(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0090147
(43) 공개일자 2013년08월13일

(51) 국제특허분류(Int. Cl.)
G06N 3/02 (2006.01)
(21) 출원번호 10-2012-0011256
(22) 출원일자 2012년02월03일
심사청구일자 없음

(71) 출원인
안병익
서울특별시 송파구 올림픽로 212, A동 1705호 (잠실동, 갤러리아팰리스)
(72) 발명자
안병익
서울특별시 송파구 올림픽로 212, A동 1705호 (잠실동, 갤러리아팰리스)
(74) 대리인
특허법인 신성

전체 청구항 수 : 총 68 항

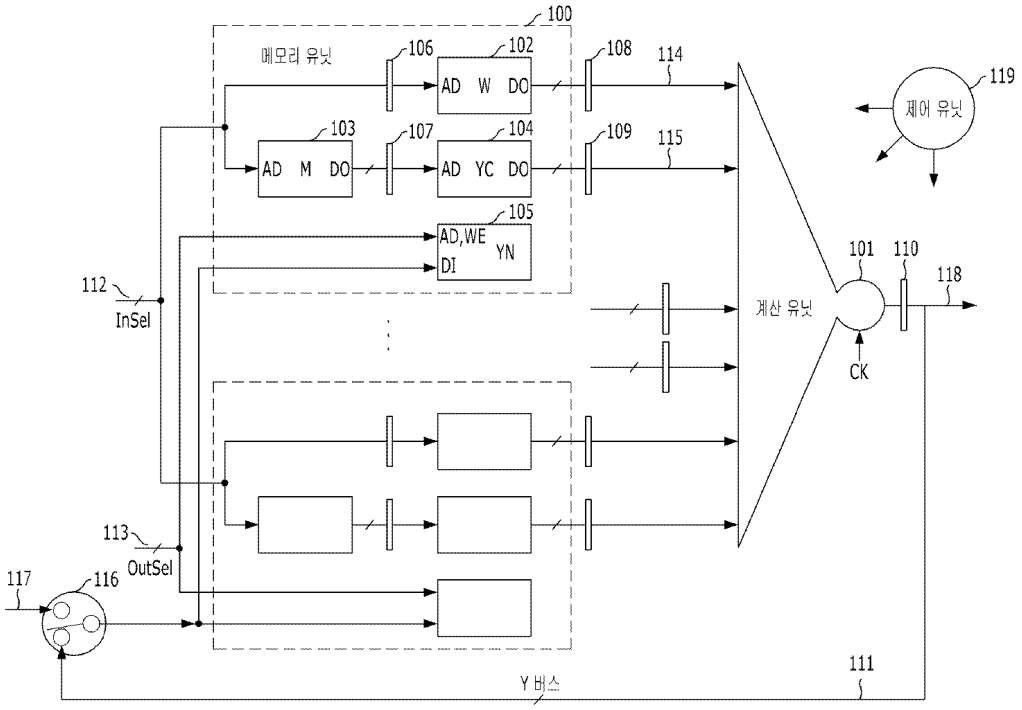
(54) 발명의 명칭 **신경망 컴퓨팅 장치 및 시스템과 그 방법**

(57) 요 약

본 발명은 신경망 컴퓨팅 장치 및 시스템과 그 방법에 관한 것으로, 전체 구성 요소가 하나의 시스템 클록에 동기화되는 동기화 회로로 동작하고, 인공 신경망 데이터를 저장하는 분산형 메모리 구조와 모든 뉴런을 파이프라인 회로에서 시분할로 처리하는 계산 구조를 포함하는, 신경망 컴퓨팅 장치 및 시스템과 그 방법을 제공하고자 한다.

이를 위하여, 본 발명은, 신경망 컴퓨팅 장치에 있어서, 상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛; 각각 연결선 속성값과 뉴런 속성값을 출력하기 위한 복수 개의 메모리 유닛; 및 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키기 위한 하나의 계산 유닛을 포함한다.

대 표 도



특허청구의 범위

청구항 1

신경망 컴퓨팅 장치에 있어서,

상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛;

각각 연결선 속성값과 뉴런 속성값을 출력하기 위한 복수 개의 메모리 유닛; 및

상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키기 위한 하나의 계산 유닛

을 포함하는 신경망 컴퓨팅 장치.

청구항 2

제 1 항에 있어서,

상기 제어 유닛은,

신경망 갱신 주기 내의 클록 주기를 제공하기 위한 클록 주기 카운터; 및

제어 신호의 타이밍 및 제어 정보를 저장하고 있다가 상기 클록 주기 카운터로부터의 클록 주기에 따라 상기 신경망 컴퓨팅 장치로 출력하기 위한 제어 메모리

를 포함하는 신경망 컴퓨팅 장치.

청구항 3

제 1 항에 있어서,

상기 제어 유닛은 호스트 컴퓨터에 의해 제어되는, 신경망 컴퓨팅 장치.

청구항 4

제 1 항에 있어서,

상기 계산 유닛의 출력과 상기 복수 개의 메모리 유닛 사이에 구비되어, 상기 제어 유닛의 제어에 따라 상기 제어 유닛으로부터의 입력 데이터와 상기 계산 유닛으로부터의 새로운 뉴런 속성값 중 어느 하나를 선택하여 상기 복수 개의 메모리 유닛으로 스위칭하기 위한 스위칭 수단

을 더 포함하는 신경망 컴퓨팅 장치.

청구항 5

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,

상기 복수 개의 메모리 유닛 각각은,

연결선 속성값을 저장하기 위한 제1메모리;

뉴런의 고유번호를 저장하기 위한 제2메모리;

상기 제2메모리의 데이터 출력이 주소 입력으로 연결되며, 뉴런 속성값을 저장하기 위한 제3메모리; 및

상기 계산 유닛에서 계산된 새로운 뉴런 속성값을 저장하기 위한 제4메모리

를 포함하는 신경망 컴퓨팅 장치.

청구항 6

제 5 항에 있어서,

상기 복수 개의 메모리 유닛 각각은,

시스템 클록에 동기화되어 동작하며, 상기 제1메모리의 주소 입력단에 구비되어 상기 제1메모리로 입력되는 연결선 묶음 번호를 임시 저장하기 위한 제1레지스터; 및

상기 시스템 클록에 동기화되어 동작하며, 상기 제3메모리의 주소 입력단에 구비되어 상기 제2메모리에서 출력되는 뉴런의 고유번호를 임시 저장하기 위한 제2레지스터를 더 포함하고,

상기 제1메모리, 상기 제2메모리, 상기 제3메모리는 상기 제어 유닛의 제어에 따라 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 장치.

청구항 7

제 5 항에 있어서,

시스템 클록에 동기화되어 동작하며, 상기 복수 개의 메모리 유닛의 각 출력과 상기 하나의 계산 유닛의 입력 사이에 구비되어 상기 연결선 속성값과 상기 뉴런 속성값을 임시 저장하기 위한 복수의 제3레지스터; 및

상기 시스템 클록에 동기화되어 동작하며, 상기 하나의 계산 유닛의 출력단에 구비되어 상기 하나의 계산 유닛에서 출력되는 새로운 뉴런 속성값을 임시 저장하기 위한 제4레지스터를 더 포함하고,

상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛은, 상기 제어 유닛의 제어에 따라 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 장치.

청구항 8

제 5 항에 있어서,

상기 제어 유닛은,

하기의 a 과정 내지 h 과정에 따라 각각의 상기 메모리 유닛 내의 각 메모리에 데이터를 저장하는, 신경망 컴퓨팅 장치.

a. 신경망 내에서 가장 많은 수의 입력 연결선을 가진 뉴런의 입력 연결선의 수(Pmax)를 찾는 과정

b. 상기 메모리 유닛의 수를 p라 할 때, 신경망 내의 모든 뉴런이 $\lceil Pmax/p \rceil * p$ 개의 연결선을 갖도록 각각의 뉴런에 어떤 뉴런이 연결되어도 인접 뉴런에 영향을 미치지 않는 연결선 속성값을 갖는 가상의 연결선을 추가하는 과정

c. 신경망 내 모든 뉴런을 임의의 순서로 정렬하고 일련번호를 부여하는 과정

d. 모든 뉴런 각각의 연결선을 p개씩 나누어 $\lceil Pmax/p \rceil$ 개의 묶음으로 분류하고 묶음들을 임의의 순서로 정렬하는 과정

e. 첫 번째 뉴런의 첫 번째 연결선 묶음부터 마지막 번째 뉴런의 마지막 연결선 묶음까지 순서대로 일련 번호 k를 부여하는 과정

f. 상기 메모리 유닛 중 i번째 메모리 유닛의 제1메모리의 k번째 주소에는 k번째 연결선 묶음의 i번째 연결선의 속성값을 저장하는 과정

g. 상기 복수 개의 메모리 유닛의 상기 제3메모리에는 j번째 주소에 j번째 뉴런의 속성값을 저장하는 과정

h. 상기 메모리 유닛 중 i번째 메모리 유닛의 제2메모리의 k번째 주소에는 k번째 연결선 묶음의 i번째 연결선에

연결된 뉴런의 번호 값을 저장하는 과정

청구항 9

제 8 항에 있어서,

상기 b 과정은,

어떤 뉴런과 연결되어도 뉴런의 속성값에 영향을 주지 않는 연결선의 속성값을 갖도록 하는 방식 또는 신경망에 어떤 뉴런과 연결되어도 영향을 주지 않는 속성값을 가진 하나의 가상의 뉴런을 추가하고 모든 가상의 연결선들이 상기 가상의 뉴런과 연결되도록 하는 방식 중 어느 한 방식으로 상기 가상의 연결선을 추가하는, 신경망 컴퓨팅 장치.

청구항 10

제 5 항에 있어서,

상기 제어 유닛은,

하기의 a 과정 내지 h 과정에 따라 각각의 상기 메모리 유닛 내의 각 메모리에 데이터를 저장하는, 신경망 컴퓨팅 장치.

a. 신경망 내 모든 뉴런을 각 뉴런에 포함된 입력 연결선의 수를 기준으로 오름차순으로 정렬하고 순서대로 번호를 부여하는 과정

b. 신경망 내에 다른 뉴런과 연결선으로 연결되어도 영향을 미치지 않는 속성값을 갖는 한 개의 널(null) 뉴런을 추가하는 과정

c. 뉴런 j의 입력 연결선의 수를 pj라 할 때, 신경망 내의 뉴런 각각이 $\lceil pj/p \rceil$ *p개의 연결선을 갖도록 뉴런에 어떤 뉴런과 연결되어도 영향을 미치지 않는 연결선 속성값을 갖고 널(null) 뉴런과 연결된 $\lceil pj/p \rceil$*p - pj개의 연결선을 추가하는 과정(p는 상기 메모리 유닛의 수)

d. 모든 뉴런 각각의 연결선을 p개씩 나누어 $\lceil pj/p \rceil$ 개의 묶음으로 분류하고 묶음 내의 연결선 각각에 임의의 순서로 1부터 시작하여 1씩 증가하는 번호 i를 부여하는 과정

e. 첫 번째 뉴런의 첫 번째 연결선 묶음부터 마지막 번째 뉴런의 마지막 연결선 묶음까지 순서대로 1부터 시작하여 1씩 증가하는 번호 k를 부여하는 과정

f. 상기 메모리 유닛 중 i번째 메모리 유닛의 제1메모리의 k번째 주소에는 k번째 연결선 묶음의 i번째 연결선의 속성값을 저장하는 과정

g. 상기 메모리 유닛 중 i번째 메모리 유닛의 제2메모리의 k번째 주소에는 k번째 연결선 묶음의 i번째 연결선에 연결된 뉴런의 번호를 저장하는 과정

h. 상기 메모리 유닛 중 i번째 메모리 유닛의 제3메모리의 j번째 주소에는 j번째 뉴런의 속성값을 저장하는 과정

청구항 11

제 5 항에 있어서,

상기 제어 유닛으로부터의 제어 신호에 의해 제어되는 복수 개의 디지털 스위치를 이용하여 두 개의 동일한 메모리의 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로를 상기 제3메모리와 상기 제4메모리에 적용하는, 신경망 컴퓨팅 장치.

청구항 12

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,

상기 복수 개의 메모리 유닛 각각은,

연결선 속성값을 저장하기 위한 제1메모리;

뉴런의 고유번호를 저장하기 위한 제2메모리; 및

뉴런 속성값을 저장하기 위한 제3메모리

를 포함하는 신경망 컴퓨팅 장치.

청구항 13

제 12 항에 있어서,

기존 뉴런 속성값과 상기 계산 유닛에서 계산된 새로운 뉴런 속성값을 구분없이 상기 제3메모리에 저장하고,

상기 기존 뉴런 속성값의 읽기 과정과 상기 계산 유닛에서 계산된 새로운 뉴런 속성값의 쓰기 과정을 하나의 파이프라인 주기에 시간 분할로 처리하기 위한 단일 메모리 중복 저장 회로를 상기 제3메모리에 적용하는, 신경망 컴퓨팅 장치.

청구항 14

제 12 항에 있어서,

상기 제3메모리의 제1 반부 영역에 기존 뉴런 속성값을 저장하고, 제2 반부 영역에 상기 계산 유닛에서 계산된 새로운 뉴런 속성값을 저장하고,

상기 기존 뉴런 속성값의 읽기 과정과 상기 계산 유닛에서 계산된 새로운 뉴런 속성값의 쓰기 과정을 하나의 파이프라인 주기에 시간 분할로 처리하기 위한 단일 메모리 교체 회로를 상기 제3메모리에 적용하는, 신경망 컴퓨팅 장치.

청구항 15

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,

상기 계산 유닛 내부의 각 계산 단계 사이에 시스템 클록에 의해 동기화되는 레지스터를 더 구비하여 상기 각 계산 단계를 파이프라인 방식으로 처리하는, 신경망 컴퓨팅 장치.

청구항 16

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,

상기 계산 유닛에 구비된 전체 또는 일부의 계산 장치 각각에 대해 내부 구조를 시스템 클록에 동기화되어 동작하는 파이프라인 회로로 구현한, 신경망 컴퓨팅 장치.

청구항 17

제 16 항에 있어서,

특정 계산 장치의 입력의 수의 개수에 해당하는 분배기와 복수 개의 특정 계산 장치와 상기 특정 계산 장치의 출력의 수에 해당하는 개수의 다중화기를 사용하여, 순차적으로 인입되는 입력 데이터를 상기 분배기를 통해 상

기 복수 개의 특정 계산 장치로 분배시키고 상기 복수 개의 특정 계산 장치의 계산 결과를 상기 다중화기로 수합하는 병렬 계산 라인 기법을 적용하여 상기 각 계산 장치의 내부 구조를 파이프라인 방식으로 구현한, 신경망 컴퓨팅 장치.

청구항 18

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,

상기 계산 유닛은,

상기 복수 개의 메모리 유닛으로부터의 연결선 속성값과 뉴런 속성값에 대해 곱셈 연산을 수행하기 위한 곱셈 연산부;

상기 곱셈 연산부로부터의 복수의 출력값에 대해 하나 이상의 단계로 덧셈 연산을 수행하기 위한 트리 구조의 덧셈 연산부;

상기 덧셈 연산부로부터의 출력값을 누적 연산하기 위한 누산기; 및

상기 누산기로부터의 누적 출력값에 활성화 함수를 적용하여 다음 신경망 갱신 주기에 사용될 새로운 뉴런 속성값을 계산하기 위한 활성화 함수 연산기

를 포함하는 신경망 컴퓨팅 장치.

청구항 19

제 18 항에 있어서,

상기 누산기를, 하나의 분배기와 복수 개의 선입선출 큐와 복수 개의 누산기와 하나의 다중화기를 사용하여, 순차적으로 인입되는 입력 데이터를 상기 분배기를 통해 상기 복수 개의 선입선출 큐로 분배시키고 상기 선입선출 큐와 상기 누산기를 거쳐 누산된 결과를 상기 다중화기로 수합하는 병렬 계산 라인 기법을 적용하여 구현한, 신경망 컴퓨팅 장치.

청구항 20

제 18 항에 있어서,

상기 곱셈 연산부에 구비된 곱셈기 각각을, 하나의 뺄셈기와 하나의 제곱승 계산기로 구현하되, 두 개의 입력값이 상기 뺄셈기로 연결되고 상기 뺄셈기의 출력이 상기 제곱승 계산기로 연결되는, 신경망 컴퓨팅 장치.

청구항 21

제 18 항에 있어서,

상기 곱셈 연산부에 구비된 곱셈기 각각을, 하나의 참조 테이블과 하나의 곱셈기를 사용하여 구현한, 신경망 컴퓨팅 장치.

청구항 22

제 18 항에 있어서,

상기 누산기와 상기 활성화 함수 연산기 사이에 선입선출 큐

를 더 포함하는 신경망 컴퓨팅 장치.

청구항 23

제 18 항에 있어서,

상기 활성화 함수 연산기는,

제1입력을 통하여 상기 누산기로부터의 누적 출력값(뉴런의 순입력 데이터)을 입력받고, 제1출력을 통하여 다음 신경망 갱신 주기에 사용될 새로운 뉴런 속성값을 상기 복수 개의 메모리 유닛 각각으로 출력하며,

제2입력을 통하여 해당 뉴런의 번호를 입력받고, 상기 제1출력으로 새로운 뉴런 속성값이 출력될 때 해당 뉴런의 번호를 제2출력을 통하여 상기 복수 개의 메모리 유닛 각각의 입력으로 연결하는, 신경망 컴퓨팅 장치.

청구항 24

신경망 컴퓨팅 장치에 있어서,

상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛;

각각 연결선 속성값과 뉴런 속성값을 출력하기 위한 복수 개의 메모리 유닛;

상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하기 위한 하나의 계산 유닛;

상기 제어 유닛으로부터의 입력 데이터를 입력 뉴런에 제공하기 위한 입력 수단;

상기 입력 수단으로부터의 입력 데이터 또는 상기 계산 유닛으로부터의 새로운 뉴런 속성값을 상기 제어 유닛의 제어에 따라 상기 복수 개의 메모리 유닛으로 스위칭하기 위한 스위칭 수단; 및

상기 제어 유닛의 제어에 따라 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로로 이루어져, 상기 계산 유닛으로부터의 새로운 뉴런 속성값이 상기 제어 유닛으로 출력되도록 하기 위한 제1 및 제2출력 수단

을 포함하는 신경망 컴퓨팅 장치.

청구항 25

제 24 항에 있어서,

상기 제어 유닛으로부터의 입력 데이터를 상기 복수 개의 메모리 유닛에 저장하는 과정을 신경망 갱신 주기의 처음에 실행하는, 신경망 컴퓨팅 장치.

청구항 26

제 24 항에 있어서,

상기 제어 유닛으로부터의 입력 데이터를 상기 복수 개의 메모리 유닛에 저장하는 과정을 상기 계산 유닛의 출력이 발생하지 않는 클록 주기에 끼워 넣기(interleaving) 방식으로 실행하는, 신경망 컴퓨팅 장치.

청구항 27

신경망 컴퓨팅 시스템에 있어서,

상기 신경망 컴퓨팅 시스템을 제어하기 위한 제어 유닛;

"각각 연결선 속성값과 뉴런 속성값을 출력하는 복수의 메모리 파트"를 포함하는 복수 개의 메모리 유닛; 및

상기 복수 개의 메모리 유닛 내의 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키기 위한 복수의 계산 유닛

을 포함하는 신경망 컴퓨팅 시스템.

청구항 28

제 27 항에 있어서,

상기 복수 개의 메모리 유닛 내의 상기 복수의 메모리 파트와 상기 복수의 계산 유닛은,

상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 시스템.

청구항 29

제 27 항 또는 제 28 항에 있어서,

각각의 상기 메모리 파트는,

연결선 속성값을 저장하기 위한 제1메모리;

뉴런의 고유번호를 저장하기 위한 제2메모리;

복수 개의 메모리가 디코더 회로에 의해 복수 배 용량의 통합 메모리의 기능을 수행하여 뉴런 속성값을 저장하기 위한 제1메모리 그룹; 및

복수 개의 메모리가 공통으로 묶여서 상응하는 상기 계산 유닛에서 계산된 새로운 뉴런 속성값을 저장하기 위한 제2메모리 그룹

을 포함하는 신경망 컴퓨팅 시스템.

청구항 30

제 29 항에 있어서,

i번째 메모리 파트(i는 임의의 자연수)의 제1 메모리 그룹의 j번째 메모리(j는 임의의 자연수)와, j번째 메모리 파트의 제2 메모리 그룹의 i번째 메모리는,

상기 제어 유닛의 제어에 따라 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 방식으로 구현된, 신경망 컴퓨팅 시스템.

청구항 31

제 29 항에 있어서,

상기 제어 유닛은,

하기의 a 과정 내지 j 과정에 따라 각각의 상기 메모리 파트 내의 각 메모리에 데이터를 저장하는, 신경망 컴퓨팅 시스템.

a. 신경망 내 모든 뉴런을 H개의 균일한 뉴런 그룹으로 나누는 과정

b. 각 뉴런 그룹 내에서 가장 많은 수의 입력 연결선을 가진 뉴런의 입력 연결선의 수(Pmax)를 찾는 과정

c. 메모리 유닛의 수를 p라 할 때, 신경망 내의 모든 뉴런이 $\lceil Pmax/p \rceil * p$ 개의 연결선을 갖도록 각각의 뉴런에 어떤 뉴런과 연결되어도 인접 뉴런에 영향을 미치지 않는 연결선 속성값을 갖는 가상의 연결선을 추가하는 과정

d. 뉴런 그룹 각각에 대해, 뉴런 그룹 내 모든 뉴런 각각에 임의의 순서로 번호를 부여하는 과정

e. 뉴런 그룹 각각에 대해, 뉴런 그룹 내 모든 뉴런 각각의 연결선을 p개씩 나누어 $\lceil Pmax/p \rceil$개의 묶음으로 분류하고 묶음 내의 연결선 각각에 임의의 순서로 1부터 시작하여 1씩 증가하는 번호 i를 부여하는 과정

f. 뉴런 그룹 각각에 대해, 뉴런 그룹 내 첫 번째 뉴런의 첫 번째 연결선 묶음부터 마지막 번째 뉴런의 마지막 연결선 묶음까지 순서대로 1부터 시작하여 1씩 증가하는 번호 k를 부여하는 과정

g. 상기 메모리 유닛 중 i번째 메모리 유닛의 h번째 메모리 파트의 제1메모리의 j번째 주소에는 h번째 뉴런 그룹의 k번째 연결선 묶음의 i번째 연결선의 속성값을 저장하는 과정

h. 상기 메모리 유닛 중 i번째 메모리 유닛의 h번째 제2메모리의 j번째 주소에는 h번째 뉴런 그룹의 k번째 연결선 묶음의 i번째 연결선에 연결된 뉴런의 고유 번호를 저장하는 과정

i. 모든 상기 메모리 유닛 각각의 모든 상기 메모리 파트의 제1메모리 그룹을 구성하는 g번째 메모리의 j번째 주소에는 g번째 뉴런 그룹 내에서 j를 고유번호로 하는 뉴런의 속성값을 저장하는 과정

j. 모든 상기 메모리 유닛 각각의 h번째 메모리 파트의 제2메모리 그룹의 모든 메모리들의 j번째 주소에는 공통으로 h번째 뉴런 그룹 내에서 j를 고유번호로 하는 뉴런의 속성값을 저장하는 과정

청구항 32

제 27 항 또는 제 28 항에 있어서,

각각의 상기 계산 유닛은.

상기 상응하는 복수의 메모리 파트로부터의 연결선 속성값과 뉴런 속성값에 대해 곱셈 연산을 수행하기 위한 곱셈 연산부;

상기 곱셈 연산부로부터의 복수의 출력값에 대해 하나 이상의 단계로 덧셈 연산을 수행하기 위한 트리 구조의 덧셈 연산부;

상기 덧셈 연산부로부터의 출력값을 누적 연산하기 위한 누산기; 및

상기 누산기로부터의 누적 출력값에 활성화 함수를 적용하여 새로운 뉴런 속성값을 계산하기 위한 활성화 함수 연산기

를 포함하는 신경망 컴퓨팅 시스템.

청구항 33

신경망 컴퓨팅 장치에 있어서,

상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛;

각각 연결선 속성값과 뉴런 오차값을 출력하기 위한 복수 개의 메모리 유닛; 및

상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 오차값을 이용하여 새로운 뉴런 오차값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키기 위한 하나의 계산 유닛

을 포함하는 신경망 컴퓨팅 장치.

청구항 34

제 33 항에 있어서,

상기 계산 유닛은,

상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 오차값, 및 상기 제어 유닛을 통해 제공되는 학습 데이터를 이용하여 새로운 뉴런 오차값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백

시키는, 신경망 컴퓨팅 장치.

청구항 35

제 33 항 또는 제 34 항에 있어서,

상기 복수 개의 메모리 유닛 각각은,

연결선 속성값을 저장하기 위한 상기 제1메모리;

뉴런의 고유번호를 저장하기 위한 제2메모리;

뉴런 오차값을 저장하기 위한 제3메모리; 및

상기 계산 유닛에서 계산된 새로운 뉴런 오차값을 저장하기 위한 제4메모리

를 포함하는 신경망 컴퓨팅 장치.

청구항 36

신경망 컴퓨팅 장치에 있어서,

상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛;

각각 연결선 속성값과 뉴런 속성값을 출력하고, 연결선 속성값과 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하기 위한 복수 개의 메모리 유닛; 및

상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값과 학습 속성값을 계산하기 위한 하나의 계산 유닛

을 포함하는 신경망 컴퓨팅 장치.

청구항 37

제 36 항에 있어서,

상기 복수 개의 메모리 유닛 각각은,

연결선 속성값을 저장하기 위한 제1메모리;

뉴런의 고유번호를 저장하기 위한 제2메모리;

뉴런 속성값을 저장하기 위한 제3메모리;

상기 계산 유닛에서 계산된 새로운 뉴런 속성값을 저장하기 위한 제4메모리;

상기 제1메모리로부터의 연결선 속성값을 지연시키기 위한 제1지연수단;

상기 제3메모리로부터의 뉴런 속성값을 지연시키기 위한 제2지연수단; 및

상기 계산 유닛으로부터의 학습 속성값과 상기 제1지연수단으로부터의 연결선 속성값과 상기 제2지연수단으로부터의 뉴런 속성값을 이용하여 새로운 연결선 속성값을 계산하기 위한 연결선 조정 모듈; 및

상기 연결선 조정 모듈에서 계산된 새로운 연결선 속성값을 저장하기 위한 제5메모리

를 포함하는 신경망 컴퓨팅 장치.

청구항 38

제 37 항에 있어서,

상기 제어 유닛의 제어에 따라 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로를 상기 제1메

모리와 상기 제5메모리에 적용하고 또한 상기 제3메모리와 상기 제4메모리에 적용하는, 신경망 컴퓨팅 장치.

청구항 39

제 37 항에 있어서,

상기 제1메모리와 상기 제5메모리, 상기 제3메모리와 상기 제4메모리를 각각 하나의 메모리로 구현하고, 읽기 과정과 쓰기 과정을 시간 분할로 처리하는, 신경망 컴퓨팅 장치.

청구항 40

제 37 항에 있어서,

상기 연결선 조정 모듈은,

상기 제1지연수단으로부터의 연결선 속성값을 지연시키기 위한 제3지연수단;

상기 계산 유닛으로부터의 학습 속성값과 상기 제2지연수단으로부터의 뉴런 속성값에 대하여 곱셈 연산을 수행하기 위한 곱셈기; 및

상기 제3지연수단으로부터의 연결선 속성값과 상기 곱셈기의 출력 값에 대하여 덧셈 연산을 수행하여 새로운 연결선 속성값을 출력하기 위한 덧셈기

를 포함하는 신경망 컴퓨팅 장치.

청구항 41

신경망 컴퓨팅 장치에 있어서,

상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛;

뉴런의 학습 속성값을 저장하기 위한 제1학습 속성값 메모리;

각각 연결선 속성값과 뉴런 속성값을 출력하고, 연결선 속성값과 뉴런 속성값과 상기 제1학습 속성값 메모리의 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하기 위한 복수 개의 메모리 유닛;

상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값과 학습 속성값을 계산하기 위한 하나의 계산 유닛; 및

상기 하나의 계산 유닛에서 계산된 새로운 학습 속성값을 저장하기 위한 제2학습 속성값 메모리

를 포함하는 신경망 컴퓨팅 장치.

청구항 42

제 41 항에 있어서,

상기 복수 개의 메모리 유닛 각각은,

연결선 속성값을 저장하기 위한 제1메모리;

뉴런의 고유번호를 저장하기 위한 제2메모리;

뉴런 속성값을 저장하기 위한 제3메모리;

상기 계산 유닛에서 계산된 새로운 뉴런 속성값을 저장하기 위한 제4메모리; 및

연결선 속성값과 뉴런 속성값과 상기 제1학습 속성값 메모리의 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하기 위한 연결선 조정 모듈; 및

상기 연결선 조정 모듈에서 계산된 새로운 연결선 속성값을 저장하기 위한 제5메모리

를 포함하는 신경망 컴퓨팅 장치.

청구항 43

제 42 항에 있어서,

상기 제어 유닛의 제어에 따라 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로를 상기 제1학습 속성값 메모리와 상기 제2학습 속성값 메모리, 상기 제1메모리와 상기 제5메모리, 및 상기 제3메모리와 상기 제4메모리에 각각 적용하는, 신경망 컴퓨팅 장치.

청구항 44

제 42 항에 있어서,

상기 제1학습 속성값 메모리와 상기 제2학습 속성값 메모리, 상기 제1메모리와 상기 제5메모리, 상기 제3메모리와 상기 제4메모리를 각각 하나의 메모리로 구현하고, 읽기 과정과 쓰기 과정을 시간 분할로 처리하는, 신경망 컴퓨팅 장치.

청구항 45

제 42 항에 있어서,

상기 연결선 조정 모듈은,

상기 메모리 유닛으로부터의 연결선 속성값을 지연시키기 위한 제1지연수단;

상기 제1학습 속성값 메모리로부터의 학습 속성값과 상기 메모리 유닛으로부터의 뉴런 속성값에 대하여 곱셈 연산을 수행하기 위한 곱셈기; 및

상기 제1지연수단으로부터의 연결선 속성값과 상기 곱셈기의 출력 값에 대하여 덧셈 연산을 수행하여 새로운 연결선 속성값을 출력하기 위한 덧셈기

를 포함하는 신경망 컴퓨팅 장치.

청구항 46

신경망 컴퓨팅 장치에 있어서,

상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛;

각각 연결선 속성값, 순방향 뉴런 속성값 및 역방향 뉴런 속성값을 저장하고 출력하며, 새로운 연결선 속성값을 계산하기 위한 복수 개의 메모리 유닛; 및

상기 복수 개의 메모리 유닛으로부터 각각 입력되는 데이터를 바탕으로 새로운 순방향 뉴런 속성값과 역방향 뉴런 속성값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키기 위한 하나의 계산 유닛

을 포함하는 신경망 컴퓨팅 장치.

청구항 47

제 46 항에 있어서,

상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛은,

상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 장치.

청구항 48

제 46 항 또는 제 47 항에 있어서,

상기 복수 개의 메모리 유닛 각각은,

제2메모리의 주소값을 저장하기 위한 제1메모리;

연결선 속성값을 저장하기 위한 상기 제2메모리;

뉴런의 고유번호를 저장하기 위한 제3메모리;

역방향 뉴런 속성값을 저장하기 위한 제4메모리;

상기 계산 유닛에서 계산된 새로운 역방향 뉴런 속성값을 저장하기 위한 제5메모리;

뉴런의 고유번호를 저장하기 위한 제6메모리;

순방향 뉴런 속성값을 저장하기 위한 제7메모리;

상기 계산 유닛에서 계산된 새로운 순방향 뉴런 속성값을 저장하기 위한 제8메모리;

상기 제2메모리의 입력을 선택하기 위한 제1스위치;

상기 제4메모리 또는 상기 제7메모리의 출력을 상기 계산 유닛으로 스위칭하기 위한 제2스위치;

상기 계산 유닛의 출력을 상기 제5메모리 또는 상기 제8메모리로 스위칭하기 위한 제3스위치; 및

아웃셀(OutSel) 입력을 상기 제5메모리 또는 상기 제8메모리로 스위칭하기 위한 제4스위치

를 포함하는 신경망 컴퓨팅 장치.

청구항 49

제 48 항에 있어서,

상기 제어 유닛의 제어에 따라 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로를 상기 제4메모리와 상기 제5메모리에 적용하고 또한 상기 제7메모리와 상기 제8메모리에 적용하는, 신경망 컴퓨팅 장치.

청구항 50

제 48 항에 있어서,

상기 제4메모리와 상기 제5메모리, 상기 제7메모리와 상기 제8메모리를 각각 하나의 메모리로 구현하고, 읽기 과정과 쓰기 과정을 시간 분할로 처리하는, 신경망 컴퓨팅 장치.

청구항 51

제 48 항에 있어서,

상기 제어 유닛은,

하기의 a 과정 내지 q 과정에 따라 각각의 상기 메모리 유닛 내의 각 메모리에 데이터를 저장하는, 신경망 컴퓨팅 장치.

a. 인공 신경망 순방향 네트워크에서 모든 연결선 각각의 양쪽 끝을 화살표가 시작되는 한쪽 끝과 화살표가 끝

나는 다른 한쪽 끝으로 구분할 때, 모든 연결선 양 쪽에 하기의 1 내지 4의 조건을 만족하는 번호를 부여하는 과정

1. 모든 뉴런 각각에서 다른 뉴런으로 나가는 아웃바운드(outbound) 연결선들의 번호는 중복되지 않고 고유한 번호를 갖는 조건

2. 모든 뉴런 각각에서 다른 뉴런으로부터 들어오는 인바운드(inbound) 연결선들의 번호는 중복되지 않고 고유한 번호를 갖는 조건

3. 모든 연결선 양쪽의 번호는 같은 번호를 갖는 조건

4. 상기 1 내지 3의 조건을 만족하되 가능한 한 낮은 숫자의 번호를 갖는 조건

b. 모든 뉴런의 아웃바운드(outbound) 또는 인바운드(inbound) 연결선에 부여된 번호 중 가장 큰 수(Pmax)를 찾는 과정

c. 신경망의 순방향 네트워크 내부에 다른 뉴런과 연결선으로 연결되어도 영향을 미치지 않는 속성값을 갖는 한 개의 널(null) 뉴런을 추가하는 과정

d. 순방향 네트워크 내의 모든 뉴런 각각의 연결선에 할당된 번호를 유지한 채로 1부터 $\lceil Pmax/p \rceil * p$ 번까지 중 비어 있는 모든 번호에 새로운 연결선을 추가하여 총 $\lceil Pmax/p \rceil * p$ 개의 입력 연결선을 갖도록 확장하고, 추가된 연결선 각각은 어떤 뉴런과 연결되어도 영향을 미치지 않는 연결선 속성값을 갖거나 널(null) 뉴런과 연결되도록 설정하는 과정(p는 상기 신경망 컴퓨팅 장치 내 상기 메모리 유닛의 수)

e. 순방향 네트워크 내 모든 뉴런 각각에 임의의 순서로 번호를 부여하는 과정

f. 순방향 네트워크 내 모든 뉴런 각각의 연결선을 1번부터 순서대로 p개씩 나누어 $\lceil Pmax/p \rceil$개의 순방향 연결선 묶음으로 분류하고 묶음 내의 연결선 각각에 순서대로 1부터 시작하여 1씩 증가하는 새로운 번호 i를 부여하는 과정

g. 첫 번째 뉴런의 첫 번째 순방향 연결선 묶음부터 마지막 번째 뉴런의 마지막 순방향 연결선 묶음까지 순서대로 1부터 시작하여 1씩 증가하는 번호 k를 부여하는 과정

h. 상기 메모리 유닛 중 i번째 메모리 유닛의 제2메모리 및 제9메모리의 k번째 주소에는 k번째 순방향 연결선 묶음의 i번째 연결선의 속성값의 초기값을 저장하는 과정

i. 상기 메모리 유닛 중 i번째 메모리 유닛의 제6메모리의 k번째 주소에는 k번째 순방향 연결선 묶음의 i번째 연결선에 연결된 뉴런의 고유 번호를 저장하는 과정

j. 모든 상기 메모리 유닛 각각의 제7메모리와 제8메모리 각각의 j번째 주소에는 j를 고유번호로 하는 뉴런의 순방향 뉴런 속성값을 저장하는 과정

k. 신경망의 역방향 네트워크 내부에 다른 뉴런과 연결선으로 연결되어도 영향을 미치지 않는 속성값을 갖는 한 개의 널(null) 뉴런을 추가하는 과정

l. 역방향 네트워크 내의 모든 뉴런 각각의 연결선에 할당된 번호를 유지한 채로 1부터 $\lceil Pmax/p \rceil * p$ 번까지 중 비어 있는 모든 번호에 새로운 연결선을 추가하여 총 $\lceil Pmax/p \rceil * p$ 개의 입력 연결선을 갖도록 확장하고, 추가된 연결선 각각은 어떤 뉴런과 연결되어도 영향을 미치지 않는 연결선 속성값을 갖거나 널(null) 뉴런과 연결되도록 설정하는 과정

m. 역방향 네트워크 내 모든 뉴런 각각의 연결선을 1번부터 순서대로 p개씩 나누어 $\lceil Pmax/p \rceil$ 개의 역방향 연결선 묶음으로 분류하고 묶음 내의 연결선 각각에 순서대로 1부터 시작하여 1씩 증가하는 새로운 번호 i를 부여하는 과정

n. 첫 번째 뉴런의 첫 번째 역방향 연결선 묶음부터 마지막 번째 뉴런의 마지막 역방향 연결선 묶음까지 순서대로 1부터 시작하여 1씩 증가하는 번호 k를 부여하는 과정

o. 상기 메모리 유닛 중 i번째 메모리 유닛의 제1메모리의 k번째 주소에는 k번째 역방향 연결선 묶음의 i번째

연결선이 상기 메모리 유닛 중 i번째 메모리 유닛의 제2메모리에서 위치하는 위치 값을 저장하는 과정

p. 상기 메모리 유닛 중 i번째 메모리 유닛의 제3메모리의 k번째 주소에는 k번째 역방향 연결선 묶음의 i번째 연결선에 연결된 뉴런의 고유 번호를 저장하는 과정

q. 모든 상기 메모리 유닛 각각의 제4메모리와 제5메모리 각각의 j번째 주소에는 j를 고유번호로 하는 뉴런의 역방향 뉴런 속성값을 저장하는 과정

청구항 52

제 51 항에 있어서,

상기 a 과정의 조건을 만족하는 해를 호의 색칠 알고리즘(edge coloring algorithm)을 사용하여 구하는, 신경망 컴퓨팅 장치.

청구항 53

제 46 항 또는 제 47 항에 있어서,

상기 복수 개의 메모리 유닛 각각은,

제2메모리의 주소값을 저장하기 위한 제1메모리;

연결선 속성값을 저장하기 위한 상기 제2메모리;

뉴런의 고유번호를 저장하기 위한 제3메모리;

역방향 뉴런 속성값 또는 순방향 뉴런 속성값을 저장하기 위한 제4메모리;

상기 계산 유닛에서 계산된 새로운 역방향 뉴런 속성값 또는 순방향 뉴런 속성값을 저장하기 위한 제5메모리; 및

상기 제2메모리의 입력을 선택하기 위한 스위치

를 포함하는 신경망 컴퓨팅 장치.

청구항 54

제 46 항 또는 제 47 항에 있어서,

상기 계산 유닛은,

상기 복수 개의 메모리 유닛으로부터의 연결선 속성값과 순방향 뉴런 속성값 또는 연결선 속성값과 역방향 뉴런 속성값에 대해 곱셈 연산을 수행하기 위한 곱셈 연산부;

상기 곱셈 연산부로부터의 복수의 출력값에 대해 하나 이상의 단계로 덧셈 연산을 수행하기 위한 트리 구조의 덧셈 연산부;

상기 덧셈 연산부로부터의 출력값을 누적 연산하기 위한 누산기; 및

상기 제어 유닛으로부터의 학습 데이터(Teach)와 상기 누산기로부터의 누적 출력값을 입력받아 새로운 순방향 뉴런 속성값 또는 역방향 뉴런 속성값을 계산하기 위한 소마(soma) 처리기

를 포함하는 신경망 컴퓨팅 장치.

청구항 55

제 54 항에 있어서,

상기 소마 처리기는,

제1입력을 통하여 상기 누산기로부터 뉴런의 순 입력 또는 오차의 총 합을 입력받고, 제2입력을 통하여 출력 뉴런의 학습 데이터를 입력받으며, 제1출력을 통하여 새로 계산된 뉴런의 속성값 또는 오차값을 출력하고, 제2출력을 통하여 연결선 조정을 위한 뉴런의 속성값을 출력하며,

출력 뉴런의 오차를 계산하는 주기에는 입력받은 학습 데이터(Teach)와 내부에 저장된 뉴런의 속성값의 차이로 오차값을 계산하여 내부에 저장하고 상기 제1출력을 통하여 출력하고,

비 출력 뉴런의 오차를 계산하는 주기에는 상기 누산기로부터 오차 입력의 총합을 받아서 내부에 저장하고 상기 제1출력을 통하여 출력하며,

회상 주기에는 상기 누산기로부터 뉴런의 순입력 값을 제공받아 활성화 함수를 적용하여 새로운 뉴런의 속성값을 계산하여 내부에 저장하고 상기 제1출력을 통하여 출력하고, 연결선 조정에 필요한 뉴런의 속성값을 계산하여 상기 제2출력을 통하여 출력하는, 신경망 컴퓨팅 장치.

청구항 56

제 54 항에 있어서,

상기 소마 처리기를 병렬 계산 라인 기법을 적용하여 구현한, 신경망 컴퓨팅 장치.

청구항 57

신경망 컴퓨팅 시스템에 있어서,

상기 신경망 컴퓨팅 시스템을 제어하기 위한 제어 유닛;

"각각 연결선 속성값과 역방향 뉴런 속성값을 출력하거나, 각각 연결선 속성값과 순방향 뉴런 속성값을 출력하고 연결선 속성값과 순방향 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하는 복수의 메모리 파트"를 포함하는 복수 개의 메모리 유닛; 및

상기 복수 개의 메모리 유닛 내의 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 역방향 뉴런 속성값을 이용하여 새로운 역방향 뉴런 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키거나, 상기 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 순방향 뉴런 속성값을 이용하여 새로운 순방향 뉴런 속성값과 학습 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키기 위한 복수의 계산 유닛

을 포함하는 신경망 컴퓨팅 시스템.

청구항 58

제 57 항에 있어서,

상기 복수 개의 메모리 유닛 내의 상기 복수의 메모리 파트와 상기 복수의 계산 유닛은,

상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 시스템.

청구항 59

제 57 항 또는 제 58 항에 있어서,

각각의 상기 메모리 파트는,

제2메모리의 주소값을 저장하기 위한 제1메모리;

연결선 속성값을 저장하기 위한 상기 제2메모리;

뉴런의 고유번호를 저장하기 위한 제3메모리;

역방향 뉴런 속성값을 저장하기 위한 제1메모리 그룹;

상기 계산 유닛에서 계산된 새로운 역방향 뉴런 속성값을 저장하기 위한 제2메모리 그룹;

뉴런의 고유번호를 저장하기 위한 제4메모리;

순방향 뉴런 속성값을 저장하기 제3메모리 그룹;

상기 계산 유닛에서 계산된 새로운 순방향 뉴런 속성값을 저장하기 위한 제4메모리 그룹;

상기 제2메모리의 입력을 선택하기 위한 제1스위치;

상기 제1메모리 그룹 또는 상기 제3메모리 그룹의 출력을 상기 계산 유닛으로 스위칭하기 위한 제2스위치;

상기 계산 유닛의 출력을 상기 제2메모리 그룹 또는 상기 제4메모리 그룹으로 스위칭하기 위한 제3스위치; 및

아웃셀(OutSel) 입력을 상기 제2메모리 그룹 또는 상기 제4메모리 그룹으로 스위칭하기 위한 제4스위치

를 포함하는 신경망 컴퓨팅 시스템.

청구항 60

제 57 항 또는 제 58 항에 있어서,

상기 계산 유닛은,

상기 상응하는 복수의 메모리 파트로부터의 연결선 속성값과 순방향 뉴런 속성값 또는 연결선 속성값과 역방향 뉴런 속성값에 대해 곱셈 연산을 수행하기 위한 곱셈 연산부;

상기 곱셈 연산부로부터의 복수의 출력값에 대해 하나 이상의 단계로 덧셈 연산을 수행하기 위한 트리 구조의 덧셈 연산부;

상기 덧셈 연산부로부터의 출력값을 누적 연산하기 위한 누산기; 및

상기 제어 유닛으로부터의 학습 데이터(Teach)와 상기 누산기로부터의 누적 출력값을 입력받아 새로운 순방향 뉴런 속성값 또는 역방향 뉴런 속성값을 계산하기 위한 소마(soma) 처리기

를 포함하는 신경망 컴퓨팅 시스템.

청구항 61

디지털 시스템의 메모리 장치에 있어서,

외부의 제어 유닛으로부터의 제어 신호에 의해 제어되는 복수 개의 디지털 스위치를 이용하여 두 개의 메모리의 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로를 상기 두 개의 메모리에 적용한, 메모리 장치.

청구항 62

신경망 컴퓨팅 방법에 있어서,

제어 유닛의 제어에 따라, 복수 개의 메모리 유닛이 각각 연결선 속성값과 뉴런 속성값을 출력하는 단계; 및

상기 제어 유닛의 제어에 따라, 하나의 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백 시키는 단계를 포함하되,

상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에

동기화되어 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 방법.

청구항 63

신경망 컴퓨팅 방법에 있어서,

제어 유닛의 제어에 따라, 상기 제어 유닛으로부터 입력 뉴런에 제공하기 위한 데이터를 입력받는 단계;

상기 입력받은 데이터 또는 계산 유닛으로부터의 새로운 뉴런 속성값을 상기 제어 유닛의 제어에 따라 복수 개의 메모리 유닛으로 스위칭하는 단계;

상기 제어 유닛의 제어에 따라, 상기 복수 개의 메모리 유닛이 각각 연결선 속성값과 뉴런 속성값을 출력하는 단계;

상기 제어 유닛의 제어에 따라, 하나의 상기 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하는 단계; 및

상기 제어 유닛의 제어에 따라 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로로 이루어진 제1 및 제2출력 수단이, 상기 계산 유닛으로부터의 새로운 뉴런 속성값이 상기 제어 유닛으로 출력되도록 하는 단계

를 포함하는 신경망 컴퓨팅 방법.

청구항 64

신경망 컴퓨팅 방법에 있어서,

제어 유닛의 제어에 따라, 복수 개의 메모리 유닛 내의 복수의 메모리 파트가 각각 연결선 속성값과 뉴런 속성값을 출력하는 단계; 및

상기 제어 유닛의 제어에 따라, 복수의 계산 유닛이 상기 복수 개의 메모리 유닛 내의 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키는 단계를 포함하되,

상기 복수 개의 메모리 유닛 내의 상기 복수의 메모리 파트와 상기 복수의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 방법.

청구항 65

신경망 컴퓨팅 방법에 있어서,

제어 유닛의 제어에 따라, 복수 개의 메모리 유닛이 각각 연결선 속성값과 뉴런 오차값을 출력하는 단계; 및

상기 제어 유닛의 제어에 따라, 하나의 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 오차값을 이용하여 새로운 뉴런 오차값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키는 단계를 포함하되,

상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 방법.

청구항 66

신경망 컴퓨팅 방법에 있어서,

제어 유닛의 제어에 따라, 복수 개의 메모리 유닛이 각각 연결선 속성값과 뉴런 속성값을 출력하는 단계;

상기 제어 유닛의 제어에 따라, 하나의 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선

속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값과 학습 속성값을 계산하는 단계; 및

상기 제어 유닛의 제어에 따라, 상기 복수 개의 메모리 유닛이 연결선 속성값과 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하는 단계를 포함하되,

상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 방법.

청구항 67

신경망 컴퓨팅 방법에 있어서,

제어 유닛의 제어에 따라, 복수 개의 메모리 유닛이 각각 연결선 속성값, 순방향 뉴런 속성값 및 역방향 뉴런 속성값을 저장하고 출력하며, 새로운 연결선 속성값을 계산하는 단계; 및

상기 제어 유닛의 제어에 따라, 하나의 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 데이터를 바탕으로 새로운 순방향 뉴런 속성값과 역방향 뉴런 속성값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키는 단계를 포함하되,

상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 방법.

청구항 68

신경망 컴퓨팅 방법에 있어서,

제어 유닛의 제어에 따라, 복수 개의 메모리 유닛 내의 복수의 메모리 파트가 각각 연결선 속성값과 역방향 뉴런 속성값을 출력하는 단계;

상기 제어 유닛의 제어에 따라, 복수 개의 계산 유닛이 상기 복수 개의 메모리 유닛 내의 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 역방향 뉴런 속성값을 이용하여 새로운 역방향 뉴런 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키는 단계;

상기 제어 유닛의 제어에 따라, 상기 복수 개의 메모리 유닛 내의 상기 복수의 메모리 파트가 각각 연결선 속성값과 순방향 뉴런 속성값을 출력하고 연결선 속성값과 순방향 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하는 단계; 및

상기 제어 유닛의 제어에 따라, 상기 복수 개의 계산 유닛이 상기 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 순방향 뉴런 속성값을 이용하여 새로운 순방향 뉴런 속성값과 학습 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키는 단계를 포함하되,

상기 복수 개의 메모리 유닛 내의 상기 복수의 메모리 파트와 상기 복수 개의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작하는, 신경망 컴퓨팅 방법.

명 세 서

기 술 분 야

[0001] 본 발명은 디지털 신경망 컴퓨팅 기술 분야에 관한 것으로, 더욱 상세하게는 전체 구성 요소가 하나의 시스템 클록에 동기화되는 동기화 회로(Synchronized Circuit)로 동작하고, 인공 신경망 데이터를 저장하는 분산형 메모리 구조와 모든 뉴런을 파이프라인 회로에서 시분할로 처리하는 계산 구조를 포함하는, 신경망 컴퓨팅 장치 및 시스템과 그 방법에 관한 것이다.

배 경 기 술

[0002] 디지털 신경망 컴퓨터는 생물학적 신경망을 시뮬레이션하여 두뇌의 역할과 유사한 기능을 구현하려는 목적으로

구현된 전자 회로이다.

[0003] 생물학적 신경망을 인공적으로 구현하기 위해 이와 유사한 구조와 연산 방법들이 다양한 형태로 제시되고 있는데, 이러한 인공 신경망의 구성 방법론을 신경망 모델이라고 한다. 대부분의 신경망 모델에서는 인공 뉴런이 방향성이 있는 연결선으로 연결되어 네트워크를 형성하고, 각 뉴런은 고유의 속성(attribute) 값을 가지며 그 값을 연결선을 통해 전달하는 방법으로 인접한 뉴런의 속성값에 영향을 미친다. 뉴런과 뉴런 사이의 연결선도 고유의 속성값을 가지고 있어서 전달하는 신호의 세기를 조절하는 역할을 한다. 다양한 신경망 모델에서 가장 일반적으로 사용하는 뉴런의 속성값은 뉴런의 출력값에 해당하는 상태(state) 값이며, 가장 일반적으로 사용하는 연결선의 속성값은 연결선의 연결 강도를 나타내는 가중치(weight) 값이다.

[0004] 인공 신경망 내의 뉴런들은 외부로부터 입력 값을 받아들이는 입력 뉴런들과 처리한 결과를 외부로 전달하는 역할을 하는 출력 뉴런들, 그리고 나머지 은닉 뉴런들로 구분할 수 있다.

[0005] 생물학적 신경망과는 다르게 디지털 신경망 컴퓨터에서는 뉴런의 값을 선형적으로 변화시킬 수 없기 때문에 전체의 뉴런에 대해 한 번씩 계산한 후 그 결과 값을 다음 계산 시에 반영하는 방식으로 계산을 진행하며, 전체 뉴런을 한 번씩 계산하는 주기를 신경망 갱신 주기라 한다. 디지털 인공 신경망의 실행은 신경망 갱신 주기를 반복적으로 실행하는 방법으로 진행된다.

[0006] 인공 신경망이 바람직한 결과 값을 도출하기 위하여 신경망 내부에 지식 정보가 연결선 속성값의 형태로 저장된다. 인공 신경망의 연결선의 값을 조정하여 지식을 축적하는 단계를 학습 모드라 하고, 입력 데이터를 제시하여 저장된 지식을 찾는 단계를 회상 모드라 한다.

[0007] 대부분의 신경망 모델에서, 회상 모드에는 입력 뉴런에 입력 데이터 값을 지정한 후 신경망 갱신 주기를 반복하여 수행함으로써 출력 뉴런의 상태값을 도출하는 방식으로 진행되며, 하나의 신경망 갱신 주기 내에서 신경망 내 모든 뉴런 j 각각에 대해 계산하는 뉴런의 상태값은 하기의 [수학식 1]과 같이 계산된다.

수학식 1

[0008]
$$y_j(T+1) = f\left(\sum_{i=1}^{p_j} w_{ij} \cdot y_{Mij}(T)\right)$$

[0009] 여기서, $y_j(T)$는 T번째 신경망 갱신 주기에서 계산된 뉴런 j의 상태값(속성값), f는 뉴런 j의 출력을 결정하는 활성화 함수, p_j는 뉴런 j의 입력 연결선의 수, w_{ij}는 뉴런 j의 i번째 입력 연결선의 가중치 값(속성값), M_{ij}는 뉴런 j의 i번째 입력 연결선에 연결된 뉴런의 번호이다.

[0010] 한편, 상기 [수학식 1]을 사용하는 경우보다는 드물게 "Radial Basis Function"이나 "Self-Organizing Feature Map" 등과 같은 일부 신경망 모델에서는 하기의 [수학식 2]와 같은 계산식을 사용하기도 한다.

수학식 2

[0011]
$$y_j(T+1) = f\left(\sum_{i=1}^{p_j}\left(y_{Mij}(T) - w_{ij}\right)^2\right)$$

[0012] 최근에 대두되고 있는 동적 시냅스 모델 또는 스파이킹 신경망 모델에서는 뉴런이 순간적인 스파이크 신호를 송출하고, 이 스파이크 신호를 전달받은 연결선(시냅스)이 일정 시간 동안 다양한 패턴으로 신호를 생성하며 이 신호들이 합산되어 전달되는 방식을 사용한다. 신호가 전달되는 패턴 유형은 연결선마다 다를 수 있다.

[0013] 학습 모드에는, 하나의 신경망 갱신 주기에 뉴런의 속성값뿐만 아니라 연결선의 속성값이 함께 갱신된다.

[0014] 학습에 가장 많이 사용되는 학습 모델은 역전파(back-propagation) 알고리즘이다. 역전파 알고리즘은 학습 모드

에 시스템 외부의 지도자(supervisor)가 특정 입력값에 상응하는 가장 바람직한 출력값을 지정하는 지도 학습(supervised learning) 방법으로서, 하나의 신경망 갱신 주기(update cycle) 내에서 다음의 1 내지 4와 같은 서브 주기(sub-cycle)를 포함한다.

[0015] 1. 모든 출력 뉴런 각각에 대하여 외부에서 제공된 바람직한 출력 값과 현재의 출력 값을 바탕으로 출력 뉴런의 오차값을 구하는 제 1 서브 주기

[0016] 2. 신경망 내 연결선의 방향이 원래의 방향과 반대 방향인 역방향 네트워크에서, 출력 뉴런의 오차값을 다른 뉴런으로 전파시켜 비 출력 뉴런도 오차값을 갖도록 하는 제 2 서브 주기

[0017] 3. 신경망 내 연결선의 방향이 원래의 방향인 순방향 네트워크에서, 입력 뉴런의 값을 다른 뉴런으로 전파시켜 모든 뉴런의 새로운 상태값을 계산하는 제 3 서브 주기(상기 회상 모드의 내용과 동일)

[0018] 4. 신경망 내 연결선의 방향이 원래의 방향인 순방향 네트워크에서, 모든 뉴런 각각의 모든 연결선 각각에 대해 그 연결선에 연결되어 값을 제공하는 뉴런의 상태값과 값을 받아들이는 뉴런의 속성값을 바탕으로 연결선의 가중치 값을 조정하는 제 4 서브 주기

[0019] 이때, 신경망 갱신 주기 내에서 상기 4개 서브 주기의 실행 순서는 중요하지 않다.

[0020] 상기 제 1 서브 주기는 모든 출력 뉴런에 대하여 하기의 [수학식 3]을 계산하는 단계이다.

수학식 3

[0021]
$$\delta_j(T+1) = teach_j - y_j(T)$$

[0022] 여기서, $teach_j$는 출력 뉴론 j에 제공되는 학습 값(학습 데이터)이고, δ_j는 뉴런 j의 오차이다.

[0023] 상기 제 2 서브 주기는 출력 뉴런 이외의 모든 뉴런에 대하여 하기의 [수학식 4]를 계산하는 단계이다.

수학식 4

[0024]
$$\delta_j(T+1) = \sum_{i=1}^{p'_j} w'_{ij} \cdot \delta_{Rij}(T)$$

[0025] 여기서, $\delta_j(T)$는 신경망 갱신 주기 T에서 뉴런 j의 오차값, p'_j 는 역방향 네트워크에서 뉴런 j의 역방향 연결선의 수, w'_{ij} 는 뉴런 j의 역방향 연결선 중 i번째 연결선의 가중치 값, R_{ij}는 뉴런 j의 역방향 i번째 연결선에 연결된 뉴런의 번호이다.

[0026] 상기 제 3 서브 주기는 모든 뉴런 각각에 대하여 상기 [수학식 1]을 계산하는 단계이다. 이는 상기 제 3 서브 주기가 회상 모드와 동일하기 때문이다.

[0027] 상기 제 4 서브 주기는 모든 뉴런 각각에 대하여 하기의 [수학식 5]를 계산하는 단계이다.

수학식 5

$$w_{ij}(T+1) = w_{ij}(T) + \eta \cdot \delta_j \cdot \frac{df(net_j)}{dnet_j} \cdot y_{Mij}$$

[0028]

[0029] 여기서, η는 상수, net_j는 뉴런 j의 입력 값 $\sum_{i=1}^{p_j} w_{ij} \cdot y_{Mij}(T)$ 이다.

[0030] 인공 신경망의 학습 방법은 신경망 모델에 따라 상기 역전파 알고리즘 이외에도 학습을 위하여 델타 학습법(Delta Learning Rule)이나 헤브의 법칙(Hebb's Rule) 등이 사용될 수 있으나 상기 [수학식 5]를 포함하여 이들의 학습 방법은 하기와 같은 [수학식 6]으로 일반화될 수 있는 특징이 있다.

수학식 6

[0031] $$w_{ij}(T+1) = w_{ij}(T) + \{\text{뉴런 j 의 고유한 값}\} * y_{Mij}$$

[0032] 참고로, 상기 [수학식 6]에서 {뉴런 j의 고유한 값}은 $\eta \cdot \delta_j \cdot \frac{df(net_j)}{dnet_j}$ 이다.

[0033] 그리고 역전파 학습 알고리즘 외에도 심도 신뢰망(Deep Belief Network)과 같은 신경망 모델에서 하나의 신경망의 전체 또는 일부 네트워크에 순방향 전파와 역방향 전파를 번갈아 계산하는 경우가 있다.

[0034] 신경망 컴퓨터는 주어진 입력에 가장 적절한 패턴을 찾아내는 패턴 인식이나 선험적 지식을 바탕으로 미래를 예측하는 용도로 활용되어 로봇 제어, 군사용 장비, 의학, 게임, 기상 정보 처리, 및 인간-기계 인터페이스 등과 같은 다양한 분야에 사용될 수 있다.

[0035] 기존의 신경망 컴퓨터는 크게 직접적(direct) 구현 방법과 가상형(virtual) 구현 방법으로 구분된다. 직접적 구현 방법은 인공 신경망의 논리적 뉴런을 물리적 뉴런에 1대 1로 매핑시켜 구현하는 방식으로, 대부분의 아날로그 신경망칩이 이 범주에 속한다. 이와 같은 직접적 구현 방법은 빠른 처리 속도를 낼 수는 있으나 신경망 모델을 다양하게 적용하기 어렵고 대규모 신경망에 적용이 어려운 단점이 있다.

[0036] 가상형 구현 방법은 대부분 기존의 폰노이만형 컴퓨터를 이용하거나 이와 같은 컴퓨터가 병렬로 연결된 다중 프로세서 시스템을 사용하는 방식으로, "HNC사"의 "ANZA Plus"나 "CNAPS", "IBM사"의 "NEP"나 "SYNAPSE-1" 등이 이와 같은 범주에 속한다. 이와 같은 가상형 구현 방법은 다양한 신경망 모델과 대규모 신경망을 실행할 수 있으나 높은 속도를 얻기 어려운 단점이 있다.

발명의 내용

해결하려는 과제

[0037] 전술한 바와 같이, 종래의 직접적 구현 방법은 빠른 처리 속도를 낼 수는 있으나 신경망 모델을 다양하게 적용할 수 없고 대규모 신경망에 적용이 어려운 문제점이 있으며, 종래의 가상형 구현 방법은 다양한 신경망 모델과 대규모 신경망을 실행할 수 있으나 높은 속도를 얻기 어려운 문제점이 있으며, 이러한 문제점을 해결하고자 하는 것이 본 발명의 과제이다.

[0038] 본 발명은 전체 구성 요소가 하나의 시스템 클록에 동기화되는 동기화 회로(Synchronized Circuit)로 동작하고, 인공 신경망 데이터를 저장하는 분산형 메모리 구조와 모든 뉴런을 파이프라인 회로에서 시분할로 처리하는 계산 구조를 포함함으로써, 다양한 신경망 모델과 대규모 신경망의 적용이 가능하면서 동시에 고속 처리가 가능한

신경망 컴퓨팅 장치 및 시스템과 그 방법을 제공하는 데 그 목적이 있다.

[0039] 본 발명의 목적들은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 본 발명의 다른 목적 및 장점들은 하기의 설명에 의해서 이해될 수 있으며, 본 발명의 실시 예에 의해 보다 분명하게 알게 될 것이다. 또한, 본 발명의 목적 및 장점들은 특허 청구 범위에 나타낸 수단 및 그 조합에 의해 실현될 수 있음을 쉽게 알 수 있을 것이다.

과제의 해결 수단

[0040] 상기 목적을 달성하기 위한 본 발명의 제1장치는, 신경망 컴퓨팅 장치에 있어서, 상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛; 각각 연결선 속성값과 뉴런 속성값을 출력하기 위한 복수 개의 메모리 유닛; 및 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키기 위한 하나의 계산 유닛을 포함한다.

[0041] 또한, 상기 목적을 달성하기 위한 본 발명의 제2장치는, 신경망 컴퓨팅 장치에 있어서, 상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛; 각각 연결선 속성값과 뉴런 속성값을 출력하기 위한 복수 개의 메모리 유닛; 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하기 위한 하나의 계산 유닛; 상기 제어 유닛으로부터의 입력 데이터를 입력 뉴런에 제공하기 위한 입력 수단; 상기 입력 수단으로부터의 입력 데이터 또는 상기 계산 유닛으로부터의 새로운 뉴런 속성값을 상기 제어 유닛의 제어에 따라 상기 복수 개의 메모리 유닛으로 스위칭하기 위한 스위칭 수단; 및 상기 제어 유닛의 제어에 따라 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로로 이루어져, 상기 계산 유닛으로부터의 새로운 뉴런 속성값이 상기 제어 유닛으로 출력되도록 하기 위한 제1 및 제2출력 수단을 포함한다.

[0042] 한편, 상기 목적을 달성하기 위한 본 발명의 제1시스템은, 신경망 컴퓨팅 시스템에 있어서, 상기 신경망 컴퓨팅 시스템을 제어하기 위한 제어 유닛; "각각 연결선 속성값과 뉴런 속성값을 출력하는 복수의 메모리 파트"를 포함하는 복수 개의 메모리 유닛; 및 상기 복수 개의 메모리 유닛 내의 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키기 위한 복수의 계산 유닛을 포함한다.

[0043] 한편, 상기 목적을 달성하기 위한 본 발명의 제3장치는, 신경망 컴퓨팅 장치에 있어서, 상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛; 각각 연결선 속성값과 뉴런 오차값을 출력하기 위한 복수 개의 메모리 유닛; 및 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 오차값을 이용하여 새로운 뉴런 오차값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키기 위한 하나의 계산 유닛을 포함한다.

[0044] 또한, 상기 목적을 달성하기 위한 본 발명의 제4장치는, 신경망 컴퓨팅 장치에 있어서, 상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛; 각각 연결선 속성값과 뉴런 속성값을 출력하고, 연결선 속성값과 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하기 위한 복수 개의 메모리 유닛; 및 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값과 학습 속성값을 계산하기 위한 하나의 계산 유닛을 포함한다.

[0045] 또한, 상기 목적을 달성하기 위한 본 발명의 제5장치는, 신경망 컴퓨팅 장치에 있어서, 상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛; 뉴런의 학습 속성값을 저장하기 위한 제1학습 속성값 메모리; 각각 연결선 속성값과 뉴런 속성값을 출력하고, 연결선 속성값과 뉴런 속성값과 상기 제1학습 속성값 메모리의 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하기 위한 복수 개의 메모리 유닛; 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값과 학습 속성값을 계산하기 위한 하나의 계산 유닛; 및 상기 하나의 계산 유닛에서 계산된 새로운 학습 속성값을 저장하기 위한 제2학습 속성값 메모리를 포함한다.

[0046] 또한, 상기 목적을 달성하기 위한 본 발명의 제6장치는, 신경망 컴퓨팅 장치에 있어서, 상기 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛; 각각 연결선 속성값, 순방향 뉴런 속성값 및 역방향 뉴런 속성값을 저장하고 출력하며, 새로운 연결선 속성값을 계산하기 위한 복수 개의 메모리 유닛; 및 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 데이터를 바탕으로 새로운 순방향 뉴런 속성값과 역방향 뉴런 속성값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키기 위한 하나의 계산 유닛을 포함한다.

[0047] 한편, 상기 목적을 달성하기 위한 본 발명의 제2시스템은, 신경망 컴퓨팅 시스템에 있어서, 상기 신경망 컴퓨팅

시스템을 제어하기 위한 제어 유닛; "각각 연결선 속성값과 역방향 뉴런 속성값을 출력하거나, 각각 연결선 속성값과 순방향 뉴런 속성값을 출력하고 연결선 속성값과 순방향 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하는 복수의 메모리 파트"를 포함하는 복수 개의 메모리 유닛; 및 상기 복수 개의 메모리 유닛 내의 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 역방향 뉴런 속성값을 이용하여 새로운 역방향 뉴런 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키거나, 상기 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 순방향 뉴런 속성값을 이용하여 새로운 순방향 뉴런 속성값과 학습 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키기 위한 복수의 계산 유닛을 포함한다.

[0048] 한편, 상기 목적을 달성하기 위한 본 발명의 제7장치는, 디지털 시스템의 메모리 장치에 있어서, 외부의 제어 유닛으로부터의 제어 신호에 의해 제어되는 복수 개의 디지털 스위치를 이용하여 두 개의 메모리의 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로를 상기 두 개의 메모리에 적용한 것을 특징으로 한다.

[0049] 한편, 상기 목적을 달성하기 위한 본 발명의 제1방법은, 신경망 컴퓨팅 방법에 있어서, 제어 유닛의 제어에 따라, 복수 개의 메모리 유닛이 각각 연결선 속성값과 뉴런 속성값을 출력하는 단계; 및 상기 제어 유닛의 제어에 따라, 하나의 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키는 단계를 포함하되, 상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0050] 또한, 상기 목적을 달성하기 위한 본 발명의 제2방법은, 신경망 컴퓨팅 방법에 있어서, 제어 유닛의 제어에 따라, 상기 제어 유닛으로부터 입력 뉴런에 제공하기 위한 데이터를 입력받는 단계; 상기 입력받은 데이터 또는 계산 유닛으로부터의 새로운 뉴런 속성값을 상기 제어 유닛의 제어에 따라 복수 개의 메모리 유닛으로 스위칭하는 단계; 상기 제어 유닛의 제어에 따라, 상기 복수 개의 메모리 유닛이 각각 연결선 속성값과 뉴런 속성값을 출력하는 단계; 상기 제어 유닛의 제어에 따라, 하나의 상기 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하는 단계; 및 상기 제어 유닛의 제어에 따라 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 회로로 이루어진 제1 및 제2출력 수단이, 상기 계산 유닛으로부터의 새로운 뉴런 속성값이 상기 제어 유닛으로 출력되도록 하는 단계를 포함한다.

[0051] 또한, 상기 목적을 달성하기 위한 본 발명의 제3방법은, 신경망 컴퓨팅 방법에 있어서, 제어 유닛의 제어에 따라, 복수 개의 메모리 유닛 내의 복수의 메모리 파트가 각각 연결선 속성값과 뉴런 속성값을 출력하는 단계; 및 상기 제어 유닛의 제어에 따라, 복수의 계산 유닛이 상기 복수 개의 메모리 유닛 내의 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키는 단계를 포함하되, 상기 복수 개의 메모리 유닛 내의 상기 복수의 메모리 파트와 상기 복수의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0052] 또한, 상기 목적을 달성하기 위한 본 발명의 제4방법은, 신경망 컴퓨팅 방법에 있어서, 제어 유닛의 제어에 따라, 복수 개의 메모리 유닛이 각각 연결선 속성값과 뉴런 오차값을 출력하는 단계; 및 상기 제어 유닛의 제어에 따라, 하나의 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 오차값을 이용하여 새로운 뉴런 오차값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키는 단계를 포함하되, 상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0053] 또한, 상기 목적을 달성하기 위한 본 발명의 제5방법은, 신경망 컴퓨팅 방법에 있어서, 제어 유닛의 제어에 따라, 복수 개의 메모리 유닛이 각각 연결선 속성값과 뉴런 속성값을 출력하는 단계; 상기 제어 유닛의 제어에 따라, 하나의 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값과 학습 속성값을 계산하는 단계; 및 상기 제어 유닛의 제어에 따라, 상기 복수 개의 메모리 유닛이 연결선 속성값과 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하는 단계를 포함하되, 상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0054] 또한, 상기 목적을 달성하기 위한 본 발명의 제6방법은, 신경망 컴퓨팅 방법에 있어서, 제어 유닛의 제어에 따라, 복수 개의 메모리 유닛이 각각 연결선 속성값, 순방향 뉴런 속성값 및 역방향 뉴런 속성값을 저장하고 출력

하며, 새로운 연결선 속성값을 계산하는 단계; 및 상기 제어 유닛의 제어에 따라, 하나의 계산 유닛이 상기 복수 개의 메모리 유닛으로부터 각각 입력되는 데이터를 바탕으로 새로운 순방향 뉴런 속성값과 역방향 뉴런 속성값을 계산하여 상기 복수 개의 메모리 유닛 각각으로 피드백시키는 단계를 포함하되, 상기 복수 개의 메모리 유닛과 상기 하나의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0055] 또한, 상기 목적을 달성하기 위한 본 발명의 제7방법은, 신경망 컴퓨팅 방법에 있어서, 제어 유닛의 제어에 따라, 복수 개의 메모리 유닛 내의 복수의 메모리 파트가 각각 연결선 속성값과 역방향 뉴런 속성값을 출력하는 단계; 상기 제어 유닛의 제어에 따라, 복수 개의 계산 유닛이 상기 복수 개의 메모리 유닛 내의 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 역방향 뉴런 속성값을 이용하여 새로운 역방향 뉴런 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키는 단계; 상기 제어 유닛의 제어에 따라, 상기 복수 개의 메모리 유닛 내의 상기 복수의 메모리 파트가 각각 연결선 속성값과 순방향 뉴런 속성값을 출력하고 연결선 속성값과 순방향 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하는 단계; 및 상기 제어 유닛의 제어에 따라, 상기 복수 개의 계산 유닛이 상기 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 순방향 뉴런 속성값을 이용하여 새로운 순방향 뉴런 속성값과 학습 속성값을 각각 계산하여 상기 상응하는 복수의 메모리 파트 각각으로 피드백시키는 단계를 포함하되, 상기 복수 개의 메모리 유닛 내의 상기 복수의 메모리 파트와 상기 복수 개의 계산 유닛이, 상기 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

발명의 효과

[0056] 상기와 같은 본 발명은, 신경망의 네트워크 토폴로지, 뉴런의 수, 연결선의 수에 제약이 없고, 임의의 활성화 함수가 포함된 다양한 신경망 모델을 실행할 수 있는 효과가 있다.

[0057] 또한, 본 발명은, 신경망 컴퓨팅 시스템이 동시에 처리할 수 있는 연결선의 수 p를 임의로 정하여 설계할 수 있으며, 매 메모리 접근 주기마다 최고 p개의 연결선을 동시에 회상(recall)하거나 학습(train)할 수 있어서 고속 실행이 가능한 장점이 있다.

[0058] 또한, 본 발명은, 구현 가능한 최고 속도를 떨어뜨리지 않고 연산의 정밀도(precision)를 임의로 높일 수 있는 장점이 있다.

[0059] 또한, 본 발명을 적용하면 대용량 범용 신경망 컴퓨터의 구현이 가능할 뿐만 아니라 소형 반도체에도 집적이 가능하여 다양한 인공 신경망 응용 분야에 적용할 수 있는 효과가 있다.

도면의 간단한 설명

[0060] 도 1은 본 발명에 따른 신경망 컴퓨팅 장치의 일실시예 구성도,

도 2는 본 발명에 따른 제어 유닛의 일실시예 상세 구성도,

도 3은 본 발명에 따른 제어 신호에 의하여 진행되는 데이터의 흐름을 나타내는 일예시도,

도 4는 본 발명에 따른 신경망 컴퓨팅 장치의 파이프라인 구조를 설명하기 위한 일예시도,

도 5는 본 발명에 따른 이중 메모리 교체(SWAP) 방식을 설명하기 위한 일예시도,

도 6 및 도 7은 본 발명에 따른 단일 메모리 교체(SWAP) 방식을 설명하기 위한 일예시도,

도 8은 본 발명에 따른 계산 유닛의 일실시예 상세 구성도,

도 9는 본 발명에 따른 계산 유닛에서의 데이터 흐름을 나타내는 일실시예 도면,

도 10은 본 발명에 따른 신경망 컴퓨팅 장치의 다단계 파이프라인 구조를 설명하기 위한 상세 예시도,

도 11은 본 발명에 따른 병렬 계산 라인 기법을 설명하기 위한 일예시도,

도 12는 본 발명에 따른 병렬 계산 라인 기법에 따른 입출력 데이터의 흐름을 나타내는 도면,

도 13은 본 발명에 따른 병렬 계산 라인 기법을 곱셈기 또는 덧셈기 또는 활성화 함수 연산기에 적용한 경우를 나타내는 일예시도,

도 14는 본 발명에 따른 병렬 계산 라인 기법을 누산기에 적용한 경우를 나타내는 일예시도,

도 15는 본 발명에 따른 병렬 계산 라인 기법을 누산기에 적용한 경우의 입출력 데이터의 흐름을 나타내는 도면,

도 16은 본 발명에 따른 신경망 컴퓨팅 장치에 병렬 계산 라인 기법을 적용한 경우 다단계 파이프라인 구조를 설명하기 위한 상세 예시도,

도 17은 본 발명에 따른 계산 유닛의 다른 구조를 설명하기 위한 도면,

도 18은 본 발명에 따른 도 17의 다른 구조의 계산 유닛에서의 입출력 데이터 흐름을 나타내는 도면,

도 19는 본 발명에 따른 활성화 함수 연산기와 YN 메모리의 다른 구조를 설명하기 위한 도면,

도 20은 본 발명에 따른 신경망 컴퓨팅 장치의 다른 실시예 구성도,

도 21은 본 발명에 따른 신경망 갱신 주기를 설명하기 위한 일실시예 도면,

도 22는 [수학식 2]를 계산하는 계산 유닛의 곱셈기에 대한 일실시예 상세 구성도,

도 23은 본 발명에 따른 신경망 컴퓨팅 시스템의 일실시예 구성도,

도 24는 본 발명에 따른 역전파 학습 알고리즘의 제1 서브 주기와 제2 서브 주기를 함께 실행하는 신경망 컴퓨팅 장치의 구조를 설명하기 위한 도면,

도 25는 본 발명에 따른 학습 알고리즘을 실행하는 신경망 컴퓨팅 장치의 구조를 설명하기 위한 도면,

도 26은 본 발명에 따른 도 25의 신경망 컴퓨팅 장치에서의 데이터 흐름을 나타내는 도면,

도 27은 본 발명에 따른 하나의 신경망의 전체 또는 일부 네트워크에 대해 역방향 전파 주기와 순방향 전파 주기를 번갈아 실행하는 신경망 컴퓨팅 장치를 나타내는 도면,

도 28은 본 발명에 따른 도 27의 신경망 컴퓨팅 장치를 간략화한 다른 계산 구조를 설명하기 위한 도면,

도 29는 본 발명에 따른 도 27 및 도 28의 신경망 컴퓨팅 장치 중 계산 유닛의 상세 구성도,

도 30은 본 발명에 따른 도 29의 계산 유닛 중 소마 처리기의 상세 구성도,

도 31은 본 발명에 따른 신경망 컴퓨팅 시스템의 다른 실시예 구성도,

도 32는 계산 유닛에서 실행하는 신경망의 계산 모델이 동적 시냅스 모델 또는 스파이킹 신경망 모델인 경우 계산 유닛의 곱셈기에 대한 일실시예 상세 구성도,

도 33은 본 발명에 따른 학습 알고리즘을 실행하는 신경망 컴퓨팅 장치의 다른 구조를 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0061] 상술한 목적, 특징 및 장점은 첨부된 도면을 참조하여 상세하게 후술되어 있는 상세한 설명을 통하여 보다 명확해 질 것이며, 그에 따라 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 또한, 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에 그 상세한 설명을 생략하기로 한다. 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 실시 예를 상세히 설명하기로 한다. 그리고 본 발명에 따른 장치 및 시스템의 구성 설명과 함께 그 동작도 함께 설명하기로 한다.

[0062] 그리고 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때 이는 "직접적으로 연결"되어 있는 경우뿐만 아니라 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한, 어떤 부분이 어떤 구성요소를 "포함" 또는 "구비"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함하거나 구비할 수 있는 것을 의미한다.

[0063] 도 1은 본 발명에 따른 신경망 컴퓨팅 장치의 일실시예 구성도로서, 그 기본적인 상세 구조를 나타내고 있다.

[0064] 도 1에 도시된 바와 같이, 본 발명에 따른 신경망 컴퓨팅 장치는, 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛(119), 각각 연결선 속성값과 뉴런 속성값을 출력하기 위한 복수 개의 메모리 유닛(일명 시냅스 유닛이라 함, 100), 및 상기 복수 개의 메모리 유닛(100)으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값(다음 신경망 갱신 주기의 뉴런 속성값으로 사용됨)을 계산하여 상기 복수 개의 메모리 유닛(100) 각각으로 피드백시키기 위한 한 개의 계산 유닛(101)을 포함한다.

[0065] 여기서, 각각 제어 유닛(119)과 연결되는 InSel 입력(연결선 묶음 번호, 112)과 OutSel 입력(다음 신경망 갱신 주기의 뉴런 속성값이 저장될 주소와 쓰기 허용 신호, 113)은 각각 상기 복수 개의 모든 메모리 유닛(100)에 공통으로 연결된다. 상기 복수 개의 메모리 유닛(100)의 각 출력(연결선 속성값과 뉴런 속성값, 114, 115)은 상기 계산 유닛(101)의 입력으로 연결된다. 그리고 상기 계산 유닛(101)의 출력(다음 신경망 갱신 주기의 뉴런 속성값)은 Y 버스(bus)(111)를 통해 상기 복수 개의 모든 메모리 유닛(100)의 입력에 공통으로 연결된다.

[0066] 각각의 메모리 유닛(100)은, 연결선 속성값을 저장하기 위한 W메모리(제1메모리, 102), 뉴런의 고유번호(예 : 뉴런 속성값이 저장되어 있는 YC메모리의 주소값)를 저장하기 위한 M메모리(제2메모리, 103), 뉴런 속성값을 저장하기 위한 YC메모리(제3메모리, 104) 및 상기 계산 유닛(101)에서 계산된 새로운 뉴런 속성값(다음 신경망 갱신 주기의 뉴런 속성값)을 저장하기 위한 YN메모리(제4메모리, 105)를 포함한다.

[0067] 이때, W메모리(102)와 M메모리(103)의 주소 입력(AD : Address Input)은 공통으로 묶여 InSel 입력(112)과 연결되고, 상기 M메모리(103)의 데이터 출력(DO : Data Output)은 상기 YC메모리(104)의 주소 입력과 연결된다. W메모리(102)와 YC메모리(104)의 데이터 출력은 각각 계산 유닛(101)의 입력으로 연결된다. OutSel 입력(113)은 YN메모리(105)의 주소 입력과 WE(Write Enable) 입력에 연결되고, Y 버스(111)는 YN메모리(105)의 데이터 입력(DI : Data Input)으로 연결된다.

[0068] 상기 메모리 유닛(100)의 W메모리(102)의 주소 입력단에는 제1레지스터(W메모리로 입력되는 연결선 묶음 번호를 임시 저장함, 106)가 더 포함될 수 있고, 상기 YC메모리(104)의 주소 입력단에는 제2레지스터(M메모리에서 출력되는 뉴런의 고유번호를 임시 저장함, 107)가 더 포함될 수 있다.

[0069] 상기 메모리 유닛(100)의 W메모리(102)의 주소 입력단에는 제1레지스터(W메모리로 입력되는 연결선 묶음 번호를 임시 저장함, 106)가 더 포함될 수 있고, 상기 YC메모리(104)의 주소 입력단에는 제2레지스터(M메모리에서 출력되는 뉴런의 고유번호를 임시 저장함, 107)가 더 포함될 수 있다. 상기 제1 및 제2 레지스터(106, 107)는 하나의 시스템 클록에 동기화되어 상기 W메모리(102), M메모리(103) 및 YC메모리가 제어 유닛(119)의 제어에 따라 파이프라인 방식으로 동작하도록 한다.

[0070] 그리고 상기 복수 개의 모든 메모리 유닛(100)의 출력과 상기 계산 유닛(101)의 입력 사이에 복수의 제3레지스터(W메모리로부터의 연결선 속성값과 YC메모리로부터의 뉴런 속성값을 임시 저장함, 108, 109)가 더 포함될 수 있다. 또한, 상기 계산 유닛(101)의 출력단에 제4레지스터(계산 유닛에서 출력되는 새로운 뉴런 속성값을 임시 저장함, 110)가 더 포함될 수 있다. 상기 제3 및 제4레지스터(108 내지 110)는 하나의 시스템 클록에 의해 동기화되어 상기 복수 개의 메모리 유닛(100)과 상기 하나의 계산 유닛(101)이 제어 유닛(119)의 제어에 따라 파이프라인 방식으로 동작하도록 한다.

[0071] 또한, 상기 계산 유닛(101)의 출력과 상기 복수 개의 모든 메모리 유닛(100)의 입력 사이에는, 제어 유닛(119)으로부터 입력 뉴런의 값이 인입되는 라인(117)과 계산 유닛(101)에서 새로 계산된 뉴런의 속성값이 출력되는 Y 버스(111) 중 하나를 선택하여 각 메모리 유닛(100)으로 연결하는 디지털 스위치(116)를 더 포함할 수 있다. 그리고 계산 유닛(101)의 출력(118)은 제어 유닛(119)과 연결되어 뉴런의 값을 외부로 전달한다.

[0072] 상기 메모리 유닛(100)의 W메모리(102)와 M메모리(103) 및 YC메모리(104)의 초기값은 제어 유닛(119)에 의해 미리 저장된다. 제어 유닛(119)이 상기 메모리 유닛(100) 내부의 각 메모리에 값을 저장하는 방식으로는, 다음과 같은 a 내지 h의 절차에 따라 각 메모리에 값을 저장할 수 있다.

[0073] a. 신경망 내에서 가장 많은 수의 입력 연결선을 가진 뉴런의 입력 연결선의 수(Pmax)를 찾는 단계

[0074] b. 상기 메모리 유닛의 수를 p라 할 때, 신경망 내의 모든 뉴런이 $\lceil Pmax/p \rceil * p$ 개의 연결선을 갖도록 각각의 뉴런에 어떤 뉴런이 연결되어도 인접 뉴런에 영향을 미치지 않는 연결선 속성값을 갖는 가상의 연결선을 추가하는 단계

[0075] (1) 상기 가상의 연결선을 추가하는 방식의 하나로서 어떤 뉴런과 연결되어도 뉴런의 속성값에 영향을 주지 않는 연결선의 속성값을 갖도록 하는 방식

[0076] (2) 상기 가상의 연결선을 추가하는 방식 중 하나로서 신경망에 어떤 뉴런과 연결되어도 영향을 주지 않는 속성값을 가진 하나의 가상의 뉴런을 추가하고 모든 가상의 연결선들이 이 가상의 뉴런과 연결되도록 하는 방식

[0077] c. 신경망 내 모든 뉴런을 임의의 순서로 정렬하고 일련번호를 부여하는 단계

[0078] d. 모든 뉴런 각각의 연결선을 p개씩 나누어 $\lceil Pmax/p \rceil$ 개의 묶음으로 분류하고 묶음들을 임의의 순서로 정렬하는 단계

[0079] e. 첫 번째 뉴런의 첫 번째 연결선 묶음부터 마지막 번째 뉴런의 마지막 연결선 묶음까지 순서대로 일련 번호 k를 부여하는 단계

[0080] f. 상기 메모리 유닛(100) 중 i번째 메모리 유닛의 W메모리(102)의 k번째 주소에는 k번째 연결선 묶음의 i번째 연결선의 속성값을 저장하는 단계

[0081] g. 상기 모든 메모리 유닛의 YC메모리(104)에는 j번째 주소에 j번째 뉴런의 속성값을 저장하는 단계

[0082] h. 상기 메모리 유닛 중 i번째 메모리 유닛의 M메모리(103)의 k번째 주소에는 k번째 연결선 묶음의 i번째 연결선에 연결된 뉴런의 번호 값(뉴런의 속성값이 상기 메모리 유닛 중 i번째 메모리 유닛의 YC메모리(104)에 저장된 주소값)을 저장하는 단계

[0083] 상기와 같이 메모리에 초기값을 저장한 후, 신경망 갱신 주기를 시작하면 제어 유닛(119)은 InSel 입력에 1부터 시작해서 매 시스템 클록 주기마다 1씩 증가하는 연결선 묶음의 번호 값을 공급하고, 신경망 갱신 주기가 시작되고 나서 일정 시스템 클록 주기가 지난 후부터 복수 개의 메모리 유닛(100)의 출력에는 매 시스템 클록 주기마다 특정 연결선 묶음에 포함된 모든 연결선 각각의 연결선 속성값과 그 연결선에 입력으로 연결된 뉴런의 속성값이 순차적으로 출력된다. 이와 같이 순차적으로 출력되는 연결선 묶음의 순서는 1번 뉴런의 첫 번째 연결선 묶음부터 마지막 연결선 묶음까지, 그리고 그 다음 뉴런의 첫 번째 연결선 묶음부터 마지막 연결선 묶음까지의 순서로 반복되고, 마지막 뉴런의 마지막 연결선 묶음이 출력될 때까지 반복된다.

[0084] 그리고 계산 유닛(101)은 메모리 유닛(100)의 출력(연결선 속성값과 뉴런 속성값)을 입력으로 받아 뉴런의 새로운 속성값을 계산한다. 모든 뉴런이 각각 n개의 연결선 묶음을 가진 경우 신경망 갱신 주기가 시작되고 나서 일정 시스템 클록 주기가 지난 후부터 계산 유닛(101)의 입력으로는 각 뉴런의 연결선 묶음의 데이터가 순차적으로 입력되고, 계산 유닛(101)의 출력에는 매 n번의 시스템 클록 주기마다 새로운 뉴런의 속성값이 계산되어 출력된다.

[0085] 도 2는 본 발명에 따른 제어 유닛의 일실시예 상세 구성도이다.

[0086] 도 2에 도시된 바와 같이, 본 발명에 따른 제어 유닛(201)은, 도 1에서 전술한 바와 같은 신경망 컴퓨팅 장치(202)에 각종 제어 신호를 제공하고 메모리 유닛 내 각 메모리의 초기화, 실시간 또는 비 실시간 입력 데이터 로딩, 실시간 또는 비 실시간 출력 데이터 인출 등의 역할을 수행한다. 그리고 제어 유닛(201)은 호스트 컴퓨터(200)에 연결되어 사용자로부터의 제어를 받을 수 있다.

[0087] 그리고 제어 메모리(204)는 신경망 갱신 주기 내에서 각각의 연결선 묶음과 뉴런 하나하나를 처리하기 위해 필요한 모든 제어 신호(205)의 타이밍 및 제어 정보를 저장하며, 클록 주기 카운터(203)로부터 제공되는 신경망 갱신 주기 내의 클록 주기에 따라 제어 신호가 추출될 수 있다.

[0088] 도 3은 본 발명에 따른 제어 신호에 의하여 진행되는 데이터의 흐름을 나타내는 일예시도이다.

[0089] 도 3에 도시된 일예에서는 모든 뉴런이 각각 2개씩의 연결선 묶음을 갖는 것으로 가정하였다($\lceil Pmax/p \rceil = 2$).

[0090] 하나의 신경망 갱신 주기가 시작되면, 제어 유닛(201)에 의해 InSel 입력(112)을 통해 연결선 묶음의 고유 번호가 순차적으로 입력된다. 특정 클록 주기에 InSel 입력(112)에 특정 연결선 묶음의 번호인 k 값이 제공되면, 다

음 클록 주기에 제1 및 제2레지스터(106, 107)에는 각각 k 값과 k번째 연결선 묶음의 i번째 연결선에 속성값을 제공하는 뉴런의 고유번호가 저장된다. 그 다음 클록 주기가 되면 복수의 제3레지스터(108, 109)에 각각 k번째 연결선 묶음의 i번째 연결선의 속성값과 k번째 연결선 묶음의 i번째 연결선에 속성값을 제공하는 뉴런의 속성값이 저장된다.

[0091] p개의 메모리 유닛(100)은 하나의 연결선 묶음에 속한 p개의 연결선의 속성값과 각 연결선에 연결된 뉴런의 속성값을 동시에 출력하여 계산 유닛(101)에 제공하고, 뉴런 j의 2개의 연결선 묶음의 데이터가 계산 유닛(101)에 입력되고 난 후 계산 유닛(101)에서 새로운 뉴런 속성값을 계산하고 나면 제4레지스터(110)에 뉴런 j의 새로 계산된 속성값이 저장된다. 상기 제4레지스터(110)에 저장된 새로운 뉴런 속성값은 다음 클록 주기에 모든 메모리 유닛(100)의 YN메모리(104) 각각에 공통으로 저장된다(각 YN메모리에 저장된 새로운 뉴런 속성값은 다음 신경망 갱신 주기의 뉴런 속성값으로 이용됨). 이때, 저장될 주소와 쓰기 허용 신호(WE)는 제어 유닛(201)에 의해 OutSel 입력(113)을 통해 제공된다. 도 3에서 굵은 선으로 표시된 칸은 j=2인 뉴런 j의 새로운 속성값을 계산하는 데이터의 흐름을 구분한 것이다.

[0092] 신경망 내 모든 뉴런의 새로운 속성값이 모두 계산되어 마지막 뉴런의 새로운 속성값이 YN메모리(104)에 저장이 완료되고 나면, 하나의 신경망 갱신 주기가 종료되고 다음 차례의 신경망 갱신 주기가 시작될 수 있다.

[0093] 도 4는 본 발명에 따른 신경망 컴퓨팅 장치의 파이프라인 구조를 설명하기 위한 일예시도이다.

[0094] 도 4에 도시된 바와 같이, 본 발명에 따른 신경망 컴퓨팅 장치는 제어 유닛의 제어에 따라 다단계(stage)로 이루어진 파이프라인 회로와 같이 동작한다. 파이프라인 이론에 따르면 파이프라인 회로에서 클록의 주기, 즉 파이프라인 주기는 파이프라인 각 단계 중에서 가장 시간이 많이 걸리는 단계의 시간까지 단축이 가능하다. 따라서 tmem을 메모리 접근 시간이라 하고 tcalc를 계산 유닛의 계산 주기(throughput)라 하면, 본 발명에 따른 신경망 컴퓨팅 장치의 이상적인 파이프라인 주기는 max(tmem, tcalc)이다. 하기에 후술하는 바와 같이 계산 유닛을 내부적으로 파이프라인 회로로 구성하면 계산 유닛의 계산 주기(tcalc)를 더 단축할 수 있다.

[0095] 상기 계산 유닛은 입력 데이터가 순차적으로 입력되고 계산 결과가 순차적으로 출력되며 입출력 간의 시간적인 의존성이 없는 특징이 있다. 따라서 입력 데이터가 입력되고 나서 출력 데이터가 계산되는 지연 시간(latency)은 계산할 데이터가 많은 경우 시스템의 성능에 크게 영향을 주지 않으나 대신 출력 데이터가 계산되는 계산 주기(throughput)가 시스템의 성능에 영향을 미친다. 따라서 계산 주기를 단축하기 위하여 계산 유닛의 내부 구조를 파이프라인 방식으로 설계하는 것이 바람직하다.

[0096] 즉, 계산 유닛의 계산 주기를 줄이기 위한 방법의 하나로서, 계산 유닛 내부의 계산 단계 사이에 시스템 클록에 의해 동기화되는 레지스터를 추가하여 각 계산 단계를 파이프라인으로 처리하는 방법을 사용할 수 있다. 이 경우 계산 유닛의 계산 주기는 각 계산 단계의 계산 주기 중 최대값으로 단축될 수 있다. 이 내용은 계산 유닛이 수행하는 계산식의 종류에 관계없이 적용될 수 있으며, 예를 들어 특정 계산식의 전제 하에 설명하는 하기의 도 8의 실시예를 통해 보다 명확해 질 것이다.

[0097] 계산 유닛의 파이프라인 주기를 줄이기 위한 추가적인 방법으로서, 계산 유닛에 속한 전체 또는 일부의 계산 장치 각각에 대해, 계산 장치 내부 구조를 시스템 클록에 동기화되는 파이프라인 회로로 구현하는 방법을 사용할 수 있다. 이 경우 각 계산 장치의 계산 주기는 내부 구조의 파이프라인 주기로 단축될 수 있다.

[0098] 상기에서 설명한 바와 같이, 계산 유닛 내부의 특정 계산 장치의 내부 구조를 파이프라인화하는 방법으로서, 그 계산 장치의 입력의 개수에 해당하는 분배기와 복수 개의 계산 장치와 그 계산 장치의 출력의 수에 해당하는 개수의 다중화기를 사용하여, 순차적으로 인입되는 입력 데이터를 분배기를 통해 복수 개의 계산 장치로 분산시키고 복수 개의 계산 장치의 계산 결과를 다중화기로 수합하는 병렬 계산 라인 기법을 적용할 수 있다. 이 내용은 계산 유닛이 수행하는 계산식의 종류에 관계없이 적용될 수 있으며, 예를 들어 특정 계산식의 전제 하에 설명하는 하기의 도 11의 실시예를 통해 보다 명확해 질 것이다.

[0099] 한편, 하나의 신경망 갱신 주기에서 생산된 뉴런의 속성값은 다음 신경망 갱신 주기에 입력 데이터로 사용되므로, 하나의 신경망 갱신 주기가 끝나고 다음 신경망 갱신 주기가 시작될 때 YN메모리(401)의 내용은 YC메모리(400)의 위치에 저장되어 있어야 한다. 그러나 YN메모리(401)의 내용을 YC메모리(400)로 복사하는 경우 처리 시간이 소요되어 시스템의 성능을 크게 저하시킬 수 있다. 이를 해결하는 방법으로는 하기에 설명하는 (1) 상기

두 개의 메모리를 이중 메모리 교체(SWAP) 방식으로 구현하는 방법, (2) 단일 메모리 중복 저장 방법, (3) 단일 메모리 교체 회로를 사용하는 방법 등이 있다.

[0100] 먼저, 이중 메모리 교체 방식은 1비트 디지털 스위치를 복수 개 사용하여 두 개의 동일한 장치(메모리)의 입출력을 완전히 바꾸어 연결하는 것과 동일한 효과를 낼 수 있는 방식이다.

[0101] 도 5는 본 발명에 따른 이중 메모리 교체(SWAP) 방식을 설명하기 위한 일예시도이다.

[0102] 1비트 스위치를 구현하는 방식의 하나로서 도 5의 (a)와 같은 논리 회로를 사용할 수 있다. 일예로 1비트 스위치는 도 5의 (b)에 도시된 "500"과 같이 표현하고, 1비트 스위치 N개로 구성되는 N-비트 스위치는 도 5의 (b2)와 같이 표기하기로 한다.

[0103] 도 5의 (c)는 3비트의 입력과 1비트의 출력을 가진 두 개의 물리적 장치(D1, D2)가 교체 회로로 구현된 구조를 예시한 것이다. 제어 신호에 따라 모든 스위치가 우측 위치로 연결된 경우, 물리적 장치 D1(501)은 a11, a21, a31이 입력으로 연결되고 a41이 출력으로 연결되며, 물리적 장치 D2(502)는 a12, a22, a32가 입력으로 연결되고 a42가 출력으로 연결된다. 한편, 제어 신호에 의해 모든 스위치가 좌측 위치로 연결되면 물리적 장치 D1(501)은 a12, a22, a32가 입력으로 연결되고 a42가 출력으로 연결되며, 물리적 장치 D2(502)는 a11, a21, a31이 입력으로 연결되고 a41이 출력으로 연결되어 두 물리적 장치(501, 502)가 그 역할을 서로 바꾸게 수행하게 된다. 도 5의 (d)와 같이 교체 회로는 두 개의 물리적 장치(503, 504)를 점선으로 연결하고 교체(swap)를 표기하는 것으로 단순화하여 표현할 수 있다.

[0104] 이와 같은 방식으로 교체 회로를 두 개의 메모리(505, 506)에 적용한 이중 메모리 교체 회로는 도 5의 (e)과 같다.

[0105] 상기 도 1에서 YC메모리(104)와 YN메모리(105)에 이중 메모리 교체 방식을 적용하고 사용하지 않는 입출력을 생략한 회로는 도 5의 (f)와 같이 표기하기로 한다.

[0106] 이러한 이중 메모리 교체 방식을 적용하면, 하나의 신경망 갱신 주기가 끝나고 다음 신경망 갱신 주기를 시작하기 전에 제어 유닛의 제어에 따라 두 개의 메모리의 역할을 교체함으로써, 물리적으로 메모리의 내용을 이동하지 않고서도 이전 갱신 주기에서 저장된 YN메모리(105)의 내용을 YC메모리(104)에서 바로 이용할 수 있다.

[0107] 다음으로, 단일 메모리 중복 저장 방법은, 전술한 바와 같이 두 개의 메모리(도 1의 YC메모리와 YN메모리)를 사용하는 대신에, 하나의 메모리를 사용하고 읽기 과정(도 1의 YC메모리의 역할)과 쓰기 과정(도 1의 YN메모리의 역할)을 하나의 파이프라인 주기에 시간 분할로 처리하고 뉴런의 속성값은 기존 값과 새로운 값의 구분이 없이 같은 저장 장소(메모리)에 저장하는 방법이다.

[0108] 다음으로, 단일 메모리 교체(SWAP) 방법은, 전술한 바와 같이 두 개의 메모리(도 1의 YC메모리와 YN메모리)를 사용하는 대신에, 하나의 메모리를 사용하고 읽기 과정(도 1의 YC메모리의 역할)과 쓰기 과정(도 1의 YN메모리의 역할)을 하나의 파이프라인 주기에 시간 분할로 처리하고 기존의 뉴런의 속성값은 메모리 저장 공간의 반부 영역에 저장하고 계산 유닛에서 계산된 다음 신경망 갱신 주기의 뉴런의 속성값은 다른 반부 영역에 저장하는 방법이다. 다음 신경망 갱신 주기에는 두 메모리 영역의 역할을 바꾸어 사용한다.

[0109] 도 6 및 도 7은 본 발명에 따른 단일 메모리 교체(SWAP) 방식을 설명하기 위한 일예시도이다.

[0110] 도 6에 도시된 바와 같이, 본 발명에 따른 단일 메모리 교체(SWAP) 방식은 한 개의 N비트 스위치(601), 한 개의 배타적 논리합(Exclusive OR) 게이트(603) 및 한 개의 메모리(602)를 이용하여 구현할 수 있다.

[0111] N비트 스위치(601)의 읽기/쓰기(READ/WRITE) 제어 입력(604)은 배타적 논리합 게이트(603)의 입력 중 하나로 연결되고, 이븐사이클(EVENCYCLE) 제어 입력(605)은 배타적 논리합 게이트(603)의 다른 입력으로 연결된다. 그리고 배타적 논리합 게이트(603)의 출력은 메모리(602)의 주소입력 중 최상위 비트로 연결된다.

[0112] 도 7에 도시된 바와 같이, 하나의 파이프라인 주기는 디지털 스위치(601)의 위치가 상단으로 연결되어 읽기 모드로 동작하는 단계와 디지털 스위치(601)의 위치가 하단으로 연결되어 쓰기 모드로 동작하는 단계로 구분된다.

[0113] 읽기/쓰기(READ/WRITE) 제어 입력(604)으로는 현재 갱신 주기의 뉴런의 속성값을 읽을 때는 1의 값이, 새로 계산된 뉴런의 속성값을 저장할 때에는 0의 값이 제공된다. 그리고 이븐사이클(EVENCYCLE) 제어 입력(605)으로는 신경망 갱신 주기 번호가 짝수일 때는 0의 값이, 홀수 일 때는 1의 값이 제공된다.

[0114] 메모리(602)의 전체 영역은 상반부 영역과 하반부 영역으로 구분되며, 신경망 갱신 주기 번호가 홀수일 때에는 메모리(602)의 상반부 영역은 YC메모리로 사용되고 하반부 영역은 YN메모리로 사용되며, 신경망 갱신 주기 번호가 짝수일 때에는 메모리(602)의 상반부 영역은 YN메모리로 사용되고 하반부 영역은 YC메모리로 역할을 번갈아 가며 사용된다.

[0115] 이러한 본 발명에 따른 단일 메모리 교체(SWAP) 방식은, 하나의 파이프라인 클록 주기 내에서 읽기 및 쓰기와 같이 두 번의 메모리 접근이 필요하여 처리 속도가 느려지는 단점이 있는 반면에, 두 개의 메모리(도 1의 YC메모리와 YN메모리) 대신에 하나의 메모리로 구현할 수 있는 장점이 있다.

[0116] 도 8은 본 발명에 따른 계산 유닛(101)의 일실시예 상세 구성도이다.

[0117] 예를 들어, 도 1에서 실행하는 신경망의 계산 모델이 상기 [수학식 1]과 같은 경우, 계산 유닛(101)의 기본적인 구조는 도 8과 같이 구현될 수 있다.

[0118] 도 8에 도시된 바와 같이, 본 발명에 따른 계산 유닛(101)은, 메모리 유닛(100)의 수만큼의 곱셈기로 이루어져 각 메모리 유닛(100)으로부터의 연결선 속성값과 뉴런 속성값에 대해 곱셈 연산을 수행하기 위한 곱셈 연산부(800), 트리 구조로 이루어져 곱셈 연산부(800)로부터의 복수의 출력값에 대해 다단으로 덧셈 연산을 수행하기 위한 덧셈 연산부(802, 804, 806), 덧셈 연산부(802, 804, 806)로부터의 출력값을 누적 연산하기 위한 하나의 누산기(accumulator, 808), 및 누산기(808)로부터의 누적 출력값에 활성화 함수를 적용하여 다음 신경망 갱신 주기에 사용될 새로운 뉴런 속성값을 계산하기 위한 하나의 활성화 함수 연산기(811)를 포함한다.

[0119] 여기서, 본 발명에 따른 계산 유닛(101)은, 각 연산 스텝 사이마다 레지스터(801, 803, 805, 807, 809)를 더 포함할 수 있다.

[0120] 즉, 본 발명에 따른 계산 유닛(101)은, 곱셈 연산부(800)와 덧셈 연산부(802, 804, 806) 트리 중 첫 번째 덧셈 연산부(802) 사이에 구비되는 복수 개의 레지스터(801), 덧셈 연산부(802, 804, 806) 트리의 각 스텝 사이에 구비되는 복수 개의 레지스터(803, 805), 덧셈 연산부(802, 804, 806) 트리의 마지막 덧셈 연산부(806)와 누산기(808) 사이에 구비되는 레지스터(807), 및 누산기(808)와 활성화 함수 연산기(811) 사이에 구비되는 레지스터(809)를 더 포함한다. 여기서, 각 레지스터는 하나의 시스템 클록에 따라 동기화되고 각 계산 단계는 파이프라인 방식으로 동작한다.

[0121] 다음으로, 본 발명에 따른 계산 유닛(101)의 동작을 좀 더 구체적으로 예를 들어 살펴보면, 곱셈 연산부(800)와 트리 형태의 덧셈 연산부(802, 804, 806)는 총체적으로 일련의 신경망 연결선 묶음에 포함된 연결선을 통해 들어오는 입력의 총 합을 순차적으로 계산한다.

[0122] 그리고 누산기(808)는 연결선 묶음의 입력의 총 합을 누적 계산하여 뉴런의 입력의 총 합을 계산하는 역할을 한다. 이때, 덧셈 연산부 트리의 출력에서 누산기(808)로 입력되는 데이터가 특정 뉴런의 첫 번째 연결선 묶음의 데이터이면 디지털 스위치(810)가 제어 유닛(201)에 의해 좌측 단자로 전환되어 0 값이 누산기(808)의 다른 입력에 제공되어 누산기(808)의 출력이 새로운 값으로 초기화된다.

[0123] 그리고 활성화 함수 연산기(811)는 뉴런의 입력의 총 합에 활성화 함수를 적용하여 새로운 뉴런 속성값(상태값)을 계산하는 역할을 한다. 이때, 활성화 함수 연산기(811)는 메모리 참조 테이블과 같은 단순한 구조로 구현할 수도 있고, 또는 마이크로 코드로 실행되는 전용 프로세서로 구현할 수도 있다.

[0124] 도 9는 본 발명에 따른 계산 유닛에서의 데이터 흐름을 나타내는 일실시예 도면이다.

[0125] 도 9에 도시된 바와 같이, 특정 시점에 곱셈 연산부(800)의 입력단에 어떤 연결선 묶음 k의 데이터가 제공되면, 연결선 묶음 k의 데이터는 다음 클록 주기에 곱셈 연산부(800)의 출력단에 나타나고, 그 다음 클록 주기에는 첫

번째 덧셈 연산부(802)의 출력단에 나타나는 방식으로 한 단계씩 전진하면서 데이터가 처리되며, 최종적으로 마지막 덧셈 연산부(806)에 이르면 연결선 묶음 k의 순입력으로 계산된다. 이 연결선 묶음 k의 순입력은 누산기(808)에 의해 하나씩 합산되어 한 뉴런의 연결선 묶음의 수가 n일 때 n회 합산되어 한 뉴런 j의 순입력으로 계산된다. 뉴런 j의 순입력은 n개의 클록 주기 동안 활성화 함수에 의해 뉴런의 새로운 속성값으로 계산되어 출력된다.

[0126] 이때, 특정 처리 스텝에서 연결선 묶음 k의 데이터가 처리되면, 그 전 처리 스텝에서는 연결선 묶음 k-1의 데이터가 처리되고, 다음 처리 스텝에서는 연결선 묶음 k+1의 데이터가 동시에 처리된다.

[0127] 도 10은 본 발명에 따른 신경망 컴퓨팅 장치의 다단계 파이프라인 구조를 설명하기 위한 상세 예시도로서, 다단계로 이루어진 파이프라인 회로를 나타내고 있다.

[0128] 도 10에서 tmem을 메모리 접근 시간이라 하고, tmul을 곱셈기 처리 시간이라 하며, tadd를 덧셈기 처리 시간이라 하고, tacti를 활성화 함수의 계산 시간이라 하면, 이상적인 파이프라인 주기는 max(tmem, tmul, tadd, tacti/B)이다. 여기서, B는 각 뉴런 당 연결선 묶음의 수이다.

[0129] 도 10에서 곱셈기와 덧셈기와 활성화 함수 연산기는 각각 내부적으로 파이프라인 방식으로 처리되는 회로로 구성될 수 있다. 곱셈기의 파이프라인 단계의 수를 smul, 덧셈기의 파이프라인 단계의 수를 sadd, 활성화 함수 연산기의 파이프라인 단계의 수를 sacti라 할 때, 전체 시스템의 파이프라인 주기는 max(tmem, tmul/smul, tadd/sadd, tacti/(B*sacti))이다. 이는 곱셈기, 덧셈기, 활성화 함수 연산기가 내부적으로 충분히 파이프라인 방식으로 동작할 수 있다면 파이프라인 주기를 추가로 단축할 수 있다는 것을 의미한다. 그러나 내부적으로 파이프라인 방식으로 동작할 수 없는 경우에도 복수 개의 계산 장치를 사용하여 파이프라인 방식의 회로로 변환할 수 있으며 하기에 설명하는 이 방법을 병렬 계산 라인 기법이라 하기로 한다.

[0130] 도 11은 본 발명에 따른 병렬 계산 라인 기법을 설명하기 위한 일예시도이고, 도 12는 본 발명에 따른 병렬 계산 라인 기법에 따른 입출력 데이터의 흐름을 나타내는 도면이다.

[0131] 상호 의존성이 없는 일련의 동일한 단위 계산을 특정 디바이스 C(1102)에서 실행할 때, 디바이스 C(1102)가 단위 계산을 처리하는데 소요되는 시간을 t_c라 하면 입력 후 결과가 출력될 때까지의 계산 소요 시간(latency)은 t_c이고 계산 처리량(throughput)은 t_c시간 당 하나의 계산이다. 만일, 계산 처리량을 t_c보다 작은 값 t_{ck}시간 당 하나의 계산으로 높이려면 도 11에 도시된 바와 같은 기법을 사용할 수 있다.

[0132] 도 11에 도시된 바와 같이, 입력단에 하나의 분배기(demultiplexer, 1101)가 사용되고 내부에 $\lceil t_c/t_{ck} \rceil$개의 디바이스 C(1102)가 사용되며, 출력단에 하나의 다중화기(multiplexer, 1103)가 사용되고, 분배기(1101)와 다중화기(1103)는 클록 t_{ck}에 의해 동기화된다. 입력단에는 매 t_{ck}클록 주기마다 하나씩의 입력 데이터가 인입되고, 이 입력 데이터는 분배기(1101)에서 각각의 내부 디바이스 C(1102)에 순차적으로 분배된다. 각각의 내부 디바이스 C(1102)는 입력 데이터를 받은 후 t_c시간에 계산을 완료하여 출력하며, 다중화기(1103)에서는 매 t_{ck}시간마다 계산이 완료된 디바이스 C(1102)의 출력을 선택하여 래치(1104)에 저장한다.

[0133] 여기서, 분배기(1101)와 다중화기(1103)는 단순한 로직 게이트와 디코더 회로를 사용하여 구현이 가능하며, 처리 속도에 거의 영향을 미치지 않는다. 이를 본 발명에서는 "병렬 계산 라인 기법"이라 하기로 한다.

[0134] 이와 같은 병렬 계산 라인 기법의 회로는 매 t_{ck}마다 하나의 결과를 출력하는 $\lceil t_c/t_{ck} \rceil$단(stage)의 파이프라인(1105)과 기능적으로 같으며, 계산 처리량(throughput)은 t_{ck}당 1회의 계산으로 높아진다. 이러한 병렬 계산 라인 기법을 사용하면 특정 디바이스 C(1102)의 처리 속도가 낮더라도 복수 개의 디바이스 C(1102)를 사용하여 처리량(throughput)을 원하는 수준까지 임의로 높일 수 있다. 이는 생산 공장에서 생산량을 높이기 위하여 생산 라인을 늘리는 것과 같은 원리이다. 일 예로서 디바이스 C의 수가 4일 때 입출력 데이터의 흐름은 도 12에 도시된 바와 같다.

[0135] 도 13은 본 발명에 따른 병렬 계산 라인 기법을 곱셈기 또는 덧셈기 또는 활성화 함수 연산기에 적용한 경우를

나타내는 일예시도이다.

[0136] 도 13에 도시된 바와 같이, 전술한 바와 같은 병렬 계산 라인 기법으로 디바이스 C(1102)에 곱셈기(1301) 또는 덧셈기(1303) 또는 활성화 함수 연산기(1305)를 대입하면 각각 투입한 디바이스의 수에 비례하여 시간당 계산량(throughput)이 향상된 곱셈기(1302) 또는 덧셈기(1304) 또는 활성화 함수 연산기(1306)를 구현할 수 있다.

[0137] 예를 들어, 곱셈 연산부(800) 내의 각 곱셈기는 하나의 분배기와 복수 개의 곱셈기(1301)와 하나의 다중화기로 이루어져, 클록 주기로 인입되는 입력 데이터를 분배기에 의해 복수 개의 곱셈기(1301)로 차례대로 분배하고 계산이 완료된 데이터를 다중화기에 의해 순서대로 다중화하여 클록 주기로 출력한다.

[0138] 그리고 덧셈 연산부(802, 804, 806) 내의 각 덧셈기는 하나의 분배기와 복수 개의 덧셈기(1303)와 하나의 다중화기로 이루어져, 클록 주기로 인입되는 입력 데이터를 분배기에 의해 복수 개의 덧셈기(1303)로 차례대로 분배하고 계산이 완료된 데이터를 다중화기에 의해 순서대로 다중화하여 클록 주기로 출력한다.

[0139] 그리고 활성화 함수 연산기(811)는 하나의 분배기와 복수 개의 활성화 함수 연산기(1305)와 하나의 다중화기로 이루어져, 클록 주기로 인입되는 입력 데이터를 분배기에 의해 복수 개의 활성화 함수 연산기(1305)로 차례대로 분배하고 계산이 완료된 데이터를 다중화기에 의해 순서대로 다중화하여 클록 주기로 출력한다.

[0140] 도 14는 본 발명에 따른 병렬 계산 라인 기법을 누산기에 적용한 경우를 나타내는 일예시도이다.

[0141] 도 14에 도시된 바와 같이, 전술한 바와 같은 병렬 계산 라인 기법을 누산기에 적용한 경우, 분배기(1400)와 다중화기(1401)는 전술한 바와 같이 구현하나, 내부의 디바이스 각각은 선입선출(FIFO) 큐(1402)와 누산기(1403)가 직렬로 연결된 회로로 대체된다. 이와 같이 구성한 디바이스를 "1405"와 같이 표기하기로 한다. 이때, 클록 주기로 인입되는 입력 데이터는 분배기(1400)에 의해 선입선출(FIFO) 큐(1402)에 차례대로 분배되고 누산기(1403)에서 계산이 완료된 데이터는 다중화기(1401)에 의해 순서대로 다중화되어 클록 주기로 출력된다.

[0142] 일 예로서 누산기(1403)의 단위 합산 계산 시간이 taccum이고, 파이프라인 주기가 t_{ck}이며 $\lceil t_{accum}/t_{ck} \rceil = 2$ 일 때 도 14에 도시된 회로의 구현에 필요한 누산기(1403)의 수는 2개이다. 이러한 일 예에 추가로 뉴런 당 2개씩의 연결선 묶음이 있다고 가정하면, 입출력 데이터의 흐름은 도 15에 도시된 바와 같다.

[0143] 도 15는 본 발명에 따른 병렬 계산 라인 기법을 누산기에 적용한 경우의 입출력 데이터의 흐름을 나타내는 도면이다.

[0144] 도 15에 도시된 바와 같이, 분배기(1400)의 입력에 순차적으로 제공되는 뉴런의 연결선 묶음의 순입력 데이터 net_j는 뉴런 당 연결선 묶음의 수인 2개 단위로 첫 번째 선입선출 큐 q1과 두 번째 선입선출 큐 q2에 번갈아 가며 저장된다. 단위 누산기 acc1, acc2 각각은 앞 단에 있는 선입선출 큐 q1, q2에 데이터가 저장되면 하나씩 인출하여 누산 계산을 진행하여 계산이 완료되면, 계산된 값은 다중화기(1401)와 레지스터(1404)에 의해 선택되어 출력된다.

[0145] 전술한 바와 같이 병렬 계산 라인 기법이 적용된 곱셈기, 덧셈기, 누산기 및 활성화 함수 연산기로 도 10의 각 구성 요소를 상응하여 대체하면 도 16에 도시된 바와 같다.

[0146] 도 16은 본 발명에 따른 신경망 컴퓨팅 장치에 병렬 계산 라인 기법을 적용한 경우 다단계 파이프라인 구조를 설명하기 위한 상세 예시도이다.

[0147] 도 16에 도시된 바와 같이, 모든 곱셈기(1601)와 모든 덧셈기(1602), 그리고 누산기(1603)와 활성화 함수 연산기(1604) 각각은 병렬 계산 라인 기법이 적용되어 필요한 경우 단위 계산 디바이스를 추가하는 방식으로 계산 주기를 임의적으로 단축할 수 있다. 파이프라인 주기는 파이프라인의 각 단계 중에서 가장 시간이 많이 걸리는 단계의 시간까지 단축이 가능한데 메모리 접근 주기인 tmem을 제외하면 나머지 단계는 모두 임의로 단축이 가능하므로, 병렬 계산 라인 기법이 적용된 신경망 컴퓨팅 장치의 이상적인 파이프라인 주기는 tmem이다. 그리고 p를 메모리 유닛의 수라 할 때, 최고 처리 속도는 p/tmem CPS(Connection Per Second)이다.

[0148] 도 22는 [수학식 2]를 계산하는 계산 유닛의 곱셈기에 대한 일실시예 상세 구성도이다.

[0149] 도 1의 계산 유닛(101)에서 실행하는 신경망의 계산 모델이 상기 [수학식 2]와 같은 경우, 도 8의 계산 유닛에서 각각의 곱셈기는, 두 개의 입력 값(연결선 속성값과 뉴런 속성값)이 하나의 뺄셈기(2200)로 연결되고, 뺄셈기(2200)의 출력은 제곱승 계산기(2201)로 연결된 회로로 대체될 수 있다.

[0150] 도 32는 계산 유닛에서 실행하는 신경망의 계산 모델이 동적 시냅스 모델 또는 스파이킹 신경망 모델인 경우 계산 유닛의 곱셈기에 대한 일실시예 상세 구성도이다.

[0151] 도 1의 계산 유닛(101)에서 실행하는 신경망의 계산 모델이 동적 시냅스 모델 또는 스파이킹 신경망 모델인 경우, 도 8의 계산 유닛에서 각각의 곱셈기는, 하나의 참조 테이블(3200)과 하나의 곱셈기(3201)로 이루어진 회로로 대체될 수 있다. 도 32의 (a)에 도시된 바와 같이, 메모리 유닛의 W메모리에 저장되는 연결선의 속성값은 연결선의 가중치값(w_{ij})과 연결선의 동적 유형 식별자($type_{ij}$)로 구분되어 저장되고, 동적 유형 식별자는 참조 테이블(3200)의 복수 개의 테이블 중 하나를 선택한다. 뉴런의 속성값($y_{W(i,j)}$)은 참조 테이블(3200) 내에서 시간축의 값을 나타낸다. 도 32의 (b)에 도시된 바와 같이, 활성화 함수 연산기는 특정 뉴런이 스파이크를 발생하면 출력값으로 0에서 시작해서 매 신경망 갱신 주기마다 점진적으로 증가하는 신호를 송출하며, 이 신호는 도 32의 (c)에 도시된 바와 같이 참조 테이블(3200)에 의해 시간에 따라 변화하는 신호로 변환되고 곱셈기(3201)의 입력 중 하나에 전달된다.

[0152] 한편, 모든 뉴런이 같은 연결선 묶음의 수를 갖는 메모리 저장 방식과 이를 위한 계산 유닛(101)의 구조는 뉴런 간에 연결선의 수의 차이가 큰 경우 연결선 묶음의 수가 작은 뉴런의 경우 빈(NULL) 연결선의 수가 많아져서 효율이 떨어질 수 있다. 또한, 이 경우 활성화 함수 연산기(1604)에 주어지는 계산 시간이 짧아져서 빠른 활성화 함수 연산기(1604)가 필요하거나 병렬 계산 라인 기법의 구성에 많은 수의 활성화 함수 연산기(1604)가 추가되어야 한다.

[0153] 이를 개선하기 위한 계산 유닛(101)의 구조는 도 17에 도시된 바와 같다.

[0154] 도 17은 본 발명에 따른 계산 유닛의 다른 구조를 설명하기 위한 도면이고, 도 18은 본 발명에 따른 도 17의 다른 구조의 계산 유닛에서의 입출력 데이터 흐름을 나타내는 도면이다.

[0155] 도 17에 도시된 바와 같이, 도 8 또는 도 13에서 전술한 바와 같은 누산기와 활성화 함수 연산기 사이에 선입선출(FIFO) 큐(1700)를 둘 수 있다. 이때, 활성화 함수 계산 시간은 전체 뉴런의 평균 연결선 묶음의 수에 해당하는 시간이고, 활성화 함수 연산기의 입력단은 신경망 컴퓨팅 장치의 파이프라인 주기에 동기화되지 않고 입력값이 필요한 임의의 시간에 선입선출 큐(1700)에서 가장 오래전에 저장된 값을 인출하여 사용한다. 이 경우 활성화 함수 연산기는 선입선출 큐(1700)에 누적된 데이터를 하나씩 인출하여 계산할 수 있기 때문에 모든 뉴런에 균등한 계산 시간을 할애하여 계산할 수 있는 장점이 있다.

[0156] 상기와 같은 방법을 사용할 때 활성화 함수 연산기가 선입선출 큐(1700)에서 데이터를 안정적으로 인출하기 위하여, 제어 유닛은 도 1의 메모리 유닛(100) 내부의 각 메모리에 값을 저장하는 방법으로, 다음과 같은 a 내지 h의 절차를 사용할 수 있다.

[0157] a. 신경망 내 모든 뉴런을 각 뉴런에 포함된 입력 연결선의 수를 기준으로 오름차순으로 정렬하고 순서대로 번호를 부여하는 단계

[0158] b. 신경망 내에 다른 뉴런과 연결선으로 연결되어도 영향을 미치지 않는 속성값을 갖는 한 개의 널(null) 뉴런을 추가하는 단계

[0159] c. 뉴런 j의 입력 연결선의 수를 pj라 할 때, 신경망 내의 뉴런 각각이 $\lceil pj/p \rceil * p$개의 연결선을 갖도록 뉴런에 어떤 뉴런과 연결되어도 영향을 미치지 않는 연결선 속성값을 갖고 널(null) 뉴런과 연결된 $\lceil pj/p \rceil * p - pj$개의 연결선을 추가하는 단계(p는 메모리 유닛의 수)

[0160] d. 모든 뉴런 각각의 연결선을 p개씩 나누어 $\lceil pj/p \rceil$ 개의 묶음으로 분류하고 묶음 내의 연결선 각각에 임의의

순서로 1부터 시작하여 1씩 증가하는 번호 i를 부여하는 단계

[0161] e. 첫 번째 뉴런의 첫 번째 연결선 묶음부터 마지막 번째 뉴런의 마지막 연결선 묶음까지 순서대로 1부터 시작하여 1씩 증가하는 번호 k를 부여하는 단계

[0162] f. 메모리 유닛(100) 중 i번째 메모리 유닛의 W메모리(102)의 k번째 주소에는 k번째 연결선 묶음의 i번째 연결선의 속성값을 저장하는 단계

[0163] g. 메모리 유닛(100) 중 i번째 메모리 유닛의 M메모리(103)의 k번째 주소에는 k번째 연결선 묶음의 i번째 연결선에 연결된 뉴런의 번호를 저장하는 단계

[0164] h. 메모리 유닛(100) 중 i번째 메모리 유닛의 YC메모리(104)의 j번째 주소에는 j번째 뉴런의 속성값을 저장하는 단계

[0165] 상기와 같은 방법을 통해 뉴런의 연결선 묶음이 메모리에 저장된 순서를 연결선의 수가 가장 적은 뉴런부터 오름차순으로 정렬하였으므로, 도 18에 도시된 바와 같이, 활성화 함수 연산기가 전체 뉴런의 평균 연결선 묶음의 수에 해당하는 주기로 선입선출 큐(1700)를 읽어 들이면 선입선출 큐(1700)에는 항상 처리할 데이터가 존재하게 되어 중단 없이 처리가 가능하게 된다.

[0166] 이러한 방식을 사용하면 뉴런 사이에 연결선 수의 불균형이 심하더라도 활성화 함수 연산기는 주기적으로 처리가 가능하여 효율을 개선할 수 있다.

[0167] 한편, 지금까지 전술한 신경망 컴퓨팅 장치에서 활성화 함수의 계산 시간은 일정하거나 미리 예측 가능한 것으로 가정하였다. 따라서 활성화 함수의 출력 데이터가 출력되는 타이밍을 미리 알 수 있으며, 모든 메모리 유닛(100)의 YN메모리(105)에 활성화 함수의 출력 데이터를 저장할 때 저장될 주소값인 OutSel 입력(113)의 값은 제어 유닛(201)에서 사전에 예정된 순서대로 생성할 수 있었다.

[0168] 만일, 활성화 함수의 계산 시간이 내부 조건에 따라 달라져서 출력되는 시점을 사전에 파악할 수 없는 경우에는 도 19에 도시된 바와 같은 방식을 사용할 수 있다.

[0169] 도 19는 본 발명에 따른 활성화 함수 연산기와 YN 메모리의 다른 구조를 설명하기 위한 도면이다.

[0170] 도 19에 도시된 바와 같이, 활성화 함수 연산기(1900)는 뉴런의 순입력 데이터를 입력받는 제1입력(1902)과 새로운 속성값(상태값)을 출력하는 제1출력(1904)에 제2입력(1903)과 제2출력(1905)이 추가된다. 이때, 제2입력(1903)으로는 제1입력(1902)에 제공되는 순입력 데이터가 뉴런 j의 데이터일 때 뉴런의 번호 j가 입력된다. 그리고 활성화 함수 연산기(1900)는 활성화 함수를 계산하는 동안 뉴런의 번호를 임시 저장하고 계산이 완료되어 제1출력(1904)으로 새로운 속성값(상태값)을 출력할 때 뉴런의 번호를 제2출력(1905)으로 출력하며, 뉴런의 속성값(상태값)이 YN메모리(1901)에 저장될 때 YN메모리(1901)의 주소 입력에 공통으로 연결된 OutSel 입력(1906)으로는 뉴런의 번호(1906)가 제공된다.

[0171] 상기와 같이 데이터에 뉴런의 번호값을 함께 연결시켜 처리함으로써, 활성화 함수 연산기의 처리 타이밍이 가변적으로 변하더라도 바른 위치의 메모리에 결과값을 저장할 수 있다.

[0172] 한편, 일반적으로 입력과 출력을 포함하는 인공 신경망의 회상(recall) 모드 실행은 다음의 1 내지 3과 같은 과정으로 실행될 수 있다.

[0173] 1. 메모리 유닛(시냅스 유닛)의 Y메모리에 입력 뉴런의 값을 저장한다.

[0174] 2. 입력 뉴런을 제외한 뉴런에 대해 신경망 갱신 주기를 반복해서 적용한다.

[0175] 3. 실행을 멈추고 메모리 유닛(시냅스 유닛)의 Y메모리에서 출력 뉴런의 값을 추출한다.

[0176] 이러한 방법은 입력 데이터를 설정하거나 출력 뉴런의 값을 추출하기 위해 계산을 중단하여야 하므로, 시스템의 처리 속도를 저하시킬 수 있는 문제가 있다. 따라서 이와 다른 방식으로, 신경망을 실행하면서 동시에 입력 뉴

런에 입력 데이터를 설정하고 출력 뉴런의 값을 추출하는 방식을 위해, 도 20에 도시된 바와 같은 방법을 사용할 수 있다.

[0177] 도 20은 본 발명에 따른 신경망 컴퓨팅 장치의 다른 실시예 구성도이다.

[0178] 도 20에 도시된 바와 같이, 본 발명에 따른 신경망 컴퓨팅 장치는, 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛(2006), 각각 연결선 속성값과 뉴런 속성값을 출력하기 위한 복수 개의 메모리 유닛(2002), 복수 개의 메모리 유닛(2002)으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 계산하기 위한 하나의 계산 유닛(2003), 제어 유닛(2006)으로부터의 입력 데이터를 입력 뉴런에 제공하기 위한 입력 메모리(2000), 입력 메모리(2000)로부터의 입력 데이터 또는 계산 유닛(2003)으로부터의 새로운 뉴런 속성값을 제어 유닛(2006)의 제어에 따라 복수 개의 메모리 유닛(2002)으로 스위칭하기 위한 디지털 스위치(2004), 및 제어 유닛(2006)의 제어에 따라 모든 입력과 모든 출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 방식으로 구현되어, 계산 유닛(2003)으로부터의 새로운 뉴런 속성값이 제어 유닛(2006)으로 출력되도록 하기 위한 제1 및 제2 출력 메모리(2001, 2005)를 포함한다.

[0179] 제어 유닛(2006)이 실시간으로 입력 뉴런의 값을 신경망에 저장할 때, 하나의 신경망 갱신 주기는 입력 뉴런의 값을 저장하는 단계와 새로 계산된 뉴런의 값을 저장하는 단계로 구분되어 운용된다.

[0180] 1. 입력 뉴런의 값을 저장하는 단계 : 디지털 스위치(2004)가 입력 메모리(2000)의 출력과 연결되어 입력 메모리에 저장된 입력 뉴런의 속성값이 입력 메모리(2000)로부터 출력되어 모든 메모리 유닛(2002)의 YN메모리에 저장되도록 한다.

[0181] 2. 새로 계산된 뉴런의 값을 저장하는 단계 : 디지털 스위치(2004)가 계산 유닛(2003)의 출력과 연결되어 계산 유닛(2003)에서 출력되는 새로 계산된 뉴런의 속성값이 모든 메모리 유닛(2002)의 YN메모리에 저장되도록 한다.

[0182] 상기 2의 과정이 진행되는 동안에 제어 유닛(2006)은 입력 메모리(2000)에 다음 신경망 갱신 주기에서 사용될 입력 뉴런의 값을 저장할 수 있다.

[0183] 신경망 갱신 주기 내에서 상기 단계를 스케줄링하는 방법의 하나로서, 신경망 갱신 주기의 처음에 "1. 입력 뉴런의 값을 저장하는 단계"를 한꺼번에 수행하는 방법을 사용할 수 있다. 이 방법을 사용하면 "1. 입력 뉴런의 값을 저장하는 단계"는 YN메모리 이외에는 사용할 필요가 없으므로 도 21의 (b)에 도시된 바와 같이 신경망 갱신 주기의 시작을 다소 앞당길 수 있으며, 그에 따라 계산 효율을 다소 높일 수 있는 장점이 있다. 그러나 입력 뉴런의 수가 많으면 여전히 입력 과정이 신경망 컴퓨팅 장치의 성능에 영향을 줄 수 있다.

[0184] 신경망 갱신 주기 내에서 상기 단계를 스케줄링하는 다른 방법으로, 각 뉴런의 연결선 묶음의 수가 둘 이상일 때, 계산 유닛의 출력은 둘 이상의 클록 주기마다 발생하므로 출력이 발생하지 않는 클록 주기마다 끼워 넣기(interleaving) 방식으로 "1. 입력 뉴런의 값을 저장하는 단계"로 전환하여 입력 데이터를 하나씩 저장하는 방법을 사용할 수 있다. 이 경우 입력 뉴런의 값을 저장하는 과정이 신경망 컴퓨팅 장치의 성능에 전혀 영향을 주지 않는 장점이 있다.

[0185] 제어 유닛(2006)이 실시간으로 출력 뉴런의 값을 추출하는 방법으로서, 제1출력 메모리(2001)와 제2출력 메모리(2005)는 제어 신호에 따라 모든 입력과 모든 출력이 서로 바뀔 수 있는 이중 메모리 교체(SWAP) 방식으로 구현된다. 신경망 갱신 주기 내에서 새로 계산된 뉴런의 속성값은 제1출력 메모리(2001)에 저장되며, 하나의 신경망 갱신 주기가 끝나면 두 메모리(제1출력 메모리와 제2출력 메모리)는 서로 교체되어 이전 갱신 주기에 저장된 데이터가 제2출력 메모리 위치에 위치하게 된다. 제어 유닛(2006)은 제2출력 메모리(2005)에서 입력 뉴런을 제외한 모든 뉴런의 속성값을 읽어 들일 수 있으며, 이 중 출력 뉴런의 속성값을 취하여 신경망의 실시간 출력 값으로 활용할 수 있다. 이 방식은 제어 유닛이 출력 뉴런의 속성값에 신경망 컴퓨팅 장치의 실행 단계 및 타이밍에 구애받지 않고 언제나 접근할 수 있는 장점이 있다.

[0186] 도 21은 본 발명에 따른 신경망 갱신 주기를 설명하기 위한 일실시예 도면이다.

[0187] 도 21의 (a)는 입력 뉴런의 속성값을 메모리 유닛(2002)에 저장하는 과정을 신경망 갱신 주기의 맨 처음에 실행하지 않는 경우를 나타내고 있다. 이 경우 이전 신경망 갱신 주기(2100)가 완전히 완료되어야 새로운 신경망 갱신 주기(2101)를 실행할 수 있다. 한편, 도 21의 (b)는 입력 뉴런의 속성값을 메모리 유닛(2002)에 저장하는 과

정을 신경망 갱신 주기의 맨 처음에 실행하는 경우를 나타내고 있다. 입력 뉴런(2102)은 그 값을 계산하기 위해 계산 유닛을 사용할 필요가 없기 때문에 도 21의 (a)보다 신경망 갱신 주기 사이의 간격을 좁힐 수 있다. 도 21의 (c)는 입력 뉴런의 속성값을 메모리 유닛(2002)에 저장하는 과정을 계산 유닛에서 출력이 발생하지 않는 틈새 시간에 삽입(interleaving)하는 방법을 도시한 것이다. 이 경우 입력 뉴런의 수가 아무리 많아도 전체 처리 속도에 영향을 주지 않는 장점이 있다.

[0188] 이러한 신경망 컴퓨팅 장치는 가능한 최고의 처리 속도가 메모리 접근 주기 tmem에 의해 한정되는 단점이 있다. 일례로, 신경망 컴퓨팅 장치가 동시에 처리할 수 있는 연결선의 수 p = 1024, tmem = 10nS라 하면 신경망 컴퓨팅 장치의 최고 처리 속도는 102.4 GCPS이다.

[0189] 신경망 컴퓨팅 장치의 최대 속도를 더욱더 높이는 다른 방식의 하나로서, 여러 개의 신경망 컴퓨팅 장치를 서로 연결하는 방식을 사용할 수 있다.

[0190] 여러 개의 신경망 컴퓨팅 장치를 연결하여 전체의 성능을 높이는 일반적인 방법으로, 복수 개의 신경망 컴퓨팅 장치의 입출력을 서로 연결하여 네트워크를 형성하고, 하나의 신경망 컴퓨팅 시스템이 전체 신경망의 서브 네트워크를 처리하게 하면, 각각의 신경망 컴퓨팅 장치는 동시에 병렬로 실행되므로 신경망 컴퓨팅 장치의 처리 속도를 증가시킬 수 있다. 이러한 방식의 단점은 네트워크를 서브 네트워크로 분할하기 위하여 네트워크 구성에 제약이 있고 시스템 간 통신이 발생하여 오버헤드와 성능 저하가 수반된다는 점이다.

[0191] 이러한 방식에 대한 대안으로서, 복수 개의 신경망 컴퓨팅 장치를 도 23에 도시된 바와 같이 하나의 대규모 동기화 회로로 결합할 수 있다.

[0192] 도 23은 본 발명에 따른 신경망 컴퓨팅 시스템의 일실시예 구성도이다.

[0193] 도 23에 도시된 바와 같이, 본 발명에 따른 신경망 컴퓨팅 시스템은, 신경망 컴퓨팅 시스템을 제어하기 위한 제어 유닛(도면에 도시되지 않음, 도 2 및 후술되는 설명 참조), "각각 연결선 속성값과 뉴런 속성값을 출력하는 복수의 메모리 파트(2309)"를 포함하는 복수 개의 메모리 유닛(2300), 및 복수 개의 메모리 유닛(2300) 내의 상응하는 복수의 메모리 파트(2309)로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값을 각각 계산하여 상응하는 복수의 메모리 파트(2309) 각각으로 피드백시키기 위한 복수의 계산 유닛(2301)을 포함한다.

[0194] 여기서, 복수 개의 메모리 유닛(2300) 내의 복수의 메모리 파트(2309)와 복수의 계산 유닛(2301)은, 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0195] 그리고 각각의 메모리 파트는, 연결선 속성값을 저장하기 위한 W메모리(제1메모리, 2302), 뉴런의 고유번호를 저장하기 위한 M메모리(제2메모리, 2303), 뉴런 속성값을 저장하기 위한 YC메모리 그룹(제1메모리 그룹, 2304), 및 상응하는 계산 유닛(2301)에서 계산된 새로운 뉴런 속성값을 저장하기 위한 YN메모리 그룹(제2메모리 그룹, 2305)을 포함한다.

[0196] 이처럼, 도 1에서 전술한 바와 같은 신경망 컴퓨팅 장치 H개가 하나의 통합 시스템으로 결합될 때, 결합되기 전 h번째 신경망 컴퓨팅 장치의 i번째 메모리 유닛은 결합된 신경망 컴퓨팅 시스템에서 i번째 메모리 유닛의 h번째 메모리 파트가 되며, 그에 따라 다중 신경망 컴퓨팅 시스템에서 하나의 메모리 유닛(2300)은 H개의 메모리 파트로 이루어진다. 하나의 메모리 파트는 도 1에서 전술한 메모리 유닛의 구조와 기본적으로 같으나 다음의 1 및 2와 같은 차이가 있다.

[0197] 1. YC메모리 위치(2304)에는 H개의 YC메모리가 디코더 회로에 의해 H배 용량의 메모리로 결합한 형태로 위치한다.

[0198] 2. YN메모리 위치(2305)에는 H개의 YN메모리가 공통으로 묶인 형태로 위치한다.

[0199] H개의 신경망 컴퓨팅 장치로 이루어진 다중 신경망 컴퓨팅 시스템은 H개의 계산 유닛(2301)을 포함하며, h번째 계산 유닛은 각 메모리 유닛의 h번째 메모리 파트와 연결된다.

[0200] 이때, 제어 유닛이 메모리 유닛(2300) 내부의 각 메모리 파트의 각각의 메모리에 값을 저장하는 방식으로는, 다음과 같은 a 내지 j의 절차에 따라 각 메모리에 값을 저장할 수 있다.

[0201] a. 신경망 내 모든 뉴런을 H개의 균일한 뉴런 그룹으로 나누는 단계

[0202] b. 각 뉴런 그룹 내에서 가장 많은 수의 입력 연결선을 가진 뉴런의 입력 연결선의 수 Pmax를 찾는 단계

[0203] c. 메모리 유닛의 수를 p라 할 때, 신경망 내의 모든 뉴런이 $\lceil Pmax/p \rceil * p$ 개의 연결선을 갖도록 각각의 뉴런에 어떤 뉴런과 연결되어도 인접 뉴런에 영향을 미치지 않는 연결선 속성값을 갖는 가상의 연결선을 추가하는 단계

[0204] d. 뉴런 그룹 각각에 대해, 뉴런 그룹 내 모든 뉴런 각각에 임의의 순서로 번호를 부여하는 단계

[0205] e. 뉴런 그룹 각각에 대해, 뉴런 그룹 내 모든 뉴런 각각의 연결선을 p개씩 나누어 $\lceil Pmax/p \rceil$ 개의 묶음으로 분류하고 묶음 내의 연결선 각각에 임의의 순서로 1부터 시작하여 1씩 증가하는 번호 i를 부여하는 단계

[0206] f. 뉴런 그룹 각각에 대해, 뉴런 그룹 내 첫 번째 뉴런의 첫 번째 연결선 묶음부터 마지막 번째 뉴런의 마지막 연결선 묶음까지 순서대로 1부터 시작하여 1씩 증가하는 번호 k를 부여하는 단계

[0207] g. 메모리 유닛 중 i번째 메모리 유닛의 h번째 메모리 파트의 W메모리(제1메모리, 2302)의 j번째 주소에는 h번째 뉴런 그룹의 k번째 연결선 묶음의 i번째 연결선의 속성값을 저장하는 단계

[0208] h. 메모리 유닛 중 i번째 메모리 유닛의 h번째 M메모리(제2메모리, 2303)의 j번째 주소에는 h번째 뉴런 그룹의 k번째 연결선 묶음의 i번째 연결선에 연결된 뉴런의 고유 번호를 저장하는 단계

[0209] i. 모든 메모리 유닛 각각의 모든 메모리 파트의 YC메모리 그룹(제1메모리 그룹, 2304)을 구성하는 g번째 메모리의 j번째 주소에는 g번째 뉴런 그룹 내에서 j를 고유번호로 하는 뉴런의 속성값을 저장하는 단계

[0210] j. 모든 메모리 유닛 각각의 h번째 메모리 파트의 YN메모리 그룹(제2메모리 그룹, 2305)의 모든 메모리들의 j번째 주소에는 공통으로 h번째 뉴런 그룹 내에서 j를 고유번호로 하는 뉴런의 속성값을 저장하는 단계

[0211] a와 b를 임의의 정수라 할 때, 도 23의 각각의 메모리 유닛 내에서 YCa-b로 표기되는 모든 메모리 각각은 동일한 a, b인 YNa-b로 표기되는 메모리와 전술한 바와 같은 이중 메모리 교체(SWAP) 방식(505, 506)으로 구현된다(2306, 2307). 즉, 임의의 자연수 i,j에 대해 i번째 메모리 파트의 YC메모리 그룹(제1메모리 그룹)의 j번째 메모리와 j번째 메모리 파트의 YN메모리 그룹(제2메모리 그룹)의 i번째 메모리는, 제어 유닛의 제어에 따라 모든 입출력을 서로 바꾸어 연결하는 이중 메모리 교체 방식으로 구현된다.

[0212] 하나의 신경망 갱신 주기가 시작되면, 제어 유닛은 각 메모리 파트별로 InSel 입력(2308)에 1부터 시작해서 매 시스템 클록 주기마다 1씩 증가하는 연결선 묶음의 번호 값을 공급하며, 신경망 갱신 주기가 시작되고 나서 일정 시스템 클록 주기가 지난 후부터 메모리 유닛(2300)에서 h번째 메모리 파트의 메모리들(2302 내지 2305)은 h번째 뉴런 그룹 내의 연결선 묶음의 연결선의 속성값과 그 연결선에 연결된 뉴런의 속성값을 순차적으로 출력한다. 모든 메모리 유닛에서 h번째 메모리 파트의 출력은 h번째 계산 유닛의 입력으로 입력되며, h번째 뉴런 그룹의 연결선 묶음의 데이터를 구성한다. 이 연결선 묶음의 순서는 h번째 뉴런 그룹 내의 1번 뉴런의 첫 번째 연결선 묶음부터 마지막 연결선 묶음까지, 그리고 그 다음 뉴런의 첫 번째 연결선 묶음부터 마지막 연결선 묶음까지의 순서로 반복되고 마지막 뉴런의 마지막 연결선 묶음이 출력될 때까지 반복된다.

[0213] h번째 뉴런 그룹의 모든 뉴런이 각각 n개의 연결선 묶음을 가진 경우 신경망 갱신 주기가 시작되고 나서 일정 시스템 클록 주기가 지난 후부터 h번째 계산 유닛의 입력으로는 h번째 뉴런 그룹의 각 뉴런의 연결선 묶음의 데이터가 순차적으로 입력되고, 계산 유닛의 출력에는 매 n번의 시스템 클록 주기마다 새로운 뉴런의 속성값이 계산되어 출력된다. h번째 계산 유닛(2301)에서 계산된 h번째 뉴런 그룹 내의 새로운 뉴런의 값은 모든 메모리 유닛의 h번째 메모리 파트의 모든 YN메모리(2305)에 공통으로 저장된다. 이때, 저장될 주소와 쓰기 허용 신호(WE)는 제어 유닛(201)에 의해 각 메모리 파트별 OutSel 입력(2310)을 통해 제공된다.

[0214] 하나의 신경망 갱신 주기가 끝나면, 제어 유닛은 모든 YC메모리들과 각각 대응되는 YN메모리들을 서로 교체하여, 새로운 신경망 갱신 주기에는 이전에 따로 저장되었던 YN메모리들의 값을 하나의 대규모 YC메모리(2304)로 결합한다. 그 결과로 모든 메모리 파트의 대규모 YC메모리(2304)는 신경망 내 모든 뉴런의 속성값을 저장하게 된다.

[0215] 이러한 신경망 컴퓨팅 시스템의 경우 p를 메모리 유닛의 수, H를 신경망 컴퓨팅 장치의 수, tmem을 메모리 접근 시간이라 할 때, 신경망 컴퓨팅 시스템의 최고 처리 속도는 p*H/tmem CPS이다. 예를 들어, 하나의 신경망 컴퓨

팅 시스템이 동시에 처리하는 연결선의 수를 p = 1,024, tmem = 10nS, 신경망 컴퓨팅 장치의 수 H = 16인 경우 신경망 컴퓨팅 시스템의 최고 처리 속도는 1638.4 GCPS이다.

[0216] 상기와 같은 다중 신경망 컴퓨팅 시스템의 구성 방식은 신경망 네트워크 토폴로지의 제약이 전혀 없이 시스템의 규모를 무한정 확대할 수 있으며, 일반적으로 다중 시스템에서 발생하는 통신 오버헤드 없이 투입한 자원에 비례하여 성능을 증가시킬 수 있는 장점이 있다.

[0217] 한편, 지금까지는 회상 모드를 위한 시스템 구조에 대하여 설명하였다. 이하에서는 학습 모드를 지원하는 시스템 구조에 대해 설명하기로 한다.

[0218] 전술한 바와 같이 역전파 학습 알고리즘의 신경망 갱신 주기는 제1,2,3,4 서브 주기를 포함한다. 본 발명에서는 먼저 제1,2 서브 주기만을 수행하는 계산 구조와 제3,4 서브 주기를 수행하는 계산 구조를 별도로 설명한 후, 이 두 계산 구조를 하나로 통합하는 방식에 대해 설명하기로 한다.

[0219] 도 24는 본 발명에 따른 역전파 학습 알고리즘의 제1 서브 주기와 제2 서브 주기를 함께 실행하는 신경망 컴퓨팅 장치의 구조를 설명하기 위한 도면이다.

[0220] 도 24에 도시된 바와 같이, 역전파 학습 알고리즘의 제1 서브 주기와 제2 서브 주기를 함께 실행하는 신경망 컴퓨팅 장치는, 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛, 각각 연결선 속성값과 뉴런 오차값을 출력하기 위한 복수 개의 메모리 유닛(2400), 및 복수 개의 메모리 유닛(2400)으로부터 각각 입력되는 연결선 속성값과 뉴런 오차값을 이용(또는 시스템 외부의 지도자(supervisor)로부터 제어 유닛을 통해 제공되는 학습 데이터를 더 이용)하여 새로운 뉴런 오차값(다음 신경망 갱신 주기의 뉴런 오차값으로 사용됨)을 계산하여 복수 개의 메모리 유닛(2400) 각각으로 피드백시키기 위한 하나의 계산 유닛(2401)을 포함한다.

[0221] 이때, 복수 개의 메모리 유닛(2400)과 하나의 계산 유닛(2401)은, 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0222] 여기서, 각각 제어 유닛과 연결되는 InSel 입력(2408)과 OutSel 입력(2409)은 모든 메모리 유닛(2400)에 공통으로 연결된다. 그리고 모든 메모리 유닛(2400)의 출력은 각각 계산 유닛(2401)의 입력으로 연결된다. 그리고 계산 유닛(2401)의 출력은 모든 메모리 유닛(2400)의 입력에 공통으로 연결된다.

[0223] 그리고 각각의 메모리 유닛(2400)은, 연결선 속성값을 저장하기 위한 W메모리(제1메모리, 2403), 뉴런의 고유번호를 저장하기 위한 R2메모리(제2메모리, 2404), 뉴런 오차값을 저장하기 위한 EC메모리(제3메모리, 2405), 및 계산 유닛(2401)에서 계산된 새로운 뉴런 오차값을 저장하기 위한 EN메모리(제4메모리, 2406)를 포함한다.

[0224] 이때, 각각의 메모리 유닛(2400) 내에서 InSel 입력(2408)은 공통으로 W메모리(2403)의 주소 입력과 R2메모리(2404)의 주소 입력으로 연결된다. 그리고 R2메모리(2404)의 데이터 출력은 EC메모리(2405)의 주소 입력에 연결된다. 그리고 W메모리(2403)의 데이터 출력과 EC메모리(2405)의 데이터 출력은 각각 메모리 유닛(2400)의 출력이 되어 계산 유닛(2401)의 입력에 공통으로 연결된다. 그리고 계산 유닛(2401)의 출력은 메모리 유닛(2400)의 EN메모리(2406)의 데이터 입력과 연결되고, EN메모리(2406)의 주소 입력은 OutSel 입력(2409)과 연결된다. EC메모리(2405)와 EN메모리(2406)는 제어 유닛의 제어에 따라 모든 입력과 모든 출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 방식으로 구현된다.

[0225] 도 24에 도시된 신경망 컴퓨팅 장치는 전술한 도 1의 신경망 컴퓨팅 장치의 기본 구조와 유사하나, 다음과 같은 차이점을 갖는다.

[0226] ● 도 1의 M메모리 대신 R2메모리(2404)에는 역방향 네트워크에서 특정 연결선에 연결된 뉴런의 고유 번호가 저장된다.

[0227] ● 도 1의 YC메모리(104)와 YN메모리(105)를 대신하여 EC메모리(2405)와 EN메모리(2406)에는 뉴런의 속성값 대신 뉴런의 오차값이 저장된다.

[0228] ● 도 1의 입력 뉴런의 값을 저장하는 과정 대신, 계산 유닛에서 전체 뉴런 중 출력 뉴런(역방향 네트워크에서 입력 뉴런)은 계산 유닛의 학습 데이터(Teach) 입력(2407)을 통해 제공되는 해당 출력 뉴런의 학습 값과 그 뉴런의 속성값을 비교하여 오차값을 계산한다[수학식 2].

[0229] ● 도 1의 계산 유닛은 뉴런의 속성값을 계산하는 반면, 전체 뉴런 중 출력 뉴런 이외의 뉴런은 역방향 연결선을 통해 들어오는 오차값들을 인수로 오차값을 계산한다[수학식 3].

[0230] 하나의 신경망 갱신 주기 내에서 출력 뉴런의 오차를 계산하는 제1서브 주기가 시작되면, 제어 유닛에 의해 계산 유닛의 학습 데이터 입력(2407)을 통해 매 클록 주기마다 출력 뉴런의 학습 데이터가 입력된다. 계산 유닛이 상기 [수학식 2]를 적용하여 오차값을 계산하여 출력하면 복수 개의 메모리 유닛(2400) 각각으로 피드백되어 EN메모리(제4메모리, 2406)에 저장된다. 이 과정은 모든 출력 뉴런의 오차값이 계산될 때까지 반복된다.

[0231] 하나의 신경망 갱신 주기 내에서 출력 뉴런을 제외한 뉴런의 오차를 계산하는 제2서브 주기가 시작되면, 제어 유닛에 의해 InSel 입력에 1부터 시작해서 매 시스템 클록 주기마다 1씩 증가하는 연결선 묶음의 번호 값이 공급되며, 신경망 갱신 주기가 시작되고 나서 일정 시스템 클록 주기가 지난 후부터 메모리 유닛(2400)의 W메모리(2403)와 EC메모리(2405)의 출력을 통해 연결선 묶음의 연결선의 속성값과 그 연결선에 연결된 뉴런의 오차값이 순차적으로 출력된다. 모든 메모리 유닛(2400) 각각의 출력은 하나의 계산 유닛(2401)의 입력으로 입력되며 하나의 연결선 묶음의 데이터를 구성한다. 이 연결선 묶음의 순서는 첫 번째 뉴런의 첫 번째 연결선 묶음부터 마지막 연결선 묶음까지, 그리고 두 번째 뉴런의 첫 번째 연결선 묶음부터 마지막 연결선 묶음까지의 순서로 반복되고 마지막 뉴런의 마지막 연결선 묶음이 출력될 때까지 반복된다. 계산 유닛(2401)은 상기 [수학식 3]을 적용하여 각 뉴런의 각 연결선 묶음의 오차값의 총 합을 계산하고, 그 값은 복수 개의 메모리 유닛(2400) 각각으로 피드백되어 EN메모리(제4메모리, 2406)에 저장된다.

[0232] 도 25는 본 발명에 따른 학습 알고리즘을 실행하는 신경망 컴퓨팅 장치의 구조를 설명하기 위한 도면이다. 이 구조는 델타 학습법(Delta Learning Rule)이나 헤브의 법칙(Hebb's Rule)을 사용하는 신경망 모델에서도 동일하게 사용할 수 있다.

[0233] 도 25에 도시된 바와 같이, 학습 알고리즘을 실행하는 신경망 컴퓨팅 장치는, 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛, 각각 연결선 속성값과 뉴런 속성값을 계산 유닛(2501)으로 출력하고, 연결선 속성값과 뉴런 속성값과 계산 유닛(2501)으로부터의 학습 속성값을 이용하여 새로운 연결선 속성값(다음 신경망 갱신 주기의 연결선 속성값으로 이용됨)을 계산하기 위한 복수 개의 메모리 유닛(2500), 및 복수 개의 메모리 유닛(2500)으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값과 학습 속성값을 계산하기 위한 하나의 계산 유닛(2501)을 포함한다.

[0234] 이때, 복수 개의 메모리 유닛(2500)과 하나의 계산 유닛(2501)은, 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0235] 복수 개의 메모리 유닛(2500) 각각은, 연결선 속성값을 저장하기 위한 WC메모리(제1메모리, 2502), 뉴런의 고유번호를 저장하기 위한 M메모리(제2메모리, 2503), 뉴런 속성값을 저장하기 위한 YC메모리(제3메모리, 2504), 계산 유닛(2501)에서 계산된 새로운 뉴런 속성값을 저장하기 위한 YN메모리(제4메모리, 2506), WC메모리(2502)로부터의 연결선 속성값을 지연시키기 위한 제1선입선출 큐(제1지연수단, 2509), YC메모리(2504)로부터의 뉴런 속성값을 지연시키기 위한 제2선입선출 큐(제2지연수단, 2510), 및 계산 유닛(2501)으로부터의 학습 속성값과 제1선입선출 큐(2509)로부터의 연결선 속성값과 제2선입선출 큐(2510)로부터의 뉴런 속성값을 이용하여 새로운 연결선 속성값을 계산하기 위한 연결선 조정 모듈(2511), 및 연결선 조정 모듈(2511)에서 계산된 새로운 연결선 속성값을 저장하기 위한 WN메모리(제5메모리, 2505)를 포함한다.

[0236] 이때, 첫 번째 선입선출 큐(FIFO Queue, 2509)와 두 번째 선입선출 큐(FIFO Queue, 2510)는 연결선의 속성값(W)과 연결선에 연결된 뉴런의 속성값(Y)을 지연시키는 역할을 하며, 계산 유닛(2501)의 X출력으로는 뉴런의 학습에 필요한 학습 속성값을 출력한다. 특정 연결선이 뉴런 j의 연결선 중의 하나일 때, 그 연결선의 속성값(W)과 연결선에 연결된 뉴런의 속성값(Y)은 각각 선입선출 큐들(2509, 2510) 내에서 한 단계씩 진행되다가 계산 유닛(2501)의 X출력(즉, 뉴런 j의 학습에 필요한 속성값)이 레지스터(2515)에서 출력되는 타이밍에 각 선입선출 큐(2509, 2510)에서 출력되어 연결선 조정 모듈(2511)의 세 개의 입력에 제공된다. 연결선 조정 모듈(2511)은 이 세 개의 입력 데이터(W,Y,X)를 제공받아 다음 신경망 갱신 주기의 새로운 연결선의 속성값을 계산한 후 WN메모리(2505)에 저장한다.

[0237] YC메모리(2504)와 YN메모리(2506), WC메모리(2502)와 WN메모리(2505)는 각각 제어 유닛의 제어에 따라 모든 입력과 모든 출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 방식으로 구현된다. 이에 대한 다른 대안으로 YC메모리(2504)와 YN메모리(2506), WC메모리(2502)와 WN메모리(2505)는 각각 하나의 메모리를 사용하여 단일 메

모리 중복 저장 방법이나 단일 메모리 교체 방법으로 구현할 수도 있다.

[0238] 그리고 연결선 조정 모듈(2511)은 하기의 [수학식 7]과 같은 계산을 수행한다.

수학식 7

[0239] $$W_{ij}(T+1) = f(W_{ij}(T), Y_j(T), L_j)$$

[0240] 여기서, W_{ij}는 뉴런 j의 i번째 연결선의 속성값, Y_j는 뉴런 j의 속성값, L_j는 뉴런 j의 학습에 필요한 속성값을 나타낸다.

[0241] 상기 [수학식 7]은 상기 [수학식 5]를 포괄하는 보다 일반화된 함수로서, 상기 [수학식 5]와 대비하여 W_{ij}는 연결선의 가중치 값 w_{ij}, Y_j는 뉴런의 상태값 y_j, L_j는 $\eta \cdot \delta_j \cdot \frac{df(sum_j)}{dsum_j}$ 이고, 계산식은 하기의 [수학식 8]과 같다.

수학식 8

[0242] $$W_{ij}(T+1) = W_{ij}(T) + Y_j(T) * L_j$$

[0243] 상기 [수학식 8]을 계산하는 연결선 조정 모듈(2511)의 구조는 한 개의 곱셈기(2513)와 선입선출 큐(FIFO Queue, 2512), 및 하나의 덧셈기(2514)로 구현될 수 있다. 즉, 연결선 조정 모듈(2511)은, 제1선입선출 큐(2509)로부터의 연결선 속성값을 지연시키기 위한 제3선입선출 큐(제3지연수단, 2512), 계산 유닛(2501)으로부터의 학습 속성값과 제2선입선출 큐(2510)로부터의 뉴런 속성값에 대하여 곱셈 연산을 수행하기 위한 곱셈기(2513), 및 제3선입선출 큐(2512)로부터의 연결선 속성값과 곱셈기(2513)의 출력 값에 대하여 덧셈 연산을 수행하여 새로운 연결선 속성값을 출력하기 위한 덧셈기(2514)를 포함한다. 여기서, 선입선출 큐(FIFO Queue, 2512)는 곱셈기(2513)에서 계산하는 동안 $W_{ij}(T)$값을 지연시키는 역할을 한다.

[0244] 도 26은 본 발명에 따른 도 25의 신경망 컴퓨팅 장치에서의 데이터 흐름을 나타내는 도면이다.

[0245] 도 26에서, 뉴런 당 연결선 묶음의 수 = 2, 계산 유닛, 곱셈기, 덧셈기 각각의 파이프라인 단계 = 1로 가정하였다. 연결선 묶음 k는 뉴런 j의 첫 번째 연결선 묶음이다.

[0246] 도 25에서 설명한 신경망 컴퓨팅 장치의 대안으로서, 도 33에서 도시된 바와 같은 신경망 컴퓨팅 장치를 사용할 수 있다.

[0247] 도 33에 도시된 바와 같이, 학습 알고리즘을 실행하는 신경망 컴퓨팅 장치는, 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛, 각각 연결선 속성값과 뉴런의 속성값을 계산 유닛(3301)으로 출력하고, 연결선 속성값과 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하기 위한 복수 개의 메모리 유닛(3300), 복수 개의 메모리 유닛(3300)으로부터 각각 입력되는 연결선 속성값과 뉴런 속성값을 이용하여 새로운 뉴런 속성값과 학습 속성값을 계산하기 위한 하나의 계산 유닛(3301), 및 학습 속성값을 저장하기 위한 LC메모리(제1학습 속성값 메모리, 3321)와 LN메모리(제2학습 속성값 메모리, 3322)를 포함한다.

[0248] 이때, 복수 개의 메모리 유닛(3300)과 하나의 계산 유닛(3301)은, 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0249] 복수 개의 메모리 유닛(3300) 각각은, 연결선 속성값을 저장하기 위한 WC메모리(제1메모리, 3302), 뉴런의 고유번호를 저장하기 위한 M메모리(제2메모리, 3303), 뉴런 속성값을 저장하기 위한 YC메모리(제3메모리, 3304), 계

산 유닛(3301)에서 계산된 새로운 뉴런 속성값을 저장하기 위한 YN메모리(제4메모리, 3306), WC메모리(제1메모리, 3302)로부터의 연결선 속성값과 YC메모리(제3메모리, 3304)로부터의 입력 뉴런의 속성값 및 뉴런의 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하기 위한 연결선 조정 모듈(3311), 및 연결선 조정 모듈(3311)에서 계산된 새로운 연결선 속성값을 저장하기 위한 WN메모리(제5메모리, 3305)를 포함한다. 이때, 메모리 유닛 내의 메모리들은 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0250] 계산 유닛(3301)은 뉴런의 새로운 속성값을 계산하여 Y출력으로 출력하고 동시에 뉴런의 학습에 필요한 학습 속성값을 계산하여 X출력으로 출력한다. 계산 유닛(3301)의 X출력은 LN메모리(3322)와 연결되며, LN메모리(3322)는 새로 계산된 학습 속성값 Lj(T+1)을 저장하는 역할을 한다.

[0251] LC메모리(3321)는 이전 신경망 갱신 주기에서 계산된 뉴런의 학습 속성값 Lj(T)을 저장하며, 이 메모리의 데이터 출력은 모든 메모리 유닛(3300)의 연결선 조정 모듈(3311)의 X입력으로 연결된다. 메모리 유닛(3300)에서 출력되는 특정 연결선의 속성값 출력과 연결선에 연결된 뉴런의 속성값 출력은 각각 메모리 유닛(3300) 내에서 연결선 조정 모듈(3311)의 W입력과 Y입력으로 연결된다. 특정 시점에 특정 연결선의 정보를 출력할 때 그 연결선이 뉴런 j의 연결선 중의 하나라고 할 때 LC메모리(3321)로부터 뉴런 j의 학습 속성값이 동시에 제공된다. 연결선 조정 모듈(3311)은 이 세 개의 입력 데이터(W,Y,L)를 제공받아 다음 신경망 갱신 주기의 새로운 연결선 속성값을 계산한 후 WN메모리(3305)에 저장한다.

[0252] YC메모리(3304)와 YN메모리(3306), WC메모리(3302)와 WN메모리(3305), 및 LC메모리(3321)와 LN메모리(3322)는 각각 제어 유닛의 제어에 따라 모든 입력과 모든 출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 방식으로 구현된다. 이에 대한 다른 대안으로 YC메모리(3304)와 YN메모리(3306), WC메모리(3302)와 WN메모리(3305), 및 LC메모리(3321)와 LN메모리(3322)는 각각 하나의 메모리를 사용하여 단일 메모리 중복 방법이나 단일 메모리 교체 방법으로 구현할 수도 있다.

[0253] 그리고 연결선 조정 모듈(3311)에 대한 설명은 도 25에서 전술한 바와 유사하므로 여기서는 더 이상 설명하지 않기로 한다.

[0254] 도 27은 본 발명에 따른 하나의 신경망의 전체 또는 일부 네트워크에 대해 역방향 전파 주기와 순방향 전파 주기를 번갈아 실행하는 신경망 컴퓨팅 장치를 나타내는 도면이다. 이러한 본 발명의 구조는 역전파 학습 알고리즘 이외에도 심도 신뢰망(Deep Belief Network)과 같이 신경망의 부분망에 대해 역방향 전파 주기와 순방향 전파 주기를 번갈아 실행하는 신경망 모델의 학습 모드를 실행할 수 있다. 역전파 학습 알고리즘의 경우 제1,2 서브 주기가 역방향 전파 주기에 해당하고, 제3,4 서브 주기가 순방향 전파 주기에 해당한다.

[0255] 도 27에 도시된 바와 같이, 본 발명에 따른 하나의 신경망의 전체 또는 일부 네트워크에 대해 역방향 전파 주기와 순방향 전파 주기를 번갈아 실행하는 신경망 컴퓨팅 장치는, 신경망 컴퓨팅 장치를 제어하기 위한 제어 유닛, 각각 연결선 속성값, 순방향 뉴런 속성값 및 역방향 뉴런 속성값을 저장하고 출력하며, 새로운 연결선 속성값을 계산하기 위한 복수 개의 메모리 유닛(2700), 및 복수 개의 메모리 유닛(2700)으로부터 각각 입력되는 데이터를 바탕으로 새로운 순방향 뉴런 속성값과 역방향 뉴런 속성값을 계산하여 복수 개의 메모리 유닛(2700) 각각으로 피드백시키기 위한 하나의 계산 유닛(2701)을 포함한다. 여기서, 역전파 학습 알고리즘의 경우 뉴런 속성값이 순방향 뉴런 속성값에 해당하며, 뉴런 오차값은 역방향 뉴런 속성값에 해당한다. 도 27에서 새로운 연결 속성값을 계산하는 회로는, 도 25와 도 33의 설명을 토대로 당업자가 용이하게 유추할 수 있으므로 생략하기로 한다.

[0256] 이때, 복수 개의 메모리 유닛(2700)과 하나의 계산 유닛(2701)은, 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0257] 그리고 복수 개의 메모리 유닛(2700) 각각은, 역방향 네트워크에서 WC메모리(제2메모리, 2704)의 주소값을 저장하기 위한 R1메모리(제1메모리, 2705), 연결선 속성값을 저장하기 위한 WC메모리(제2메모리, 2704), 역방향 네트워크에서 뉴런의 고유번호를 저장하기 위한 R2메모리(제3메모리, 2706), 역방향 뉴런 속성값을 저장하기 위한 EC메모리(제4메모리, 2707), 계산 유닛(2701)에서 계산된 새로운 역방향 뉴런 속성값을 저장하기 위한 EN메모리(제5메모리, 2710), 순방향 네트워크에서 뉴런의 고유번호를 저장하기 위한 M메모리(제6메모리, 2702), 순방향 뉴런 속성값을 저장하기 위한 YC메모리(제7메모리, 2703), 계산 유닛(2701)에서 계산된 새로운 순방향 뉴런 속성값을 저장하기 위한 YN메모리(제8메모리, 2709), WC메모리(2704)의 입력을 선택하기 위한 제1디지털스위치(2712), EC메모리(2707) 또는 YC메모리(2703)의 출력을 계산 유닛(2701)으로 스위칭하기 위한 제2디지털스위치

(2713), 계산 유닛(2701)의 출력을 EN메모리(2710) 또는 YN메모리(2709)로 스위칭하기 위한 제3디지털스위치(2714), 및 아웃셀(OutSel) 입력을 EN메모리(2710) 또는 YN메모리(2709)로 스위칭하기 위한 제4디지털스위치(2715)를 포함한다.

[0258] 여기서, 역방향 전파 주기(역전파 학습 알고리즘의 경우 학습 모드의 제1,2 서브 주기)를 계산할 때에는 신경망 컴퓨팅 장치 내 N-비트 스위치들(2712 내지 2715)의 위치가 각각 하단부에 위치하고, 순방향 전파 주기(역전파 학습 알고리즘의 경우 제3,4 서브 주기)를 계산할 때에는 N-비트 스위치들(2712 내지 2715)의 위치가 각각 상단부에 위치하도록 제어 유닛에 의해 제어된다.

[0259] 그리고 YC메모리(2703)와 YN메모리(2709), EC메모리(2707)와 EN메모리(2710), WC메모리(2704)와 WN메모리(2708)는 각각 제어 유닛의 제어에 따라 모든 입력과 모든 출력을 서로 바꾸어 연결하는 이중 메모리 교체(SWAP) 방식으로 구현된다. 이에 대한 다른 대안으로 YC메모리(2703)와 YN메모리(2709), EC메모리(2707)와 EN메모리(2710), WC메모리(2704)와 WN메모리(2708)는 각각 하나의 메모리를 사용하여 단일 메모리 중복 저장 방법이나 단일 메모리 교체 방법으로 구현할 수도 있다.

[0260] 제어 유닛은 신경망 갱신 주기가 시작되면 N-비트 스위치들(2712 내지 2715)을 각각 하단부에 위치시키고 역방향 전파 주기를 수행한다. 그 다음에는 N-비트 스위치들(2712 내지 2715)을 상단부로 전환하고 순방향 전파 주기를 수행한다. 여기서, N-비트 스위치들(2712 내지 2715)이 하단부에 위치할 때 유효한 시스템의 구성도는 도 24와 같으나 InSel 입력과 WC메모리가 직접 연결되지 않고 R1메모리(2705)를 거치는 차이점이 있다. 그리고 N-비트 스위치들(2712 내지 2715)이 상단부에 위치할 때 유효한 시스템의 구성도는 도 25와 같다.

[0261] 역방향 전파 주기에 시스템이 동작하는 절차는 도 24에서 전술한 바와 기본적으로 같으나 R1메모리(2705)를 통해 간접적으로 매핑되어 WC메모리(2704)의 내용이 선택되는 차이점이 있다. 이는 WC메모리(2704)의 내용이 역방향 네트워크의 연결선 묶음의 순서와 일치하지 않아도 메모리 유닛 내에 있기만 하면 R1메모리(2705)를 통해 참조할 수 있는 특징을 추가로 갖는다. 그리고 순방향 전파 주기에 시스템이 동작하는 절차는 도 25 및 도 33의 설명에서 전술한 바와 같다.

[0262] 제어 유닛이 상기 메모리 유닛(2700) 내부의 각 메모리에 값을 저장하는 방식으로는, 다음과 같은 a 내지 q의 절차에 따라 각 메모리에 값을 저장할 수 있다.

[0263] a. 인공 신경망 순방향 네트워크에서 모든 연결선 각각의 양쪽 끝을 화살표가 시작되는 한쪽 끝과 화살표가 끝나는 다른 한쪽 끝으로 구분할 때, 모든 연결선 양 쪽에 다음의 1 내지 4와 같은 조건을 만족하는 번호를 부여하는 단계

[0264] 1. 모든 뉴런 각각에서 다른 뉴런으로 나가는 아웃바운드(outbound) 연결선들의 번호는 중복되지 않고 고유한 번호를 갖는 조건

[0265] 2. 모든 뉴런 각각에서 다른 뉴런으로부터 들어오는 인바운드(inbound) 연결선들의 번호는 중복되지 않고 고유한 번호를 갖는 조건

[0266] 3. 모든 연결선 양쪽의 번호는 같은 번호를 갖는 조건

[0267] 4. 상기 1 내지 3의 조건을 만족하되 가능한 한 낮은 숫자의 번호를 갖는 조건

[0268] b. 모든 뉴런의 아웃바운드(outbound) 또는 인바운드(inbound) 연결선에 부여된 번호 중 가장 큰 수 Pmax를 찾는 단계

[0269] c. 신경망의 순방향 네트워크 내부에 다른 뉴런과 연결선으로 연결되어도 영향을 미치지 않는 속성값을 갖는 한 개의 널(null) 뉴런을 추가하는 단계

[0270] d. 순방향 네트워크 내의 모든 뉴런 각각의 연결선에 할당된 번호를 유지한 채로 1부터 $\lceil Pmax/p \rceil * p$ 번까지 중 비어 있는 모든 번호에 새로운 연결선을 추가하여 총 $\lceil Pmax/p \rceil * p$ 개의 입력 연결선을 갖도록 확장하고, 추가된 연결선 각각은 어떤 뉴런과 연결되어도 영향을 미치지 않는 연결선 속성값을 갖거나 널(null) 뉴런과 연결되도록 설정하는 단계(여기서, p는 신경망 컴퓨팅 장치 내 메모리 유닛(2700)의 수)

[0271] e. 순방향 네트워크 내 모든 뉴런 각각에 임의의 순서로 번호를 부여하는 단계

[0272] f. 순방향 네트워크 내 모든 뉴런 각각의 연결선을 1번부터 순서대로 p개씩 나누어 $\lceil Pmax/p \rceil$개의 순방향

연결선 묶음으로 분류하고 묶음 내의 연결선 각각에 순서대로 1부터 시작하여 1씩 증가하는 새로운 번호 i를 부여하는 단계

[0273] g. 첫 번째 뉴런의 첫 번째 순방향 연결선 묶음부터 마지막 번째 뉴런의 마지막 순방향 연결선 묶음까지 순서대로 1부터 시작하여 1씩 증가하는 번호 k를 부여하는 단계

[0274] h. 메모리 유닛(2700) 중 i번째 메모리 유닛의 WC메모리(2704) 및 WN메모리(2708)의 k번째 주소에는 k번째 순방향 연결선 묶음의 i번째 연결선의 속성값의 초기값을 저장하는 단계

[0275] i. 메모리 유닛(2700) 중 i번째 메모리 유닛의 M메모리(2702)의 k번째 주소에는 k번째 순방향 연결선 묶음의 i번째 연결선에 연결된 뉴런의 고유 번호를 저장하는 단계

[0276] j. 모든 메모리 유닛 각각의 YC메모리(2703)와 YN메모리(2709) 각각의 j번째 주소에는 j를 고유번호로 하는 뉴런의 순방향 뉴런 속성값을 저장하는 단계

[0277] k. 신경망의 역방향 네트워크 내부에 다른 뉴런과 연결선으로 연결되어도 영향을 미치지 않는 속성값을 갖는 한 개의 널(null) 뉴런을 추가하는 단계

[0278] l. 역방향 네트워크 내의 모든 뉴런 각각의 연결선에 할당된 번호를 유지한 채로 1부터 $\lceil Pmax/p \rceil * p$ 번까지 중 비어 있는 모든 번호에 새로운 연결선을 추가하여 총 $\lceil Pmax/p \rceil * p$ 개의 입력 연결선을 갖도록 확장하고, 추가된 연결선 각각은 어떤 뉴런과 연결되어도 영향을 미치지 않는 연결선 속성값을 갖거나 널(null) 뉴런과 연결되도록 설정하는 단계(여기서, p는 신경망 컴퓨팅 장치 내 메모리 유닛(2700)의 수)

[0279] m. 역방향 네트워크 내 모든 뉴런 각각의 연결선을 1번부터 순서대로 p개씩 나누어 $\lceil Pmax/p \rceil$개의 역방향 연결선 묶음으로 분류하고 묶음 내의 연결선 각각에 순서대로 1부터 시작하여 1씩 증가하는 새로운 번호 i를 부여하는 단계

[0280] n. 첫 번째 뉴런의 첫 번째 역방향 연결선 묶음부터 마지막 번째 뉴런의 마지막 역방향 연결선 묶음까지 순서대로 1부터 시작하여 1씩 증가하는 번호 k를 부여하는 단계

[0281] o. 메모리 유닛(2700) 중 i번째 메모리 유닛의 R1메모리(2705)의 k번째 주소에는 k번째 역방향 연결선 묶음의 i번째 연결선이 메모리 유닛(2700) 중 i번째 메모리 유닛의 WC메모리(2704)에서 위치하는 위치 값을 저장하는 단계

[0282] p. 메모리 유닛(2700) 중 i번째 메모리 유닛의 R2메모리(2706)의 k번째 주소에는 k번째 역방향 연결선 묶음의 i번째 연결선에 연결된 뉴런의 고유 번호를 저장하는 단계

[0283] q. 모든 메모리 유닛 각각의 EC메모리(2707)와 EN메모리(2710) 각각의 j번째 주소에는 j를 고유번호로 하는 뉴런의 역방향 뉴런 속성값을 저장하는 단계

[0284] 상기 a단계를 만족하면, 순방향 신경망 네트워크의 특정 연결선이 i번째 메모리 유닛에 저장될 때, 역방향 네트워크에서 같은 연결선이 동일하게 i번째 메모리 유닛에 저장되는 특징을 갖게 된다. 따라서 전술한 바와 같이 역방향 전파 주기에 WC메모리(2704)를 순방향의 WC메모리와 같은 메모리를 사용해서 저장 순서가 역방향 네트워크의 연결선 묶음의 순서와 일치하지 않아도 R1메모리(2705)를 통해 참조할 수 있게 된다.

[0285] 상기 a단계를 해결하는 문제는 그래프 이론에서 모든 노드 각각에 붙은 호(edge)에 각기 다른 색깔을 칠하는 호의 색칠 문제(edge coloring problem)와 같은 문제이며, 각 뉴론에 연결된 연결선의 번호가 각기 다른 색깔을 대표한다고 가정하고 호의 색칠 알고리즘을 적용하여 해결할 수 있다.

[0286] 그래프 이론 중 하나인 바이징 이론(Vizing's theorem)과 쾌니히의 양분 그래프 이론(Konig's bipartite theorem)에 따르면 그래프 내 노드 중에서 가장 많은 호를 가진 노드의 호의 수를 n개라 할 때, 이 그래프에 호의 색칠 문제를 해결하기 위해 필요한 색깔의 수는 n개이다. 이는 상기 a단계에 호의 색칠 알고리즘을 적용하여 번호를 지정하면 전체 네트워크를 통틀어 연결선 번호는 전체 뉴런 중 가장 많은 수의 연결선을 가진 뉴런의 연결선의 수를 초과하지 않음을 의미한다.

[0287] 도 28은 본 발명에 따른 도 27의 신경망 컴퓨팅 장치를 간략화한 다른 계산 구조를 설명하기 위한 도면이다.

[0288] 도 27의 M메모리(2702), YC메모리(2703), YN메모리(2709) 각각을 메모리 영역 분할하여 각각 R2메모리(2706), EC메모리(2707), EN메모리(2710)의 용도로도 활용함으로써, 도 28에 도시된 바와 같이 단순화시킬 수 있다.

[0289] 그에 따라, 도 28의 M메모리(2802)의 메모리 영역의 절반은 도 27의 신경망 컴퓨팅 장치의 M메모리(2702)의 용도로 사용하고, 다른 절반은 도 27의 신경망 컴퓨팅 장치의 R2메모리(2706)의 용도로 사용한다. 그리고 도 28의 YEC메모리(2803)의 메모리 영역의 절반은 도 27의 신경망 컴퓨팅 장치의 YC메모리(2703)의 용도로 사용하고, 다른 절반은 도 27의 신경망 컴퓨팅 장치의 EC메모리(2707)의 용도로 사용한다. 그리고 도 28의 YEN메모리(2823)의 메모리 영역의 절반은 도 27의 신경망 컴퓨팅 장치의 YN메모리(2709)의 용도로 사용하고, 다른 절반은 도 27의 신경망 컴퓨팅 장치의 EN메모리(2710)의 용도로 사용한다.

[0290] 결과적으로, 도 28의 복수 개의 메모리 유닛(2800) 각각은, WC메모리(제2메모리, 2804)의 주소값을 저장하기 위한 R1메모리(제1메모리, 2805), 연결선 속성값을 저장하기 위한 WC메모리(제2메모리, 2804), 뉴런의 고유번호를 저장하기 위한 M메모리(제3메모리, 2802), 역방향 뉴런 속성값 또는 순방향 뉴런 속성값을 저장하기 위한 YEC메모리(제4메모리, 2803), 계산 유닛(2801)에서 계산된 새로운 역방향 뉴런 속성값 또는 순방향 뉴런 속성값을 저장하기 위한 YEN메모리(제5메모리, 2823), 및 WC메모리(2804)의 입력을 선택하기 위한 디지털스위치(2812)를 포함한다.

[0291] 도 29는 본 발명에 따른 도 27 및 도 28의 신경망 컴퓨팅 장치 중 계산 유닛(2701, 2801)의 상세 구성도이다.

[0292] 도 29에 도시된 바와 같이, 본 발명에 따른 계산 유닛(2701, 2801)은, 메모리 유닛(2700, 2800)의 수만큼의 곱셈기로 이루어져 각 메모리 유닛(2700, 2800)으로부터의 연결선 속성값과 순방향 뉴런 속성값 또는 연결선 속성값과 역방향 뉴런 속성값에 대해 곱셈 연산을 수행하기 위한 곱셈 연산부(2900), 트리 구조로 이루어져 곱셈 연산부(2900)로부터의 복수의 출력값에 대해 다단으로 덧셈 연산을 수행하기 위한 덧셈 연산부(2901), 덧셈 연산부(2901)로부터의 출력값을 누적 연산하기 위한 하나의 누산기(2902), 및 시스템 외부의 지도자(supervisor)로부터 제어 유닛을 통해 제공되는 학습 데이터(Teach)와 누산기(2902)로부터의 누적 출력값을 입력받아 다음 신경망 갱신 주기에 사용될 새로운 순방향 뉴런 속성값 또는 역방향 뉴런 속성값을 계산하기 위한 하나의 소마(soma) 처리기(2903)를 포함한다.

[0293] 여기서, 본 발명에 따른 계산 유닛(2701, 2801)은, 내부에 각 연산 단계 사이마다 레지스터를 더 포함할 수 있다. 이 경우 레지스터는 시스템 클록으로 동기화되고 각 연산 단계는 파이프라인 방식으로 처리된다.

[0294] 이처럼, 도 29의 계산 유닛의 구조는 전술한 도 8의 계산 유닛의 구조와 같으나, 활성화 함수 연산기 대신 소마 처리기(2903)가 사용되는 점이 다르다.

[0295] 상기 소마 처리기(2903)는 신경망 갱신 주기 내의 서브 주기에 따라 다음의 a 내지 c와 같은 다양한 계산을 수행한다.

[0296] a. 역전파 학습 알고리즘을 실행하는 경우, 오차 계산 서브 주기에 출력 뉴런을 계산하는 차례에는 학습 데이터(Teach) 입력(2904)으로부터 각 뉴런의 학습 값을 제공받아 상기 [수학식 3]을 적용하여 새로운 오차값을 계산하여 내부에 저장하고 Y출력에 출력한다. 즉, 출력 뉴런의 오차를 계산하는 주기에는 입력받은 학습 데이터(Teach)와 내부에 저장된 뉴런의 속성값의 차이로 오차값을 계산하여 내부에 저장하고 Y출력으로 출력한다. 역전파 학습 알고리즘이 아닌 경우 이 과정은 생략될 수 있다.

[0297] b. 역전파 학습 알고리즘을 실행하는 경우, 오차 계산 서브 주기에 출력 뉴런이 아닌 뉴런의 차례에는 누산기(2902)로부터 오차 입력의 총합을 받아서 내부에 저장하고 Y출력에 출력한다. 역전파 학습 알고리즘이 아닌 경우 해당 신경망 모델의 역방향 계산식에 따라 계산하여 Y출력에 출력한다.

[0298] c. 역전파 학습 알고리즘을 실행하는 경우, 뉴런 속성값 계산 서브 주기(회상 주기)에는 누산기(2902)로부터 뉴런의 순입력 값 NETk을 제공받아 활성화 함수를 적용하여 새로운 뉴런의 속성값(상태값)을 계산하여 내부에 저장하고 Y출력에 출력한다. 이와 함께 연결선 조정에 필요한 뉴런의 속성값 $\eta \cdot \delta_j \cdot \frac{df(sum_j)}{dsum_j}$ 을 계산하여 X출력에 출력한다. 역전파 학습 알고리즘이 아닌 경우 해당 신경망 모델의 순방향 계산식에 따라 계산하여 Y출력에 출력한다.

[0299] 도 30은 본 발명에 따른 도 29의 계산 유닛 중 소마 처리기(2903)의 상세 구성도이다.

[0300] 하나의 단위 소마 처리기는 도 30의 (a)와 같은 입출력을 가지며, 내부에 뉴런의 각종 속성 정보를 저장할 수 있다. 그리고 병렬 계산 라인 기법으로 처리량을 높인 소마 처리기는 도 30의 (b)와 같이 구현할 수 있다.

[0301] 도 30의 (a)에 도시된 바와 같이, 소마 처리기는, 제1입력(3000)을 통하여 누산기(2902)로부터 뉴런의 순 입력 또는 오차의 총 합을 입력받고, 제2입력(3001)을 통하여 출력 뉴런의 학습 데이터를 입력받으며, 제1출력(3003)을 통하여 새로 계산된 뉴런의 속성값 또는 오차값을 출력하고, 제2출력(3002)을 통하여 연결선 조정을 위한 뉴런의 속성값을 출력한다.

[0302] 도 30의 (b)에 도시된 바와 같이, 소마 처리기는, 각 입력에 대응되는 분배기(3004, 3005)와 복수 개의 소마 처리기(3006)와 각 출력에 대응되는 다중화기(3007, 3008)를 포함하고, 클록 주기로 인입되는 입력 데이터가 분배기(3004, 3005)에 의해 복수 개의 소마 처리기(3006)로 차례대로 분배되고 계산이 완료된 데이터는 다중화기(3007, 3008)에 의해 순서대로 다중화되어 클록 주기로 출력된다.

[0303] 한편, 전술한 회상 모드 전용 신경망 컴퓨팅 장치에서 실시간 입출력을 제공하는 방식의 확장으로 학습 모드에서도 입력 메모리를 통한 입력 뉴런의 값의 실시간 제공, 출력 메모리를 통한 출력 뉴런의 값의 실시간 인출과 함께 학습 데이터(Teach) 입력부(2723)에 메모리를 두어 실시간으로 학습 데이터를 제공할 수 있다.

[0304] 도 27 또는 도 28의 신경망 컴퓨팅 장치에 계산 유닛으로 도 29의 구조를 적용한 신경망 컴퓨팅 장치는 모든 학습의 전 과정이 파이프라인 회로에 의해 처리되며, 파이프라인 주기를 제한하는 요소는 메모리 접근 시간 tmem 뿐이다. 학습 모드에서 하나의 신경망 갱신 주기 내에는 2번의 내부 주기(제1,2 서브 주기와 제3,4 서브 주기)가 있으므로 최고 학습 처리 속도는 p/(2*tmem) CUPS이다.

[0305] 전술한 도 27의 학습을 지원하는 신경망 컴퓨팅 장치를 복수 개로 묶어서 복수 배의 성능을 갖도록 하는 신경망 컴퓨팅 시스템의 구조는 도 31과 같다.

[0306] 도 31은 본 발명에 따른 신경망 컴퓨팅 시스템의 다른 실시예 구성도이다.

[0307] 도 31에 도시된 바와 같이, 본 발명에 따른 신경망 컴퓨팅 시스템은, 신경망 컴퓨팅 시스템을 제어하기 위한 제어 유닛, "각각 연결선 속성값과 역방향 뉴런 속성값을 출력하거나, 각각 연결선 속성값과 순방향 뉴런 속성값을 출력하고 연결선 속성값과 순방향 뉴런 속성값과 학습 속성값을 이용하여 새로운 연결선 속성값을 계산하는 복수의 메모리 파트"를 포함하는 복수 개의 메모리 유닛(3100), 및 복수 개의 메모리 유닛(3100) 내의 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 역방향 뉴런 속성값을 이용하여 새로운 역방향 뉴런 속성값을 각각 계산하여 상응하는 복수의 메모리 파트 각각으로 피드백시키거나, 상응하는 복수의 메모리 파트로부터 각각 입력되는 연결선 속성값과 순방향 뉴런 속성값을 이용하여 새로운 순방향 뉴런 속성값과 학습 속성값을 각각 계산하여 상응하는 복수의 메모리 파트 각각으로 피드백시키기 위한 복수의 계산 유닛(3101)을 포함한다. 도 31에서 새로운 연결 속성값을 계산하는 회로는, 도 25와 도 33의 설명을 토대로 당업자가 용이하게 유추할 수 있으므로 생략하기로 한다.

[0308] 이때, 복수 개의 메모리 유닛(3100) 내의 복수의 메모리 파트와 복수의 계산 유닛(3101)은, 제어 유닛의 제어에 따라 하나의 시스템 클록에 동기화되어 파이프라인 방식으로 동작한다.

[0309] 그리고 각각의 메모리 파트는, WC메모리(제2메모리, 3102)의 주소값을 저장하기 위한 R1메모리(제1메모리, 3103), 연결선 속성값을 저장하기 위한 WC메모리(제2메모리, 3102), 뉴런의 고유번호를 저장하기 위한 R2메모리(제3메모리, 3115), 역방향 뉴런 속성값을 저장하기 위한 EC메모리 그룹(제1메모리 그룹, 3106), 계산 유닛(3101)에서 계산된 새로운 역방향 뉴런 속성값을 저장하기 위한 EN메모리 그룹(제2메모리 그룹, 3108), 뉴런의 고유번호를 저장하기 위한 M메모리(제4메모리, 3104), 순방향 뉴런 속성값을 저장하기 YC메모리 그룹(제3메모리 그룹, 3105), 계산 유닛(3101)에서 계산된 새로운 순방향 뉴런 속성값을 저장하기 위한 YN메모리 그룹(제4메모리 그룹, 3107), WC메모리(3102)의 입력을 선택하기 위한 제1디지털스위치, EC메모리 그룹(3106) 또는 YC메모리 그룹(3105)의 출력을 계산 유닛(3101)으로 스위칭하기 위한 제2디지털스위치, 계산 유닛(3101)의 출력을 EN메모리 그룹(3108) 또는 YN메모리 그룹(3107)으로 스위칭하기 위한 제3디지털스위치, 및 아웃셀(OutSel) 입력을 EN메모리 그룹(3108) 또는 YN메모리 그룹(3107)으로 스위칭하기 위한 제4디지털스위치를 포함한다.

[0310] 이처럼, 도 27 또는 도 28에서 전술한 바와 같은 학습을 지원하는 n개의 신경망 컴퓨팅 장치가 하나의 통합 시스템으로 결합될 때 메모리 유닛(3100)은 개별 신경망 컴퓨팅 장치의 메모리 유닛(2700) n개가 하나로 합해진 형태의 회로로 구현되고, 각각의 YC메모리 위치(3105)에는 n개의 YC메모리가 디코더 회로에 의해 n배의 대규모 메모리 용량으로 결합한 형태로 구현되며, YN메모리 위치(3107)에는 n개의 YN메모리가 공통으로 묶인 형태로 구현된다. 또한, 각각의 EC메모리 위치(3106)에는 n개의 EC메모리가 디코더 회로에 의해 n배의 대규모 메모리 용량으로 결합한 형태로 구현되고, EN메모리 위치(3108)에는 n개의 EN메모리가 공통으로 묶인 형태로 구현된다.

[0311] n개의 신경망 컴퓨팅 장치로 이루어진 신경망 컴퓨팅 시스템에서 h번째 신경망 컴퓨팅 장치는 전체 시스템의 뉴런을 n개의 그룹으로 나누었을 때 h번째 그룹의 뉴런을 처리한다.

[0312] 도 31에서 a와 b를 임의의 정수라 할 때, 각각의 메모리 유닛 내에서 YCa-b로 표기되는 모든 메모리 각각은 동일한 a, b인 YNa-b로 표기되는 메모리와 전술한 바와 같은 이중 메모리 교체(SWAP) 방식(3111, 3112)으로 구현된다. 또한, ECa-b로 표기되는 모든 메모리 각각은 동일한 a, b인 ENa-b로 표기되는 메모리와 전술한 바와 같은 이중 메모리 교체(SWAP) 방식(3113, 3114)으로 구현된다.

[0313] 이러한 시스템의 동작 절차는 도 23의 시스템 동작 절차와 비교하여 학습 절차를 지원하는 차이가 있으나 대체로 유사하며 당업자의 수준에서 유추가 가능하므로 더 이상 자세한 설명은 생략하기로 한다.

[0314] 이러한 경우 p를 메모리 유닛의 수, h를 신경망 컴퓨팅 장치의 수, tmem을 메모리 접근 시간이라 할 때, 신경망 컴퓨팅 시스템의 최고 처리 속도는 p*h/tmem CUPS이다.

[0315] 한편, 전술한 바와 같은 본 발명에 따른 신경망 컴퓨팅 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 상기 매체는 프로그램 명령, 데이터 구조 등을 지정하는 신호를 전송하는 반송파를 포함하는 광 또는 금속선, 도파관 등의 전송 매체일 수도 있다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용하여 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0316] 이상과 같이 본 발명은 비록 한정된 실시 예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시 예에 한정되는 것은 아니며, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 다양한 치환, 변형 및 변경이 가능하다.

[0317] 그러므로 본 발명의 범위는 설명된 실시 예에 국한되어 정해져서는 아니 되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

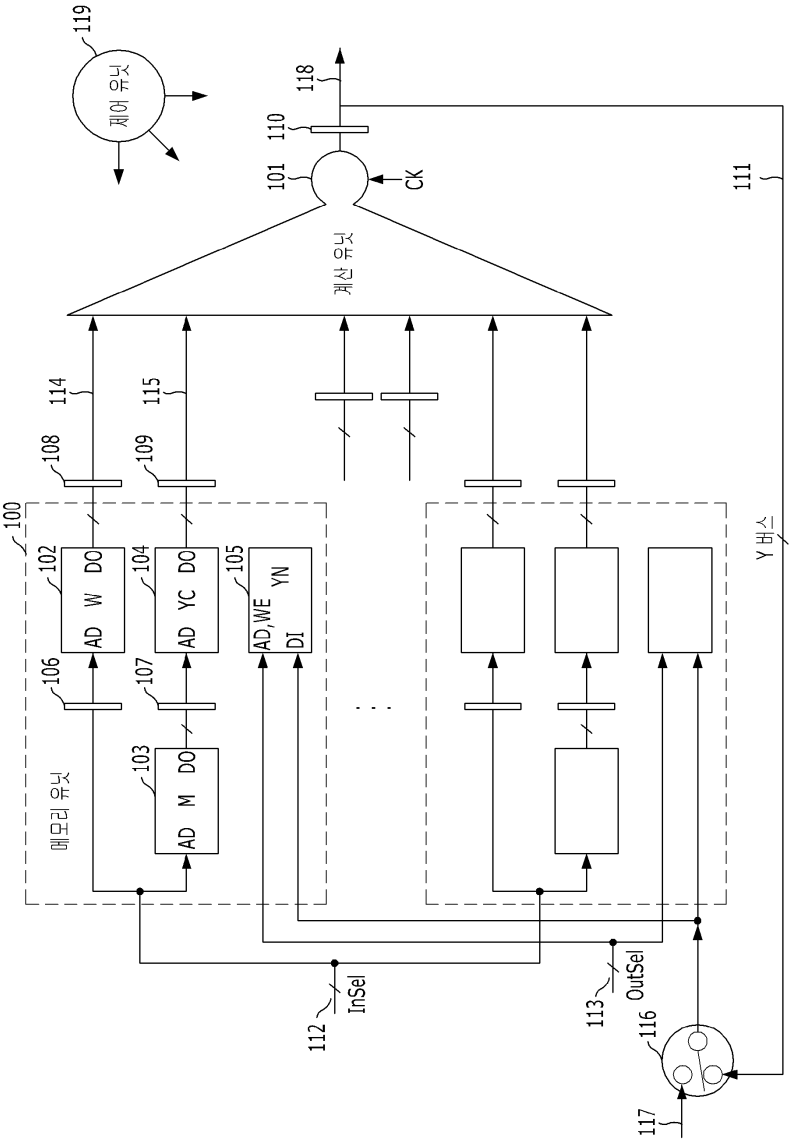
부호의 설명

[0318]

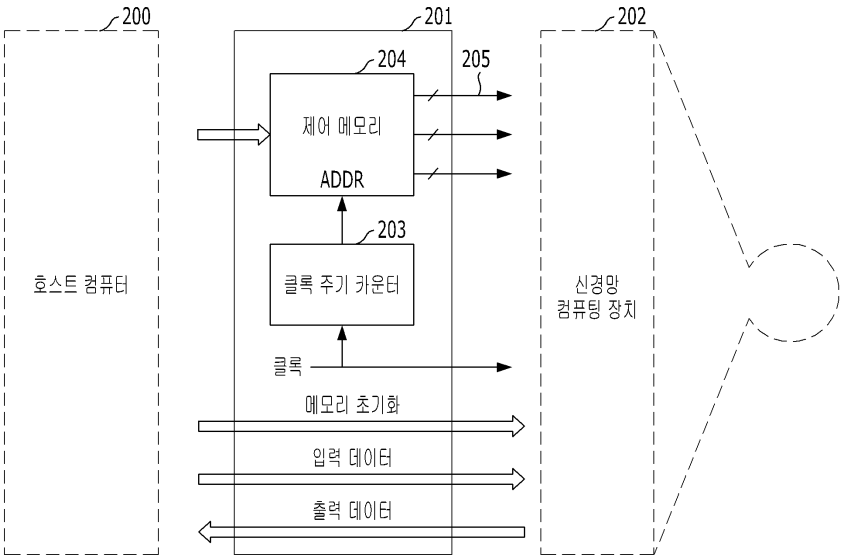
100 : 메모리 유닛	101 : 계산 유닛
102 : W메모리(제1메모리)	103 : M메모리(제2메모리)
104 : YC메모리(제3메모리)	105 : YN메모리(제4메모리)
106 : 제1레지스터	107 : 제2레지스터
108, 109 : 제3레지스터	110 : 제4레지스터
116 : 디지털 스위치	119 : 제어 유닛

도면

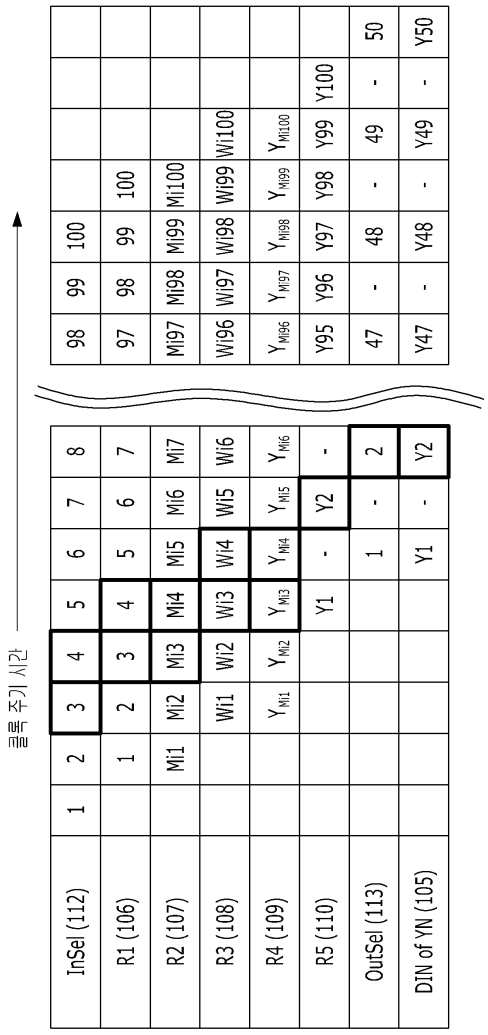
도면1


119
제어 유닛
118
110
101
CK
111
계산 유닛
114
115
108
109
100
102
AD W DO
104
AD YC DO
105
AD,WE YN
DI
106
107
103
AD M DO
메모리 유닛
Y 버스
112
InSel
113
OutSel
116
117

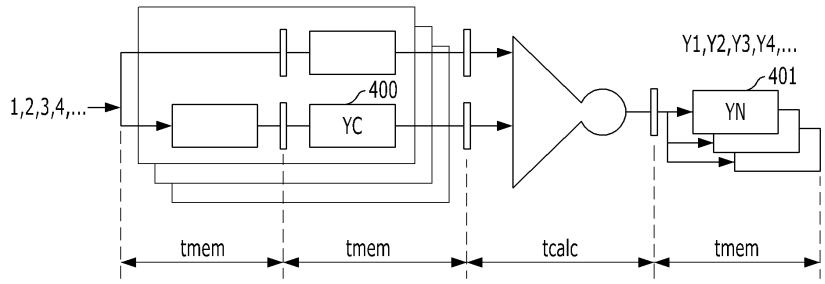
도면2


200
201
202
204
205
제어 메모리
ADDR
203
클록 주기 카운터
호스트 컴퓨터
신경망
컴퓨팅 장치
클록
메모리 초기화
입력 데이터
출력 데이터

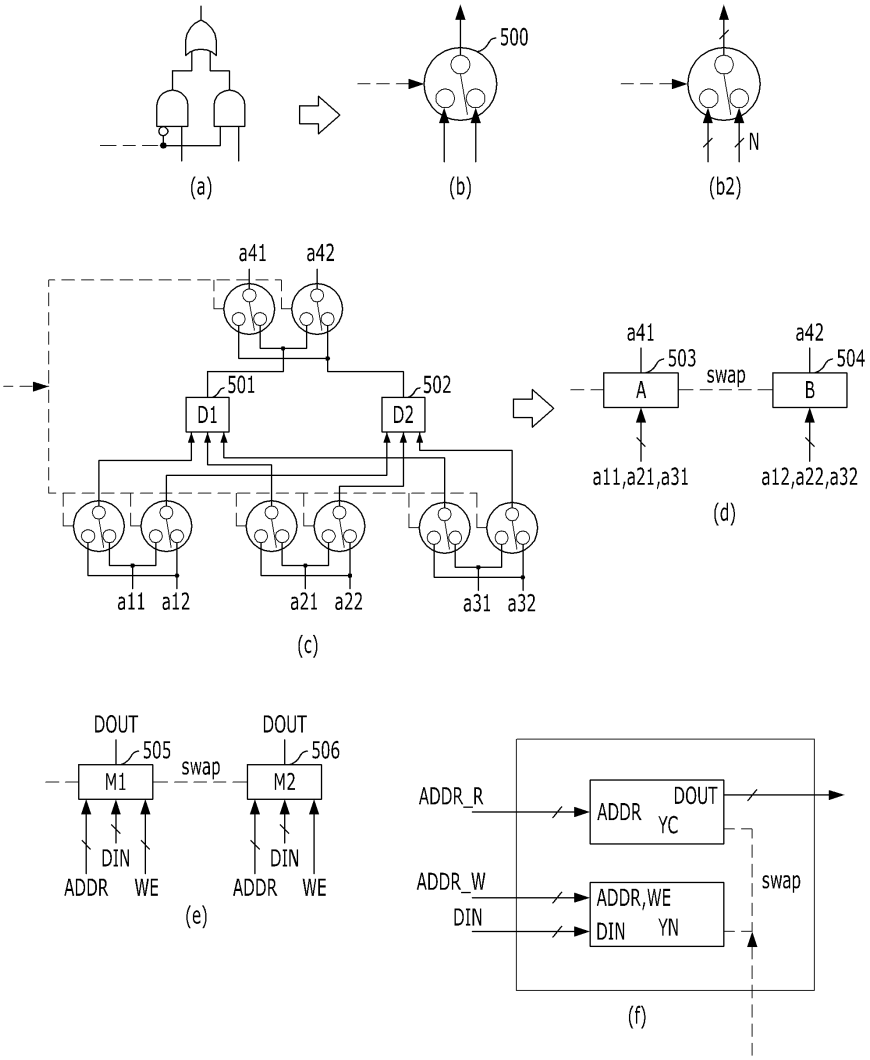
도면3


클록 주기 시간
InSel (112)
R1 (106)
R2 (107)
R3 (108)
R4 (109)
R5 (110)
OutSel (113)
DIN of YN (105)

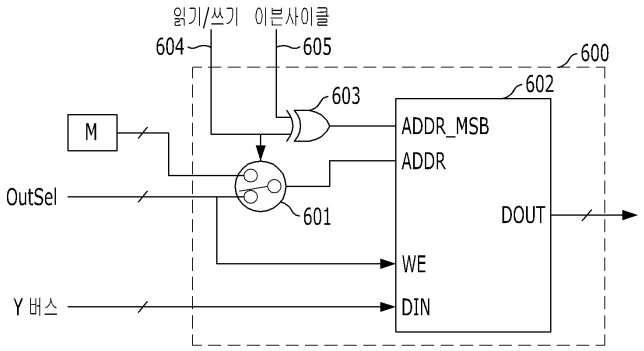
도면4


1,2,3,4,...
400
YC
Y1,Y2,Y3,Y4,...
401
YN
tmem
tmem
tcalc
tmem

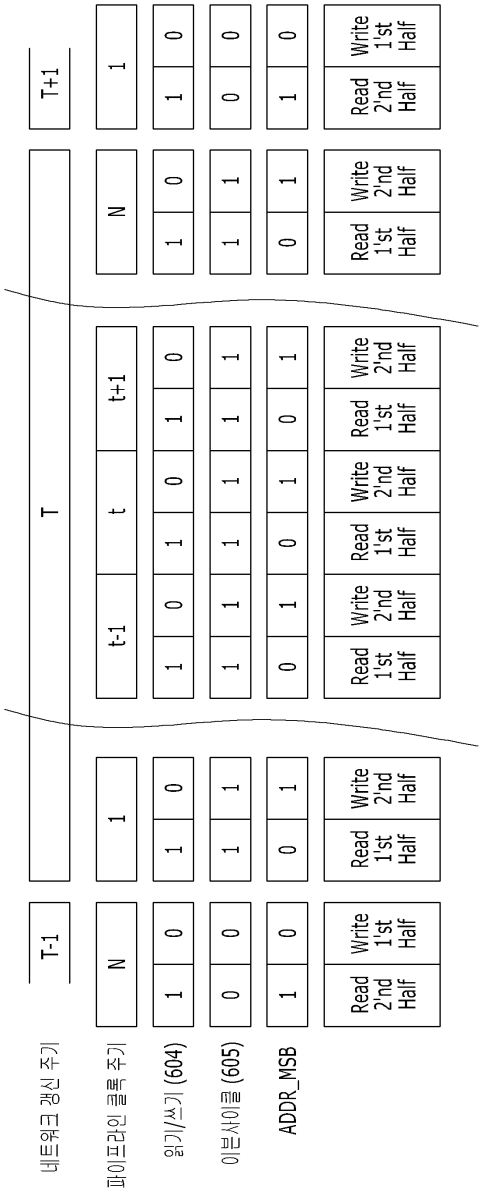
도면5


(a)
500
(b)
N
(b2)
a41
a42
501
502
D1
D2
a11
a12
a21
a22
a31
a32
(c)
a41
503
A
swap
a42
504
B
a11,a21,a31
a12,a22,a32
(d)
DOUT
505
M1
swap
DOUT
506
M2
DIN
ADDR
WE
DIN
ADDR
WE
(e)
ADDR_R
ADDR
DOUT
YC
ADDR_W
DIN
ADDR,WE
DIN
YN
swap
(f)

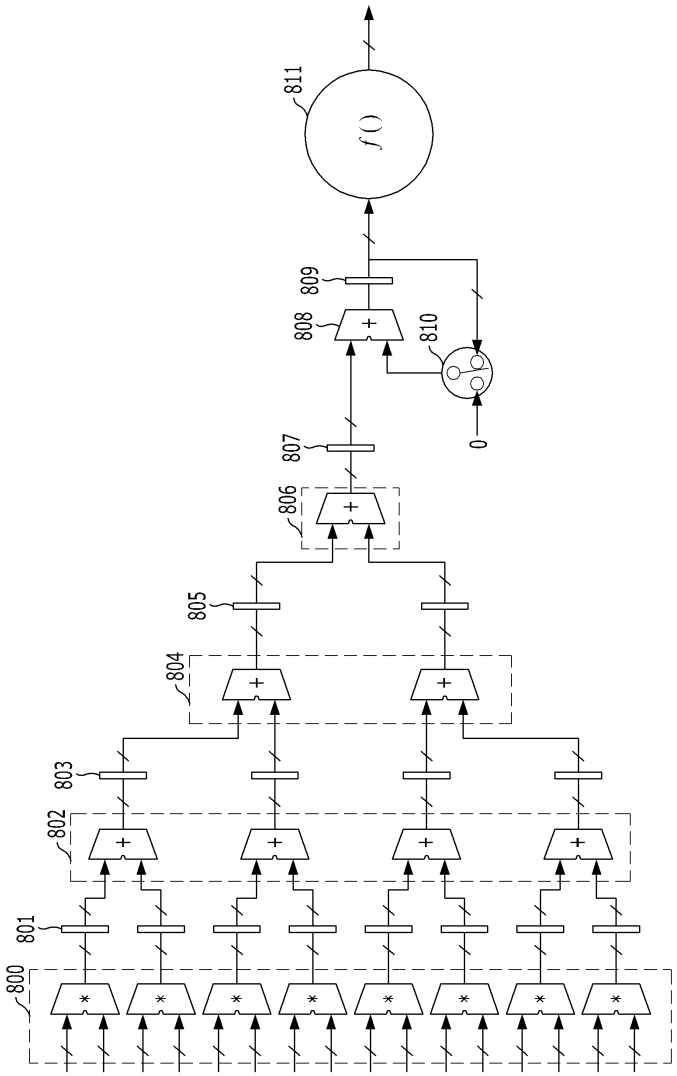
도면6


읽기/쓰기
이븐사이클
604
605
600
603
602
M
ADDR_MSB
ADDR
OutSel
601
DOUT
WE
Y 버스
DIN

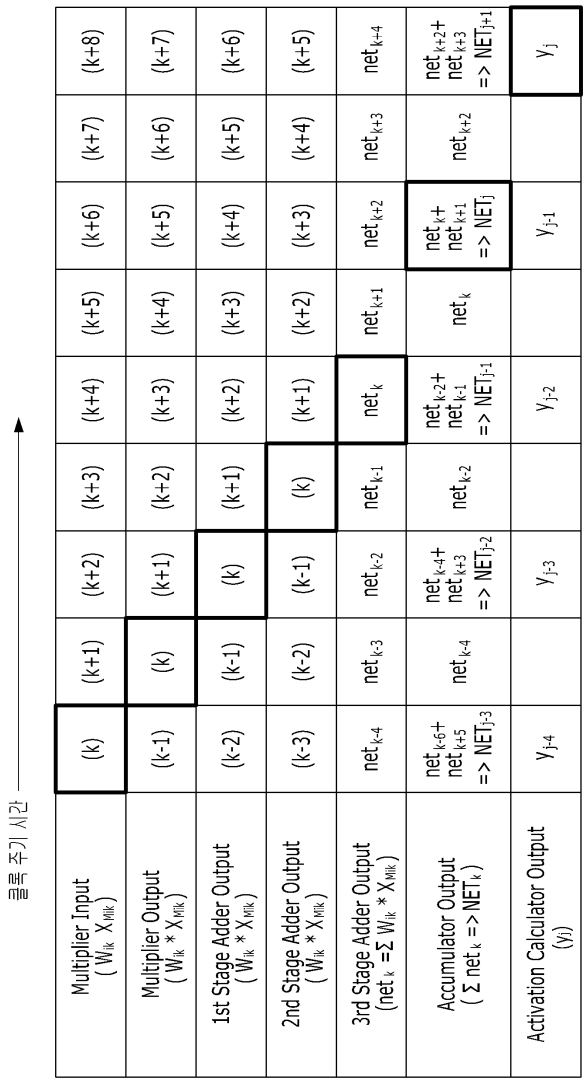
도면7


네트워크 갱신 주기
파이프라인 클록 주기
읽기/쓰기 (604)
이븐사이클 (605)
ADDR_MSB
T-1
T
T+1
N
t-1
t
t+1
Read 2'nd Half
Write 1'st Half
Read 1'st Half
Write 2'nd Half

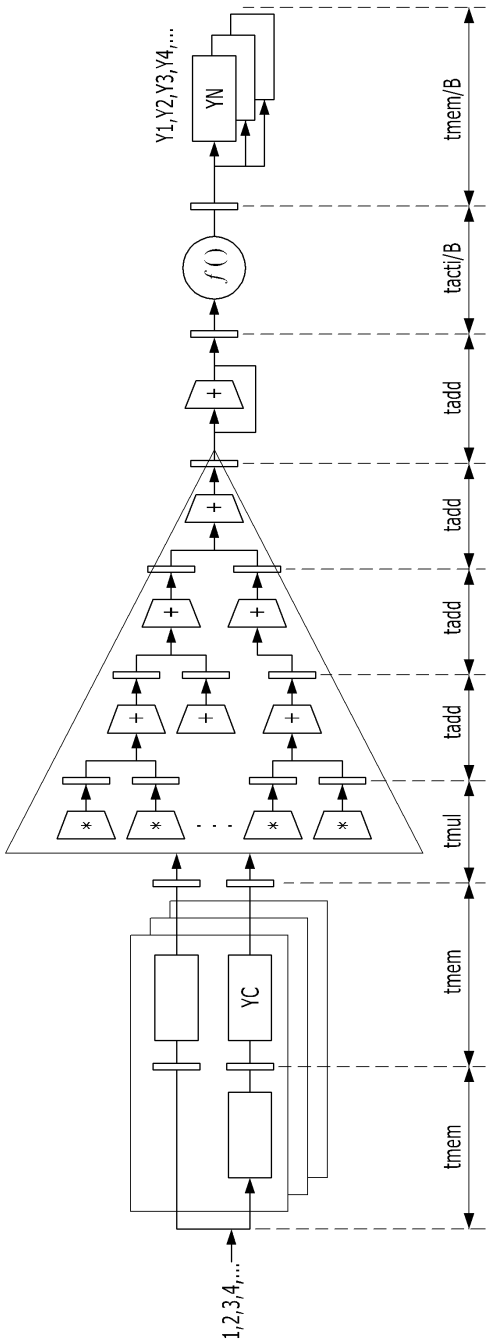
도면8



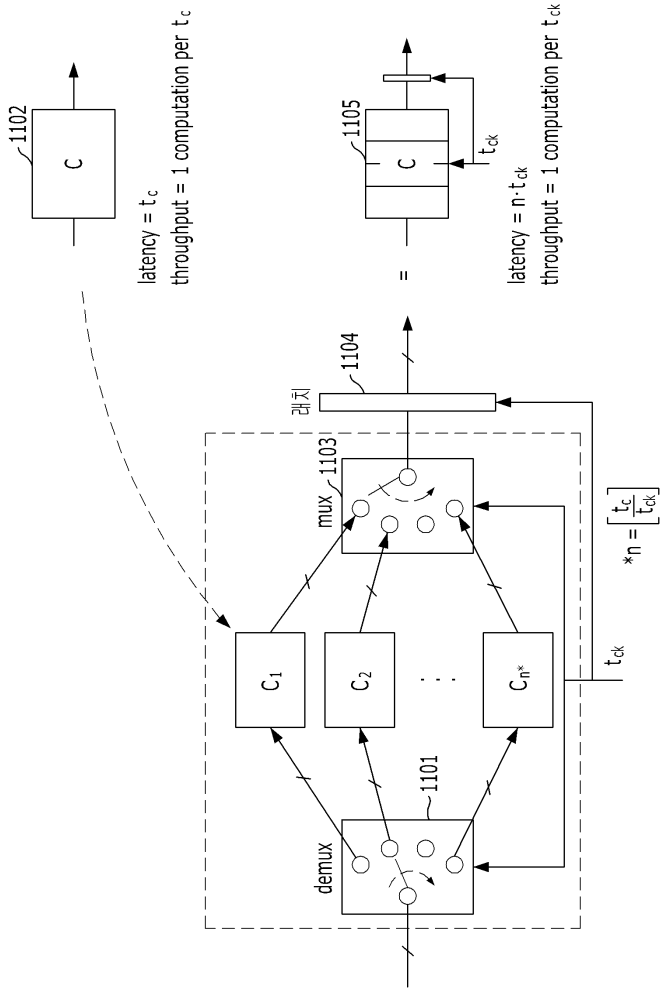
도면9


클록 주기 시간
Multiplier Input (W_{ik} X_{Mik})
Multiplier Output (W_{ik} * X_{Mik})
1st Stage Adder Output (W_{ik} * X_{Mik})
2nd Stage Adder Output (W_{ik} * X_{Mik})
3rd Stage Adder Output (net_k = Σ W_{ik} * X_{Mik})
Accumulator Output (Σ net_k => NET_k)
Activation Calculator Output (y_j)
(k) (k+1) (k+2) (k+3) (k+4) (k+5) (k+6) (k+7) (k+8)
(k-1) (k) (k+1) (k+2) (k+3) (k+4) (k+5) (k+6) (k+7)
(k-2) (k-1) (k) (k+1) (k+2) (k+3) (k+4) (k+5) (k+6)
(k-3) (k-2) (k-1) (k) (k+1) (k+2) (k+3) (k+4) (k+5)
net_{k-4} net_{k-3} net_{k-2} net_{k-1} net_k net_{k+1} net_{k+2} net_{k+3} net_{k+4}
net_{k-6} + net_{k+5} => NET_{j-3} net_{k-4} net_{k-4} + net_{k+3} => NET_{j-2} net_{k-2} net_{k-2} + net_{k-1} => NET_{j-1} net_k net_k + net_{k+1} => NET_j net_{k+2} net_{k+2} + net_{k+3} => NET_{j+1}
y_{j-4} y_{j-3} y_{j-2} y_{j-1} y_j

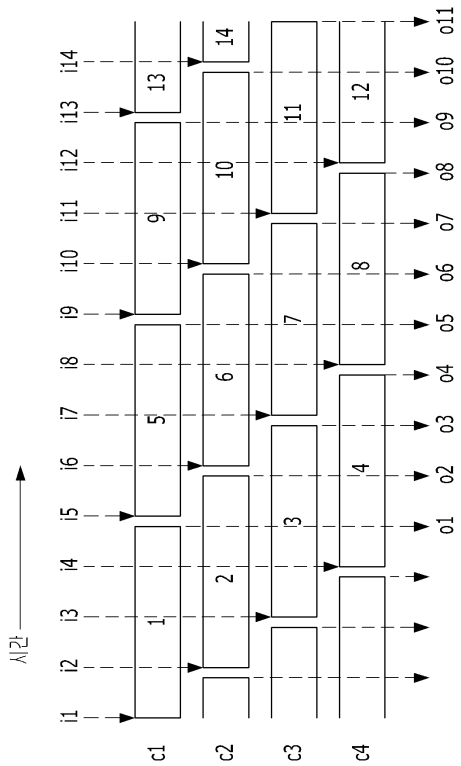
도면10


Y1,Y2,Y3,Y4,...
YN
f ( )
+
*
YC
1,2,3,4,...
tmem/B
tacti/B
tadd
tadd
tadd
tadd
tmul
tmem
tmem

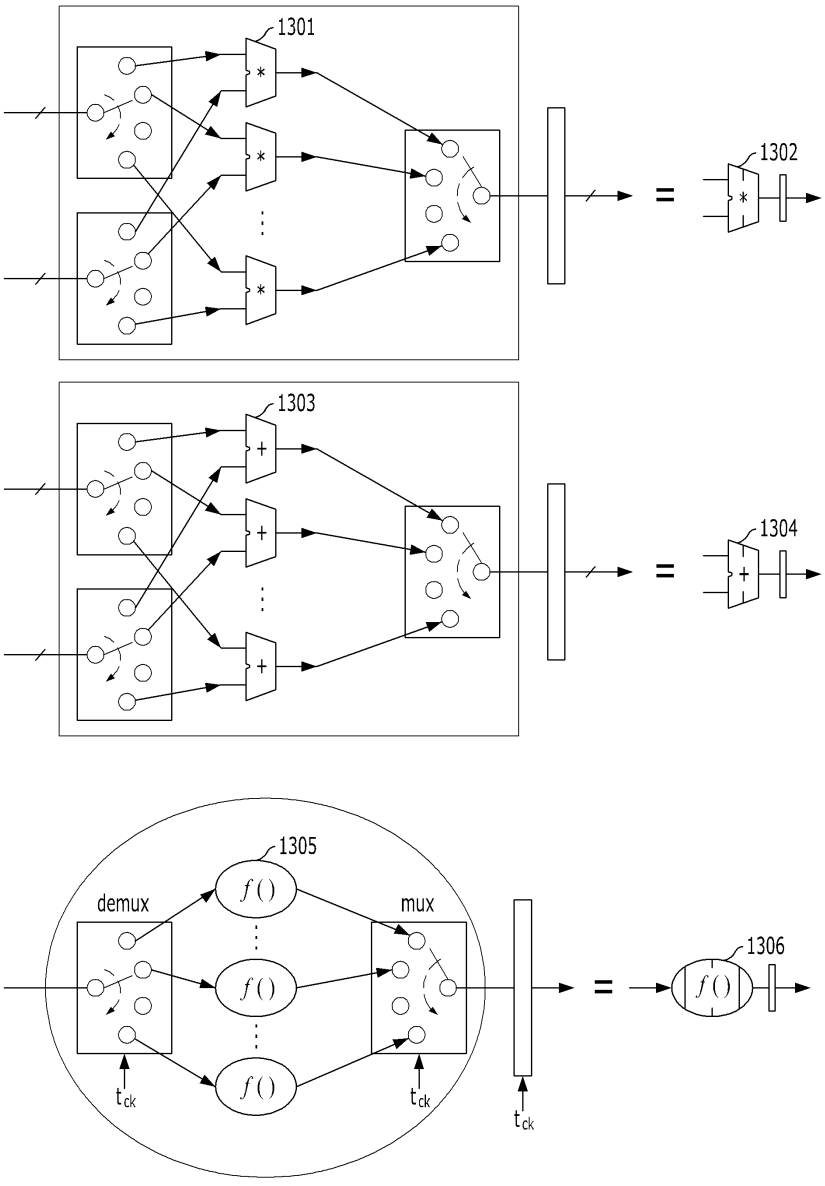
도면11


1102
C
latency = t_c
throughput = 1 computation per t_c
=
1105
C
t_{ck}
latency = $n \cdot t_{ck}$
throughput = 1 computation per t_{ck}
1104
래치
1103
mux
C_1
C_2
C_{n^*}
1101
demux
t_{ck}
$*n = \left\lceil \frac{t_c}{t_{ck}} \right\rceil$

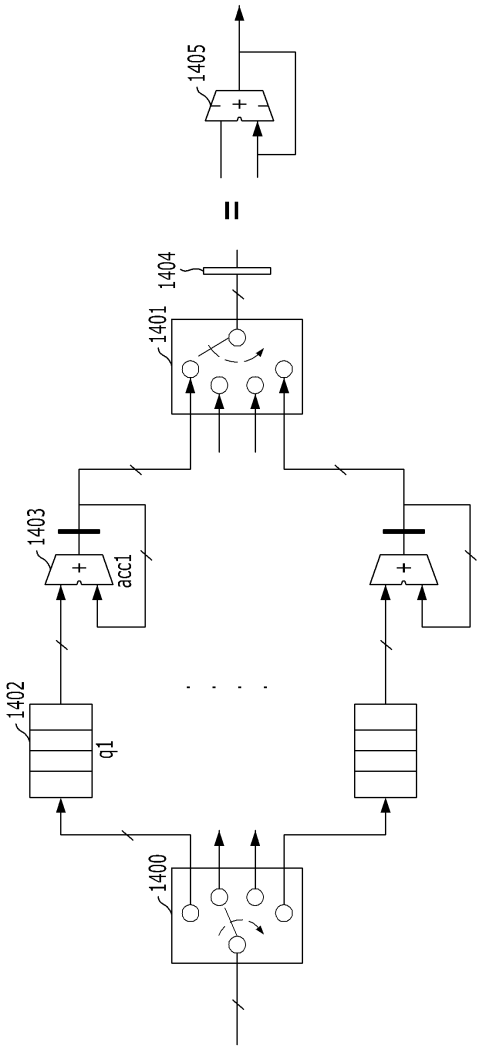
도면12


시간
i1
i2
i3
i4
i5
i6
i7
i8
i9
i10
i11
i12
i13
i14
o1
o2
o3
o4
o5
o6
o7
o8
o9
o10
o11
c1
c2
c3
c4
1
2
3
4
5
6
7
8
9
10
11
12
13
14

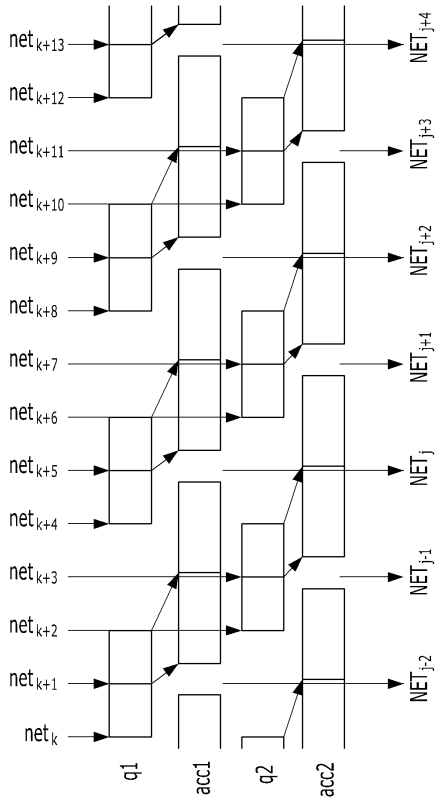
도면13



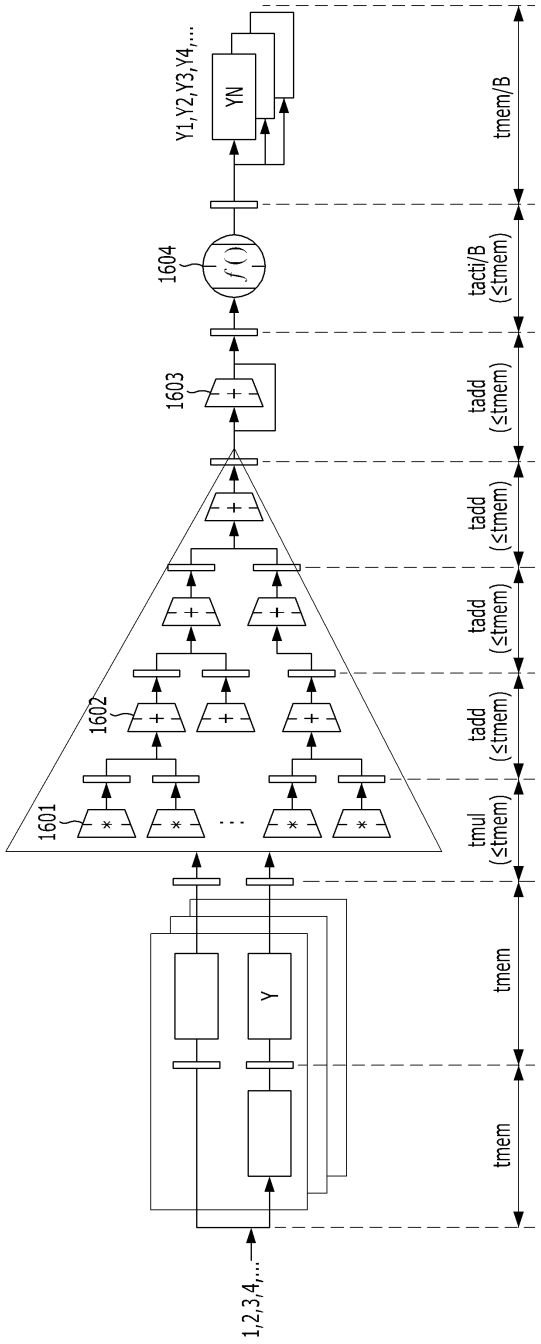
도면14



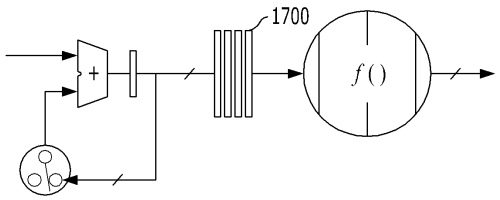
도면15


net k+13
net k+12
net k+11
net k+10
net k+9
net k+8
net k+7
net k+6
net k+5
net k+4
net k+3
net k+2
net k+1
net k
NET j+4
NET j+3
NET j+2
NET j+1
NET j
NET j-1
NET j-2
q1
acc1
q2
acc2

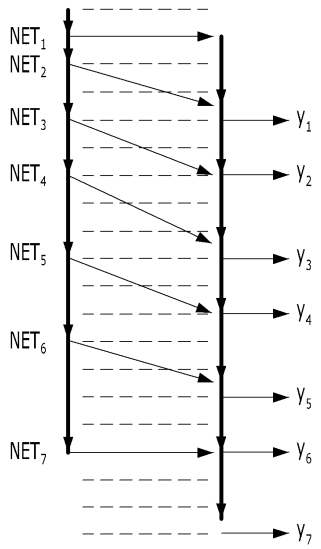
도면16


Y1,Y2,Y3,Y4,...
YN
tmem/B
1604
f ( )
tacti/B (≤tmem)
1603
tadd (≤tmem)
tadd (≤tmem)
tadd (≤tmem)
1602
tadd (≤tmem)
1601
tmul (≤tmem)
tmem
Y
tmem
1,2,3,4,...

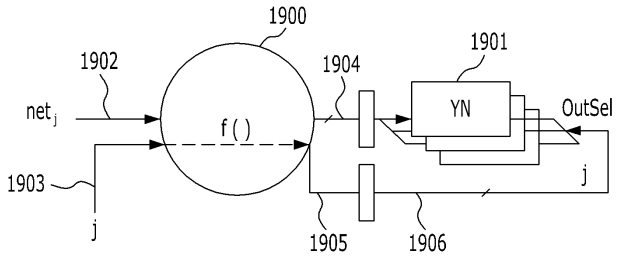
도면17


1700
f ( )

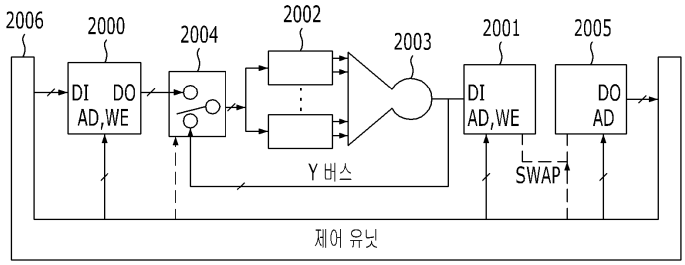
도면18



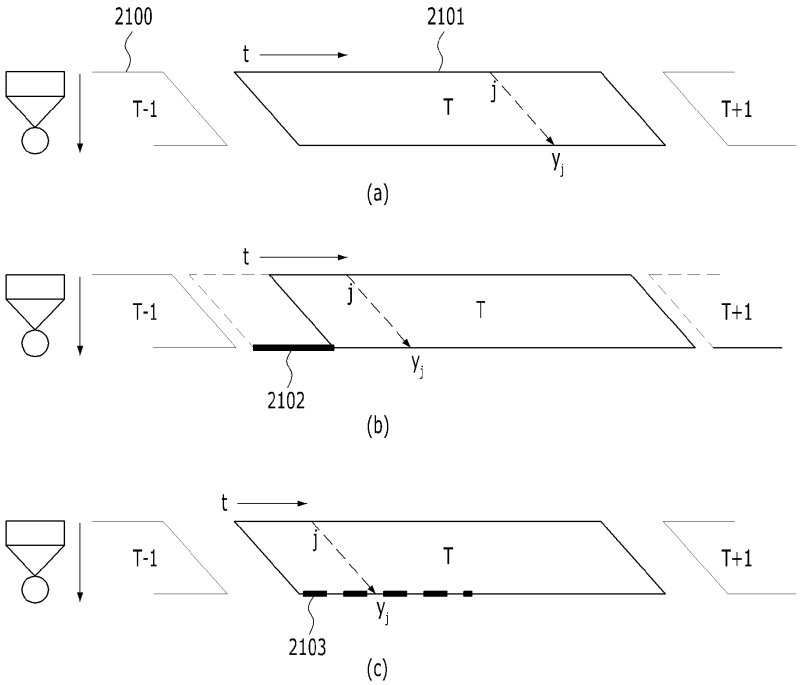
도면19



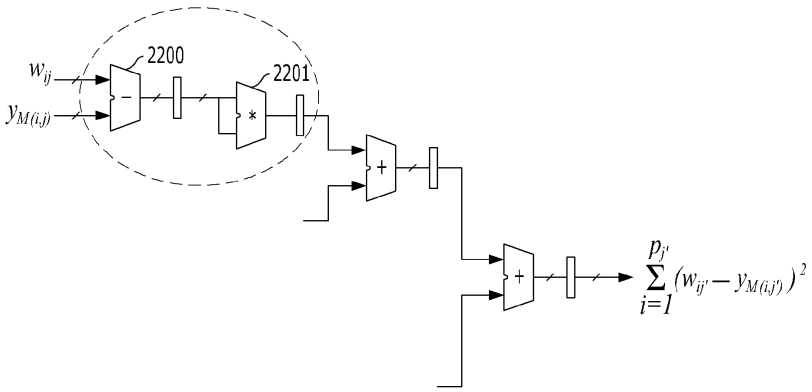
도면20



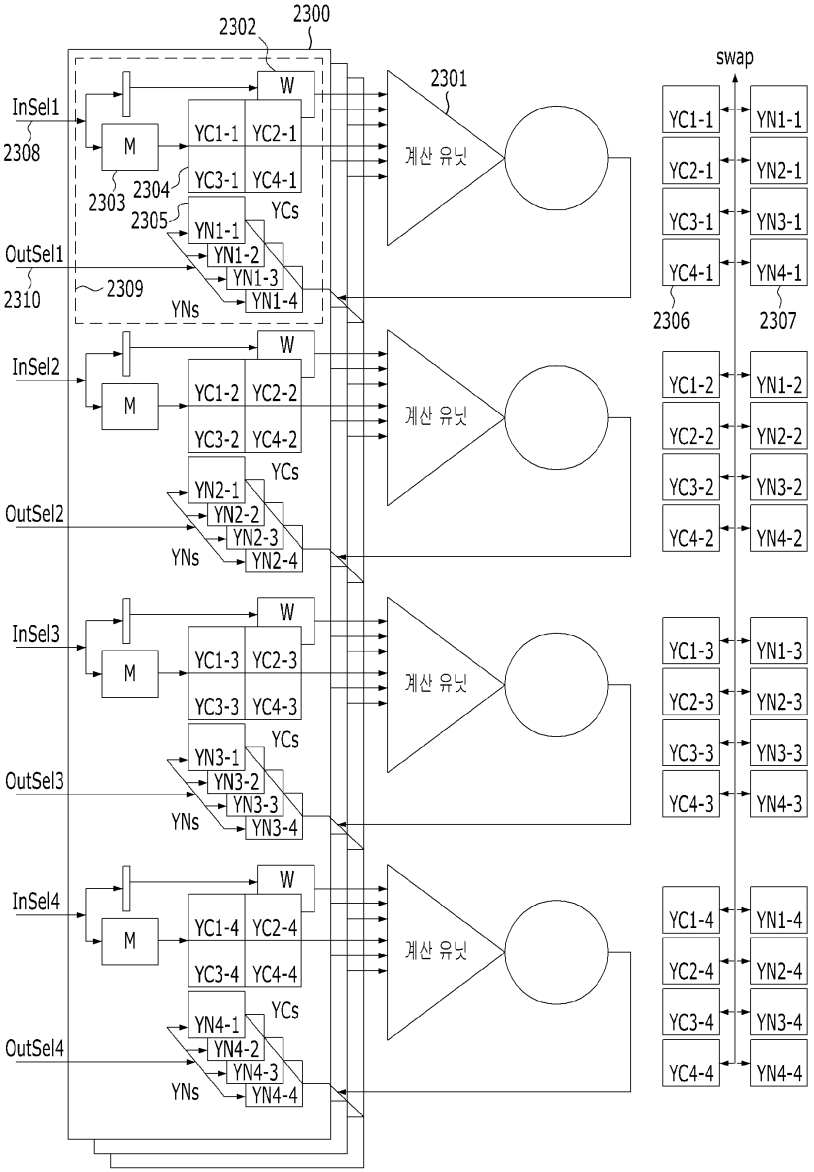
도면21


2100
2101
t
T-1
T
T+1
j
y_j
(a)
t
T-1
j
T
T+1
y_j
2102
(b)
t
T-1
j
T
T+1
y_j
2103
(c)

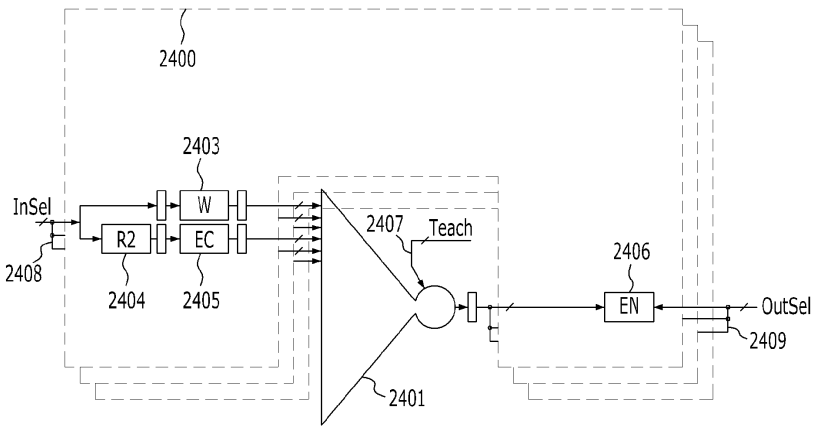
도면22


w_{ij}
$y_{M(i,j)}$
2200
2201
$\sum_{i=1}^{p_{j'}} (w_{ij'} - y_{M(i,j')})^2$

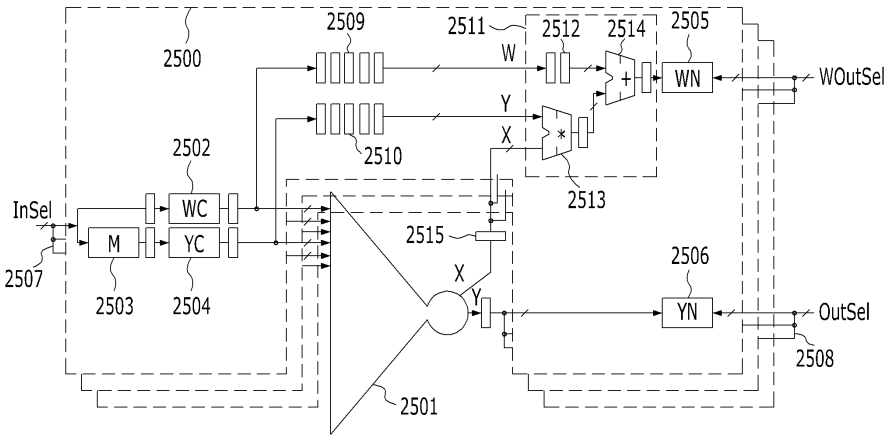
도면23


2302
2300
2301
swap
InSel1
2308
M
W
YC1-1
YC2-1
YC3-1
YC4-1
2303
2304
2305
YCs
YN1-1
YN1-2
YN1-3
YN1-4
OutSel1
2310
2309
YNs
계산 유닛
YC1-1
YN1-1
YC2-1
YN2-1
YC3-1
YN3-1
YC4-1
YN4-1
2306
2307
InSel2
YC1-2
YC2-2
YC3-2
YC4-2
YN2-1
YN2-2
YN2-3
YN2-4
OutSel2
YC1-2
YN1-2
YC2-2
YN2-2
YC3-2
YN3-2
YC4-2
YN4-2
InSel3
YC1-3
YC2-3
YC3-3
YC4-3
YN3-1
YN3-2
YN3-3
YN3-4
OutSel3
YC1-3
YN1-3
YC2-3
YN2-3
YC3-3
YN3-3
YC4-3
YN4-3
InSel4
YC1-4
YC2-4
YC3-4
YC4-4
YN4-1
YN4-2
YN4-3
YN4-4
OutSel4
YC1-4
YN1-4
YC2-4
YN2-4
YC3-4
YN3-4
YC4-4
YN4-4

도면24


2400
2403
InSel
2408
W
R2
EC
2404
2405
2407
Teach
2406
EN
OutSel
2409
2401

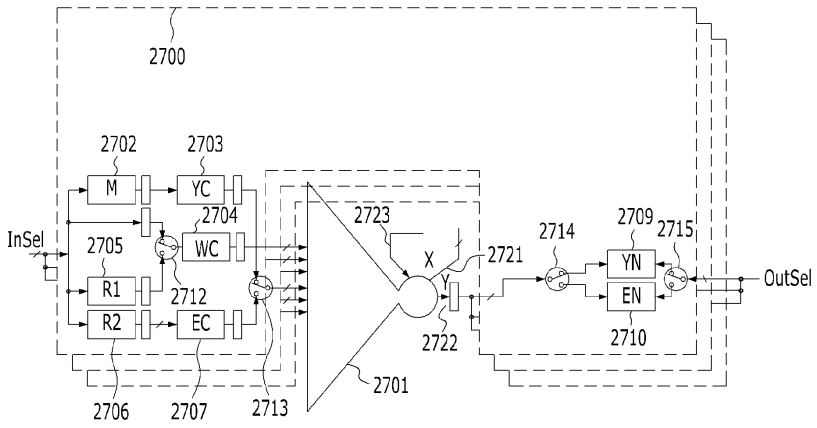
도면25



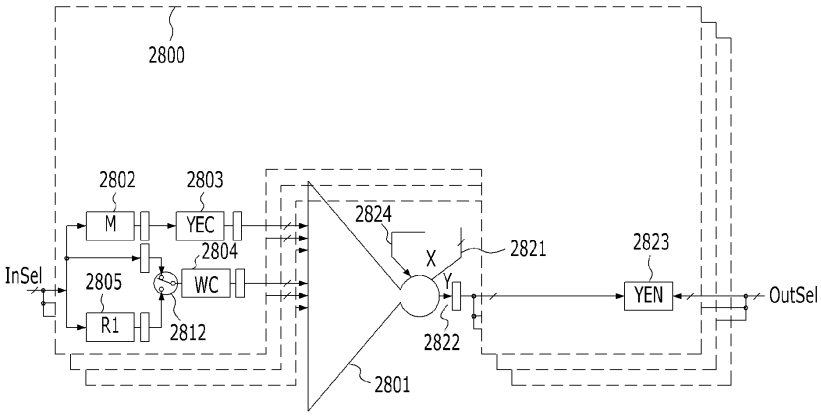
도면26

Register (1619) on WC data Output (Wik)	(k)	(k+1)	(k+2)	(k+3)	(k+4)	(k+5)	(k+6)	(k+7)	(k+7)	(j+2)	(k+4)	(k+4)	(j+2)	(k+3)	(k+3,j+1)	(k+2,j+1)
Register (1620) on YC data Output (Yik)	(k)	(k+1)	(k+2)	(k+3)	(j+1)	(j+1)	(j+2)	(j+2)	(k+6)	(j+1)	(k+3)	(k+3)	(j+1)	(k+2)	(k+2,j+1)	(k+1,j)
X output of Calculation Unit(Lj)	(j-1)	(j-1)	(j)	(j)	(j+1)	(j+1)	(j+1)	(j+2)	(k+5)	(j+1)	(k+2)	(k+2)	(j+1)	(k+1)	(k+1,j)	(k,j)
W input of Connection Adjust Module (Wik)	(k-3)	(k-2)	(k-1)	(k)	(k+1)	(k+2)	(k+3)	(k+4)	(k+4)	(j+2)	(k+1)	(k+1)	(j)	(k)	(k,j)	(k-1,j-1)
Y input of Connection Adjust Module (Yik)	(k-3)	(k-2)	(k-1)	(k)	(k+1)	(k+2)	(k+3)	(k+4)	(k+4)	(j+2)	(k+1)	(k+1)	(j)	(k)	(k,j)	(k-1,j-1)
X input of Connection Adjust Module (Lj)	(j-2)	(j-1)	(j-1)	(j)	(j)	(j+1)	(j+1)	(j+2)	(k+3)	(j+1)	(k)	(k)	(j)	(k-1)	(k-1,j-1)	(k-2,j-1)
Output of FIFO Queue (1612) (Wik)	(k-4)	(k-3)	(k-2)	(k-1)	(k)	(k+1)	(k+2)	(k+3)	(k+3)	(k+1)	(k-1)	(k-1)	(k-2)	(k-1)	(k-1,j-1)	(k-2,j-2)
Register (1615) of Output of Multiplier (Yik*Lj)	(k-4,j-2)	(k-3,j-2)	(k-2,j-1)	(k-1,j-1)	(k,j)	(k+1,j)	(k+2,j+1)	(k+3,j+1)	(k+3,j+1)	(k+1,j)	(k-1,j-1)	(k-1,j-1)	(k-2,j-1)	(k-2,j-2)	(k-3,j-2)	(k-4,j-2)
Register (1616) of Adder (Wik + YMij*Lj)	(k-5,j-3)	(k-4,j-2)	(k-3,j-2)	(k-2,j-1)	(k-1,j-1)	(k,j)	(k+1,j)	(k+2,j+1)	(k+2,j+1)	(k,j)	(k-1,j-1)	(k-1,j-1)	(k-2,j-1)	(k-3,j-2)	(k-4,j-2)	(k-5,j-3)

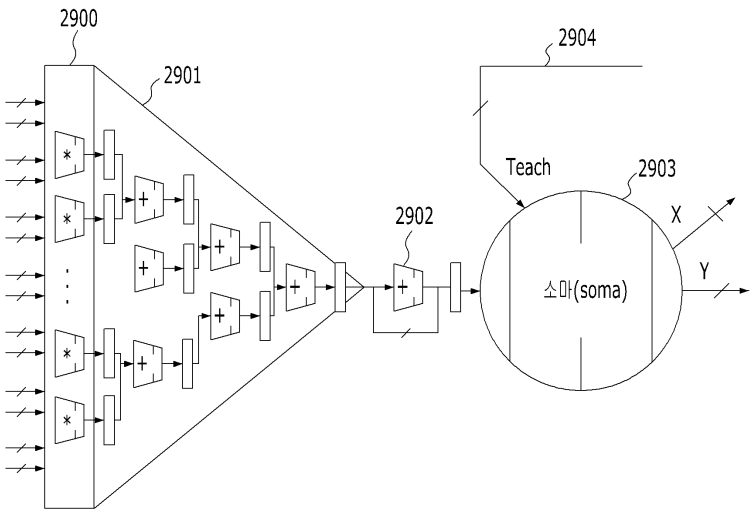
도면27



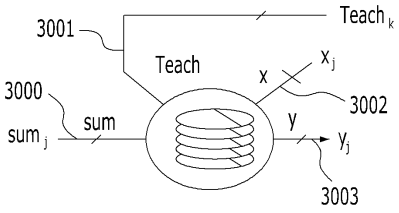
도면28



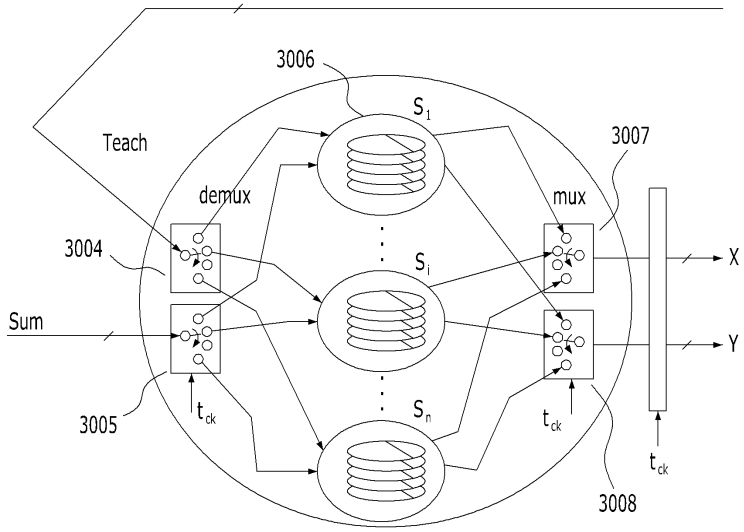
도면29



도면30

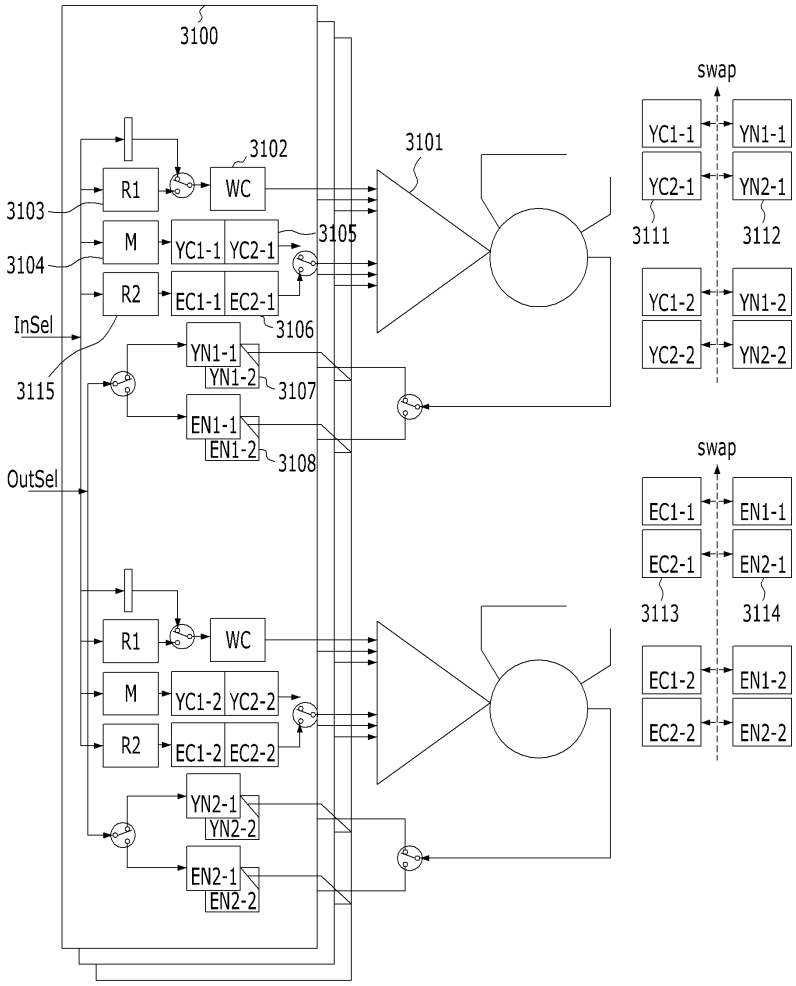

3001
Teach k
Teach
x
x j
3000
3002
sum
sum j
y
y j
3003

(a)

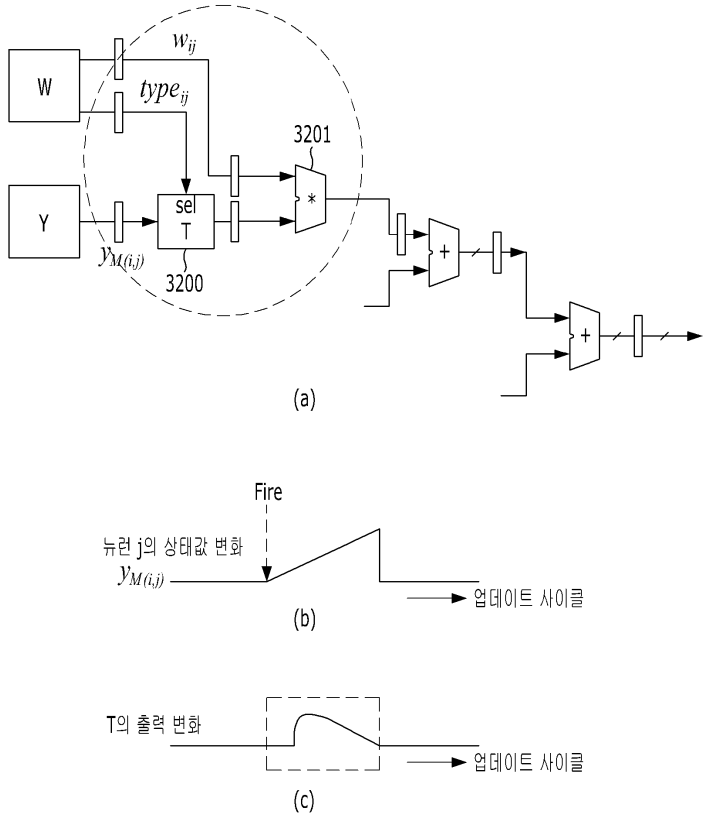

3006
S 1
Teach
3007
demux
mux
3004
S i
X
Sum
Y
S n
t ck
3005
t ck
t ck
3008

(b)

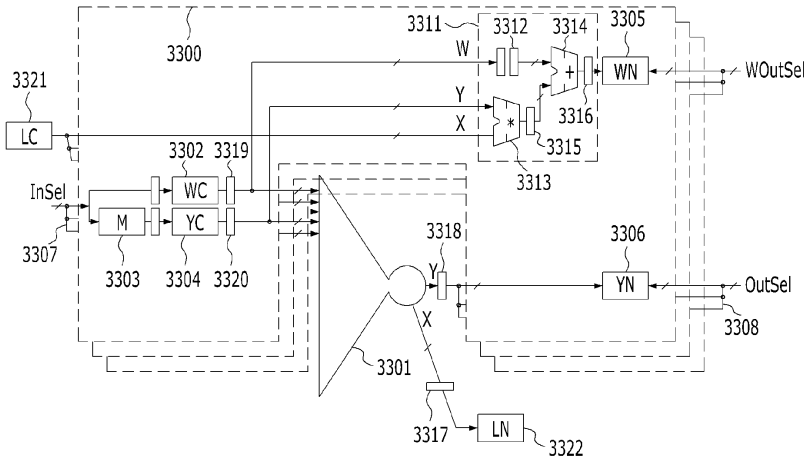
도면31


3100
3101
3102
3103
3104
3105
3106
3107
3108
3111
3112
3113
3114
3115
InSel
OutSel
R1
WC
M
R2
YC1-1
YC2-1
EC1-1
EC2-1
YN1-1
YN1-2
EN1-1
EN1-2
YC1-2
YC2-2
EC1-2
EC2-2
YN2-1
YN2-2
EN2-1
EN2-2
swap
swap

도면32


W
Y
w_{ij}
$type_{ij}$
$y_{M(i,j)}$
sel
T
3200
3201
*
+
+
(a)
Fire
뉴런 j의 상태값 변화
$y_{M(i,j)}$
업데이트 사이클
(b)
T의 출력 변화
업데이트 사이클
(c)

도면33


3300
3311
3312
3314
3305
W
Y
X
WN
WOutSel
3316
3315
3313
3321
LC
3302
3319
WC
YC
InSel
M
3307
3303
3304
3320
3318
3306
YN
OutSel
3308
3301
3317
LN
3322