



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2022년04월14일

(11) 등록번호 10-2386806

(24) 등록일자 2022년04월11일

(51) 국제특허분류(Int. Cl.)
G06N 3/04 (2006.01) G06N 3/08 (2006.01)(52) CPC특허분류
G06N 3/0445 (2013.01)
G06N 3/0454 (2013.01)

(21) 출원번호 10-2019-7012084

(22) 출원일자(국제) 2017년10월27일

심사청구일자 2019년04월25일

(85) 번역문제출일자 2019년04월25일

(65) 공개번호 10-2019-0052143

(43) 공개일자 2019년05월15일

(86) 국제출원번호 PCT/US2017/058760

(87) 국제공개번호 WO 2018/081563

국제공개일자 2018년05월03일

(30) 우선권주장

62/414,300 2016년10월28일 미국(US)

(56) 선행기술조사문헌

“Neural Architecture Search With
Reinforcement Learning”, ARXIV.ORG,
2017.02.15.“Learning to Compose Neural Networks for
Question Answering”, ARXIV.ORG, 2016.06.07.“Learning to learn by gradient descent”,
ARXIV.ORG, 14 June 2016.

(73) 특허권자

구글 엘엘씨

미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이
1600 (우:94043)

(72) 발명자

조프 바렛

미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이
1600 (우:94043)

리 꾸옥 브이.

미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이
1600 (우:94043)

(74) 대리인

박장원

전체 청구항 수 : 총 22 항

심사관 : 박승철

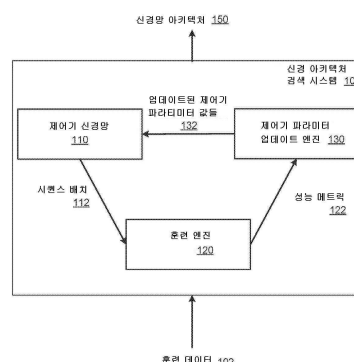
(54) 발명의 명칭 신경 아키텍처 검색

(57) 요약

신경망 아키텍처를 결정하기 위한, 컴퓨터 저장 매체상에 인코딩된 컴퓨터 프로그램을 포함하는 방법, 시스템 및 장치가 개시된다. 방법들 중 하나는, 제어기 신경망을 사용하여 출력 시퀀스들의 배치를 생성하는 단계 -상기 배치 내의 각각의 출력 시퀀스는 특정 신경망 태스크를 수행하도록 구성된 차일드 신경망의 각 아키텍처를 정의함

(뒷면에 계속)

대표도 - 도1



-; 상기 배치의 각 출력 시퀀스에 대해: 상기 출력 시퀀스에 의해 정의된 상기 아키텍처를 갖는 상기 차일드 신경망의 각 인스턴스를 훈련시키는 단계; 상기 특정 신경망 태스크에 대한 차일드 신경망의 상기 훈련된 인스턴스의 성능을 평가하여 상기 특정 신경망 태스크에 대한 상기 차일드 신경망의 상기 훈련된 인스턴스에 대한 성능 메트릭을 결정하는 단계; 그리고 상기 차일드 신경망의 상기 훈련된 인스턴스에 대한 상기 성능 메트릭을 사용하여 상기 제어기 신경망의 제어기 파라미터들의 현재 값들을 조정하는 단계를 포함한다.

(52) CPC특허분류

G06N 3/08 (2013.01)

명세서

청구범위

청구항 1

컴퓨터 시스템에 의해 각 단계가 수행되는 방법으로서,

복수의 제어기 파라미터들을 갖는 제어기 신경망을 사용하여, 상기 제어기 파라미터들의 현재 값들에 따라 출력 시퀀스들의 배치를 생성하는 단계 -상기 배치 내의 각각의 출력 시퀀스는 특정 신경망 태스크를 수행하도록 구성된 차일드 신경망의 각 아키텍처를 정의함 -;

상기 배치의 각 출력 시퀀스에 대해:

상기 특정 신경망 태스크를 수행하기 위해 상기 출력 시퀀스에 의해 정의된 상기 아키텍처를 갖는 상기 차일드 신경망의 각 인스턴스를 훈련시키는 단계;

상기 특정 신경망 태스크에 대한 차일드 신경망의 상기 훈련된 인스턴스의 성능을 평가하여 상기 특정 신경망 태스크에 대한 상기 차일드 신경망의 상기 훈련된 인스턴스에 대한 성능 메트릭을 결정하는 단계; 그리고

상기 차일드 신경망의 상기 훈련된 인스턴스에 대한 상기 성능 메트릭을 사용하여 상기 제어기 신경망의 제어기 파라미터들의 현재 값들을 조정하는 단계를 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 2

제1항에 있어서,

상기 차일드 신경망의 훈련된 인스턴스에 대한 성능 메트릭을 사용하여 상기 제어기 신경망의 제어기 파라미터들의 현재 값들을 조정하는 단계는,

강화 학습(reinforcement learning) 기술을 사용하여 증가된 성능 메트릭을 갖는 차일드 신경망을 초래하는 출력 시퀀스들을 생성하도록 상기 제어기 신경망을 훈련시키는 단계를 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 3

제2항에 있어서, 상기 강화 학습 기술은 PG(policy gradient) 기술인 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 4

제3항에 있어서, 상기 강화 학습 기술은 강화(REINFORCE) 기술인 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 5

제1항에 있어서, 각각의 출력 시퀀스는 복수의 시간 단계들 각각에서 상기 차일드 신경망의 각 하이퍼파라미터에 대한 값을 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 6

제5항에 있어서, 상기 제어기 신경망은 순환 신경망이고,

상기 순환 신경망은,

소정의 출력 시퀀스에 대해 그리고 각각의 시간 단계에서, 상기 소정의 출력 시퀀스의 선행 시간 단계에서 하이퍼파라미터의 값을 입력으로서 수신하고, 상기 입력을 처리하여 상기 순환 신경망의 현재 은닉 상태를 업데이트

하도록 구성된 하나 이상의 순환 신경망 계층들;

각각의 시간 단계에 대한 각각의 출력층을 포함하며,

각각의 출력층은, 소정의 출력 시퀀스에 대해, 시간 단계에서 상기 업데이트된 은닉 상태를 포함하는 출력층 입력을 수신하고, 상기 시간 단계에서 상기 하이퍼파라미터의 가능한 값들에 대한 스코어 분포를 정의하는 시간 단계에 대한 출력을 생성하도록 구성되는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 7

제6항에 있어서,

복수의 제어기 파라미터들을 갖는 제어기 신경망을 사용하여, 상기 제어기 파라미터들의 현재 값들에 따라 출력 시퀀스들의 배치를 생성하는 단계는,

상기 배치 내의 각각의 출력 시퀀스 및 상기 복수의 시간 단계들 각각에 대해,

상기 시간 단계에서 상기 하이퍼파라미터의 가능한 값들에 대한 스코어 분포를 정의하는 상기 시간 단계에 대한 출력을 생성하도록 상기 출력 시퀀스의 선행 시간 단계에서 상기 하이퍼파라미터의 값을 상기 제어기 신경망에 입력으로서 제공하는 단계; 그리고

상기 스코어 분포에 따라 가능한 값들로부터 샘플링하여 상기 출력 시퀀스의 시간 단계에서 하이퍼파라미터의 값을 결정하는 단계를 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 8

제5항에 있어서,

상기 차일드 신경망은 콘볼루션 신경망이고, 상기 하이퍼파라미터들은 상기 차일드 신경망의 각각의 콘볼루션 신경망 계층에 대한 하이퍼파라미터들을 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 9

제8항에 있어서,

상기 콘볼루션 신경망 계층들 각각에 대한 하이퍼파라미터들은,

복수의 필터들, 각 필터에 대한 필터 높이, 각 필터에 대한 필터 폭, 각 필터에 대한 스트라이드(stride) 높이, 각 필터에 대한 스트라이드 폭 중에서 하나 이상을 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 10

제5항에 있어서,

상기 차일드 신경망은 복수의 계층 유형들을 포함하고, 상기 하이퍼파라미터들은, 각 계층에 대해, 상기 계층의 유형에 대응하는 값을 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 11

제6항에 있어서,

하나 이상의 순환 신경망 계층들 각각에 대해, 상기 하이퍼파라미터들은 스킵 연결 하이퍼파라미터를 포함하며, 상기 스킵 연결 하이퍼파라미터는 이전의 어느 계층들이 상기 계층에 대한 스킵 연결을 갖는지를 정의하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 12

제11항에 있어서,

상기 복수의 시간 단계들은 하이퍼파라미터가 스킵 연결 하이퍼파라미터인 상기 하나 이상의 계층들 각각에 대한 각

각의 앵커 포인트 시간 단계를 포함하고,

현재 계층에 대한 앵커 포인트 시간 단계에 대해, 상기 출력층은 상기 차일드 신경망의 현재 계층 이전의 각 계층에 대응하는 각 노드를 포함하며,

각 노드는, 파라미터들의 세트의 현재 값들에 따라 이전 계층에 대한 상기 앵커 포인트 시간 단계에 대한 상기 업데이트된 은닉 상태 및 상기 앵커 포인트 시간 단계에 대한 상기 업데이트된 은닉 상태를 처리하고, 상기 이전 계층이 차일드 신경망의 현재 계층에 연결될 우도를 나타내는 스코어를 생성하도록 구성되는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 13

제1항에 있어서,

상기 차일드 신경망은 순환 신경망이고,

상기 출력 시퀀스는 상기 순환 신경망에서의 순환 셀에 대한 아키텍처를 정의하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 14

제13항에 있어서,

상기 출력 시퀀스는 상기 순환 셀에 의해 수행되는 계산을 나타내는 계산 단계들의 트리에서 각 노드에 대한 각 계산 단계를 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 15

제14항에 있어서,

상기 트리 내의 각 노드는 2개의 입력을 병합하여 출력을 생성하고,

각각의 노드에 대해, 상기 출력 시퀀스는 상기 2개의 입력을 결합하는 결합 방법을 식별하는 데이터 및 상기 출력을 생성하도록 상기 2개의 입력의 결합에 적용되는 활성화 함수를 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 16

제14항에 있어서,

상기 출력 시퀀스는 상기 순환 셀의 메모리 상태가 상기 셀에 어떻게 주입되는지를 정의하는 값들을 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 17

제1항에 있어서,

제어기 파라미터들의 조정된 값들에 따라, 차일드 신경망의 최종 아키텍처를 정의하는 최종 출력 시퀀스를 생성하는 단계를 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 18

제17항에 있어서,

상기 최종 아키텍처를 갖는 차일드 신경망을 사용하여 수신된 네트워크 입력들을 처리함으로써 상기 수신된 네트워크 입력들에 대한 특정 신경망 태스크를 수행하는 단계를 포함하는 것을 특징으로 하는 컴퓨터 시스템에 의해 각 단계가 수행되는 방법.

청구항 19

하나 이상의 컴퓨터들 및 상기 하나 이상의 컴퓨터들에 의해 실행될 때 상기 하나 이상의 컴퓨터들로 하여금 제 1항 내지 제18항 중 어느 한 항의 각각의 방법의 동작들을 수행하게 하는 명령어들을 저장하는 하나 이상의 저

장 장치들을 포함하는 시스템.

청구항 20

하나 이상의 컴퓨터들에 의해 실행될 때 상기 하나 이상의 컴퓨터들로 하여금 제1항 내지 제18항 중 어느 한 항의 각각의 방법의 동작들을 수행하게 하는 명령어들을 저장하는 하나 이상의 컴퓨터 저장 매체.

청구항 21

시스템으로서,

하나 이상의 컴퓨터들에 의해 구현되는 제어기 신경망 -상기 제어기 신경망은 순환 신경망이며; 상기 순환 신경망은, 특정 신경망 태스크를 수행하도록 구성된 차일드 신경망의 아키텍처를 정의하는 하이퍼파라미터 값들의 출력 시퀀스의 각 시간 단계에서, 상기 출력 시퀀스에서 선행 시간 단계에 대응하는 상기 차일드 신경망의 하이퍼파라미터의 값을 입력으로서 수신하고, 상기 입력을 처리하여 상기 순환 신경망의 현재 은닉 상태를 업데이트하도록 구성된 하나 이상의 순환 신경망 계층들을 포함하며 -;

상기 출력 시퀀스의 각 시간 단계에 대응하는 각각의 출력층을 포함하며,

각각의 출력층은, 상기 시간 단계에서 상기 업데이트된 은닉 상태를 포함하는 출력층 입력을 수신하고, 상기 시간 단계에 대응하는 상기 차일드 신경망의 하이퍼파라미터의 가능한 값들에 대한 스코어 분포를 정의하는 시간 단계에 대한 출력을 생성하도록 구성되는 것을 특징으로 하는 시스템.

청구항 22

하나 이상의 컴퓨터들에 의해 실행될 때 상기 하나 이상의 컴퓨터들로 하여금 제21항의 시스템을 구현하게 하는 명령어들을 저장하는 하나 이상의 컴퓨터 저장 매체.

발명의 설명

기술 분야

[0001] 본 명세서는 신경망 아키텍처를 수정하는 것에 관한 것이다.

배경 기술

[0002] 신경망은 수신된 입력에 대한 출력을 예측하기 위해 하나 이상의 비선형 유닛 계층을 사용하는 기계 학습 모델이다. 일부 신경망은 출력층 외에도 하나 이상의 은닉층을 포함한다. 각각의 은닉층의 출력은 네트워크의 다음 계층, 즉 다음 은닉층 또는 출력층에 대한 입력으로 사용된다. 네트워크의 각 계층은 각각의 파라미터 세트의 현재 값에 따라 수신된 입력으로부터 출력을 생성한다.

[0003] 일부 신경망은 순환 신경망이다. 순환 신경망은 입력 시퀀스를 수신하고 입력 시퀀스로부터 출력 시퀀스를 생성하는 신경망이다. 특히, 순환 신경망은 현재 시간 단계에서 출력을 계산할 때 이전 시간 단계에서 네트워크의 내부 상태 중 일부 또는 전부를 사용할 수 있다. 순환 신경망의 예는 하나 이상의 LSTM(long short term) 메모리 블록을 포함하는 LSTM 신경망이다. 각각의 LSTM 메모리 블록은, 입력 게이트, 삭제 게이트(forget gate), 셀이 예를 들어 현재 활성화를 생성하거나 LSTM 신경망의 다른 컴포넌트에 제공되는데 사용하기 위해 셀에 대한 이전 상태를 저장할 수 있게 하는 출력 게이트를 각각 포함하는 하나 이상의 셀을 포함할 수 있다.

발명의 내용

해결하려는 과제

[0004] 본 명세서는 하나 이상의 위치에 있는 하나 이상의 컴퓨터상의 컴퓨터 프로그램으로 구현된 시스템이 제어기 신경망을 사용하여 특정 신경망 태스크를 수행하도록 구성된 차일드(child) 신경망을 위한 아키텍처(구조)를 결정하는 방법을 설명한다.

발명의 효과

[0005] 본 명세서에서 설명된 요지의 특정 실시 예는 다음의 장점 중 하나 이상을 실현하도록 구현될 수 있다. 시스템

은 효율적이고 자동적으로, 즉 사용자 개입없이, 특정 태스크에 대해 고성능 신경망을 초래할 신경망 아키텍처를 선택할 수 있다. 이 시스템은 특정 태스크에 적합한 새로운 신경망 아키텍처를 효과적으로 결정할 수 있으므로 결과적인 차일드 신경망이 태스크에 대해 개선된 성능을 가질 수 있게 한다. 상기 시스템은 보강 학습을 통해 제어기 신경망을 학습하여 아키텍처(구조)를 결정하기 때문에, 시스템은 특정 태스크에 적합한 차일드 신경망을 위한 아키텍처를 식별하기 위해 가능한 아키텍처의 넓은 공간을 효과적으로 탐구(explore)할 수 있다.

[0006] 본 명세서에 기술된 요지의 하나 이상의 실시 예의 세부 사항은 첨부된 도면 및 이하의 설명에서 설명된다. 요지의 다른 특징, 양태 및 장점은 상세한 설명, 도면 및 청구 범위로부터 명백해질 것이다.

도면의 간단한 설명

[0007] 도 1은 예시적인 신경 아키텍처 검색 시스템을 도시한다.

도 2a는 출력 시퀀스를 생성하는 제어기 신경망의 예를 나타낸 도면이다.

도 2b는 스킵 연결을 포함하는 아키텍처를 정의하는 출력 시퀀스를 생성하는 제어기 신경망의 예를 도시한 도면이다.

도 2c는 순환 셀에 대한 아키텍처를 정의하는 출력 시퀀스를 생성하는 제어기 신경망의 일례를 나타낸 도이다.

도 3은 제어기 파라미터의 현재 값을 업데이트하기 위한 예시적인 프로세스의 흐름도이다.

다양한 도면에서 동일한 참조 번호 및 명칭은 동일한 구성요소를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0008] 본 명세서는 제어기 신경망을 사용하여 특정 신경망 태스크를 수행하도록 구성된 차일드 신경망용 아키텍처를 결정하는 하나 이상의 위치에 있는 하나 이상의 컴퓨터 상의 컴퓨터 프로그램으로 구현되는 시스템을 설명한다.

[0009] 차일드 신경망은 모든 유형의 디지털 데이터 입력을 수신하고, 입력에 따라 스코어(점수), 분류 또는 회귀 출력의 모든 유형을 생성하도록 구성될 수 있다.

[0010] 예를 들어, 차일드 신경망에 대한 입력이 이미지로부터 추출된 특징 또는 이미지인 경우, 소정의 이미지에 대해 차일드 신경망에 의해 생성된 출력은 객체 카테고리들의 세트의 각각에 대한 스코어일 수 있으며, 각각의 스코어는 이미지가 카테고리에 속하는 객체의 이미지를 포함하는 추정된 우도를 나타낸다.

[0011] 또 다른 예로서, 차일드 신경망에 대한 입력이 인터넷 자원(예: 웹 페이지), 문서 또는 인터넷 자원, 문서 또는 그 문서의 일부에서 추출된 특징 또는 문서의 일부인 경우, 소정의 인터넷 자원, 문서, 또는 그 문서의 일부분에 대한 차일드 신경망에 의해 생성된 출력은 토픽들의 세트 각각에 대한 스코어일 수 있고, 각 스코어는 인터넷 자원, 문서 또는 문서 부분이 토픽에 관한 것으로 추정되는 우도를 나타낸다.

[0012] 또 다른 예로서, 차일드 신경망에 대한 입력이 특정 광고에 대한 노출 컨텍스트의 특징인 경우, 차일드 신경망에 의해 생성된 출력은 특정 광고가 클릭될 것으로 추정되는 우도를 나타내는 스코어일 수 있다.

[0013] 또 다른 예로서, 차일드 신경망에 대한 입력이 사용자에게 개인화된 추천의 특징, 예를 들어, 추천에 대한 컨텍스트를 특징짓는 특징, 예컨대 사용자에게 의해 취해진 이전 액션을 특징짓는 특징인 경우, 차일드 신경망에 의해 생성된 출력은 콘텐츠 아이템들의 세트 각각에 대한 스코어일 수 있고, 각 스코어는 사용자가 콘텐츠 아이템을 추천하는 것에 호의적으로 응답할 것으로 예상되는 우도를 나타낸다.

[0014] 또 다른 예로서, 차일드 신경망에 대한 입력이 하나의 언어로 된 텍스트의 시퀀스라면, 차일드 신경망에 의해 생성된 출력은 다른 언어로 된 텍스트의 부분들(pieces) 세트의 각각에 대한 스코어일 수 있으며, 각각의 스코어는 다른 언어의 텍스트의 부분들이 다른 언어로의 입력 텍스트의 적절한 번역인 것으로 추정된 우도를 나타낸다.

[0015] 또 다른 예로서, 차일드 신경망에 대한 입력이 발언(발화)을 나타내는 시퀀스라면, 차일드 신경망에 의해 생성된 출력은 텍스트의 부분들의 세트 각각에 대한 스코어 일 수 있고, 각각의 스코어는 텍스트 부분이 발화에 대한 정확한 전사(transcript)일 것으로 추정되는 우도를 나타낸다.

[0016] 도 1은 예시적인 신경 아키텍처 검색 시스템(100)을 도시한다. 신경 아키텍처 검색 시스템(100)은 하나 이상의 위치에서 하나 이상의 컴퓨터상의 컴퓨터 프로그램으로 구현되는 시스템의 예이며, 여기서 설명되는 시스템, 컴포넌트 및 기술이 구현될 수 있다.

- [0017] 신경 아키텍처 검색 시스템(100)은 특정 태스크를 수행하기 위해 신경망을 훈련시키기 위한 훈련 데이터(102) 및 특정 태스크에 대해 신경망의 성능을 평가하기 위한 유효성(validation) 세트(104)를 획득하고, 훈련 데이터(102) 및 유효성 세트(104)를 사용하여 특정 태스크를 수행하도록 구성된 차일드 신경망에 대한 아키텍처를 결정하는 시스템이다. 상기 아키텍처는 차일드 신경망에서의 계층(layer)의 수, 각 계층에 의해 수행되는 동작, 및 차일드 신경망에서의 계층들 사이의 연결성, 즉 어느 계층이 차일드 신경망에서 다른 계층으로부터 입력을 수신하는지를 정의한다.
- [0018] 일반적으로, 훈련 데이터(102) 및 유효성 세트(104) 모두는 신경망 입력들의 세트 및 각각의 네트워크 입력에 대해 특정 태스크를 수행하기 위해 차일드 신경망에 의해 생성되어야 하는 각각의 목표 출력을 포함한다. 예를 들어, 훈련 데이터(102) 및 유효성 세트(104)를 생성하기 위해 더 큰 세트의 훈련 데이터가 랜덤하게 분할될 수 있다.
- [0019] 시스템(100)은 다양한 방식들 중 임의의 방식으로 훈련 데이터(102) 및 유효성 세트(104)를 수신할 수 있다. 예를 들어, 시스템(100)은 시스템(100)에 의해 이용 가능한 애플리케이션 프로그래밍 인터페이스(API)를 사용하여 데이터 통신 네트워크를 통해 시스템의 원격 사용자로부터 업로드로서 훈련 데이터를 수신할 수 있고, 업로드된 데이터를 무작위로 훈련 데이터(102)와 유효성 세트(104)로 나눈다(분할). 또 다른 예로서, 시스템(100)은 시스템(100)에 의해 이미 유지된 데이터가 신경망을 훈련시키기 위해 사용되어야 하는지를 지정하는 사용자로부터의 입력을 수신할 수 있고, 이어서 지정된 데이터를 훈련 데이터(102) 및 유효성 세트(104)로 분할한다.
- [0020] 신경 아키텍처 검색 시스템(100)은 제어기 신경망(110), 훈련 엔진(120) 및 제어기 파라미터 업데이트 엔진(130)을 포함한다.
- [0021] 제어기 신경망(110)은 본 명세서에서 "제어기 파라미터"로 언급되고 제어기 파라미터에 따라 출력 시퀀스를 생성하도록 구성된 파라미터를 갖는 신경망이다. 제어기 신경망(110)에 의해 생성된 각각의 출력 시퀀스는 차일드 신경망에 대한 각각의 가능한 아키텍처를 정의한다.
- [0022] 특히, 각각의 출력 시퀀스는 복수의 시간 단계 각각에서 각각의 출력을 포함하고, 출력 시퀀스의 각각의 시간 단계는 차일드 신경망의 아키텍처의 상이한 하이퍼 파라미터에 대응한다. 따라서, 각각의 출력 시퀀스는 각 시간 단계에서 해당 하이퍼파라미터의 각 값을 포함한다. 집합적으로, 소정의 출력 시퀀스에서 하이퍼파라미터의 값은 차일드 신경망에 대한 아키텍처를 정의한다. 일반적으로, 하이퍼파라미터는 차일드 신경망의 훈련이 시작되기 전에 설정되고 차일드 신경망에 의해 수행되는 동작에 영향을 미치는 값이다. 출력 시퀀스 및 가능한 하이퍼파라미터는 2a-2c를 참조하여 아래에서보다 상세히 설명된다.
- [0023] 일반적으로, 시스템(100)은 제어기 신경망(110)을 훈련시켜 제어기 파라미터의 값을 조정함으로써 차일드 신경망에 대한 아키텍처를 결정한다.
- [0024] 특히, 훈련 과정의 반복 동안, 시스템(100)은 제어기 파라미터의 현재 값에 따라 제어기 신경망(110)을 사용하여 시퀀스의 배치(batch)(112)를 생성한다. 배치(112)의 각 출력 시퀀스에 대해, 훈련 엔진(120)은 훈련 데이터(102)에 대한 출력 시퀀스에 의해 정의된 아키텍처를 갖는 차일드 신경망의 인스턴스를 훈련시키고, 유효성 세트(104)에서 훈련된 인스턴스의 성능을 평가한다. 그 다음, 제어기 파라미터 업데이트 엔진(130)은 배치(112) 내의 출력 시퀀스에 대한 평가 결과를 사용하여 제어기 파라미터의 현재 값을 업데이트하여 태스크에 대한 제어기 신경망(110)에 의해 생성된 출력 시퀀스에 의해 정의된 아키텍처의 예상된 성능을 향상시킨다. 훈련된 인스턴스의 성능을 평가하고 제어기 파라미터의 현재 값을 업데이트하는 것은 도 3을 참조하여 이하에서보다 상세하게 설명된다.
- [0025] 이러한 방식으로 제어기 파라미터의 값을 반복적으로 업데이트함으로써, 시스템(100)은 제어기 신경망(110)을 훈련시켜, 특정 태스크에 대해 증가된 성능을 갖는 차일드 신경망, 즉 제어기 신경망(110)에 의해 제안된 아키텍처의 유효성 세트(104)상의 예상된 정확성을 최대화하는 출력 시퀀스를 생성할 수 있다.
- [0026] 일단 제어기 신경망(110)이 훈련되면, 시스템(100)은 차일드 신경망의 최종 아키텍처로서 유효성 세트(104)에서 최상으로 수행된 아키텍처를 선택할 수 있거나, 제어기 파라미터의 훈련된 값에 따라 새로운 출력 시퀀스를 생성할 수 있고 새로운 출력 시퀀스에 의해 정의된 아키텍처를 차일드 신경망의 최종 아키텍처로 사용할 수 있다.
- [0027] 그 후, 신경망 검색 시스템(100)은 차일드 신경망의 아키텍처, 즉 차일드 신경망의 일부인 계층을 지정하는 데이터, 계층들 간 연결성 및 계층들에 의해 수행된 동작을 지정하는 아키텍처 데이터(150)를 출력할 수 있다. 예를 들어, 신경망 검색 시스템(100)은 훈련 데이터를 제출한 사용자에게 아키텍처 데이터(150)를 출력할 수

있다. 일부 경우에, 데이터(150)는 또한 아키텍처를 갖는 차일드 신경망의 훈련된 인스턴스의 훈련으로부터 차일드 신경망의 파라미터의 훈련된 값을 포함한다.

[0028] 일부 구현 예에서, 아키텍처 데이터(150)를 출력하는 대신에 또는 그에 추가하여, 시스템(100)은, 예를 들어, 아키텍처를 갖는 차일드 신경망의 인스턴스를 훈련한 결과 생성된 파라미터 값을 스크래치(scratch)로부터 또는 미세조정하기 위해 상기 결정된 아키텍처를 갖는 신경망의 인스턴스를 훈련시키고, 그 다음 훈련된 신경망을 사용하여, 예를 들어 시스템에 의해 제공된 API를 통해 사용자에게 의해 수신된 요청을 처리한다. 즉, 시스템(100)은 처리될 입력을 수신하고, 훈련된 차일드 신경망을 사용하여 입력을 처리할 수 있고, 훈련된 신경망에 의해 생성된 출력 또는 수신된 입력에 응답하여 생성된 출력으로부터 유도된 데이터를 제공한다.

[0029] 일부 구현 예에서, 시스템(100)은 분산 방식으로 제어기 신경망을 훈련시킨다. 즉, 시스템(100)은 제어기 신경망의 복수의 레플리카(replicas)를 포함한다. 훈련이 배포되는 이러한 구현예 중 일부에서, 각 레플리카는 레플리카가 출력한 출력 시퀀스의 배치들에 대한 성능 메트릭과 그 성능 메트릭을 사용하여 제어기 파라미터에 대한 업데이트를 결정하는 전용 제어기 파라미터 업데이트 엔진을 생성하는 전용 훈련 엔진을 갖는다. 제어기 파라미터 업데이트 엔진이 업데이트를 결정하면, 제어기 파라미터 업데이트 엔진은 업데이트를 모든 제어기 파라미터 업데이트 엔진에 액세스할 수 있는 중앙 파라미터 업데이트 서버로 전송할 수 있다. 중앙 파라미터 업데이트 서버는 서버에 의해 유지되는 제어기 파라미터의 값을 업데이트하고, 그 업데이트된 값을 제어기 파라미터 업데이트 엔진에 송신할 수 있다. 일부 경우에 따라, 복수의 레플리카들 및 이들의 해당 훈련 엔진들 및 파라미터 업데이트 엔진들 각각은 훈련 엔진들 및 파라미터 업데이트 엔진들의 서로 다른 세트로부터 비동기적으로 동작할 수 있다.

[0030] 도 2a는 출력 신경망(110)이 출력 시퀀스를 생성하는 일련의 다이어그램(200)이다.

[0031] 특히, 다이어그램(200)은 출력 시퀀스의 생성 동안 7개의 예시적인 시간 단계(202-214)에 대해 제어기 신경망(110)에 의해 수행되는 처리(프로세싱)를 도시한다. 아래에서보다 상세히 설명되는 바와 같이, 7개의 시간 단계들(202-214) 각각은 신경망 아키텍처의 상이한 하이퍼파라미터에 대응한다.

[0032] 제어기 신경망(110)은 하나 이상의 순환 신경망 계층들, 예를 들어 계층들(220 및 230)을 포함하는 순환 신경망이며, 이는 각각의 시간 단계에 대해 상기 소정의 출력 시퀀스에서 선행 시간 단계에 대응하는 하이퍼파라미터의 값을 입력으로서 수신하고, 상기 입력을 처리하여 상기 순환 신경망의 현재 은닉 상태를 업데이트하도록 구성된다. 예를 들어, 제어기 신경망(110)의 순환 계층들은 LSTM(long-short term memory) 계층 또는 GRU(gated recurrent unit) 계층들일 수 있다. 도 2a의 예에서는, 시간 단계(208)에서, 계층들(220 및 230)은 이전(선행) 시간 단계(206)로부터의 하이퍼파라미터의 값을 입력으로서 수신하고, 업데이트된 은닉 상태(232)를 출력으로서 생성하기 위해 시간 단계(206)로부터 계층들(220 및 230)의 은닉 상태들을 업데이트한다.

[0033] 제어기 신경망(110)은 또한 출력 시퀀스의 각 시간 단계에 대한 각각의 출력 계층, 예를 들어, 시간 단계들(202-214)에 대한 출력 계층들(242-254)을 각각 포함한다. 각각의 출력 계층은 시간 단계에서 업데이트된 은닉 상태를 포함하는 출력 계층 입력을 수신하고, 시간 단계에서 하이퍼파라미터의 가능한 값에 대한 스코어 분포를 정의하는 시간 단계에 대한 출력을 생성하도록 구성된다. 예를 들어, 각 출력 계층은 먼저 해당 하이퍼파라미터의 복수의 가능한 값들에 대한 적절한 차원으로 출력 계층 입력을 프로젝트(project)하고, 그 후 시간 단계에서 하이퍼파라미터에 대한 복수의 가능한 값들 각각에 대한 각각의 스코어를 생성하도록 상기 프로젝트된 출력 계층 입력에 소프트맥스(softmax)를 적용할 수 있다. 예를 들어, 시간 단계(208)에 대한 출력 계층(248)은 은닉 상태(232)를 포함하는 입력을 수신하고, 스트라이드(stride) 높이의 하이퍼파라미터에 대한 복수의 가능한 값들 각각에 대한 각각의 스코어를 생성하도록 구성된다.

[0034] 따라서, 출력 시퀀스에서 소정의 시간 단계에 대한 하이퍼파라미터 값을 생성하기 위해, 시스템(100)은 출력 시퀀스의 이전 시간 단계에서의 하이퍼파라미터의 값을 제어기 신경망에 입력으로서 제공하고, 제어기 신경망은 시간 단계에서 하이퍼파라미터의 가능한 값에 대한 스코어 분포를 정의하는 시간 단계에 대한 출력을 생성한다. 이전 시간 단계가 없기 때문에, 출력 시퀀스의 맨 처음 시간 단계에서, 시스템(100)은 그 대신에 미리 결정된 플레이스홀더(placeholder) 입력을 제공할 수 있다. 그 다음, 시스템(100)은 출력 시퀀스의 시간 단계에서 하이퍼파라미터의 값을 결정하기 위해 스코어 분포에 따라 가능한 값들로부터 샘플링한다. 소정의 하이퍼파라미터가 취할 수 있는 가능한 값은 훈련 전에 고정되어 있으며 복수의 가능한 값들은 다른 하이퍼파라미터들마다 다를 수 있다.

[0035] 일반적으로 소정의 출력 시퀀스에 의해 정의된 아키텍처에 포함될 계층들의 수는 시퀀스를 생성하기 전에 고정

된다. 일부 구현 예에서, 제어기 신경망의 훈련 동안 생성된 출력 시퀀스에 의해 정의된 각각의 아키텍처는 동일한 수의 계층을 갖는다. 다른 구현예에서, 시스템은 훈련이 진행됨에 따라 차일드 신경망에서 계층의 수를 증가시키는 스케줄을 사용한다. 일례로서, 시스템은 훈련 동안 6개의 계층에서 시작하여 1,600개의 샘플마다 하나 이상의 계층으로 깊이(depth)를 증가시킬 수 있다.

[0036] 도 2a의 예에서, 차일드 신경망은 콘볼루션 신경망이고, 하이퍼파라미터들은 차일드 신경망에서 각 콘볼루션 신경망 계층에 대한 하이퍼파라미터들을 포함한다. 특히, 도 2a에서, 시간 단계(202)는 차일드 신경망의 콘볼루션 계층 N-1의 하이퍼 파라미터에 대응하고, 시간 단계들(204-212)은 콘볼루션 계층 N의 하이퍼파라미터들에 대응하고, 시간 단계(214)는 콘볼루션 계층 N + 1의 하이퍼파라미터에 대응한다. 예를 들어, 콘볼루션 계층들은 스택으로 배열될 수 있으며, 계층 N은 계층 N-1에 의해 생성된 출력을 입력으로서 수신하고 계층 N + 1에 입력으로서 제공되는 출력을 생성한다.

[0037] 도 2a의 예에서, 콘볼루션 계층의 경우, 계층에 의해 수행하는 동작들을 정의하는 하이퍼파라미터들은 계층의 필터 수, 각 필터의 필터 높이, 각 필터의 필터 폭, 각 필터를 적용하는 스트라이드(stride) 높이 및 각 필터의 스트라이드 폭이다. 다른 예에서, 이들 중 일부는 제거될 수 있으며, 예를 들어 이러한 하이퍼파라미터들의 특징적인 것들은 고정된 것으로 가정될 수 있고, 다른 하이퍼파라미터들, 예를 들어 활성화 함수의 유형, 콘볼루션이 딜레이션되거나(dilated) 마스크되는지 여부 등이 추가될 수 있거나, 또는 둘 모두가 추가될 수 있다.

[0038] 일례의 구현예에서, 필터 높이에 대한 가능한 값은 [1, 3, 5, 7]이고, 필터 폭에 대한 가능한 값은 [1, 3, 5, 7]이고, 필터의 수에 대한 가능한 값은 [24, 36, 48, 64]이고, 스트라이드 높이와 폭에 대한 가능한 값은 [1, 2, 3]이다.

[0039] 도 2a의 예에서, 차일드 신경망에서의 계층들의 구성, 즉 어떤 계층들이 다른 계층들로부터의 계층들을 수신하는지가 고정된다. 그러나, 다른 예들에서, 상기 하이퍼파라미터들은 차일드 신경망의 계층들 사이의 연결성을 정의하는 하이퍼파라미터들을 포함한다.

[0040] 도 2b는 스킵 연결(skip connections)을 포함하는 아키텍처를 정의하는 출력 시퀀스를 생성하는 제어기 신경망(110)의 일례의 다이어그램(250)이다.

[0041] 특히, 도 2b의 예에서, 차일드 신경망의 하나 이상의 계층에 대해, 하이퍼파라미터는 스킵 연결 하이퍼파라미터를 포함하는데, 이 스킵 연결 하이퍼파라미터는 이전의 어느 계층들이 그 계층에 대한 스킵 연결을 갖는지를 정의한다. 더 구체적으로, 출력 시퀀스의 시간 단계들은 하이퍼미터(hypermeter)가 스킵 연결 하이퍼파라미터, 예를 들어, 계층 N-1에 대한 시간 단계(252) 및 계층 N에 대한 시간 단계(254)인 하나 이상의 계층들 각각에 대한 각 앵커(anchor) 포인트 시간 단계를 포함한다.

[0042] 소정의 계층의 소정의 앵커 포인트 시간 단계에 대한 출력 계층은 차일드 신경망에서 현재 계층보다 이전의 각 계층에 대응하는 개별 노드를 포함한다. 각각의 노드는 (i) 앵커 포인트 단계에 대한 업데이트된 은닉 상태, 및 (ii) 대응하는 이전 계층, 즉 노드에 대응하는 이전 계층의 앵커 포인트 시간 단계에 대한 업데이트된 은닉 상태를, 상기 이전 계층이 차일드 신경망의 현재 계층에 연결될 가능성을 나타내는 스코어를 생성하도록 하는 파라미터들의 세트에 따라 처리하도록 구성된다. 예를 들어, 이전 계층 j에 대응하는 계층 i의 출력 계층에 있는 노드는 수학적 식 1을 만족하는 상기 대응하는 이전 계층에 대한 스코어를 생성할 수 있다.

수학적 식 1

[0043]
$$P(\text{Layer } j \text{ is an input to layer } i) = \text{sigmoid}(v^T \tanh(W_{\text{prev}} * h_j + W_{\text{curr}} * h_i))$$

[0044] 여기서, v^T , W_{prev} , W_{curr} 는 상기 노드에 대한 파라미터들이며, h_j 는 상기 대응하는 이전 계층 j의 앵커 포인트 시간 단계에 대한 업데이트된 은닉 상태이고, h_i 는 계층 i의 앵커 포인트 시간 단계에 대한 업데이트된 은닉 상태이다.

[0045] 그 후, 시스템(100)은 각각의 이전 계층에 대해, 상기 계층이 이전 계층에 대응하는 노드에 의해 생성된 스코어에 따라 "yes" 또는 "no"중 어느 하나를 샘플링함으로써 스킵 연결로 소정의 계층에 연결되어 있는지를 결정한다. 시스템이 다중 계층이 상기 소정의 계층에 연결되어야 한다고 결정하면, 다중 계층들 모두에 의해 생성된

출력들은 소정의 계층에 대한 입력을 생성하기 위해 깊이 차원에서 연결된다. 하나의 계층이 다른 계층과 호환되지 않고 입력 또는 출력이 없는 계층이 포함되지 않은 네트워크에서 스킵 연결이 "컴파일 실패(compilation failures)"를 발생시키지 않도록 하려면, (i) 계층이 입력 계층에 연결되어 있지 않으면 네트워크 입력이 계층의 입력으로 사용되며, (ii) 최종 계층에서, 시스템은 연결되지 않은 모든 계층 출력을 가져 와서 이를 최종 연결 출력을 네트워크의 출력 계층으로 전송하기 전에 연결하며, (iii) 연결될 입력이 다른 크기를 갖는다면, 시스템은 연결될 입력이 동일한 크기를 갖도록 스몰 계층을 제로로 패드(pad)한다.

[0046] 일부 예에서, 차일드 신경망은 복수의 상이한 층 유형을 포함한다. 예를 들어, 차일드 신경망은 다른 유형의 신경망 계층들, 예를 들어 완전히 연결된 계층, 풀링 계층, 깊이 연결 계층, 로컬 콘트라스트(contrast) 정규화, 배치 정규화, 출력 계층(예: softmax 계층 또는 다른 분류기 계층) 등을 포함하는 콘볼루션 신경망이 될 수 있다.

[0047] 이들 예들 중 일부에서, 다른 계층들의 하이퍼파라미터들 및 위치들은 고정되어 있고, 출력 시퀀스는 차일드 신경망의 콘볼루션 신경망 계층들에 대한 하이퍼파라미터 값들만을 포함한다. 예를 들어, 출력 계층의 위치는 차일드 신경망의 마지막 계층으로 고정될 수 있으며, 일부 또는 전부의 콘볼루션 계층들은 배치 정규화 계층들을 뒤따르거나 선행될 수 있다.

[0048] 이 예들 중 다른 예들에서, 출력 시퀀스에 의해 정의된 하이퍼파라미터는 각 계층에 대해, 계층의 유형에 대응하는 값을 포함한다. 서로 다른 유형의 계층에는 서로 다른 하이퍼파라미터가 있으므로 이 예제에서 시스템은 소정의 계층에 대해 어떤 유형의 신경망 계층이 선택되는지에 기초하여 출력 시퀀스의 생성 중에 어떤 하이퍼파라미터가 동적으로 어떤 시간 단계에 해당하는지를 결정한다. 즉, 시스템이 소정의 시간 단계 동안 출력 계층으로 사용하는 출력 계층은 가장 최근에 샘플링된 계층 유형 하이퍼파라미터의 값에 따라 달라진다.

[0049] 일부 예에서, 차일드 신경망은 순환 신경망이다. 이러한 경우, 출력 시퀀스는 순환(recurrent) 셀의 아키텍처를 정의할 수 있으며, 순환 셀은 차일드 신경망 내에서 여러 번 반복되어 신경망을 위한 아키텍처를 생성할 수 있다. 전술한 바와 같이, 일부 경우에는 반복 횟수가 훈련 동안 고정되고, 다른 경우에는 시스템이 훈련이 진행되는 동안 따라 반복 횟수를 증가시킨다.

[0050] 도 2c는 순환 셀에 대한 아키텍처를 정의하는 출력 시퀀스를 생성하는 제어기 신경망(110)의 일례의 다이어그램(270)이다.

[0051] 특히, 도 2c에 도시된 바와 같이, 출력 시퀀스는 순환 셀에 의해 수행된 계산을 나타내는 계산 단계의 트리(tree)의 각각의 노드에 대한 각각의 계산 단계를 포함한다. 순환 셀은, 현재 시간 단계에 대한 입력(즉 시간 단계에서의 차일드 네트워크로의 입력 또는 차일드 네트워크의 다른 컴포넌트에 의해 생성된 출력), 및 이전 시간 단계로부터의 셀의 출력과 같은 2개의 입력을 수신한다. 순환 셀은 이 두 입력을 처리하여 셀 출력을 생성한다. 후술되는 바와 같이, 일부 경우에, 순환 셀은 또한 제3 입력, 즉 메모리 상태를 수신한다.

[0052] 보다 구체적으로, 도 2c에서, 출력 시퀀스는 트리의 3개의 노드, 즉 트리 인덱스(index) 0에서의 하나의 리프 노드, 트리 인덱스 1에서의 다른 하나의 리프 노드, 트리 인덱스 2에서의 내부 노드에 대한 세팅을 정의한다.

[0053] 트리의 각 노드는 두 개의 입력을 병합하여 출력을 생성하고, 각 노드에 대해 출력 시퀀스는 (i) 두 입력을 결합하는 결합(combination) 방법을 식별하는 데이터 및 (ii) 출력을 생성하기 위해 두 입력의 조합에 적용될 활성화 함수를 포함한다. 일반적으로, 셀의 리프 노드는 먼저 셀에 대한 두 개의 입력 각각에 각각의 파라미터 행렬을 적용하지만 내부 노드는 어떠한 파라미터들을 가지지 않는다. 상기 설명한 바와 같이, 결합 방법은 가능한 결합 방법들(예를 들어, 합산(add); 요소별 곱셈(element wise multiply))의 한 세트로부터 선택되고, 상기 활성화 함수는 가능한 활성화 함수들(예를 들어, 아이덴티티(identity); tanh; sigmoid; relu)의 한 세트로부터 선택된다.

[0054] 예를 들어, 트리 인덱스 0의 리프 노드의 경우, 시스템이 결합 함수로 "add"를 선택하고, 활성화 함수로 "tanh"를 선택했기 때문에, 셀의 트리 인덱스 0에 있는 리프 노드는 출력 a^0 을 생성하기 위해 수학식 2의 연산을 수행할 수 있다.

수학식 2

$$a_0 = \tanh(W_1 * x_t + W_2 * h_{t-1})$$

[0055]

[0056]

여기서, W_1 및 W_2 는 노드의 파라미터 행렬이고, x_t 는 시간 단계에서 셀에 대한 입력이고, h_{t-1} 는 이전 시간 단계에서의 셀의 출력이다.

[0057]

트리 인덱스 2에서의 노드에 대해, 셀이 메모리 상태를 갖지 않을 때, 도 2c의 예에서는 노드에 대한 두 개의 입력, 즉 두 리프 노드의 출력이 요소별로 곱해지고, 요소별 시그모이드 함수가 내부 노드의 출력, 즉 셀의 출력을 생성하기 위해 요소별 곱셈의 출력에 적용되도록 특정한다.

[0058]

선택적으로, 셀의 아키텍처는 입력으로서 선행 메모리 상태를 수신하는 것을 포함할 수 있다. 이 경우, 출력 시퀀스는 셀의 메모리 상태가 셀에 주입(injected)되는 방법, 즉 선행 메모리 상태가 어떻게 업데이트되는지, 트리의 다음 노드로 전달되기 전에 선행 메모리 상태를 사용하여 수정된 출력을 갖는 노드를 정의하는 값들을 포함한다.

[0059]

특히, 출력 시퀀스는 노드에 대한 업데이트된 출력을 생성하기 위해 선행 메모리 상태가 트리에서의 노드들 중 하나의 출력과 결합되는 방법(즉 결합 방법 및 결합에 대한 활성화 함수)을 특정하는 두 개의 셀 주입 값들과, (i) 출력이 메모리 상태를 사용하여 업데이트되는 노드 및 (ii) 출력이 업데이트된 메모리 상태로 설정되는 노드(노드에 대한 활성화 함수의 적용 전에)를 특정하는 두 개의 셀 인덱스 값을 포함한다.

[0060]

도 2c의 예에서, 두 번째 셀 인덱스에 대해 생성된 값이 0이고, 주입(injection)에 대한 결합 방법이 "add"이고, 활성화 함수가 ReLU이므로, 셀은 선행 셀 상태와 트리 인덱스 0(a_0 라고 칭함)에서의 노드의 출력을 합산할 수 있고, 그 다음 트리 인덱스 0에서 노드의 업데이트된 출력을 생성하기 위해 그 합산에 ReLU를 적용할 수 있다. 그 다음, 상기 셀은 업데이트된 출력을 트리 인덱스 2의 노드에 입력으로서 제공할 수 있다.

[0061]

첫 번째 셀 인덱스에 대해 생성된 값이 1이기 때문에, 셀은 활성화 함수가 적용되기 전에 상기 업데이트된 메모리 상태를 인덱스 1의 트리 출력으로 설정한다.

[0062]

한편, 도 2c는 트리가 설명을 용이하게 하기 위해 2개의 리프 노드를 포함하는 예를 도시하고, 실제로 리프 노드의 수는 4, 8 또는 16과 같이 더 클 수 있다.

[0063]

도 3은 제어기 파라미터의 현재 값을 업데이트하기 위한 예시적인 프로세스(300)의 흐름도이다. 편의상, 프로세스(300)는 하나 이상의 위치에 위치한 하나 이상의 컴퓨터의 시스템에 의해 수행되는 것으로 설명될 것이다. 예를 들어, 적절하게 프로그램된 도 1의 신경 아키텍처 검색 시스템(100)과 같은 신경 아키텍처 검색 시스템은 프로세스(300)를 수행할 수 있다.

[0064]

상기 시스템은 제어기 신경망을 훈련시키기 위해, 즉 제어기 파라미터의 초기 값으로부터 제어기 파라미터의 훈련된 값을 결정하기 위해 프로세스(300)를 반복적으로 수행할 수 있다.

[0065]

상기 시스템은 제어기 신경망을 사용하고 반복시 제어기 파라미터의 현재 값에 따라 출력 시퀀스의 배치를 생성한다(단계 302). 배치의 각 출력 시퀀스는 차일드 신경망의 개별 아키텍처를 정의한다. 특히, 전술한 바와 같이, 상기 시스템은 출력 시퀀스에서 각각의 하이퍼파라미터 값을 생성할 때 스코어 분포로부터 샘플링하기 때문에, 배치의 시퀀스는 일반적으로 동일한 제어기 파라미터 값에 따라 생성되더라도 서로 다를 수 있다. 배치는 일반적으로 미리 결정된 수의 출력 시퀀스, 예를 들어 8, 16, 32 또는 64개의 시퀀스를 포함한다.

[0066]

배치 내의 각 출력 시퀀스에 대해, 시스템은 특정 신경망 태스크를 수행하기 위해 출력 시퀀스에 의해 정의된 아키텍처를 갖는 차일드 신경망의 인스턴스를 훈련시킨다(단계 304). 즉, 배치의 각 출력 시퀀스에 대해, 상기 시스템은 출력 시퀀스에 의해 정의된 아키텍처를 갖는 신경망을 인스턴스화하고, 태스크에 적합한 종래의 기계 학습 훈련 기술(예를 들어, 역 전파(또는 backpropagation-through-time)를 통한 확률적인 그래디언트 디센트 (gradient descent))을 사용하여 특정 신경망 태스크를 수행하기 위해 상기 수신된 훈련 데이터에 대한 인스턴스를 훈련시킨다. 일부 구현 예에서, 상기 시스템은 제어기 신경망에 대한 전체 훈련 시간을 줄이기 위해 차일드 신경망의 훈련을 병렬화한다. 상기 시스템은 지정된 시간 동안 또는 지정된 수의 훈련 반복 동안 각 차일드

신경망을 훈련시킬 수 있다.

- [0067] 배치의 각 출력 시퀀스에 대해, 상기 시스템은 특정 신경망 태스크에 대한 훈련된 인스턴스에 대한 성능 메트릭을 결정하기 위해 특정 신경망 태스크에 대한 차일드 신경망의 훈련된 인스턴스의 성능(performance)을 평가한다(단계 306). 예를 들어, 성능 메트릭은 적절한 정확도 측정에 의해 측정된 유효성 세트에서의 훈련된 인스턴스의 정확도일 수 있다. 예를 들어, 상기 정확도는 출력이 시퀀스일 때 난난도(perplexity measure)이거나 태스크가 분류 태스크일 때 분류 오류 비율일 수 있다. 또 다른 예로서, 성능 메트릭은 인스턴스의 훈련의 마지막 2, 5 또는 10 에포크(epoch)마다 인스턴스의 정확도의 평균 또는 최대값이 될 수 있다.
- [0068] 상기 시스템은 훈련된 인스턴스에 대한 성능 메트릭을 사용하여 제어기 파라미터의 현재 값을 조정한다(단계 308).
- [0069] 특히, 상기 시스템은 보강 학습 기술을 사용하여 증가된 성능 메트릭을 갖는 차일드 신경망을 초래하는 출력 시퀀스를 생성하도록 제어기 신경망을 훈련시킴으로써 현재 값을 조정한다. 보다 구체적으로, 상기 시스템은 훈련된 인스턴스의 성능 메트릭에 기초하여 결정되는 수신된 보상(reward)을 최대화하는 출력 시퀀스를 생성하기 위해 제어기 신경망을 훈련시킨다. 특히, 소정의 출력 시퀀스에 대한 보상은 훈련된 인스턴스의 성능 메트릭의 함수이다. 예를 들어, 상기 보상은 성능 메트릭, 성능 메트릭의 제곱, 성능 메트릭의 세제곱, 성능 메트릭의 제곱근 중 하나 일 수 있다.
- [0070] 일부 경우에는, 상기 시스템은 PG(policy gradient) 기술을 사용하여 예상되는 보상을 최대화하기 위해 제어기 신경망을 훈련시킨다. 예를 들어, PG 기술은 "REINFORCE 기술" 또는 PPO(Proximal Policy Optimization) 기술 일 수 있다. 예를 들어, 상기 시스템은 수학적 식 3을 만족하는 그라디언트의 추정기를 사용하여 제어기 파라미터에 대해 예상되는 보상의 그라디언트를 추정할 수 있다.

수학적 식 3

$$\frac{1}{m} \sum_{k=1}^m \sum_{t=1}^T \nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c) (R_k - b)$$

- [0071]
- [0072] 여기서, m은 배치의 시퀀스 수이며, T는 배치에서 각 시퀀스의 시간 단계의 수이고, a_t 는 소정의 출력 시퀀스의 시간 단계(t)에서의 출력이며, R_k 는 출력 시퀀스 k에 대한 보상이며, θ_c 는 제어기 파라미터이며, b는 베이스라인 함수, 예를 들어 이전 아키텍처 정확도의 지수(exponential) 이동 평균이다.
- [0073] 일부 구현 예에서, 시스템은 분산 방식으로 제어기 신경망을 훈련시킨다. 즉, 상기 시스템은 제어기 신경망의 여러 레플리카(replica)를 유지하고, 훈련 도중 비동기적으로 레플리카의 파라미터 값을 업데이트한다. 즉, 상기 시스템은 각 레플리카에 대해 단계(302-306)를 비동기적으로 수행할 수 있고, 각 레플리카에 대해 결정된 그라디언트를 사용하여 제어기 파라미터를 업데이트할 수 있다.
- [0074] 본 명세서는 시스템 및 컴퓨터 프로그램 구성 요소와 관련하여 "구성된"이라는 용어를 사용한다. 특정 동작이나 액션을 수행하도록 구성된 하나 이상의 컴퓨터 시스템은 시스템이 소프트웨어, 펌웨어, 하드웨어 또는 이들의 조합으로 인해 시스템이 동작 또는 액션을 수행하게 하는 것을 의미한다. 특정 동작 또는 액션을 수행하도록 구성된 하나 이상의 컴퓨터 프로그램은 하나 이상의 컴퓨터 프로그램이 데이터 처리 장치에 의해 실행될 때 장치로 하여금 동작 또는 액션을 수행하게 하는 명령어들을 포함함을 의미한다.
- [0075] 본 명세서에서 설명된 요지 및 기능적 동작의 실시 예는 본 명세서 및 그의 구조적 균등물에 개시된 구조들 또는 그들 중 하나 이상의 조합을 포함하여, 디지털 전자 회로, 유형적으로- 구현된 컴퓨터 소프트웨어 또는 펌웨어, 컴퓨터 하드웨어에서 구현될 수 있다. 본 명세서에서 설명된 요지의 실시 예는 하나 이상의 컴퓨터 프로그램, 즉 데이터 처리 장치에 의해 실행되거나 또는 데이터 처리 장치의 동작을 제어하기 위해 유형의 일시적 저장 매체상에 인코딩된 컴퓨터 프로그램 명령어들의 하나 이상의 모듈로서 구현될 수 있다. 상기 컴퓨터 저장 매체는 기계 판독가능 저장 장치, 기계 판독가능 저장 기판, 랜덤 또는 직렬 액세스 메모리 장치, 또는 이들 중 하나 이상의 조합일 수 있다. 대안으로 또는 부가적으로, 상기 프로그램 명령어들은 데이터 처리 장치에 의한 실행을 위해 적절한 수신기 장치로의 송신을 위해 정보를 인코딩하기 위해 생성되는 인위적으로 생성된 전파된

신호, 예를 들어, 기계-발생 전기, 광학 또는 전자기 신호상에 인코딩될 수 있다.

- [0076] "데이터 처리 장치"라는 용어는 데이터 처리 하드웨어를 의미하며, 예를 들어 프로그램 가능 프로세서, 컴퓨터, 또는 복수의 프로세서 또는 컴퓨터를 포함하여 데이터를 처리하기 위한 모든 종류의 장치, 디바이스 및 기계를 포함한다. 이 장치는 또한 특수 목적 논리 회로, 예를 들어 FPGA (field programmable gate array) 또는 ASIC(application specific integrated circuit)일 수 있다. 상기 장치는 하드웨어 이외에, 컴퓨터 프로그램들의 실행 환경을 생성하는 코드, 예를 들어 프로세서 펌웨어, 프로토콜 스택, 데이터베이스 관리 시스템, 운영 체제 또는 이들 중 하나 이상의 조합을 구성하는 코드를 선택적으로 포함할 수 있다.
- [0077] 컴퓨터 프로그램(프로그램, 소프트웨어, 소프트웨어 애플리케이션, 모듈, 소프트웨어 모듈, 스크립트 또는 코드로 지칭되거나 설명될 수 있음)은 컴파일된 또는 해석된 언어, 또는 선언적 또는 절차적 언어를 포함한 모든 형태의 프로그래밍 언어로 작성될 수 있으며, 독립 실행형 프로그램이나 모듈, 컴포넌트, 서브루틴 또는 컴퓨팅 환경에서 사용하기에 적합한 다른 장치를 포함하여 어떤 형태로든 배포될 수 있다. 컴퓨터 프로그램은 파일 시스템의 파일에 해당할 수 있지만 반드시 그런 것은 아니다. 프로그램은 프로그램 전용 단일 파일, 여러 개의 조정된 파일(예를 들어, 하나 이상의 모듈, 하위 프로그램 또는 코드의 부분들을 저장하는 파일들), 또는 마크업 언어 문서에 저장된 하나 이상의 스크립트와 같은 다른 프로그램들 또는 데이터를 보유하고 있는 파일의 부분에 저장될 수 있다. 컴퓨터 프로그램은 한 사이트에 있거나 여러 사이트에 분산되어 있으며 통신 네트워크로 상호 연결된 여러 대의 컴퓨터 또는 하나의 컴퓨터에서 실행되도록 배포될 수 있다.
- [0078] 본 명세서에서, "데이터베이스"라는 용어는 모든 데이터 수집을 나타내기 위해 광범위하게 사용되며, 그 데이터는 특정 방식으로 구조화되거나 전혀 구조화될 필요가 없으며 하나 이상의 위치에 있는 저장 장치에 저장할 수 있다. 따라서, 예를 들어, 인텍스 데이터베이스는 복수의 데이터 집합을 포함할 수 있으며, 각각의 집합은 다르게 구성 및 액세스될 수 있다.
- [0079] 유사하게, 본 명세서에서, 용어 "엔진"은 하나 이상의 특정 기능을 수행하도록 프로그램된 소프트웨어 기반 시스템, 서브 시스템 또는 프로세스를 지칭하기 위해 광범위하게 사용된다. 일반적으로 엔진은 하나 이상의 소프트웨어 모듈 또는 구성 요소로 구현되며 하나 이상의 위치에 있는 하나 이상의 컴퓨터에 설치된다. 일부 경우에 따라 하나 이상의 컴퓨터가 특정 엔진 전용으로 사용되며, 다른 경우에는 여러 대의 엔진을 동일한 컴퓨터나 컴퓨터들에 설치하여 실행할 수 있다.
- [0080] 본 명세서에서 설명되는 프로세스들 및 로직 흐름은 입력 데이터를 조작하고 출력을 생성함으로써 기능을 수행하도록 하나 이상의 컴퓨터 프로그램을 실행하는 하나 이상의 프로그램 가능 컴퓨터에 의해 수행될 수 있다. 상기 프로세스들 및 로직 흐름은 또한 FPGA(field programmable gate array) 또는 ASIC(application specific integrated circuit)과 같은 특수 목적 논리 회로에 의해 수행될 수 있고, 장치는 또한 상기 특수 목적 논리 회로로 구현될 수 있다.
- [0081] 컴퓨터 프로그램의 실행에 적합한 컴퓨터는 예를 들어 범용 또는 특수 목적 마이크로프로세서 또는 둘 모두, 또는 임의의 다른 종류의 중앙 처리 장치를 포함하고, 이들에 기반할 수 있다. 일반적으로, 중앙 처리 장치는 판독 전용 메모리 또는 랜덤 액세스 메모리 또는 둘 모두로부터 명령어들 및 데이터를 수신할 것이다. 컴퓨터의 필수 구성요소들은 명령어들을 수행하거나 실행하기 위한 중앙 처리 장치 및 명령어들 및 데이터를 저장하기 위한 하나 이상의 메모리 장치이다. 일반적으로, 컴퓨터는 데이터(예를 들어, 자기, 광 자기 디스크 또는 광 디스크)를 저장하기 위한 하나 이상의 대용량 저장 장치를 포함하거나, 그 하나 이상의 대용량 저장 장치로부터 데이터를 수신하거나 전송하기 위해 동작 가능하게 결합될 것이다. 그러나, 컴퓨터는 그러한 장치들을 가질 필요는 없다. 또한, 컴퓨터는 다른 장치, 예를 들어, 이동 전화기, 개인 휴대 정보 단말기(PDA), 이동 오디오 또는 비디오 플레이어, 게임 콘솔, GPS 수신기 또는 휴대용 저장 장치(예를 들어, 범용 직렬 버스(USB) 플래시 드라이브)에 내장될 수 있다.
- [0082] 컴퓨터 프로그램 명령어들 및 데이터를 저장하기에 적합한 컴퓨터 판독 가능 매체는 예를 들어 EPROM, EEPROM 및 플래시 메모리 장치와 같은 반도체 메모리 장치, 내부 하드 디스크 또는 이동식 디스크와 같은 자기 디스크, 광 자기 디스크, 및 CD ROM 및 DVD-ROM 디스크를 포함하는 모든 형태의 비휘발성 메모리, 매체 및 메모리 장치를 포함한다.
- [0083] 사용자와의 상호 작용을 제공하기 위해, 본 명세서에서 설명된 요지의 실시예들은 사용자에게 정보를 제공하기 위한 CRT(cathode ray tube) 또는 LCD(liquid crystal display) 모니터와 같은 디스플레이 장치, 사용자가 입력을 컴퓨터에 제공할 수 있는 마우스 또는 트랙볼과 같은 키보드 및 포인팅 장치를 갖는 컴퓨터에서 구현될 수

있다. 다른 종류의 장치들은 사용자와의 상호 작용을 제공하는 데 사용될 수 있으며, 예를 들어, 사용자에게 제공되는 피드백은 시각 피드백, 청각 피드백 또는 촉각 피드백과 같은 임의의 형태의 감각 피드백일 수 있고, 사용자로부터의 입력은 음향, 음성 또는 촉각 입력을 포함하는 임의의 형태로 수신될 수 있다. 또한, 컴퓨터는 사용자가 사용하는 장치로 문서를 보내고 문서를 수신하여 사용자와 상호 작용할 수 있으며, 예를 들어, 웹 브라우저로부터 수신된 요청에 응답하여 사용자의 클라이언트 장치상의 웹 브라우저에 웹 페이지를 전송함으로써 수행될 수 있다. 또한, 컴퓨터는 문자 메시지 또는 다른 형태의 메시지를 개인용 장치(예를 들어, 메시징 애플리케이션을 실행 중인 스마트폰)에 송신하고 사용자로부터 응답 메시지를 수신함으로써 사용자와 상호 작용할 수 있다.

[0084] 기계 학습 모델을 구현하기 위한 데이터 처리 장치는 또한, 예를 들어, 기계 학습 훈련 또는 생산, 즉 추론, 작업부하의 공통 및 연산 중심 부분을 처리하기 위한 특수 목적 하드웨어 가속기 유닛을 포함할 수 있다.

[0085] 기계 학습 모델은 "TensorFlow" 프레임워크, "Microsoft Cognitive Toolkit" 프레임워크, "Apache Singa" 프레임워크 또는 "Apache MXNet" 프레임워크와 같은 기계 학습 프레임워크를 사용하여 구현 및 배치할 수 있다.

[0086] 본 명세서에서 설명된 요지의 실시예들은 데이터 서버와 같은 백 엔드 컴포넌트; 애플리케이션 서버와 같은 미들웨어 컴포넌트; 예를 들어 관계 그래픽 사용자 인터페이스 또는 사용자가 본 명세서에 설명된 요지의 구현예와 상호 작용할 수 있는 웹 브라우저를 갖는 클라이언트 컴퓨터와 같은 프론트 엔드 컴포넌트; 또는 하나 이상의 백 엔드, 미들웨어, 프론트 엔드 컴포넌트들의 임의의 조합을 포함하는 컴퓨팅 시스템에서 구현될 수 있다. 상기 시스템의 컴포넌트들은 디지털 데이터 통신의 임의의 형태 또는 매체, 예를 들어 통신 네트워크에 의해 상호 접속될 수 있다. 예시적인 통신 네트워크는 근거리 통신망("LAN") 및 광역 통신망("WAN"), 예를 들어 인터넷을 포함한다.

[0087] 상기 컴퓨팅 시스템은 클라이언트들과 서버들을 포함할 수 있다. 클라이언트와 서버는 일반적으로 서로 멀리 떨어져 있으며, 일반적으로 통신 네트워크를 통해 상호 작용한다. 클라이언트와 서버의 관계는 각각의 컴퓨터에서 실행되고 서로 클라이언트-서버 관계를 갖는 컴퓨터 프로그램들로 인해 발생한다. 일부 실시 예에서, 서버는 데이터, 예를 들어, 데이터를 디스플레이하고, 클라이언트로서 동작하는 장치와 상호 작용하는 사용자로부터 사용자 입력을 수신하기 위해 HTML 페이지를 사용자 장치로 송신한다. 사용자 장치에서 생성된 데이터, 예를 들어 사용자 상호 작용의 결과는 상기 장치로부터 서버에서 수신될 수 있다.

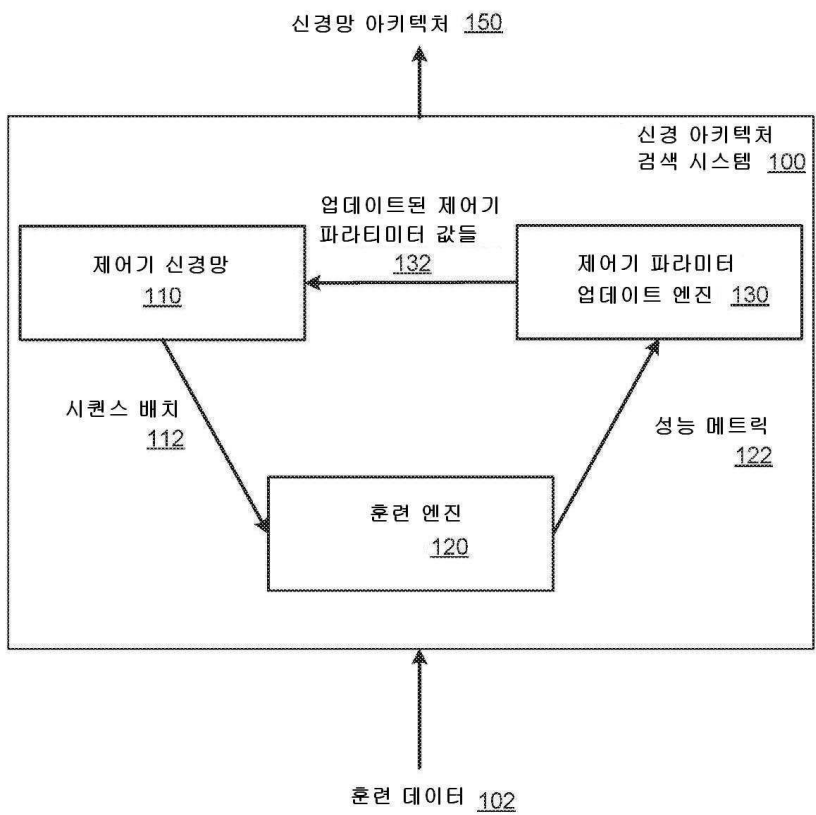
[0088] 본 명세서는 다수의 특정 구현 세부 사항을 포함하지만, 이들은 임의의 발명 또는 청구될 수 있는 범위에 대한 제한으로서 해석되어서는 안되며, 오히려 특정 발명의 특정 실시예에 특정될 수 있는 특징에 대한 설명으로 해석되어야 한다. 별도의 실시예들과 관련하여 본 명세서에서 설명되는 특정 특징들은 또한 단일 실시예에서 조합하여 구현될 수 있다. 반대로, 단일 실시예의 컨텍스트에서 설명된 다양한 특징들은 또한 다수의 실시예에서 개별적으로 또는 임의의 적합한 서브조합으로 구현될 수 있다. 더욱이, 특징들은 소정의 조합으로 작용하고 상술한 바와 같이 초기에 청구된 것으로서 설명될 수 있지만, 청구된 조합의 하나 이상의 특징이 어떤 경우 그 조합으로부터 제거될 수 있고, 그 청구된 조합은 서브조합 또는 그 서브조합의 변형을 지향할 수 있다.

[0089] 유사하게, 동작들이 특정 순서로 도면들에 도시되어 있지만, 이는 바람직한 동작들을 달성하기 위해, 그러한 동작들이 도시된 순서 또는 순차적인 순서로 수행되거나, 도시된 모든 동작들이 수행될 것을 요구하는 것으로 이해되어서는 안된다. 특정 상황에서 멀티 태스킹 및 병렬 처리가 유리할 수 있다. 또한, 상술한 실시 예에서 다양한 시스템 모듈 및 컴포넌트의 분리는 모든 실시예에서 그러한 분리를 필요로 하는 것으로 이해되어서는 안되며, 서술된 프로그램 컴포넌트들 및 시스템들은 일반적으로 단일 소프트웨어 제품에 함께 통합되거나 다중 소프트웨어 제품들로 패키징될 수 있다는 것을 이해해야 한다.

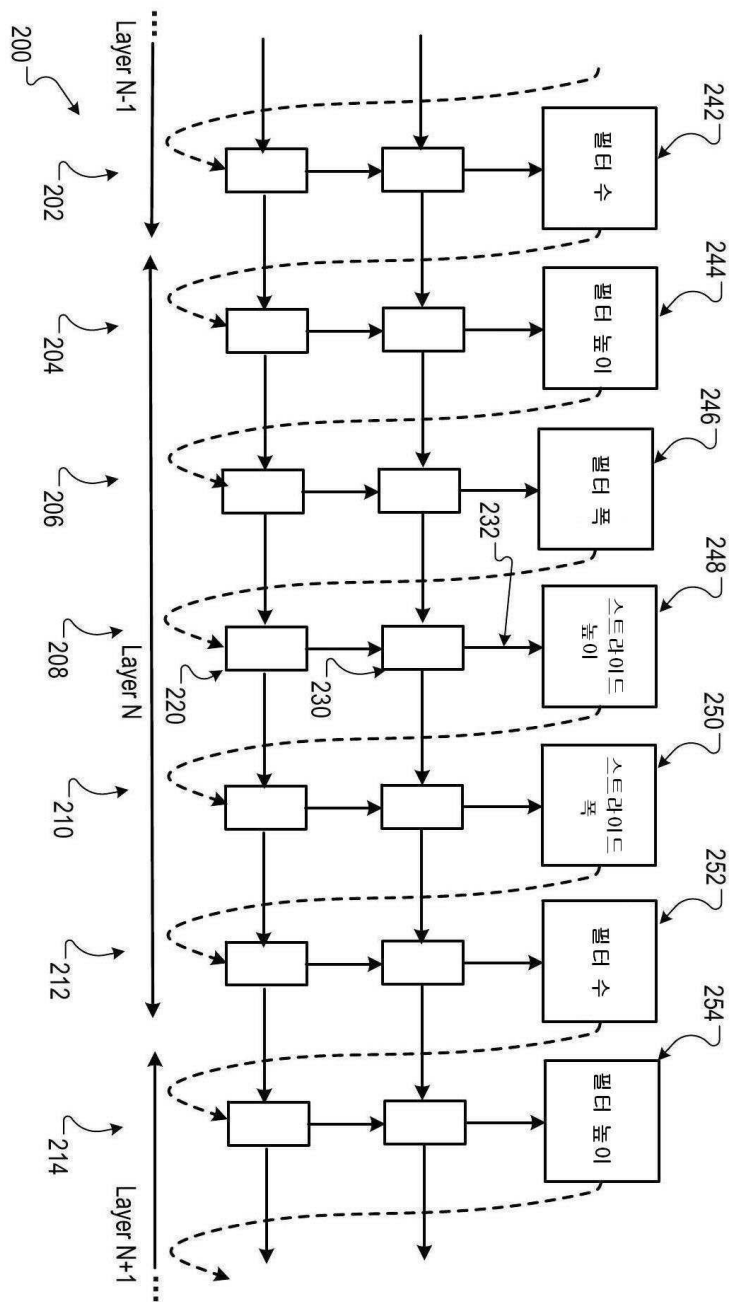
[0090] 본 발명의 특정 실시예들이 설명되었다. 다른 실시예들은 다음의 청구항들의 범위 내에 있다. 예를 들어, 청구 범위에 열거된 동작들은 상이한 순서로 수행될 수 있으며 여전히 바람직한 결과를 달성한다. 하나의 예로서, 첨부된 도면에 도시된 프로세스는 바람직한 결과를 얻기 위해 도시된 특정 순서 또는 순차적 순서를 반드시 필요로 하지는 않는다. 특정 구현예들에서, 멀티태스킹 및 병렬 처리가 유리할 수 있다.

도면

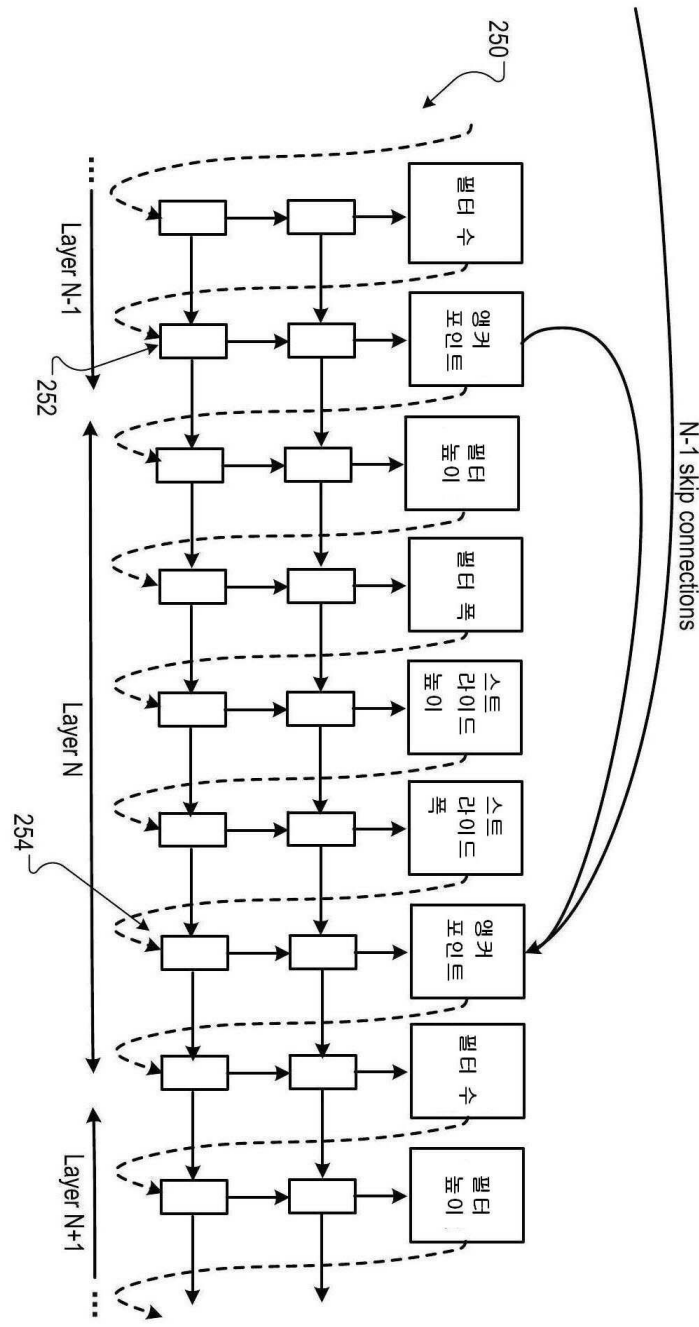
도면1



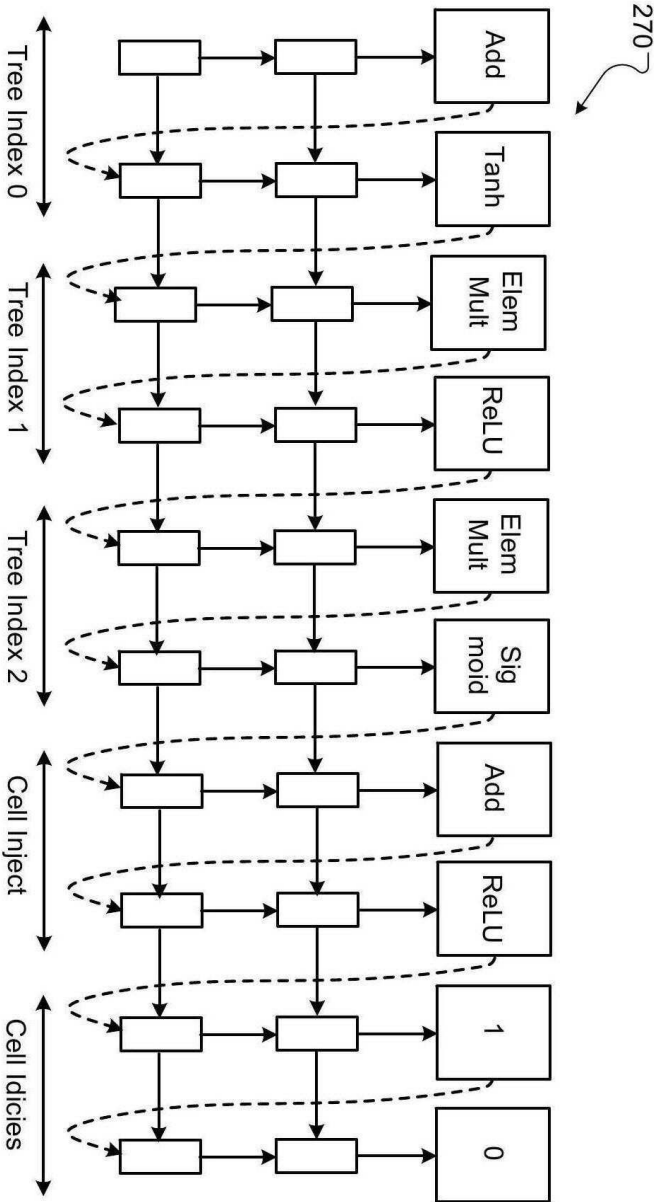
도면2a



도면2b



도면2c



도면3

300

