(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년05월02일
(11) 등록번호 10-2663080
(24) 등록일자 2024년04월29일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2023.01) *G06N 3/04* (2023.01)
G06N 3/063 (2023.01)
(52) CPC특허분류
G06N 3/08 (2023.01)
G06N 3/04 (2023.01)
(21) 출원번호 10-2020-0168245
(22) 출원일자 2020년12월04일
심사청구일자 2020년12월04일
(65) 공개번호 10-2022-0078950
(43) 공개일자 2022년06월13일
(56) 선행기술조사문헌
JP2018169960 A*
Jun Zhou et al., "Polynomial activation neural networks: Modeling, stability analysis and coverage BP-training," Neurocomputing (2019.06.04.)*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
김하영
서울특별시 서대문구 명지길 30, 107동 602호(홍은동, 신원지벤스타)
김정은
서울특별시 강남구 영동대로 230, 3동 502호(대치동, 우성1차아파트)
(74) 대리인
민영준

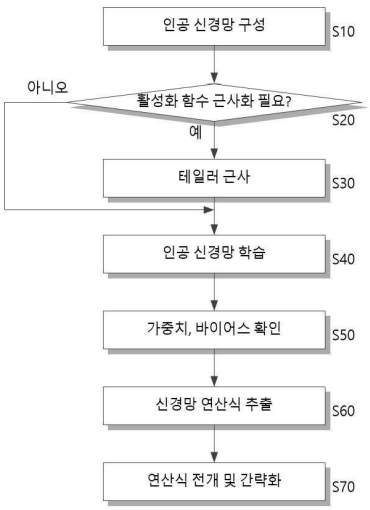
전체 청구항 수 : 총 11 항 심사관 : 송근배

(54) 발명의 명칭 **인공 신경망 수학적 경량화 장치 및 방법**

(57) 요 약

본 발명은 학습이 완료된 인공 신경망이 수행하는 전체 연산을 수학적으로 다항식 형식으로 도출하고, 도출된 다항식에 대한 연산을 수행하는 연산 장치로 인공 신경망을 대체함으로써, 딥러닝 모델로서의 인공 신경망을 수학적으로 경량화하여 학습이 완료된 인공 신경망을 사용함으로써 요구되었던 연산량과 메모리 용량을 저감시키고, 연산 속도를 고속화할 수 있는 인공 신경망 수학적 경량화 장치 및 방법을 제공할 수 있다.

대 표 도 - 도6



(52) CPC특허분류
G06N 3/063 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1415169319
과제번호 20202020800030
부처명 산업통상자원부
과제관리(전문)기관명 한국에너지기술평가원
연구사업명 에너지수요관리핵심기술
연구과제명 제로에너지건축물 구현을 위한 스마트 외장재·설비 융복합 기술개발 및 성능평가 체계 구축, 실증
기 여 율 1/1
과제수행기관명 한국건설기술연구원
연구기간 2020.05.01 ~ 2020.12.31

명 세 서

청구범위

청구항 1

인공 신경망의 구조를 설정하기 위한 신경망 설정값을 획득하는 구조 설정 모듈;

상기 신경망 설정값에 따라 인공 신경망을 구성하는 신경망 모듈;

상기 신경망 모듈에 구성된 인공 신경망에서 각 레이어별로 노드에 지정된 활성화 함수 중 근사화가 요구되는 활성화 함수가 존재하는지 여부를 판단하고, 근사화가 요구되는 활성화 함수가 존재할 경우 해당 활성화 함수를 테일러 근사 기법에 따라 다항식으로 치환하는 활성 함수 근사 모듈;

상기 신경망 모듈에 구성된 인공 신경망을 미리 지정된 방식으로 학습시키는 학습 모듈;

학습된 인공 신경망이 수행하는 전체 연산에 대한 연산식을 획득하는 연산식 추출 모듈; 및

획득된 연산식을 수학적으로 전개하고, 전개된 연산식을 다항식으로 간략화하여 학습된 인공 신경망에 대응하는 동작을 수행하는 간략식을 획득하는 간략식 전환 모듈을 포함하되,

상기 학습 모듈은

활성화 함수가 근사 다항식으로 치환된 경우, 치환된 근사 다항식에 따라 상기 인공 신경망을 학습시키는 인공 신경망 수학적 경량화 장치.

청구항 2

제1항에 있어서, 상기 구조 설정 모듈은

상기 신경망 설정값으로 인공 신경망에 포함되는 레이어 개수와 각 레이어에 포함되는 노드 개수, 각 레이어별로 노드에 지정되는 활성화 함수 및 가중치 행렬과 바이어스 행렬의 초기값을 획득하는 인공 신경망 수학적 경량화 장치.

청구항 3

삭제

청구항 4

제1항에 있어서, 상기 근사화가 요구되는 활성화 함수는

시그모이드 함수, 하이퍼볼릭 탄젠트 함수 중 하나인 인공 신경망 수학적 경량화 장치.

청구항 5

삭제

청구항 6

제1항에 있어서, 상기 연산식 추출 모듈은

인공 신경망의 구조에 따라 입력 레이어로부터 출력 레이어까지 각 레이어에서 수행되는 연산이 순차적으로 수행되는 구조의 연산식을 추출하는 인공 신경망 수학적 경량화 장치.

청구항 7

제6항에 있어서, 상기 연산식 추출 모듈은

활성화 함수가 근사 다항식으로 치환된 경우, 치환된 근사 다항식을 적용하여 연산식을 추출하는 인공 신경망 수학적 경량화 장치.

청구항 8

인공 신경망 수학적 경량화 장치에서 수행되는 인공 신경망 수학적 경량화 방법으로서,

인공 신경망의 구조를 설정하기 위한 신경망 설정값을 획득하는 단계;

상기 신경망 설정값에 따라 인공 신경망을 구성하는 단계;

상기 인공 신경망을 구성하는 단계 이후, 상기 구성된 인공 신경망에서 각 레이어별로 노드에 지정된 활성화 함수 중 근사화가 요구되는 활성화 함수가 존재하는지 여부를 판단하고, 근사화가 요구되는 활성화 함수가 존재할 경우 해당 활성화 함수를 테일러 근사 기법에 따라 다항식으로 치환하는 단계;

구성된 인공 신경망을 미리 지정된 방식으로 학습시키는 단계;

학습된 인공 신경망이 수행하는 전체 연산에 대한 연산식을 획득하는 단계; 및

획득된 연산식을 수학적으로 전개하고, 전개된 연산식을 다항식으로 간략화하여 학습된 인공 신경망에 대응하는 동작을 수행하는 간략식을 획득하는 단계를 포함하되,

상기 학습시키는 단계는

활성화 함수가 근사 다항식으로 치환된 상기 인공 신경망을 학습시키는 인공 신경망 수학적 경량화 방법.

청구항 9

제8항에 있어서, 상기 인공 신경망을 구성하는 단계는

상기 신경망 설정값으로 인공 신경망에 포함되는 레이어 개수와 각 레이어에 포함되는 노드 개수, 각 레이어별로 노드에 지정되는 활성화 함수 및 가중치 행렬과 바이어스 행렬의 초기값을 획득하는 인공 신경망 수학적 경량화 방법.

청구항 10

삭제

청구항 11

제8항에 있어서, 상기 근사화가 요구되는 활성화 함수는

시그모이드 함수, 하이퍼볼릭 탄젠트 함수 중 하나인 인공 신경망 수학적 경량화 방법.

청구항 12

삭제

청구항 13

제8항에 있어서, 상기 연산식을 획득하는 단계는

인공 신경망의 구조에 따라 입력 레이어로부터 출력 레이어까지 각 레이어에서 수행되는 연산이 순차적으로 수행되는 구조의 연산식을 추출하는 인공 신경망 수학적 경량화 방법.

청구항 14

제13항에 있어서, 상기 연산식을 획득하는 단계는

활성화 함수가 근사 다항식으로 치환된 경우, 치환된 근사 다항식을 적용하여 연산식을 추출하는 인공 신경망 수학적 경량화 방법.

청구항 15

제8항의 인공 신경망 수학적 경량화 방법을 수행하기 위해 컴퓨팅 장치에서 판독 가능한 프로그램 명령어가 기록된 기록 매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 인공 신경망 수학적 경량화 장치 및 방법에 관한 것으로, 인공 신경망을 수학적으로 다항식으로 변환하여 에러를 줄이고 고속화 할 수 있는 인공 신경망 수학적 경량화 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 딥 러닝(Deep Learning)은 다중 레이어 구조를 갖는 인공 신경망을 기반으로 하는 머신 러닝의 한 분야로, 다량의 데이터로부터 높은 수준의 추상화 모델을 구축하고자 하는 기법이다. 현재 인공 신경망은 이미지 처리, 객체 분류, 객체 검출, 음성 인식, 자연어 처리와 같은 매우 다양한 분야에서 적용되고 있을 뿐만 아니라 그 적용 분야가 계속 확장되어 가고 있다.

[0003] 도 1은 인공 신경망의 일 예를 나타낸다.

[0004] 도 1에 도시된 바와 같이 딥 러닝 모델로서의 인공 신경망은 입력 레이어(Input layer)과 출력 레이어(Output layer) 사이에 다수의 히든 레이어(Hidden layer 1, Hidden layer 2)가 배치되는 구조를 갖는다. 이러한 구조의 인공 신경망에서 다수의 히든 레이어의 각 노드(뉴런 또는 커널이라고도 함)는 이전 레이어의 노드들로부터 인가되는 값에 대해 학습에 의해 결정된 가중치(weight)를 가중하고 바이어스(bias)와 합한 후, 기지정된 활성화 함수(activation function)에 따라 기지정된 연산을 수행하여 활성화하여 출력한다. 즉 인공 신경망의 경우, 다수의 레이어가 각각 이전 레이어의 노드별 연산 결과를 기반으로 순차적으로 기지정된 연산을 수행하게 된다.

[0005] 이와 같은 인공 신경망은 개략적인 수학식에 따라 우선 구조가 결정되고, 딥 러닝 기반 학습을 통해 수학식의 여러 파라미터로서 가중치 및 바이어스가 결정된다. 그리고 학습이 완료된 인공 신경망은 실제 이용되는 분야에서 입력되는 입력 데이터에 대해 결정된 파라미터에 따른 연산을 수행하여 연산 결과를 출력값으로 반환한다.

[0006] 이렇게 학습이 완료된 인공 신경망의 구조 그대로 인공 신경망 사용을 목표로 하는 데이터에 적용이 되기 때문에 실제 이용되는 경우에도 학습 시와 마찬가지로 다수의 레이어가 순차적으로 연산을 수행하므로 많은 연산량이 요구된다. 뿐만 아니라, 레이어별로 연산된 데이터를 임시로 저장하기 위해 대용량의 메모리가 함께 요구되며, 상당한 연산을 수행하므로 연산 시간이 길다는 문제가 있다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 한국 등록 특허 제10-2137802호 (2020.07.20 등록)

발명의 내용

해결하려는 과제

[0008] 본 발명의 목적은 학습이 완료된 인공 신경망을 수학적으로 경량화할 수 있는 인공 신경망 수학적 경량화 장치 및 방법을 제공하는데 있다.

[0009] 본 발명의 다른 목적은 학습이 완료된 인공 신경망에서 수행되는 전체 연산을 하나의 다항식으로 도출하고, 타겟 데이터를 학습된 인공 신경망을 통해 결과값을 도출하는 대신에, 간단한 다항식 연산을 수행함으로써, 학습이 완료된 인공 신경망을 이용함으로써 요구되었던 연산량과 메모리 용량을 저감 시킬 뿐만 아니라 고속화 할 수 있는 인공 신경망 수학적 경량화 장치 및 방법을 제공하는데 있다.

과제의 해결 수단

[0010] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 인공 신경망 수학적 경량화 장치는 인공 신경망의 구조를 설정하기 위한 신경망 설정값을 획득하는 구조 설정 모듈; 상기 신경망 설정값에 따라 인공 신경망을 구성하는 신경망 모듈; 상기 신경망 모듈에 구성된 인공 신경망을 미리 지정된 방식으로 학습시키는 학습 모듈; 학습된 인공 신경망이 수행하는 전체 연산에 대한 연산식을 획득하는 연산식 추출 모듈; 및 획득된 연산식을 수학적으로 전개하고, 전개된 연산식을 다항식으로 간략화하여 학습된 인공 신경망에 대응하는 동작을 수행하는 간략식을 획득하는 간략식 전환 모듈을 포함한다.

[0011] 상기 구조 설정 모듈은 상기 신경망 설정값으로 인공 신경망에 포함되는 레이어 개수와 각 레이어에 포함되는 노드 개수, 각 레이어별로 노드에 지정되는 활성화 함수 및 가중치 행렬과 바이어스 행렬의 초기값을 획득할 수 있다.

[0012] 상기 인공 신경망 수학적 경량화 장치는 상기 신경망 모듈에 구성된 인공 신경망에서 각 레이어별로 노드에 지정된 활성화 함수 중 기지정된 다항식 근사 대상 활성화 함수를 탐색하고, 탐색된 활성화 함수를 테일러 근사 기법에 따라 근사 다항식으로 치환하는 활성 함수 근사 모듈을 더 포함할 수 있다.

[0013] 상기 기지정된 다항식 근사 대상 활성화 함수는 시그모이드 함수, 하이퍼볼릭 탄젠트 함수 중 하나일 수 있다.

[0014] 상기 학습 모듈은 활성화 함수가 근사 다항식으로 치환된 경우, 치환된 근사 다항식에 따라 상기 인공 신경망을 학습시킬 수 있다.

[0015] 상기 연산식 추출 모듈은 인공 신경망의 구조에 따라 입력 레이어로부터 출력 레이어까지 각 레이어에서 수행되는 연산이 순차적으로 수행되는 구조의 연산식을 추출할 수 있다.

[0016] 상기 연산식 추출 모듈은 활성화 함수가 근사 다항식으로 치환된 경우, 치환된 근사 다항식을 적용하여 연산식을 추출할 수 있다.

[0017] 상기 목적을 달성하기 위한 본 발명의 다른 실시예에 따른 인공 신경망 수학적 경량화 방법은 인공 신경망의 구조를 설정하기 위한 신경망 설정값을 획득하는 단계; 상기 신경망 설정값에 따라 인공 신경망을 구성하는 단계; 구성된 인공 신경망을 미리 지정된 방식으로 학습시키는 단계; 학습된 인공 신경망이 수행하는 전체 연산에 대한 연산식을 획득하는 단계; 및 획득된 연산식을 수학적으로 전개하고, 전개된 연산식을 다항식으로 간략화하여 학습된 인공 신경망에 대응하는 동작을 수행하는 간략식을 획득하는 단계를 포함한다.

발명의 효과

[0018] 따라서, 본 발명의 실시예에 따른 인공 신경망 수학적 경량화 장치 및 방법은 학습이 완료된 인공 신경망이 수행하는 전체 연산을 수학적으로 하나의 다항식 형식으로 도출하고, 도출된 다항식에 대한 연산을 수행하는 연산 장치로 인공 신경망을 대체함으로써, 딥러닝 모델로서의 인공 신경망을 수학적으로 경량화하여 학습이 완료된 인공 신경망을 이용하면서 요구되었던 연산량과 메모리 용량을 저감시키고, 연산 속도를 고속화할 수 있다.

도면의 간단한 설명

[0019] 도 1은 인공 신경망의 일 예를 나타낸다.

도 2는 인공 신경망에서 가중치를 설명하기 위한 도면이다.

도 3은 인공 신경망의 개별 노드의 동작을 설명하기 위한 도면이다.

도 4는 인공 신경망의 활성화 함수의 다양한 예를 나타낸다.

도 5는 본 발명의 일 실시예에 따른 인공 신경망 수학적 경량화 장치의 개략적 구조를 나타낸다.

도 6은 본 발명의 일 실시예에 따른 인공 신경망 수학적 경량화 방법을 나타낸다.

도 7은 도 6의 인공 신경망 수학적 경량화 방법을 수행하기 위한 컴퓨팅 장치의 예를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0020] 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본

발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.

[0021] 이하, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 설명함으로써, 본 발명을 상세히 설명한다. 그러나, 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 설명하는 실시예에 한정되는 것이 아니다. 그리고, 본 발명을 명확하게 설명하기 위하여 설명과 관계없는 부분은 생략되며, 도면의 동일한 참조부호는 동일한 부재임을 나타낸다.

[0022] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있는 것을 의미한다. 또한, 명세서에 기재된 "...부", "...기", "모듈", "블록" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0023] 여기서는 예시로서 메모리(720)를 프로세서(710)와 별도의 구성 요소로 표시하였으나, 메모리(720)는 프로세서(710)에 포함되어 구성될 수도 있다. 일 예로 메모리(720)는 프로세서(710) 내에 포함되는 캐시 메모리 등으로 구현될 수도 있다.

[0024] 도 2는 인공 신경망에서 가중치를 설명하기 위한 도면이고, 도 3은 인공 신경망의 개별 노드의 동작을 설명하기 위한 도면이다.

[0025] 도 2에서도 간단한 일 예로서 도 1의 인공 신경망과 동일한 구조의 인공 신경망을 가정하였으며, 이에 입력 레이어(Input layer)에는 2개의 입력 노드가 포함되고, 제1 히든 레이어(Hidden layer 1)의 4개의 노드가 포함되며, 제2 히든 레이어(Hidden layer 2)의 3개의 노드가 포함된다. 그리고 출력 레이어(Output layer)에는 하나의 출력 노드가 포함된다.

[0026] 도 2를 참조하면, 입력 레이어(Input layer)의 2개의 입력 노드는 입력 노드의 개수에 대응하는 개수의 입력값(x_1, x_2)을 원소로 포함하는 입력 행렬(X) 중 대응하는 입력값을 인가받고, 인가된 입력값(x_1, x_2)을 제1 히든 레이어(Hidden layer 1)의 4개의 노드의 입력값으로 전달한다.

[0027] 그리고 제1 히든 레이어(Hidden layer 1)의 4개의 노드 각각은 입력 레이어(Input layer)의 2개의 입력 노드에서 전달되는 입력값(x_1, x_2) 각각에 미리 수행된 학습에 의해 결정된 대응하는 가중치((w^1_{11}, w^1_{21}) ~ (w^1_{14}, w^1_{24}))를 가중한다. 여기서는 설명의 편의를 위하여 각 레이어의 식별자(i)를 위첨자로 표기하고, 서로 인접하여 배치된 2개의 레이어(i-1, i) 중 이전 배치된 레이어(i-1)의 노드 식별자(j)와 이후 배치된 레이어(i)의 노드 식별자(k)를 아래첨자로 표기하였다. 일 예로 W^i 는 i-1번째 레이어에서 i 번째 레이어로의 가중치 행렬을 나타내고, w^i_{jk} 는 i-1 번째 레이어의 j번째 노드와 i번째 레이어의 k번째 노드 사이의 가중치를 나타낸다.

[0028] 따라서 입력 레이어(Input layer)의 2개의 입력 노드와 제1 히든 레이어(Hidden layer 1)의 4개의 노드 사이의 가중치 행렬(W^1)은 수학식 1과 같이 표현될 수 있다.

수학식 1

[0029]
$$W^1 = \begin{bmatrix} w^1_{11} & \cdots & w^1_{14} \\ \vdots & \ddots & \vdots \\ w^1_{21} & \cdots & w^1_{24} \end{bmatrix}$$

[0030] 여기서 제1 히든 레이어(Hidden layer 1)의 4개의 노드 각각은 대응하는 2개의 입력 노드에서 입력값(x_1, x_2)이 전달되면, 각 입력값(x_1, x_2)에 대응하는 가중치((w^1_{11}, w^1_{21}) ~ (w^1_{14}, w^1_{24}))를 가중하고, 가중치((w^1_{11}, w^1_{21}) ~ (w^1_{14}, w^1_{24}))가 가중된 입력값(($x_1w^1_{11}$, $x_2w^1_{21}$) ~ ($x_1w^1_{14}$, $x_2w^1_{24}$))과 학습에 의해 미리 설정된 바이어스(b^1_1)를 합산하고, 합산 결과에 대해 미리 지정된 활성화 함수(f^1)로 기지정된 연산을 수행하여 각 노드별로 노드 출력값

(u^1_1 ~ u^1_4)을 제2 히든 레이어(Hidden layer 2)의 3개의 노드 각각의 입력값으로 전달한다. 여기서 제1 히든 레이어(Hidden layer 1)의 노드들이 출력하는 노드 출력값(u^1_1 ~ u^1_4)은 레이어 출력 행렬(U^1)이라 할 수 있으며, 각 노드의 바이어스(b^1_1, ~ b^1_4)는 바이어스 행렬(B^1)이라 할 수 있다.

[0031] 따라서 제1 히든 레이어(Hidden layer 1)의 출력 행렬(U^1)은 입력 행렬(X)에 가중치 행렬(W^1)을 가중($X * W^1$)하고, 바이어스(B^1)를 합산한 합산값($(W^1X)+B^1$)을 활성화 함수(f^1)의 입력으로 인가된 결과이므로 수학식 2로 표현될 수 있다.

수학식 2

$$U^1 = f^1(W^1X + B^1)$$

[0032]

[0033] 일반적으로 인공 신경망에서 동일 레이어에 포함되는 노드들은 동일한 활성화 함수를 사용하므로, 제1 히든 레이어(Hidden layer 1)의 다수의 노드에 설정된 활성화 함수(f^1)는 제1 히든 레이어(Hidden layer 1)의 활성화 함수라 할 수 있다.

[0034] 한편, 제2 히든 레이어(Hidden layer 2)의 3개의 노드 또한 제1 히든 레이어(Hidden layer 1)의 노드들과 마찬가지로, 이전 레이어인 제1 히든 레이어(Hidden layer 1)의 각 노드에서 출력되는 노드 출력값(u^1_1 ~ u^1_4) 각각에 수학식 3과 같이 미리 수행된 학습에 의해 결정된 가중치 행렬(W^2)의 대응하는 가중치((w^2_{11}, w^2_{41}), (w^2_{12}, w^2_{42}), (w^1_{13}, w^1_{43}))를 가중하고, 바이어스(b^2_1, ~ b^2_3)와 함께 합산한 후 활성화 함수(f^2)로 활성화하여 3개의 노드 출력값(u^2_1 ~ u^2_3)으로 구성된 레이어 출력 행렬(U^2)을 출력한다.

수학식 3

$$W^2 = \begin{bmatrix} w^2_{11} & \cdots & w^2_{13} \\ \vdots & \ddots & \vdots \\ w^2_{41} & \cdots & w^2_{43} \end{bmatrix}$$

[0035]

[0036] 따라서 제2 히든 레이어(Hidden layer 2)의 출력 행렬(U^2)은 수학식 4와 같이 표현될 수 있다.

수학식 4

$$\begin{aligned} U^2 &= f^2(W^2U^1 + B^2) \\ &= f^2(W^2(f^1(W^1X + B^1)) + B^2) \end{aligned}$$

[0037]

[0038] 그리고 출력 레이어(Output layer)의 출력 노드는 다시 이전 레이어인 제2 히든 레이어(Hidden layer 2)의 각 노드에서 출력되는 노드 출력값(u^2_1 ~ u^2_3) 각각에 미리 결정된 가중치 행렬(W^3)의 대응하는 가중치((w^3_{11}, w^3_{21},

w^{3}_{31})를 가중하고, 바이어스(b^{3}_{1})와 함께 합산한 후 활성화 함수(f^{3})로 활성화하여 노드 출력값(u^{3}_{1})을 출력한다.

[0039] 결과적으로 도 3에 도시된 바와 같이, 인공 신경망에서 각 노드는 이전 배치된 레이어(i-1)에 포함된 j개의 노드 각각에서 출력되는 노드 출력값(u^{i-1}_{1} ~ u^{i-1}_{j})를 인가받아, 학습에 의해 결정된 가중치(w^{i}_{1k} ~ w^{i}_{jk})를 가중하고, 가중치와 마찬가지로 학습에 의해 결정된 바이어스(b^{i}_{k})와 함께 합산한 후, 미리 지정된 활성화 함수(f^{i})로 활성화하여, 노드 출력값(u^{i}_{k})을 출력한다.

[0040] 여기서는 출력 레이어(Output layer)가 하나의 출력 노드만을 포함하므로, 노드 출력값(u^{3}_{1})을 곧 레이어 출력 행렬(U^{3})로 볼 수 있으며, 레이어 출력 행렬(U^{3})은 인공 신경망의 최종 출력이므로, 인공 신경망의 출력 행렬(Y)이라 할 수 있으며, 수학식 5와 같이 계산될 수 있다.

수학식 5

[0041]
$$
\begin{aligned}
Y &= U^{3} \\
&= f^{3}(W^{3}U^{2} + B^{3}) \\
&= f^{3}(W^{3}(f^{2}(W^{2}(f^{1}(W^{1}X + B^{1})) + B^{2})) + B^{3})
\end{aligned}
$$

[0042] 상기한 바와 같이, 인공 신경망은 다수의 레이어 각각에서 다수의 노드들이 이전 레이어(i-1)의 다수의 노드에서 출력되는 레이어 출력값(U^{i-1})에 대해 지정된 가중치 행렬(W^{i})을 가중하고, 바이어스 행렬(B^{i})을 합산한 후, 활성화 함수에 대입하는 과정을 레이어의 수에 대응하여 반복 수행하는 연산 장치로 볼 수 있다.

[0043] 따라서 학습이 완료된 인공 신경망의 경우, 수학식 5와 같이, 인공 신경망에서 수행되는 연산 전체를 수학적으로 표현할 수 있다. 다만 수학식 5에서는 인공 신경망의 구조에서 각 레이어의 배치 순서에 따라 우선 배치된 레이어의 연산이 우선 수행되고, 이후 배치된 레이어에 대한 연산들이 순차적으로 연산되는 구조이다. 이때 각 레이어(i)별 가중치 행렬(W^{i})과 바이어스 행렬(B^{i})이 학습에 의해 미리 결정되어 있을 뿐만 아니라 활성화 함수(f^{i}) 또한 인공 신경망을 구성할 때 미리 결정된 함수이다.

[0044] 따라서 인공 신경망의 출력 행렬(Y)은 수학식 5를 입력 행렬(X)에 대한 다항식 형태로 전개될 수 있다. 그리고 전개된 입력 행렬(X)에 대한 다항식에서는 레이어 배치에 따른 연산 우선 순위가 필요하지 않다. 즉 일반 다항식의 연산과 동일한 방식으로 연산이 수행될 수 있다. 그러므로 적용 대상에 학습 완료된 인공 신경망의 원 구조를 그대로 적용하지 않아도 무방하며, 더 적은 연산을 수행하면서, 기존의 연산 프로세서 등을 이용하여도 인공 신경망과 동일한 연산 결과를 도출할 수 있다.

[0045] 따라서 다양한 분야에서 인공 신경망의 적용이 매우 용이해질 수 있다. 뿐만 아니라, 레이어별 연산 결과를 별도로 저장할 필요가 없으므로, 연산 결과를 임시 저장하기 위해 요구되는 메모리 용량을 저감시킬 수 있고, 연산 속도를 향상시킬 수 있다.

[0046] 다만 활성화 함수에 따라 다항식 형태의 전개가 용이하지 않은 경우가 발행할 수 있다. 따라서 활성화 함수를 다항식으로 전개할 수 있는 방안이 고려되어야 한다.

[0047] 도 4는 인공 신경망의 활성화 함수의 다양한 예를 나타낸다.

[0048] 도 4는 일반적으로 인공 신경망에서 주로 이용되는 대표적인 3가지 활성화 함수에 대한 그래프를 도시하였다. 도 4에서 (a)는 시그모이드 함수(Sigmoid function)를 나타내고, (b)는 하이퍼볼릭 탄젠트 함수(Hyperbolic tangent/Tang function)를 나타내며, (c)는 ReLU 함수(Rectified Linear Unit function)를 나타낸다.

[0049] 이중 (c)에 도시된 ReLU 함수의 경우, 미분 불가능한 비선형 함수이지만 단순 구조의 함수이므로, 다항식으로의 전개가 가능하다. 그러나 (a)와 (b)의 시그모이드 함수나 하이퍼볼릭 탄젠트 함수의 경우, 다항식으로의 전개가 용이하지 않다.

[0050] 따라서 본 실시예에서는 인공 신경망의 각 레이어에 설정된 활성화 함수가 다항식으로 전개가 용이하지 않은 함수인 경우, 활성화 함수(f)를 테일러 근사를 이용하여 테일러 급수 형태의 근사 다항식으로 치환함으로써, 인공 신경망의 연산이 다항식 연산으로 전환될 수 있도록 한다.

[0051] 다만, 기지정된 활성화 함수로 인공 신경망을 학습 시키고, 인공 신경망의 학습이 완료된 이후, 활성화 함수(f)를 테일러 근사 기법에 따라 근사 다항식으로 치환하게 되면, 학습에 이용된 활성화 함수와 실제 적용시에 이용되는 근사 다항식 사이의 오차로 인해 인공 신경망 연산의 신뢰도가 하락하게 된다. 본 실시예에서는 이와 같이 인공 신경망의 학습과 실제 이용 시, 서로 다른 함수를 이용함으로 인해 발생할 수 있는 오차 문제를 해결하기 위해, 학습 시에도 인공 신경망의 활성화 함수를 테일러 근사 기법으로 근사한 근사 다항식으로 구성한다. 즉 인공 신경망의 활성화 함수를 미리 근사 다항식으로 치환한 후, 인공 신경망을 학습시키고, 학습 결과로 가중치와 바이어스를 획득한 후, 획득된 가중치와 바이어스가 적용된 연산식을 전개함으로써, 인공 신경망의 연산에 대응하는 다항식을 획득할 수 있다.

[0052] 도 5는 본 발명의 일 실시예에 따른 인공 신경망 수학적 경량화 장치의 개략적 구조를 나타낸다.

[0053] 도 5를 참조하면, 본 실시예에 따른 인공 신경망 수학적 경량화 장치는 구조 설정 모듈(510), 신경망 모듈(520), 활성 함수 근사 모듈(530), 학습 모듈(540), 연산식 추출 모듈(550) 및 간략식 전환 모듈(560)을 포함할 수 있다.

[0054] 구조 설정 모듈(510)은 인공 신경망의 구조에 대한 신경망 설정값을 획득한다. 이때 구조 설정 모듈(510)은 신경망 설정값으로 인공 신경망의 레이어 개수와 레이어별 노드 개수, 가중치 및 바이어스 초기값 등을 인가받아 설정할 수 있다. 또한 각 레이어의 노드에서 수행될 활성화 함수(f)를 설정한다. 여기서 신경망 설정값은 사용자 명령으로 인가되거나, 미리 저장된 메모리 등으로부터 획득될 수 있다.

[0055] 그리고 신경망 모듈(520)은 구조 설정 모듈(510)에서 획득된 구조 설정값에 따라 인공 신경망을 구성한다.

[0056] 활성 함수 근사 모듈(530)은 구조 설정 모듈(510)에 의해 구성된 신경망 모듈(520)에서 각 레이어별 활성화 함수(f)를 확인하고, 확인된 활성화 함수 중 근사화가 요구되는 기지정된 활성화 함수가 존재하면, 테일러 근사 기법에 따라 테일러 급수 형태의 근사 다항식으로 활성화 함수를 치환한다. 여기서 근사화가 요구되는 기지정된 활성화 함수에는 시그모이드 함수, 하이퍼볼릭 탄젠트 함수 등이 포함될 수 있다.

[0057] 학습 모듈(540)은 기지정된 방식으로 신경망 모듈(520)을 학습시킨다. 학습 모듈(540)은 신경망 모듈(520)에 구성된 인공 신경망을 학습시키기 위해 미리 지정된 학습 방식으로 학습을 수행할 수 있으며, 지도 학습(Supervised Learning) 방식 또는 비지도 학습(Unsupervised Learning) 방식 등을 이용하여 인공 신경망을 학습시킬 수 있다. 학습 모듈(540)은 인공 신경망이 요구되는 성능을 나타낼 수 있도록, 인공 신경망의 출력에 대해 기지정된 방식으로 손실(Loss)을 계산하고, 계산된 손실을 인공 신경망으로 역전파하여 각 레이어별 노드들의 가중치와 바이어스를 반복적으로 업데이트함으로 인공 신경망을 학습시킨다.

[0058] 한편, 학습 모듈(540)에 의해 신경망 모듈(520)의 인공 신경망에 대한 학습이 완료되면, 연산식 추출 모듈(540)은 학습된 인공 신경망 모듈(520)에서 최종 업데이트된 레이어별 가중치 행렬(W)과 바이어스 행렬(B), 그리고 활성화 함수(f)를 획득하여, 인공 신경망 전체에서 수행되는 연산에 대한 연산식을 추출한다. 이때 활성화 함수(f)에서 근사 다항식으로 치환된 경우에는 치환된 근사 다항식이 추출된다.

[0059] 연산식 추출 모듈(550)은 수학식 5에서와 같이 인공 신경망의 구조에 따라 입력 레이어(Input Layer)로부터 출력 레이어(Output Layer)까지 각 레이어에서 수행되는 연산이 순차적으로 수행되는 구조의 연산식을 추출할 수 있다.

[0060] 간략식 전환 모듈(560)은 연산식 추출 모듈(550)이 인공 신경망의 구조에 따라 획득한 연산식을 수학적으로 전개하고, 전개된 수학식을 간략화하여 간략화된 연산식으로 전환하여 출력한다.

[0061] 이후 인공 신경망을 적용하고자 하는 적용 대상은 신경망 모듈(520)에서 구성되는 인공 신경망 대신 간략식에 대한 연산을 수행하는 연산 모듈을 이용할 수 있다.

[0062] 특히 학습 모듈(540)에 의한 학습 과정에서 인공 신경망의 활성화 함수가 미리 활성 함수 근사 모듈(530)에 의해 근사 다항식으로 치환되어 학습될 수 있으므로, 인공 신경망의 학습과 인공 신경망이 전환된 간략식 사이의 오차가 발생하는 것을 방지될 수 있다. 결과적으로 인공 신경망을 대체하여 간략식을 연산하는 연산 모듈을 이용함에 따라 적은 연산량으로 고속으로 인공 신경망과 동일한 결과를 도출할 수 있게 되며, 더 적은 메모리 용

량으로 연산이 수행될 수 있다.

[0063] 도 6은 본 발명의 일 실시예에 따른 인공 신경망 수학적 경량화 방법을 나타낸다.

[0064] 도 5를 참조하여, 도 6의 인공 신경망 수학적 경량화 방법을 설명하면, 우선 요구되는 인공 신경망을 구성한다(S10). 이때 인공 신경망은 지정된 신경망 설정값에 의해 구성되어진다.

[0065] 그리고 구성된 인공 신경망에서 근사화가 요구되는 활성화 함수(f)가 존재하는지 판별한다(S20). 만일 근사화가 요구되는 활성화 함수(f)가 존재하는 것으로 판별되면, 해당 활성화 함수(f)를 테일러 근사 기법으로 근사하여 근사화 함수로 대체한다(S30).

[0066] 이후 구성된 인공 신경망을 기지정된 방식으로 학습시킨다(S40). 인공 신경망에 대한 학습이 완료되면, 학습에 의해 최종 업데이트된 가중치 행렬(W)과 바이어스 행렬(B)을 확인한다(S50). 그리고 인공 신경망의 각 레이어에서 확인된 가중치 행렬(W)과 바이어스 행렬(B) 및 활성화 함수(f)를 기반으로 인공 신경망 전체에서 수행되는 연산에 대한 연산식을 추출한다(S60).

[0067] 연산식이 추출되면, 추출된 연산식을 전개하고, 전개된 연산식을 수학적으로 다항식 구조로 간략화하여 간략화된 연산식을 획득한다(S70).

[0068] 도 7은 도 6의 인공 신경망 수학적 경량화 방법을 수행하기 위한 컴퓨팅 장치의 예를 나타낸다.

[0069] 도 7을 참조하면, 본 발명의 일 실시 예에 따른 컴퓨팅 장치는 프로세서(710) 및 메모리(720)를 포함할 수 있다. 프로세서(710)는 MPU(micro processing unit), CPU(central processing unit)등으로 구현될 수 있다. 그리고 도 5에 도시된 인공 신경망 수학적 경량화 장치의 구조 설정 모듈(510), 신경망 모듈(520), 활성 함수 근사 모듈(530), 학습 모듈(540), 연산식 추출 모듈(550) 및 간략식 전환 모듈(560) 각각은 프로세서(710) 내에 구현되는 하드웨어 또는 프로세서(710) 내에서 실행되는 소프트웨어 모듈로 구현될 수 있다.

[0070] 메모리(720)는 프로세서(710)에서 동작을 수행하기 위한 각종 데이터를 저장하여, 프로세서(710)로 저장된 데이터를 전달하거나, 프로세서(710)에서 인가되는 데이터를 저장한다. 메모리(720)는 인공 신경망의 학습 시에 프로세서(710)에서 인공 신경망의 각 레이어의 연산 결과가 도출되면, 도출된 연산 결과를 임시로 저장하고, 프로세서(710)에서 다음 레이어에 대한 연산이 수행될 때, 저장된 연산 결과를 프로세서(710)로 전달할 수 있다.

[0071] 본 발명에 따른 방법은 컴퓨터에서 실행시키기 위한 매체에 저장된 컴퓨터 프로그램으로 구현될 수 있다. 여기서 컴퓨터 판독가능 매체는 컴퓨터에 의해 액세스 될 수 있는 임의의 가용 매체일 수 있고, 또한 컴퓨터 저장 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함하며, ROM(판독 전용 메모리), RAM(랜덤 액세스 메모리), CD(컴팩트 디스크)-ROM, DVD(디지털 비디오 디스크)-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등을 포함할 수 있다.

[0072] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다.

[0073] 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

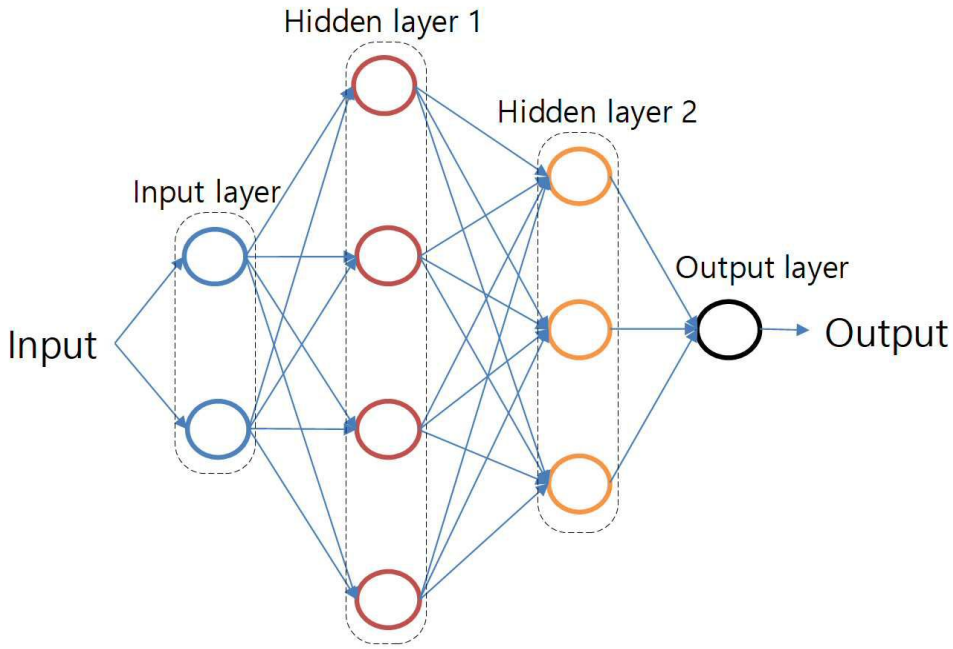
[0074] 510: 구조 설정 모듈 520: 신경망 모듈

530: 활성 함수 근사 모듈 540: 학습 모듈

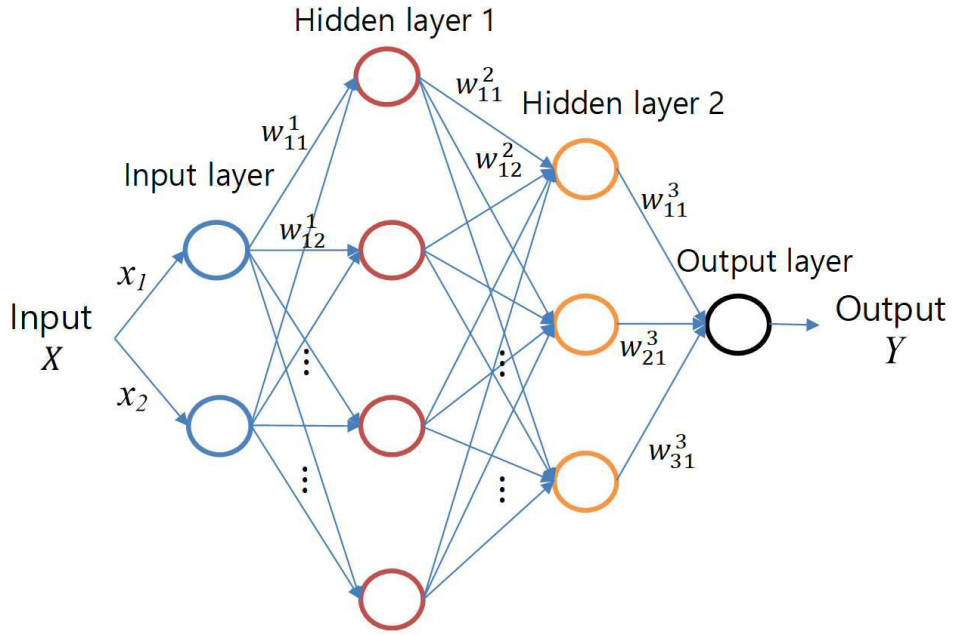
550: 연산식 추출 모듈 560: 간략식 전환 모듈

도면

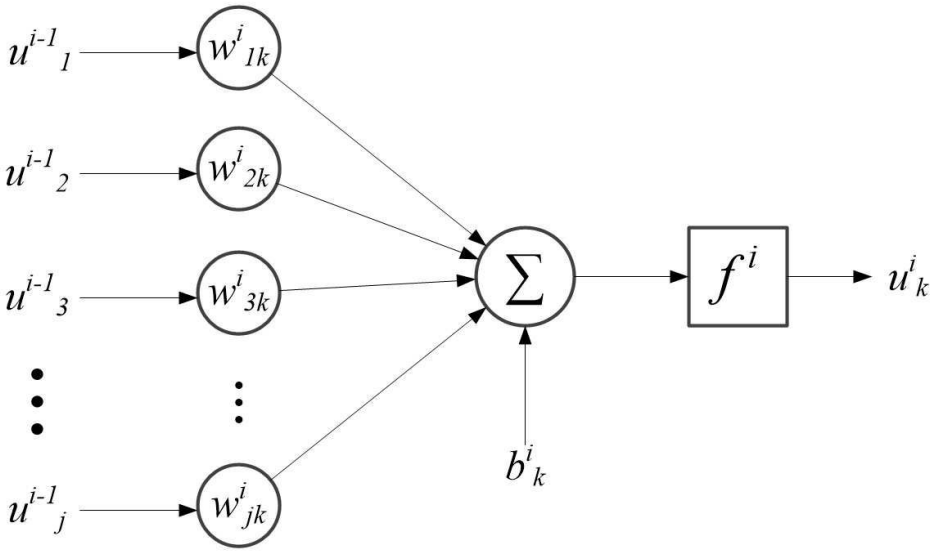
도면1


Hidden layer 1
Hidden layer 2
Input layer
Output layer
Input
Output

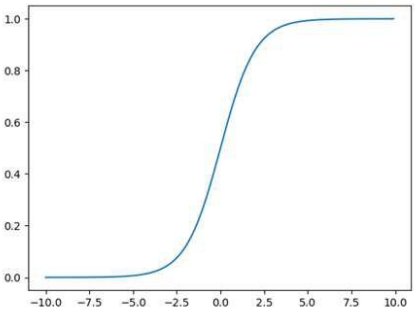
도면2


Hidden layer 1
Hidden layer 2
Input layer
Output layer
Input
X
Output
Y
x_1
x_2
w_{11}^1
w_{12}^1
w_{11}^2
w_{12}^2
w_{11}^3
w_{21}^3
w_{31}^3

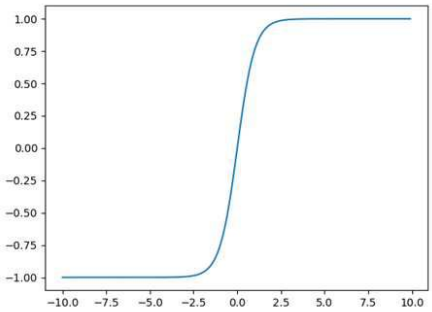
도면3


$u^{i-1}{}_1$
$u^{i-1}{}_2$
$u^{i-1}{}_3$
$u^{i-1}{}_j$
$w^i{}_{1k}$
$w^i{}_{2k}$
$w^i{}_{3k}$
$w^i{}_{jk}$
Σ
f^i
$u^i{}_k$
$b^i{}_k$

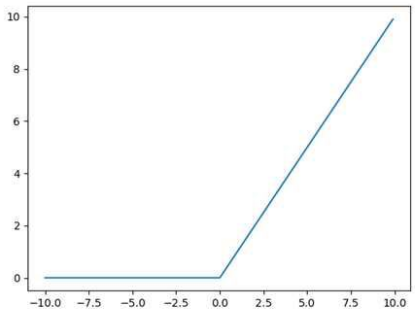
도면4


1.0
0.8
0.6
0.4
0.2
0.0
−10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0 7.5 10.0

(a)

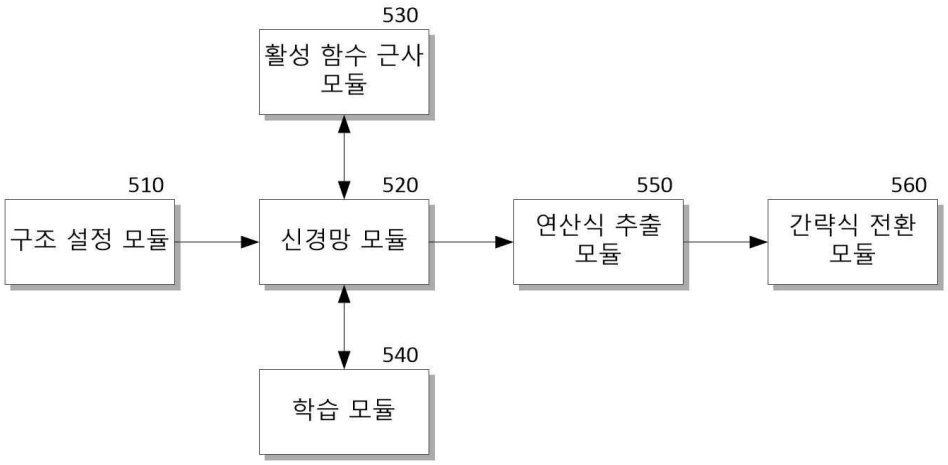

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0 7.5 10.0

(b)

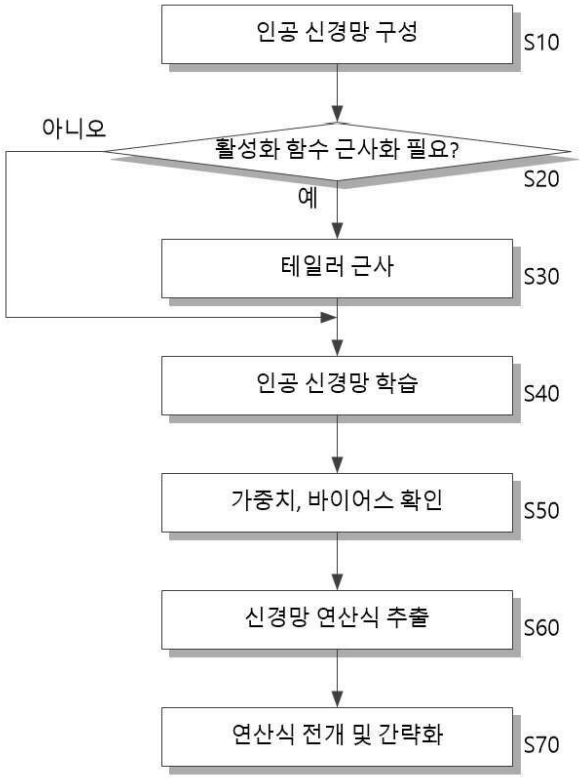

10
8
6
4
2
0
−10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0 7.5 10.0

(c)

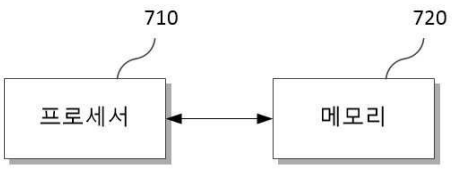
도면5


530
활성 함수 근사 모듈
510
구조 설정 모듈
520
신경망 모듈
550
연산식 추출 모듈
560
간략식 전환 모듈
540
학습 모듈

도면6


인공 신경망 구성
S10
아니오
활성화 함수 근사화 필요?
S20
예
테일러 근사
S30
인공 신경망 학습
S40
가중치, 바이어스 확인
S50
신경망 연산식 추출
S60
연산식 전개 및 간략화
S70

도면7


710
720
프로세서
메모리