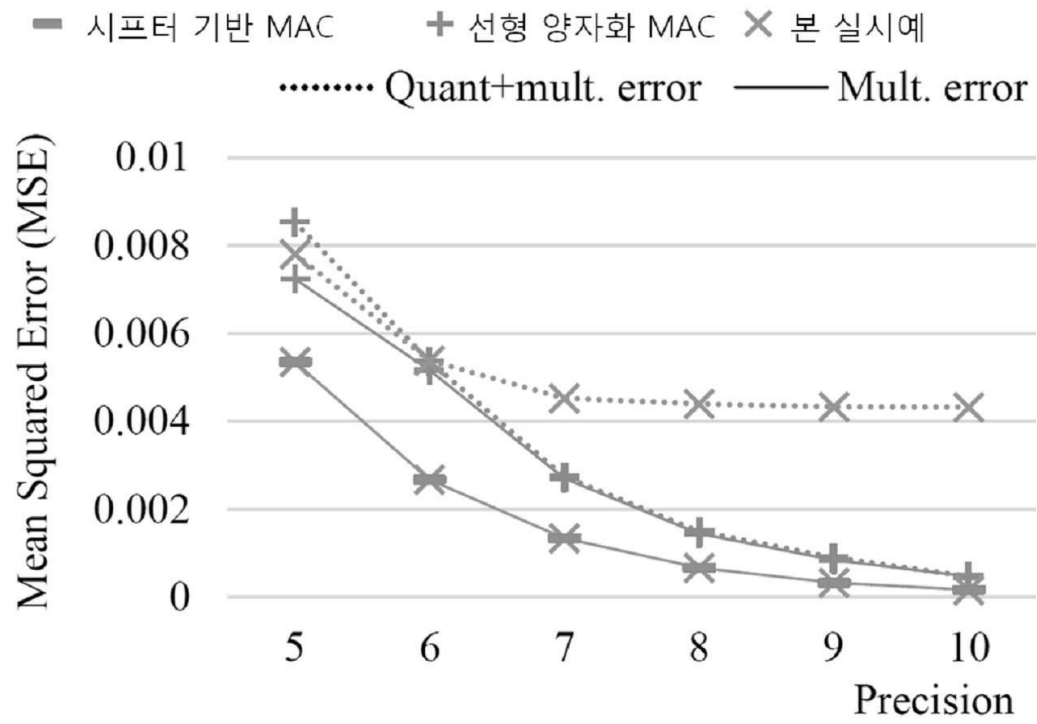
130



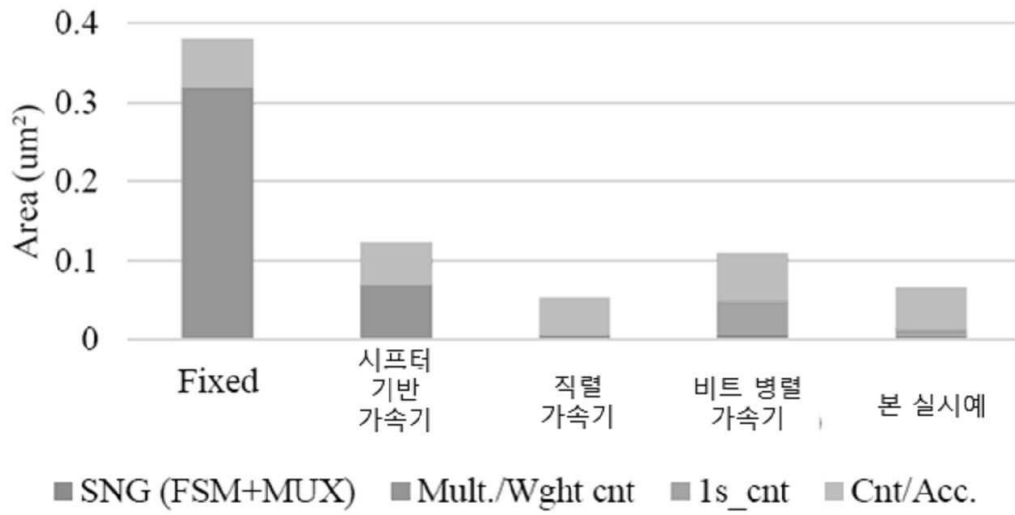
도면5



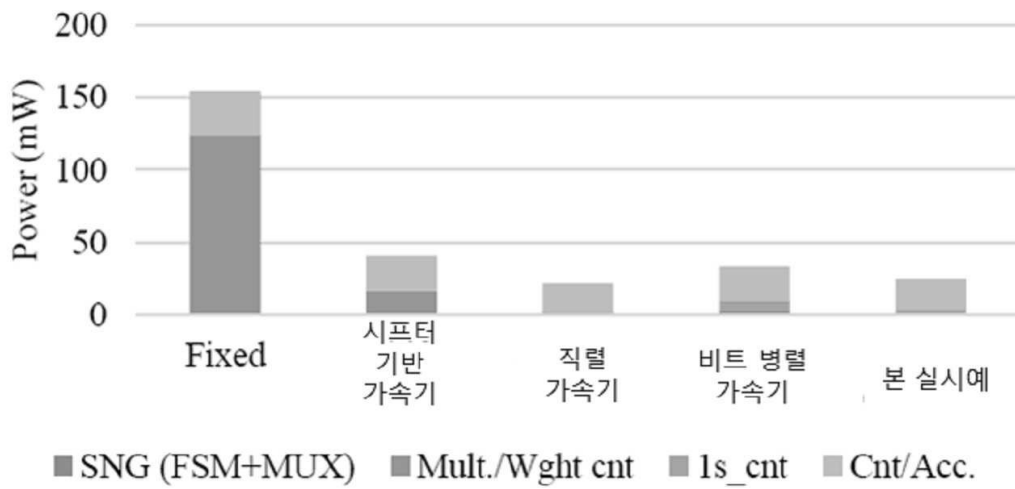
도면6



도면7



(a)



(b)

도면8

