(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년04월11일
(11) 등록번호 10-2385843
(24) 등록일자 2022년04월07일

(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 3/0454 (2013.01)
(21) 출원번호 10-2019-7016416
(22) 출원일자(국제) 2017년08월23일
심사청구일자 2019년06월10일
(85) 번역문제출일자 2019년06월07일
(65) 공개번호 10-2019-0084088
(43) 공개일자 2019년07월15일
(86) 국제출원번호 PCT/US2017/048123
(87) 국제공개번호 WO 2018/089079
국제공개일자 2018년05월17일
(30) 우선권주장
15/348,199 2016년11월10일 미국(US)
15/467,382 2017년03월23일 미국(US)

(73) 특허권자
구글 엘엘씨
미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이 1600 (우:94043)
(72) 발명자
영, 레지널드 클리포드
미국 94043 캘리포니아 마운틴 뷰 엠피씨어터 파크웨이 1600
굴랜드, 윌리엄 존
미국 94043 캘리포니아 마운틴 뷰 엠피씨어터 파크웨이 1600
(74) 대리인
특허법인 남앤남

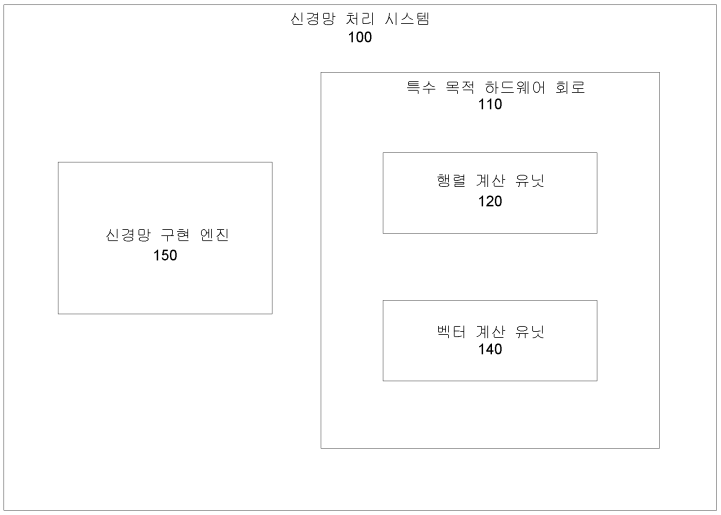
전체 청구항 수 : 총 12 항 심사관 : 양대경

(54) 발명의 명칭 **하드웨어에서의 커널 스트라이딩 수행**

(57) 요 약

1보다 더 큰 스트라이드를 갖는 제1 컨볼루션 신경망 계층을 포함하는 신경망을 하드웨어 회로 상에서 처리하라는 요청을 수신하고, 그리고 이에 대응하여, 하드웨어 회로로 하여금, 제1 텐서를 생성하기 위해, 1과 같은 스트라이드를 갖지만 그 외에는 제1 컨볼루션 신경망 계층과 동일한 제2 컨볼루션 신경망 계층을 사용하여 입력 텐서를 처리하고; 제2 텐서를 생성하기 위해, 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트들을 0으로 설정하고; 그리고 계층 출력 텐서를 생성하기 위해, 제2 텐서에 대한 최대 풀링을 수행함으로써, 입력 텐서의 처리 동안, 제1 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서를 생성하게 하는 명령들을 생성하기 위한 방법이 개시된다.

대 표 도 - 도1



(52) CPC특허분류
G06N 3/08 (2013.01)

명 세 서

청구범위

청구항 1

컴퓨터로 구현되는 방법으로서,

특수 목적 하드웨어 회로 상에 컨볼루션 신경망을 구현하라는 요청을 수신하고, 그리고 상기 하드웨어 회로로 하여금 명령들을 실행하게 함으로써 상기 신경망을 사용하여 신경망 입력들을 수신 및 처리하는 단계 - 상기 신경망은 1보다 더 큰 스트라이드(stride)를 갖는 제1 컨볼루션 신경망 계층을 포함하고, 상기 하드웨어 회로는 신경망 계산들을 수행하기 위한 집적 회로이고, 벡터 행렬 곱셈들을 수행하도록 구성된 행렬 계산 유닛, 및 벡터 계산 유닛을 포함하며, 상기 벡터 계산 유닛은 상기 행렬 계산 유닛의 출력들에 대한 풀링(pool)을 수행하도록 구성된 풀링 회로를 포함함 -; 및

이에 대응하여, 상기 하드웨어 회로에 의해 실행될 때, 상기 신경망에 의한 입력 텐서(tensor)의 처리 동안, 상기 하드웨어 회로로 하여금 상기 제1 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서를 생성하게 하는 명령들을 생성하는 단계를 포함하고,

상기 제1 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서는,

상기 행렬 계산 유닛이 제1 텐서를 생성하기 위해, 1의 스트라이드를 갖지만 그 외에는 상기 제1 컨볼루션 신경망 계층과 동일한 제2 컨볼루션 신경망 계층을 사용하여, 상기 제1 컨볼루션 신경망 계층에 대한 상기 입력 텐서를 처리하는 동작;

상기 벡터 계산 유닛이 제2 텐서를 생성하기 위해, 상기 제2 컨볼루션 신경망 계층이 상기 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 상기 제1 텐서의 엘리먼트들을 0으로 설정(zero out)하는 동작; 및

상기 벡터 계산 유닛의 풀링 회로가 계층 출력 텐서를 생성하기 위해, 상기 제2 텐서에 대한 최대 풀링을 수행하는 동작

을 포함하는 동작들을 수행함으로써 생성되고,

상기 벡터 계산 유닛이 상기 제1 텐서의 엘리먼트들을 0으로 설정하는 동작은, 상기 벡터 계산 유닛이 상기 제2 텐서를 생성하기 위해, 마스킹 텐서와 상기 제1 텐서의 엘리먼트별 곱셈을 수행하는 동작을 포함하며,

상기 마스킹 텐서는 (i) 상기 제2 컨볼루션 신경망 계층이 상기 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 상기 제1 텐서의 엘리먼트에 대응하는 상기 마스킹 텐서의 각각의 엘리먼트 위치에서 0들을 포함하고, 그리고 (ii) 상기 마스킹 텐서의 각각의 다른 엘리먼트 위치에서 1들을 포함하는,

컴퓨터로 구현되는 방법.

청구항 2

제1항에 있어서,

상기 벡터 계산 유닛이 상기 제1 텐서의 엘리먼트들을 0으로 설정하는 동작은,

상기 제1 텐서의 엘리먼트들의 서브세트를 0으로 곱하는 동작; 및

상기 서브세트에 포함되지 않은 상기 제1 텐서의 엘리먼트들을 1로 곱하는 동작

을 포함하는,

컴퓨터로 구현되는 방법.

청구항 3

컴퓨터로 구현되는 방법으로서,

특수 목적 하드웨어 회로 상에 컨볼루션 신경망을 구현하라는 요청을 수신하고, 그리고 상기 하드웨어 회로로 하여금 명령들을 실행하게 함으로써 상기 신경망을 사용하여 신경망 입력들을 수신 및 처리하는 단계 - 상기 신경망은 1보다 더 큰 스트라이드를 갖는 제1 컨볼루션 신경망 계층을 포함하고, 상기 하드웨어 회로는 신경망 계산들을 수행하기 위한 집적 회로이고, 벡터 행렬 곱셈들을 수행하도록 구성된 행렬 계산 유닛, 및 벡터 계산 유닛을 포함하며, 상기 벡터 계산 유닛은 상기 행렬 계산 유닛의 출력들에 대한 풀링을 수행하도록 구성된 풀링 회로를 포함함 -; 및

이에 대응하여, 상기 하드웨어 회로에 의해 실행될 때, 상기 신경망에 의한 입력 텐서의 처리 동안, 상기 하드웨어 회로로 하여금 상기 제1 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서를 생성하게 하는 명령들을 생성하는 단계를 포함하고,

상기 제1 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서는,

상기 행렬 계산 유닛이 제1 텐서를 생성하기 위해, 1의 스트라이드를 갖지만 그 외에는 상기 제1 컨볼루션 신경망 계층과 동일한 제2 컨볼루션 신경망 계층을 사용하여, 상기 제1 컨볼루션 신경망 계층에 대한 상기 입력 텐서를 처리하는 동작;

상기 벡터 계산 유닛이 제2 텐서를 생성하기 위해, 상기 제2 컨볼루션 신경망 계층이 상기 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 상기 제1 텐서의 엘리먼트들을 0으로 설정하는 동작; 및

상기 벡터 계산 유닛의 풀링 회로가 계층 출력 텐서를 생성하기 위해, 상기 제2 텐서에 대한 최대 풀링을 수행하는 동작

을 포함하는 동작들을 수행함으로써 생성되고,

상기 벡터 계산 유닛이 상기 제1 텐서의 엘리먼트들을 0으로 설정하는 동작은,

상기 벡터 계산 유닛이 수정된 제1 텐서를 생성하기 위해, 제1 마스킹 텐서와 상기 제1 텐서의 엘리먼트별 곱셈을 수행하는 동작 - 상기 제1 마스킹 텐서는 (ⅰ) 상기 제2 컨볼루션 신경망 계층이 상기 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 상기 제1 텐서의 엘리먼트에 대응하는 상기 마스킹 텐서의 각각의 엘리먼트 위치에서 0들을 포함하고, 그리고 (ii) 상기 제2 컨볼루션 신경망 계층이 상기 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되었을 상기 제1 텐서의 엘리먼트에 대응하는 상기 마스킹 텐서의 각각의 엘리먼트 위치에서 각각 0이 아닌 값을 포함함 -; 및

상기 벡터 계산 유닛이 제2 마스킹 텐서와 상기 수정된 제1 텐서의 엘리먼트별 곱셈을 수행하는 동작

을 포함하며,

상기 제2 마스킹 텐서는, 상기 제2 컨볼루션 신경망 계층이 상기 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되었을 상기 제1 텐서의 엘리먼트에 대응하는 각각의 엘리먼트 위치에서의 상기 제1 마스킹 텐서의 각각의 0이 아닌 값의 역(inverse)을 포함하는,

컴퓨터로 구현되는 방법.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 마스킹 텐서는 상기 하드웨어 회로에 의해 액세스 가능한 메모리에 저장되는,

컴퓨터로 구현되는 방법.

청구항 5

제4항에 있어서,

복수의 마스킹 텐서들이 각각 1보다 더 큰 복수의 스트라이드들에 대응하도록 상기 메모리에 저장되고,

상기 방법은 상기 복수의 마스킹 텐서들 중에서 상기 제1 컨볼루션 신경망 계층의 스트라이드에 대응하는 마스킹 텐서를 선택하는 단계를 더 포함하는,

컴퓨터로 구현되는 방법.

청구항 6

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 벡터 계산 유닛의 풀링 회로가 최대 풀링을 수행하는 동작은, 상기 제1 컨볼루션 신경망 계층의 스트라이드에 의해 정의되는 상기 제2 텐서의 하나 이상의 윈도우들 각각에 대해 상기 윈도우 내의 엘리먼트들 중 최대 값 엘리먼트를 획득하는 동작을 포함하는,

컴퓨터로 구현되는 방법.

청구항 7

제6항에 있어서,

상기 제2 텐서의 하나 이상의 윈도우들 각각은 상기 컨볼루션 신경망 계층의 스트라이드에 대응하는 치수들을 갖는 직사각형 윈도우이고, 상기 제2 텐서의 서로 다른 엘리먼트들을 포함하는,

컴퓨터로 구현되는 방법.

청구항 8

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 벡터 계산 유닛의 풀링 회로가 최대 풀링을 수행하는 동작은, 상기 제2 텐서의 엘리먼트들의 하나 이상의 서브세트들 각각에 대해 상기 서브세트의 최대 값 엘리먼트를 획득하는 동작을 포함하는,

컴퓨터로 구현되는 방법.

청구항 9

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 컨볼루션 신경망 계층은 상기 컨볼루션 신경망에서의 제1 신경망 계층이고,

상기 입력 텐서는 디지털 이미지의 픽셀들에 대응하는 엘리먼트들을 포함하는 상기 디지털 이미지의 표현인,

컴퓨터로 구현되는 방법.

청구항 10

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 입력 텐서는 상기 하드웨어 회로의 통합 버퍼에 저장되고,

상기 제2 컨볼루션 신경망 계층의 가중치들은 상기 하드웨어 회로의 동적 메모리에 저장되며,

상기 제2 컨볼루션 신경망 계층을 사용하여 상기 제1 컨볼루션 신경망 계층에 대한 입력 텐서를 처리하는 동작은,

상기 통합 버퍼로부터 상기 행렬 계산 유닛으로 상기 입력 텐서를 전송하는 동작;

상기 동적 메모리로부터 상기 행렬 계산 유닛으로 상기 제2 컨볼루션 신경망 계층의 가중치들을 전송하는 동작; 및

상기 제1 텐서를 생성하기 위해, 상기 행렬 계산 유닛에 의해 상기 제2 컨볼루션 신경망 계층의 가중치들을 사용하여 상기 입력 텐서를 처리하는 동작

을 포함하는,

컴퓨터로 구현되는 방법.

청구항 11

시스템으로서,

신경망 계산들을 수행하기 위한 집적 회로인 특수 목적 하드웨어 회로 — 상기 특수 목적 하드웨어 회로는 벡터 행렬 곱셈들을 수행하도록 구성된 행렬 계산 유닛 및 벡터 계산 유닛을 포함하고, 상기 벡터 계산 유닛은 상기 행렬 계산 유닛의 출력들에 대한 풀링을 수행하도록 구성된 풀링 회로를 포함함 —; 및

상기 하드웨어 회로에 의해 실행될 때, 상기 하드웨어 회로로 하여금 제1항 내지 제3항 중 어느 한 항에 따른 방법을 수행하게 하도록 동작 가능한 명령들을 저장하는 하나 이상의 저장 디바이스들을 포함하는,

시스템.

청구항 12

컴퓨터 프로그램이 인코딩된 컴퓨터 판독가능 저장 매체로서,

상기 컴퓨터 프로그램은, 하나 이상의 컴퓨터들에 의해 실행될 때, 상기 하나 이상의 컴퓨터들로 하여금 제1항 내지 제3항 중 어느 한 항에 따른 방법을 수행하게 하는 명령들을 포함하는,

컴퓨터 판독가능 저장 매체.

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

발명의 설명

기 술 분 야

[0001] 본 명세서는 하드웨어에서의 신경망 추론들의 계산에 관한 것이다.

배 경 기 술

[0002] 신경망들은 하나 이상의 계층들을 이용하여, 수신된 입력에 대한 출력, 예컨대 분류를 생성하는 기계 학습 모델들이다. 일부 신경망들은 출력 계층 외에도 하나 이상의 숨겨진 계층들을 포함한다. 각각의 숨겨진 계층의 출력은 망의 다른 계층, 예컨대 다음 숨겨진 계층 또는 망의 출력 계층에 대한 입력으로서 사용된다. 망의 각각

의 계층은 각각의 세트의 파라미터들의 현재 값들에 따라, 수신된 입력으로부터 출력을 생성한다.

발명의 내용

[0003] 일반적으로, 본 명세서는 신경망 추론들을 계산하는 특수 목적 하드웨어 회로를 설명한다.

[0004] 일반적으로, 본 명세서에서 설명되는 청구 대상의 하나의 혁신적인 양상은, 하드웨어 회로 상에서 신경망을 처리하라는 요청을 수신하고 - 신경망은 1보다 더 큰 스트라이드(stride)를 갖는 제1 컨볼루션 신경망 계층을 포함함 -, 그리고 응답으로, 하드웨어 회로에 의해 실행될 때 하드웨어 회로로 하여금, 제1 텐서(tensor)를 생성하기 위해, 1과 같은 스트라이드를 갖지만 그 외에는 제1 컨볼루션 신경망 계층과 동일한 제2 컨볼루션 신경망 계층을 사용하여 제1 컨볼루션 신경망 계층에 대한 입력 텐서를 처리하는 동작, 제2 텐서를 생성하기 위해, 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트들을 0으로 설정(zero out)하는 동작, 및 계층 출력 텐서를 생성하기 위해, 제2 텐서에 대한 최대 풀링(pool)을 수행하는 동작을 포함하는 동작들을 수행함으로써, 신경망에 의한 입력 텐서의 처리 동안, 제1 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서를 생성하게 하는 명령들을 생성하기 위한 방법들 및 시스템들로 구현될 수 있다.

[0005] 구현들은 다음 특징들 중 하나 이상의 특징을 포함할 수 있다. 일부 구현들에서, 제1 텐서의 엘리먼트들을 0으로 설정하는 것은, 제1 텐서의 엘리먼트들의 서브세트를 0과 곱하는 것, 그리고 서브세트에 포함되지 않은, 제1 텐서의 엘리먼트들을 1과 곱하는 것을 포함한다. 제1 텐서의 엘리먼트들을 0으로 설정하는 것은 제2 텐서를 생성하기 위해, 마스킹 텐서와 제1 텐서의 엘리먼트별 곱셈을 수행하는 것을 포함하며, 마스킹 텐서는 (i) 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트에 대응하는 마스킹 텐서의 각각의 엘리먼트 위치에서 0들을, 그리고 (ii) 마스킹 텐서의 각각의 다른 엘리먼트 위치에서 1들을 포함한다. 일부 구현들에서, 마스킹 텐서는 하드웨어 회로에 의해 액세스 가능한 메모리에 저장되고, 마스킹 텐서와 제1 텐서의 엘리먼트별 곱셈은 하드웨어 회로에 포함되는 하드웨어에 구현된 벡터 계산 유닛에 의해 수행된다.

[0006] 구현들은 다음 특징들 중 하나 이상의 특징을 더 포함할 수 있다. 일부 구현들에서, 제1 텐서의 엘리먼트들을 0으로 설정하는 것은, 수정된 제1 텐서를 생성하기 위해, 제1 마스킹 텐서와 제1 텐서의 엘리먼트별 곱셈을 수행하는 것 - 제1 마스킹 텐서는 (i) 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트에 대응하는 마스킹 텐서의 각각의 엘리먼트 위치에서 0들을, 그리고 (ii) 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되었을 제1 텐서의 엘리먼트에 대응하는 마스킹 텐서의 각각의 엘리먼트 위치에서 각각 0이 아닌 값을 포함함 -, 그리고 제2 마스킹 텐서와 수정된 제1 텐서의 엘리먼트별 곱셈을 수행하는 것을 포함하며, 제2 마스킹 텐서는 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성될 제1 텐서의 엘리먼트에 대응하는 각각의 엘리먼트 위치에서의, 제1 마스킹 텐서의 각각의 0이 아닌 값의 역(inverse)을 포함한다.

[0007] 구현들은 다음 특징들 중 하나 이상의 특징을 더 포함할 수 있다. 일부 구현들에서, 최대 풀링을 수행하는 것은, 제1 컨볼루션 신경망 계층의 스트라이드에 의해 정의되는 제2 텐서의 하나 이상의 윈도우들 각각에 대해 윈도우 내의 엘리먼트들 중 최대 값 엘리먼트를 획득하는 것을 포함한다. 제2 텐서의 하나 이상의 윈도우들 각각은 컨볼루션 신경망 계층의 스트라이드에 대응하는 치수들을 갖는 직사각형 윈도우이고, 제2 텐서의 서로 다른 엘리먼트들을 포함한다. 일부 구현들에서, 최대 풀링을 수행하는 것은, 제2 텐서의 엘리먼트들의 하나 이상의 서브세트들 각각에 대해 서브세트의 최대 값 엘리먼트를 획득하는 것을 포함한다. 제2 텐서에 대해 수행된 최대 풀링은 하드웨어 회로의 풀링 회로에 의해 수행된다. 컨볼루션 신경망 계층은 신경망에서의 제1 신경망 계층이고, 입력 텐서는 디지털 이미지의 픽셀들에 대응하는 엘리먼트들을 포함하는 디지털 이미지의 표현이다.

[0008] 구현들은 다음 특징들 중 하나 이상의 특징을 더 포함할 수 있다. 일부 구현들에서, 입력 텐서는 하드웨어 회로의 통합 버퍼에 저장되고, 제2 컨볼루션 신경망 계층의 가중치들은 하드웨어 회로의 동적 메모리에 저장되며, 제2 컨볼루션 신경망 계층을 사용하여 제1 컨볼루션 신경망 계층에 대한 입력 텐서를 처리하는 것은, 통합 버퍼로부터, 하드웨어로 구현되는 하드웨어 회로의 행렬 계산 유닛으로 입력 텐서를 전송하는 것, 동적 메모리로부터 하드웨어 회로의 행렬 계산 유닛으로 제2 컨볼루션 신경망 계층의 가중치들을 전송하는 것, 그리고 제1 텐서를 생성하기 위해, 하드웨어 회로의 행렬 계산 유닛에 의해 제2 컨볼루션 신경망 계층의 가중치들을 사용하여 입력 텐서를 처리하는 것을 포함한다.

[0009] 본 명세서에서 설명되는 청구 대상의 특정 실시예들은 다음 이점들 중 하나 이상을 실현하도록 구현될 수 있다.

특수 목적 하드웨어 회로가 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망을 사용하여 입력 텐서를 직접 처리할 수 없는 경우에도, 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층에 대응하는 출력 텐서가 하드웨어 회로에 의해 하드웨어에서 생성될 수 있다. 특수 목적 하드웨어 회로를 사용하여 적절한 출력을 생성함으로써, 특수 목적 하드웨어 회로가 1보다 더 큰 스트라이드를 갖는 신경망 계층의 처리를 직접 지원하지 않더라도, 데이터를 호스트 컴퓨터로 다시 전달하지 않고, 즉 오프-칩(off-chip) 계산의 적어도 일부를 수행하지 않고, 그러한 처리가 수행될 수 있다. 이는 1보다 더 큰 스트라이드를 갖는 컨볼루션 계층을 포함하는 신경망의 추론이 특수 목적 하드웨어 회로의 하드웨어 아키텍처를 수정하지 않고 효율적으로 결정될 수 있게 한다. 즉, 오프-칩이나 소프트웨어에서 또는 둘 모두에서 처리의 일부를 수행하는 것으로부터 발생하는 처리 지연들이 회피된다.

[0010] 본 명세서에서 설명되는 청구 대상은 또한 예를 들어, 신경망 추론들을 계산할 때 커널 스트라이딩을 수행하기 위해 개시된 기술들 및 하드웨어를 사용하는 이미지 인식 또는 분류 방법 및 시스템에 관한 것이다.

[0011] 본 명세서의 청구 대상의 하나 이상의 실시예들의 세부사항들은 아래 첨부 도면들 및 설명에서 제시된다. 청구 대상의 다른 특징들, 양상들 및 이점들은 설명, 도면들 및 청구항들로부터 자명해질 것이다.

도면의 간단한 설명

[0012] 도 1은 예시적인 신경망 처리 시스템을 도시한다.

도 2는 신경망의 주어진 계층에 대한 계산을 수행하기 위한 예시적인 방법의 흐름도이다.

도 3은 예시적인 신경망 처리 시스템을 도시한다.

도 4는 행렬 계산 유닛을 포함하는 예시적인 아키텍처를 도시한다.

도 5는 수축기 어레이 내부의 셀의 예시적인 아키텍처를 도시한다.

도 6은 벡터 계산 유닛의 예시적인 아키텍처를 도시한다.

도 7은 풀링 회로에 대한 예시적인 아키텍처를 도시한다.

도 8은 1보다 더 큰 스트라이드를 갖는 신경망의 주어진 계층에 대한 계산을 수행하도록 신경망 처리 시스템에 명령하기 위한 예시적인 방법의 흐름도이다.

도 9는 1보다 더 큰 스트라이드를 갖는 신경망의 주어진 계층에 대한 계산을 수행하기 위한 예시적인 방법의 흐름도이다.

도 10은 1보다 더 큰 스트라이드를 갖는 신경망의 주어진 계층에 대한 계산의 일례이다. 다양한 도면들에서 유사한 참조 번호들 및 지정들은 유사한 엘리먼트들을 표시한다.

발명을 실시하기 위한 구체적인 내용

[0013] 추론들을 계산하기 위해 다수의 계층들을 갖는 신경망이 사용될 수 있다. 예를 들어, 입력이 주어지면, 신경망은 입력에 대한 추론을 계산할 수 있다. 신경망은 신경망의 계층들 각각을 통해 입력을 처리함으로써 이 추론을 계산한다. 각각의 계층은 입력을 수신하고 계층에 대한 가중치들의 세트에 따라 입력을 처리하여 출력을 생성한다.

[0014] 따라서 수신된 입력으로부터의 추론을 계산하기 위해, 신경망은 입력을 수신하고 이를 신경망 계층들 각각을 통해 처리하여 추론을 생성하며, 하나의 신경망 계층으로부터의 출력은 다음 신경망 계층에 대한 입력으로서 제공된다. 신경망 계층에 대한 데이터 입력들, 예컨대 신경망에 대한 입력 또는 신경망 계층으로의 시퀀스의 계층 아래에 있는 계층의 출력들이 해당 계층에 대한 활성화 입력들로 지칭될 수 있다.

[0015] 일부 구현들에서, 신경망의 계층들은 시퀀스로 배열된다. 다른 일부 구현들에서, 계층은 방향 그래프로서 배열된다. 즉, 임의의 특정 계층은 다수의 입력들이나 다수의 출력들, 또는 이들 모두를 수신할 수 있다. 신경망의 계층들은 또한 계층의 출력이 이전 계층에 대한 입력으로서 다시 전송될 수 있도록 배열될 수 있다.

[0016] 일부 신경망들은 하나 이상의 신경망 계층들로부터의 출력들을 풀링하여, 후속 신경망 계층들에 대한 입력들로서 사용되는 풀링된 값들을 생성한다. 일부 구현들에서, 신경망은 출력들의 그룹의 최대, 최소 또는 평균을 결정하고 최대, 최소 또는 평균을 그룹에 대한 풀링된 출력으로서 사용함으로써 출력들의 그룹을 풀링한다. 출력들을 풀링하는 것은 어떤 공간적으로 불변성을 유지할 수 있어, 다양한 구성들로 배열된 출력들이 동일한 추론

을 갖도록 처리될 수 있다. 출력들을 풀링하는 것은 또한 풀링 전에 출력들의 원하는 특성들을 유지하면서 후속 신경망 계층들에서 수신된 입력들의 차원을 감소시킬 수 있으며, 이는 신경망들에 의해 생성된 추론들의 품질을 크게 손상시키지 않으면서 효율을 향상시킬 수 있다.

[0017] 일부 신경망들은 1보다 더 큰 스트라이드를 갖는 하나 이상의 컨볼루션 신경망 계층들을 포함한다. 개념상, 1의 스트라이드에 대해, 컨볼루션 신경망 계층은 가중치들의 세트들을 활성화 입력들에 순차적으로 적용할 수 있다. 즉, 활성화 입력 어레이에 대해, 가중치들이 활성화 입력들의 서브세트에 적용될 수 있고, 컨볼루션 계산이 완료될 때까지 활성화 입력들의 다른 각각의 서브세트에 한 위치, 예컨대 행 또는 열씩 이동될 수 있다. 1보다 더 큰 스트라이드를 가지며, 스트라이드가 정수인 컨볼루션 신경망 계층에 대해, 가중치들이 활성화 입력들의 서브세트에 적용될 수 있고, 컨볼루션 계산이 완료될 때까지 활성화 입력들의 다른 각각의 서브세트에 스트라이드와 동일한 위치들의 수만큼, 예컨대 스트라이드에 의해 표시된 행들 또는 열들의 수만큼 이동될 수 있다.

[0018] 본 명세서는 신경망 계층들을 처리하고 선택적으로 하나 이상의 신경망 계층들의 출력들에 대한 풀링을 수행하는 특수 목적 하드웨어 회로를 설명한다. 특수 목적 하드웨어 회로는 1의 스트라이드를 갖는 신경망 계층들을 처리할 수 있는 회로를 포함한다. 특수 목적 하드웨어 회로가 1보다 더 큰 스트라이드를 갖는 신경망 계층들의 처리를 직접적으로 지원하지는 않지만, 특수 목적 하드웨어 회로는 1보다 더 큰 스트라이드를 갖는 신경망 계층의 출력과 동일한 출력을 생성하도록 제어될 수 있다. 이에 따라, 개시된 기술의 하나의 기술적 효과 및 이점은, 1의 스트라이드를 갖는 신경망 계층들을 처리할 수 있는 회로가 보다 유연한 방식으로 그리고 1보다 더 큰 스트라이드를 갖는 신경망 계층에 대한 신경망 추론들을 계산하기 위해 사용될 수 있다는 점이다.

[0019] 도 1은 예시적인 신경망 처리 시스템(100)을 도시한다. 신경망 처리 시스템(100)은 아래에서 설명되는 시스템들, 컴포넌트들 및 기술들이 구현될 수 있는 하나 이상의 위치들에서 하나 이상의 컴퓨터들로서 구현되는 시스템의 일례이다.

[0020] 신경망 처리 시스템(100)은 특수 목적 하드웨어 회로(110)를 사용하여 신경망 계산들을 수행하는 시스템이다. 하드웨어 회로(110)는 신경망 계산들을 수행하기 위한 집적 회로이고, 하드웨어에 벡터 행렬 곱셈들을 수행하는 행렬 계산 유닛(120)을 포함한다. 하드웨어 회로(110)는 또한 행렬 계산 유닛(120)의 출력들에 풀링을 수행하기 위한 풀링 회로를 포함하는 벡터 계산 유닛(140)을 포함한다. 예시적인 특수 목적 하드웨어 회로(110)가 아래에서 도 3을 참조로 보다 상세히 설명된다.

[0021] 특히, 신경망 처리 시스템(100)은 특수 목적 하드웨어 회로(110) 상에 신경망들을 구현하라는 요청들을 수신하고, 특수 목적 하드웨어 회로(110) 상에 신경망들을 구현하고, 일단 주어진 신경망이 구현되면, 특수 목적 집적 회로(110)를 사용하여 신경망에 대한 입력들을 처리하여 신경망 추론들을 생성한다.

[0022] 즉, 신경망 처리 시스템(100)은 입력들을 처리하는 데 사용될 신경망에 대한 신경망 아키텍처를 지정하는 요청을 수신할 수 있다. 신경망 아키텍처는 신경망에서 계층들의 수와 구성 그리고 파라미터들을 가진 계층들 각각에 대한 파라미터들의 값들을 정의한다.

[0023] 특수 목적 집적 회로(110) 상에 신경망을 구현하기 위해, 신경망 처리 시스템(100)은 하나 이상의 물리적 위치들에 있는 하나 이상의 컴퓨터들 상의 하나 이상의 컴퓨터 프로그램들로서 구현되는 신경망 구현 엔진(150)을 포함한다.

[0024] 신경망 구현 엔진(150)은 특수 목적 하드웨어 회로(110)에 의해 실행될 때, 하드웨어 회로(110)로 하여금 신경망에 의해 지정된 동작들을 수행하여, 수신된 신경망 입력으로부터 신경망 출력을 생성하게 하는 명령들을 생성한다.

[0025] 명령들이 신경망 구현 엔진(150)에 의해 생성되어 하드웨어 회로(110)에 제공되면, 신경망 처리 시스템(100)은 신경망 입력들을 수신할 수 있고, 하드웨어 회로(110)로 하여금 생성된 명령들을 실행하게 함으로써 신경망을 사용하여 신경망 입력들을 처리할 수 있다.

[0026] 그러나 일부 신경망들은 하나 이상의 비호환 신경망 계층들을 포함한다. 본 명세서에서 사용되는 비호환 신경망 계층이라는 용어는 특수 목적 하드웨어 회로(110)에 의해 하드웨어에서 직접 수행될 수 없는 동작을 지정하는 신경망 계층을 의미한다. 하드웨어 회로(110) 상에 이러한 신경망들을 구현하기 위해, 신경망 구현 엔진(150)은 하드웨어 회로(110)에 의해 실행될 때 하드웨어 회로(110)로 하여금, 신경망 계층에 의해 지정된 것들과는 다르지만 비호환 신경망 계층의 규격을 충족하는 계층 출력, 예컨대 계층 출력 텐서, 즉 계층에 의해 지정된 동작들을 직접 수행함으로써 생성되었을 출력과 동일한 계층 출력이 생성되게 하는 동작들을 하드웨어에서

수행함으로써 비호환 신경망 계층에 대한 출력을 생성하게 하는 명령들을 생성한다.

[0027] 특히, 일부 신경망들은 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층을 포함한다. 이러한 신경망 계층은 입력 텐서로 비순차적으로 처리되는 하나 이상의 커널들을 특징으로 한다. 예를 들어, 1인 스트라이드로 커널 스트라이딩을 수행할 때, 커널은 입력 텐서의 엘리먼트들에 순차적으로 적용된다. 그러나 2인 스트라이드로 커널 스트라이딩을 수행할 때, 신경망 계층의 커널은 커널의 특정 엘리먼트가 입력 텐서의 모든 각각의 다른 엘리먼트에 적용되어 출력 텐서를 생성하도록 시프트된다. 그 다음, 출력 텐서는 신경망의 다른 계층에 의해 입력으로서 사용될 수 있다.

[0028] 하드웨어 회로(110) 상에서 행렬 연산들을 수행하는 메인 하드웨어 유닛은 행렬 계산 유닛(120)이기 때문에, 집적 회로는 1보다 더 큰 스트라이드를 갖는 신경망 계층을 직접 계산할 수 없다. 1보다 더 큰 스트라이드를 갖는 계층을 포함하는 신경망을 구현하기 위해, 신경망 구현 엔진(150)은 신경망에 의한 신경망 입력의 처리 중에 특수 목적 하드웨어 회로(110)에 의해 실행될 때 하드웨어 회로(110)로 하여금, 행렬 곱셈 유닛(120), 및 풀링 회로를 특징으로 하는 벡터 계산 유닛(140)을 사용하여 1보다 더 큰 스트라이드를 갖는 신경망 계층의 규격을 충족시키는 출력 텐서를 생성하도록 하드웨어에서 다른 동작들을 수행하게 하는 명령들을 생성한다. 이러한 명령들 및 다른 동작들은 도 7 - 도 10을 참조하여 아래에서 보다 상세하게 설명된다.

[0029] 도 2는 특수 목적 하드웨어 회로를 사용하여 신경망의 주어진 계층에 대한 계산을 수행하기 위한 예시적인 프로세스(200)의 흐름도이다. 편의상, 이 방법(200)은 방법(200)을 수행하는 하나 이상의 회로들을 갖는 시스템과 관련하여 설명될 것이다. 이 방법(200)은 수신된 입력으로부터 추론을 계산하기 위해 신경망의 각각의 계층에 대해 수행될 수 있다.

[0030] 시스템은 주어진 계층에 대한 가중치 입력들의 세트들(단계(202)) 및 활성화 입력들의 세트들(단계(204))을 수신한다. 가중치 입력들의 세트들 및 활성화 입력들의 세트들은 각각 특수 목적 하드웨어 회로의 동적 메모리 및 통합 버퍼로부터 수신될 수 있다. 일부 구현들에서는, 가중치 입력들의 세트들과 활성화 입력들의 세트들 모두가 통합 버퍼로부터 수신될 수 있다.

[0031] 시스템은 특수 목적 하드웨어 회로의 행렬 곱셈 유닛을 사용하여 가중치 입력들 및 활성화 입력들로부터 누산 값들을 생성한다(단계(206)). 일부 구현들에서, 누산 값들은 가중치 입력들의 세트들과 활성화 입력들의 세트들의 내적들이다. 즉, 계층의 모든 가중치들의 서브세트인 가중치들의 한 세트에 대해, 시스템은 각각의 가중치 입력에 각각의 활성화 입력을 곱하고 그 곱들을 서로 합하여 누산 값을 형성할 수 있다. 그 다음, 시스템은 가중치들의 다른 세트와 활성화 입력들의 다른 세트들과의 내적들을 계산할 수 있다. 일부 구현들에서, 특수 목적 하드웨어 회로는 특정 신경망 계층의 스트라이드, 즉 신경망 계층이 1의 스트라이드를 갖는지 또는 1보다 더 큰 스트라이드를 갖는지와 상관없이 유사하게 이러한 동작들을 수행할 수 있다. 행렬 곱셈 유닛으로부터의 출력들의 후속 처리는 신경망 계층이 1보다 더 큰 지정된 스트라이드로 처리되었다면 생성될 출력과 동등한 출력을 생성하도록 수행될 수 있다.

[0032] 시스템은 특수 목적 하드웨어 회로의 벡터 계산 유닛을 사용하여 누산 값들로부터 계층 출력을 생성할 수 있다(단계(208)). 일부 구현들에서, 벡터 계산 유닛은 누산 값들에 활성화 함수를 적용하는데, 이는 도 5를 참조하여 아래에서 더 설명될 것이다. 계층의 출력은 신경망에서 후속 계층에 대한 입력으로서 사용하기 위해 통합 버퍼에 저장될 수 있거나 추론을 결정하는 데 사용될 수 있다. 일부 구현들에서, 신경망 계층은 1보다 더 큰 스트라이드를 지정할 수 있고, 시스템은 1보다 더 큰 스트라이드를 갖는 신경망 계층의 출력과 동일한 계층 출력을 획득하기 위해 누산 값들에 대한 추가 처리를 수행할 수 있다. 시스템은 수신된 입력이 신경망의 각각의 계층을 통해 처리되었을 때 신경망의 처리를 완료하여, 수신된 입력에 대한 추론을 생성한다.

[0033] 도 3은 신경망 계산들을 수행하기 위한 예시적인 특수 목적 하드웨어 회로(300)를 도시한다. 시스템(300)은 호스트 인터페이스(302)를 포함한다. 호스트 인터페이스(302)는 신경망 계산을 위한 파라미터들을 포함하는 명령들을 수신할 수 있다. 파라미터들은 다음 중 하나 이상을 포함할 수 있다: 몇 개의 계층들이 처리되어야 하는지, 모델의 각각의 계층에 대한 가중치 입력들의 해당 세트들, 활성화 입력들의 초기 세트, 즉 추론이 계산될 신경망에 대한 입력, 각각의 계층의 대응하는 입력 및 출력 크기들, 신경망 계산을 위한 스트라이드 값, 및 처리될 계층의 타입, 예컨대 컨볼루션 계층 또는 완전히 접속된 계층.

[0034] 호스트 인터페이스(302)는 명령들을 시퀀서(306)에 전송할 수 있으며, 시퀀서(306)는 신경망 계산들을 수행하도록 회로를 제어하는 저레벨 제어 신호들로 명령들을 변환한다. 일부 구현들에서, 제어 신호들은 회로에서 데이터 흐름, 예컨대 가중치 입력들의 세트들 및 활성화 입력들의 세트들이 회로를 통해 어떻게 흐르는지를 조절한

다. 시퀀서(306)는 제어 신호들을 통합 버퍼(308), 행렬 연산 유닛(312) 및 벡터 계산 유닛(314)에 전송할 수 있다. 일부 구현들에서, 시퀀서(306)는 또한 직접 메모리 액세스 엔진(304) 및 동적 메모리(310)에 제어 신호들을 전송한다. 일부 구현들에서, 시퀀서(306)는 제어 신호들을 생성하는 프로세서이다. 시퀀서(306)는 제어 신호들의 타이밍을 사용하여, 적절한 시점들에 제어 신호들을 회로(300)의 각각의 컴포넌트에 전송할 수 있다. 일부 다른 구현들에서, 호스트 인터페이스(302)는 외부 프로세서로부터 제어 신호를 전달한다.

[0035] 호스트 인터페이스(302)는 가중치 입력들의 세트들 및 활성 입력들의 초기 세트를 직접 메모리 액세스 엔진(304)에 전송할 수 있다. 직접 메모리 액세스 엔진(304)은 통합 버퍼(308)에서 활성화 입력들의 세트들을 저장할 수 있다. 일부 구현들에서, 직접 메모리 액세스는 메모리 유닛일 수 있는 동적 메모리(310)에 가중치들의 세트들을 저장한다. 일부 구현들에서, 동적 메모리(310)는 회로로부터 벗어난 위치에 있다.

[0036] 통합 버퍼(308)는 메모리 버퍼이다. 이는 직접 메모리 액세스 엔진(304)으로부터의 활성화 입력들의 세트 및 벡터 계산 유닛(314)의 출력들을 저장하는 데 사용될 수 있다. 벡터 계산 유닛(314)은 도 6을 참조하여 아래에서 보다 상세히 설명될 것이다. 직접 메모리 액세스 엔진(304)은 또한 통합 버퍼(308)로부터 벡터 계산 유닛(314)의 출력들을 판독할 수 있다.

[0037] 동적 메모리(310) 및 통합 버퍼(308)는 가중치 입력들의 세트들 및 활성화 입력들의 세트들을 각각 행렬 계산 유닛(312)에 전송할 수 있다. 일부 구현들에서, 행렬 계산 유닛(312)은 2차원 수축기 어레이이다. 행렬 계산 유닛(312)은 또한 수학 연산들, 예컨대 곱셈 및 가산을 수행할 수 있는 1차원 수축기 어레이 또는 다른 회로일 수 있다. 일부 구현들에서, 행렬 계산 유닛(312)은 범용 행렬 프로세서이다.

[0038] 행렬 계산 유닛(312)은 가중치 입력들 및 활성화 입력들을 처리할 수 있고, 벡터 계산 유닛(314)에 출력들의 벡터를 제공할 수 있다. 일부 구현들에서, 행렬 계산 유닛(312)은 출력들의 벡터를 통합 버퍼(308)에 전송하는데, 통합 버퍼는 출력들의 벡터를 벡터 계산 유닛(314)으로 전송한다. 벡터 계산 유닛(314)은 출력들의 벡터를 처리할 수 있고 처리된 출력들의 벡터를 통합 버퍼(308)에 저장할 수 있다. 벡터 계산 유닛(314)은 1보다 더 큰 스트라이드를 갖는 신경망 계층들에 대해, 출력들의 벡터를 처리하여 1보다 더 큰 스트라이드를 갖는 신경망 계층의 출력과 동일한 계층 출력 텐서를 생성할 수 있고, 통합 버퍼(308)에서 계층 출력 텐서를 저장할 수 있다. 처리된 출력들의 벡터는 예컨대, 신경망의 후속 계층에서 사용할, 행렬 계산 유닛(312)에 대한 활성화 입력들로서 사용될 수 있다. 행렬 계산 유닛(312) 및 벡터 계산 유닛(314)은 도 4 및 도 6을 각각 참조하여 아래에서 보다 상세하게 설명될 것이다.

[0039] 도 4는 행렬 계산 유닛을 포함하는 예시적인 아키텍처(400)를 도시한다. 행렬 계산 유닛은 2차원 수축기 어레이(406)이다. 어레이(406)는 다수의 셀들(404)을 포함한다. 일부 구현들에서, 수축기 어레이(406)의 제1 차원(420)은 셀들의 열들에 대응하고 수축기 어레이(406)의 제2 차원(422)은 셀들의 행들에 대응한다. 수축기 어레이는 열들보다 더 많은 행들, 행들보다 더 많은 열들 또는 동일한 수의 열들과 행들을 가질 수 있다.

[0040] 예시된 예에서, 값 로더들(402)은 어레이(406)의 행들에 활성화 입력들을 전송하고, 가중치 페처 인터페이스(408)는 어레이(406)의 열들에 가중치 입력들을 전송한다. 그러나 일부 다른 구현들에서, 활성화 입력들은 열들로 전송되고 가중치 입력들은 어레이(406)의 행들로 전송된다.

[0041] 값 로더들(402)은 통합 버퍼, 예컨대 도 3의 통합 버퍼(308)로부터 활성화 입력들을 수신할 수 있다. 각각의 값 로더는 대응하는 활성화 입력을 어레이(406)의 별개의 최좌측 셀에 전송할 수 있다. 예를 들어, 값 로더(412)는 셀(414)에 활성화 입력을 전송할 수 있다.

[0042] 가중치 페처 인터페이스(408)는 메모리 유닛, 예컨대 도 3의 동적 메모리(310)로부터 가중치 입력을 수신할 수 있다. 가중치 페처 인터페이스(408)는 대응하는 가중치 입력을 어레이(406)의 별개의 최상위 셀에 전송할 수 있다. 예를 들어, 가중치 페처 인터페이스(408)는 가중치 입력들을 셀들(414, 416)에 전송할 수 있다. 가중치 페처 인터페이스(408)는 추가로, 메모리 유닛, 예컨대 동적 메모리(310)로부터 다수의 가중치들을 수신하고 병렬로 어레이(406)의 별개의 최상위 셀들로 다수의 가중치들을 전송할 수 있다. 예를 들어, 가중치 페처 인터페이스(408)는 서로 다른 가중치들을 동시에 셀들(414, 416)에 전송할 수 있다.

[0043] 일부 구현들에서, 호스트 인터페이스, 예컨대 도 3의 호스트 인터페이스(302)는 하나의 차원을 따라 어레이(406) 전체에 걸쳐, 예컨대 우측으로 활성화 입력들을 시프트하면서, 다른 차원을 따라 어레이(406) 전체에 걸쳐, 예컨대 최하단으로 가중치 입력들을 시프트한다. 예를 들어, 1 클록 사이클에 걸쳐, 셀(414)에서의 활성화 입력은 셀(414)의 오른쪽에 있는 셀(416)의 활성화 레지스터로 시프트될 수 있다. 마찬가지로, 셀(416)에서의 가중치 입력은 셀(414) 아래에 있는 셀(418)의 가중치 레지스터로 시프트될 수 있다.

[0044] 각각의 클록 사이클에서, 각각의 셀은 주어진 가중치 입력, 주어진 활성화 입력 및 인접한 셀로부터의 누산 출력을 처리하여 누산 출력을 생성할 수 있다. 누산 출력은 또한 주어진 가중치 입력과 동일한 차원을 따라 인접한 셀로 전달될 수 있다. 각각의 셀은 또한 인접한 셀로부터의 누산 출력을 처리하지 않고 주어진 가중치 입력 및 주어진 활성화 입력을 처리하여 출력을 생성할 수 있다. 출력은 누산되지 않고 주어진 가중치 입력 및 출력과 동일한 차원들을 따라 인접한 셀들로 전달될 수 있다. 개개의 셀은 도 5를 참조하여 아래에서 추가 설명된다.

[0045] 일부 구현들에서, 단위 행렬, 즉 주 대각선에 1들을 그리고 다른 곳에는 0들을 갖는 행렬이 어레이(406)로 전달될 수 있으며, 이로써 값 로더들(402)에서 제공된 입력들을 수정 없이 누산기(410)에 전달할 수 있다. 이것은 2개의 입력들의 엘리먼트별 곱셈을 수행하는 데 사용될 수 있는데, 여기서 누산기들에서의 제1 출력은 output = MatMul(input1, identity)로서 표현될 수 있고, MatMul은 행렬 계산 유닛이 행렬 곱셈을 수행하기 위한 명령이며, 엘리먼트별 곱셈 결과에 대응하는 제2 출력은 output *= MatMul(input2, identity)로서 표현된다. *= 연산, 즉 output = output * MatMul(input2, identity) 연산을 수행하기 위해, 아키텍처(400)는 += 또는 *= 계산들을 수행하기 위한 컴포넌트를 포함할 수 있다. += 또는 *= 연산들을 수행하기 위한 컴포넌트는 누산기들(410) 앞에, 즉 셀들(404)의 마지막 행 뒤에 위치될 수 있다. 일부 구현들에서, 도 3의 벡터 계산 유닛(314)은 += 또는 *= 연산들을 수행하기 위한 컴포넌트를 포함할 수 있는데, 즉 벡터 계산 유닛(314)이 output *= MatMul(input2,identity) 연산을 수행하여 엘리먼트별 곱셈을 수행한다.

[0046] 누산 출력은 가중치 입력과 동일한 열을 따라, 예컨대 어레이(406)의 열의 최하단을 향해 전달될 수 있다. 일부 구현들에서, 각각의 열의 하단에서, 어레이(406)는 열들보다 더 많은 활성화 입력들을 갖는 계층들로 계산들을 수행할 때 각각의 열로부터의 각각의 누산 출력을 저장하고 누산하는 누산기 유닛들(410)을 포함할 수 있다. 일부 구현들에서, 각각의 누산기 유닛은 다수의 병렬 누산들을 저장한다. 누산기 유닛들(410)은 각각의 누산 출력을 누산하여 최종 누산 값을 생성할 수 있다. 최종 누산 값은 벡터 계산 유닛, 예컨대 도 6의 벡터 계산 유닛으로 전송될 수 있다. 일부 다른 구현들에서, 누산기 유닛들(410)은 행들보다 더 적은 활성화 입력들을 갖는 계층들을 갖는 계층들을 처리할 때 어떠한 누산들도 수행하지 않고 벡터 계산 유닛에 누산 값들을 전달한다.

[0047] 도 5는 수축기 어레이 내부의 셀, 예컨대 도 4의 수축기 어레이(406)의 셀들(414, 416 또는 418) 중 하나의 셀의 예시적인 아키텍처(500)를 도시한다.

[0048] 셀은 활성화 입력을 저장하는 활성화 레지스터(506)를 포함할 수 있다. 활성화 레지스터는 수축기 어레이 내의 셀 위치에 따라 왼쪽 인접 셀, 즉 주어진 셀의 왼쪽에 위치된 인접 셀로부터 또는 통합 버퍼로부터 활성화 입력을 수신할 수 있다. 셀은 가중치 입력을 저장하는 가중치 레지스터(502)를 포함할 수 있다. 가중치 입력은 수축기 어레이 내의 셀의 위치에 따라, 최상부 인접 셀로부터 또는 가중치 페처 인터페이스로부터 전송될 수 있다. 셀은 또한 합산 레지스터(504)를 포함할 수 있다. 합산 레지스터(504)는 최상부 인접 셀로부터의 누산 값을 저장할 수 있다. 곱셈 회로(508)는 가중치 레지스터(502)로부터의 가중치 입력을 활성화 레지스터(506)로부터의 활성화 입력과 곱하는 데 사용될 수 있다. 곱셈 회로(508)는 곱을 합산 회로(510)에 출력할 수 있다.

[0049] 합산 회로(510)는 곱과 합산 레지스터(504)로부터의 누산 값을 합산하여 새로운 누산 값을 생성할 수 있다. 그 다음, 합산 회로(510)는 새로운 누산 값을 최하단 인접 셀에 위치된 다른 합산 레지스터로 전송할 수 있다. 새로운 누산 값은 최하단 인접 셀의 합산에 대한 피연산자로서 사용될 수 있다. 합산 회로(510)는 또한, 합산 레지스터(504)로부터의 값을 받아, 합산 레지스터(504)로부터의 값을 곱셈 회로(508)로부터의 곱과 합하지 않고 합산 레지스터(504)로부터의 값을 최하단 인접 셀에 전송할 수 있다.

[0050] 셀은 또한 처리를 위해 가중치 입력 및 활성화 입력을 인접 셀들로 시프트할 수 있다. 예를 들어, 가중치 경로 레지스터(512)는 최하단 인접 셀의 다른 가중치 레지스터에 가중치 입력을 전송할 수 있다. 활성화 레지스터(506)는 우측 인접 셀의 다른 활성화 레지스터에 활성화 입력을 전송할 수 있다. 따라서 가중치 입력과 활성화 입력이 둘 다 후속 클록 사이클에서 어레이의 다른 셀들에 의해 재사용될 수 있다.

[0051] 일부 구현들에서, 셀은 또한 제어 레지스터를 포함한다. 제어 레지스터는 셀이 인접 셀들로 가중치 입력을 시프트해야 하는지 또는 활성화 입력을 시프트해야 하는지를 결정하는 제어 신호를 저장할 수 있다. 일부 구현들에서, 가중치 입력 또는 활성화 입력을 시프트하는 것은 하나 이상의 클록 사이클들이 걸린다. 제어 신호는 또한 곱셈 회로(508)로 활성화 입력이 전송되는지 또는 가중치 입력들이 전송되는지를 결정할 수 있거나, 곱셈 회로(508)가 활성화 및 가중치 입력들에 대해 동작하는지 여부를 결정할 수 있다. 제어 신호는 또한 예컨대, 배선을 사용하여 하나 이상의 인접한 셀들로 전달될 수 있다.

[0052] 일부 구현들에서, 가중치들은 가중치 경로 레지스터(512)로 사전 시프트된다. 가중치 경로 레지스터(512)는 예컨대, 최상부 인접 셀로부터 가중치 입력을 수신할 수 있고, 제어 신호에 기초하여 가중치 레지스터(502)에 가중치 입력을 전송할 수 있다. 가중치 레지스터(502)는 활성화 입력들이 다수의 클록 사이클들에 걸쳐, 예컨대 활성화 레지스터(506)를 통해 셀로 전송될 때, 가중치 입력이 셀 내에 유지되고 인접한 셀로 전송되지 않도록 가중치 입력을 정적으로 저장할 수 있다. 따라서 가중치 입력은 예컨대, 곱셈 회로(508)를 사용하여 다수의 활성화 입력들에 적용될 수 있고, 각각의 누산 값들은 인접한 셀로 전송될 수 있다.

[0053] 도 6은 벡터 계산 유닛(602)의 예시적인 아키텍처(600)를 도시한다. 벡터 계산 유닛(602)은 행렬 계산 유닛, 예컨대 도 3을 참조하여 설명한 행렬 계산 유닛(312) 또는 도 4의 행렬 계산 유닛의 누산기들(410)로부터 누산 값들의 벡터를 수신할 수 있다.

[0054] 벡터 계산 유닛(602)은 활성화 유닛(604)에서 누산 값들의 벡터를 처리할 수 있다. 일부 구현들에서, 활성화 유닛은 각각의 누산 값에 비선형 함수를 적용하여 활성화 값들을 생성하는 회로를 포함한다. 예를 들어, 비선형 함수는 tanh(x)일 수 있으며, 여기서 x는 누산 값이다.

[0055] 선택적으로, 벡터 계산 유닛(602)은 풀링 회로(608)를 사용하여 값들, 예컨대 활성화 값들을 풀링할 수 있다. 풀링 회로(608)는 값들 중 하나 이상의 값에 집계 함수를 적용하여 풀링된 값들을 생성할 수 있다. 일부 구현들에서, 집계 함수들은 값들의 또는 값들의 서브세트의 최대, 최소 또는 평균을 리턴하는 함수들이다.

[0056] 제어 신호들(610)은 예컨대, 도 3의 시퀀서(306)에 의해 전송될 수 있고, 벡터 계산 유닛(602)이 누산 값들의 벡터를 어떻게 처리하는지를 조절할 수 있다. 즉, 제어 신호들(610)은 활성화 값들이 풀링되는지 여부를 조절할 수 있거나- 활성화 값들은 예컨대, 통합 버퍼(308)에 저장됨 -, 또는 활성화 값들의 처리를 다른 식으로 조절할 수 있다. 제어 신호들(610)은 또한 활성화 또는 풀링 함수들뿐만 아니라, 활성화 값들 또는 풀링 값들, 예컨대 스트라이드 값을 처리하기 위한 다른 파라미터들을 지정할 수 있다.

[0057] 벡터 계산 유닛(602)은 값들, 예컨대 활성화 값들 또는 풀링된 값들을 통합 버퍼, 예컨대 도 3의 통합 버퍼(308)에 전송할 수 있다. 일부 구현들에서, 풀링 회로(608)는 활성화 값들 또는 풀링된 값들을 수신하고 활성화 값들 또는 풀링된 값들을 통합 버퍼에 저장한다.

[0058] 도 7은 풀링 회로에 대한 예시적인 아키텍처(700)를 도시한다. 풀링 회로는 하나 이상의 활성화된 값들에 집계 함수를 적용하여 풀링된 값들을 생성할 수 있다. 예시로, 아키텍처(700)는 활성화된 값들의 4×4 세트의 풀링을 수행할 수 있다. 도 7에 도시된 풀링은 정사각형 영역, 즉 4×4를 갖지만, 직사각형 영역들이 가능하다. 예를 들어, 영역이 n×m의 윈도우를 갖는다면, 아키텍처(700)는 n * m 레지스터들, 즉 n개의 열들과 m개의 행들을 가질 수 있다.

[0059] 풀링 회로 아키텍처(700)는 값들의 벡터로부터, 예컨대 도 6의 활성화 회로(604)로부터 엘리먼트들의 시퀀스를 수신할 수 있다. 예를 들어, 시퀀스는 이미지의 8×8 부분의 픽셀들을 나타낼 수 있고, 풀링 회로 아키텍처(700)는 8×8 부분의 4×4 서브세트로부터의 값들을 풀링할 수 있다. 일부 구현들에서, 풀링된 값들은 풀링 회로 아키텍처(700)에 의해 일단 계산되면 시퀀스에 부가된다. 일부 구현들에서, 신경망 프로세서는 다수의 병렬 풀링 회로들을 포함한다. 각각의 클록 사이클 동안, 각각의 풀링 회로는 활성화 회로(604)로부터의 값들의 벡터로부터 각각의 엘리먼트를 수신할 수 있다. 각각의 풀링 회로는 활성화 회로(604)로부터 수신된 엘리먼트들을 래스터 순서로 도달하는 2차원 이미지로서 해석할 수 있다.

[0060] 풀링 회로는 일련의 레지스터들 및 메모리 유닛들을 포함할 수 있다. 각각의 레지스터는 레지스터들 내부에 저장된 값들에 걸쳐 집계 함수를 적용하는 집계 회로(706)에 출력을 전송할 수 있다. 집계 함수는 값들의 세트로부터 최소, 최대 또는 평균 값을 리턴할 수 있다.

[0061] 제1 값은 레지스터(702)로 전송되어 레지스터(702) 내에 저장될 수 있다. 후속 클록 사이클에서, 제1 값은 후속 레지스터(708)로 시프트하여 메모리(704)에 저장될 수 있고, 제2 값은 레지스터(702)로 전송되어 레지스터(702) 내에 저장될 수 있다.

[0062] 4개의 클록 사이클들 후에, 4개의 값들이 처음 4개의 레지스터들(702, 708-712) 내에 저장된다. 일부 구현들에서, 메모리 유닛(704)은 선입 선출(FIFO: first-in-first-out)에 따라 동작한다. 각각의 메모리 유닛은 최대 8개의 값들을 저장할 수 있다. 메모리 유닛(704)이 픽셀들의 완전한 행을 포함한 후, 메모리 유닛(704)은 레지스터(714)에 값을 전송할 수 있다.

[0063] 임의의 주어진 시점에서, 집계 회로(706)는 각각의 레지스터로부터의 값들에 액세스할 수 있다. 레지스터들의

값들은 이미지의 4×4 부분에 대한 값들을 나타내야 한다.

[0064] 풀링 회로는 집계 회로(706)를 사용함으로써 액세스된 값들로부터 풀링된 값, 예컨대 최대, 최소 또는 평균 값을 생성할 수 있다. 풀링된 값은 통합 버퍼, 예컨대 도 3의 통합 버퍼(308)로 전송될 수 있다.

[0065] 제1 풀링된 값을 생성한 후에, 풀링 회로는 새로운 값들이 레지스터들에 저장되고 집계 회로(706)에 의해 풀링될 수 있도록 각각의 레지스터를 통해 값들을 시프트함으로써 풀링된 값들을 생성하는 것을 계속할 수 있다. 예를 들어, 아키텍처(700)에서, 풀링 회로는 4개 이상의 클록 사이클들에 걸쳐 값들을 시프트함으로써, 메모리 유닛들의 값들을 레지스터들로 시프트할 수 있다. 일부 구현들에서, 풀링 회로는 새로운 값이 마지막 최상위 레지스터, 예를 들어 레지스터(716)에 저장될 때까지 새로운 값들을 시프트한다.

[0066] 그 후, 집계 회로(706)는 레지스터들에 저장된 새로운 값들을 풀링할 수 있다. 새로운 값들을 풀링한 결과가 통합 버퍼에 저장될 수 있다.

[0067] 도 8은 1보다 더 큰 스트라이드를 갖는 신경망의 주어진 컨볼루션 계층에 대한 계산을 수행하기 위한 예시적인 프로세스(800)의 흐름도이다. 일반적으로, 프로세스(700)는 특수 목적 하드웨어 회로를 포함하는 하나 이상의 컴퓨터들의 시스템에 의해 수행된다. 일부 구현들에서, 예시적인 프로세스(800)는 도 1의 시스템에 의해 수행될 수 있다.

[0068] 시스템은 특수 목적 하드웨어 회로 상에 신경망을 구현하라는 요청을 수신한다(단계(802)). 특히, 신경망은 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층을 포함한다. 요청은 신경망을 구현하기 위한 다른 파라미터들, 이를테면 신경망을 사용하여 처리할 입력, 신경망에 의해 생성된 출력 텐서를 저장할 위치들, 또는 다른 파라미터들을 추가로 지정할 수 있다.

[0069] 시스템은 1보다 더 큰 스트라이드를 갖는 신경망 계층을 처리하는 데 사용될 마스킹 텐서를 요청에 기초하여 생성한다(단계(804)). 예를 들어, 신경망을 구현하라는 요청 및 신경망에 대한 입력을 지정하는 정보를 수신하는 것에 기초하여, 시스템은 1보다 더 큰 스트라이드를 갖는 신경망 계층을 처리하기 위한 마스킹 텐서를 생성한다.

[0070] 마스킹 텐서의 크기는 지정된 입력의 치수들 또는 1보다 더 큰 스트라이드를 갖는 신경망 계층에 대한 입력 텐서의 예상 크기에 기초하여 결정될 수 있다. 마스킹 텐서에 포함된 값들은 1보다 더 큰 스트라이드를 갖는 신경망 계층의 지정된 스트라이드에 기초하여 결정될 수 있다. 예를 들어, 신경망 계층이 4의 지정된 스트라이드를 갖는다면, 마스킹 텐서의 네 번째 엘리먼트마다 1로 설정될 수 있는 한편, 마스킹 텐서의 다른 모든 항목들은 0으로 설정될 수 있다. 일부 구현들에서, 신경망은 1보다 더 큰 스트라이드를 갖는 다수의 계층들을 포함할 수 있고, 시스템은 1보다 더 큰 스트라이드를 갖는 계층들 각각에 대한 대응하는 마스킹 텐서들을 생성할 수 있다. 추가로, 일부 구현들에서, 시스템은 예컨대, 메모리에 마스킹 행렬들 또는 마스킹 행렬 컴포넌트들의 라이브러리를 저장할 수 있으며, 라이브러리의 사용에 기반하여 마스킹 행렬을 선택 또는 생성할 수 있다.

[0071] 시스템은 특수 목적 하드웨어 회로(110)에 의해 실행될 때, 특수 목적 하드웨어 회로(110)로 하여금 신경망에 의한 입력 텐서의 처리 동안, 마스킹 텐서를 사용하여 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서를 생성하게 하는 명령들을 생성한다(단계(806)). 예를 들어, 요청에 대한 응답으로, 신경망 구현 엔진(150)은 특수 목적 하드웨어 회로(110)가 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층을 사용하여 입력 텐서를 처리한 경우와 동일한 출력 텐서, 즉 출력 벡터를 생성하도록 특수 목적 하드웨어 회로(110)에 지시하거나 특수 목적 하드웨어 회로(110)를 제어하는 명령들을 생성할 수 있다.

[0072] 시스템은 명령들 및 마스킹 텐서를 특수 목적 하드웨어 회로(110)에 송신한다(단계(808)). 예를 들어, 신경망 구현 엔진(150)은 명령들을 특수 목적 하드웨어 회로(110)에 제공할 수 있고, 특수 목적 하드웨어 회로(110)는 예컨대, 도 3의 호스트 인터페이스(302)에서 명령들을 수신할 수 있다. 신경망 구현 엔진(150)은 또한 호스트 인터페이스(302)에 의해 또한 수신될 수 있는 신경망 계산을 위한 다른 명령들 및/또는 파라미터들을 제공할 수 있다.

[0073] 도 9는 1보다 더 큰 스트라이드를 갖는 신경망 계산 계층을 계산하기 위한 예시적인 프로세스(900)의 흐름도이다. 예를 들어, 프로세스(900)는 신경망 구현 엔진(150)으로부터 수신된 명령들에 기초하여 도 1의 특수 목적 하드웨어 회로(110)에 의해 수행될 수 있다.

[0074] 예를 들어, 1보다 더 큰 스트라이드를 갖는 신경망 계층을 구현하기 위한 명령들을 수신하면, 호스트 인터페이스(302)는 명령들을 도 3의 시퀀서(306)에 전송할 수 있고, 시퀀서(306)는 신경망 계산을 수행하도록 도 3의 특

수 목적 하드웨어 회로(300)를 제어하는 저레벨 제어 신호들로 명령들을 변환할 수 있다.

[0075] 수신된 명령들에 기초하여, 특수 목적 하드웨어 회로(300)는 1의 스트라이드를 갖는 제2 컨볼루션 신경망 계층을 사용하여 컨볼루션 신경망 계층에 대한 입력 텐서를 처리한다(단계(902)). 예를 들어, 수신된 명령들로부터 발생된 제어 신호들은 1과 같은 스트라이드를 갖지만 그 외에는 컨볼루션 신경망 계층과 동일한 제2 컨볼루션 신경망 계층을 사용하여 입력 텐서, 예컨대 통합 버퍼(308)에 저장된 신경망의 선행 계층의 출력 또는 특수 목적 하드웨어 회로(300)에 지정된 또는 제공된 신경망에 대한 입력을 처리하도록 특수 목적 하드웨어 회로(300)를 제어하여, 컨볼브(convolve)된 텐서를 생성한다.

[0076] 제2 컨볼루션 신경망 계층을 사용하여 입력 텐서를 처리하기 위해, 제어 신호들은 입력 텐서, 즉 신경망에 대한 입력 또는 선행 신경망 계층의 출력에 대응할 수 있는 활성화 입력들을 도 3의 행렬 계산 유닛(312)에 제공하도록 통합 버퍼(308)를 제어할 수 있다. 제어 신호들은 또한 1의 스트라이드, 즉 단위(unity) 스트라이드를 갖지만 그 외에는 1보다 더 큰 스트라이드를 갖는 신경망 계층과 동일한 제2 신경망 계층에 대응하는 행렬 계산 유닛(312)에 가중치들을 제공하도록 도 3의 직접 메모리 액세스 엔진(304) 및/또는 동적 메모리(310)에 명령할 수 있다.

[0077] 시퀀서(306)는 가중치들을 사용하여, 예컨대 도 3과 관련하여 설명한 프로세스를 사용하여 입력 텐서를 처리하도록 행렬 계산 유닛(312)을 제어하는 명령들을 추가로 생성할 수 있다. 일부 구현들에서, 행렬 계산 유닛(312)은 2015년 9월 3일자로 출원된 미국 특허출원 제14/844,738호에 기술된 기술들을 사용하여 컨볼루션을 수행하는데, 이 출원은 이로써 그 전체가 인용에 의해 포함된다.

[0078] 행렬 계산 유닛(312)은 제어 신호들에 기초하여 계산들을 수행하고, 컨볼브된 텐서를 벡터 계산 유닛(314)으로 출력한다. 예를 들어, 행렬 계산 유닛(312)은 행렬 계산 유닛(312)에 의해 생성된 출력들의 벡터를 벡터 계산 유닛(314)에 전송한다. 출력들의 벡터는, 1의 스트라이드를 갖지만 그 외에는 1보다 더 큰 스트라이드를 갖는 신경망 계층과 동일한 신경망 계층에 대응하는 가중치들을 사용하여 입력 텐서를 처리하는 것에 기초하여 결정될 수 있다. 벡터 계산 유닛(314)은 컨볼브된 텐서를 통합 버퍼(308)에 저장할 수 있다.

[0079] 1의 스트라이드를 갖는 컨볼루션 신경망 계층을 통해 활성화 입력들을 처리하여 컨볼브된 텐서를 생성한 후, 특수 목적 하드웨어 회로(300)는 제2 컨볼루션 신경망 계층이 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 엘리먼트들을 0으로 설정한다(단계(904)). 엘리먼트를 0으로 설정하는 것은 일반적으로 엘리먼트의 현재 값들을 0으로 대체하는 것을 의미한다. 값들을 무효화하는 것, 즉 0으로 설정하는 것은 컨볼브된 텐서와 마스킹 텐서, 즉 신경망 구현 엔진(150)에 의해 생성되어 특수 목적 신경망에 송신된 마스킹 텐서와의 엘리먼트별 곱셈을 수행함으로써 달성될 수 있다.

[0080] 입력 텐서가 지정된 스트라이드를 갖는 컨볼루션 신경망 계층에 의해 처리되었다면 생성되지 않았을 컨볼브된 텐서의 그러한 값들을 무효화하기 위해, 시퀀서(306)는 컨볼브된 텐서와 마스킹 텐서의 엘리먼트별 곱셈을 수행하도록 행렬 곱셈 유닛(312)을 제어하기 위한 제어 신호들을 전송할 수 있다. 컨볼브된 텐서는 시퀀서(306)로부터의 다른 제어 신호들에 기초하여 통합 버퍼(308)로부터 행렬 곱셈 유닛(312)으로 전송될 수 있고, 마스킹 텐서는 시퀀서(306)로부터 직접 메모리 액세스 엔진(304) 또는 동적 메모리(310)로의 제어 신호들에 기초하여, 즉 마스킹 텐서가 특수 목적 하드웨어 회로(300)에 의해 수신되어 동적 메모리(310)에 저장된 후에 행렬 계산 유닛(312)으로 전송될 수 있다.

[0081] 일반적으로, 도 8과 관련하여 설명한 바와 같이, 마스킹 텐서는, 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층으로 입력 텐서를 처리함으로써 생성될 엘리먼트들에 대응하는 엘리먼트 위치들에 단위 값 엘리먼트들, 즉 1의 값들을 포함하고, 다른 모든 위치들, 즉 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층으로 활성화 값들을 처리함으로써 생성되지 않을 엘리먼트들에 대응하는 위치들에 0 값 엘리먼트들을 포함하는 벡터이다.

[0082] 마스킹 텐서는 예를 들어, 동적 메모리(310)에 저장될 수 있고, 시퀀서(306)는 동적 메모리(310)로부터 행렬 계산 유닛(312)으로 마스킹 텐서를 전송하도록 제어 신호들을 전송할 수 있다. 예를 들어, 특수 목적 하드웨어 회로(300)에 제공되는 명령들은 동적 메모리(310)에서의 마스킹 텐서의 위치를 식별, 예컨대 제공할 수 있거나, 이후에 동적 메모리(310)에 저장되는 마스킹 텐서를 정의하는 데이터를 포함할 수 있으며, 시퀀서(306)는 동적 메모리(310) 내의 위치에 저장된 마스킹 텐서가 행렬 계산 유닛(312)에 전송되게 하는 제어 신호들을 전송할 수 있다. 추가로, 시퀀서(306)는 통합 버퍼(308)에 저장된 컨볼브된 텐서가 행렬 계산 유닛(312)에 제공되게 하도록 제어 신호들을 제공할 수 있다. 그 다음, 행렬 계산 유닛(312)은 컨볼브된 텐서와 마스킹 텐서의 엘리먼트별 곱셈을 수행하여 수정된 컨볼브된 텐서를 생성한다. 수정된 컨볼브된 텐서는 벡터 계산 유닛(314)에 의해

행렬 계산 유닛(312)으로부터 수신될 수 있다. 벡터 계산 유닛(314)은 수정된 컨볼브된 텐서를 통합 버퍼(308)에 선택적으로 저장할 수 있다.

[0083] 마스킹 텐서와의 엘리먼트별 곱셈으로 인해, 수정된 컨볼브된 텐서는 입력 텐서가 1보다 더 큰 지정된 스트라이드를 갖는 신경망 계층을 사용하여 처리되었다면 출력될 값들을 포함한다. 수정된 컨볼브된 텐서는 입력 텐서가 지정된 스트라이드를 갖는 컨볼루션 신경망으로 처리되었다면 출력되지 않았을 1의 스트라이드를 갖는 컨볼루션 신경망 계층을 사용한 입력 텐서의 계산에서 출력된 값들에 대응하는 위치들에 0들을 포함한다. 다른 구현들에서, 컨볼브된 텐서의 엘리먼트들을 0으로 하는 다른 방법들이 이용될 수 있다. 예를 들어, 컨볼브된 행렬은 수정된 형태로 통합 버퍼(308) 또는 다른 메모리에 재기록될 수 있으며, 여기서 지정된 스트라이드를 갖는 컨볼루션 신경망을 사용한 입력 텐서의 계산에서 출력된 값들에 대응하는 엘리먼트들은 변경되지 않고, 다른 엘리먼트들은 0으로 기록된다.

[0084] 벡터 계산 유닛(314)은 수정된 컨볼브된 텐서를 수신하고, 수정된 컨볼브된 텐서에 대한 최대 풀링을 수행하여, 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층에 대한 계층 출력 텐서를 생성한다(단계(906)). 예를 들어, 벡터 계산 유닛(314)은 행렬 계산 유닛(312)으로부터 수정된 컨볼브된 텐서를 수신할 수 있고, 풀링 회로(608)를 사용하여, 수정된 컨볼브된 텐서에 대한 최대 풀링을 수행할 수 있다. 최대 풀링은 한 세트의 데이터를 수신하고 데이터의 하나 이상의 서브세트들 각각에 대해 서브세트의 엘리먼트들의 최대 값을 출력하는 연산이다. 수정된 컨볼브된 텐서의 최대 풀링을 수행하는 것은 수정된 컨볼브된 텐서의 엘리먼트들의 다수의 서브세트들 각각에 대해, 서브세트의 최대 값을 포함하는 텐서를 야기한다. 벡터 계산 유닛(314)은 컨볼루션 신경망 계층의 지정된 스트라이드에 기초하여 결정된 수정된 컨볼브된 텐서의 윈도우들에 대해 최대 풀링을 수행할 수 있다. 예를 들어, 스트라이드가 2인 경우, 풀링 회로(608)는 각각의 2×2 윈도우로부터의 최대 값 엘리먼트를 포함하는 계층 출력 텐서를 생성하기 위해, 2×2 윈도우를 사용하여 최대 풀링을 수행할 것이다. 스트라이드가 4인 신경망 계층의 경우, 풀링 회로(608)는 각각의 4×4 윈도우로부터의 최대 값 엘리먼트를 포함하는 계층 출력 텐서를 생성하기 위해, 4×4 윈도우를 사용하여 최대 풀링을 수행할 것이다. 최대 풀링 연산의 결과는 벡터 연산 유닛(314)에 의해 통합 버퍼(308)에 저장되는데, 여기서 결과는 특수 목적 하드웨어 회로(300)가 1보다 더 큰 스트라이드를 갖는 신경망 계층을 사용하여 입력 텐서를 처리했다면 생성될 출력과 동일한 출력 텐서이다. 계층 출력 텐서를 사용하여 신경망의 후속 계층의 처리가 수행되어 결국 신경망의 추론을 얻을 수 있다.

[0085] 도 10은 1보다 더 큰 스트라이드를 갖는 신경망의 주어진 계층에 대한 계산의 일례를 도시한다. 도 10의 예는 도 7의 프로세스 및 도 2의 특수 목적 하드웨어 회로(300)를 사용하여 수행될 수 있다. 예시로, 도 10의 예는 활성화 값들의 8×8 어레이에 4의 스트라이드를 갖는 컨볼루션 신경망 계층을 적용한다. 컨볼루션 신경망 계층은 활성화 값들의 8×8 어레이에 적용될 가중치들의 4×4 커널을 가질 수 있다. 활성화 값들은 신경망에 입력되는 이미지의 8×8 부분, 즉 이미지의 8×8 부분에 대응하는 값들의 시퀀스를 나타낼 수 있다. 대안으로, 활성화 값들의 8×8 어레이는 다른 입력 텐서의 8×8 부분, 예컨대 신경망의 선행 계층의 출력에 대응하는 입력 텐서를 나타낼 수 있다.

[0086] 도 10의 부분(a)에서, 8×8 입력 텐서는, 1의 스트라이드를 가지며 그 외에는 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층과 동일한 컨볼루션 신경망 계층을 사용하여 처리된다. 따라서 부분(a)에 도시된 가중치들의 4×4 커널은 입력 텐서의 처음 4개의 행들과 처음 4개의 열들에 대응하는 입력 텐서의 엘리먼트들에 먼저 적용될 수 있다(값들은 도시되지 않음). 프로세스의 결과는 결과적인 컨볼브된 텐서에서 제1 엘리먼트, 즉 도 10의 부분(a)에 도시된 결과적인 컨볼브된 텐서의 "a" 엘리먼트일 수 있다.

[0087] 입력 텐서의 처리는 4의 지정된 스트라이드 대신에 1의 스트라이드를 갖는 컨볼루션 신경망 계층을 사용하여 수행되기 때문에, 부분(a)에 도시된 가중치들의 4×4 세트가 다음에, 활성화 값 어레이의 처음 4개의 행들 및 입력 텐서의 제2 열 내지 제5 열에 대응하는 입력 텐서의 엘리먼트들(값들은 도시되지 않음)에 적용될 수 있다. 처리의 결과는 컨볼브된 텐서의 제2 엘리먼트, 즉 도 10의 부분(a)에 도시된 컨볼루션 결과의 "b" 엘리먼트이다. 1의 스트라이드를 사용하여 활성화 값 어레이에 가중치들의 4×4 세트를 적용함으로써, 즉 가중치들의 4×4 세트를 열 방향과 행 방향 모두로 점진적으로 활성화 값 어레이에 적용함으로써 프로세스가 반복될 수 있다. 처리는 도 10의 부분(a)에 도시된 8×8 컨볼브된 텐서를 야기한다.

[0088] 그 다음, 도 9의 부분(b)에 도시된 바와 같이, 컨볼브된 텐서와 마스킹 텐서 사이에서 엘리먼트별 곱셈이 수행되어 수정된 컨볼브된 텐서를 얻는다. 마스킹 텐서의 크기는 입력 텐서의 크기 또는 컨볼브된 텐서의 크기에 기초하여 결정되는데, 이는 1의 스트라이드를 갖는 컨볼루션 신경망 계층을 사용하는 도 10의 부분(a)에서의 처리로 인해 일반적으로 동일할 것이다. 마스킹 텐서는 입력 텐서가 지정된 스트라이드를 갖는 컨볼루션 신경망

계층을 사용하여 처리되었다면 생성될 값들에 대응하는 위치들에 단위 값들, 즉 1들을 포함한다. 그 다음, 일반적으로, 마스킹 텐서의 단위 값 항목들의 위치들은 컨볼루션 신경망 계층의 지정된 스트라이드에 좌우된다. 도 10의 예에서, 컨볼루션 신경망 계층은 4의 스트라이드를 갖기 때문에, 마스킹 텐서는 열 방향과 행 방향 모두에서 네 번째 위치마다 단위 값들을 포함할 것이다. 마스킹 텐서의 다른 항목들에는 0 값들이 할당되어, 컨볼브된 텐서와 마스킹 텐서의 엘리먼트별 곱셈은 입력 텐서가 지정된 스트라이드를 갖는 컨볼루션 신경망으로 처리되었다면 생성되지 않았을 모든 값들을 0으로 설정되게 할 것이다.

[0089] 컨볼브된 텐서와 마스킹 텐서의 엘리먼트별 곱셈이 수행되어 수정된 컨볼브된 텐서를 생성한다. 도 10에 도시된 바와 같이, 엘리먼트별 곱셈 후에, 컨볼브된 텐서의 네 번째 엘리먼트마다 유지되고, 컨볼브된 텐서의 엘리먼트들 중 나머지는 마스킹 행렬의 대응하는 0 값 엘리먼트와의 곱셈으로 인해 0들이 된다. 따라서 8×8 컨볼브된 텐서의 엘리먼트들 중 4개의 엘리먼트들만이 0이 아닌 값을 유지한다.

[0090] 일부 구현들에서, 컨볼브된 텐서의 엘리먼트들에 비-단위(non-unity) 팩터들을 먼저 곱하고, 이어서 그러한 엘리먼트들에 제2 비-단위 팩터들을 곱함으로써 유사한 결과가 얻어질 수 있다. 예컨대, 마스킹 텐서는 입력 텐서가 지정된 스트라이드를 갖는 컨볼루션 신경망 계층을 사용하여 처리되었다면 생성될 값들에 대응하는 위치들에 2들(또는 다른 값)을 포함할 수 있다. 따라서 위의 예에 따라, 컨볼브된 텐서와 마스킹 텐서의 엘리먼트별 곱셈은 컨볼브된 텐서의 네 번째 엘리먼트마다 2배가 되고 나머지 엘리먼트들은 0인 수정된 컨볼브된 텐서를 생성한다. 이어서, 수정된 컨볼브된 텐서와 1/2(또는 다른 값의 역)의 스칼라 곱셈이 수행될 수 있다. 대안으로, 수정된 컨볼브된 텐서와 제2 마스킹 텐서의 엘리먼트별 곱셈이 수행될 수 있으며, 여기서 제2 마스킹 텐서는 입력 텐서가 지정된 스트라이드를 갖는 컨볼루션 신경망 계층을 사용하여 처리되었다면 생성될 값들에 대응하는 위치들에 1/2의 값들을 포함한다.

[0091] 이후에, 도 10의 부분(c)에서 수정된 컨볼루션 결과 어레이에 대해 최대 풀링이 수행된다. 최대 풀링의 결과는 입력 텐서가 4의 스트라이드를 갖는 컨볼루션 신경망 계층에 의해 처리되었다면 얻어질 결과와 동일하다. 도 6의 프로세스를 사용하여, 수정된 컨볼브된 텐서에 대해 최대 풀링이 수행되어, 수정된 컨볼브된 텐서의 각각의 4×4 윈도우의 최대 값을 식별한다. 그 다음, 최대 풀링의 결과가 4의 스트라이드를 갖는 컨볼루션 신경망 계층의 출력 텐서로서 저장된다. 입력 텐서가 8×8 어레이였기 때문에, 4의 스트라이드를 갖는 신경망 계층에 의한 처리는 2×2 출력 어레이가 된다. 2×2 출력 어레이는 도 3의 통합 버퍼(308)에, 예컨대 래스터 순서로 저장될 수 있다. 2×2 출력 어레이의 값들은 신경망의 후속 계층에 대한 입력들로서 제공될 수 있다.

[0092] 본 명세서에서 설명한 기능적 동작들 및 청구 대상의 실시예들은 디지털 전자 회로에, 유형적으로 구현된 컴퓨터 소프트웨어 또는 펌웨어에, 본 명세서에 개시된 구조들 및 이들의 구조적 등가물들을 포함하는 컴퓨터 하드웨어에, 또는 이들 중 하나 이상에 대한 결합들에 구현될 수 있다. 본 명세서에서 설명한 청구 대상의 실시예들은 하나 이상의 컴퓨터 프로그램들, 즉 데이터 처리 장치에 의한 실행을 위해 또는 그 동작을 제어하기 위해 유형의 비일시적 프로그램 운반체(carrier) 상에 인코딩되는 컴퓨터 프로그램 명령들의 하나 이상의 모듈들로서 구현될 수 있다. 대안으로 또는 추가로, 프로그램 명령들은 데이터 처리 장치에 의한 실행을 위해 적당한 수신기 장치로의 송신을 위한 정보를 인코딩하도록 발생되는 인공적으로 발생한 전파 신호, 예를 들면 기계 발생 전기, 광 또는 전자기 신호에 대해 인코딩될 수 있다. 컴퓨터 저장 매체는 기계 판독 가능 저장 디바이스, 기계 판독 가능 저장 기판, 랜덤 또는 직렬 액세스 메모리 디바이스, 또는 이들 중 하나 이상에 대한 결합일 수 있다.

[0093] "데이터 처리 장치"라는 용어는 예로서 프로그래밍 가능 프로세서, 컴퓨터 또는 다수의 프로세서들이나 컴퓨터들을 포함하여, 데이터를 처리하기 위한 모든 종류들의 장치, 디바이스들 및 기계들을 포괄한다. 장치는 특수 목적 로직 회로, 예를 들면 필드 프로그래밍 가능 게이트 어레이(FPGA: field programmable gate array) 또는 주문형 집적 회로(ASIC: application specific integrated circuit)를 포함할 수 있다. 장치는 또한 하드웨어뿐만 아니라, 해당 컴퓨터 프로그램에 대한 실행 환경을 생성하는 코드, 예를 들면 프로세서 펌웨어, 프로토콜 스택, 데이터베이스 관리 시스템, 운영 시스템, 또는 이들의 하나 이상에 대한 결합을 구성하는 코드를 포함할 수 있다.

[0094] (프로그램, 소프트웨어, 소프트웨어 애플리케이션, 모듈, 소프트웨어 모듈, 스크립트 또는 코드로 또한 지칭되거나 이로서 설명될 수 있는) 컴퓨터 프로그램은 컴파일링된 또는 해석된 언어들, 또는 서술적 또는 절차적 언어들을 포함하는 임의의 형태의 프로그래밍 언어로 작성될 수 있고, 이는 독립형 프로그램으로서 또는 모듈, 컴포넌트, 서브루틴, 또는 컴퓨팅 환경에 사용하기에 적당한 다른 유닛으로서의 형태를 포함하는 임의의 형태로 전개될 수 있다. 컴퓨터 프로그램은 파일 시스템 내의 파일에 대응할 수 있지만 필요한 것은 아니다. 프로그

램은 다른 프로그램들 또는 데이터, 예를 들면 마크업 언어 문서에 저장된 하나 이상의 스크립트들을 보유하는 파일의 일부에, 해당 프로그램에 전용된 단일 파일에, 또는 다수의 조정된 파일들, 예를 들면 하나 이상의 모듈들, 하위 프로그램들, 또는 코드의 부분들을 저장하는 파일들에 저장될 수 있다. 컴퓨터 프로그램은 하나의 컴퓨터 상에서 또는 한 사이트에 로케이팅되거나 다수의 사이트들에 걸쳐 분포되어 통신 네트워크에 의해 상호 접속되는 다수의 컴퓨터들 상에서 실행되도록 전개될 수 있다.

[0095] 본 명세서에서 설명한 프로세스들 및 로직 플로우들은 입력 데이터에 대해 동작하여 출력을 발생시킴으로써 기능들을 수행하도록 하나 이상의 컴퓨터 프로그램들을 실행하는 하나 이상의 프로그래밍 가능 컴퓨터들에 의해 수행될 수 있다. 프로세스들 및 논리 흐름들은 또한 특수 목적 로직 회로, 예를 들면 필드 프로그래밍 가능 게이트 어레이(FPGA: field programmable gate array) 또는 주문형 집적 회로(ASIC: application specific integrated circuit)에 의해 수행될 수 있으며, 장치가 또한 이로서 구현될 수 있다.

[0096] 컴퓨터 프로그램의 실행에 적합한 컴퓨터들은 범용 또는 특수 목적 마이크로프로세서들 또는 둘 다, 또는 다른 어떤 종류의 중앙 처리 유닛을 포함하며, 예로서 이에 기반할 수 있다. 일반적으로, 중앙 처리 유닛은 판독 전용 메모리 또는 랜덤 액세스 메모리 또는 둘 다로부터 명령들 및 데이터를 수신할 것이다. 컴퓨터의 필수 엘리먼트들은 명령들을 수행 또는 실행하기 위한 중앙 처리 유닛 그리고 명령들 및 데이터를 저장하기 위한 하나 이상의 메모리 디바이스들이다. 일반적으로, 컴퓨터는 또한 데이터를 저장하기 위한 하나 이상의 대용량 저장 디바이스들, 예를 들면, 자기, 마그네토 광 디스크들, 또는 광 디스크들을 포함하거나, 이들로부터 데이터를 수신하고 또는 이들에 데이터를 전송하도록, 또는 둘 다를 위해 동작 가능하게 연결될 것이다. 그러나 컴퓨터가 이러한 디바이스들을 가질 필요는 없다. 더욱이, 컴퓨터는 다른 디바이스, 몇 가지만 예로 들자면, 예를 들어 모바일 전화, 개인용 디지털 보조기기(PDA: personal digital assistant), 모바일 오디오 또는 비디오 플레이어, 게임 콘솔, 글로벌 포지셔닝 시스템(GPS: Global Positioning System) 수신기, 또는 휴대용 저장 디바이스, 예를 들면 범용 직렬 버스(USB: universal serial bus) 플래시 드라이브에 내장될 수 있다.

[0097] 컴퓨터 프로그램 명령들 및 데이터를 저장하기에 적합한 컴퓨터 판독 가능 매체들은 예로서 반도체 메모리 디바이스들, 예를 들면 EPROM, EEPROM, 및 플래시 메모리 디바이스들; 자기 디스크들, 예를 들면 내부 하드 디스크들 또는 착탈식 디스크들; 마그네토 광 디스크들; 그리고 CD ROM 및 DVD-ROM 디스크들을 포함하는 모든 형태들의 비휘발성 메모리, 매체들 및 메모리 디바이스들을 포함한다. 프로세서 및 메모리는 특수 목적 로직 회로에 의해 보완되거나 특수 목적 로직 회로에 포함될 수 있다.

[0098] 사용자와의 상호 작용을 전송하기 위해, 본 명세서에서 설명한 청구 대상의 실시예들은 사용자에게 정보를 디스플레이하기 위한 디스플레이 디바이스, 예를 들면 음극선관(CRT: cathode ray tube) 또는 액정 디스플레이(LCD: liquid crystal display) 모니터 및 사용자가 컴퓨터에 입력을 전송할 수 있게 하는 키보드와 포인팅 디바이스, 예를 들면 마우스 또는 트랙볼을 갖는 컴퓨터 상에 구현될 수 있다. 다른 종류들의 디바이스들이 사용자와의 상호 작용을 전송하는 데 역시 사용될 수 있는데; 예를 들어, 사용자에게 제공되는 피드백은 임의의 형태의 감각 피드백, 예컨대 시각 피드백, 청각 피드백 또는 촉각 피드백일 수 있고; 사용자로부터의 입력은 음향, 음성 또는 촉각 입력을 포함하는 임의의 형태로 수신될 수 있다. 추가로, 컴퓨터는 사용자에 의해 사용되는 디바이스에 문서들을 전송하고 이러한 디바이스로부터 문서들을 수신함으로써; 예를 들어, 웹 브라우저로부터 수신된 요청들에 대한 응답으로 사용자의 클라이언트 디바이스 상의 웹 브라우저에 웹 페이지들을 전송함으로써 사용자와 상호 작용할 수 있다.

[0099] 본 명세서에서 설명한 청구 대상의 실시예들은 예를 들어, 데이터 서버로서 백엔드 컴포넌트를 포함하는, 또는 미들웨어 컴포넌트, 예를 들어, 애플리케이션 서버를 포함하는, 또는 프론트엔드 컴포넌트, 예를 들어, 본 명세서에서 설명한 청구 대상의 구현과 사용자가 상호 작용할 수 있게 하는 그래픽 사용자 인터페이스 또는 웹 브라우저를 갖는 클라이언트 컴퓨터를 포함하는 컴퓨팅 시스템, 또는 이러한 하나 이상의 백엔드, 미들웨어 또는 프론트엔드 컴포넌트들의 임의의 결합으로 구현될 수 있다. 시스템의 컴포넌트들은 임의의 형태 또는 매체의 디지털 데이터 통신, 예를 들면 통신 네트워크에 의해 상호 접속될 수 있다. 통신 네트워크들의 예들은 근거리 네트워크("LAN") 및 광역 네트워크("WAN"), 예를 들면 인터넷을 포함한다.

[0100] 컴퓨팅 시스템은 클라이언트들 및 서버들을 포함할 수 있다. 클라이언트 및 서버는 일반적으로 서로로부터 원거리이며 일반적으로 통신 네트워크를 통해 상호 작용한다. 클라이언트와 서버의 관계는, 각각의 컴퓨터들 상에서 구동되며 서로 클라이언트-서버 관계를 갖는 컴퓨터 프로그램들에 의해 발생한다.

[0101] 본 명세서는 많은 특정 구현 세부사항들을 포함하지만, 이들은 청구될 수 있는 것의 또는 임의의 발명의 범위에 대한 한정들로서가 아니라, 그보다는 특정 발명들의 특정 실시예들에 특정할 수 있는 특징들의 설명으로서 해석

되어야 한다. 개별 실시예들과 관련하여 본 명세서에 설명되는 특정 특징들은 또한 단일 실시예로 결합하여 구현될 수 있다. 반대로, 단일 실시예와 관련하여 설명되는 다양한 특징들은 또한 다수의 실시예들로 개별적으로 또는 임의의 적절한 하위 결합으로 구현될 수 있다. 아울러, 특징들이 특정한 결합들로 작용하는 것으로 앞서 설명되고 심지어 초기에 이와 같이 청구될 수 있다 하더라도, 어떤 경우들에는 청구된 결합으로부터의 하나 이상의 특징들이 그 결합으로부터 삭제될 수 있고, 청구된 결합은 하위 결합 또는 하위 결합의 변형에 관련될 수 있다.

[0102] 마찬가지로, 동작들이 특정 순서로 도면들에 도시되지만, 이는 바람직한 결과들을 달성하기 위해 이러한 동작들이 도시된 특정 순서로 또는 순차적인 순서로 수행될 것을, 또는 예시된 모든 동작들이 수행될 것을 요구하는 것으로 이해되지는 않아야 한다. 특정 상황들에서는, 다중 작업 및 병렬 처리가 유리할 수 있다. 더욱이, 앞서 설명한 실시예들에서 다양한 시스템 모듈들 및 컴포넌트들의 분리는 모든 실시예들에서 이러한 분리를 필요로 하는 것으로 이해되지 않아야 하며, 설명한 프로그램 컴포넌트들 및 시스템들은 일반적으로 단일 소프트웨어 제품으로 함께 통합되거나 다수의 소프트웨어 제품들로 패키지화될 수 있다고 이해되어야 한다.

[0103] 추가 구현들은 다음의 예들로 요약된다:

[0104] 예 1: 방법은: 하드웨어 회로 상에서 신경망을 처리하라는 요청을 수신하는 단계 - 신경망은 1보다 더 큰 스트라이드를 갖는 제1 컨볼루션 신경망 계층을 포함함 -; 및 응답으로, 하드웨어 회로에 의해 실행될 때 하드웨어 회로로 하여금: 제1 텐서를 생성하기 위해, 1과 같은 스트라이드를 갖지만 그 외에는 제1 컨볼루션 신경망 계층과 동일한 제2 컨볼루션 신경망 계층을 사용하여 제1 컨볼루션 신경망 계층에 대한 입력 텐서를 처리하는 동작; 제2 텐서를 생성하기 위해, 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트들을 0으로 설정하는 동작; 및 계층 출력 텐서를 생성하기 위해, 제2 텐서에 대한 최대 풀링을 수행하는 동작을 포함하는 동작들을 수행함으로써, 신경망에 의한 입력 텐서의 처리 동안, 제1 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서를 생성하게 하는 명령들을 생성하는 단계를 포함한다.

[0105] 예 2: 예 1의 방법에서, 제1 텐서의 엘리먼트들을 0으로 설정하는 것은: 제1 텐서의 엘리먼트들의 서브세트를 0과 곱하는 것; 그리고 서브세트에 포함되지 않은, 제1 텐서의 엘리먼트들을 1과 곱하는 것을 포함한다.

[0106] 예 3: 예 1의 방법에서, 제1 텐서의 엘리먼트들을 0으로 설정하는 것은: 제2 텐서를 생성하기 위해, 마스킹 텐서와 제1 텐서의 엘리먼트별 곱셈을 수행하는 것을 포함하며, 마스킹 텐서는 (i) 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트에 대응하는 마스킹 텐서의 각각의 엘리먼트 위치에서 0들을, 그리고 (ii) 마스킹 텐서의 각각의 다른 엘리먼트 위치에서 1들을 포함한다.

[0107] 예 4: 예 3의 방법에서, 마스킹 텐서는 하드웨어 회로에 의해 액세스 가능한 메모리에 저장되고, 마스킹 텐서와 제1 텐서의 엘리먼트별 곱셈은 하드웨어 회로에 포함되는 하드웨어에 구현된 벡터 계산 유닛에 의해 수행된다.

[0108] 예 5: 예 1의 방법에서, 제1 텐서의 엘리먼트들을 0으로 설정하는 것은, 수정된 제1 텐서를 생성하기 위해, 제1 마스킹 텐서와 제1 텐서의 엘리먼트별 곱셈을 수행하는 것 - 제1 마스킹 텐서는 (i) 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트에 대응하는 마스킹 텐서의 각각의 엘리먼트 위치에서 0들을, 그리고 (ii) 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되었을 제1 텐서의 엘리먼트에 대응하는 마스킹 텐서의 각각의 엘리먼트 위치에서 각각 0이 아닌 값을 포함함 -; 그리고 제2 마스킹 텐서와 수정된 제1 텐서의 엘리먼트별 곱셈을 수행하는 것을 포함하며, 제2 마스킹 텐서는 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성될 제1 텐서의 엘리먼트에 대응하는 각각의 엘리먼트 위치에서의, 제1 마스킹 텐서의 각각의 0이 아닌 값의 역을 포함한다.

[0109] 예 6: 예 1 내지 예 5 중 한 예의 방법에서, 최대 풀링을 수행하는 것은, 제1 컨볼루션 신경망 계층의 스트라이드에 의해 정의되는 제2 텐서의 하나 이상의 윈도우들 각각에 대해 윈도우 내의 엘리먼트들 중 최대 값 엘리먼트를 획득하는 것을 포함한다.

[0110] 예 7: 예 6의 방법에서, 제2 텐서의 하나 이상의 윈도우들 각각은 컨볼루션 신경망 계층의 스트라이드에 대응하는 치수들을 갖는 직사각형 윈도우이고, 제2 텐서의 서로 다른 엘리먼트들을 포함한다.

[0111] 예 8: 예 1 내지 예 7 중 한 예의 방법에서, 최대 풀링을 수행하는 것은, 제2 텐서의 엘리먼트들의 하나 이상의 서브세트들 각각에 대해 서브세트의 최대 값 엘리먼트를 획득하는 것을 포함한다.

[0112] 예 9: 예 1 내지 예 8 중 한 예의 방법에서, 제2 텐서에 대해 수행된 최대 풀링은 하드웨어 회로의 풀링 회로에 의해 수행된다.

[0113] 예 10: 예 1 내지 예 9 중 한 예의 방법에서, 컨볼루션 신경망 계층은 신경망에서의 제1 신경망 계층이고, 입력 텐서는 디지털 이미지의 픽셀들에 대응하는 엘리먼트들을 포함하는 디지털 이미지의 표현이다.

[0114] 예 11: 예 1 내지 예 10 중 한 예의 방법에서, 입력 텐서는 하드웨어 회로의 통합 버퍼에 저장되고, 제2 컨볼루션 신경망 계층의 가중치들은 하드웨어 회로의 동적 메모리에 저장되며, 제2 컨볼루션 신경망 계층을 사용하여 제1 컨볼루션 신경망 계층에 대한 입력 텐서를 처리하는 것은: 통합 버퍼로부터, 하드웨어로 구현되는 하드웨어 회로의 행렬 계산 유닛으로 입력 텐서를 전송하는 것; 동적 메모리로부터 하드웨어 회로의 행렬 계산 유닛으로 제2 컨볼루션 신경망 계층의 가중치들을 전송하는 것; 그리고 제1 텐서를 생성하기 위해, 하드웨어 회로의 행렬 계산 유닛에 의해 제2 컨볼루션 신경망 계층의 가중치들을 사용하여 입력 텐서를 처리하는 것을 포함한다.

[0115] 예 12: 시스템은: 하드웨어 회로; 및 하드웨어 회로에 의해 실행될 때 하드웨어 회로로 하여금: 제1 텐서를 생성하기 위해, 1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층에 대한 입력 텐서를, 1과 같은 스트라이드를 갖지만 그 외에는 상기 컨볼루션 신경망 계층과 동일한 제2 컨볼루션 신경망 계층을 사용하여 처리하는 동작; 제2 텐서를 생성하기 위해, 제2 컨볼루션 신경망 계층이 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트들을 0으로 설정하는 동작; 및 계층 출력 텐서를 생성하기 위해, 제2 텐서에 대한 최대 풀링을 수행하는 동작을 포함하는 동작들을 수행하게 하도록 동작 가능한 명령들을 저장하는 하나 이상의 저장 디바이스들을 포함한다.

[0116] 예 13: 예 12의 시스템에서, 제1 텐서의 엘리먼트들을 0으로 설정하는 것은: 제2 텐서를 생성하기 위해, 마스킹 텐서와 제1 텐서의 엘리먼트별 곱셈을 수행하는 것을 포함하며, 마스킹 텐서는 (ⅰ) 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트에 대응하는 마스킹 텐서의 각각의 엘리먼트 위치에서 0들을, 그리고 (ⅱ) 마스킹 텐서의 각각의 다른 엘리먼트 위치에서 1들을 포함한다.

[0117] 예 14: 예 13의 시스템에서, 마스킹 텐서는 하드웨어 회로에 의해 액세스 가능한 메모리에 저장되고, 마스킹 텐서와 제1 텐서의 엘리먼트별 곱셈은 하드웨어 회로에 포함되는 하드웨어에 구현된 벡터 계산 유닛에 의해 수행된다.

[0118] 예 15: 예 12 내지 예 14 중 한 예의 시스템에서, 최대 풀링을 수행하는 것은, 제1 컨볼루션 신경망 계층의 스트라이드에 의해 정의되는 제2 텐서의 하나 이상의 윈도우들 각각에 대해 윈도우 내의 엘리먼트들 중 최대 값 엘리먼트를 획득하는 것을 포함한다.

[0119] 예 16: 예 15의 시스템에서, 제2 텐서의 하나 이상의 윈도우들 각각은 컨볼루션 신경망 계층의 스트라이드에 대응하는 치수들을 갖는 직사각형 윈도우이고, 제2 텐서의 서로 다른 엘리먼트들을 포함한다.

[0120] 예 17: 예 12 내지 예 16 중 한 예의 시스템에서, 제2 텐서에 대해 수행된 최대 풀링은 하드웨어 회로의 풀링 회로에 의해 수행된다.

[0121] 예 18: 예 12 내지 예 17의 시스템에서, 컨볼루션 신경망 계층은 신경망에서의 제1 신경망 계층이고, 입력 텐서는 디지털 이미지의 픽셀들에 대응하는 엘리먼트들을 포함하는 디지털 이미지의 표현이다.

[0122] 예 19: 예 12 내지 예 18 중 한 예의 시스템에서, 입력 텐서는 하드웨어 회로의 통합 버퍼에 저장되고, 제2 컨볼루션 신경망 계층의 가중치들은 하드웨어 회로의 동적 메모리에 저장되며, 제2 컨볼루션 신경망 계층을 사용하여 제1 컨볼루션 신경망 계층에 대한 입력 텐서를 처리하는 것은: 통합 버퍼로부터, 하드웨어로 구현되는 하드웨어 회로의 행렬 계산 유닛으로 입력 텐서를 전송하는 것; 동적 메모리로부터 하드웨어 회로의 행렬 계산 유닛으로 제2 컨볼루션 신경망 계층의 가중치들을 전송하는 것; 그리고 제1 텐서를 생성하기 위해, 하드웨어 회로의 행렬 계산 유닛에 의해 제2 컨볼루션 신경망 계층의 가중치들을 사용하여 입력 텐서를 처리하는 것을 포함한다.

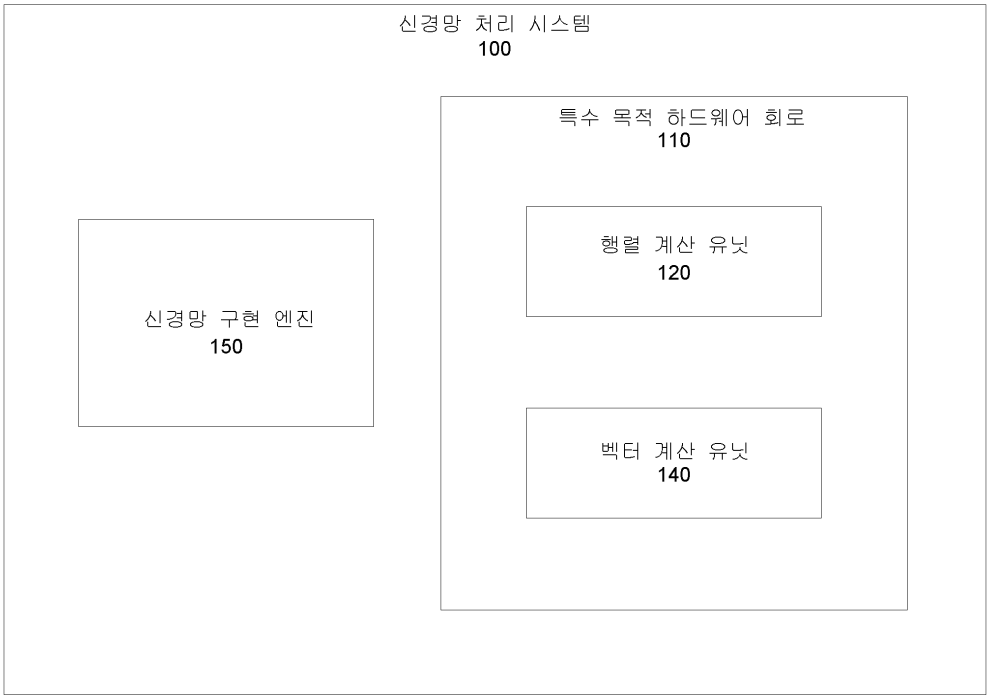
[0123] 예 20: 컴퓨터 프로그램으로 인코딩된 컴퓨터 판독 가능 저장 디바이스로서, 이 프로그램은 하나 이상의 컴퓨터들에 의해 실행된다면, 하나 이상의 컴퓨터들로 하여금: 하드웨어 회로 상에서 신경망을 처리하라는 요청을 수신하는 동작 — 신경망은 1보다 더 큰 스트라이드를 갖는 제1 컨볼루션 신경망 계층을 포함함 —; 그리고 응답으로, 하드웨어 회로에 의해 실행될 때 하드웨어 회로로 하여금: 제1 텐서를 생성하기 위해, 1과 같은 스트라이드를 갖지만 그 외에는 제1 컨볼루션 신경망 계층과 동일한 제2 컨볼루션 신경망 계층을 사용하여 제1 컨볼루션

신경망 계층에 대한 입력 텐서를 처리하는 동작; 제2 텐서를 생성하기 위해, 제2 컨볼루션 신경망 계층이 제1 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 제1 텐서의 엘리먼트들을 0으로 설정하는 동작; 및 계층 출력 텐서를 생성하기 위해, 제2 텐서에 대한 최대 풀링을 수행하는 동작을 포함하는 동작들을 수행함으로써, 신경망에 의한 입력 텐서의 처리 동안, 제1 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서를 생성하게 하는 명령들을 생성하는 동작을 포함하는 동작들을 수행하게 한다.

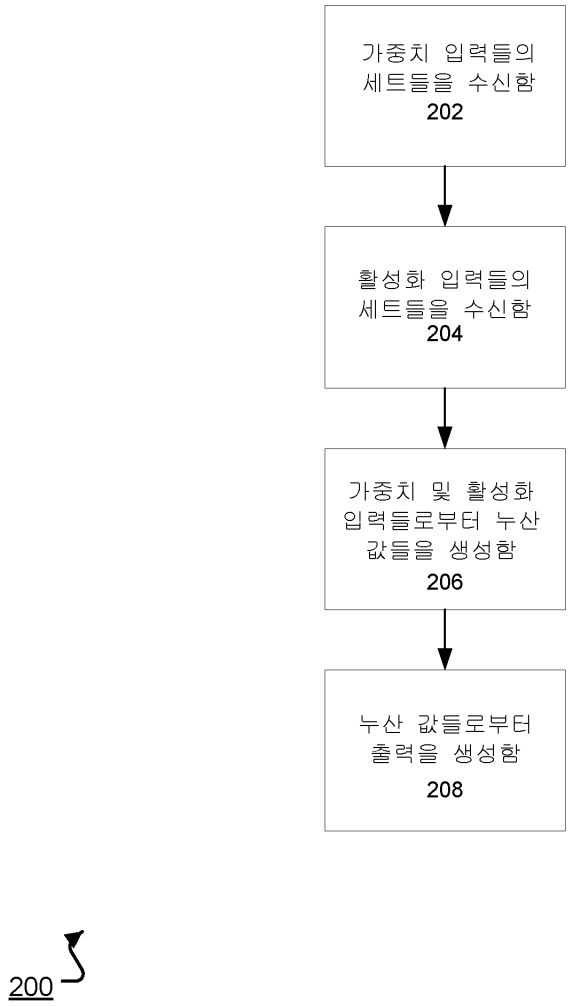
[0124] 청구 대상의 특정 실시예들이 설명되었다. 다른 실시예들이 다음의 청구항들의 범위 내에 있다. 예를 들어, 청구항들에서 언급되는 동작들은 다른 순서로 수행되며 그럼에도 바람직한 결과들을 달성할 수 있다. 일례로, 첨부 도면들에 도시된 프로세스들은 바람직한 결과들을 달성하기 위해 반드시 도시된 특정 순서 또는 순차적인 순서를 필요로 하는 것은 아니다. 특정 구현들에서는, 다중 작업 및 병렬 처리가 유리할 수도 있다.

도면

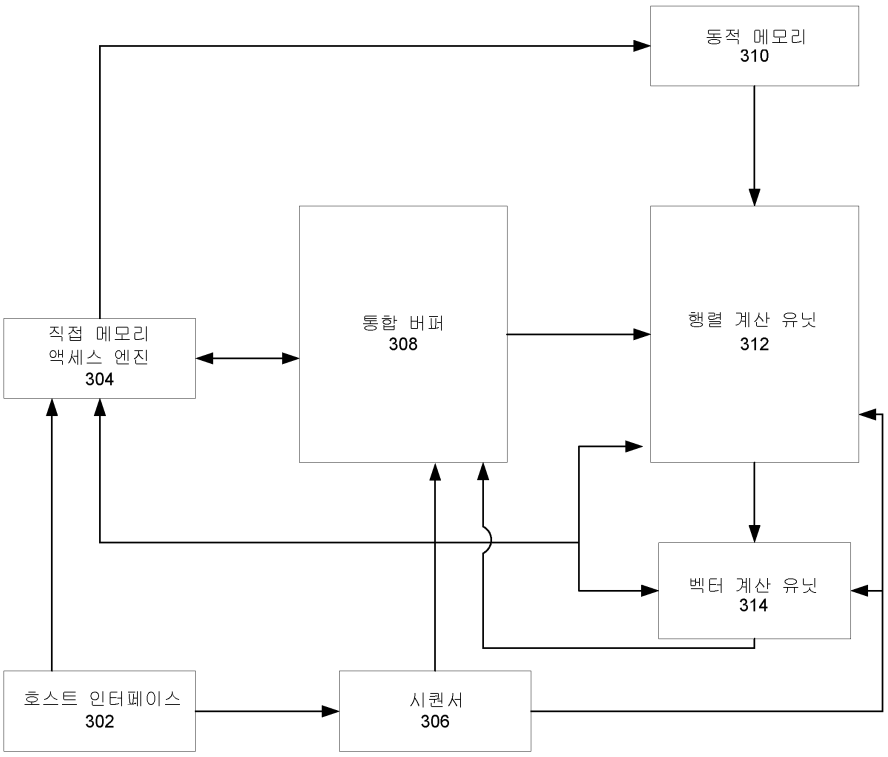
도면1



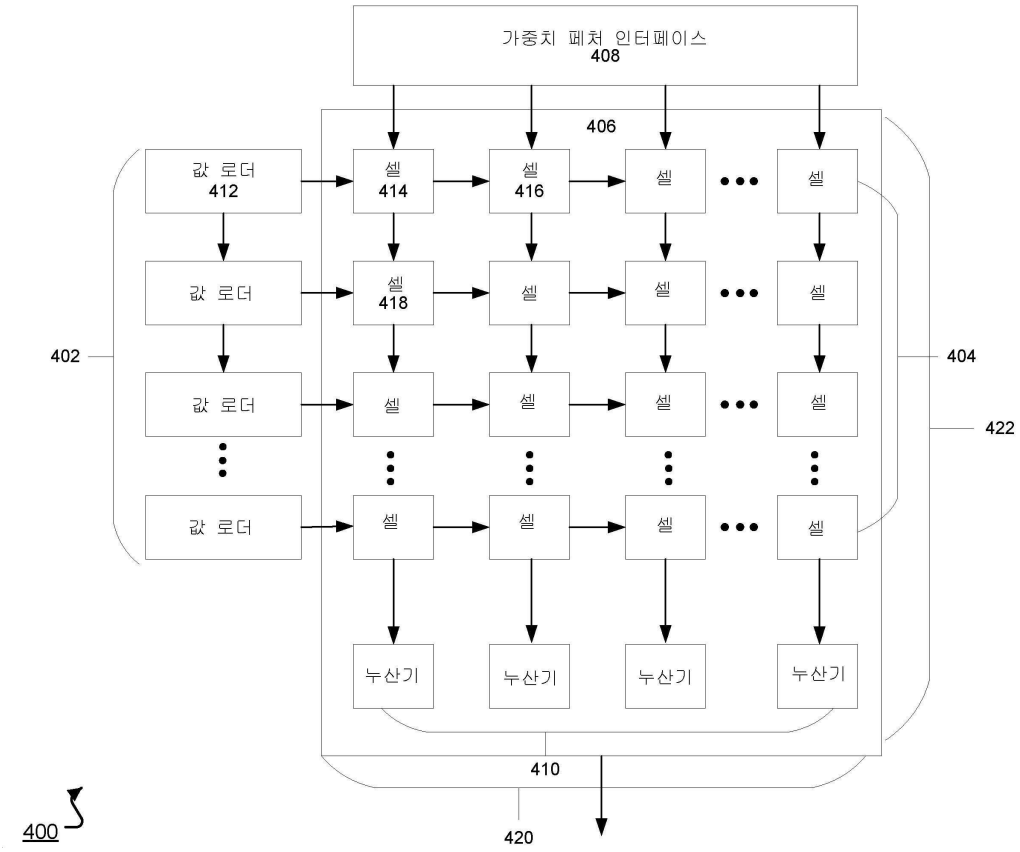
도면2


가중치 입력들의 세트들을 수신함
202
활성화 입력들의 세트들을 수신함
204
가중치 및 활성화 입력들로부터 누산 값들을 생성함
206
누산 값들로부터 출력을 생성함
208
200

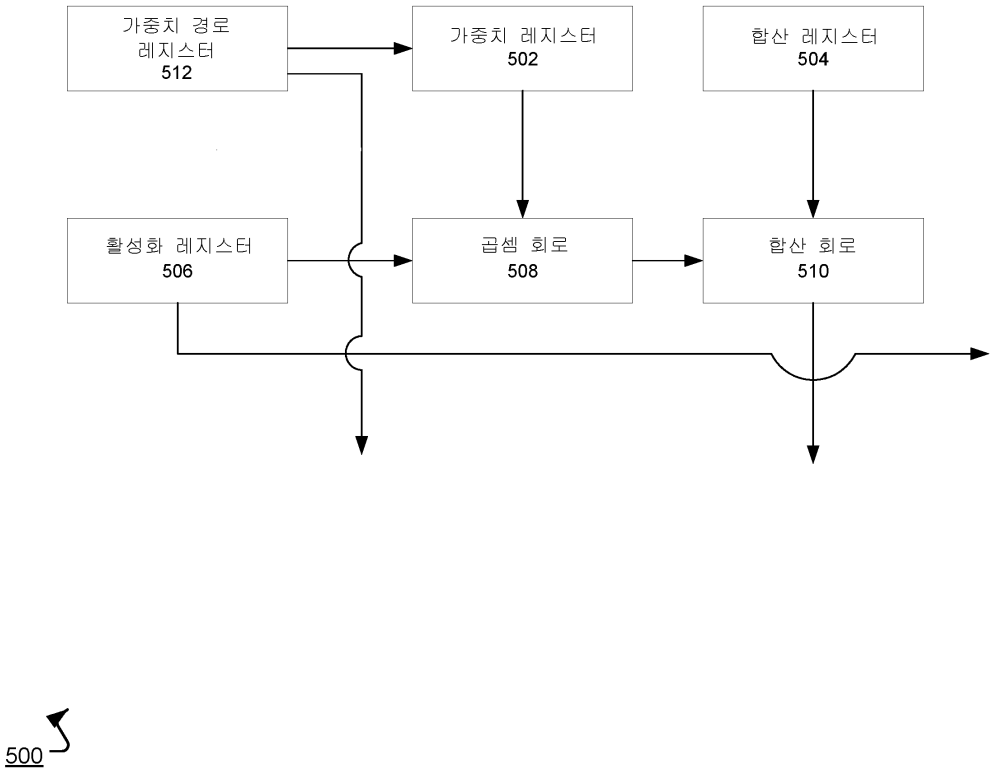
도면3


동적 메모리
310
통합 버퍼
308
행렬 계산 유닛
312
직접 메모리
액세스 엔진
304
벡터 계산 유닛
314
호스트 인터페이스
302
시퀀서
306
300

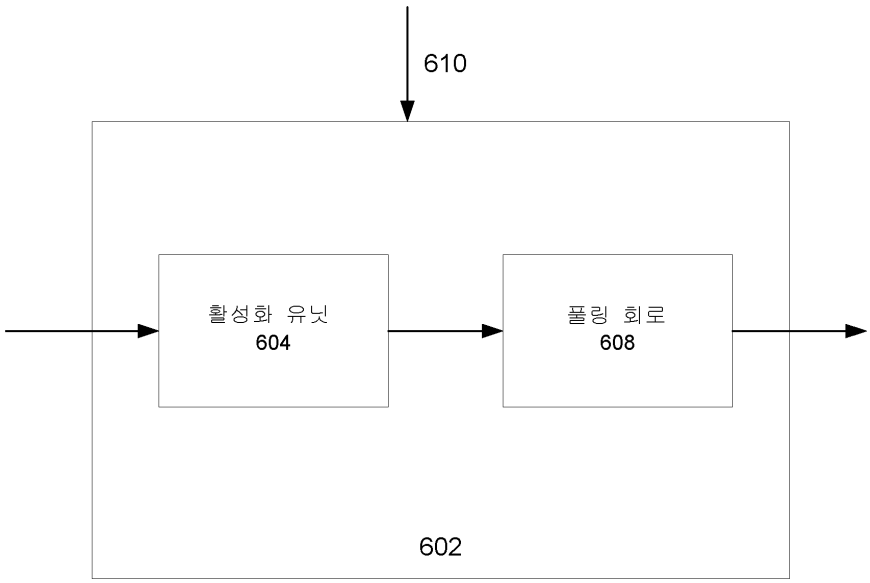
도면4


가중치 페처 인터페이스
408
406
값 로더
412
셀
414
셀
416
셀
셀
값 로더
셀
418
셀
셀
셀
402
404
값 로더
셀
셀
셀
셀
422
값 로더
셀
셀
셀
셀
누산기
누산기
누산기
누산기
410
420
400

도면5

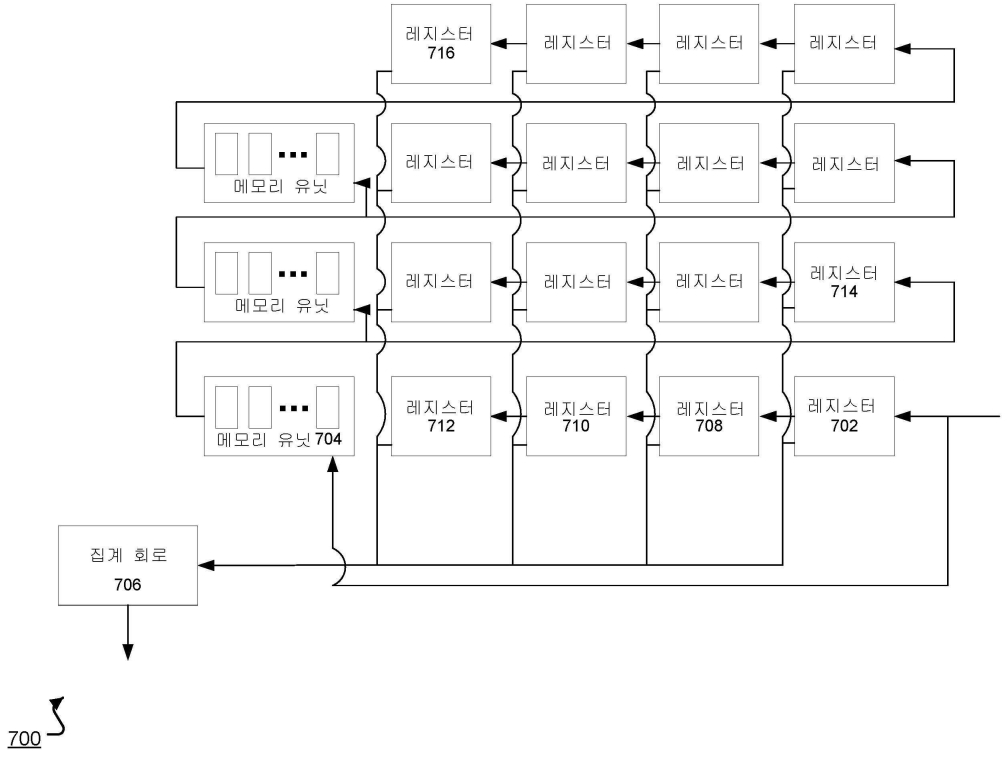

가중치 경로
레지스터
512
가중치 레지스터
502
합산 레지스터
504
활성화 레지스터
506
곱셈 회로
508
합산 회로
510
500

도면6



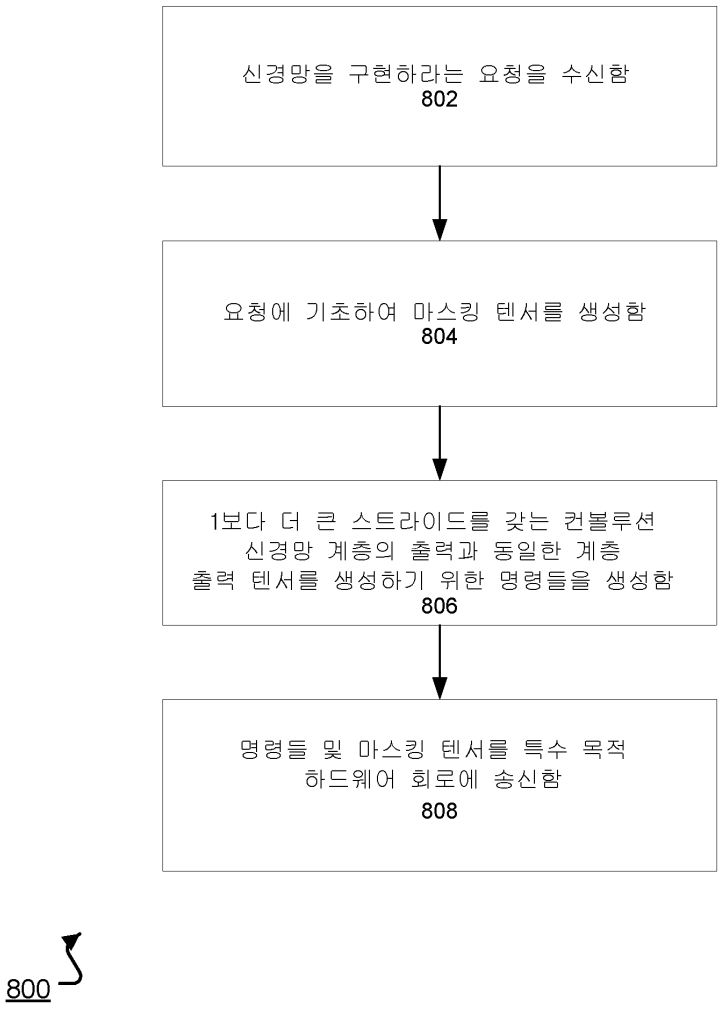
600

도면7

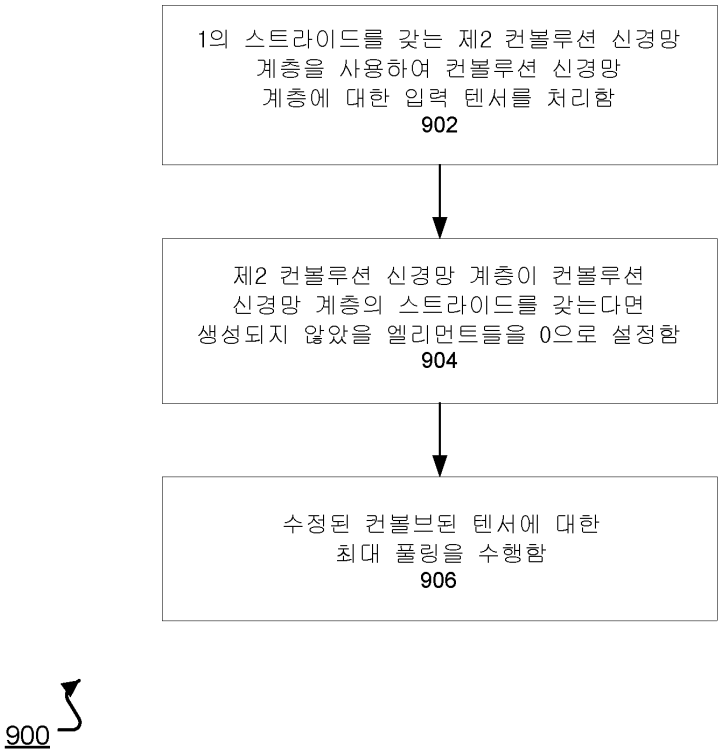


700

도면8

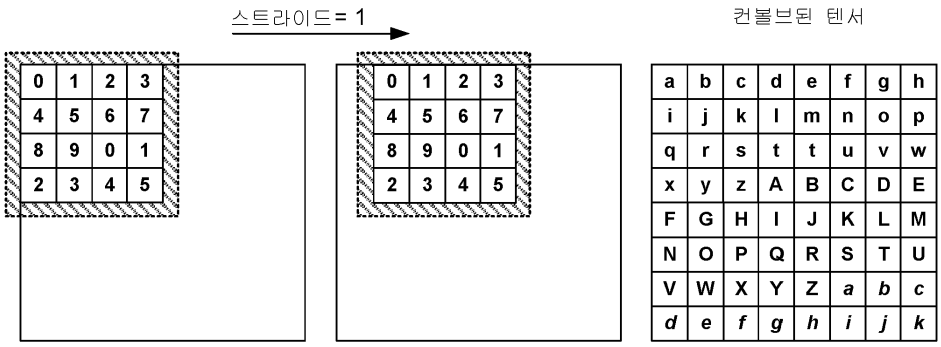

신경망을 구현하라는 요청을 수신함
802
요청에 기초하여 마스킹 텐서를 생성함
804
1보다 더 큰 스트라이드를 갖는 컨볼루션 신경망 계층의 출력과 동일한 계층 출력 텐서를 생성하기 위한 명령들을 생성함
806
명령들 및 마스킹 텐서를 특수 목적 하드웨어 회로에 송신함
808
800

도면9

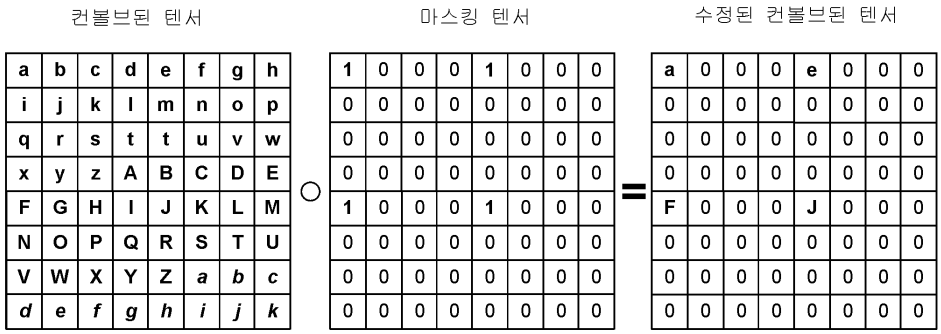

1의 스트라이드를 갖는 제2 컨볼루션 신경망 계층을 사용하여 컨볼루션 신경망 계층에 대한 입력 텐서를 처리함
902
제2 컨볼루션 신경망 계층이 컨볼루션 신경망 계층의 스트라이드를 갖는다면 생성되지 않았을 엘리먼트들을 0으로 설정함
904
수정된 컨볼브된 텐서에 대한 최대 풀링을 수행함
906
900

도면10

(a)



(b)



(c)