(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년06월29일
(11) 등록번호 10-2414583
(24) 등록일자 2022년06월24일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2006.01) *G06N 99/00* (2019.01)
(52) CPC특허분류
G06N 3/084 (2013.01)
G06N 20/00 (2021.08)
(21) 출원번호 10-2017-0036715
(22) 출원일자 2017년03월23일
심사청구일자 2020년03월19일
(65) 공개번호 10-2018-0107869
(43) 공개일자 2018년10월04일
(56) 선행기술조사문헌
US20160062947 A1
US20160283841 A1

(73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
김경훈
경기도 수원시 영통구 동탄원천로881번길 35, 503동 1503호(매탄동, 주공그린빌)
박영환
경기도 용인시 수지구 수지로 166, 107동 306호(풍덕천동, 정자뜰마을 태영 데시앙1차 아파트)
(뒷면에 계속)
(74) 대리인
정홍식, 김태헌

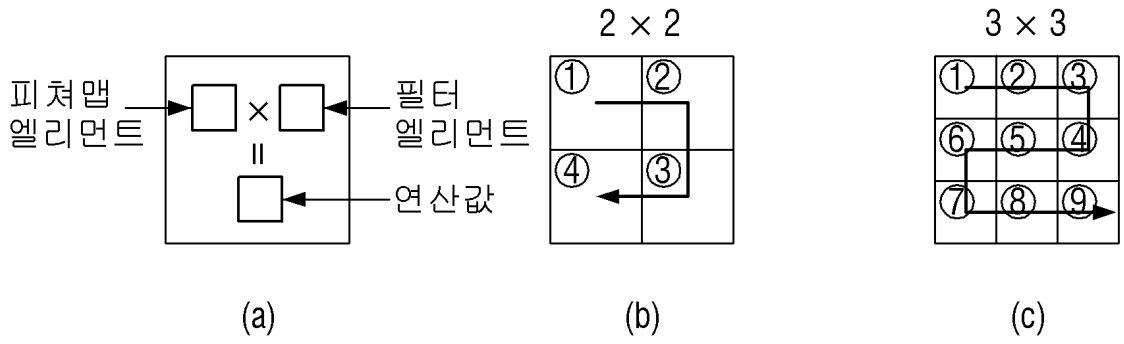
전체 청구항 수 : 총 19 항 심사관 : 양대경

(54) 발명의 명칭 **머신 러닝을 수행하는 전자 장치 및 머신 러닝 수행 방법**

(57) 요 약

머신 러닝(machine learning)을 수행하는 전자 장치가 개시된다. 전자 장치는, 기설정된 패턴으로 배열된 복수의 연산 소자를 포함하며, 서로 인접하는 연산 소자 간에 데이터를 공유하여 연산을 수행하는 연산 모듈 및 입력 데이터에 필터를 적용하여 콘볼루션 연산을 수행하도록 연산 모듈을 제어하는 프로세서를 포함하며, 프로세서는, 2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서대로 복수의 연산 소자로 입력하여 입력 데이터에 복수의 엘리먼트를 순차적으로 적용하여 콘볼루션 연산을 수행하도록 연산 모듈을 제어한다.

대 표 도 - 도6



(72) 발명자

권기석

서울특별시 송파구 중대로 24, 108동 702호 (문정동, 올림픽훼밀리타운)

김석진

서울특별시 마포구 토정로31길 23, 102동 1501호 (용강동, 마포 리버웰)

임채석

경기도 용인시 기흥구 서천동로 60, 401동 303호 (서천동, 서천마을4단지)

조한수

경기도 화성시 영통로50번길 14, 202동 204호(반월동, 반달마을두산위브아파트)

한상복

경기도 수원시 영통구 도청로 65, 5408동 2203호(이의동, 자연앤 힐스테이트)

이승원

경기도 화성시 영통로27번길 53, 212동 901호 (반월동, 신영통 현대타운)

윤강진

서울특별시 송파구 문정로 55, 104동 801호 (문정동, 대우아파트)

명 세 서

청구범위

청구항 1

머신 러닝(machine learning)을 수행하는 전자 장치에 있어서,

기설정된 패턴으로 배열된 복수의 연산 소자를 포함하며, 서로 인접하는 연산 소자 간에 데이터를 공유하여 연산을 수행하는 연산 모듈; 및

입력 데이터에 필터를 적용하여 콘볼루션 연산을 수행하도록 상기 연산 모듈을 제어하는 프로세서;를 포함하며,

상기 프로세서는,

2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서대로 상기 복수의 연산 소자로 입력하여 상기 입력 데이터에 상기 복수의 엘리먼트를 순차적으로 적용하여 콘볼루션 연산을 수행하도록 상기 연산 모듈을 제어하는, 전자 장치.

청구항 2

제1항에 있어서,

상기 복수의 엘리먼트 각각은,

상기 기설정된 순서로 배열된 1차원 데이터이며,

상기 프로세서는,

2차원 또는 3차원 입력 데이터에 상기 복수의 엘리먼트 각각을 적용하여 상기 콘볼루션 연산을 수행하도록 상기 연산 모듈을 제어하는, 전자 장치.

청구항 3

제1항에 있어서,

상기 프로세서는,

상기 2차원 필터를 상기 복수의 엘리먼트로 분할하고, 상기 복수의 엘리먼트 중 제로 값을 가지는 엘리먼트를 제외한 나머지 엘리먼트들을 각각 상기 기설정된 순서 대로 상기 복수의 연산 소자로 입력하는, 전자 장치.

청구항 4

제1항에 있어서,

상기 프로세서는,

상기 입력 데이터의 서로 다른 데이터 값과 상기 복수의 엘리먼트 각각을 곱한 값에 대한 어큐뮬레이션(accumulation)을 인접한 연산 소자로 전달하여 상기 콘볼루션 연산을 수행하도록 상기 연산 모듈을 제어하는, 전자 장치.

청구항 5

제1항에 있어서,

상기 프로세서는,

상기 입력 데이터의 제1 행에 속한 복수의 제1 데이터 값 각각에 상기 복수의 엘리먼트 중 제1 엘리먼트를 곱하는 연산을 수행하고, 상기 제1 엘리먼트를 상기 입력 데이터의 제2 행에 속한 복수의 제2 데이터 값 각각에 곱하는 연산을 수행하며,

상기 복수의 제1 데이터 값 각각에 상기 복수의 엘리먼트 중 제2 엘리먼트를 곱하는 연산을 수행하고, 상기 제2

엘리먼트를 상기 복수의 제2 데이터 값 각각에 곱하는 연산을 수행하는, 전자 장치.

청구항 6

제5항에 있어서,

상기 프로세서는,

상기 제1 행에서 상기 제1 엘리먼트 연산이 완료되고 상기 제2 엘리먼트에 대한 연산이 시작되면, 상기 제1 엘리먼트에 대한 복수의 연산 값을 기설정된 방향으로 시프트하여 연산 값들에 대한 어큐뮬레이션을 수행하며,

상기 기설정된 방향은,

상기 2차원 필터에서 상기 제1 엘리먼트를 기준으로 상기 제2 엘리먼트가 배치된 방향인, 전자 장치.

청구항 7

제1항에 있어서,

상기 프로세서는,

상기 입력 데이터에 대한 각 행에서 연산된 연산 값을 기설정된 방향으로 시프트하여 연산 값들에 대한 어큐뮬레이션을 수행하며,

상기 기설정된 방향은,

상기 2차원 필터에서 특정 엘리먼트를 기준으로 일측 방향을 진행하고, 상기 진행 방향의 마지막에 위치한 엘리먼트의 다음 행 또는 다음 열에서 상기 마지막에 위치한 엘리먼트와 인접한 엘리먼트로 진행하고, 상기 인접한 엘리먼트에서 상기 일측 방향과 반대 방향으로 진행하는 순서가 반복되는 방향인, 전자 장치.

청구항 8

제1항에 있어서,

상기 복수의 연산 소자는,

메쉬 토폴로지(mesh topology) 네트워크에 트리 토폴로지(tree topology) 네트워크가 결합된 구조의 네트워크를 형성하고,

상기 프로세서는,

상기 결합된 구조의 네트워크를 이용하여, CNN(Convolutional Neural Network) 알고리즘 및 RNN(Recurrent Neural Network) 알고리즘에 따른 연산을 수행하도록 상기 복수의 연산 소자를 제어하는, 전자 장치.

청구항 9

제8항에 있어서,

상기 프로세서는,

상기 CNN 알고리즘의 콘볼루션 레이어(convolution layer) 및 풀링 레이어(pooling layer)에서는 상기 메쉬 토폴로지 네트워크에 따른 연산을 수행하고, 상기 CNN 알고리즘의 완전 연결 레이어(fully connected layer) 및 상기 RNN 알고리즘의 각 레이어에서는 상기 트리 토폴로지 네트워크에 따른 연산을 수행하도록 상기 복수의 연산 소자를 제어하는, 전자 장치.

청구항 10

기설정된 패턴으로 배열된 복수의 연산 소자를 포함하며 서로 인접하는 연산 소자 간에 데이터를 공유하여 연산을 수행하는 연산 모듈을 이용한 머신 러닝(machine learning) 수행 방법에 있어서,

입력 데이터를 수신하는 단계; 및

상기 입력 데이터에 필터를 적용하여 콘볼루션 연산을 수행하는 단계;를 포함하며,

상기 콘볼루션 연산을 수행하는 단계는,

2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서대로 상기 복수의 연산 소자로 입력하여 상기 입력 데이터에 상기 복수의 엘리먼트를 순차적으로 적용하여 콘볼루션 연산을 수행하는, 방법.

청구항 11

제10항에 있어서,

상기 복수의 엘리먼트 각각은,

상기 기설정된 순서로 배열된 1차원 데이터이며,

상기 콘볼루션 연산을 수행하는 단계는,

2차원 또는 3차원 입력 데이터에 상기 복수의 엘리먼트 각각을 적용하여 상기 콘볼루션 연산을 수행하는, 방법.

청구항 12

제10항에 있어서,

상기 콘볼루션 연산을 수행하는 단계는,

상기 2차원 필터를 상기 복수의 엘리먼트로 분할하고, 상기 복수의 엘리먼트 중 제로 값을 가지는 엘리먼트를 제외한 나머지 엘리먼트들을 각각 상기 기설정된 순서대로 상기 복수의 연산 소자로 입력하는, 방법.

청구항 13

제10항에 있어서,

상기 콘볼루션 연산을 수행하는 단계는,

상기 입력 데이터의 서로 다른 데이터 값과 상기 복수의 엘리먼트 각각을 곱한 값에 대한 어큐뮬레이션을 인접한 연산 소자로 전달하여 상기 콘볼루션 연산을 수행하는, 방법.

청구항 14

제10항에 있어서,

상기 콘볼루션 연산을 수행하는 단계는,

상기 입력 데이터의 제1 행에 속한 복수의 제1 데이터 값 각각에 상기 복수의 엘리먼트 중 제1 엘리먼트를 곱하는 연산을 수행하고, 상기 제1 엘리먼트를 상기 입력 데이터의 제2 행에 속한 복수의 제2 데이터 값 각각에 곱하는 연산을 수행하는 단계; 및

상기 복수의 제1 데이터 값 각각에 상기 복수의 엘리먼트 중 제2 엘리먼트를 곱하는 연산을 수행하고, 상기 제2 엘리먼트를 상기 복수의 제2 데이터 값 각각에 곱하는 연산을 수행하는 단계;를 포함하는, 방법.

청구항 15

제14항에 있어서,

상기 콘볼루션 연산을 수행하는 단계는,

상기 제1 행에서 상기 제1 엘리먼트에 대한 연산이 완료되고, 상기 제2 엘리먼트에 대한 연산이 시작되면, 상기 제1 엘리먼트에 대한 복수의 연산 값을 기설정된 방향으로 시프트하여 연산 값들에 대한 어큐뮬레이션을 수행하는 단계;를 더 포함하고,

상기 기설정된 방향은,

상기 2차원 필터에서 상기 제1 엘리먼트를 기준으로 상기 제2 엘리먼트가 배치된 방향인, 방법.

청구항 16

제10항에 있어서,

상기 콘볼루션 연산을 수행하는 단계는,

상기 입력 데이터에 대해 각 행에서 연산된 연산 값을 기설정된 방향으로 시프트하여 연산 값들에 대한 어큐뮬레이션을 수행하고,

상기 기설정된 방향은,

상기 2차원 필터에서 특정 엘리먼트를 기준으로 일측 방향을 진행하고, 상기 진행 방향의 마지막에 위치한 엘리먼트의 다음 행 또는 다음 열에서 상기 마지막에 위치한 엘리먼트와 인접한 엘리먼트로 진행하고, 상기 인접한 엘리먼트에서 상기 일측 방향과 반대 방향으로 진행하는 순서가 반복되는 방향인, 방법.

청구항 17

제10항에 있어서,

상기 복수의 연산 소자는,

메쉬 토폴로지 네트워크에 트리 토폴로지 네트워크가 결합된 구조의 네트워크를 형성하고,

상기 콘볼루션 연산을 수행하는 단계는,

상기 결합된 구조의 네트워크를 이용하여, CNN 알고리즘에 따른 상기 콘볼루션 연산 또는 RNN 알고리즘에 따른 연산을 수행하는, 방법.

청구항 18

제17항에 있어서,

상기 콘볼루션 연산을 수행하는 단계는,

상기 CNN 알고리즘의 콘볼루션 레이어 및 풀링 레이어에서, 상기 메쉬 토폴로지 네트워크에 따른 연산을 수행하고, 상기 CNN 알고리즘의 완전 연결 레이어 및 상기 RNN 알고리즘의 각 레이어에서는 상기 트리 토폴로지 네트워크에 따른 연산을 수행하는, 방법.

청구항 19

기설정된 패턴으로 배열된 복수의 연산 소자를 포함하며 서로 인접하는 연산 소자 간에 데이터를 공유하여 연산을 수행하는 연산 모듈을 이용하여 머신 러닝(machine learning)을 수행하기 위한 프로그램이 저장된 기록 매체에 있어서,

입력 데이터를 수신하는 단계; 및

상기 입력 데이터에 필터를 적용하여 콘볼루션 연산을 수행하는 단계;를 포함하며,

상기 콘볼루션 연산을 수행하는 단계는,

2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서 대로 상기 복수의 연산 소자로 입력하여 상기 입력 데이터에 상기 복수의 엘리먼트를 순차적으로 적용하여 콘볼루션 연산을 수행하는, 방법을 실행시키는 프로그램이 저장된 기록매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 머신 러닝을 수행하는 전자 장치 및 머신 러닝 수행 방법에 관한 것으로, 보다 상세하게는 머신 러닝에 이용되는 뉴럴 네트워크에 따른 연산을 수행하기 위한 방법에 관한 것이다.

배 경 기 술

[0002] 인공지능의 한 분야인 머신 러닝(machine learning)은 대규모의 빅데이터를 수집 및 분석하여 미래를 예측하고 스스로의 성능을 향상시키는 시스템과 이를 위한 알고리즘을 연구하고 구축하는 기술을 의미한다.

[0003] 최근, 하드웨어 기술의 발전에 힘입어 빅데이터의 수집과 저장이 가능해지고, 이를 분석하는 컴퓨터 능력과 기

술이 정교해지고 빨라짐에 따라, 인간처럼 사물을 인식하고 정보를 이해할 수 있는 알고리즘인 머신러닝에 대한 연구가 활발히 진행되고 있다. 특히, 머신 러닝 기술분야에서도 뉴럴 네트워크(neural network)를 이용한 자율 학습 방식의 딥 러닝에 대한 연구가 활발하다.

[0004] 뉴럴 네트워크는 인간의 뇌의 기능을 적극적으로 모방하려는 의도에 기초하여, 복수의 입력에 가중치를 곱한 총합에 대하여 활성 함수가 특정 경계값과 비교하여 최종 출력을 결정하는 알고리즘으로, 일반적으로 복수의 레이어로 구성되어 있다. 이미지 인식에 많이 이용되는 컨볼루셔널 뉴럴 네트워크(Convolutional Neural Network, 이하 CNN), 음성 인식에 많이 이용되는 리커런트 뉴럴 네트워크(Recurrent Neural Network, 이하 RNN) 등이 대표적이다.

[0005] 그러나, 기존의 CNN에서는 기본적으로 2차원의 콘볼루션 연산이 수행되므로, 콘볼루션 연산에 이용되는 필터의 스파시티(sparsity)에 의한 불필요한 연산이 발생됨에 따라 연산속도 및 메모리의 사용 측면에서 효율적이지 못한 문제가 있다. 또한, CNN 및 RNN에 따라서, 각 데이터를 전달하는 연산소자(Processing Element, 이하, PE)의 경로(path)가 상이하기 때문에 별개의 PE 구조를 필요로 한다는 문제가 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 상술한 문제점을 해결하기 위한 것으로, 본 발명의 목적은 CNN 및 RNN에 의한 연산을 수행함에 있어, 연산 효율을 높이기 위한 콘볼루션 연산 방법 및 CNN 및 RNN에 의한 연산을 동시에 지원하는 통합된 PE 구조를 제공하는 전자 장치 및 그 머신 러닝 수행 방법을 제공함에 있다.

과제의 해결 수단

[0007] 상술한 문제점을 해결하기 위한 본 발명의 일 실시 예에 따른, 머신 러닝(machine learning)을 수행하는 전자 장치는 기설정된 패턴으로 배열된 복수의 연산 소자를 포함하며, 서로 인접하는 연산 소자 간에 데이터를 공유하여 연산을 수행하는 연산 모듈 및, 입력 데이터에 필터를 적용하여 콘볼루션 연산을 수행하도록 상기 연산 모듈을 제어하는 프로세서를 포함하며, 상기 프로세서는, 2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서대로 상기 복수의 연산 소자로 입력하여 상기 입력 데이터에 상기 복수의 엘리먼트를 순차적으로 적용하여 콘볼루션 연산을 수행하도록 상기 연산 모듈을 제어한다.

[0008] 여기서, 상기 복수의 엘리먼트 각각은, 상기 기설정된 순서로 배열된 1차원 데이터이며, 상기 프로세서는, 2차원 또는 3차원 입력 데이터에 상기 복수의 엘리먼트 각각을 적용하여 상기 콘볼루션 연산을 수행하도록 상기 연산 모듈을 제어할 수 있다.

[0009] 또한, 상기 프로세서는, 상기 2차원 필터를 상기 복수의 엘리먼트로 분할하고, 상기 복수의 엘리먼트 중 제로 값을 가지는 엘리먼트를 제외한 나머지 엘리먼트들을 각각 상기 기설정된 순서 대로 상기 복수의 연산 소자로 입력할 수 있다.

[0010] 또한, 상기 프로세서는, 상기 입력 데이터의 서로 다른 데이터 값과 상기 복수의 엘리먼트 각각을 곱한 값에 대한 어큐뮬레이션(accumulation)을 인접한 연산 소자로 전달하여 상기 콘볼루션 연산을 수행하도록 상기 연산 모듈을 제어할 수 있다.

[0011] 또한, 상기 프로세서는, 상기 입력 데이터의 제1 행에 속한 복수의 제1 데이터 값 각각에 상기 복수의 엘리먼트 중 제1 엘리먼트를 곱하는 연산을 수행하고, 상기 제1 엘리먼트를 상기 입력 데이터의 제2 행에 속한 복수의 제2 데이터 값 각각에 곱하는 연산을 수행하며, 상기 복수의 제1 데이터 값 각각에 상기 복수의 엘리먼트 중 제2 엘리먼트를 곱하는 연산을 수행하고, 상기 제2 엘리먼트를 상기 복수의 제2 데이터 값 각각에 곱하는 연산을 수행할 수 있다.

[0012] 또한, 상기 프로세서는, 상기 제1 행에서 상기 제1 엘리먼트 연산이 완료되고 상기 제2 엘리먼트에 대한 연산이 시작되면, 상기 제1 엘리먼트에 대한 복수의 연산 값을 기설정된 방향으로 시프트하여 연산 값들에 대한 어큐뮬레이션을 수행하며, 상기 기설정된 방향은, 상기 2차원 필터에서 상기 제1 엘리먼트를 기준으로 상기 제2 엘리먼트가 배치된 방향일 수 있다.

[0013] 또한, 상기 프로세서는, 상기 입력 데이터에 대한 각 행에서 연산된 연산 값을 기설정된 방향으로 시프트하여 연산 값들에 대한 어큐뮬레이션을 수행하며, 상기 기설정된 방향은, 상기 2차원 필터에서 특정 엘리먼트를 기준

으로 일측 방향을 진행하고, 상기 진행 방향의 마지막에 위치한 엘리먼트의 다음 행 또는 다음 열에서 상기 해당 엘리먼트와 인접한 엘리먼트로 진행하고, 상기 인접한 엘리먼트에서 상기 일측 방향과 반대 방향으로 진행하는 순서가 반복되는 방향일 수 있다.

[0014] 또한, 상기 복수의 연산 소자는, 메쉬 토폴로지(mesh topology) 네트워크에 트리 토폴로지(tree topology) 네트워크가 결합된 구조의 네트워크를 형성하고, 상기 프로세서는, 상기 결합된 구조의 네트워크를 이용하여, CNN(Convolutional Neural Network) 알고리즘 및 RNN(Recurrent Neural Network) 알고리즘에 따른 연산을 수행하도록 상기 복수의 연산 소자를 제어할 수 있다.

[0015] 또한, 상기 프로세서는, 상기 CNN 알고리즘의 콘볼루션 레이어(convolution layer) 및 풀링 레이어(pooling layer)에서는 상기 메쉬 토폴로지 네트워크에 따른 연산을 수행하고, 상기 CNN 알고리즘의 완전 연결 레이어(fully connected layer) 및 상기 RNN 알고리즘의 각 레이어에서는 상기 트리 토폴로지 네트워크에 따른 연산을 수행하도록 상기 복수의 연산 소자를 제어할 수 있다.

[0016] 한편, 본 발명의 일 실시 예에 따른, 기설정된 패턴으로 배열된 복수의 연산 소자를 포함하며 서로 인접하는 연산 소자 간에 데이터를 공유하여 연산을 수행하는 연산 모듈을 이용하여 머신 러닝(machine learning)을 수행하는 방법은 입력 데이터를 수신하는 단계 및, 상기 입력 데이터에 필터를 적용하여 콘볼루션 연산을 수행하는 단계를 포함하며, 상기 콘볼루션 연산을 수행하는 단계는, 2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서대로 상기 복수의 연산 소자로 입력하여 상기 입력 데이터에 상기 복수의 엘리먼트를 순차적으로 적용하여 콘볼루션 연산을 수행한다.

[0017] 여기서, 상기 복수의 엘리먼트 각각은, 상기 기설정된 순서로 배열된 1차원 데이터이며, 상기 콘볼루션 연산을 수행하는 단계는, 2차원 또는 3차원 입력 데이터에 상기 복수의 엘리먼트 각각을 적용하여 상기 콘볼루션 연산을 수행할 수 있다.

[0018] 또한, 상기 콘볼루션 연산을 수행하는 단계는, 상기 2차원 필터를 상기 복수의 엘리먼트로 분할하고, 상기 복수의 엘리먼트 중 제로 값을 가지는 엘리먼트를 제외한 나머지 엘리먼트들을 각각 상기 기설정된 순서대로 상기 복수의 연산 소자로 입력할 수 있다.

[0019] 또한, 상기 콘볼루션 연산을 수행하는 단계는, 상기 입력 데이터의 서로 다른 데이터 값과 상기 복수의 엘리먼트 각각을 곱한 값에 대한 어큐뮬레이션을 인접한 연산 소자로 전달하여 상기 콘볼루션 연산을 수행할 수 있다.

[0020] 또한, 상기 콘볼루션 연산을 수행하는 단계는, 상기 입력 데이터의 제1 행에 속한 복수의 제1 데이터 값 각각에 상기 복수의 엘리먼트 중 제1 엘리먼트를 곱하는 연산을 수행하고, 상기 제1 엘리먼트를 상기 입력 데이터의 제2 행에 속한 복수의 제2 데이터 값 각각에 곱하는 연산을 수행하는 단계 및, 상기 복수의 제1 데이터 값 각각에 상기 복수의 엘리먼트 중 제2 엘리먼트를 곱하는 연산을 수행하고, 상기 제2 엘리먼트를 상기 복수의 제2 데이터 값 각각에 곱하는 연산을 수행하는 단계를 포함할 수 있다.

[0021] 또한, 상기 콘볼루션 연산을 수행하는 단계는, 상기 제1 행에서 상기 제1 엘리먼트에 대한 연산이 완료되고, 상기 제2 엘리먼트에 대한 연산이 시작되면, 상기 제1 엘리먼트에 대한 복수의 연산 값을 기설정된 방향으로 시프트하여 연산 값들에 대한 어큐뮬레이션을 수행하는 단계를 더 포함하고, 상기 기설정된 방향은, 상기 2차원 필터에서 상기 제1 엘리먼트를 기준으로 상기 제2 엘리먼트가 배치된 방향일 수 있다.

[0022] 또한, 상기 콘볼루션 연산을 수행하는 단계는, 상기 입력 데이터에 대해 각 행에서 연산된 연산 값을 기설정된 방향으로 시프트하여 연산 값들에 대한 어큐뮬레이션을 수행하고, 상기 기설정된 방향은, 상기 2차원 필터에서 특정 엘리먼트를 기준으로 일측 방향을 진행하고, 상기 진행 방향의 마지막에 위치한 엘리먼트의 다음 행 또는 다음 열에서 상기 해당 엘리먼트와 인접한 엘리먼트로 진행하고, 상기 인접한 엘리먼트에서 상기 일측 방향과 반대 방향으로 진행하는 순서가 반복되는 방향일 수 있다.

[0023] 또한, 상기 복수의 연산 소자는, 메쉬 토폴로지 네트워크에 트리 토폴로지 네트워크가 결합된 구조의 네트워크를 형성하고, 상기 콘볼루션 연산을 수행하는 단계는, 상기 결합된 구조의 네트워크를 이용하여, CNN 알고리즘에 따른 상기 콘볼루션 연산 또는 RNN 알고리즘에 따른 연산을 수행할 수 있다.

[0024] 또한, 상기 콘볼루션 연산을 수행하는 단계는, 상기 CNN 알고리즘의 콘볼루션 레이어 및 풀링 레이어에서, 상기 메쉬 토폴로지 네트워크에 따른 연산을 수행하고, 상기 CNN 알고리즘의 완전 연결 레이어 및 상기 RNN 알고리즘의 각 레이어에서는 상기 트리 토폴로지 네트워크에 따른 연산을 수행할 수 있다.

[0025] 한편, 본 발명의 일 실시 예에 따른, 기설정된 패턴으로 배열된 복수의 연산 소자를 포함하며 서로 인접하는 연

산 소자 간에 데이터를 공유하여 연산을 수행하는 연산 모듈을 이용하여 머신 러닝(machine learning)을 수행하기 위한 프로그램이 저장된 기록 매체는, 입력 데이터를 수신하는 단계 및, 상기 입력 데이터에 필터를 적용하여 콘볼루션 연산을 수행하는 단계를 포함하며, 상기 콘볼루션 연산을 수행하는 단계는, 2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서 대로 상기 복수의 연산 소자로 입력하여 상기 입력 데이터에 상기 복수의 엘리먼트를 순차적으로 적용하여 콘볼루션 연산을 수행한다.

발명의 효과

[0026] 상술한 본 발명의 다양한 실시 예에 따르면, CNN 및 RNN에 따른 연산을 수행함에 있어, 연산 속도 및 효율을 높일 수 있다.

도면의 간단한 설명

[0027] 도 1은 본 발명의 일 실시 예에 따른, 전자 장치의 구성을 간략히 도시한 블록도,

도 2는 본 발명의 일 실시 예에 따른, PE의 구성을 설명하기 위한 도면,

도 3은 CNN에 따른 콘볼루션 연산을 수행하기 위한 PE의 네트워크 구조를 설명하기 위한 도면,

도 4a 내지 4c는 기존의 CNN에 따른 콘볼루션 연산 방법을 설명하기 위한 도면,

도 5는 기존의 CNN에 따른 콘볼루션 연산 방법의 문제점을 설명하기 위한 도면,

도 6은 본 발명의 일 실시 예에 따른, PE에 대한 필터의 입력 순서를 설명하기 위한 도면,

도 7a 내지 7j는 본 발명의 일 실시 예에 따른, CNN에 따른 콘볼루션 연산 방법을 구체적으로 설명하기 위한 도면,

도 8a 내지 8n은 본 발명의 다른 실시 예에 따른, CNN에 따른 콘볼루션 연산 방법을 구체적으로 설명하기 위한 도면,

도 9는 RNN에 따른 연산을 수행하기 위한 PE의 네트워크 구조를 설명하기 위한 도면,

도 10은 본 발명의 일 실시 예에 따른, CNN 및 RNN에 따른 연산을 모두 지원할 수 있는 PE의 통합 네트워크 구조를 설명하기 위한 도면,

도 11은 본 발명의 일 실시 예에 따른, PE의 통합 네트워크 구조를 이용하여 완전 연결(fully connected)에 의한 연산 및 1X1 커널의 콘볼루션 연산을 수행하는 방법을 간략히 설명하기 위한 도면,

도 12는 본 발명의 일 실시 예에 따른, PE의 통합 네트워크 구조를 이용하여 RNN에 의한 연산을 수행하는 방법을 간략히 설명하기 위한 도면,

도 13a 내지 13g는 본 발명의 일 실시 예에 따른, PE의 통합 네트워크 구조를 이용하여 완전 연결에 의한 연산을 수행하는 방법을 구체적으로 설명하기 위한 도면,

도 14a 및 14b는 일정한 뎁스(depth)를 갖는 피쳐맵에 대하여 콘볼루션을 수행하는 방법을 설명하기 위한 도면,

도 15는 본 발명의 일 실시 예에 따른, 머신 러닝을 수행하는 방법을 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0028] 본 발명에 대하여 구체적으로 설명하기에 앞서, 본 명세서 및 도면의 기재 방법에 대하여 설명한다.

[0029] 먼저, 본 명세서 및 청구범위에서 사용되는 용어는 본 발명의 다양한 실시 예들에서의 기능을 고려하여 일반적인 용어들을 선택하였다. 하지만, 이러한 용어들은 당 분야에 종사하는 기술자의 의도나 법률적 또는 기술적 해석 및 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한, 일부 용어는 출원인이 임의로 선정한 용어일 수 있다. 이러한 용어에 대해서는 본 명세서에서 정의된 의미로 해석될 수 있으며, 구체적인 용어 정의가 없으면 본 명세서의 전반적인 내용 및 당해 기술 분야의 통상적인 기술 상식을 토대로 해석될 수도 있다.

[0030] 또한, 본 명세서에 첨부된 각 도면에 기재된 동일한 참조 번호 또는 부호는 실질적으로 동일한 기능을 수행하는 부품 또는 구성요소를 나타낸다. 설명 및 이해의 편의를 위해서 서로 다른 실시 예들에서도 동일한 참조번호 또는 부호를 사용하여 설명하도록 한다. 즉, 복수의 도면에서 동일한 참조 번호를 가지는 구성 요소를 모두 도시

하고 있다고 하더라도, 복수의 도면들이 하나의 실시 예를 의미하는 것은 아니다.

[0031] 또한, 본 명세서 및 청구범위에서는 구성요소들 간의 구별을 위하여 “제1”, “제2” 등과 같이 서수를 포함하는 용어가 사용될 수 있다. 이러한 서수는 동일 또는 유사한 구성 요소들을 서로 구별하기 위하여 사용하는 것이며, 이러한 서수 사용으로 인하여 용어의 의미가 한정 해석되어서는 안될 것이다. 일 예로, 이러한 서수와 결합된 구성 요소는 그 숫자에 의해 사용 순서나 배치 순서 등이 제한 해석되어서는 안된다. 필요에 따라서는, 각 서수들은 서로 교체되어 사용될 수도 있다.

[0032] 본 명세서에서 단수의 표현은 문맥상 명백하게 다름을 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, “포함하다” 또는 “구성하다” 등의 용어는 명세서 상에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0033] 본 발명의 실시 예에서 “모듈”, “유닛”, “부(Part)” 등과 같은 용어는 적어도 하나의 기능이나 동작을 수행하는 구성 요소를 지칭하기 위한 용어이며, 이러한 구성 요소는 하드웨어 또는 소프트웨어로 구현되거나 하드웨어 및 소프트웨어의 결합으로 구현될 수도 있다. 또한, 복수의 “모듈”, “유닛”, “부(part)” 등은 각각이 개별적인 특정한 하드웨어로 구현될 필요가 있는 경우를 제외하고는, 적어도 하나의 모듈이나 칩으로 일체화되어 적어도 하나의 프로세서(미도시)로 구현될 수 있다.

[0034] 또한, 본 발명의 실시 예에서, 어떤 부분이 다른 부분과 연결되어 있다고 할 때, 이는 직접적인 연결뿐 아니라, 다른 매체를 통한 간접적인 연결의 경우도 포함한다. 또한 어떤 부분이 어떤 구성 요소를 포함한다는 의미는, 특별히 반대되는 기재가 없는 한 다른 구성 요소를 제외하는 것이 아니라 다른 구성 요소를 더 포함할 수 있다는 것을 의미한다.

[0035] 이하, 첨부된 도면을 이용하여 본 발명에 대하여 구체적으로 설명한다.

[0036] 도 1은 본 발명의 일 실시 예에 따른, 전자 장치의 구성을 간략히 도시한 블록도이다.

[0037] 도 1을 참조하면, 머신 러닝(machine learning)을 수행하기 위한 전자 장치(100)는 연산모듈(110) 및 프로세서(120)를 포함한다.

[0038] 연산모듈(110)은 복수의 PE를 포함하는 구성이다. 복수의 PE는 기설정된 패턴의 배열 구조로 구성된 처리 소자로서, 동기적으로 서로 인접하는 PE 간에 데이터를 병렬처리하고, 동시에 같은 기능을 수행하는 구성이다. PE는 연산 작업 및 PE 간 데이터 교환 작업을 수행하며, 하나의 클럭에 동기되어 연산을 수행할 수 있다. 즉, 복수의 PE는 클럭마다 동일한 연산 작업을 수행할 수 있다. 복수의 PE는 동일한 경로로 이웃한 PE와 데이터를 공유하므로 PE 간의 연결 구조가 기하학적으로 간단한 대칭 구조를 이룰 수 있다.

[0039] 예를 들어, PE는 메쉬 토폴로지 네트워크(mesh topology network), 트리 토폴로지 네트워크(tree topology network) 등 다양한 형태의 네트워크의 구조로 배열될 수 있다. 메쉬 토폴로지 네트워크(mesh topology network) 및 트리 토폴로지 네트워크(tree topology network)의 구조에 대하여는 도 3 및 도 9와 관련하여 후술하도록 한다.

[0040] 한편, PE는 도 2에 도시된 바와 같이 기본적으로, 한 쌍의 곱셈기(multiplier)(21) 및 산술 논리 연산 장치(Arithmetic Logic Unit, 이하 ALU)(22)를 포함하며, ALU는 적어도 하나 이상의 가산기(adder)를 포함할 수 있다. PE는 곱셈기(21) 및 ALU(22)를 이용하여 사칙 연산을 수행할 수 있다.

[0041] 프로세서(120)는 전자 장치(100)의 전반적인 동작을 제어하는 구성이다. 특히, 프로세서(120)는 연산 모듈(110)을 이용하여 이미지 분류, 이미지 인식, 음성 인식, 자연 언어 프로세싱 등에 있어 이용되는 뉴럴 네트워크에 의한 연산을 수행할 수 있다.

[0042] 구체적으로, 프로세서(120)는 연산 모듈(110)로 입력되는 입력 데이터에 필터(filter)를 적용하여 뉴럴 네트워크에 기반한 콘볼루션 연산을 수행하도록 연산 모듈(110)을 제어할 수 있다. 여기서, 필터는 가중치를 갖는 마스크로서 행렬로 정의된다. 필터는 윈도우(windows) 또는 커널(kernel)이라고도 한다.

[0043] 예를 들어, 주로 영상을 처리하기 위한 CNN에 있어서, 프로세서(120)는 가중치를 갖는 필터를 입력되는 영상에 씌우고, 영상과 필터의 가중치를 각각 곱한 값에 대한 합(콘볼루션 연산)을 출력영상의 픽셀값으로 결정하여 피처맵(feature map)을 추출할 수 있다. 입력 영상은 강인한 특징을 추출하기 위해 다중 필터를 통해 복수 개로

추출될 수 있으며, 필터의 개수에 따라 복수 개의 피쳐 맵이 추출될 수 있다. 이와 같은 콘볼루션 영상은 다중 레이어에 의해 반복될 수 있다.

[0044] 이와 같이 프로세서(120)는 각기 다른 특징을 추출할 수 있는 다중 필터를 조합하여 CNN에 적용함으로써, 입력되는 원본 데이터가 어떤 형태의 특징을 가지고 있는지 판단할 수 있다.

[0045] 한편, 본 발명의 일 실시 예에 따른 프로세서(120)는 2차원으로 구현되는 필터를 구성하는 복수의 엘리먼트(element)를 각각 기설정된 순서대로 연산 모듈(110)로 입력하고, 입력된 복수의 엘리먼트를 입력 데이터에 순차적으로 적용하여 콘볼루션 연산을 수행하도록 연산 모듈(110)을 제어할 수 있다. 여기서, 복수의 엘리먼트 각각은 기설정된 순서로 배열된 1차원 데이터이며, 프로세서(120)는 2차원 또는 3차원의 입력 데이터에 복수의 엘리먼트 각각을 적용하여 콘볼루션 연산을 수행한다. 이에 대한 구체적인 설명은 이하, 도면을 참조하여 후술하도록 한다.

[0046] 도 3은 CNN에 따른 콘볼루션 연산을 수행하기 위한 PE의 네트워크 구조를 설명하기 위한 도면이다.

[0047] 도 3을 참조하면, 연산 모듈(110)에 포함된 복수의 PE(20-1 내지 20-n)는 메쉬 토폴로지 네트워크 구조의 시스톨릭 배열(systolic array)을 형성한다. 복수의 PE(20-1 내지 20-n)는 이러한 메쉬 토폴로지 네트워크 구조를 이용하여, 서로 인접하는 PE 를 연결하는 라인을 통해 데이터를 공유하여 연산을 수행할 수 있다. 메쉬 토폴로지 네트워크 구조는 도 3에 도시된 바와 같이 각 PE가 그물망처럼 인접한 PE들끼리 연결되어 데이터를 교환할 수 있는 구조로 형성된다.

[0048] 복수의 PE(20-1 내지 20-n)은 각각 곱셈기(20-1 내지 20-n) 및 ALU(21-1 내지 21-n)을 포함하며, 각각의 곱셈기(20-1 내지 20-n) 및 ALU(21-1 내지 21-n)에 의해 연산된 값이 인접한 PE로 전달되어 연산이 수행될 수 있다.

[0049] 도 4a 내지 4c는 기존의 CNN에 따른 콘볼루션 연산 방법을 설명하기 위한 도면이다.

[0050] 도 4a는 2차원의 입력 데이터에 2차원의 필터(41)를 적용하여 CNN에 따른 콘볼루션 연산을 수행하는 기존의 방법을 도시한 것으로, 입력 데이터 및 필터(41)는 적어도 하나 이상의 특정 값을 갖는 엘리먼트를 포함하는 행렬 데이터로 이루어질 수 있다. 콘볼루션은 입력 데이터의 각 엘리먼트(예를 들어, 픽셀)와 그 주변 엘리먼트들에 가중치를 곱한 값의 합으로 연산되며, 필터(41)는 이러한 합을 구하는 연산에 사용되는 가중치들의 행렬을 의미한다. 필터(41)는 커널 또는 윈도우로 명명되기도 한다. 필터(41)는 그 기능에 따라 다양한 형태로 구현될 수 있다.

[0051] 이하에서는, 설명의 편의를 위해 입력 데이터가 이미지 형식의 데이터인 경우를 가정하도록 한다.

[0052] 도 4a에 도시된 바와 같이, 먼저, 입력 이미지로부터 필터(41)와 곱할 부분인 슬라이스(10)가 추출되고, 슬라이스(10)에 필터(41)가 적용된다. 여기서, 입력 이미지로부터 추출된 슬라이스(10)를 피쳐맵이라고도 한다. 이때, 필터(41)는 슬라이스(10) 상에서 한 칸씩 이동되면서 적용된다. 예를 들어, 10X10의 슬라이스(10)가 존재할 때, 도 4b와 같이 3X3의 필터가 좌측 상단에서부터 왼쪽으로 한 칸씩 적용되고, 그 다음 줄에서 다시 왼쪽으로 한 칸씩 적용된다.

[0053] 필터(41)가 이동되면서, 필터(41)의 각 엘리먼트(41-1 내지 41-9)가 슬라이스(10)에 있어서 대응되는 엘리먼트와 곱해지며, 곱한 값이 더해진 값이 필터(41)가 이동함에 따라 각각 출력되게 된다. 출력된 결과는 다시 행렬을 이루며, 출력된 결과인 피쳐맵에 대하여는 다른 필터를 통해 콘볼루션 연산이 반복적으로 수행되게 된다. 필요에 따라서는 샘플링 과정인 풀링(pooling)이 수행되며, 이러한 연산 과정에서 출력값이 손실되는 것을 방지하기 위해서 패딩(padding)이 적용될 수 있다.

[0054] 도 4c는 슬라이스(10)에 서로 다른 제1 필터(41) 및 제2 필터(42)을 각각 적용하여 제1 피쳐맵(11a) 및 제2 피쳐맵(11b)을 출력하는 과정을 나타낸 것이다. 피쳐맵(11a, 11b)에서 가장 외곽에 존재하는 엘리먼트들은 패딩에 의해 생성된 앨리먼트이며, 슬라이스(10)에서 가장 처음 콘볼루션 연산이 수행된 값은 제1 및 제2 피쳐맵(11a)에서 패딩 영역을 제외하고 각각의 첫번째 엘리먼트(11a-1, 11b-1)가 된다.

[0055] CNN의 콘볼루셔널 레이어에서 최종적으로 특징 값들이 추출되면, 완전 연결 레이어에서, 추출된 특징 값들을 뉴럴 네트워크에 입력하여 분류를 수행하게 된다.

[0056] 한편, 도 5는 기존의 CNN에 따른 콘볼루션 연산 방법의 문제점을 설명하기 위한 도면이다.

[0057] 먼저, 도 4a 내지 4b에 도시된 콘볼루션 연산 방법은 연산 효율성이 떨어지는 문제가 있다. 도 5에 도시된 바와 같이 필터(51)는 일반적으로 복수의 엘리먼트(51-1 내지 51-9) 중에서 제로 값을 갖는 엘리먼트(51-2, 51-5,

51-8, 51-9)를 포함하는 경우가 많다.

[0058] 3X3의 2차원 필터(51)를 그대로 피처맵(10)에 적용하는 경우에는, 콘볼루션 과정에서 제로 값을 가지는 엘리먼트에 대한 불필요한 연산을 수행할 수 밖에 없으므로, 결과 값 도출에 영향을 미치지 않는 불필요한 연산 및 그에 따른 메모리 사용량의 증가에 따른 효율 저하의 문제가 발생되게 된다.

[0059] 도 6은 상술한 2차원 필터(51)의 적용으로 인한 비효율 문제를 극복하기 위한 본 발명의 일 실시 예에 따른 콘볼루션 방법을 간략히 도시한 것이다. 구체적으로, 도 6은 PE에 대한 필터의 입력 순서를 나타내고 있다.

[0060] 도 6의 (a)에 도시된 바와 같이, 단일의 PE에서는 곱셈기에 의해, 피처맵 엘리먼트와 필터의 엘리먼트가 각각 곱해진 값이 연산값으로 출력되게 된다. 이때, 연산 모듈(110)에 입력되는 필터는 2차원의 행렬 형식의 데이터가 그대로 적용되는 것이 아니라, 필터를 구성하는 1차원 데이터인 엘리먼트 별로 분할되어, 순차적으로 연산 모듈(110)에 입력된다. 즉, 2차원의 피처맵에 1차원의 필터가 적용되는 것과 같다.

[0061] 구체적으로, 프로세서(120)는 2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서대로 복수의 연산 소자로 입력하고, 피처맵에 복수의 엘리먼트를 순차적으로 적용하여 CNN에 의한 콘볼루션 연산을 수행하도록 연산 모듈(110)을 제어할 수 있다. 이때, 프로세서(120)는 2차원 필터를 복수의 엘리먼트로 분할하면서, 복수의 엘리먼트 중 제로 값을 가지는 엘리먼트를 제외하고, 제로 값을 가지지 않는 나머지 엘리먼트들을 각각 기설정된 순서대로 연산 모듈(110)로 입력할 수 있다.

[0062] 도 6의 (b) 및 (c)는 본 발명의 일 실시 예에 따라, 2차원 필터의 엘리먼트들이 연산 모듈(110)로 입력되는 순서를 나타낸 것이다. 기본적인 입력 순서는 2차원 필터에서 특정 엘리먼트를 기준으로 일측 방향으로 진행하고, 진행 방향의 마지막에 위치한 엘리먼트의 다음 행 또는 다음 열에서 해당 엘리먼트와 인접한 엘리먼트로 진행하며, 인접한 엘리먼트에서 일측 방향과 반대 방향으로 진행하는 순서로 반복된다. 도 6의 (b)에서는 엘리먼트①부터 엘리먼트④까지 번호 순서대로 연산 모듈(110)에 입력되게 된다. 도 6의 (c)에서는 엘리먼트①부터 엘리먼트⑨까지 번호 순서대로 연산 모듈(110)에 입력되게 된다.

[0063] 도 6에 따른 구체적인 콘볼루션 연산 방법은 이하, 도 7a 내지 7j를 참조하여 설명하도록 한다.

[0064] 도 7a 내지 7j는 본 발명의 일 실시 예에 따라, 2X2의 2차원 필터를 이용하여 CNN에 따른 콘볼루션 연산을 수행하는 방법을 구체적으로 설명하기 위한 도면이다.

[0065] 도 7a의 (a)에 도시된 바와 같이, 제로값을 포함하지 않는 2X2의 필터(70)가 피처맵에 적용된다고 가정할 경우, 프로세서(120)는 필터(70)를 4개의 엘리먼트로 분할하여 도 7a의 (b)에 도시된 연산 모듈(110)에 순차적으로 입력할 수 있다. 여기서, 연산 모듈(110)은 메쉬 토폴로지 네트워크 구조의 PE에 피처맵의 각 엘리먼트가 1:1로 맵핑(mapping)되어 입력될 수 있도록 구성된다.

[0066] 도 7b를 참조하면, 첫 번째 클럭에서, 엘리먼트①이 연산 모듈(110)의 1행에 포함된 PE에 각각 입력된다. 연산 모듈(110)에 입력되는 필터(70)의 각 엘리먼트는 연산 모듈(110)의 일 방향으로 순차적으로 진행하며, 프로세서(120)는 입력되는 피처맵의 1행에 속한 복수의 제1 데이터 값 각각에 필터(70)를 구성하는 엘리먼트 중 엘리먼트①을 곱하는 연산을 수행하고, 엘리먼트①을 피처맵의 2행에 속한 복수의 제2 데이터 값 각각에 곱하는 연산을 수행할 수 있다. 즉, 엘리먼트①이 입력된 각각의 PE는 엘리먼트①과 각각 PE에 대응되는 피처맵의 엘리먼트와의 곱에 대한 결과 값(A1 내지 A8)을 도출할 수 있다.

[0067] 한편, 프로세서(120)는 입력 데이터의 서로 다른 데이터 값, 즉, 피처맵의 서로 다른 엘리먼트 값과 필터(70)의 각 엘리먼트 각각을 곱한 값(A1 내지 A8)에 대한 어큐뮬레이션(accumulation)을 각각 인접한 PE로 전달하여 콘볼루션 연산을 수행하도록 연산 모듈(110)을 제어할 수 있다. 즉, 1차원으로 분할된 필터(70)의 각 엘리먼트를 PE 간에 전달하면서 필터의 각 엘리먼트와 피처맵의 데이터 값에 대한 어큐뮬레이션을 수행하여 기존의 2차원 필터(70)에 의한 콘볼루션 연산과 동일한 연산을 수행하도록 할 수 있다.

[0068] 구체적으로는, 프로세서(120)에 의해, 입력 데이터의 1행에 속한 복수의 제1 데이터 값 각각에 필터(70)의 복수의 엘리먼트 중 제1 엘리먼트를 곱하는 연산을 수행하고, 제1 엘리먼트를 입력 데이터의 2행에 속한 복수의 제2 데이터 값 각각에 곱하는 연산이 수행된다. 또한, 복수의 제1 데이터 값 각각에 필터(70)의 복수의 엘리먼트 중 제2 엘리먼트를 곱하는 연산이 수행되며, 제2 엘리먼트를 복수의 제2 데이터 값 각각에 곱하는 연산이 수행된다. 이와 같은 연산은 필터(70)의 모든 엘리먼트에 대해 반복된다.

[0069] 이후, 프로세서(120)는 1행에서 제1 엘리먼트 연산이 완료되고, 제2 엘리먼트에 대한 연산이 시작되면, 제1 엘리먼트에 대한 복수의 연산 값을 기설정된 방향으로 시프트(shift)하여 연산 값들에 대한 어큐뮬레이션을 수행

한다. 여기서, 기설정된 방향은 2차원 필터에서 제1 엘리먼트를 기준으로 제2 엘리먼트가 배치된 방향과 동일하다.

[0070] 도 7c에 도시된 바와 같이, 다음 클럭에서는, 엘리먼트②가 연산 모듈(110)의 1행에 포함된 PE에 각각 입력되고, 엘리먼트①은 2행에 포함된 PE로 각각 입력된다. 이때, 엘리먼트①과 2행의 PE에 매핑된 피처맵의 엘리먼트가 각각 곱해진 결과 값(B1 내지 B8)이 도출될 수 있다. 또한, 엘리먼트②와 1행의 PE에 매핑된 피처맵의 엘리먼트가 곱해진 결과 값(C1 내지 C8) 역시 도출된다.

[0071] 이때, 이전 클럭에서 도출된 A1 내지 A8 역시, 우측에 인접한 PE로 한 칸씩 이동되어, C1 내지 C8과 각각 합산되는 어큐뮬레이션 동작이 수행된다. 어큐뮬레이션 동작이 수행되는 방향은 필터(70)에서 번호가 매겨진 순서에 대응되는 방향(도 7a의 (a)에서 화살표의 방향)과 동일하다. 도 7c에 도시된 바와 같이, C1 내지 C8은 A0 내지 A7과 각각 합산된다. 도 7d는 C1 내지 C8이 A0 내지 A7과 각각 합산된 제1 어큐뮬레이션 값(D0 내지 D7)을 나타낸 것이다.

[0072] 이후, 다음 클럭에서는, 도 7e에 도시된 바와 같이, 엘리먼트③이 연산 모듈(110)의 1행에 포함된 PE에 각각 입력되고, 엘리먼트①은 3행에 포함된 PE로, 엘리먼트②는 2행에 포함된 PE로 각각 입력된다. 이때, 엘리먼트①과 3행의 PE에 매핑된 피처맵의 엘리먼트가 각각 곱해진 결과 값(F1 내지 F8)이 도출될 수 있다. 또한, 엘리먼트②와 2행의 PE에 매핑된 피처맵의 엘리먼트가 곱해진 결과 값(E1 내지 E8) 역시 도출된다. 엘리먼트③ 역시 1행의 PE에 매핑된 피처맵의 엘리먼트와 곱해지게 되는데, 설명의 편의를 위해, 그 결과값은 도면에서 생략하였다.

[0073] 한편, 이전 클럭에서 도출된 제1 어큐뮬레이션 값(D0 내지 D7)은 하단에 인접한 PE로 한 칸씩 이동되어 2행의 PE에 포함된 메모리에 각각 임시 저장될 수 있다(이를 위해, 각각의 PE는 메모리를 포함할 수 있다). 이와 함께, 이전 클럭에서 도출된 B1 내지 B7은, 이전 클럭에서의 A1 내지 A8과 마찬가지로 우측에 인접한 PE로 한 칸씩 이동되어, E1 내지 E8과 각각 합산되는 어큐뮬레이션 동작이 수행된다. 즉, 각 행의 PE마다 필터(70)에서 번호가 매겨진 순서에 대응되는 방향(도 7a의 (a)에서 화살표가 이동하는 방향)에 따른 어큐뮬레이션이 별개로 수행될 수 있다. 도 7f는 E1 내지 E8이 B0 내지 B7과 각각 합산된 어큐뮬레이션 값(G0 내지 G7)을 나타낸 것이다.

[0074] 이후, 다음 클럭에서는, 도 7g에 도시된 바와 같이, 필터(70)의 마지막 엘리먼트인 엘리먼트④가 연산 모듈(110)의 1행에 포함된 PE에 각각 입력되고, 엘리먼트①은 4행에 포함된 PE로, 엘리먼트②는 3행에 포함된 PE로, 엘리먼트③는 2행에 포함된 PE로 각각 입력된다. 이때, 엘리먼트①와 4행의 PE에 매핑된 피처맵의 엘리먼트가 각각 곱해진 결과 값(J1 내지 J8)이 도출될 수 있다. 또한, 엘리먼트②와 3행의 PE에 매핑된 피처맵의 엘리먼트가 곱해진 결과 값(I1 내지 I8), 엘리먼트③와 2행의 PE에 매핑된 피처맵의 엘리먼트가 곱해진 결과 값(H1 내지 H8) 역시 도출된다. 엘리먼트④ 역시 1행의 PE에 매핑된 피처맵의 엘리먼트와 곱해지게 되는데, 설명의 편의를 위해, 그 결과값은 도면에서 생략하였다.

[0075] 한편, 이전 클럭에서 도출된 어큐뮬레이션 값(D1 내지 D7)은 H1 내지 H8과 각각 합산되는 어큐뮬레이션 동작이 수행되어, 도 7h에 도시된 바와 같이 제2 어큐뮬레이션 값(K0 내지 K7)이 도출될 수 있다. 3행의 L1 내지 L7은 우측에 인접한 PE로 한 칸씩 이동된 F0 내지 F7과 I1 내지 I8이 각각 합산된 값을 나타낸 것이다.

[0076] 이후, 다음 클럭에서는, 도 7i에 도시된 바와 같이, 엘리먼트①은 5행에 포함된 PE로, 엘리먼트②는 4행에 포함된 PE로, 엘리먼트③는 3행에 포함된 PE로, 엘리먼트④는 2행에 포함된 PE로 각각 입력된다. 이때, 엘리먼트④와 2행의 PE에 매핑된 피처맵의 엘리먼트가 각각 곱해진 결과 값(M1 내지 M8)이 도출될 수 있다. 이때, 이전 클럭에서 도출된 제2 어큐뮬레이션 값(K1 내지 K8)은 좌측에 인접한 PE로 한 칸씩 이동되어, M1 내지 M8과 각각 합산되는 어큐뮬레이션 동작이 수행될 수 있다. 이하, 엘리먼트①, 엘리먼트② 및 엘리먼트③은 도 7i에 도시된 바와 같이, 각각 연산 모듈(110)의 3행, 4행 및 5행에서 별도의 어큐뮬레이션 동작을 수행하게 된다.

[0077] 도 7j는 제2 어큐뮬레이션 값(K1 내지 K8)이 M1 내지 M8과 합산된 제3 어큐뮬레이션 값(Q1 내지 Q8)을 도시한 것이다. 제3 어큐뮬레이션 값(Q1 내지 Q8)은 필터(70)의 엘리먼트① 내지 ④가 피처맵의 엘리먼트 I 내지 IV와 각각 곱해진 값을 합산한 값과 같으며, 제3 어큐뮬레이션 값(Q1 내지 Q8)은 출력 단자를 통해 출력되어, 상술한 콘볼루션 연산에 의해 도출되는 새로운 피처맵의 1행 1열의 엘리먼트가 된다.

[0078] 도 8a 내지 8n은 본 발명의 다른 실시 예에 따른, CNN에 따른 콘볼루션 연산 방법을 구체적으로 설명하기 위한 도면이다.

[0079] 도 8a의 (a)에 도시된 바와 같이, 제로값을 포함하는 3X3의 필터(80)가 피처맵에 적용된다고 가정할 경우, 프로세서(120)는 필터(80)에서 제로값을 가지는 엘리먼트를 제거하고, 제로값을 가지지 않는 5개의 엘리먼트를 도

8b의 (b)에 도시된 연산 모듈(110)에 순차적으로 입력할 수 있다.

[0080] 도 8b를 참조하면, 첫 번째 클럭에서, 엘리먼트①가 연산 모듈(110)의 1행에 포함된 PE에 각각 입력된다. 연산 모듈(110)에 입력되는 필터(80)의 각 엘리먼트는 연산 모듈(110)의 일 방향으로 순차적으로 진행하며, 프로세서(120)는 입력되는 피처맵의 1행에 속한 복수의 제1 데이터 값 각각에 필터(80)를 구성하는 엘리먼트 중 엘리먼트①을 곱하는 연산을 수행한다. 즉, 엘리먼트①이 입력된 각각의 PE는 엘리먼트①과 각각 PE에 대응되는 피처맵의 엘리먼트와의 곱에 대한 결과 값(A1 내지 A8)을 도출할 수 있다.

[0081] 이후, 다음 클럭에서는 도 8c에 도시된 바와 같이, 엘리먼트②가 연산 모듈(110)의 1행에 포함된 PE에 각각 입력되고, 엘리먼트①은 2행에 포함된 PE로 각각 입력된다. 이때, 엘리먼트①과 2행의 PE에 매핑된 피처맵의 엘리먼트가 각각 곱해진 결과 값(B1 내지 B8)이 각각 도출될 수 있다. 또한, 엘리먼트②와 1행의 PE에 매핑된 피처맵의 엘리먼트가 곱해진 결과 값(C1 내지 C8) 역시 도출된다.

[0082] 이때, 이전 클럭에서 도출된 A1 내지 A8은 우측의 PE로 이동되어, C1 내지 C8과 합산되는 어큐뮬레이션 동작이 수행된다. 어큐뮬레이션 동작이 수행되는 방향은 필터(80)에서 번호가 매겨진 순서에 대응되는 방향(도 8a의 (a)에서 화살표의 방향)과 동일하다. 다만, 필터가 제로값을 포함하지 않는 도 7a 내지 7j의 실시 예와는 달리, 도 8a의 (a)에 도시된 필터(80)는 제로값을 포함하고 있으므로, 필터(80)에서 엘리먼트①과 엘리먼트②가 떨어진 만큼, A1 내지 A8도 우측으로 두 칸씩 이동되어 어큐뮬레이션이 수행된다. 즉, C1 내지 C8은 A(-1) 내지 A(6)와 각각 각각 합산된다. 도 8d는 C1 내지 C8이 A(-1) 내지 A(6)와 각각 합산된 제1 어큐뮬레이션 값(D(-1) 내지 D6)을 나타낸 것이다.

[0083] 이후, 다음 클럭에서는, 도 8e에 도시된 바와 같이, 엘리먼트③가 연산 모듈(110)의 1행에 포함된 PE에 각각 입력되고, 엘리먼트①은 3행에 포함된 PE로, 엘리먼트②는 2행에 포함된 PE로 각각 입력된다. 이때, 엘리먼트②와 2행의 PE에 매핑된 피처맵의 엘리먼트가 곱해진 결과 값(E1 내지 E8)이 도출되며, 이전 클럭에서 도출된 제1 어큐뮬레이션 값(D(-1) 내지 D6)은 하단에 인접한 PE로 한 칸씩 이동되어 2행의 PE에 포함된 메모리에 각각 임시 저장될 수 있다. 이와 함께, 이전 클럭에서 도출된 B1 내지 B7은, 이전 클럭에서의 A1 내지 A8과 마찬가지로 우측으로 두 칸씩 이동되어, E1 내지 E8과 각각 합산되는 어큐뮬레이션 동작이 수행된다. 도 8f는 E1 내지 E8이 B(-1) 내지 B6과 각각 합산된 어큐뮬레이션 값(G(-1) 내지 G6)을 나타낸 것이다.

[0084] 이후, 다음 클럭에서는, 도 8g에 도시된 바와 같이, 엘리먼트④가 연산 모듈(110)의 1행에 포함된 PE에 각각 입력되고, 엘리먼트①은 4행에 포함된 E로, 엘리먼트②는 3행에 포함된 PE로, 엘리먼트③은 2행에 포함된 PE로 각각 입력된다. 이때, 엘리먼트③과 2행의 PE에 매핑된 피처맵의 엘리먼트가 곱해진 결과 값(H1 내지 H8)이 도출되며, H1 내지 H8은 2행의 PE에 각각 임시저장된 D(-1) 내지 D6와 각각 합산되는 어큐뮬레이션 동작이 수행된다. 도 8h는 H1 내지 H8이 D(-1) 내지 D6와 각각 합산된 제2 어큐뮬레이션 값(J(-1) 내지 J6)을 나타낸 것이다.

[0085] 이후, 다음 클럭에서는, 도 8i에 도시된 바와 같이, 필터(70)의 마지막 엘리먼트인 엘리먼트⑤가 연산 모듈(110)의 1행에 포함된 PE에 각각 입력되고, 엘리먼트①은 5행에 포함된 PE로, 엘리먼트②는 4행에 포함된 PE로, 엘리먼트③은 3행에 포함된 PE로, 엘리먼트④는 2행에 포함된 PE로 각각 입력된다. 이때, 엘리먼트④와 2행의 PE에 매핑된 피처맵의 엘리먼트가 각각 곱해진 결과 값(L1 내지 L8)이 도출될 수 있다. 또한, 이전 클럭에서 도출된 제2 어큐뮬레이션 값(J(-1) 내지 J6)은 도 8i에 도시된 바와 같이 좌측의 PE로 두 칸씩 이동되어, L1 내지 L8과 각각 합산되는 어큐뮬레이션 동작이 수행될 수 있다. 도 8j는 제2 어큐뮬레이션 값(J(-1) 내지 J6)이 L1 내지 L8과 각각 합산된 제3 어큐뮬레이션 값(N1 내지 N8)을 나타낸 것이다.

[0086] 이후, 다음 클럭에서는, 도 8k 및 도 8l에 도시된 바와 같이, 엘리먼트①은 6행에 포함된 PE로, 엘리먼트②는 5행에 포함된 PE로, 엘리먼트③은 4행에 포함된 PE로, 엘리먼트④는 3행에 포함된 PE로, 엘리먼트⑤는 2행에 포함된 PE로 각각 입력된다. 이때, 이전 클럭에서 도출된 제3 어큐뮬레이션 값(N1 내지 N8)은 하단에 인접한 PE로 한 칸씩 이동되어 3행의 PE에 포함된 메모리에 각각 임시 저장될 수 있다.

[0087] 이후, 다음 클럭에서는, 도 8m에 도시된 바와 같이, 엘리먼트⑤는 3행에 포함된 PE로 입력되며, 엘리먼트⑤와 3행의 PE에 매핑된 피처맵의 엘리먼트가 각각 곱해진 결과 값(R1 내지 R8)이 도출될 수 있다. 이때, 3행의 PE에 임시 저장된 제3 어큐뮬레이션 값(N1 내지 N8)이 R1 내지 R8과 각각 합산되는 어큐뮬레이션 동작이 수행되어, 도 8m에 도시된 바와 같이, 제4 어큐뮬레이션 값(S1 내지 S8)이 도출될 수 있다. 제4 어큐뮬레이션 값(S1 내지 S8)은 필터(80)의 엘리먼트① 내지 ⑤가 피처맵의 엘리먼트 I 내지 V와 각각 곱해진 값을 합산한 값과 같으며, 제4 어큐뮬레이션 값(S1 내지 S8)은 출력 단자를 통해 출력되어, 상술한 콘볼루션 연산에 의해 도출되는 새로운

피처맵의 1행 1열의 엘리먼트가 된다.

[0088] 상술한 바와 같이, 1차원으로 분할된 필터를 피처맵에 순차적으로 적용하면, 필터에서 제로값에 대한 연산을 생략할 수 있으므로, 메모리의 부담이 낮아지고, 연산 효율이 높아지게 된다.

[0089] 도 9는 RNN에 따른 연산을 수행하기 위한 PE의 네트워크 구조를 설명하기 위한 도면이다.

[0090] 도 9는 트리 토폴로지 네트워크 구조의 시스톨릭 배열(systolic array)을 형성하는 연산 모듈(110)을 나타낸 것이다. 복수의 PE(20-1 내지 20-n)는 이러한 트리 토폴로지 네트워크 구조를 이용하여, 서로 인접하지 않은 PE와도 데이터를 공유하여 연산을 수행할 수 있다. 이러한 트리 토폴로지 네트워크 구조는 RNN에 따른 연산을 수행할 때 일반적으로 이용된다.

[0091] 이와 같이, CNN 및 RNN에 따른 연산은 데이터를 전달하는 경로가 상이하여, 메쉬 토폴로지 네트워크 구조 또는 트리 토폴로지 네트워크 구조의 단일한 PE의 배열 구조만으로는 CNN 및 RNN에 따른 연산을 모두 지원할 수 없는 문제가 있다. 특히, CNN의 완전 연결 레이어(fully connected layer)에서도 트리 토폴로지 네트워크 구조의 시스톨릭 배열을 형성하는 연산 모듈이 필요하게 되므로, 메쉬 토폴로지 네트워크 구조의 연산 모듈 및 트리 토폴로지 네트워크 구조의 연산 모듈이 모두 필요했었다.

[0092] 본 발명의 일 실시 예에 따른, 연산 모듈(110)의 구조는 CNN 및 RNN에 따른 연산을 모두 수행할 수 있도록 메쉬 토폴로지 네트워크 구조 및 트리 토폴로지 네트워크 구조가 결합된 구조의 통합 네트워크를 형성할 수 있다.

[0093] 도 10은 본 발명의 일 실시 예에 따른 연산 모듈(110)의 네트워크 구조를 도시한 것이다. 프로세서(120)는 이와 같이 결합된 구조의 네트워크를 이용하여, CNN 알고리즘 및 RNN 알고리즘에 따른 연산을 수행할 수 있도록 연산 모듈(110)을 제어할 수 있다. 특히, 프로세서(120)는 CNN 알고리즘의 콘볼루션 레이어 및 풀링 레이어(pooling layer)에서 메쉬 토폴로지 네트워크에 따른 연산을 수행하고, CNN 알고리즘의 완전 연결 레이어(fully connected layer) 및 RNN 알고리즘의 각 레이어에서는 트리 토폴로지 네트워크에 따른 연산을 수행하도록 연산 모듈(110)을 제어할 수 있다.

[0094] 이에 따라, 각 PE는 인접한 상하좌우의 PE 뿐만 아니라, 비인접한 PE 간에서도 어큐뮬레이션 값 등의 데이터를 전달할 수 있게 되므로, 데이터의 재활용 및 연산의 효율화 측면에서 장점이 있다. 또한, 하나의 연산 모듈(110)에서 CNN에 의한 연산, 완전 연결(fully connected)에 의한 분류 및 RNN에 의한 연산을 모두 수행할 수 있으므로, 머신 러닝에 필요한 PE 개수가 감축됨으로써 비용절감의 효과가 있다.

[0095] 도 11은 본 발명의 일 실시 예에 따른, PE의 통합 네트워크 구조를 이용하여 완전 연결(fully connected)에 의한 연산 및 1X1 커널의 콘볼루션 연산을 수행하는 방법을 간략히 설명하기 위한 도면이다.

[0096] 도 11의 (a)는 CNN 알고리즘의 완전 연결 레이어(fully connected layer)에 의한 분류 과정을 나타낸 것이다. 완전 연결 레이어는 CNN의 콘볼루션 레이어 및 풀링 레이어에 의해 압축·요약된 특징 데이터를 딥 뉴럴 네트워크(Deep Neural Network)에 의해 분류하여 최종적인 결과 값을 도출한다. 콘볼루션 레이어 및 풀링 레이어의 반복에 의해 산출되는 특징 데이터는 완전 연결 레이어의 딥 뉴럴 네트워크에 입력값(i_1 내지 i_{1000})으로 입력되며, 각각의 입력값은 가중치($W_{i,j}$)를 가지는 edge와 연결되어 있다. 각 입력값과 가중치의 곱을 합한 값은 활성함수(예를 들어, signoid function)에 입력되어 액티베이션 값(j_1 내지 j_{800})이 출력되고, 액티베이션 값들은 다음 레이어에서 또다른 가중치와 동일한 동작을 수행하여 최종 출력값이 출력되게 된다.

[0097] 도 11의 (a)에 도시된 바와 같이, 프로세서(120)는 완전 연결 레이어에 의한 분류 작업을 도 10에 도시된 통합 네트워크 구조의 연산 모듈(110)을 이용하여 수행하도록 연산 모듈(110)을 제어할 수 있다. 구체적으로, 각 입력값(i_1 내지 i_{1000})은 연산 모듈(110)의 상단부에 일렬로 배열되어 입력되며, 메쉬 토폴로지 네트워크에 의해 하단부로 한 칸씩 순차적으로 이동한다. 이때, 연산 모듈(110)의 각 PE에는 각각의 가중치가 미리 저장되어 있으며, 각 PE에서는 입력값과 가중치가 각각 곱해지는 연산이 수행되게 된다. 각 PE에서 곱해진 값은 트리 토폴로지 네트워크에 의해 어큐뮬레이션되어 액티베이션 값(j_1 내지 j_{800})이 출력되게 된다.

[0098] 또한, 도 10의 통합 네트워크 구조의 연산 모듈(110)은 하나의 엘리먼트만을 포함하는 1x1 사이즈의 필터에 의한 콘볼루션을 수행할 때에도 이용될 수 있다. 이때, 본 발명은 도 11의 (b)에 도시된 바와 같이, 1x1 필터를 3D 피처맵의 뎁스 방향(l)으로 우선 적용하여 콘볼루션을 수행하는 방법을 사용하여, 결과 값을 출력하기 위한 연산 속도를 높일 수 있다. 이때, 뎁스 방향(l)의 복수의 1x1 필터에 대한 1:1 콘볼루션을 수행할 때, 도 10의 통합 네트워크 구조의 연산 모듈(110)을 이용할 수 있다. 이때, 3D 피처맵의 뎁스만큼의 입력값이 연산 모듈

(110)의 상단부에 일렬로 배열되어 입력되고, 메쉬 토폴로지 네트워크에 의해 하단부로 한 칸씩 순차적으로 이동하며, 트리 토폴로지에 의한 네트워크에 의해 어큐뮬레이션된 액티베이션 값이 출력된다.

[0099] 도 12는 본 발명의 일 실시 예에 따른, PE의 통합 네트워크 구조를 이용하여 RNN에 의한 연산을 수행하는 방법을 간략히 설명하기 위한 도면이다.

[0100] RNN은 시간의 흐름에 따라 변하는 시계열 데이터(Time-series data)에 대한 딥 러닝을 수행하기 위한 알고리즘으로, 예를 들어, 음성인식, 음악 장르 분류, 문자열 생성, 동영상 분류 등의 작업을 처리하는데에 이용된다. RNN은 매 순간의 뉴럴 네트워크를 쌓아 올린 형태를 가지므로, 과거(t-1)의 뉴럴 네트워크에서의 입력 값부터 현재(t)의 뉴럴 네트워크에서의 입력값(121-1 ~ 121-n)까지 순차적으로 도 10의 연산 모듈(110)에 입력될 수 있다. 연산 모듈(110)의 각 PE에는 뉴럴 네트워크의 가중치(W_0)가 각각 미리 저장되어 있을 수 있다. 연산 모듈(110)은 RNN의 각각의 레이어마다 연산 값이 어큐뮬레이션된 값(122)을 출력하며, 각각의 레이어에서 연산된 과거의 값은 각각의 PE에 임시 저장되어 현재의 연산 과정에 전달되어 영향을 줄 수 있다. 즉, 각각의 PE에 과거 값이 임시로 저장되어 현재 레이어에서의 연산에 반영되는 것은 RNN에 있어서, 리커런트 가중치(recurrent weight)가 적용되는 것과 동일하다.

[0101] 도 13a 내지 13g는 본 발명의 일 실시 예에 따른, PE의 통합 네트워크 구조를 이용하여 완전 연결에 의한 연산을 수행하는 방법을 구체적으로 설명하기 위한 도면이다.

[0102] 도 13a의 (a)에 도시된 바와 같이, 시간 t-1에서의 제1 입력값(i_1 내지 i_8)이 연산 모듈(110)에 입력된다고 가정할 경우, 프로세서(120)는 제1 입력값과 연산 모듈(110)의 1행에 포함된 PE에 저장된 가중치와의 곱(A1 내지 A8)을 도출할 수 있다. 이후, 다음 클럭(t)에서 제2 입력값이 연산 모듈(110)에 입력될 때, A1 내지 A8은 PE에 임시 저장되어 리커런트 가중치로 활용된다.

[0103] 한편, 도 13b에 도시된 바와 같이, 제1 입력값은 다음 클럭(t)에서, 메쉬 토폴로지 네트워크 연결을 이용하여, 연산 모듈(110)의 2행에 포함된 PE에 입력되어, 제1 입력값과 2행에 포함된 PE에 저장된 가중치와의 곱(B1 내지 B8)이 도출된다. 또한, 1행에서의 A1 내지 A8은 트리 토폴로지 네트워크 연결을 이용하여, 제1 어큐뮬레이션이 수행된다. 즉, A1은 A2와, A3는 A4와, A5는 A6와, 그리고 A7은 A8과 각각 합산된다. 도 13c는 1행에서 트리 토폴로지 네트워크에 의해 각각의 A1 내지 A8이 인접한 값과 합산된 값(C1 내지 C4)를 나타낸 것이다. 한편, 시간 t에서 입력되는 제2 입력값에 의한 연산은 설명의 편의를 위해 도면에서 생략하였다.

[0104] 또한, 도 13d에 도시된 바와 같이, 다음 클럭(t+1)에서는 제1 입력 값이 연산 모듈(110)의 3행에 포함된 PE로 이동되면서, 제1 입력값과 3행에 포함된 PE에 저장된 가중치와의 곱(D1 내지 D8)을 도출하고, 1행에서의 C1 내지 C4는 트리 토폴로지 네트워크 연결을 이용하여 제2 어큐뮬레이션이 수행된다. 즉, C1은 C2와, C3는 4와 각각 합산된다. 도 13e는 1행에서 합산된 값(E1, E2)를 나타낸 것이다. 한편, 시간 t+1에서 입력되는 제3 입력값에 의한 연산 또한 설명의 편의를 위해 도면에서 생략하였다.

[0105] 이후, 도 13f에 도시된 바와 같이, 다음 클럭(t+2)에서는 제1 입력 값이 연산 모듈(110)의 4행에 포함된 PE로 이동하여 상술한 바와 같은 동일한 방식의 연산을 수행한다. 1행에서 제2 어큐뮬레이션에 의해 도출된 E1 및 E2에 대하여는, 트리 토폴로지 네트워크 연결을 이용하여 제3 어큐뮬레이션이 수행되며, 최종적으로 도 13g에 도시된 바와 같이 E1 및 E2가 합산된 값인 H1을 출력하게 된다. 이러한 방식으로 각각의 행에서 출력되는 값은 다시 다음 레이어의 입력값이 된다.

[0106] 도 14a 및 14b는 일정한 뎁스(depth)를 갖는 피처맵에 대하여 콘볼루션을 수행하는 방법을 설명하기 위한 도면이다.

[0107] 도 14a에 도시된 바와 같이, CNN에 있어, 피처맵(14)은 RGB(Red-Green-Blue)에 의한 인코딩에 의해 최초에는 3차원의 행렬로 표현될 수 있으며, 적용되는 커널의 개수에 따라 레이어마다 그 뎁스가 결정될 수 있다. 즉, 피처맵(14)은 3차원의 직육면체로 표현될 수 있다. 이때, 필터(15-1, 15-2)는 피처맵(14)의 뎁스 방향에 대한 콘볼루션을 추가적으로 수행하며, 콘볼루션에 의한 최종 값은 새로운 2차원 피처맵(16-1, 16-2)의 일 엘리먼트가 된다.

[0108] 필터가 복수 개인 경우에는, 도 14b에 도시된 바와 같이 g개의 필터(15-1 내지 15-g) 중 하나의 필터가 적용될 때마다 하나의 슬라이스(피처맵)(16-1 내지 16-g)가 도출되며, 최종적으로 도출되는 슬라이스의 뎁스는 g가 된다.

[0109] 도 15는 본 발명의 일 실시 예에 따른, 머신 러닝을 수행하는 방법을 설명하기 위한 흐름도이다.

[0110] 먼저, 입력 데이터를 수신한다(S1510). 이후, 2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서대로 복수의 연산소자로 입력한다(S1520). 여기서, 복수의 엘리먼트 각각은, 기설정된 순서로 배열된 1차원 데이터이다.

[0111] 이후, 수신된 입력 데이터에 복수의 엘리먼트를 순차적으로 적용하여 콘볼루션 연산을 수행한다(S1530). 이때, 2차원 또는 3차원의 입력 데이터에 복수의 엘리먼트 각각을 적용하여 콘볼루션 연산을 수행할 수 있다.

[0112] 또한, 2차원 필터를 복수의 엘리먼트로 분할하고, 복수의 엘리먼트 중 제로 값을 가지는 엘리먼트를 제외한 나머지 엘리먼트들을 각각 기설정된 순서대로 복수의 연산 소자로 입력할 수 있다. 이때, 입력 데이터의 서로 다른 데이터 값과 복수의 엘리먼트 각각을 곱한 값에 대한 어큐뮬레이션을 인접한 연산 소자로 전달하여 콘볼루션 연산을 수행할 수 있다.

[0113] 구체적으로, 입력 데이터의 제1행에 속한 복수의 제1 데이터 값 각각에 복수의 엘리먼트 중 제1 엘리먼트를 곱하는 연산을 수행하고, 제1 엘리먼트를 입력 데이터의 제2행에 속한 복수의 제2 데이터 값 각각에 곱하는 연산을 수행할 수 있다. 또한, 복수의 제1 데이터 값 각각에 복수의 엘리먼트 중 제2 엘리먼트를 곱하는 연산을 수행하고, 제2 엘리먼트를 복수의 제2 데이터 값 각각에 곱하는 연산을 수행할 수 있다.

[0114] 이후, 제1행에서 제1 엘리먼트에 대한 연산이 완료되고, 제2 엘리먼트에 대한 연산이 시작되면, 제1 엘리먼트에 대한 복수의 연산 값을 기설정된 방향으로 시프트하여 연산 값들에 대한 어큐뮬레이션을 수행할 수 있다. 여기서, 기설정된 방향은, 2차원 필터에서 제1 엘리먼트를 기준으로 제2 엘리먼트가 배치된 방향일 수 있다.

[0115] 더욱 구체적으로는, 기설정된 방향은 2차원 필터에서 특정 엘리먼트를 기준으로 일측 방향으로 진행하고, 진행 방향의 마지막에 위치한 엘리먼트의 다음 행 또는 다음 열에서 해당 엘리먼트와 인접한 엘리먼트로 진행하고, 인접한 엘리먼트에서 일측 방향과 반대 방향으로 진행하는 순서가 반복되는 방향일 수 있다.

[0116] 또한, 복수의 연산 소자는 메쉬 토폴로지 네트워크에 트리 토폴로지 네트워크가 결합된 구조의 네트워크를 형성할 수 있다. 이와 같은 결합된 구조의 네트워크를 이용하여, CNN 알고리즘에 따른 콘볼루션 연산 또는 RNN 알고리즘에 따른 연산을 수행할 수 있다. 예를 들어, CNN 알고리즘의 콘볼루션 레이어 및 풀링 레이어에서는 메쉬 토폴로지 네트워크에 따른 연산을 수행하고, CNN 알고리즘의 완전 연결 레이어 및 RNN 알고리즘의 각 레이어에서는 트리 토폴로지 네트워크에 따른 연산을 수행할 수 있다.

[0117] 이상과 같이, 본 발명의 다양한 실시 예에 따르면, 이미지, 음성 등의 데이터에 대한 머신 러닝을 수행함에 있어, 연산 속도 및 효율을 높일 수 있으며, PE 소자의 통합 네트워크 구조에 의해, 머신 러닝에 필요한 PE 소자의 개수를 줄임으로써 비용을 절감할 수 있다.

[0118] 상술한 다양한 실시 예에 따른 머신 러닝 수행 방법은 프로그램으로 구현되어 다양한 기록 매체에 저장될 수 있다. 즉, 각종 프로세서에 의해 처리되어 상술한 다양한 머신 러닝 수행 방법을 실행할 수 있는 컴퓨터 프로그램이 기록 매체에 저장된 상태로 사용될 수도 있다.

[0119] 일 예로, 입력 데이터를 수신하는 단계, 2차원 필터를 구성하는 복수의 엘리먼트를 각각 기설정된 순서대로 복수의 연산소자로 입력하는 단계 및 수신된 입력 데이터에 복수의 엘리먼트를 순차적으로 적용하여 콘볼루션 연산을 수행하는 프로그램이 저장된 비일시적 판독 가능 매체(non-transitory computer readable medium)가 제공될 수 있다.

[0120] 비일시적 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 구체적으로는, 상술한 다양한 어플리케이션 또는 프로그램들은 CD, DVD, 하드 디스크, 블루레이 디스크 USB, 메모리카드, ROM 등과 같은 비일시적 판독 가능 매체에 저장되어 제공될 수 있다.

[0121] 이상에서는 본 발명의 바람직한 실시 예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특정의 실시 예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변형 실시가 가능한 것은 물론이고, 이러한 변형실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해되어져서는 안될 것이다.

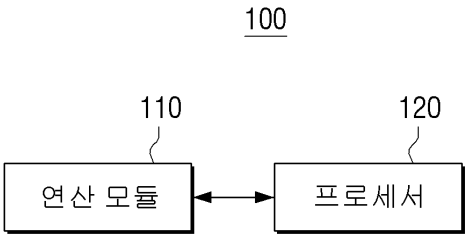
부호의 설명

[0122] 100: 전자 장치 110: 연산 모듈

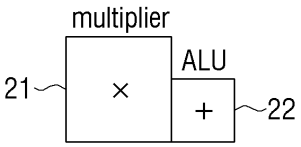
120: 프로세서

도면

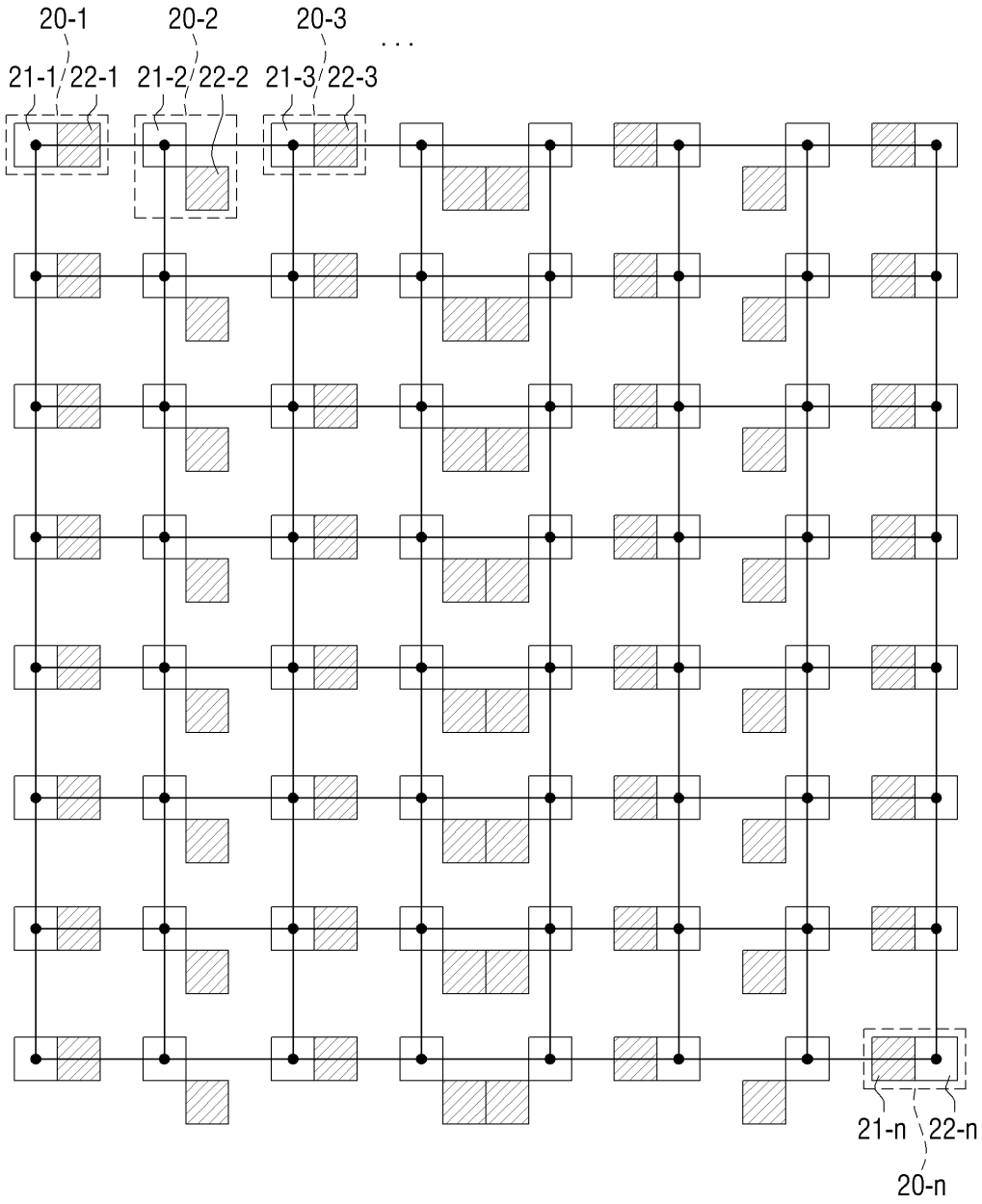
도면1



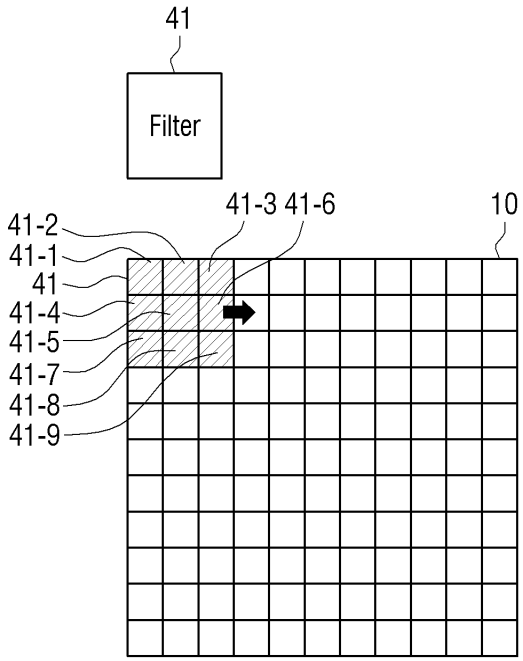
도면2



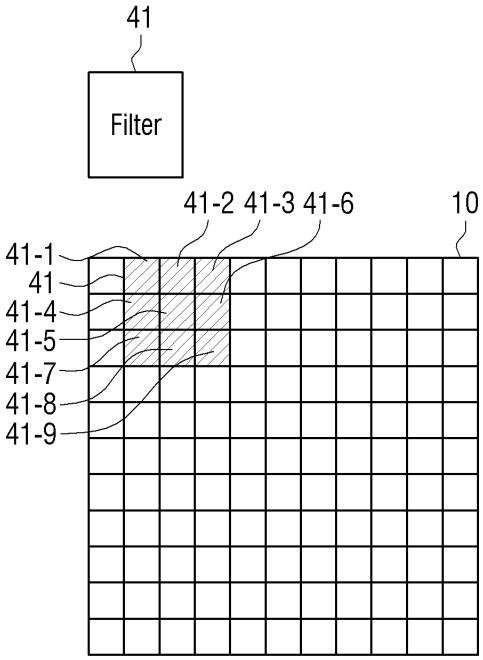
도면3



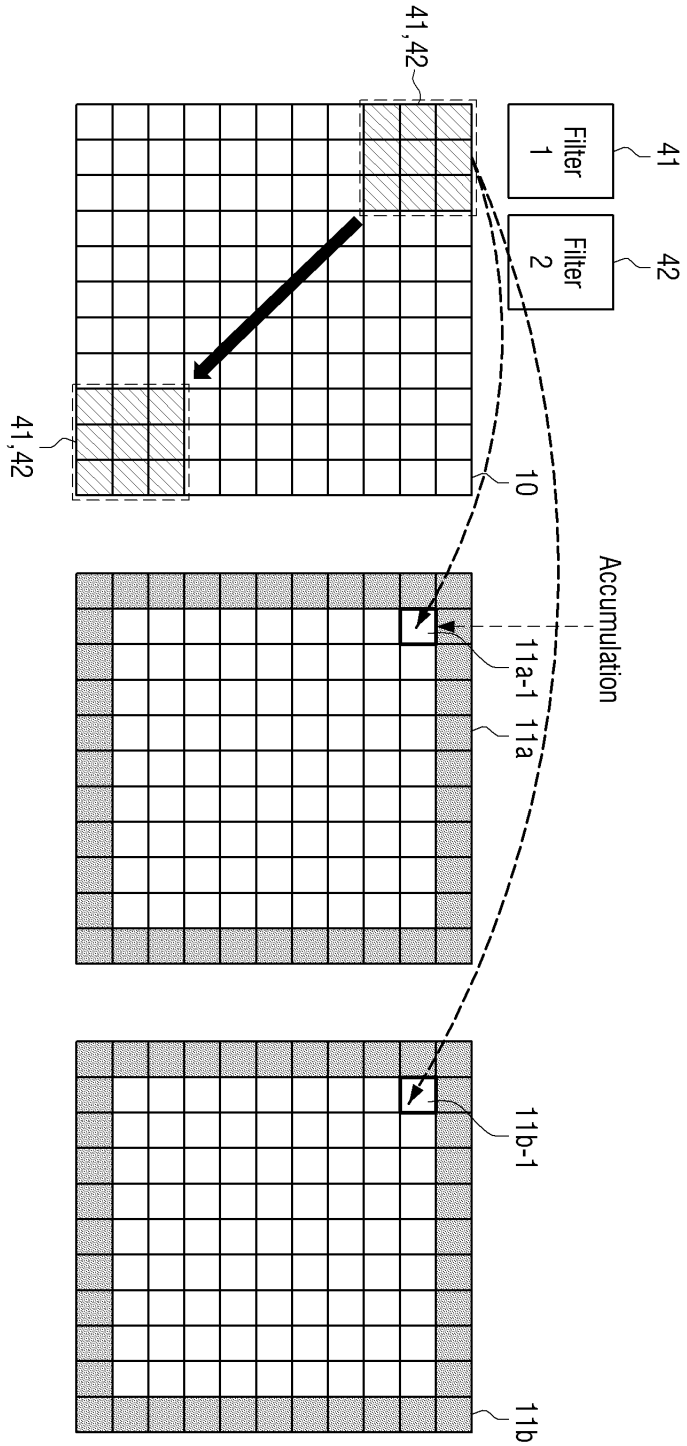
도면4a


41
Filter
41-3 41-6
10
41-2
41-1
41
41-4
41-5
41-7
41-8
41-9

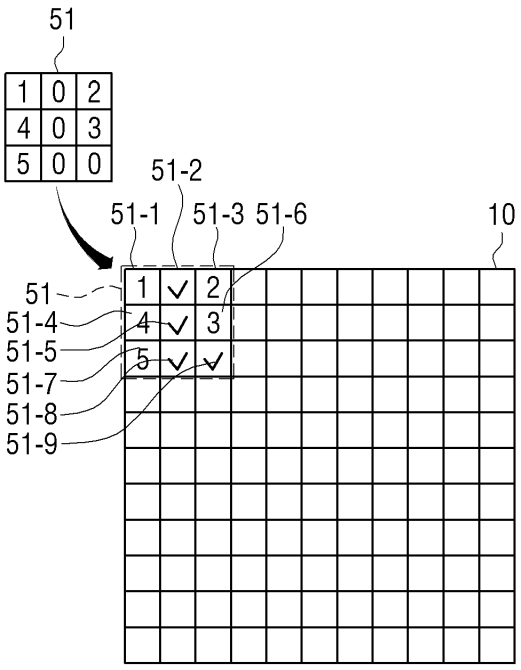
도면4b


41
Filter
41-2 41-3 41-6
10
41-1
41
41-4
41-5
41-7
41-8
41-9

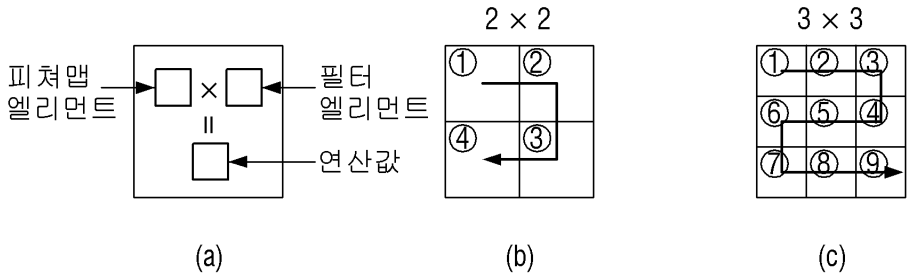
도면4c


41,42
Filter 1
41
Filter 2
42
41,42
10
Accumulation
11a-1
11a
11b-1
11b

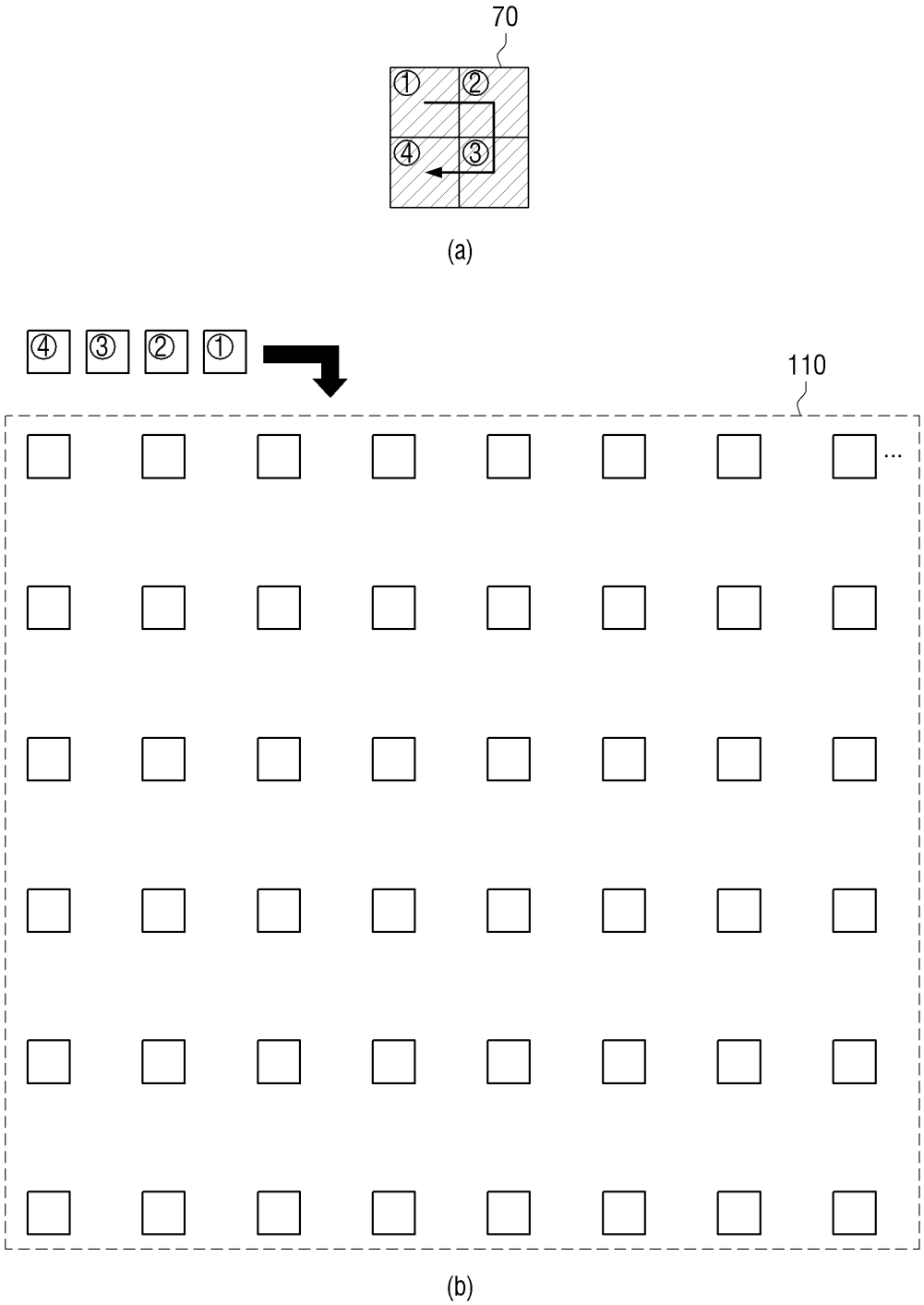
도면5


51
1 0 2
4 0 3
5 0 0
51-2
51-1
51-3
51-6
10
51
51-4
51-5
51-7
51-8
51-9
1 2
4 3
5

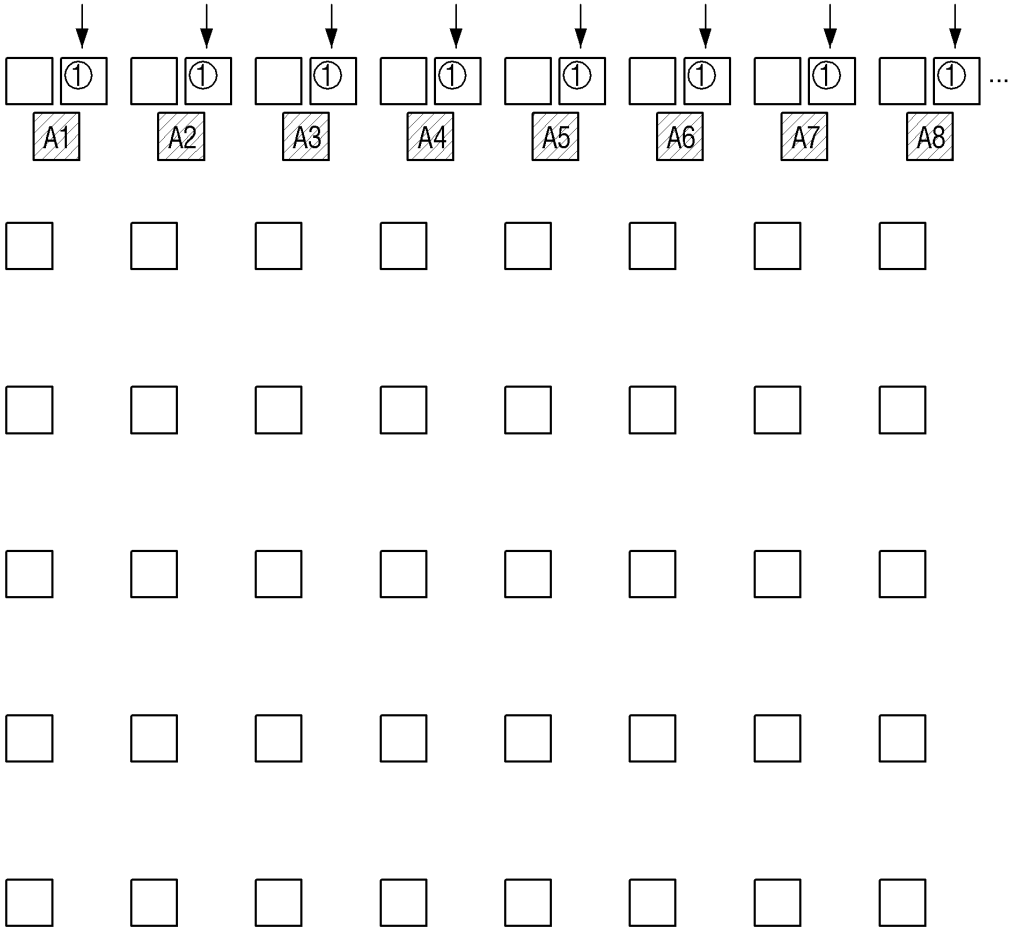
도면6


피쳐맵
엘리먼트
×
=
필터
엘리먼트
연산값
2 × 2
3 × 3
(a)
(b)
(c)

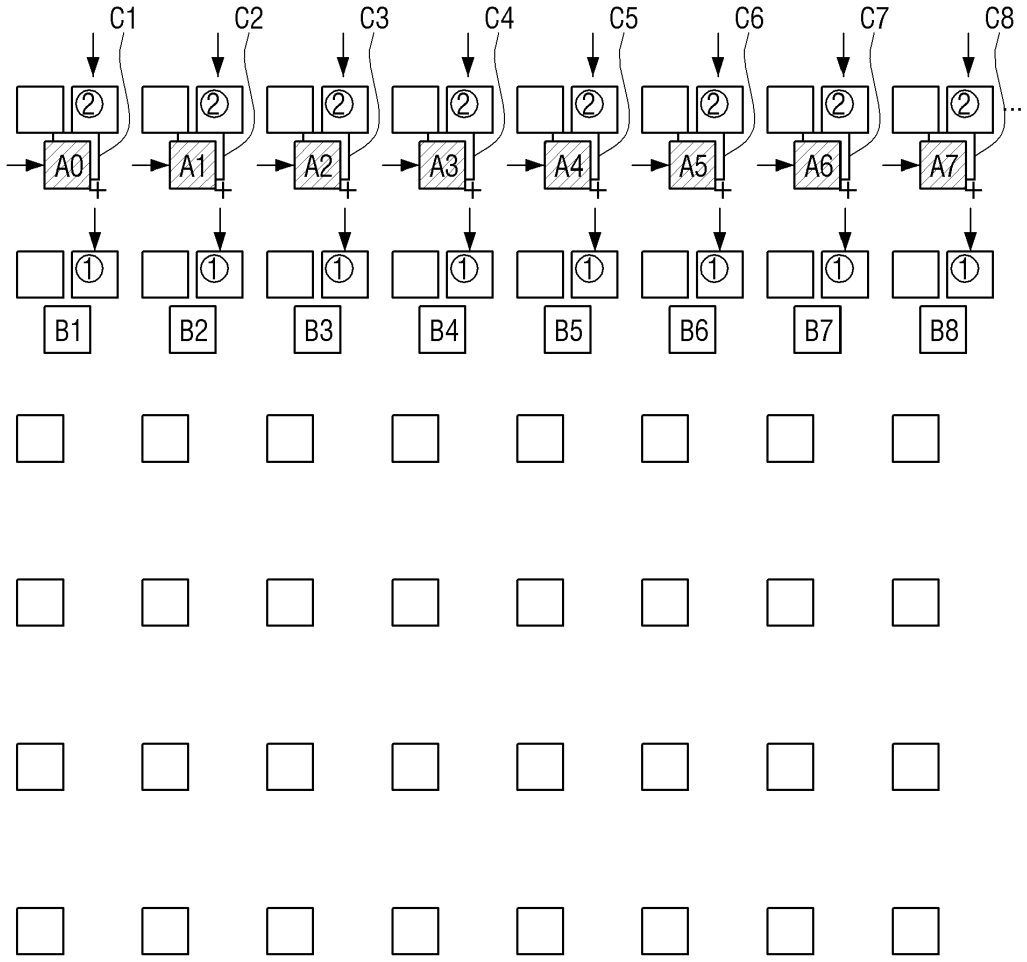
도면7a



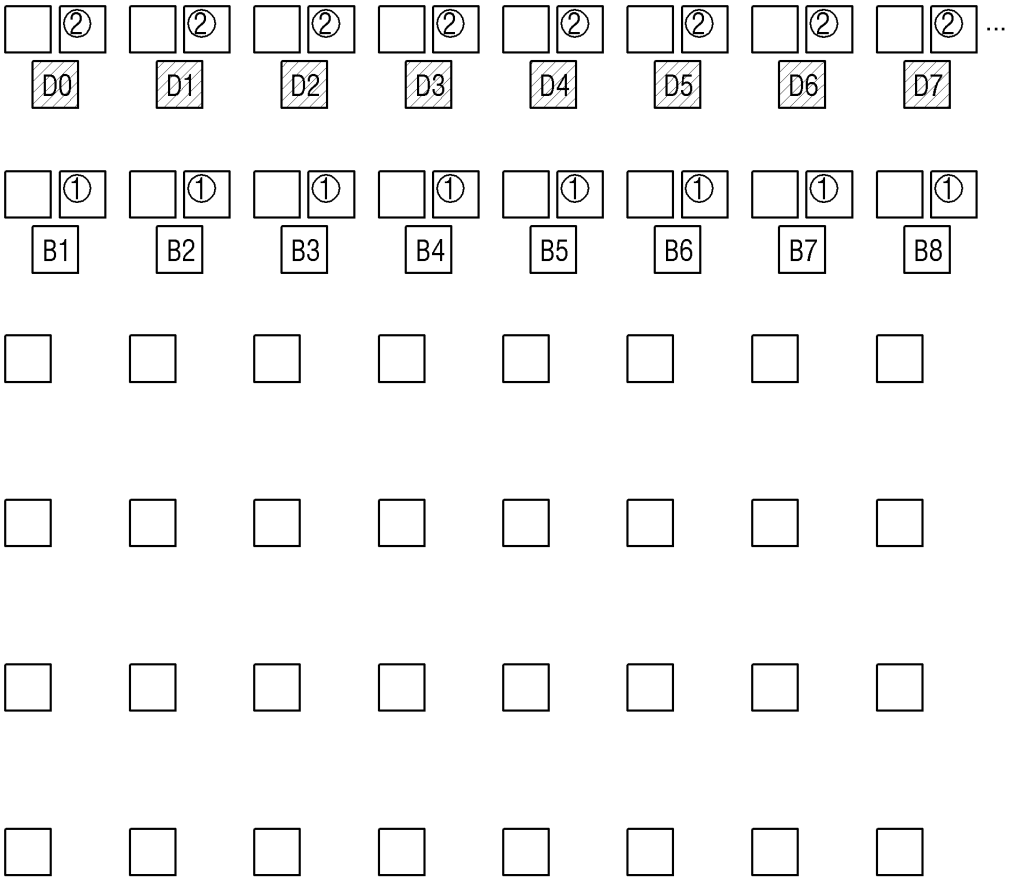
도면7b



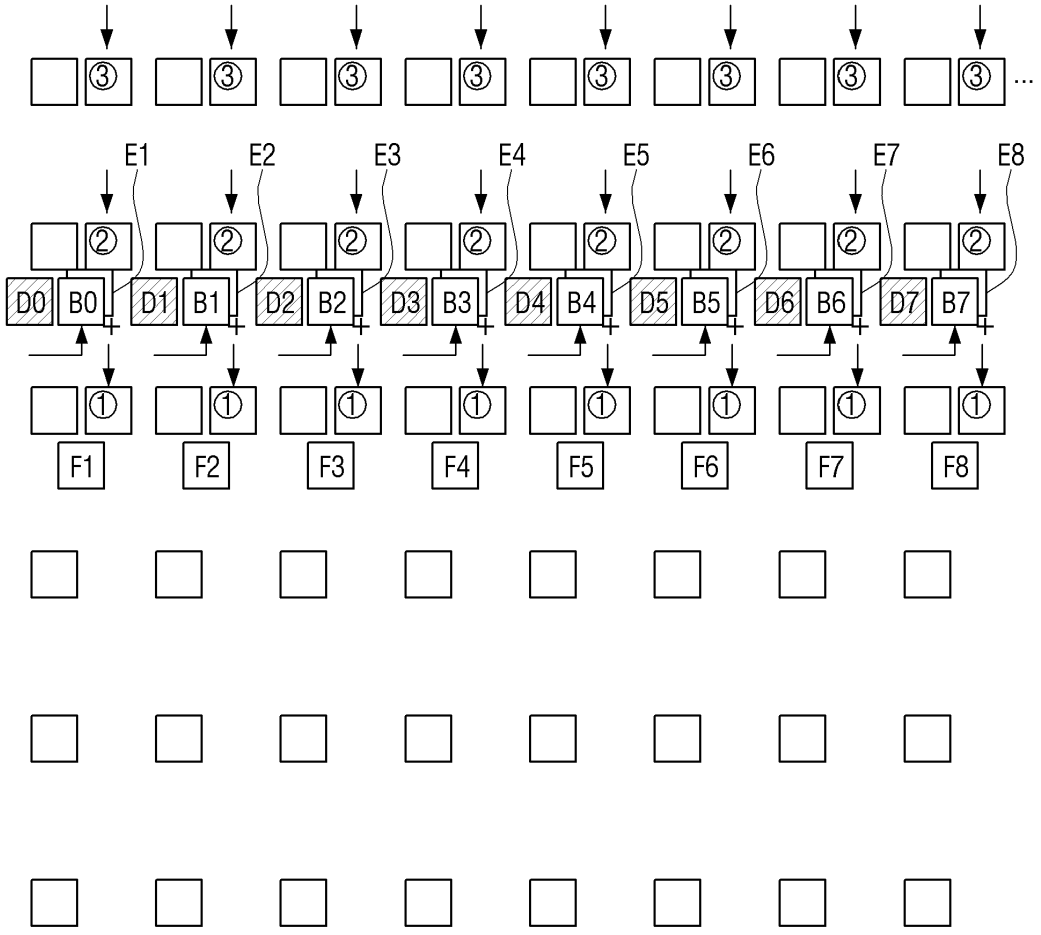
도면7c


C1
C2
C3
C4
C5
C6
C7
C8
②
A0
A1
A2
A3
A4
A5
A6
A7
①
B1
B2
B3
B4
B5
B6
B7
B8

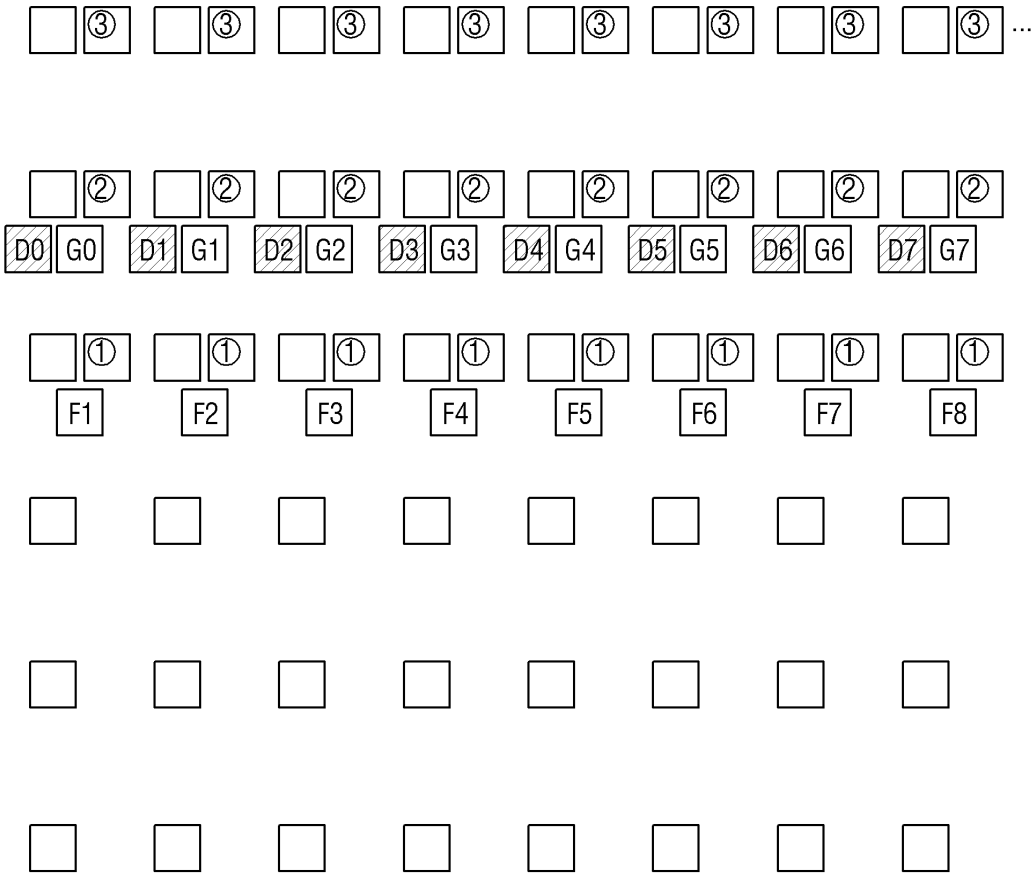
도면7d



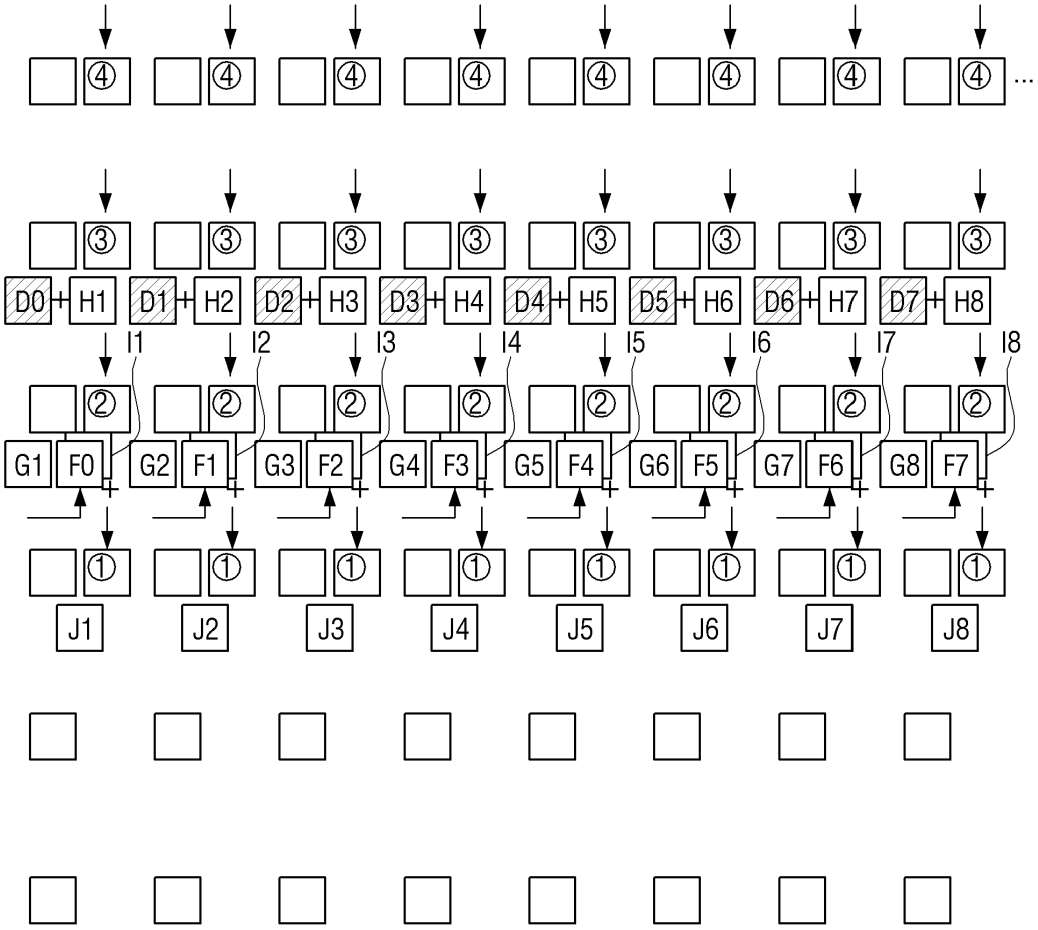
도면7e


③
E1
E2
E3
E4
E5
E6
E7
E8
②
D0
B0
D1
B1
D2
B2
D3
B3
D4
B4
D5
B5
D6
B6
D7
B7
①
F1
F2
F3
F4
F5
F6
F7
F8

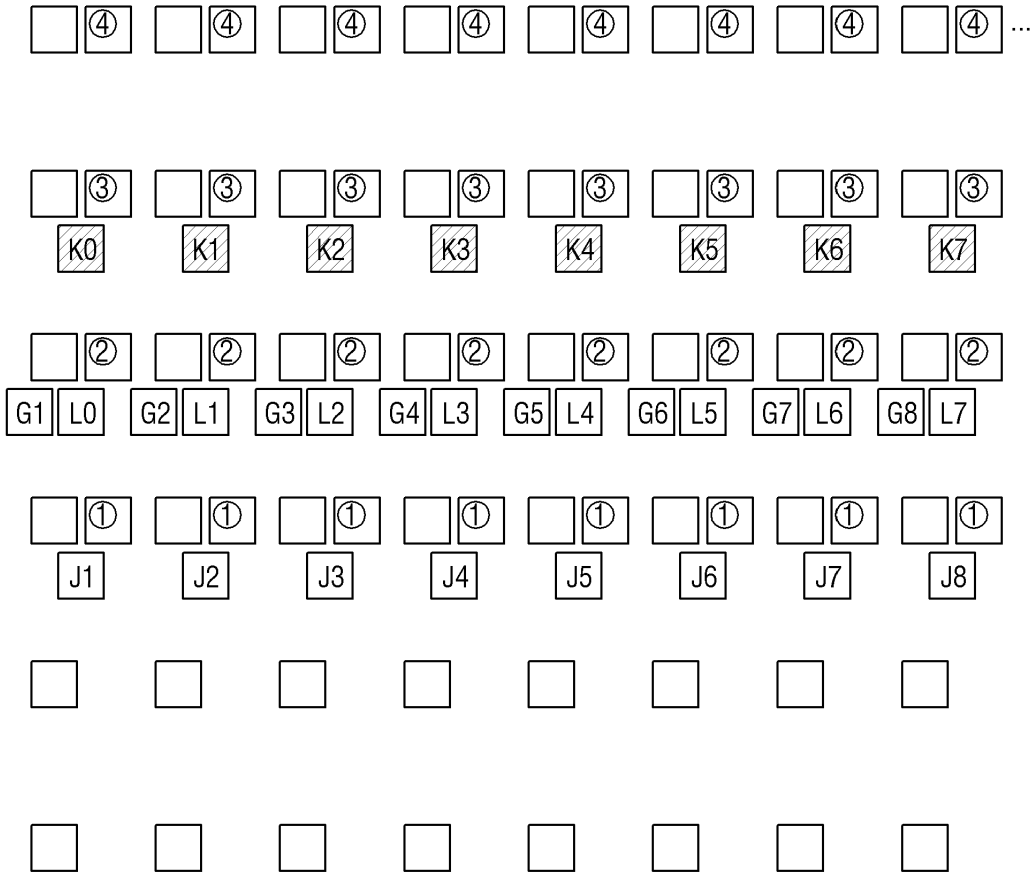
도면7f


③ ③ ③ ③ ③ ③ ③ ③ ...
② ② ② ② ② ② ② ②
D0 G0 D1 G1 D2 G2 D3 G3 D4 G4 D5 G5 D6 G6 D7 G7
① ① ① ① ① ① ① ①
F1 F2 F3 F4 F5 F6 F7 F8

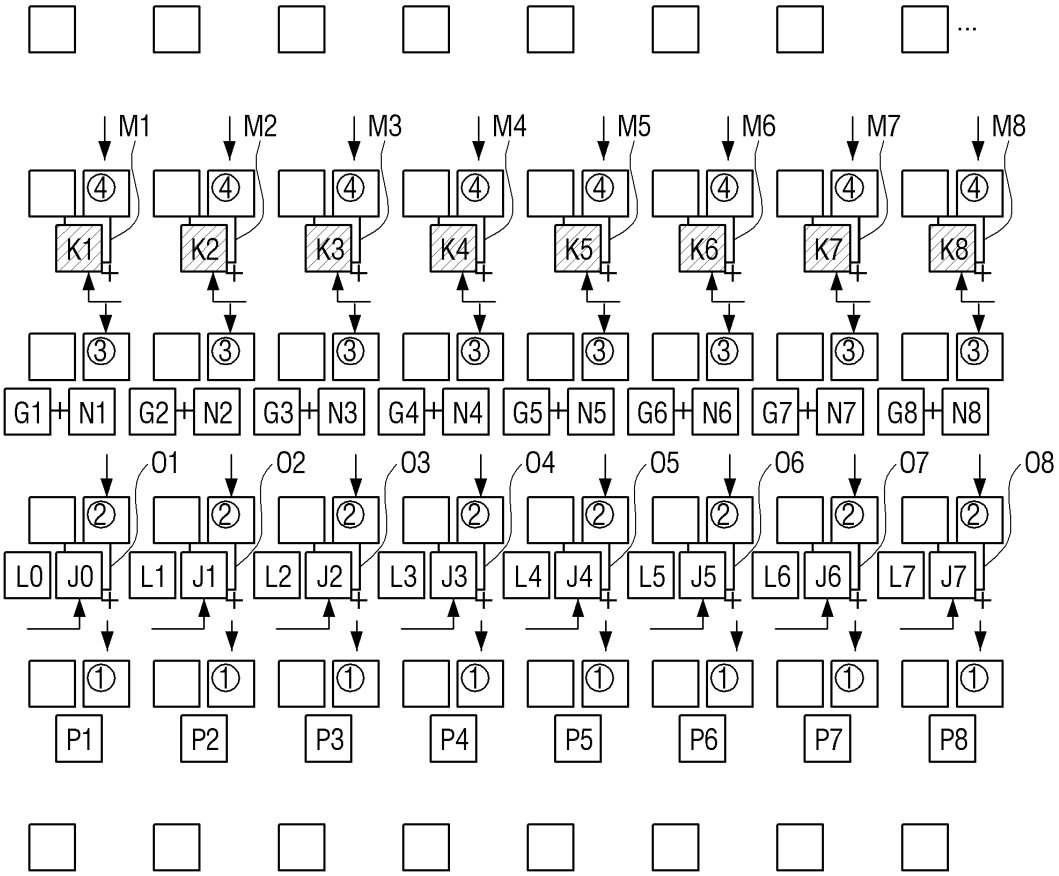
도면7g


④
③
D0
H1
D1
H2
D2
H3
D3
H4
D4
H5
D5
H6
D6
H7
D7
H8
I1
I2
I3
I4
I5
I6
I7
I8
②
G1
F0
G2
F1
G3
F2
G4
F3
G5
F4
G6
F5
G7
F6
G8
F7
①
J1
J2
J3
J4
J5
J6
J7
J8

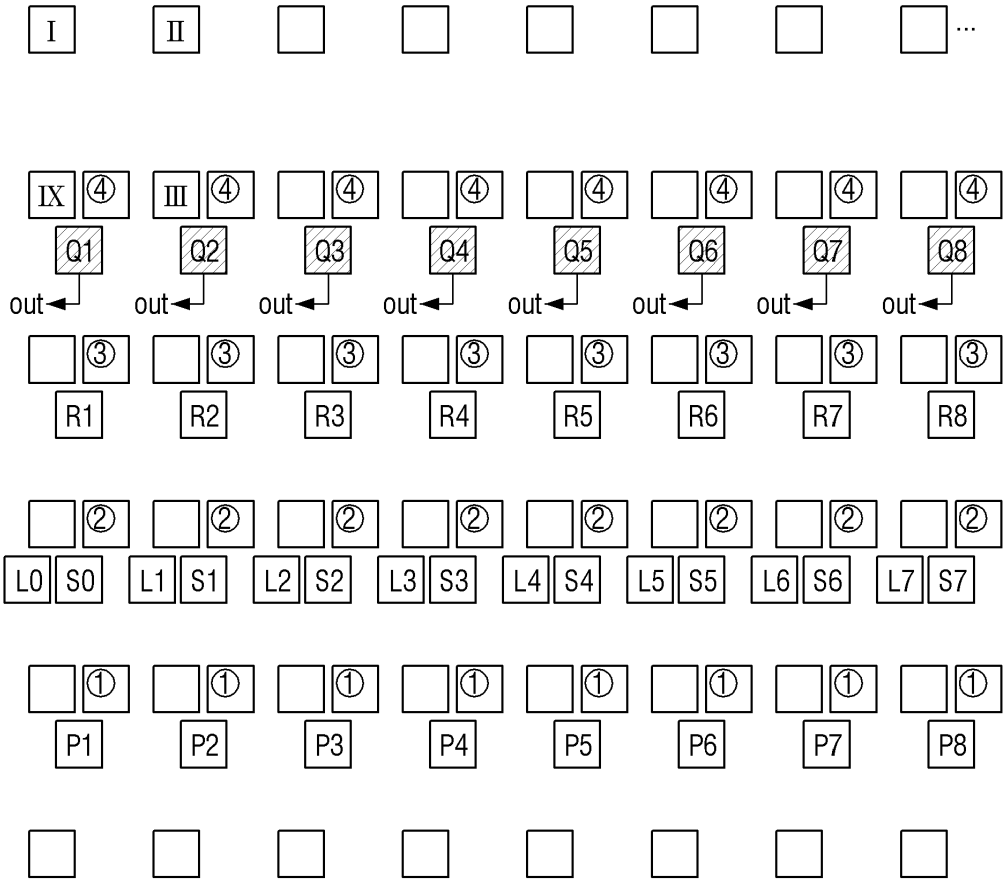
도면7h


④
③
K0
K1
K2
K3
K4
K5
K6
K7
②
G1
L0
G2
L1
G3
L2
G4
L3
G5
L4
G6
L5
G7
L6
G8
L7
①
J1
J2
J3
J4
J5
J6
J7
J8

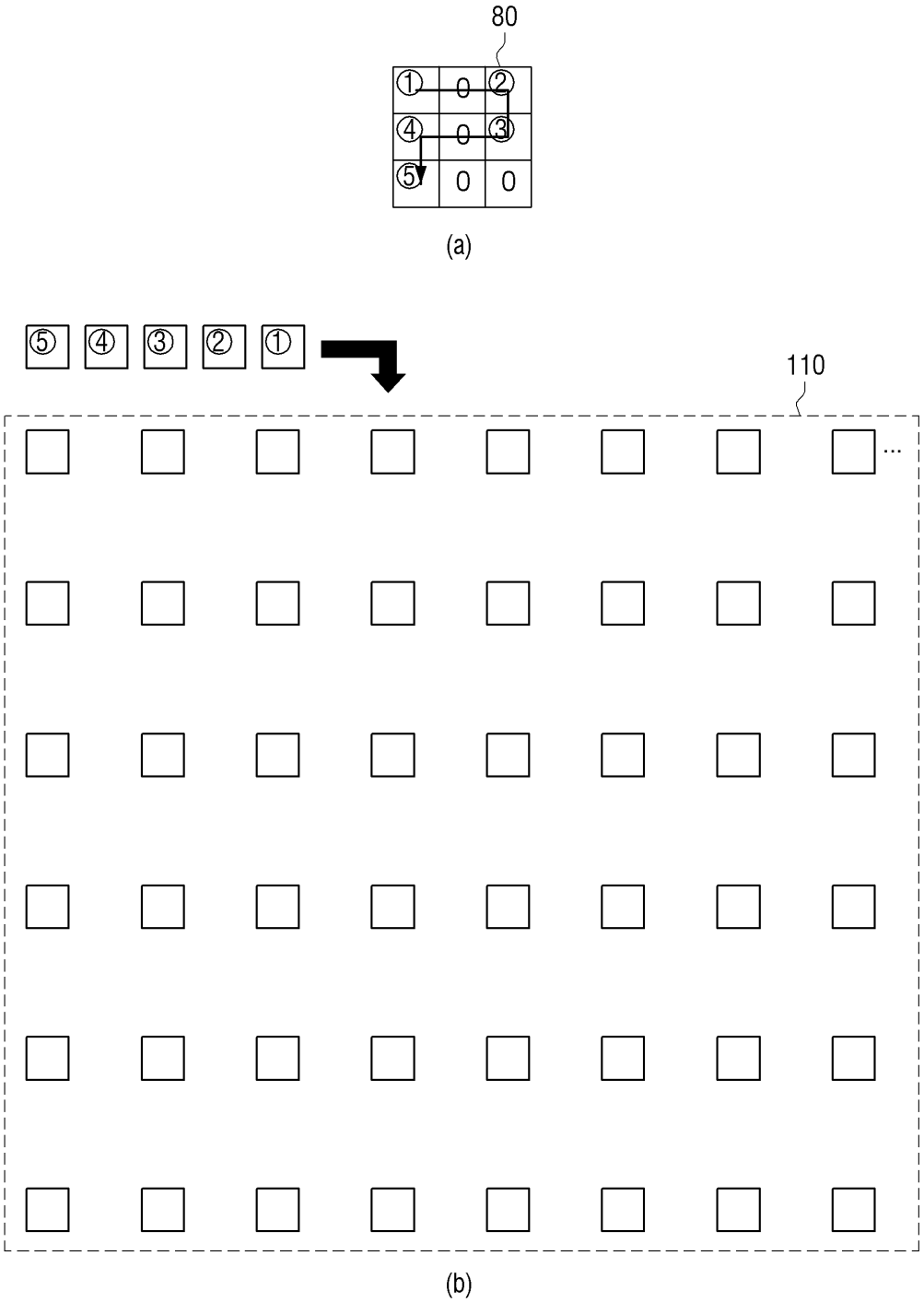
도면7i


M1
M2
M3
M4
M5
M6
M7
M8
④
K1
K2
K3
K4
K5
K6
K7
K8
③
G1
N1
G2
N2
G3
N3
G4
N4
G5
N5
G6
N6
G7
N7
G8
N8
O1
O2
O3
O4
O5
O6
O7
O8
②
L0
J0
L1
J1
L2
J2
L3
J3
L4
J4
L5
J5
L6
J6
L7
J7
①
P1
P2
P3
P4
P5
P6
P7
P8

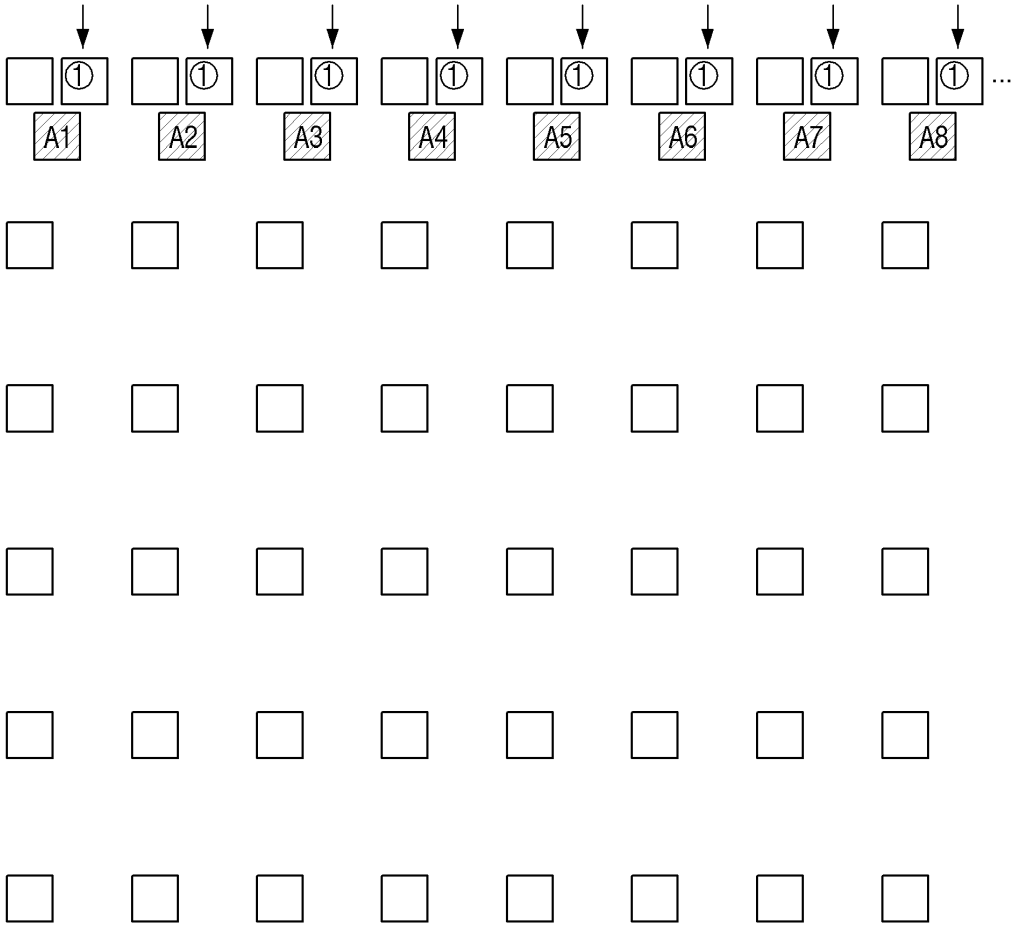
도면7j


I
II
...
IX
④
Q1
out
III
Q2
Q3
Q4
Q5
Q6
Q7
Q8
③
R1
R2
R3
R4
R5
R6
R7
R8
②
L0
S0
L1
S1
L2
S2
L3
S3
L4
S4
L5
S5
L6
S6
L7
S7
①
P1
P2
P3
P4
P5
P6
P7
P8

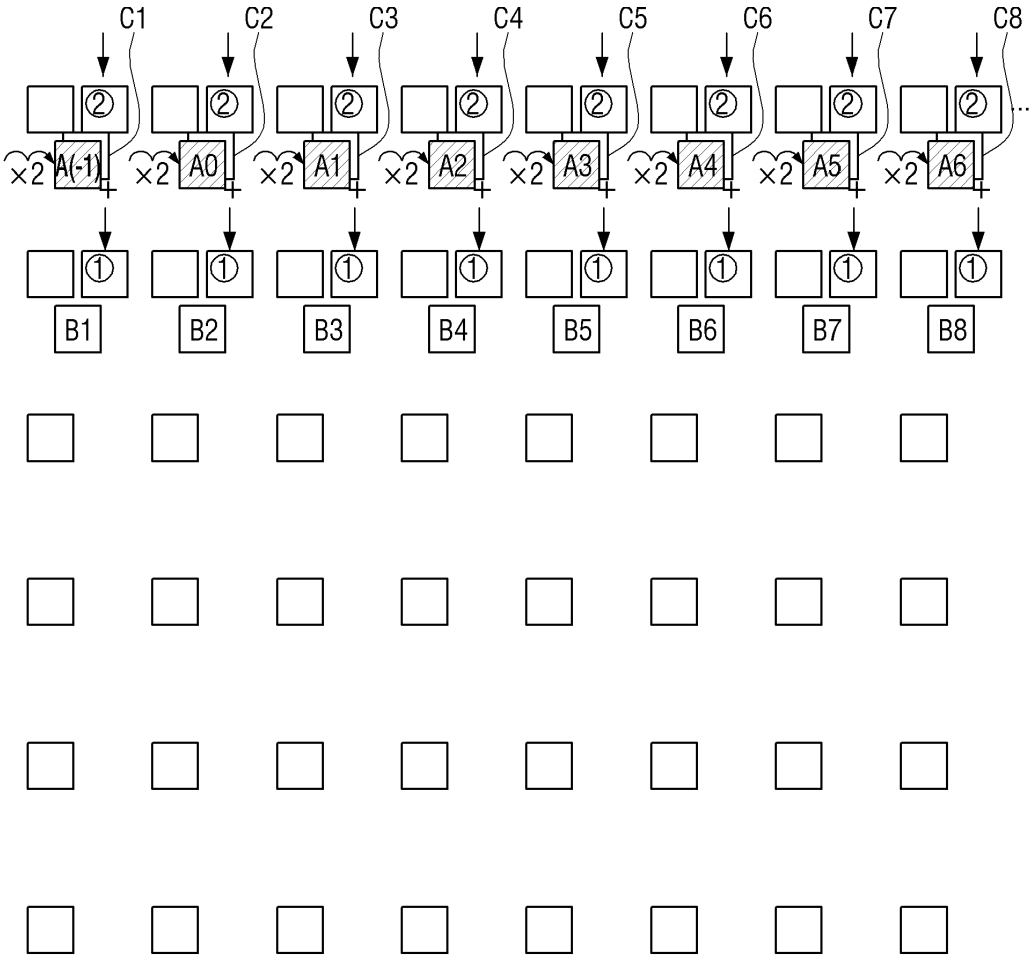
도면8a



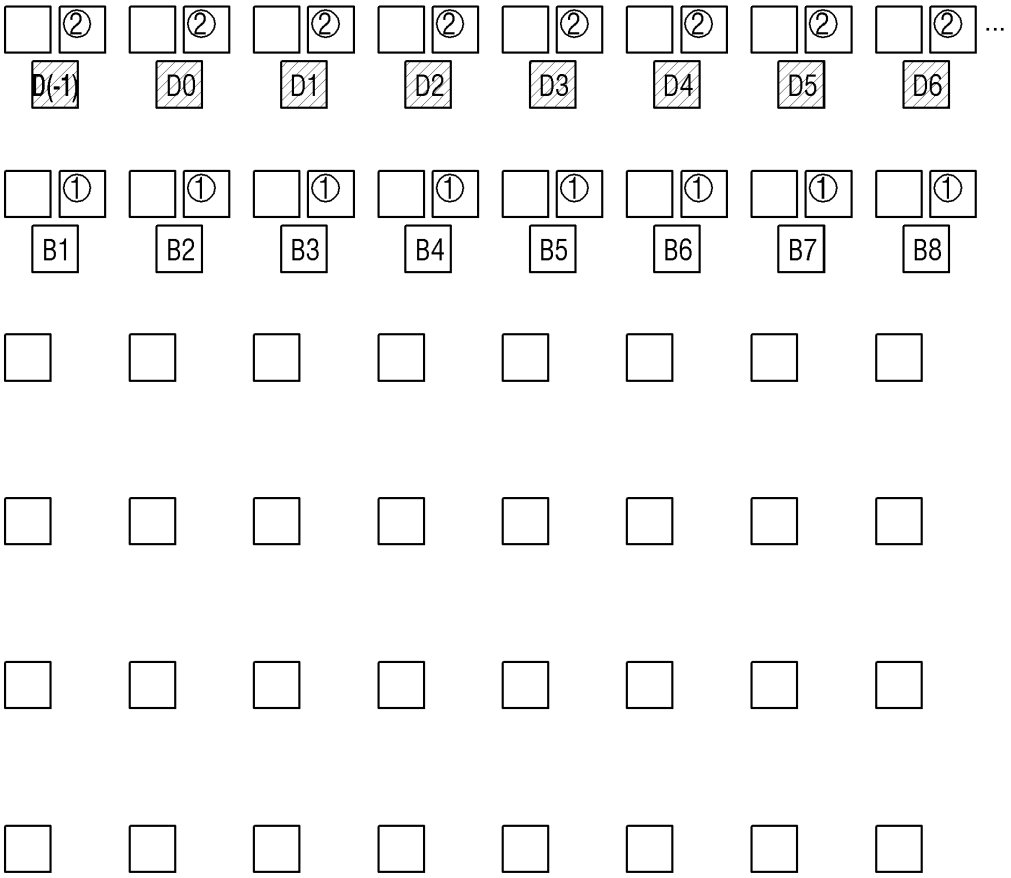
도면8b



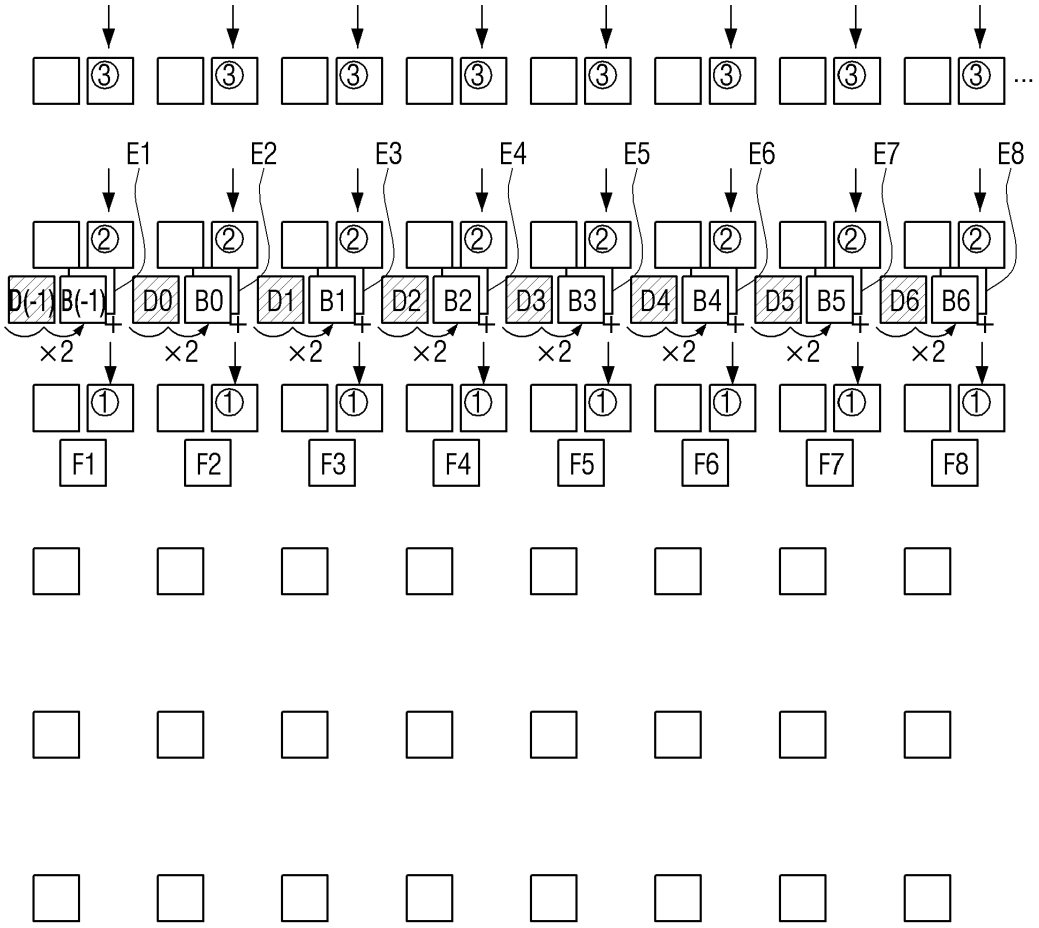
도면8c


C1
C2
C3
C4
C5
C6
C7
C8
②
×2
A(-1)
A0
A1
A2
A3
A4
A5
A6
①
B1
B2
B3
B4
B5
B6
B7
B8

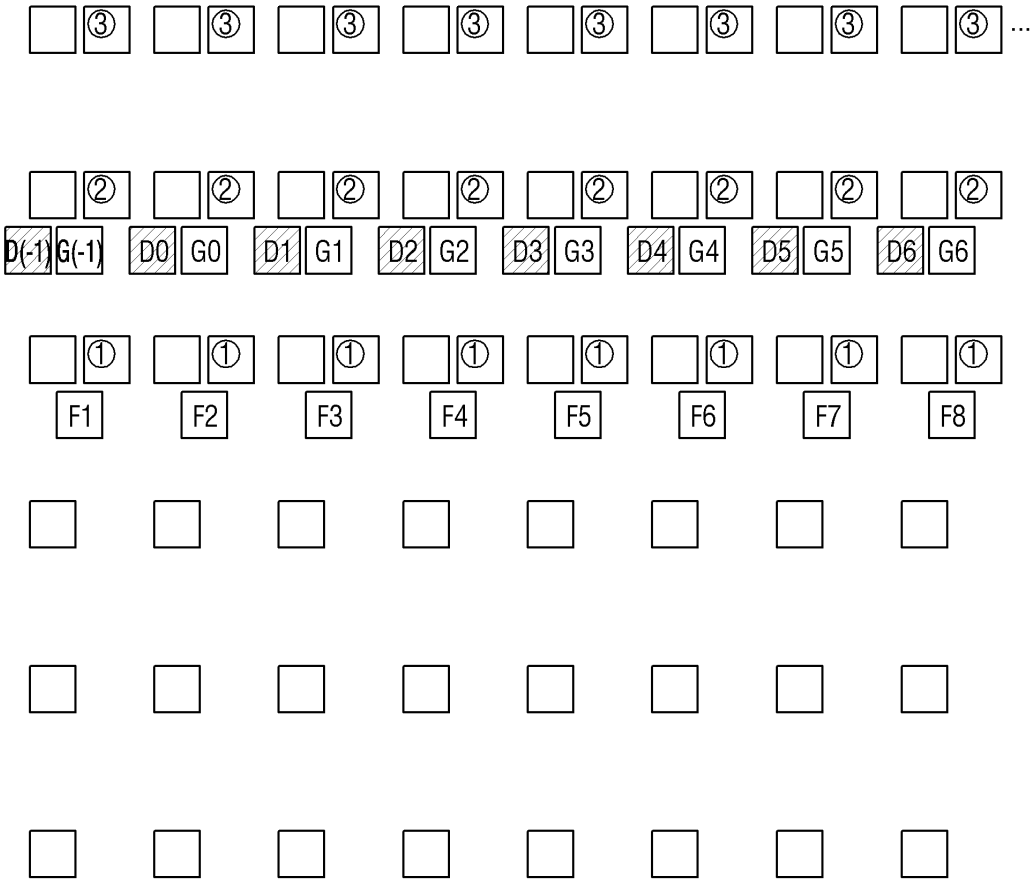
도면8d



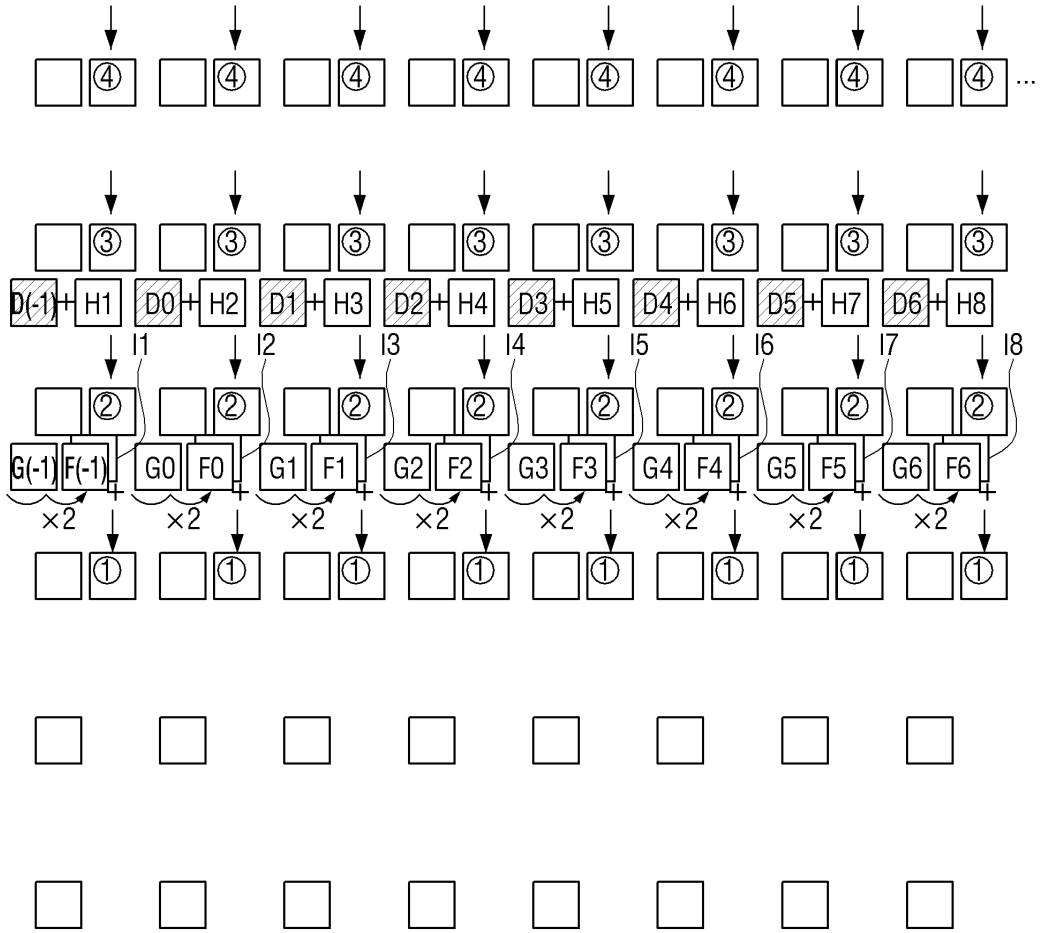
도면8e


③
E1
E2
E3
E4
E5
E6
E7
E8
②
D(-1)
B(-1)
D0
B0
D1
B1
D2
B2
D3
B3
D4
B4
D5
B5
D6
B6
×2
①
F1
F2
F3
F4
F5
F6
F7
F8

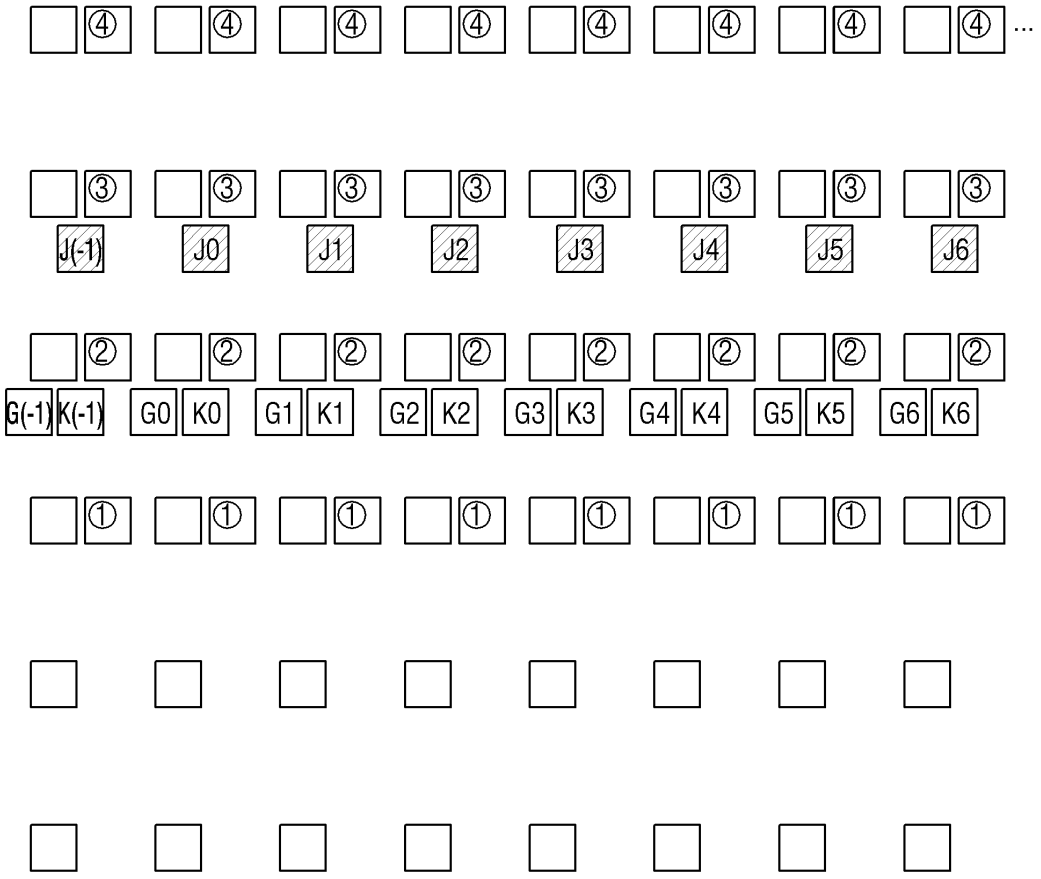
도면8f


③
②
D(-1)
G(-1)
D0
G0
D1
G1
D2
G2
D3
G3
D4
G4
D5
G5
D6
G6
①
F1
F2
F3
F4
F5
F6
F7
F8

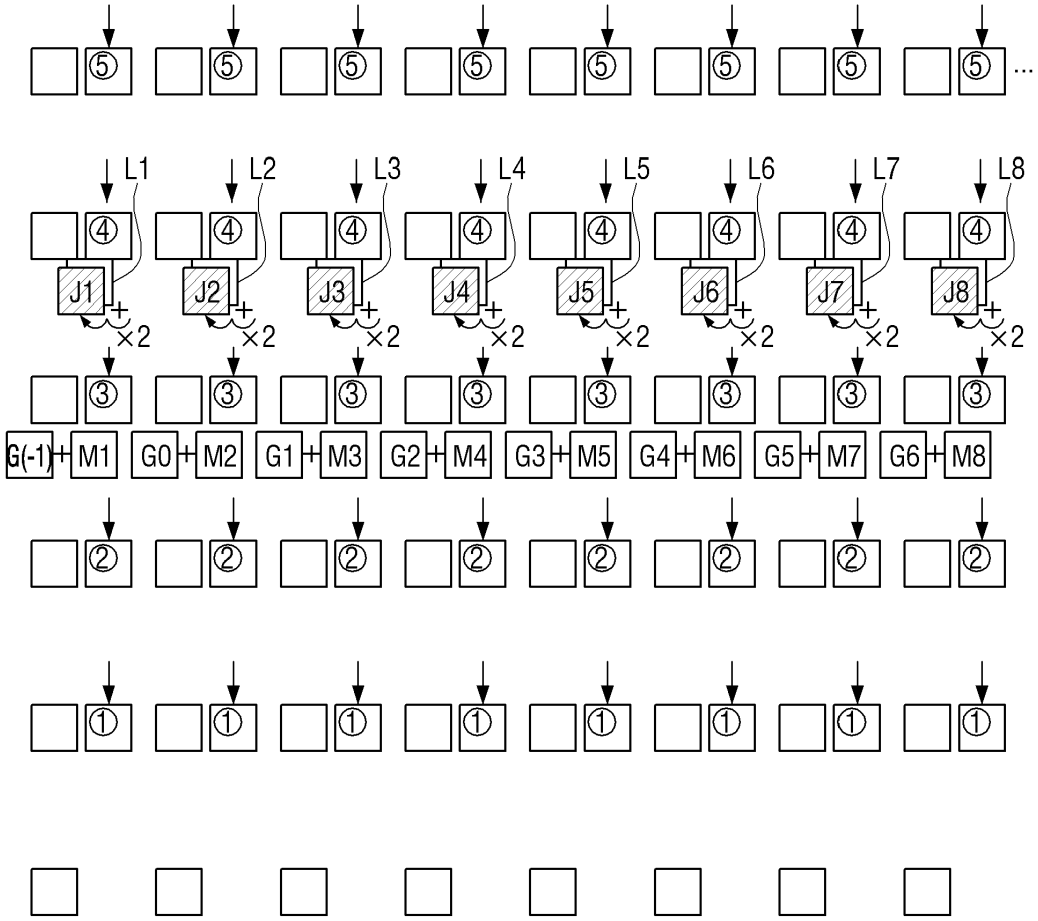
도면8g


④
③
D(-1)
H1
D0
H2
D1
H3
D2
H4
D3
H5
D4
H6
D5
H7
D6
H8
I1
I2
I3
I4
I5
I6
I7
I8
②
G(-1)
F(-1)
G0
F0
G1
F1
G2
F2
G3
F3
G4
F4
G5
F5
G6
F6
×2
①

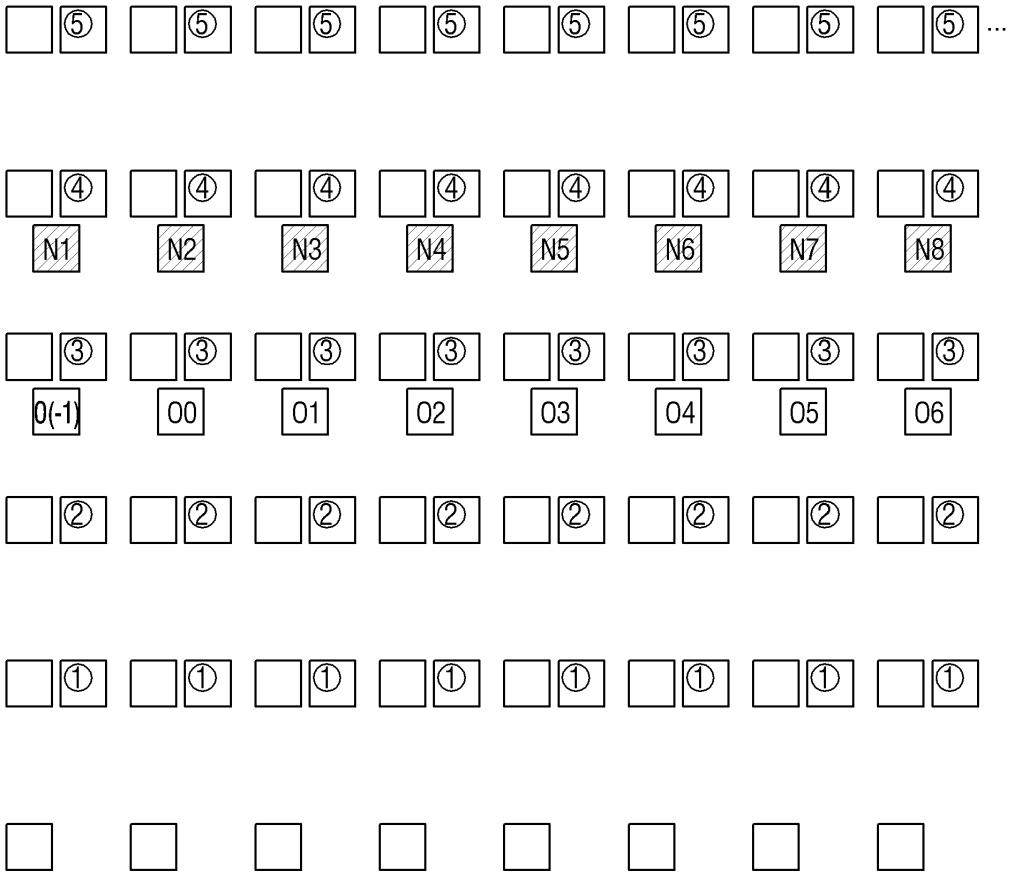
도면8h


④
③
J(-1)
J0
J1
J2
J3
J4
J5
J6
②
G(-1)
K(-1)
G0
K0
G1
K1
G2
K2
G3
K3
G4
K4
G5
K5
G6
K6
①

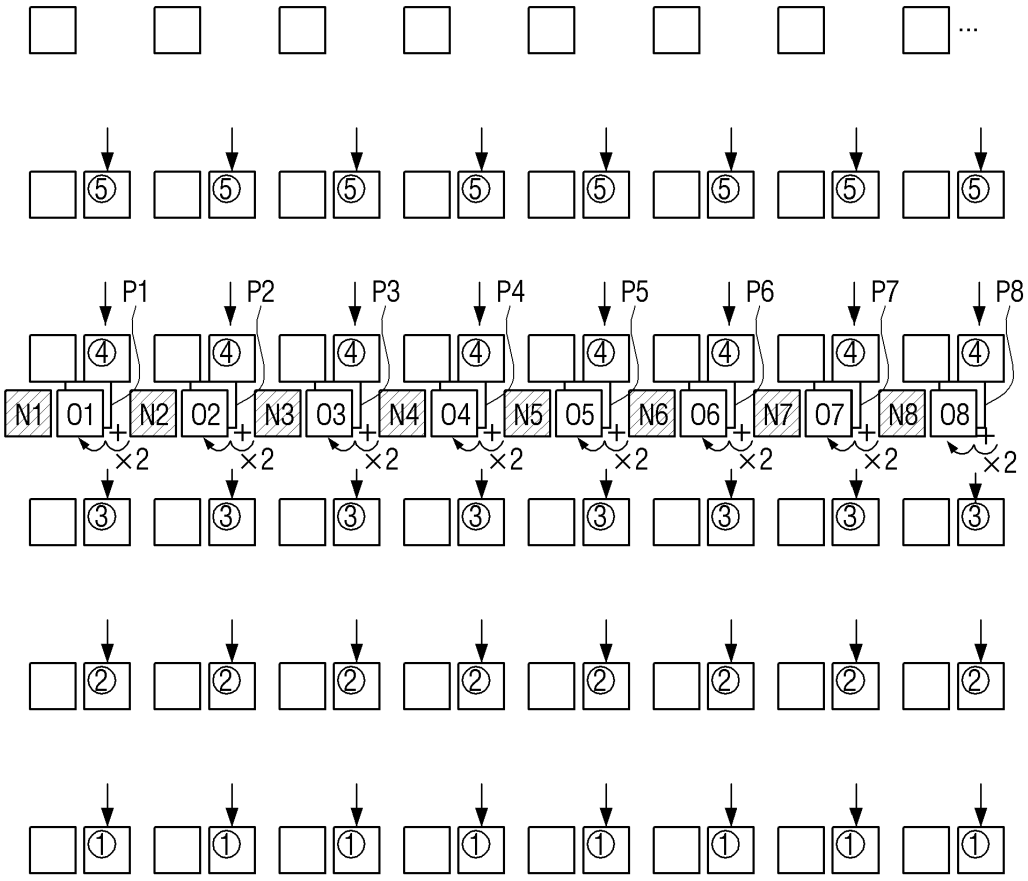
도면8i


⑤
L1
L2
L3
L4
L5
L6
L7
L8
④
J1
J2
J3
J4
J5
J6
J7
J8
×2
③
G(-1)
M1
G0
M2
G1
M3
G2
M4
G3
M5
G4
M6
G5
M7
G6
M8
②
①

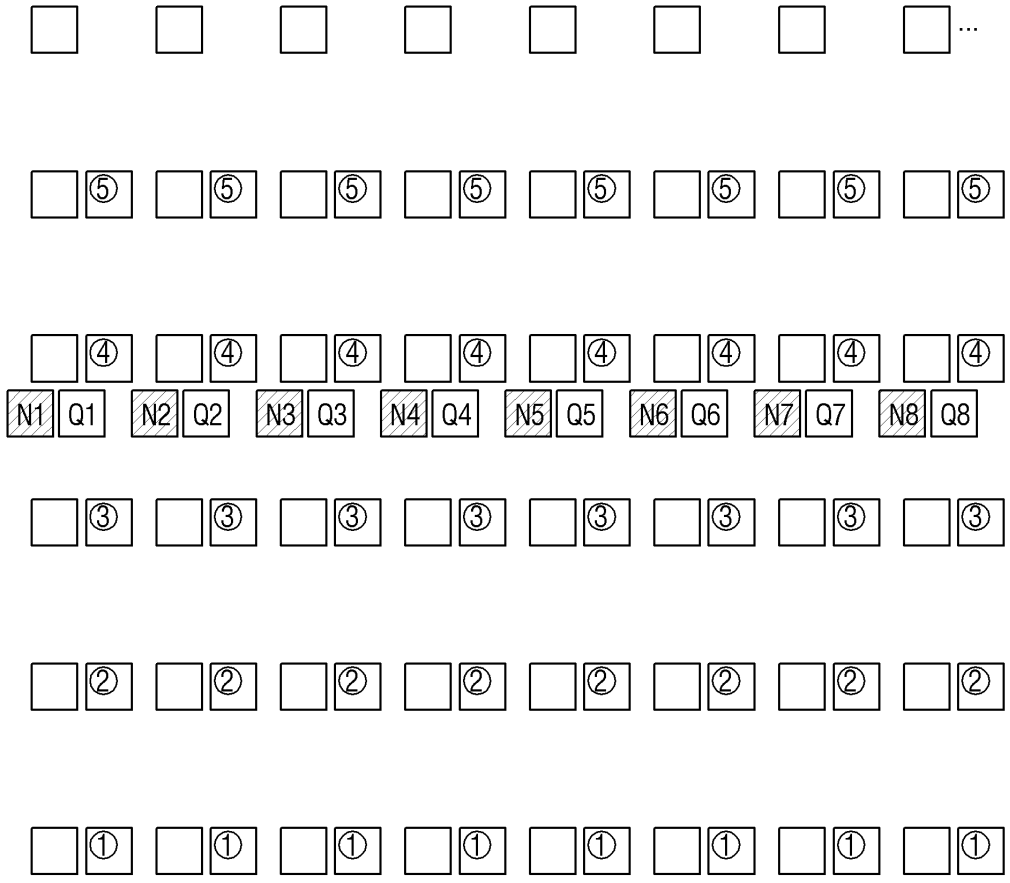
도면8j



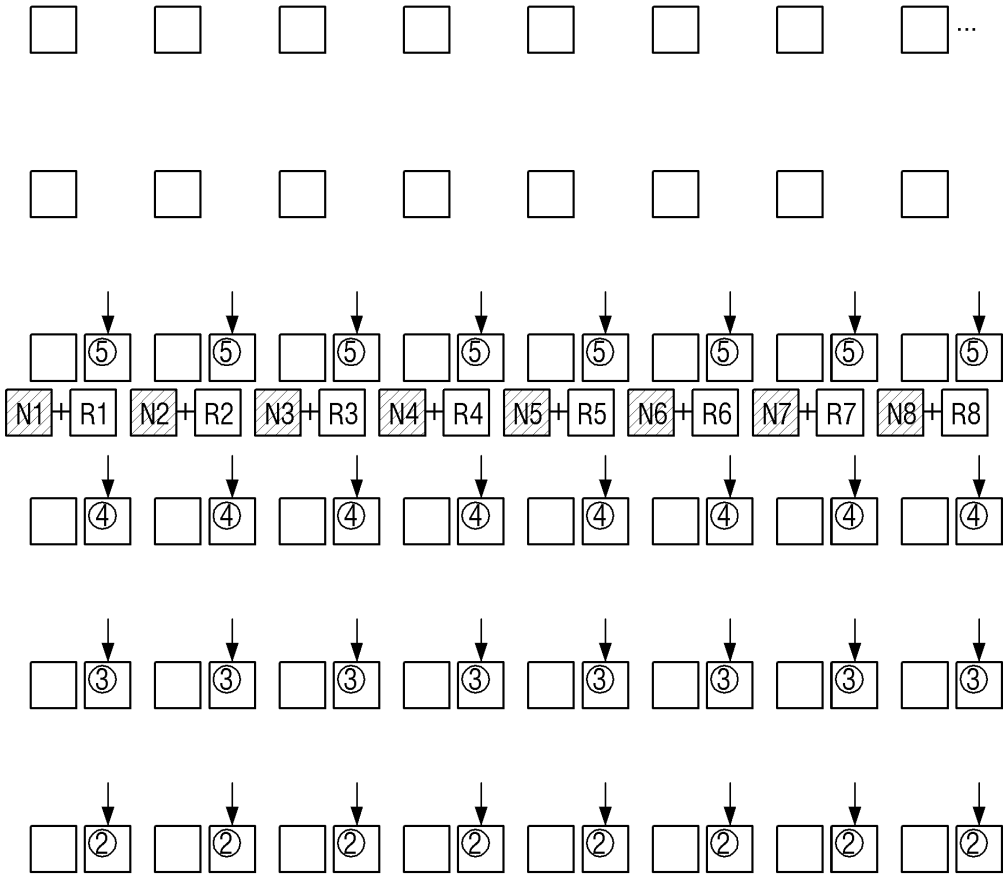
도면8k


⑤
P1
P2
P3
P4
P5
P6
P7
P8
④
N1
O1
N2
O2
N3
O3
N4
O4
N5
O5
N6
O6
N7
O7
N8
O8
×2
③
②
①

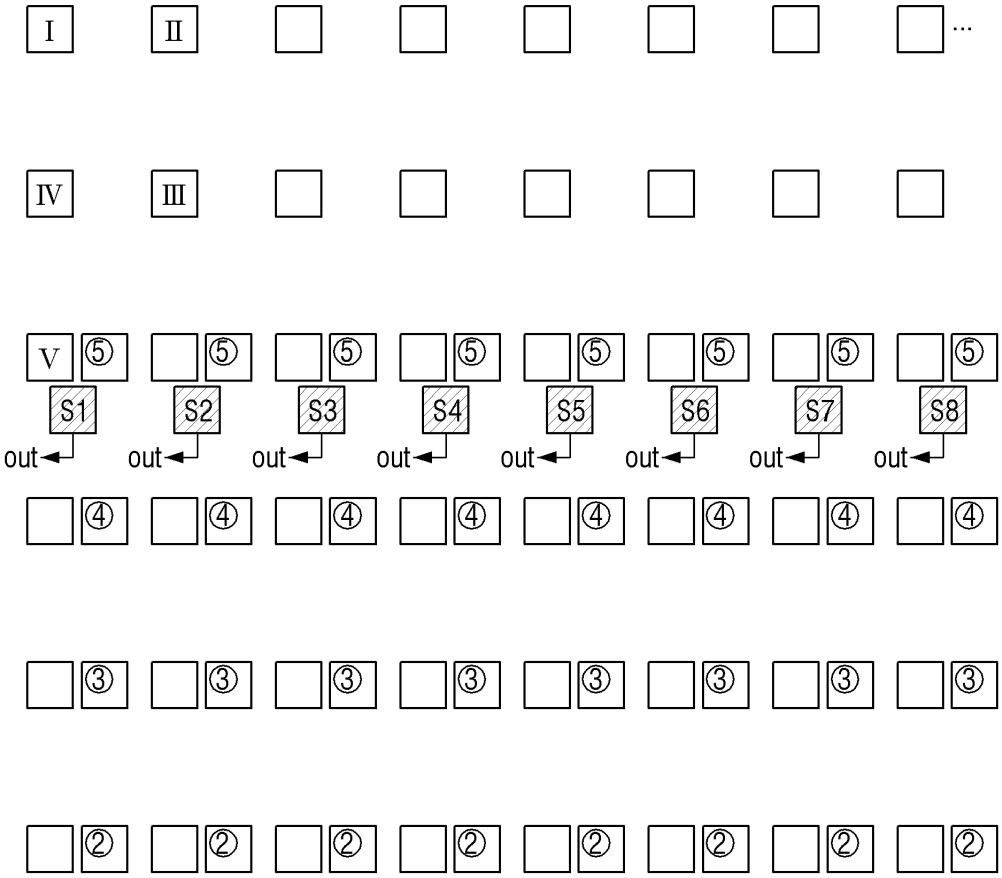
도면81



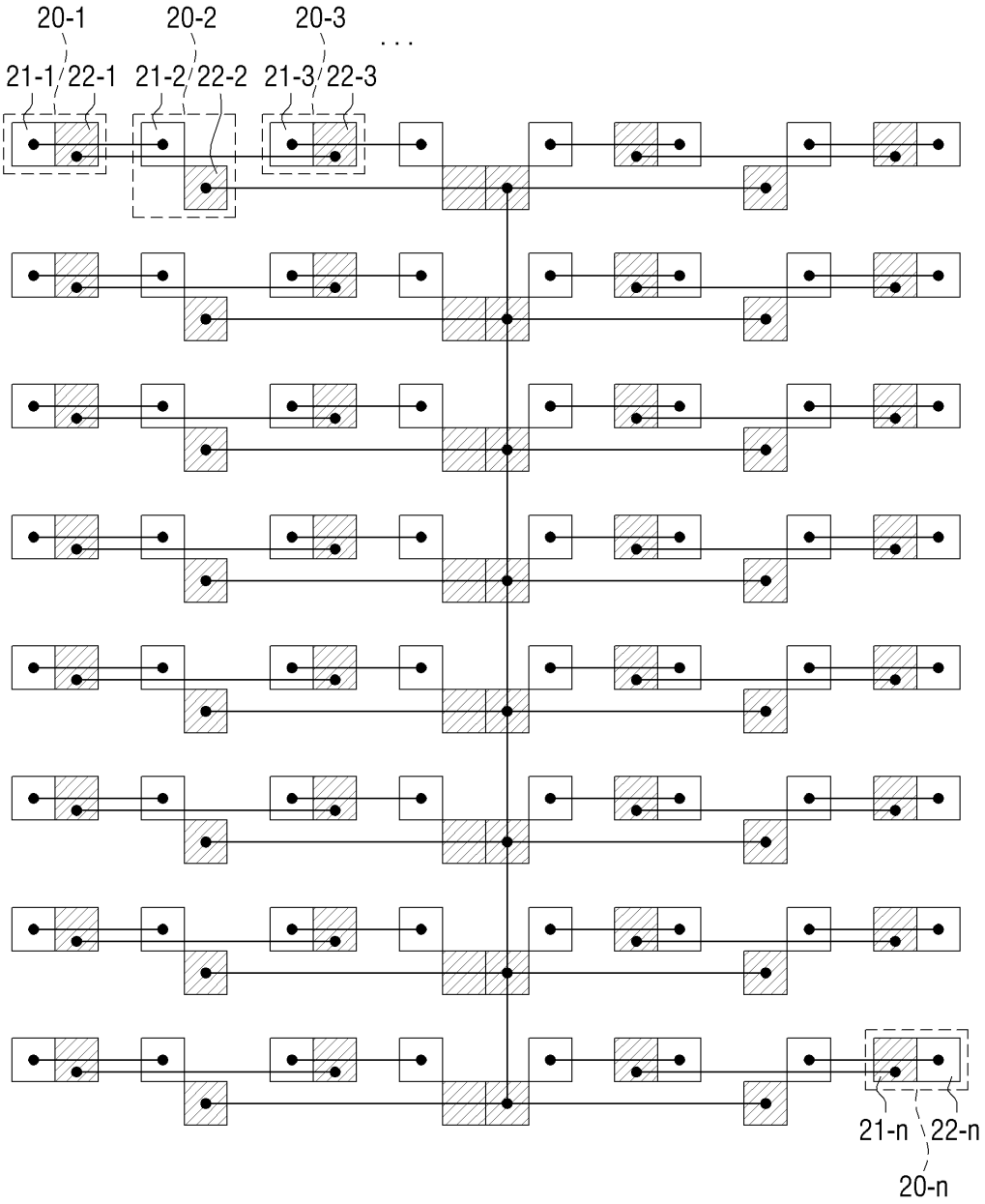
도면8m



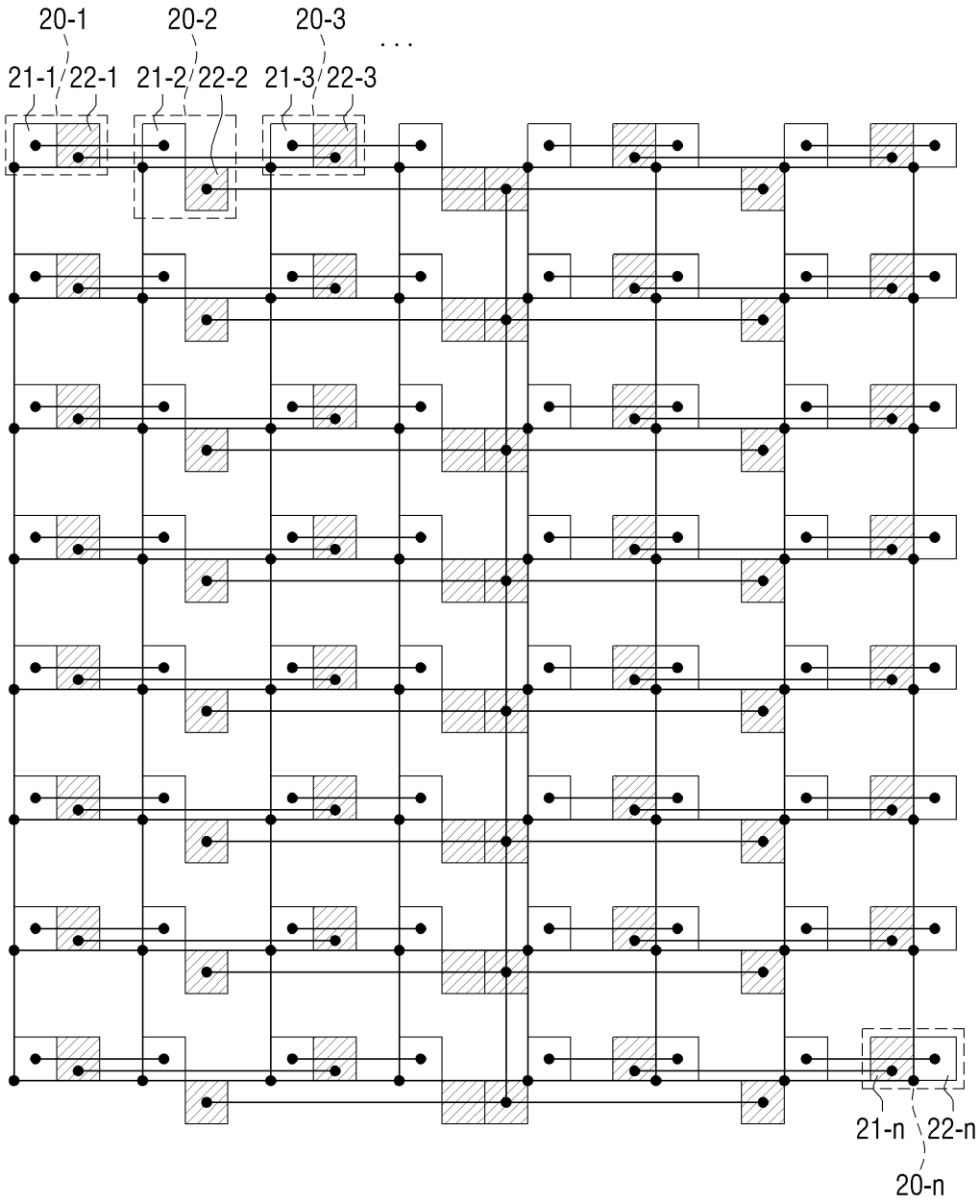
도면8n



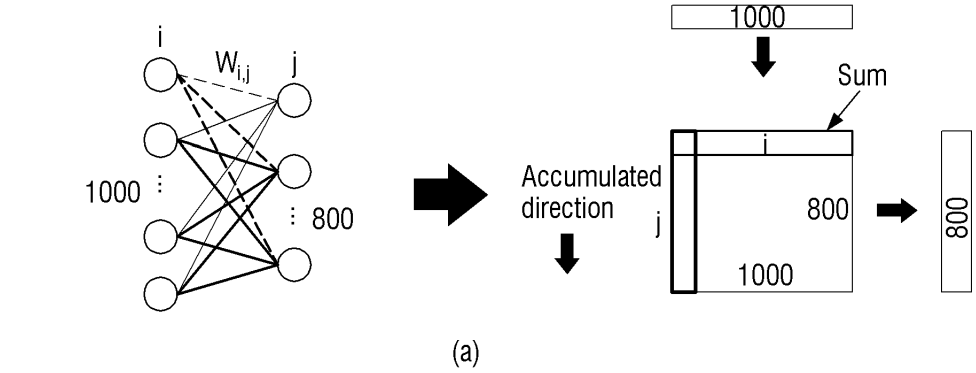
도면9



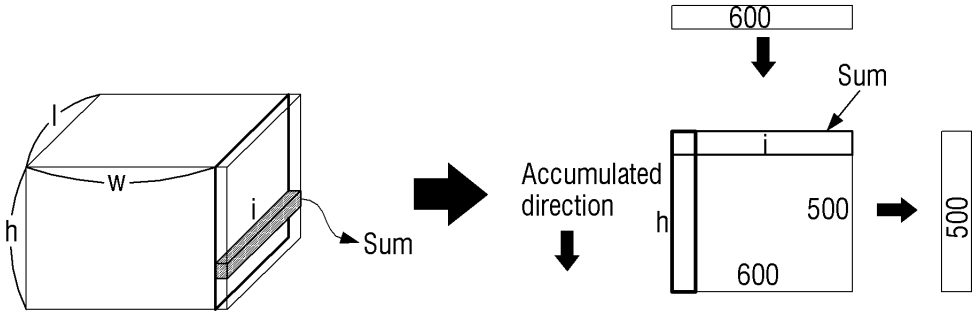
도면10



도면11

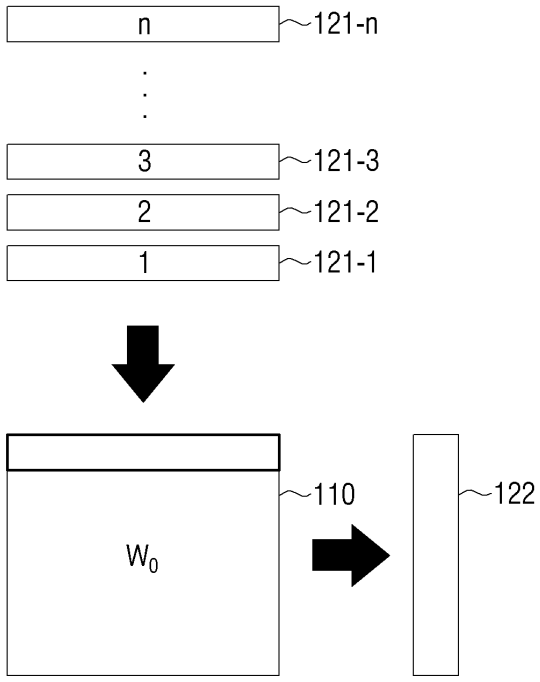

i
$W_{i,j}$
j
1000
800
1000
Sum
Accumulated direction
i
j
800
1000
800

(a)

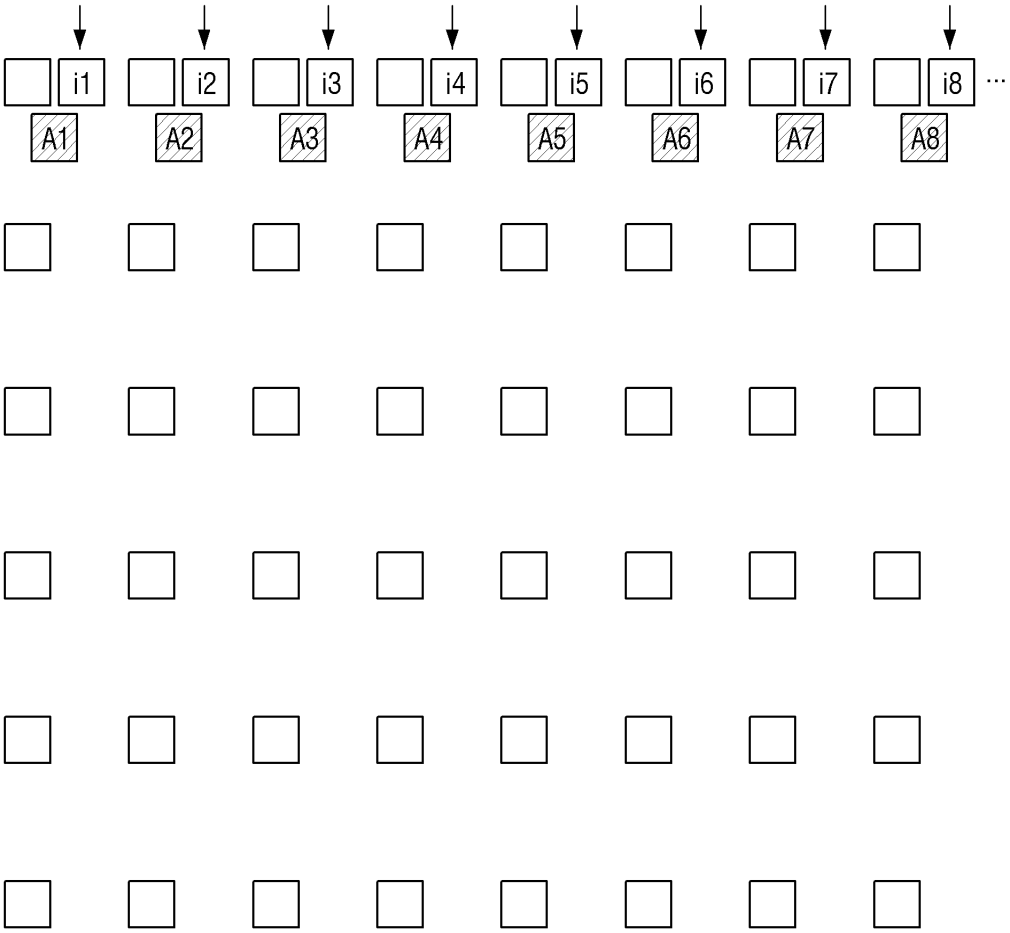

l
w
h
i
Sum
600
Sum
Accumulated direction
i
h
500
600
500

(b)

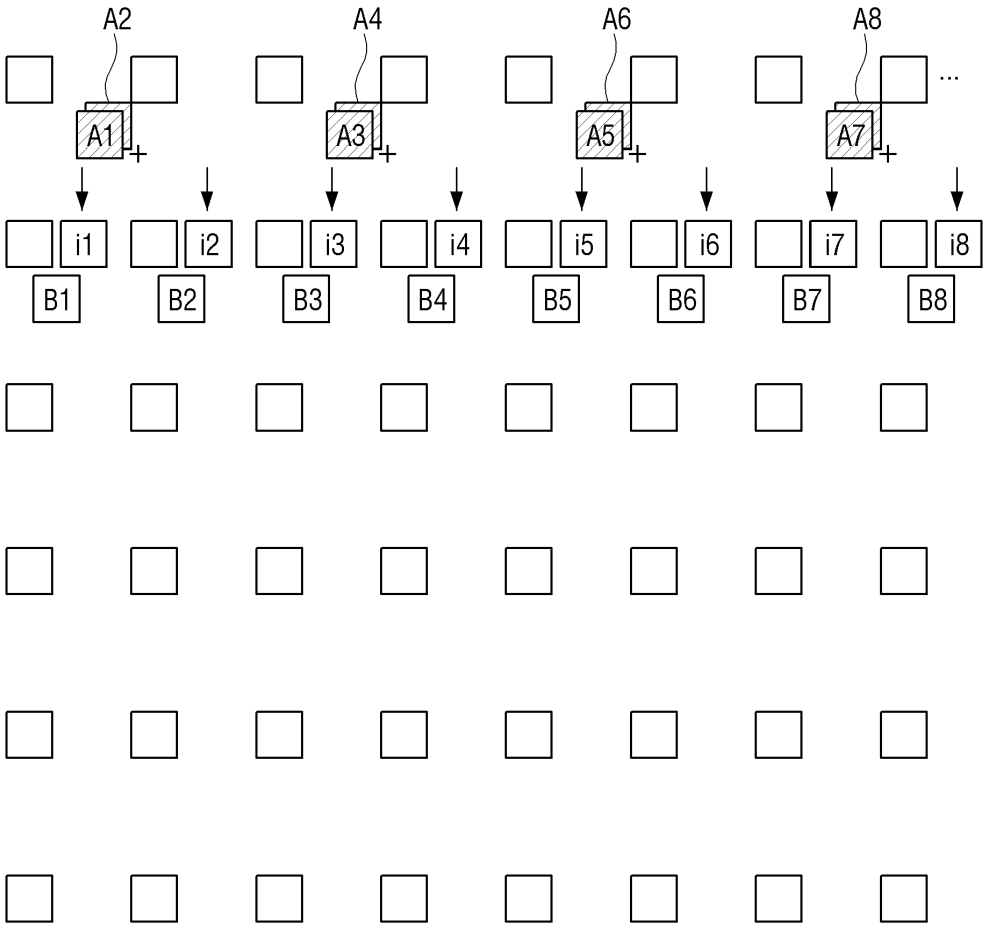
도면12


n
121-n
3
121-3
2
121-2
1
121-1
W_0
110
122

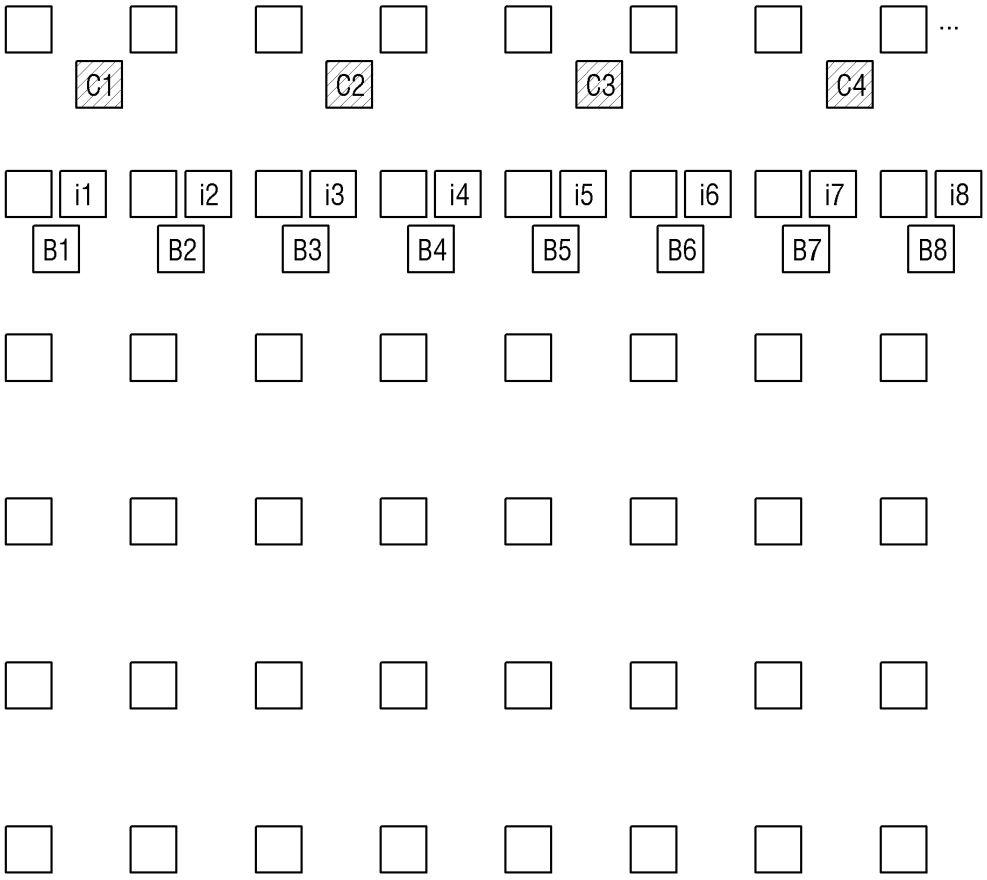
도면13a



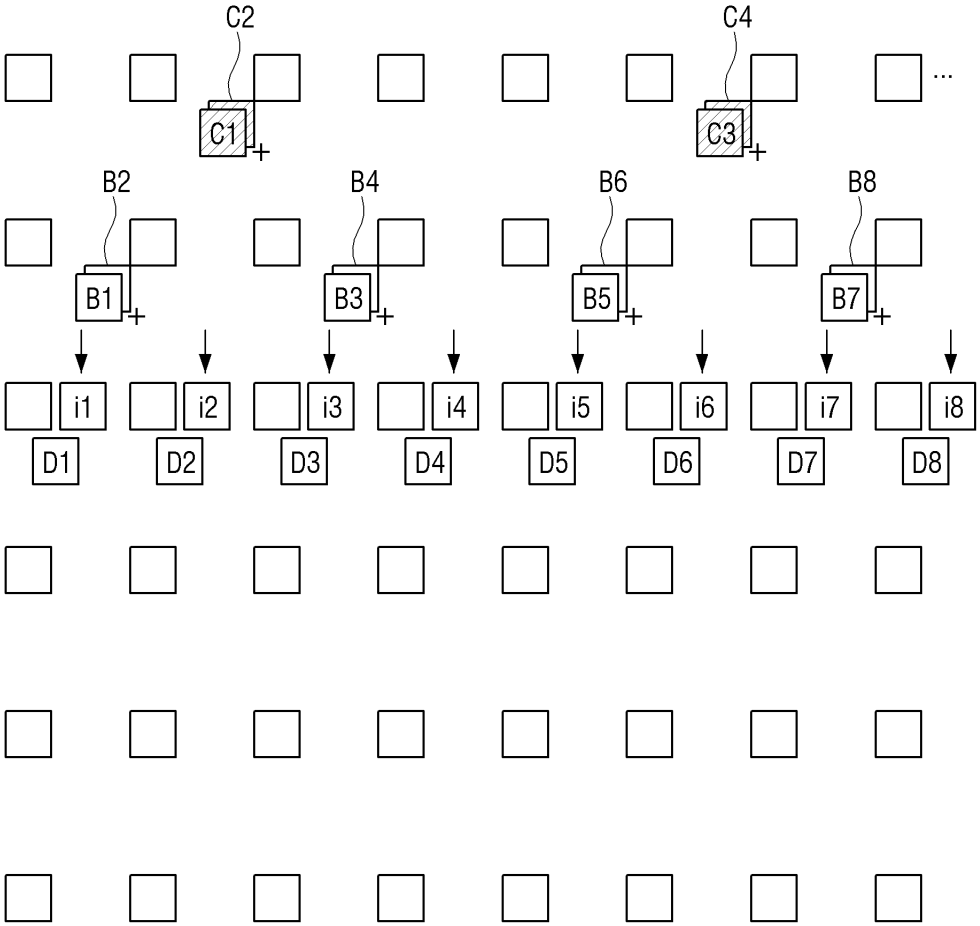
도면13b



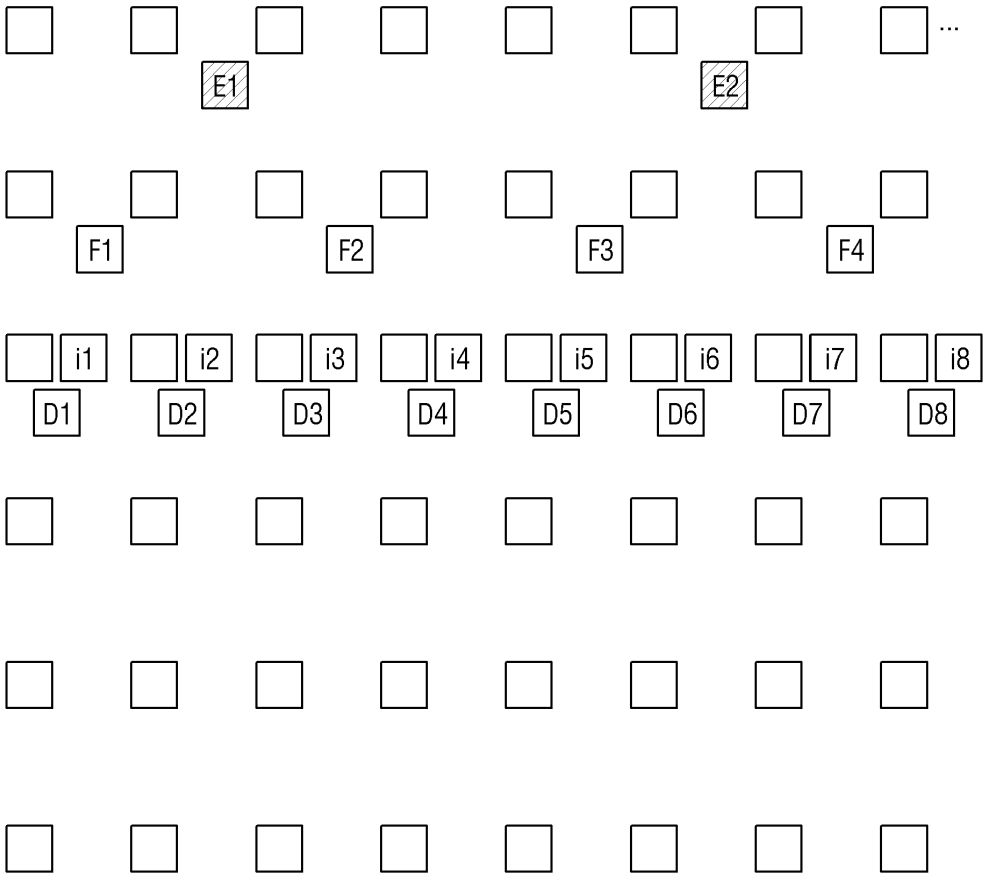
도면13c



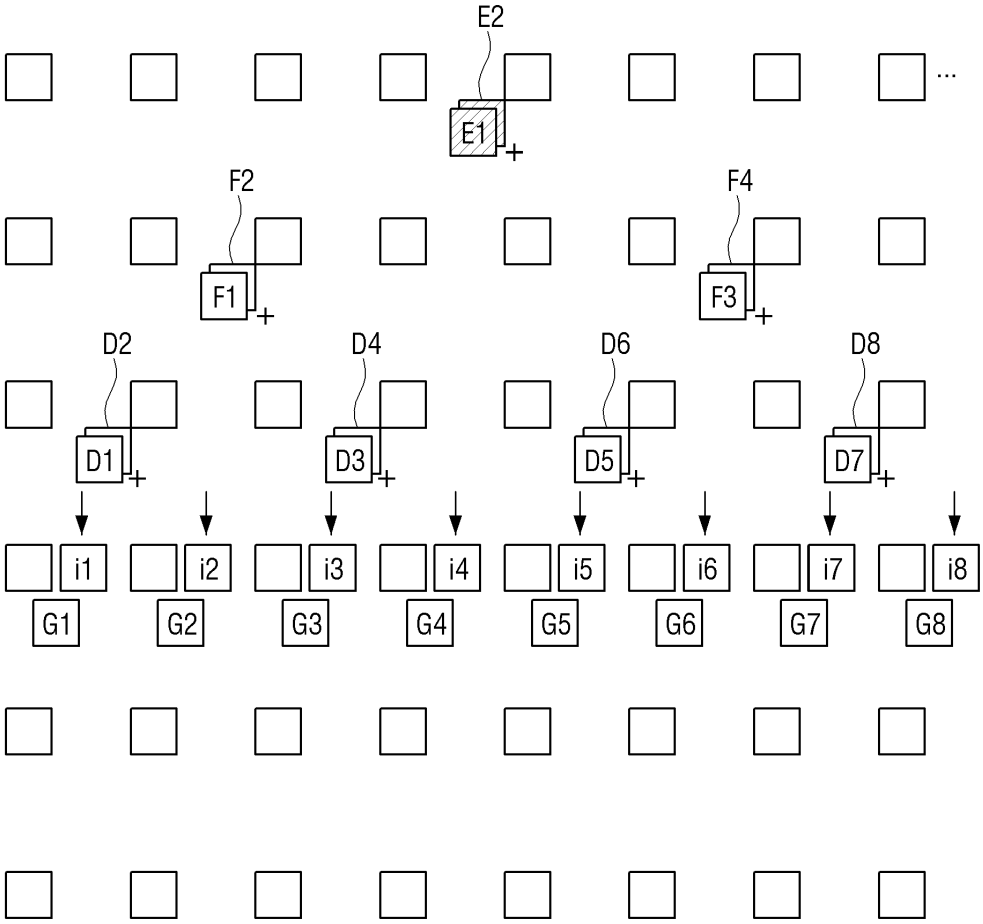
도면13d


C2
C4
C1
C3
B2
B4
B6
B8
B1
B3
B5
B7
i1
i2
i3
i4
i5
i6
i7
i8
D1
D2
D3
D4
D5
D6
D7
D8

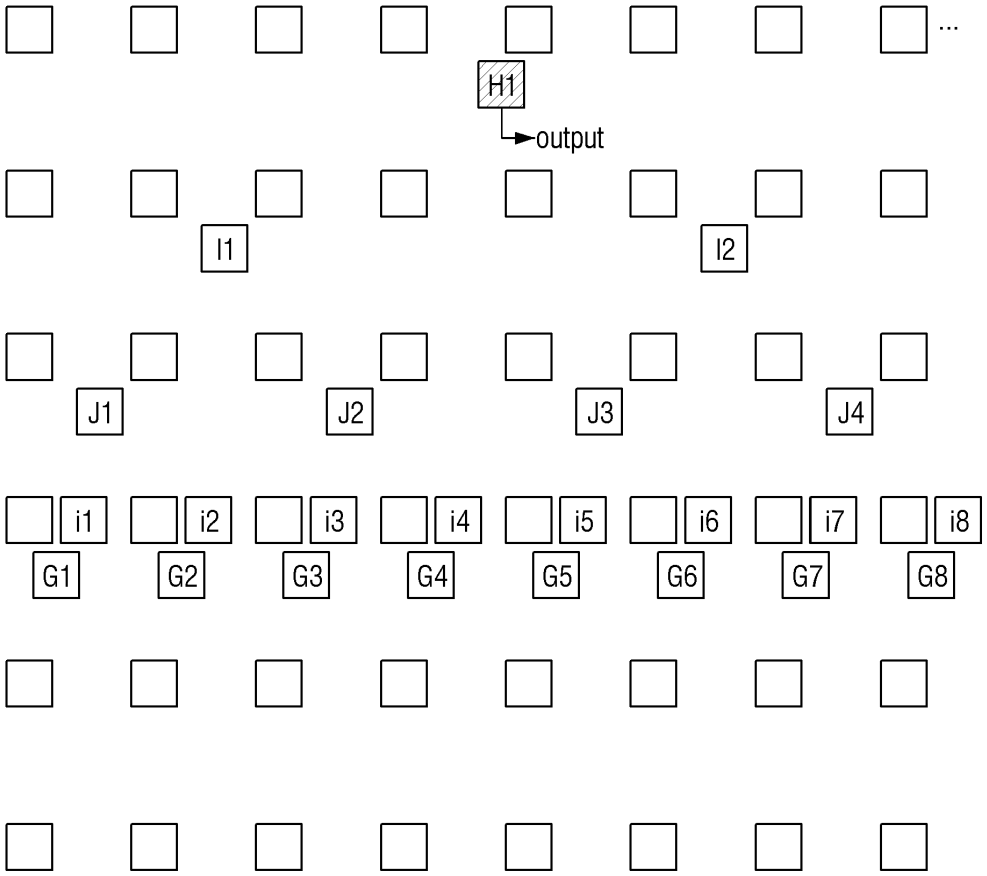
도면13e



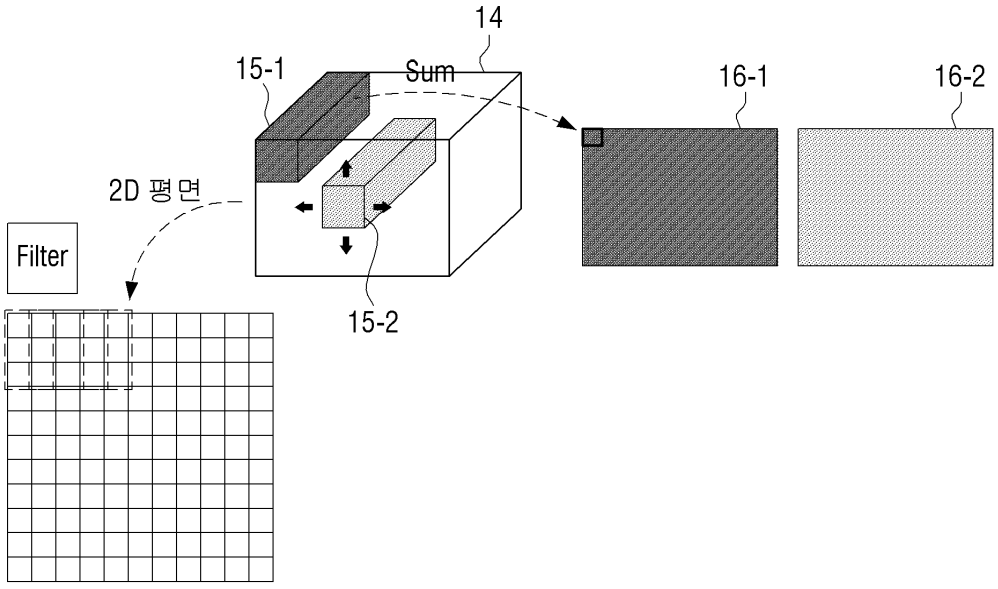
도면13f


E2
E1
F2
F1
F4
F3
D2
D1
D4
D3
D6
D5
D8
D7
i1
i2
i3
i4
i5
i6
i7
i8
G1
G2
G3
G4
G5
G6
G7
G8

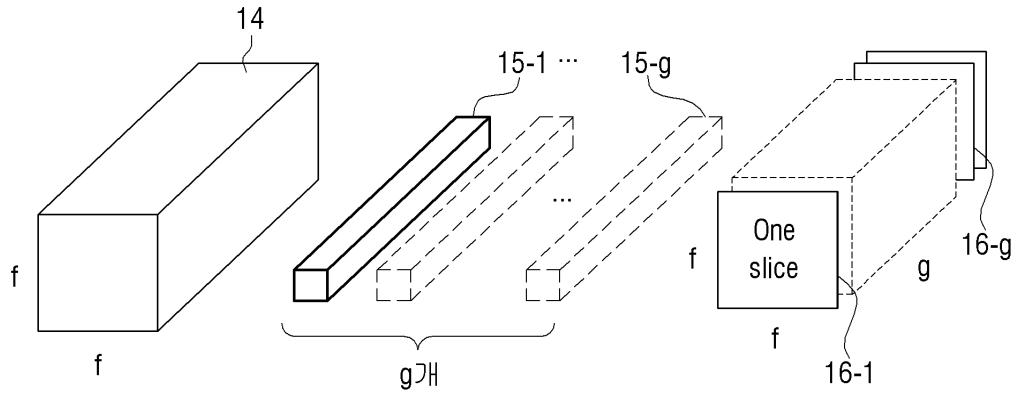
도면13g


H1
output
I1
I2
J1
J2
J3
J4
i1
i2
i3
i4
i5
i6
i7
i8
G1
G2
G3
G4
G5
G6
G7
G8

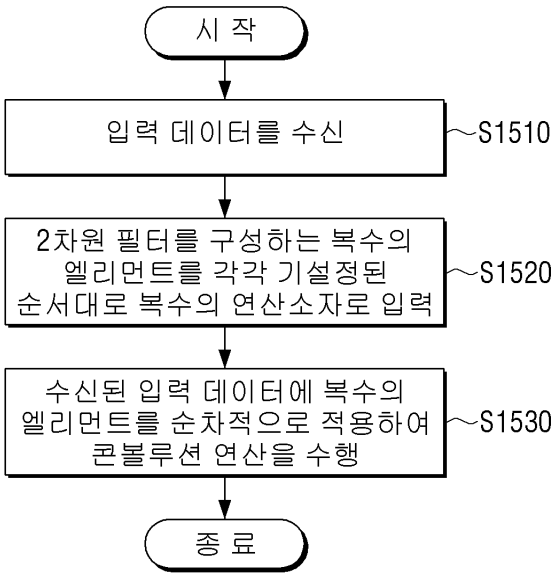
도면14a


14
15-1
Sum
16-1
16-2
2D 평면
Filter
15-2

도면14b


14
15-1 ···
15-g
f
f
g개
One
slice
f
g
16-g
f
16-1

도면15


시 작
입력 데이터를 수신
S1510
2차원 필터를 구성하는 복수의
엘리먼트를 각각 기설정된
순서대로 복수의 연산소자로 입력
S1520
수신된 입력 데이터에 복수의
엘리먼트를 순차적으로 적용하여
콘볼루션 연산을 수행
S1530
종 료