



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2025년05월12일

(11) 등록번호 10-2806514

(24) 등록일자 2025년05월08일

(51) 국제특허분류(Int. Cl.)  
G06N 3/04 (2023.01) G06N 3/08 (2023.01)(52) CPC특허분류  
G06N 3/045 (2023.01)  
G06N 3/08 (2023.01)

(21) 출원번호 10-2018-7001459

(22) 출원일자(국제) 2016년06월27일

심사청구일자 2021년06월09일

(85) 번역문제출일자 2018년01월16일

(65) 공개번호 10-2018-0034395

(43) 공개일자 2018년04월04일

(86) 국제출원번호 PCT/US2016/039661

(87) 국제공개번호 WO 2017/052709

국제공개일자 2017년03월30일

(30) 우선권주장

62/195,763 2015년07월22일 미국(US)

14/851,911 2015년09월11일 미국(US)

(56) 선행기술조사문헌

JP2012084117 A\*

US05793932 A\*

\*는 심사관에 의하여 인용된 문헌

(73) 특허권자

퀵컴 인코포레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

비어진스키 카시미어 매튜

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(74) 대리인

특허법인코리아나

전체 청구항 수 : 총 26 항

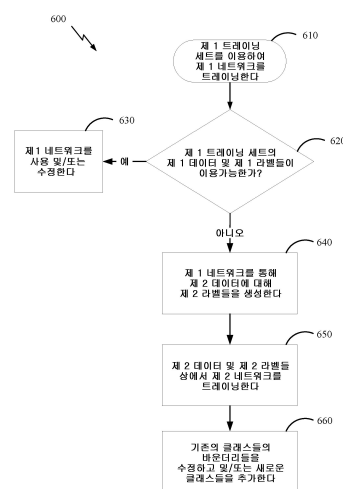
심사관 : 박요한

(54) 발명의 명칭 뉴럴 네트워크들에서의 러닝을 트랜스퍼하기 위한 장치들 및 방법들

## (57) 요약

러닝 트랜스퍼의 방법은 제 2 데이터를 수신하는 단계 및 제 1 네트워크를 통해 제 2 데이터에 대한 제 2 라벨을 생성하는 단계를 포함한다. 일 구성에서, 제 1 네트워크는 제 1 데이터에 대한 제 1 라벨들 상에서 미리 트레이닝되어 왔다. 부가적으로, 제 2 라벨들은 제 2 네트워크를 트레이닝하기 위해 생성된다.

대표도 - 도6



## 명세서

### 청구범위

#### 청구항 1

러닝 트랜스퍼 방법으로서,

제 1 트레이닝된 (trained) 뉴럴 네트워크에서, 제 2 라벨링되지 않은 (un-labeled) 데이터를 수신하는 단계;

상기 제 1 트레이닝된 뉴럴 네트워크를 통해, 상기 제 2 라벨링되지 않은 데이터를 라벨링 함으로써 제 2 라벨링된 데이터를 생성하는 단계로서, 상기 제 1 트레이닝된 뉴럴 네트워크는 상기 제 2 라벨링되지 않은 데이터를 수신하기 전에 제 1 라벨링된 데이터 상에서 트레이닝되었고, 상기 제 1 라벨링된 데이터는 상기 제 2 라벨링된 데이터와 상이한, 상기 제 2 라벨링된 데이터를 생성하는 단계;

상기 제 1 트레이닝된 뉴럴 네트워크가 상기 제 2 라벨링되지 않은 데이터를 라벨링한 이후 상기 제 2 라벨링된 데이터로 제 2 뉴럴 네트워크를 트레이닝 하는 단계로서, 상기 제 2 뉴럴 네트워크는 상기 제 1 트레이닝된 뉴럴 네트워크와 상이한, 상기 제 2 뉴럴 네트워크를 트레이닝 하는 단계;

상기 제 1 트레이닝된 뉴럴 네트워크 및 트레이닝된 상기 제 2 뉴럴 네트워크를 트레이닝한 이후, 트레이닝된 상기 제 2 뉴럴 네트워크를 업데이트된 트레이닝 데이터로 재-트레이닝 (re-training) 하는 단계로서, 상기 업데이트된 트레이닝 데이터는 상기 제 2 라벨링된 데이터 및 제 3 라벨링된 데이터의 조합을 포함하고, 상기 제 2 라벨링되지 않은 데이터는 상기 제 3 라벨링된 데이터를 생성하는데 사용된 제 3 라벨링되지 않은 데이터와 상이한, 트레이닝된 상기 제 2 뉴럴 네트워크를 재-트레이닝하는 단계; 및

트레이닝된 상기 제 2 뉴럴 네트워크를 통하여, 트레이닝된 상기 제 2 뉴럴 네트워크에서 수신된 입력을 분류하는 단계를 포함하는, 러닝 트랜스퍼 방법.

#### 청구항 2

삭제

#### 청구항 3

제 1 항에 있어서,

트레이닝된 상기 제 2 뉴럴 네트워크의 크기는 상기 제 1 트레이닝된 뉴럴 네트워크의 크기와 다른, 러닝 트랜스퍼 방법.

#### 청구항 4

삭제

#### 청구항 5

제 3 항에 있어서,

상기 제 3 라벨링된 데이터에 적어도 부분적으로 기초하여 트레이닝된 상기 제 2 뉴럴 네트워크에서 클래스의 적어도 하나의 기존 바운더리를 수정하는 단계를 더 포함하는, 러닝 트랜스퍼 방법.

#### 청구항 6

제 3 항에 있어서,

상기 제 3 라벨링된 데이터에 적어도 부분적으로 기초하여 트레이닝된 상기 제 2 뉴럴 네트워크에서 적어도 하나의 새로운 클래스를 추가하는 단계를 더 포함하는, 러닝 트랜스퍼 방법.

#### 청구항 7

제 1 항에 있어서,

상기 제 1 라벨링된 데이터는, 제 1 라벨들을 제 1 라벨링되지 않은 데이터와 연관시킴으로써 생성되고, 상기 제 1 라벨링되지 않은 데이터 또는 상기 제 1 라벨들 중 적어도 하나는 상기 제 1 트레이닝된 뉴럴 네트워크를 트레이닝한 이후에 이용 가능하지 않은, 러닝 트랜스퍼 방법.

#### 청구항 8

삭제

#### 청구항 9

제 1 항에 있어서,

상기 제 2 라벨링된 데이터를 클러스터링하는 단계;

클러스터링된 상기 제 2 라벨링된 데이터를 상기 제 3 라벨링된 데이터와 비교하는 단계; 및

상기 업데이트된 트레이닝 데이터를 위하여 상기 제 3 라벨링된 데이터와의 유사성 또는 상기 제 3 라벨링된 데이터와의 차이에 적어도 부분적으로 기초하여 상기 제 2 라벨링된 데이터를 선택하는 단계를 더 포함하는, 러닝 트랜스퍼 방법.

#### 청구항 10

러닝 트랜스퍼 방법으로서,

제 2 뉴럴 네트워크에서, 제 2 라벨링되지 않은 데이터의 분류에 기초하여 제 1 트레이닝된 뉴럴 네트워크에 의해 라벨링된 제 2 라벨링된 데이터를 수신하는 단계로서, 상기 제 1 트레이닝된 뉴럴 네트워크는 상기 제 2 라벨링되지 않은 데이터를 수신하기 전에 제 1 라벨링된 데이터 상에서 트레이닝되었고, 상기 제 1 라벨링된 데이터는 상기 제 2 라벨링된 데이터와 상이한, 상기 제 2 라벨링된 데이터를 수신하는 단계;

상기 제 1 트레이닝된 뉴럴 네트워크가 상기 제 2 라벨링되지 않은 데이터를 라벨링한 이후 상기 제 2 라벨링된 데이터 상에서 상기 제 2 뉴럴 네트워크를 트레이닝하는 단계로서, 상기 제 2 뉴럴 네트워크는 상기 제 1 트레이닝된 뉴럴 네트워크와 상이한, 상기 제 2 뉴럴 네트워크를 트레이닝 하는 단계;

상기 제 1 트레이닝된 뉴럴 네트워크 및 트레이닝된 상기 제 2 뉴럴 네트워크를 트레이닝한 이후, 트레이닝된 상기 제 2 뉴럴 네트워크를 업데이트된 트레이닝 데이터로 재-트레이닝 (re-training) 하는 단계로서, 상기 업데이트된 트레이닝 데이터는 상기 제 2 라벨링된 데이터 및 제 3 라벨링된 데이터의 조합을 포함하고, 상기 제 2 라벨링되지 않은 데이터는 상기 제 3 라벨링된 데이터를 생성하는데 사용된 제 3 라벨링되지 않은 데이터와 상이한, 트레이닝된 상기 제 2 뉴럴 네트워크를 재-트레이닝하는 단계; 및

트레이닝된 상기 제 2 뉴럴 네트워크를 통하여, 트레이닝된 상기 제 2 뉴럴 네트워크에서 수신된 입력을 분류하는 단계를 포함하는, 러닝 트랜스퍼 방법.

#### 청구항 11

삭제

#### 청구항 12

제 10 항에 있어서,

상기 제 2 라벨링된 데이터를 클러스터링하는 단계;

클러스터링된 상기 제 2 라벨링된 데이터를 상기 제 3 라벨링된 데이터와 비교하는 단계; 및

상기 업데이트된 트레이닝 데이터를 위하여 상기 제 3 라벨링된 데이터와의 유사성 또는 상기 제 3 라벨링된 데이터와의 차이에 적어도 부분적으로 기초하여 상기 제 2 라벨링된 데이터를 선택하는 단계를 더 포함하는, 러닝 트랜스퍼 방법.

#### 청구항 13

제 10 항에 있어서,

트레이닝된 상기 제 2 뉴럴 네트워크의 크기는 상기 제 1 트레이닝된 뉴럴 네트워크의 크기와 다른, 러닝 트랜스퍼 방법.

#### 청구항 14

삭제

#### 청구항 15

제 13 항에 있어서,

상기 제 3 라벨링된 데이터에 적어도 부분적으로 기초하여 트레이닝된 상기 제 2 뉴럴 네트워크에서 클래스의 적어도 하나의 기존 바운더리를 수정하는 단계를 더 포함하는, 러닝 트랜스퍼 방법.

#### 청구항 16

제 13 항에 있어서,

상기 제 3 라벨링된 데이터에 적어도 부분적으로 기초하여 트레이닝된 상기 제 2 뉴럴 네트워크에서 적어도 하나의 클래스를 추가하는 단계를 더 포함하는, 러닝 트랜스퍼 방법.

#### 청구항 17

제 10 항에 있어서,

상기 제 1 라벨링된 데이터는, 제 1 라벨들을 제 1 라벨링되지 않은 데이터와 연관시킴으로써 생성되고, 상기 제 1 라벨링되지 않은 데이터 또는 상기 제 1 라벨들 중 적어도 하나는 상기 제 1 트레이닝된 뉴럴 네트워크를 트레이닝한 이후에 이용 가능하지 않은, 러닝 트랜스퍼 방법.

#### 청구항 18

삭제

#### 청구항 19

러닝 트랜스퍼 장치로서,

메모리 유닛; 및

상기 메모리 유닛에 커플링된 적어도 하나의 프로세서를 포함하고,

상기 적어도 하나의 프로세서는:

제 1 트레이닝된 (trained) 뉴럴 네트워크에서, 제 2 라벨링되지 않은 (un-labeled) 데이터를 수신하고;

상기 제 1 트레이닝된 뉴럴 네트워크를 통해, 상기 제 2 라벨링되지 않은 데이터를 라벨링 함으로써 제 2 라벨링된 데이터를 생성하는 것으로서, 상기 제 1 트레이닝된 뉴럴 네트워크는 상기 제 2 라벨링되지 않은 데이터를 수신하기 전에 제 1 라벨링된 데이터 상에서 트레이닝되었고, 상기 제 1 라벨링된 데이터는 상기 제 2 라벨링된 데이터와 상이한, 상기 제 2 라벨링된 데이터를 생성하고;

상기 제 1 트레이닝된 뉴럴 네트워크가 상기 제 2 라벨링되지 않은 데이터를 라벨링한 이후 상기 제 2 라벨링된 데이터로 제 2 뉴럴 네트워크를 트레이닝 하는 것으로서, 상기 제 2 뉴럴 네트워크는 상기 제 1 트레이닝된 뉴럴 네트워크와 상이한, 상기 제 2 뉴럴 네트워크를 트레이닝 하고;

상기 제 1 트레이닝된 뉴럴 네트워크를 트레이닝한 이후, 트레이닝된 상기 제 2 뉴럴 네트워크를 업데이트된 트레이닝 데이터로 재-트레이닝 (re-training) 하는 것으로서, 상기 업데이트된 트레이닝 데이터는 상기 제 2 라벨링된 데이터 및 제 3 라벨링된 데이터의 조합을 포함하고, 상기 제 2 라벨링되지 않은 데이터는 상기 제 3 라벨링된 데이터를 생성하는데 사용된 제 3 라벨링되지 않은 데이터와 상이한, 트레이닝된 상기 제 2 뉴럴 네트워크를 재-트레이닝하고; 그리고

트레이닝된 상기 제 2 뉴럴 네트워크를 통하여, 트레이닝된 상기 제 2 뉴럴 네트워크에서 수신된 입력을 분류하도록 구성되는, 러닝 트랜스퍼 장치.

#### 청구항 20

삭제

#### 청구항 21

제 19 항에 있어서,

트레이닝된 상기 제 2 뉴럴 네트워크의 크기는 상기 제 1 트레이닝된 뉴럴 네트워크의 크기와 다른, 러닝 트랜스퍼 장치.

#### 청구항 22

삭제

#### 청구항 23

제 21 항에 있어서,

상기 적어도 하나의 프로세서는 또한, 상기 제 3 라벨링된 데이터에 적어도 부분적으로 기초하여 트레이닝된 상기 제 2 뉴럴 네트워크에서 클래스의 적어도 하나의 기존 바운더리를 수정하도록 구성되는, 러닝 트랜스퍼 장치.

#### 청구항 24

제 21 항에 있어서,

상기 적어도 하나의 프로세서는 또한, 상기 제 3 라벨링된 데이터에 적어도 부분적으로 기초하여 트레이닝된 상기 제 2 뉴럴 네트워크에서 적어도 하나의 새로운 클래스를 추가하도록 구성되는, 러닝 트랜스퍼 장치.

#### 청구항 25

제 19 항에 있어서,

상기 제 1 라벨링된 데이터는, 제 1 라벨들을 제 1 라벨링되지 않은 데이터와 연관시킴으로써 생성되고, 상기 제 1 라벨링되지 않은 데이터 또는 상기 제 1 라벨들 중 적어도 하나는 상기 제 1 트레이닝된 뉴럴 네트워크를 트레이닝한 이후에 이용 가능하지 않은, 러닝 트랜스퍼 장치.

#### 청구항 26

삭제

#### 청구항 27

제 19 항에 있어서,

상기 적어도 하나의 프로세서는 또한:

상기 제 2 라벨링된 데이터를 클러스터링하고;

클러스터링된 상기 제 2 라벨링된 데이터를 상기 제 3 라벨링된 데이터와 비교하고; 그리고

상기 업데이트된 트레이닝 데이터를 위하여 상기 제 3 라벨링된 데이터와의 유사성 또는 상기 제 3 라벨링된 데이터와의 차이에 적어도 부분적으로 기초하여 상기 제 2 라벨링된 데이터를 선택하도록 구성되는, 러닝 트랜스퍼 장치.

#### 청구항 28

러닝 트랜스퍼 장치로서,

메모리 유닛; 및

상기 메모리 유닛에 커플링된 적어도 하나의 프로세스를 포함하고,

상기 적어도 하나의 프로세서는:

제 2 뉴럴 네트워크에서, 제 2 라벨링되지 않은 데이터의 분류에 기초하여 제 1 트레이닝된 뉴럴 네트워크에 의해 라벨링된 제 2 라벨링된 데이터를 수신하는 것으로서, 상기 제 1 트레이닝된 뉴럴 네트워크는 상기 제 2 라벨링되지 않은 데이터를 수신하기 전에 제 1 라벨링된 데이터 상에서 트레이닝되었고, 상기 제 1 라벨링된 데이터는 상기 제 2 라벨링된 데이터와 상이한, 상기 제 2 라벨링된 데이터를 수신하고;

상기 제 1 트레이닝된 뉴럴 네트워크가 상기 제 2 라벨링되지 않은 데이터를 라벨링한 이후 상기 제 2 라벨링된 데이터 상에서 상기 제 2 뉴럴 네트워크를 트레이닝하는 것으로서, 상기 제 2 뉴럴 네트워크는 상기 제 1 트레이닝된 뉴럴 네트워크와 상이한, 상기 제 2 뉴럴 네트워크를 트레이닝하고;

상기 제 1 트레이닝된 뉴럴 네트워크를 트레이닝한 이후, 트레이닝된 상기 제 2 뉴럴 네트워크를 업데이트된 트레이닝 데이터로 재-트레이닝 (re-training) 하는 것으로서, 상기 업데이트된 트레이닝 데이터는 상기 제 2 라벨링된 데이터 및 제 3 라벨링된 데이터의 조합을 포함하고, 상기 제 2 라벨링되지 않은 데이터는 상기 제 3 라벨링된 데이터를 생성하는데 사용된 제 3 라벨링되지 않은 데이터와 상이한, 트레이닝된 상기 제 2 뉴럴 네트워크를 재-트레이닝하고; 그리고

트레이닝된 상기 제 2 뉴럴 네트워크를 통하여, 트레이닝된 상기 제 2 뉴럴 네트워크에서 수신된 입력을 분류하도록 구성되는, 러닝 트랜스퍼 장치.

#### 청구항 29

삭제

#### 청구항 30

제 28 항에 있어서,

상기 적어도 하나의 프로세서는 또한:

상기 제 2 라벨링된 데이터를 클러스터링하고;

클러스터링된 상기 제 2 라벨링된 데이터를 상기 제 3 라벨링된 데이터와 비교하고; 그리고

상기 업데이트된 트레이닝 데이터를 위하여 상기 제 3 라벨링된 데이터와의 유사성 또는 상기 제 3 라벨링된 데이터와의 차이에 적어도 부분적으로 기초하여 상기 제 2 라벨링된 데이터를 선택하도록 구성되는, 러닝 트랜스퍼 장치.

#### 청구항 31

제 28 항에 있어서,

트레이닝된 상기 제 2 뉴럴 네트워크의 크기는 상기 제 1 트레이닝된 뉴럴 네트워크의 크기와 다른, 러닝 트랜스퍼 장치.

#### 청구항 32

삭제

#### 청구항 33

제 31 항에 있어서,

상기 적어도 하나의 프로세서는 또한, 상기 제 3 라벨링된 데이터에 적어도 부분적으로 기초하여 트레이닝된 상기 제 2 뉴럴 네트워크에서 클래스의 적어도 하나의 기존 바운더리를 수정하도록 구성되는, 러닝 트랜스퍼 장치.

#### 청구항 34

제 31 항에 있어서,

상기 적어도 하나의 프로세서는 또한, 상기 제 3 라벨링된 데이터에 적어도 부분적으로 기초하여 트레이닝된 상

기 제 2 뉴럴 네트워크에서 적어도 하나의 새로운 클래스를 추가하도록 구성되는, 러닝 트랜스퍼 장치.

#### 청구항 35

제 28 항에 있어서,

상기 제 1 라벨링된 데이터는, 제 1 라벨들을 제 1 라벨링되지 않은 데이터와 연관시킴으로써 생성되고, 상기 제 1 라벨링되지 않은 데이터 또는 상기 제 1 라벨들 중 적어도 하나는 상기 제 1 트레이닝된 뉴럴 네트워크를 트레이닝한 이후에 이용 가능하지 않은, 러닝 트랜스퍼 장치.

#### 청구항 36

삭제

#### 청구항 37

삭제

#### 청구항 38

삭제

#### 청구항 39

프로그램 코드가 기록되어 있는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 프로그램 코드는:

제 1 트레이닝된 (trained) 뉴럴 네트워크에서, 제 2 라벨링되지 않은 (un-labeled) 데이터를 수신하기 위한 프로그램 코드;

상기 제 1 트레이닝된 뉴럴 네트워크를 통해, 상기 제 2 라벨링되지 않은 데이터를 라벨링 함으로써 제 2 라벨링된 데이터를 생성하기 위한 프로그램 코드로서, 상기 제 1 트레이닝된 뉴럴 네트워크는 상기 제 2 라벨링되지 않은 데이터를 수신하기 전에 제 1 라벨링된 데이터 상에서 트레이닝되었고, 상기 제 1 라벨링된 데이터는 상기 제 2 라벨링된 데이터와 상이한, 상기 제 2 라벨링된 데이터를 생성하기 위한 프로그램 코드;

상기 제 1 트레이닝된 뉴럴 네트워크가 상기 제 2 라벨링되지 않은 데이터를 라벨링한 이후 상기 제 2 라벨링된 데이터로 제 2 뉴럴 네트워크를 트레이닝 하기 위한 프로그램 코드로서, 상기 제 2 뉴럴 네트워크는 상기 제 1 트레이닝된 뉴럴 네트워크와 상이한, 상기 제 2 뉴럴 네트워크를 트레이닝 하기 위한 프로그램 코드;

상기 제 1 트레이닝된 뉴럴 네트워크를 트레이닝한 이후, 트레이닝된 상기 제 2 뉴럴 네트워크를 업데이트된 트레이닝 데이터로 재-트레이닝 (re-training) 하기 위한 프로그램 코드로서, 상기 업데이트된 트레이닝 데이터는 상기 제 2 라벨링된 데이터 및 제 3 라벨링된 데이터의 조합을 포함하고, 상기 제 2 라벨링되지 않은 데이터는 상기 제 3 라벨링된 데이터를 생성하는데 사용된 제 3 라벨링되지 않은 데이터와 상이한, 트레이닝된 상기 제 2 뉴럴 네트워크를 재-트레이닝하기 위한 프로그램 코드; 및

트레이닝된 상기 제 2 뉴럴 네트워크를 통하여, 트레이닝된 상기 제 2 뉴럴 네트워크에서 수신된 입력을 분류하기 위한 프로그램 코드를 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

#### 청구항 40

프로그램 코드가 기록되어 있는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 프로그램 코드는:

제 2 뉴럴 네트워크에서, 제 2 라벨링되지 않은 데이터의 분류에 기초하여 제 1 트레이닝된 뉴럴 네트워크에 의해 라벨링된 제 2 라벨링된 데이터를 수신하기 위한 프로그램 코드로서, 상기 제 1 트레이닝된 뉴럴 네트워크는 상기 제 2 라벨링되지 않은 데이터를 수신하기 전에 제 1 라벨링된 데이터 상에서 트레이닝되었고, 상기 제 1 라벨링된 데이터는 상기 제 2 라벨링된 데이터와 상이한, 상기 제 2 라벨링된 데이터를 수신하기 위한 프로그램 코드;

상기 제 1 트레이닝된 뉴럴 네트워크가 상기 제 2 라벨링되지 않은 데이터를 라벨링한 이후 상기 제 2 라벨링된

데이터 상에서 상기 제 2 뉴럴 네트워크를 트레이닝하기 위한 프로그램 코드로서, 상기 제 2 뉴럴 네트워크는 상기 제 1 트레이닝된 뉴럴 네트워크와 상이한, 상기 제 2 뉴럴 네트워크를 트레이닝하기 위한 프로그램 코드;

상기 제 1 트레이닝된 뉴럴 네트워크를 트레이닝한 이후, 트레이닝된 상기 제 2 뉴럴 네트워크를 업데이트된 트레이닝 데이터로 재-트레이닝 (re-training) 하기 위한 프로그램 코드로서, 상기 업데이트된 트레이닝 데이터는 상기 제 2 라벨링된 데이터 및 제 3 라벨링된 데이터의 조합을 포함하고, 상기 제 2 라벨링되지 않은 데이터는 상기 제 3 라벨링된 데이터를 생성하는데 사용된 제 3 라벨링되지 않은 데이터와 상이한, 트레이닝된 상기 제 2 뉴럴 네트워크를 재-트레이닝하기 위한 프로그램 코드; 및

트레이닝된 상기 제 2 뉴럴 네트워크를 통하여, 트레이닝된 상기 제 2 뉴럴 네트워크에서 수신된 입력을 분류하기 위한 프로그램 코드를 포함하는, 비일시적 컴퓨터 판독가능 저장 매체.

## 발명의 설명

## 기술 분야

[0001] 관련 출원의 상호 참조

[0002] 본 출원은 35 U.S.C. § 119(e) 하에서 2015년 7월 22일에 출원되고 발명의 명칭이 "TRANSFER LEARNING IN NEURAL NETWORKS"인 미국 특허 가출원 62/195,763호에 대해 우선권을 주장하며, 그 개시는 그 전체가 참조로써 본 명세서에서 명백히 인용된다.

[0003] 분야

[0004] 본 개시의 소정의 양태들은 일반적으로 머신 러닝에 관한 것으로, 보다 구체적으로 뉴럴 네트워크들에서의 러닝을 트랜스퍼하기 위한 개선된 시스템들 및 방법들에 관한 것이다.

## 배경 기술

[0005] 인공 뉴런들 (즉, 뉴런 모델들) 의 상호 연결된 그룹을 포함할 수도 있는 인공 뉴럴 네트워크는 연산 디바이스 이거나 연산 디바이스에 의해 수행될 방법을 표현한다.

[0006] 콘볼루션 뉴럴 네트워크들은 일종의 피드-포워드 인공 뉴럴 네트워크이다. 콘볼루션 뉴럴 네트워크들은 각각 수용 필드 (receptive field) 를 가지고 집합적으로 입력 공간을 타일링하는 뉴런들의 집합들을 포함할 수도 있다. 콘볼루션 뉴럴 네트워크들 (CNNs) 은 다수의 애플리케이션들을 갖는다. 특히, CNN 들은 패턴 인식 및 분류의 영역에서 널리 사용되어 왔다.

[0007] 딥 빌리프 네트워크들 (deep belief networks) 및 딥 콘볼루션 네트워크들과 같은 딥 러닝 아키텍처들은 계층화된 뉴럴 네트워크 아키텍처들이며, 여기서 제 1 계층의 뉴런들의 출력은 제 2 계층의 뉴런들에 대한 입력이 되고, 제 2 계층의 뉴런들의 출력은 제 3 계층의 뉴런들에 대한 입력이 되는 등이다. 딥 뉴럴 네트워크들은 피쳐들의 하이어라키 (hierarchy) 를 인식하도록 트레이닝될 수도 있고, 따라서 그들은 오브젝트 인식 애플리케이션들에서 점점 더 많이 사용되어 왔다. 콘볼루션 뉴럴 네트워크들처럼, 이들 딥 러닝 아키텍처들에서의 연산은 하나 이상의 연산 체인들로 구성될 수도 있는 일군의 프로세싱 노드들을 통해 딥 러닝될 수도 있다. 이들 다중-계층화 아키텍처들은 한 번에 하나의 계층이 트레이닝될 수도 있고, 후방 전파 (back propagation) 를 사용하여 미세 튜닝될 수도 있다.

[0008] 다른 모델들이 또한 오브젝트 인식을 위해 이용가능하다. 예를 들어, 서포트 벡터 머신들 (support vector machines: SVMs) 은 분류를 위해 적용될 수 있는 러닝 툴들 (learning tools) 이다. 서포트 벡터 머신들은 데이터를 카테고리화하는 분리 하이퍼플레인 (hyperplane) (예를 들어, 결정 바운더리) 을 포함한다. 하이퍼플레인은 감독된 러닝에 의해 정의된다. 원하는 하이퍼플레인은 트레이닝 데이터의 마진을 증가시킨다. 즉, 하이퍼플레인은 트레이닝 예들에 대한 최소 거리를 가져야 한다.

[0009] 이들 해결책들이 다수의 분류 벤치마크들에 대한 우수한 결과들을 달성하지만, 그들의 연산 복잡성은 엄청나게 높을 수 있다. 또, 모델들의 트레이닝이 힘들 수도 있다.

## 발명의 내용

[0010] 본 개시의 일 양태에서, 러닝을 트랜스퍼하기 위한 방법이 개시된다. 방법은 제 2 데이터를 수신하는 단계를 포함한다. 방법은 또한 제 1 네트워크를 통해 제 2 데이터에 대한 제 2 라벨들을 생성하는 단계를 포함



한다. 일 구성에서, 제 1 네트워크는 제 1 데이터에 대한 제 1 라벨들 상에서 미리 트레이닝되어 왔다. 더욱이, 제 2 라벨들은 제 2 네트워크를 트레이닝하기 위해 생성된다.

[0011] 본 개시의 또 다른 양태는 제 2 데이터를 수신하는 수단을 포함하는 장치에 관한 것이다. 장치는 또한 제 1 네트워크를 통해 제 2 데이터에 대한 제 2 라벨들을 생성하는 수단을 포함한다. 일 구성에서, 제 1 네트워크는 제 1 데이터에 대한 제 1 라벨들 상에서 미리 트레이닝되어 왔다. 더욱이, 제 2 라벨들은 제 2 네트워크를 트레이닝하기 위해 생성된다.

[0012] 본 개시의 또 다른 양태에서, 러닝을 트랜스퍼하기 위한 컴퓨터 프로그램 제품이 개시된다. 컴퓨터 프로그램 제품은 비일시적 프로그램 코드가 기록된 비일시적 컴퓨터 판독가능 매체를 갖는다. 프로그램 코드는 프로세서에 의해 실행되고 제 2 데이터를 수신하기 위한 프로그램 코드를 포함한다. 프로그램 코드는 제 1 네트워크를 통해 제 2 데이터에 대한 제 2 라벨들을 생성하기 위한 프로그램 코드를 포함한다. 일 구성에서, 제 1 네트워크는 제 1 데이터에 대한 제 1 라벨들 상에서 미리 트레이닝되어 왔다. 더욱이, 제 2 라벨들은 제 2 네트워크를 트레이닝하기 위해 생성된다.

[0013] 본 개시의 또 다른 양태는 메모리 유닛 및 메모리에 커플링된 하나 이상의 프로세서들을 갖는 러닝을 트랜스퍼하기 위한 디바이스에 관한 것이다. 프로세서(들)는 제 2 데이터를 수신하도록 구성된다. 프로세서(들)는 또한 제 1 네트워크를 통해 제 2 데이터에 대한 제 2 라벨들을 생성하도록 구성된다. 일 구성에서, 제 1 네트워크는 제 1 데이터에 대한 제 1 라벨들 상에서 미리 트레이닝되어 왔다. 더욱이, 제 2 라벨들은 제 2 네트워크를 트레이닝하기 위해 생성된다.

[0014] 본 개시의 일 양태에서, 러닝을 트랜스퍼하기 위한 방법이 개시된다. 방법은 제 2 데이터를 사용하여 제 1 네트워크에 의해 생성된 제 2 라벨들을 수신하는 단계를 포함한다. 일 구성에서, 제 1 네트워크는 제 1 라벨들 및 제 1 데이터 상에서 미리 트레이닝되어 왔다. 방법은 또한 제 2 라벨들 및 제 2 데이터 상에서 제 2 네트워크를 트레이닝하는 단계를 포함한다.

[0015] 본 개시의 또 다른 양태는 제 2 데이터를 사용하여 제 1 네트워크에 의해 생성된 제 2 라벨들을 수신하기 위한 수단을 포함하는 장치에 관한 것이다. 일 구성에서, 제 1 네트워크는 제 1 라벨들 및 제 1 데이터 상에서 미리 트레이닝되어 왔다. 장치는 또한 제 2 라벨들 및 제 2 데이터 상에서 제 2 네트워크를 트레이닝하기 위한 수단을 포함한다.

[0016] 본 개시의 또 다른 양태에서, 러닝을 트랜스퍼하기 위한 컴퓨터 프로그램 제품이 개시된다. 컴퓨터 프로그램 제품은 비일시적 프로그램 코드가 기록된 비일시적 컴퓨터 판독가능 매체를 갖는다. 프로그램 코드는 프로세서에 의해 실행되며, 제 2 데이터를 사용하여 제 1 네트워크에 의해 생성된 제 2 라벨들을 수신하기 위한 프로그램 코드를 포함한다. 일 구성에서, 제 1 네트워크는 제 1 라벨들 및 제 1 데이터 상에서 미리 트레이닝되어 왔다. 프로그램 코드는 또한 제 2 라벨들 및 제 2 데이터 상에서 제 2 네트워크를 트레이닝하기 위한 프로그램 코드를 포함한다.

[0017] 본 개시의 또 다른 양태는 메모리 및 메모리에 커플링된 하나 이상의 프로세서들을 갖는 러닝을 트랜스퍼하기 위한 장치에 관한 것이다. 프로세서(들)는 제 2 데이터를 사용하여 제 1 네트워크에 의해 생성된 제 2 라벨들을 수신하도록 구성된다. 일 구성에서, 제 1 네트워크는 제 1 라벨들 및 제 1 데이터 상에서 미리 트레이닝되어 왔다. 프로세서(들)는 또한 제 2 라벨들 및 제 2 데이터 상에서 제 2 네트워크를 트레이닝하도록 구성된다.

[0018] 본 개시의 추가적인 특징들 및 이점들이 하기에 설명될 것이다. 당업자에 의해, 본 개시가 본 개시의 동일한 목적을 수행하기 위한 다른 구조들을 수정하거나 설계하는 기초로서 쉽게 이용될 수도 있음이 이해되어야만 한다. 당업자라면, 이러한 등가의 구성들이 첨부된 청구항들에서 설명되는 본 개시의 교시들을 벗어나지 않는다는 것을 알 수 있어야 한다. 동작의 구성 및 방법 양자에 관한 본 개시의 특징으로 여겨지는 신규의 특징들은, 다른 목적들 및 이점들과 함께, 첨부된 도면과 연계하여 고려될 때 하기의 설명으로부터 더욱 이해될 것이다. 그러나, 각각의 도면은 도해 및 설명의 목적으로만 제공된 것이며, 본 개시의 제한들의 정의로서 의도된 것은 아님이 명확히 이해되어야 한다.

### 도면의 간단한 설명

[0019] 본 개시의 특징들, 속성, 및 이점들은, 도면들과 연계하여 취해지는 경우, 하기에 제시된 상세한 설명으로부터 더욱 자명해질 것이며, 도면들에서, 유사한 도면 부호들은 명세서 전체에 걸쳐 그에 대응하는 것을 식별한다.

도 1 은 본 개시의 소정의 양태들에 따른 범용 프로세서를 포함하는, 시스템-온-칩 (system-on-a-chip; SOC) 을 사용하여 뉴럴 네트워크를 설계하는 예시의 구현을 도시한다.

도 2 는 본 개시의 양태들에 따른 시스템의 예시의 구현을 도시한다.

도 3a 는 본 개시의 양태들에 따른 뉴럴 네트워크를 도시하는 다이어그램이다.

도 3b 는 본 개시의 양태들에 따른 예시적인 딥 콘볼루션 네트워크 (DCN) 를 도시하는 블록도이다.

도 4 는 본 개시의 양태들에 따른 인공 지능 (AI) 기능들을 모듈화할 수도 있는 예시적인 소프트웨어 아키텍처를 도시하는 블록도이다.

도 5 는 본 개시의 양태들에 따른 스마트폰상의 AI 애플리케이션의 런-타임 동작을 도시하는 블록도이다.

도 6 은 본 개시의 양태들에 따른 러닝 트랜스퍼를 위한 흐름도이다.

도 7a 내지 도 7e 는 본 개시의 양태들에 따른 상이한 네트워크들의 예들을 도시한다.

도 8 및 도 9 는 본 개시의 양태들에 따른 러닝 트랜스퍼를 위한 방법들을 도시한다.

### 발명을 실시하기 위한 구체적인 내용

[0020] 첨부된 도면들과 연계하여 하기에 설명되는 상세한 설명은, 여러 구성들의 설명으로서 의도된 것이며 본원에서 설명되는 개념들이 실시될 수도 있는 유일한 구성들만을 나타내도록 의도된 것은 아니다. 상세한 설명은 여러 개념들의 완전한 이해를 제공하기 위한 목적으로 특정 세부사항들을 포함한다. 그러나, 이들 개념들이 이들 특정 세부사항들 없이 실시될 수도 있음이 당업자에게는 명백할 것이다. 일부 사례들에서, 이러한 개념들을 모호하게 하는 것을 방지하기 위해 공지의 구조들 및 컴포넌트들이 블록도의 형태로 도시된다.

[0021] 본 개시물의 임의의 다른 양태들과 독립적으로 또는 결합하여 구현되는지 여부에 관계없이, 본 교시들에 기초하여, 당업자들은 본 개시의 범위가 본 개시의 임의의 양태를 커버하도록 의도된다는 것을 인정해야 한다. 예를 들어, 제시된 임의의 개수의 양태들을 이용하여 장치가 구현될 수도 있거나 방법이 실시될 수도 있다. 또한, 본 개시의 범위는 본원에 제시된 개시의 다양한 양태들에 더해 또는 그 외에 다른 구조, 기능성, 또는 구조와 기능성을 이용하여 실시되는 그러한 장치 또는 방법을 커버하고자 한다. 본원에 개시된 개시의 임의의 양태는 청구항의 하나 이상의 요소들에 의해 구체화될 수도 있다.

[0022] 단어 "예시적인" 은 본원에서 "예, 예시, 또는 설명의 역할을 하는" 것을 의미하기 위해 사용된다. "예시적인" 으로 본원에서 설명된 임의의 양태는 반드시 다른 양태들보다 바람직하거나 이로운 것으로 해석되지는 않는다.

[0023] 특정 양태들이 본원에서 설명되지만, 이러한 양태들의 많은 변형예들 및 치환예들이 본 개시의 범위 내에 속한다. 바람직한 양태들의 일부 이득들 및 이점들이 언급되었지만, 본 개시의 범위는 특정 이득들, 이용들, 또는 목적들로 제한되고자 하지 않는다. 오히려, 본 개시의 양태들은 상이한 기술들, 시스템 구성들, 네트워크들, 및 프로토콜들에 널리 적용되고자 하며, 본 개시물의 양태들 중 일부는 도면들에서 그리고 다음의 바람직한 양태들의 설명에서 예로서 예시된다. 상세한 설명 및 도면들은 제한하는 것이기 보다는 단지 본 개시물의 예시일 뿐이며, 본 개시물의 범위는 첨부된 청구항들 및 그의 등가물들에 의해 정의된다.

[0024] 뉴럴 네트워크들은 입력으로부터 오브젝트들을 분류하기 위한 라벨들 및 상응하는 데이터를 포함하는 트레이닝 세트 상에서 트레이닝될 수 있다. 예를 들어, 제 1 뉴럴 네트워크는 상이한 유형의 자동차들을 식별하기 위해 자동차의 라벨링된 이미지 상에서 트레이닝될 수 있다. 일부 경우들에서, 네트워크가 트레이닝된 이후에 새로운 클래스를 추가하고 및/또는 기존 클래스의 바운더리들을 수정하는 것이 바람직할 수도 있다. 그럼에도 불구하고, 여러 가지 이유로, 트레이닝 세트는 제 1 뉴럴 네트워크가 트레이닝된 이후에 더 이상 이용가능하지 않을 수 있다. 따라서, 그 트레이닝 세트는 더 이상 이용가능하지 않기 때문에, 네트워크가 트레이닝된 이후에는 새로운 클래스들을 추가하고 및/또는 기존 클래스들의 바운더리들을 수정하기 위한 점진적인 러닝이 네트워크 상에서 수행되지 않을 수도 있다.

[0025] 그러므로, 제 2 뉴럴 네트워크에 의한 점진적인 러닝을 허용하기 위해 제 1 뉴럴 네트워크의 러닝을 제 2 뉴럴 네트워크에 트랜스퍼하는 것이 바람직할 수 있다. 예를 들어, 제 1 뉴럴 네트워크를 트레이닝한 후에 원래의 트레이닝 세트가 이용 가능하지 않을 수도 있기 때문에, 제 1 뉴럴 네트워크는 제 1 뉴럴 네트워크에 근사하는 제 2 뉴럴 네트워크를 트레이닝시키기 위해 새로운 데이터를 라벨링하도록 특정될 수 있다. 다음, 제 2

뉴럴 네트워크는 점진적인 러닝 또는 기타 태스크를 위해 사용될 수 있다.

[0026] 도 1은 본 개시의 소정의 양태들에 따른 범용 프로세서 (CPU) 또는 다중-코어 범용 프로세서들 (CPUs) (102)를 포함할 수도 있는, 시스템-온-칩 (SOC) (100)을 사용하는 상술된 딥 뉴럴 네트워크 아키텍처의 예시의 구현 (100)을 도시한다. 변수들 (예를 들어, 뉴럴 신호들 및 시냅스 가중치들), 연산 디바이스와 연관된 시스템 파라미터들 (예를 들어, 가중치들을 갖는 뉴럴 네트워크), 지연들, 주파수 빈 정보, 및 태스크 정보는 뉴럴 프로세싱 유닛 (NPU)과 연관된 메모리 블록 (108) 또는 전용 메모리 블록 (118)에 저장될 수 있다. 범용 프로세서 (102)에서 실행되는 명령들은 CPU (102)와 연관된 프로그램 메모리로부터 로딩될 수도 있거나 전용 메모리 블록 (118)으로부터 로딩될 수도 있다. SOC (100)는 또한 그래픽 프로세싱 유닛 (GPU) (104), 디지털 신호 프로세서 (DSP) (106), 제 4세대 롱 텀 에볼루션 (4G LTE) 연결성, 비허가 Wi-Fi 연결성, USB 연결성, 블루투스 연결성 등을 포함할 수도 있는 연결성 블록 (110), 및 예를 들어, 제스처들을 검출 및 인식할 수도 있는 멀티미디어 프로세서 (112)와 같은 특정의 기능들로 테일러링된 추가적인 프로세싱 블록들을 포함할 수도 있다. SOC (100)는 또한 센서 프로세서 (114), 이미지 신호 프로세서들 (ISP들), 및/또는 글로벌 포지셔닝 시스템을 포함할 수도 있는 네비게이션 (120)을 포함할 수도 있다.

[0027] SOC (100)는 ARM 명령 세트에 기초할 수도 있다. 본 개시의 양태에서, 범용 프로세서 (102)로 로딩된 명령들은 제 2 데이터를 사용하여 제 1 네트워크에 의해 생성된 제 2 라벨들을 수신하기 위한 코드를 포함할 수도 있다. 제 1 네트워크는 제 1 라벨들 및 제 1 데이터 상에서 미리 트레이닝되었다. 범용 프로세서 (102)로 로딩된 명령들은 또한 제 2 라벨들 및 제 2 데이터 상에서 제 2 네트워크를 트레이닝하기 위한 코드를 포함할 수도 있다.

[0028] 또 다른 양태에서, 범용 프로세서 (102)에 로딩된 명령들은 제 2 데이터를 수신하기 위한 코드 및 제 1 네트워크를 통해 제 2 데이터에 대한 제 2 라벨들을 생성하기 위한 코드를 포함한다. 제 1 네트워크는 제 1 데이터에 대한 제 1 라벨들 상에서 미리 트레이닝되었다. 제 2 라벨들은 제 2 네트워크를 트레이닝하기 위해 생성된다. 제 2 데이터는 라벨링되지 않을 수 있거나 또는 그 데이터는 무시된 라벨들을 갖도록 라벨링될 수 있다. 범용 프로세서 (102)로 로딩된 명령들은 또한 제 2 라벨들 및 제 2 데이터 상에서 제 2 네트워크를 트레이닝하기 위한 코드를 포함할 수도 있다.

[0029] 도 2는 본 개시의 소정의 양태들에 따른 시스템 (200)의 예시적인 구현을 도시한다. 도 2에 도시된 바와 같이, 시스템 (200)은 여기에 기술된 방법들의 다양한 동작들을 수행할 수도 있는 다수의 로컬 프로세싱 유닛들 (202)을 가질 수도 있다. 각각의 로컬 프로세싱 유닛 (202)은 뉴럴 네트워크의 파라미터들을 저장할 수도 있는 로컬 상태 메모리 (204) 및 로컬 파라미터 메모리 (206)를 포함할 수도 있다. 또한, 로컬 프로세싱 유닛 (202)은 로컬 모델 프로그램을 저장하기 위한 로컬 (뉴럴) 모델 프로그램 (local model program; LMP) 메모리 (208), 로컬 러닝 프로그램을 저장하기 위한 로컬 러닝 프로그램 (local learning program; LLP) 메모리 (210), 및 로컬 연결 메모리 (212)를 가질 수도 있다. 또한, 도 2에 도시된 바와 같이, 각각의 로컬 프로세싱 유닛 (202)은 로컬 프로세싱 유닛의 로컬 메모리들을 위한 구성을 제공하기 위한 구성 프로세서 유닛 (214), 및 로컬 프로세싱 유닛들 (202) 사이에 라우팅을 제공하는 라우팅 연결 프로세싱 유닛 (216)과 인터페이스할 수도 있다.

[0030] 딥 러닝 아키텍처들은 각각의 계층에서 추상화의 연속적으로 더 높은 레벨들에서 입력들을 표현하기 위해 러닝에 의한 오브젝트 인식 태스크를 수행하고, 이것에 의해 입력 데이터의 유용한 특징 표현을 구축할 수도 있다. 이러한 방식으로, 딥 러닝은 전통적인 머신 러닝의 주요한 병목을 다룬다. 딥 러닝의 도래 이전에, 오브젝트 인식 문제에 대한 머신 러닝 접근법은 아마도 쉘로우 (shallow) 분류기와 결합하여, 인간 조작된 (human engineered) 특징들에 크게 의존했을 수도 있다. 쉘로우 분류기는 예를 들어 2-클래스 선형 분류기일 수도 있으며, 여기서 특징 벡터 컴포넌트들의 가중된 합은 입력이 어느 클래스에 속하는지를 예측하기 위해 임계값과 비교될 수도 있다. 인간 조작된 특징들은 도메인 전문 기술을 갖는 엔지니어들에 의해 특정의 문제 도메인으로 테일러링된 템플릿들 또는 커널들일 수도 있다. 딥 러닝 아키텍처들은, 대조적으로, 트레이닝을 통해서만 인간 엔지니어가 설계할지도 모르는 것과 유사한 특징들을 표현하는 것을 러닝할 수도 있다. 더욱이, 딥 네트워크는 인간이 고려하지 못했을 수도 있는 새로운 타입들의 특징들을 표현 및 인식하는 것을 러닝할 수도 있다.

[0031] 딥 러닝 아키텍처들은 특징들의 하이어라키 (hierarchy)를 러닝할 수도 있다. 예를 들어, 시각적 데이터가 제시되는 경우, 제 1 계층은 입력 스트림에서 에지들과 같은 간단한 특징들을 인식하는 것을 러닝할 수도 있다. 청각적 데이터가 제시되는 경우, 제 1 계층은 특정의 주파수들에서의 스펙트럼 전력을 인식하는 것을 러닝할

수도 있다. 제 2 계층은, 제 1 계층의 출력을 입력으로서 취하여, 시각적 데이터에 대한 간단한 형상들 또는 청각적 데이터에 대한 사운드들의 조합들과 같은 특징들의 조합들을 인식하는 것을 러닝할 수도 있다. 더 높은 계층들은 시각적 데이터에서 복잡한 형상들 또는 청각적 데이터에서 단어들을 표현하는 것을 러닝할 수도 있다. 여전히 더 높은 계층들은 공통적인 시각적 오브젝트들 또는 발화된 어구들을 인식하는 것을 러닝할 수도 있다.

[0032] 딥 러닝 아키텍처들은 자연의 하이이라키 구조를 갖는 문제들에 적용될 때 특히 잘 수행될 수도 있다. 예를 들어, 자동차들의 분류는 처음에 바퀴들, 전면 유리들, 및 다른 특징들을 인식하는 것을 러닝하는 것으로부터 이익을 얻을 수도 있다. 이들 특징들은 자동차들, 트럭들, 및 비행기들을 인식하기 위해 상이한 방식들로 더 높은 계층들에서 결합될 수도 있다.

[0033] 뉴럴 네트워크들은 다양한 연결성 패턴들로 설계될 수도 있다. 피드-포워드 네트워크들에서, 정보는 더 낮은 계층으로부터 더 높은 계층으로 트랜스퍼되며, 주어진 계층에서의 각각의 뉴런은 더 높은 계층들에서의 뉴런들로 통신한다. 하이이라키 표현은 상술된 바와 같이 피드-포워드 네트워크의 연속적인 계층들에서 구축될 수도 있다. 뉴럴 네트워크들은 또한 회귀 또는 피드백 (또한 톱-다운 (top-down) 으로도 불림) 연결들을 가질 수도 있다. 회귀 연결에서, 주어진 계층에서의 뉴런으로부터의 출력은 동일한 계층에서의 다른 뉴런으로 통신된다. 회귀 아키텍처는 시간에 있어서 전개되는 패턴들을 인식하는데 있어서 도움이 될 수도 있다. 주어진 계층에서의 뉴런으로부터 더 낮은 계층에서의 뉴런으로의 연결은 피드백 (또는 톱-다운) 연결로 지칭된다. 다수의 피드백 연결들을 갖는 네트워크는 고레벨 개념이 입력의 특징의 저레벨 특징들을 판별하는데 도움을 줄 수도 있는 경우에 도움이 될 수도 있다.

[0034] 도 3a 를 참조하면, 뉴럴 네트워크의 계층들 사이의 연결들은 완전히 연결되거나 (302) 로컬로 연결될 수도 있다 (304). 완전히 연결된 네트워크 (302) 에서, 주어진 계층에서의 뉴런은 다음의 계층에서의 모든 뉴런으로 그것의 출력을 통신할 수도 있다. 대안적으로, 로컬로 연결된 네트워크 (304) 에서는, 주어진 계층에서의 뉴런은 다음의 계층에서의 제한된 수의 뉴런들로 연결될 수도 있다. 콘볼루션 네트워크 (306) 는 로컬로 연결될 수도 있고, 더욱이 주어진 계층에서의 각 뉴런과 연관된 연결 강도들이 공유되는 특수한 케이스 (예를 들어, 308) 이다. 더욱 일반적으로, 네트워크의 로컬로 연결된 계층은 계층에서의 각 뉴런이 동일하거나 유사한 연결성 패턴을 갖지만, 상이한 값들 (예를 들어, 310, 312, 314, 및 316) 을 가질 수도 있는 연결 강도들을 갖도록 구성될 수도 있다. 로컬로 연결된 연결성 패턴은 더 높은 계층에서 공간적으로 별개의 각각의 필드들을 발생시킬 수도 있으며, 이는 주어진 영역에서의 더 높은 계층 뉴런들이 네트워크에 대한 총 입력의 제한된 부분의 특성들로 트레이닝을 통해 튜닝되는 입력들을 수신할 수도 있기 때문이다.

[0035] 로컬로 연결된 뉴럴 네트워크들은 입력들의 공간적 로케이션들이 의미있는 문제들에 잘 적합될 수도 있다. 예를 들어, 자동차-탐재 카메라로부터 시각적 특징들을 인식하도록 설계된 네트워크 (300) 는 상이한 특성들을 갖는 높은 계층 뉴런들을, 이미지의 하위 대 상위 부분과의 그들의 연관에 따라 개발할 수도 있다. 예를 들어, 이미지의 하위 부분과 연관된 뉴런들은 차선 표시들을 인식하는 것을 러닝할 수도 있는 반면, 이미지의 상위 부분과 연관된 뉴런들은 교통 신호등, 교통 표지판, 등을 인식하는 것을 러닝할 수도 있다.

[0036] DCN 은 감독된 러닝으로 트레이닝될 수도 있다. 트레이닝 동안, DCN 은 속도 제한 표지판의 크로핑된 이미지와 같은 이미지 (326) 가 제시될 수도 있고, "포워드 패스 (forward pass)" 는 그 후 출력 (328) 을 생성하기 위해 컴퓨팅될 수도 있다. 출력 (328) 은 "표지판", "60", 및 "100" 과 같은 특징들에 대응하는 값들의 벡터일 수도 있다. 네트워크 설계자는 DCN 이 출력 특징 벡터, 예를 들어 트레이닝되었던 네트워크 (300) 에 대한 출력 (328) 에서 보여진 바와 같은 "표지판" 및 "60" 에 대응하는 것들에서 뉴런들의 일부에 대해 높은 스코어를 출력하기를 원할 수도 있다. 트레이닝 전에, DCN 에 의해 생성된 출력은 정확하지 않기 쉽고, 따라서 에러가 실제의 출력과 목표 출력 사이에 계산될 수도 있다. DCN 의 가중치들은 그 후 DCN 의 출력 스코어들이 목표와 더 가깝게 정렬되도록 조정될 수도 있다.

[0037] 가중치들을 적절하게 조정하기 위해, 러닝 알고리즘은 가중치들에 대한 기울기 벡터를 컴퓨팅할 수도 있다. 기울기는 가중치가 약간 조정되는 경우에 에러가 증가하거나 감소할 양을 나타낼 수도 있다. 상부 (top) 계층에서, 기울기는 페널티메이트 (penultimate) 계층에서의 활성화된 뉴런 및 출력 계층에서의 뉴런을 연결하는 가중치의 값에 직접 대응할 수도 있다. 더 낮은 계층들에서, 기울기는 가중치들의 값에 및 더 높은 계층들의 컴퓨팅된 에러 기울기들에 의존할 수도 있다. 가중치들은 그 후 에러를 감소시키기 위해 조정될 수도 있다. 가중치들을 조정하는 이러한 방식은 그것이 뉴럴 네트워크를 통해 "백워드 패스" 를 수반하기 때문에 "역 전파 (back propagation)" 로 지칭될 수도 있다.



- [0038] 실제로, 가중치들의 에러 기울기는 계산된 기울기가 실제의 에러 기울기에 근사화하도록 적은 수의 예들에 대해 계산될 수도 있다. 이러한 근사화 방법은 확률적 기울기 하강으로 지칭될 수도 있다. 확률적 기울기 하강은 전체 시스템의 달성가능한 에러 레이트가 감소하기를 중단한 때까지 또는 에러 레이트가 목표 레벨에 도달한 때까지 반복될 수도 있다.
- [0039] 러닝 후에, DCN 에는 새로운 이미지들 (326) 이 제시될 수도 있고, 네트워크를 통한 포워드 패스는 DCN 의 추론 또는 예측이 고려될 수도 있는 출력 (328) 을 산출할 수도 있다.
- [0040] 딥 빌리프 네트워크들 (Deep belief networks; DBNs) 은 다수의 계층들의 은닉된 노드들을 포함하는 확률 모델들이다. DBN 들은 트레이닝 데이터 세트들의 하이얼라키 표현을 추출하기 위해 사용될 수도 있다. DBN 은 제한된 볼츠만 머신들 (Restricted Boltzmann Machines: RBMs) 의 계층들을 적층함으로써 획득될 수도 있다. RBM 은 입력들의 세트를 통한 확률 분포를 러닝할 수 있는 일 타입의 인공 뉴럴 네트워크다. RBM 들은 각 입력이 카테고리화되어야 하는 클래스에 대한 정보의 부재 시에 확률 분포를 러닝할 수 있기 때문에, RBM 들은 종종 비감독 (unsupervised) 러닝에 사용된다. 하이브리드 비감독 및 감독 파라다임을 사용하여, DBN 의 저부 RBM 들은 비감독 방식으로 트레이닝될 수도 있고 특정 추출기들로서 작용할 수도 있으며, 상부 RBM 은 (이전의 계층 및 목표 클래스들로부터의 입력들의 조인트 분포에 대해) 감독 방식으로 트레이닝될 수도 있다.
- [0041] 딥 콘볼루션 네트워크들 (DCNs) 은 추가적인 풀링 (pooling) 및 정규화 계층들로 구성된 콘볼루션 네트워크들 중의 네트워크들이다. DCN 들은 다수의 태스크들에 대한 종래 기술의 성능을 달성했다. DCN 들은 입력 및 출력 목표들 양자 모두가 다수의 전형들에 대해 알려져 있고 기울기 하강 방법들을 사용하여 네트워크의 가중치들을 변경하기 위해 사용되는 감독 러닝을 사용하여 트레이닝될 수 있다.
- [0042] DCN 들은 피드-포워드 네트워크들일 수도 있다. 또, 상술된 바와 같이, DCN 의 제 1 계층에서의 뉴런으로부터 다음의 더 높은 계층에서의 뉴런들의 그룹으로의 연결들은 제 1 계층에서의 뉴런들에 걸쳐 공유된다. DCN 들의 피드-포워드 및 공유된 연결들은 고속 프로세싱을 위해 활용될 수도 있다. DCN 의 계산적 짐은 예를 들어 회귀 또는 피드백 연결들을 포함하는 유사한 사이즈의 뉴럴 네트워크의 그것보다 훨씬 작을 수도 있다.
- [0043] 콘볼루션 네트워크의 각 계층의 프로세싱은 공간적 가변 템플릿 또는 기저 투영으로 고려될 수도 있다. 입력이 먼저 컬러 이미지의 적색, 녹색 및 청색 채널들과 같은 다수의 채널들로 분해된다면, 그 입력에 대해 트레이닝된 콘볼루션 네트워크는 이미지의 축들을 따른 2 개의 공간적 차원들 및 컬러 정보를 캡처하는 제 3 차원을 갖는 3 차원적인 것으로 고려될 수도 있다. 콘볼루션 연결들의 출력들은 후속 계층 (318, 320, 및 322) 에서 특징 맵을 형성하는 것으로 고려될 수도 있고, 특징 맵 (예를 들어, 320) 의 각 엘리먼트는 이전의 계층 (예를 들어, 318) 에서의 뉴런들의 범위로부터 및 다수의 채널들 각각으로부터 입력을 수신한다. 특징 맵에서의 값들은 정류 (rectification),  $\max(0, x)$  와 같은 비선형성으로 더욱 프로세싱될 수도 있다. 인접한 뉴런들로부터의 값들은 더욱 풀링될 수도 있으며 (324), 이것은 다운 샘플링에 대응하고, 추가적인 로컬 불변성 및 차원성 감소를 제공할 수도 있다. 화이트닝 (whitening) 에 대응하는 정규화가 또한 특징 맵에서의 뉴런들 사이의 측면 억제 (lateral inhibition) 를 통해 적용될 수도 있다.
- [0044] 딥 러닝 아키텍처들의 성능은 더 많은 라벨링된 데이터 포인트들이 이용가능하게 됨에 따라 또는 계산 전력이 증가함에 따라 증가할 수도 있다. 현대의 딥 뉴럴 네트워크들은 단지 15년 전에 통상적인 연구자들에게 이용가능했던 것보다 수천배 더 큰 컴퓨팅 자원들로 루틴하게 트레이닝된다. 새로운 아키텍처들 및 트레이닝 파라다임들은 딥 러닝의 성능을 더욱 부스팅할 수도 있다. 정류된 선형 유닛들은 소실하는 기울기들 (vanishing gradients) 로서 알려진 트레이닝 문제를 감소시킬 수도 있다. 새로운 트레이닝 기법들은 오버-피팅 (over-fitting) 을 감소시킬 수도 있고, 따라서 더 큰 모델들이 더 양호한 일반화를 달성하는 것을 가능하게 할 수도 있다. 캡슐화 기법들은 주어진 각각의 필드에서 데이터를 추출하고 전체 성능을 더욱 부스팅할 수도 있다.
- [0045] 도 3b 는 예시적인 딥 콘볼루션 네트워크 (350) 를 도시하는 블록도이다. 딥 콘볼루션 네트워크 (350) 는 연결성 및 가중치 공유에 기초하여 다수의 상이한 타입들의 계층들을 포함할 수도 있다. 도 3b 에 도시된 바와 같이, 예시적인 딥 콘볼루션 네트워크 (350) 는 다수의 콘볼루션 블록들 (예를 들어, C1 및 C2) 을 포함한다. 콘볼루션 블록들 각각은 콘볼루션 계층, 정규화 계층 (LNorm), 및 풀링 계층으로 구성될 수도 있다. 콘볼루션 계층들은 특징 맵을 생성하기 위해 입력 데이터에 적용될 수도 있는 하나 이상의 콘볼루션 필터들을 포함할 수도 있다. 단지 2 개의 콘볼루션 블록들이 도시되지만, 본 개시는 그렇게 제한하지 않고, 대신에, 임의의 수의 콘볼루션 블록들이 설계 선호에 따라 딥 콘볼루션 네트워크 (350) 에 포함될 수도 있다. 정규

화 계층은 컨볼루션 필터들의 출력을 정규화하기 위해 사용될 수도 있다. 예를 들어, 정규화 계층은 화이트닝 또는 측면 억제제를 제공할 수도 있다. 풀링 계층은 로컬 불변성 및 차원성 감소를 위해 공간에 대해 다운샘플링 집성 (aggregation) 을 제공할 수도 있다.

[0046] 예를 들어, 딥 컨볼루션 네트워크의 병렬 필터 뱅크들은 고성능 및 저전력 소비를 달성하기 위해, 선택적으로 ARM 명령 세트에 기초하여, SOC (100) 의 CPU (102) 또는 GPU (104) 상에 로딩될 수도 있다. 대안적인 실시형태들에서, 병렬 필터 뱅크들은 SOC (100) 의 DSP (106) 또는 ISP (116) 상에 로딩될 수도 있다. 또, DCN 은 센서들 (114) 및 네비게이션 (120) 에 전용된 프로세싱 블록들과 같은, SOC 상에 존재할 수도 있는 다른 프로세싱 블록들을 액세스할 수도 있다.

[0047] 딥 컨볼루션 네트워크 (350) 는 또한 하나 이상의 완전히 연결된 계층들 (예를 들어, FC1 및 FC2) 을 포함할 수도 있다. 딥 컨볼루션 네트워크 (350) 는 로지스틱 회귀분석 (logistic regression: LR) 계층을 더 포함할 수도 있다. 딥 컨볼루션 네트워크 (350) 의 각 계층 사이에 업데이트되어야 하는 가중치들 (도시하지 않음) 이 존재한다. 각 계층의 출력은 제 1 컨볼루션 블록 (C1) 에서 공급된 입력 데이터 (예를 들어, 이미지들, 오디오, 비디오, 센서 데이터 및/또는 다른 입력 데이터) 로부터 계층적 특징 표현들을 러닝하기 위해 딥 컨볼루션 네트워크 (350) 에서 계속되는 계층의 입력으로서 작용할 수도 있다.

[0048] 도 4 는 인공 지능 (AI) 기능들을 모듈화할 수도 있는 예시적인 소프트웨어 아키텍처를 도시하는 블록도이다. 그 아키텍처를 사용하여, SOC (420) 의 여러 프로세싱 블록들 (예를 들어, CPU (422), DSP (424), GPU (426) 및/또는 NPU (428)) 로 하여금 애플리케이션 (402) 의 런-타임 동작 동안 계산들을 지원하는 것을 수행하게 할 수도 있는 애플리케이션들 (402) 이 설계될 수도 있다.

[0049] AI 애플리케이션 (402) 은 예를 들어 디바이스가 현재 동작하는 로케이션을 나타내는 장면의 검출 및 인식을 제공할 수도 있는 사용자 공간 (404) 에서 정의된 함수들을 호출하도록 구성될 수도 있다. AI 애플리케이션 (402) 은 예를 들어 그 인식된 장면이 사무실, 강의실, 식당, 또는 호수와 같은 옥외 설정인지 여부에 따라 상이하게 마이크론 및 카메라를 구성할 수도 있다. AI 애플리케이션 (402) 은 현재의 장면의 추정을 제공하기 위해 장면검출 (SceneDetect) 애플리케이션 프로그래밍 인터페이스 (API) (406) 에서 정의된 라이브러리와 연관된 컴파일링된 프로그램 코드에 대한 요청을 행할 수도 있다. 이러한 요청은 예를 들어 비디오 및 포지셔닝 데이터에 기초하여 장면 추정들을 제공하도록 구성된 딥 뉴럴 네트워크의 출력에 궁극적으로 의존할 수도 있다.

[0050] 런타임 프레임워크의 컴파일링된 코드일 수도 있는 런-타임 엔진 (408) 은 AI 애플리케이션 (402) 에 대해 추가로 액세스가능할 수도 있다. AI 애플리케이션 (402) 은 런-타임 엔진으로 하여금, 예를 들어, 특정의 시간 간격으로 또는 애플리케이션의 사용자 인터페이스에 의해 검출된 이벤트에 의해 트리거되어 장면 추정을 요청하게 할 수도 있다. 장면을 추정하도록 하게 되는 경우, 런-타임 엔진은 차례로 SOC (420) 상에서 실행하는, 리눅스 커널 (412) 과 같은 운영 시스템 (410) 으로 신호를 전송할 수도 있다. 운영 시스템 (410) 은, 차례로, 계산이 CPU (422), DSP (424), GPU (426), NPU (428), 또는 이들의 일부 조합상에서 수행되게 할 수도 있다. CPU (422) 는 운영 시스템에 의해 직접 액세스될 수도 있고, 다른 프로세싱 블록들은 DSP (424) 에 대한, GPU (426) 에 대한, 또는 NPU (428) 에 대한 구동기 (414-418) 와 같은 구동기를 통해 액세스될 수도 있다. 예시적인 예에서, 딥 뉴럴 네트워크는 CPU (422) 및 GPU (426) 와 같은 프로세싱 블록들의 조합상에서 실행하도록 구성될 수도 있다.

[0051] 도 5 는 스마트폰 (502) 상의 AI 애플리케이션의 런-타임 동작 (500)을 도시하는 블록도이다. AI 애플리케이션은 이미지의 포맷을 변환하고 (506) 및 그 후 이미지를 크로핑 및/또는 리사이징하도록 (508) (예를 들어, 자바 프로그래밍 언어를 사용하여) 구성될 수도 있는 프리-프로세스 모듈 (504) 을 포함할 수도 있다. 프리-프로세싱된 이미지는 그 후 시각적 입력에 기초하여 장면들을 검출 및 분류하기 위해 (예를 들어, C 프로그래밍 언어를 사용하여) 구성될 수도 있는 장면검출 백엔드 (Backend) 엔진 (512) 을 포함하는 분류 애플리케이션 (510) 으로 통신될 수도 있다. 장면검출 백엔드 엔진 (512) 은 스케일링 (516) 및 크로핑 (518) 에 의해 이미지를 더욱 사전프로세싱하도록 (514) 구성될 수도 있다. 예를 들어, 이미지는 결과의 이미지가 224 화소 바이 (by) 224 화소가 되도록 스케일링 및 크로핑될 수도 있다. 이들 차원들은 뉴럴 네트워크의 입력 차원들로 맵핑할 수도 있다. 뉴럴 네트워크는 SOC (100) 의 여러 프로세싱 블록들로 하여금 딥 뉴럴 네트워크로 이미지 화소들을 더욱 프로세싱하게 하기 위해 딥 뉴럴 네트워크 블록 (520) 에 의해 구성될 수도 있다. 딥 뉴럴 네트워크의 결과들은 그 후 임계값이 적용되고 (522) 분류 애플리케이션 (510) 에서의 지수적 평활화 블록 (524) 을 통해 통과될 수도 있다. 평활화된 결과들은 그 후 스마트폰 (502) 의 설정들 및/또는 디스플레이

의 변경을 야기할 수도 있다.

- [0052] 일 구성에서, 머신 러닝 모델과 같은 모델은 라벨링되지 않은 제 2 데이터를 수신하도록 구성된다.      부가적으로, 모델은 제 1 네트워크를 통해 제 2 데이터에 대한 제 2 라벨들을 생성하도록 구성될 수 있다.      또한, 제 1 네트워크는 제 1 데이터에 대한 제 1 라벨들 상에서 미리 트레이닝되었을 수 있다.      더욱이, 모델은 제 2 데이터 및 제 2 라벨들 상에서 제 2 네트워크를 트레이닝하도록 구성될 수도 있다.      제 1 네트워크 및 제 2 네트워크는 동일한 디바이스 상에서 정의될 수 있거나 또는 상이한 디바이스들 상에서 정의될 수 있음에 유의해야 한다.
- [0053] 모델은 생성하는 수단 및/또는 수신하는 수단을 포함한다.      하나의 양태에서, 생성하는 수단 및/또는 수신하는 수단은 범용 프로세서 (102), 범용 프로세서 (102) 와 연관된 프로그램 메모리, 메모리 블록 (118), 로컬 프로세싱 유닛들 (202), 및 또는 기재된 기능들을 수행하도록 구성된 라우팅 연결 프로세싱 유닛들 (216) 일 수도 있다.      또 다른 구성에서, 상기 언급된 수단은 상기 언급된 수단에 의해 언급된 기능들을 수행하도록 구성된 임의의 모듈 또는 임의의 장치일 수도 있다.
- [0054] 또 다른 구성에서, 머신 러닝 모델과 같은 모델은 제 2 데이터를 사용하여 제 1 네트워크에 의해 생성된 제 2 라벨들을 수신하도록 구성된다.      부가적으로, 모델은 제 2 라벨들 및 제 2 데이터 상에서 제 2 네트워크를 트레이닝하도록 구성될 수 있다.      또한, 제 1 네트워크는 제 1 데이터에 대한 제 1 라벨들 상에서 미리 트레이닝되었을 수 있다.      제 1 네트워크 및 제 2 네트워크는 동일한 디바이스 상에서 정의될 수 있거나 또는 상이한 디바이스들 상에서 정의될 수 있음에 유의해야 한다.
- [0055] 모델은 수신하는 수단 및/또는 트레이닝하는 수단을 포함한다.      하나의 양태에서, 수신하는 수단 및/또는 트레이닝하는 수단은 범용 프로세서 (102), 범용 프로세서 (102) 와 연관된 프로그램 메모리, 메모리 블록 (118), 로컬 프로세싱 유닛들 (202), 및 또는 기재된 기능들을 수행하도록 구성된 라우팅 연결 프로세싱 유닛들 (216) 일 수도 있다.      또 다른 구성에서, 상기 언급된 수단은 상기 언급된 수단에 의해 언급된 기능들을 수행하도록 구성된 임의의 모듈 또는 임의의 장치일 수도 있다.
- [0056] 본 개시의 소정의 양태들에 따르면, 각 로컬 프로세싱 유닛 (202) 은 모델의 원하는 하나 이상의 기능적 특징들에 기초하여 모델의 파라미터들을 결정하고, 결정된 파라미터들이 추가로 적응되고, 튜닝되며, 업데이트됨에 따라 하나 이상의 기능적 특징들을 원하는 기능적 특징들로 전개하도록 구성될 수도 있다.
- [0057] 뉴럴 네트워크들에서의 러닝 트랜스퍼
- [0058] 신경 네트워크들과 같은 머신 러닝 네트워크들은 이미지 입력 및/또는 오디오 입력과 같은 입력으로부터 항목들을 분류하도록 트레이닝될 수도 있다.      일부 경우들에서, 뉴럴 네트워크는 라벨링된 데이터 상에서 역 전파를 통해 트레이닝된다.      예를 들어, 역 전파는 자동차들의 라벨링된 이미지들을 사용하여 뉴럴 네트워크를 트레이닝하여 다른 자동차 모델들을 식별할 수 있다.
- [0059] 일 예에서, 제 1 뉴럴 네트워크 (F) 는 제 1 트레이닝 세트 (D) 상에서 트레이닝될 수 있다.      이 예에서, 제 1 트레이닝 세트 (D) 는,  $D := \{(x_i, y_i)\}$ ,  $i = 1 \dots N$  이 되도록 데이터 (x) 및 대응하는 라벨 (y) 을 포함한다.      따라서, 트레이닝 이후에, 제 1 뉴럴 네트워크 (F) 는 특정 이미지 ( $x_i$ ) 에 대해 특정 라벨 ( $y_i$ ) 을 근사할 수 있다 ( $F(x_i) \approx y_i$ ).      본 개시에서, 트레이닝 세트 (예를 들어, 라벨들 및 데이터) 는 라벨링된 데이터로 지칭될 수 있다.      더욱이, 본 개시에서, 트레이닝 세트의 데이터는 이미지로 지칭될 수 있다.
- [0060] 일부 경우들에서, 제 1 트레이닝 세트 (D) 의 데이터 (x) 및/또는 라벨들 (y) 은 제 1 뉴럴 네트워크 (F) 를 트레이닝한 이후에 더 이상 이용 가능하지 않을 수 있다.      즉, 일부 경우들에서, 제 1 트레이닝 세트 (D) 의 데이터 (x) 는 라벨들 (y) 이 여전히 이용 가능한 상태에서 더 이상 이용 가능하지 않을 수 있다.      다른 경우들에서, 제 1 트레이닝 세트 (D) 의 라벨들 (y) 은 데이터 (x) 가 여전히 이용 가능한 상태에서 더 이상 이용 가능하지 않을 수 있다.      다른 예에서, 이미지들 (x) 및 라벨들 (y) 은 모두 이용 가능하지 않다.
- [0061] 일 예에서, 제 1 트레이닝 세트 (D) 는 라이선싱 제한과 같은 제한으로 인해 외부 배포를 위해 이용 가능하지 않을 수 있다.      따라서, 제 1 트레이닝 세트 (D) 가 더 이상 이용 가능하지 않기 때문에, 제 1 뉴럴 네트워크 (F) 는 점진적 러닝을 통해 업데이트되지 않을 수 있다.      본 개시의 양태들에 따르면, 점진적 러닝은 뉴럴 네트워크의 클래스의 기존 바운더리를 수정하고 및/또는 뉴럴 네트워크에 새로운 클래스를 추가하는 것을 지칭한다.
- [0062] 일 구성에서, 점진적 러닝을 수행하기 위해 원래의 트레이닝 세트가 부가적인 데이터 및 라벨들로 보강될 수 있

다. 즉, 점진적 러닝의 경우, 원래의 트레이닝 세트의 분류를 잊어버리지 않도록 원래의 트레이닝 세트를 새로운 트레이닝 세트로 보강하는 것이 바람직하다. 점진적 러닝은, 다른 점진적 러닝 기능들도 또한 고려되므로, 분류들을 보강하거나 또는 기존의 분류들의 바운더리들을 수정하는 것에 제한되지 않는다.

[0063] 일 구성에서, 제 1 트레이닝 세트 (D) 가 제 1 뉴럴 네트워크 (F) 를 트레이닝한 이후에 더 이상 이용 가능하지 않을 때, 제 2 뉴럴 네트워크 (F') 는 제 1 뉴럴 네트워크 (F) 에 근사하도록 특정된다. 구체적으로, 제 1 트레이닝 세트 (D) 가 더 이상 이용 가능하지 않을 때, 제 1 뉴럴 네트워크 (F) 는 제 2 라벨들 (y<sub>i</sub>') 을 포함하지 않는 제 2 데이터 (x<sub>i</sub>') 에 적용될 수 있다. 일 구성에서, 제 2 데이터 (x<sub>i</sub>') 는 제 1 트레이닝 세트 (D) 의 제 1 데이터 (x) 와 실질적으로 유사하거나 동일하다. 대안적으로, 제 2 데이터 (x<sub>i</sub>') 는 제 1 데이터 (x) 와 관련이 없을 수도 있다.

[0064] 더욱이, 제 2 트레이닝 세트 (D') 는 제 1 뉴럴 네트워크 (F) 를 제 2 데이터 (x<sub>i</sub>') 에 적용한 이후 생성된다. 즉, 트레이닝 이후, 제 2 라벨들 (y<sub>i</sub>') 은 제 2 데이터 (x<sub>i</sub>') 와 연관된다. 구체적으로, 트레이닝 이후, 제 2 트레이닝 세트 (D') 는 제 2 데이터 (x<sub>i</sub>') 및 제 2 라벨들 (y<sub>i</sub>') 을 포함한다. 보다 구체적으로, 제 2 데이터 (x<sub>i</sub>') 상에서 제 1 뉴럴 네트워크 (F) 를 트레이닝한 이후,  $D' := \{(x_i', y_i')\}$ ,  $i = 1 \dots N$  이다. 마지막으로, 제 2 트레이닝 세트 (D') 는 제 2 뉴럴 네트워크 (F') 를 트레이닝시키는데 사용되어, 제 2 뉴럴 네트워크 (F') 가 특정 이미지 (x<sub>i</sub>) 에 대해 특정 라벨 (y<sub>i</sub>') 을 근사하게 할 수 있다 ( $F'(x_i') \approx y_i'$ ). 일 구성에서, 제 2 뉴럴 네트워크 (F') 는 제 1 뉴럴 네트워크 (F) 과 비교하여 다른 크기를 갖는다. 예를 들어, 제 2 뉴럴 네트워크 (F') 의 크기는 제 1 뉴럴 네트워크 (F) 의 크기보다 작거나 더 클 수 있다.

[0065] 제 2 트레이닝 세트 (D') 로 트레이닝되는 것 외에도, 제 2 뉴럴 네트워크 (F') 는 제 3 트레이닝 세트 (D'') 로 트레이닝될 수 있다. 제 3 트레이닝 세트 (D'') 는 제 2 뉴럴 네트워크 (F') 에서 클래스의 하나 이상의 기존 바운더리들을 수정하는데 사용될 수 있다. 예를 들어, 제 2 트레이닝 세트 (D') 는 자동차들의 다른 클래스들에 대해 바운더리들을 설정할 수 있고, 제 3 트레이닝 세트 (D'') 는 하나 이상의 기존 자동차들의 설계 변경을 설명하기 위해 바운더리를 수정할 수 있다.

[0066] 예를 들어, 자동차 모델 A와 같은 특정 자동차 모델은 특정 라벨 (예를 들어, 클래스) 과 연관될 수 있다. 즉, 자동차 모델 A의 이미지들은 자동차 모델 A로 라벨링될 수 있다. 이 예에서, 특정 자동차 모델은 미등 설계에 대한 업데이트와 같이 설계에 대한 업데이트를 수신할 수 있다. 여전히, 제 2 트레이닝 세트는 업데이트된 자동차 설계에 대한 라벨링된 데이터를 포함하지 않을 수 있다. 따라서, 기존 클래스의 바운더리를 수정하여, 네트워크가 여전히 업데이트된 자동차 모델을 특정 자동차 모델로 라벨링하도록 업데이트된 설계를 고려할 수 있다. 즉, 이 예에서, 자동차 모델 A 클래스의 바운더리는, 업데이트된 미등을 갖는 자동차 모델 A를 잘못 분류하는 대신 업데이트된 미등 설계를 갖는 자동차 모델 A를 자동차 모델 A로 분류하도록 수정된다. 이 경우, 자동차 모델 A의 새로운 설계의 라벨링된 예들을 포함하는 제 3 트레이닝 세트 (D'') 가, 제 2 트레이닝 세트 (D') 와 함께, 자동차 모델 A의 양 설계들을 정확하게 분류하기 위해 제 2 뉴럴 네트워크 (F') 를 트레이닝하는데 사용될 수 있다.

[0067] 일 구성에서, 새로운 예들이 기존의 클래스들에 추가되는 경우, 라벨링되지 않은 데이터 (X) 및 새로운 라벨링된 데이터 (x'', y'') 를 고려할 때, 새로운 네트워크 (F) 는 세트들의 집합  $\{(X, F(X))\}$  및  $\{(x'', y'')\}$  상에서 트레이닝된다.

[0068] 부가적으로 또는 대안으로, 일 구성에서, 제 3 트레이닝 세트 (D'') 는 하나 이상의 새로운 클래스들을 제 2 뉴럴 네트워크 (F) 에 추가하도록 특정된다. 예를 들어, 제 2 트레이닝 세트 (D') 가 다른 자동차들에 대한 클래스들을 생성하도록 특정되면, 제 3 트레이닝 세트 (D'') 는 새로운 자동차 모델을 설명하기 위해 새로운 클래스를 추가하도록 특정될 수 있다.

[0069] 일 예로, 자동차 모델 A와 같은 특정 자동차 모델들은 특정 클래스와 연관될 수 있다. 또한, 이 예에서, 자동차 모델 B와 같은 새로운 자동차 모델이 제조업체에 의해 도입될 수 있다. 이 예에서, 제 2 트레이닝 세트 (D') 는 새로운 자동차 모델에 대한 라벨링된 데이터를 포함하지 않았다. 따라서, 제 2 트레이닝 세트 (D') 로부터의 트레이닝에 기초하여, 네트워크는 자동차 모델 B의 이미지들을 부정확하게 라벨링할 수 있다. 따라서, 이 예에서, 새로운 자동차 모델 B에 대한 새로운 클래스가 제 3 트레이닝 세트 (D'') 에 포함된 새로운 자동차 모델 B에 대한 라벨링된 데이터에 기초하여 제 2 뉴럴 네트워크 (F') 에 추가된다. 즉, 이 예에서, 자동차 모델 B에 대한 새로운 자동차 클래스가 생성되어 자동차 모델 B가 자동차 모델 A로 잘못 라벨링되지 않도록 한다.



- [0070] 일 구성에서, 원래의 트레이닝 데이터가 이용 가능하지 않은 때 프리-트레이닝된 네트워크에 하나 이상의 새로운 클래스들을 추가하는 경우, 라벨링되지 않은 데이터 ( $X$ ) 및 새로운 라벨링된 데이터  $\{(x'', y'')\}$  를 고려할 때, 새로운 네트워크 ( $F$ ) 는 세트들의 집합  $\{X, [F(X); 0 \cdot y'']\}$  및  $\{(x'', [F(X) \cdot 0; y''])\}$  상에서 트레이닝된다. 본 개시의 양태들에 따르면,  $y''$ 는 벡터 또는 스칼라일 수 있다.
- [0071] 일 구성에서, 제 2 데이터 ( $x_i'$ ) 는 클러스터되고 제 3 트레이닝 세트 ( $D''$ ) 의 제 3 데이터 ( $x''_i$ ) 와 비교된다. 더욱이, 비교 이후, 제 2 데이터 ( $x_i'$ ) 는 제 3 데이터 ( $x''_i$ ) 에 대한 유사성 또는 다양성에 기초하여 트레이닝하기 위해 선택된다.
- [0072] 도 6 은 본 개시의 양태들에 따른 러닝 트랜스퍼를 위한 흐름도 (600) 의 일 예를 나타낸다. 도 6에 도시된 바와 같이, 블록 (610) 에서, 제 1 네트워크는 제 1 트레이닝 세트로 트레이닝된다. 더욱이, 블록 (620) 에서, 네트워크 및/또는 사용자는 제 1 트레이닝 세트의 제 1 데이터 및 제 1 라벨들이 이용 가능한지를 결정할 수 있다. 제 1 데이터 및 제 1 라벨들이 이용 가능한 경우, 블록 (630) 에서, 제 1 네트워크는 원하는대로 사용 및/또는 수정될 수 있다.
- [0073] 제한된 배포 권한으로 인해 제 1 데이터와 제 1 라벨들이 이용 가능하지 않을 수 있다. 다른 예로서, 제 1 데이터 및 제 1 라벨들은 메모리 제한으로 인해 이용 가능하지 않을 수 있다. 이 예에서, 점진적 러닝을 수행 중인 시스템은 제 1 트레이닝 세트를 보유할 메모리 용량을 갖지 않을 수 있다. 추가적으로 또는 대안으로, 시스템은 제 1 트레이닝 세트를 다운로드하기 위해 특정된 시간 또는 연결성을 갖지 않을 수 있다. 따라서, 시스템 제약들로 인하여, 제 2 트레이닝 세트는 트레이닝 프로세스 동안 제 1 트레이닝된 네트워크로부터 생성될 수 있다. 본 개시의 양태들은, 제 1 데이터 및 제 1 라벨들이 이용 가능하지 않은 다른 이유들이 또한 고려될 수 있기 때문에, 배포 권리 또는 시스템 제약으로 인해 이용 불가능한 제 1 데이터 및 제 1 라벨들로 제한되지 않는다.
- [0074] 추가로, 제 1 라벨들 및/또는 제 1 데이터 (예를 들어, 제 1 트레이닝 세트) 가 이용 가능하지 않은 경우, 블록 (640) 에서, 제 2 라벨들은 제 1 네트워크를 통해 제 2 데이터에 대해 생성된다. 제 2 데이터는 제 1 데이터에 기초할 수도 있고 그렇지 않을 수도 있다. 제 2 라벨들을 생성한 이후, 블록 (650) 에서, 제 2 네트워크는 제 2 라벨들 및 제 2 데이터 (예를 들어, 제 2 트레이닝 세트) 상에서 트레이닝될 수 있다. 더욱이, 제 2 네트워크가 트레이닝된 이후에, 블록 (660) 에서, 제 3 데이터 및 제 3 라벨들 (예를 들어, 제 3 트레이닝 세트) 을 제 2 라벨들 및 제 2 데이터와 함께 사용하여, 새로운 클래스들을 제 2 네트워크에 추가하고 및/또는 제 2 네트워크의 기존 클래스들에 대한 바운더리들을 수정할 수 있다.
- [0075] 도 7a는 제 1 트레이닝 세트로 트레이닝된 원래의 네트워크 (710) 의 예를 도시한다. 도 7a에 도시된 바와 같이, 라벨들 ( $y$ -축) 은 이미지들 ( $x$ -축) 에 대해 특정된다. 더욱이, 원래의 네트워크 (710) 는 라벨링된 이미지들 상에서 트레이닝되며, 라벨링된 이미지들에 근사된다.
- [0076] 전술한 바와 같이, 제 1 라벨 및/또는 제 1 데이터 (예를 들어, 제 1 트레이닝 세트) 는 원래의 네트워크 (710) 를 트레이닝한 후에는 이용 가능하지 않을 수 있다. 도 7b는 제 1 트레이닝 세트가 없는 원래의 네트워크 (710) 의 예를 도시한다. 도 7b에 도시된 바와 같이, 비록 제 1 트레이닝 세트가 더 이상 이용 가능하지 않더라도, 원래의 네트워크는 제 1 트레이닝 세트로부터의 트레이닝에 기초하여 레이블링되지 않은 제 2 데이터를 여전히 분류하는데 사용될 수 있다. 여전히, 새로운 클래스들을 추가하고 및/또는 기존 클래스의 바운더리들을 수정하는 것이 바람직할 수 있다. 대부분의 경우, 새로운 클래스들을 추가하고 및/또는 기존 클래스들의 바운더리들을 수정하기 위해 네트워크를 트레이닝할 때, 네트워크는 제 1 트레이닝 세트와 같은 원래의 트레이닝 세트와 제 2 트레이닝 세트와 같은 추가 트레이닝 세트를 사용하여 트레이닝된다. 추가 트레이닝 세트의 라벨들 및 데이터는 새로운 클래스들을 추가하고 및/또는 기존 클래스들의 바운더리들을 수정하는데 사용될 수 있다.
- [0077] 따라서, 본 개시의 일 양태에 따르면, 새로운 네트워크는 새로운 클래스들을 추가하고 및/또는 기존 클래스들에 대한 바운더리들을 수정하기 위해 특정될 수 있다. 일 구성에서, 새로운 네트워크는 라벨들을 라벨링되지 않은 데이터로 원래의 네트워크를 통해 제공함으로써 생성된다. 도 7c에 도시된 바와 같이, 원래의 네트워크 (710) 는 라벨링되지 않은 데이터를 라벨링하는데 사용된다. 즉, 원래의 네트워크 (710) 는 라벨링되지 않은 제 2 데이터에 대한 제 2 라벨들을 생성하도록 특정될 수 있다. 일 구성에서, 제 2 데이터는 제 1 데이터와 실질적으로 유사하다.
- [0078] 더욱이, 도 7d에 도시된 바와 같이, 제 2 데이터 및 제 2 라벨들은 새로운 네트워크 (720) 를 트레이닝시키는 데

사용된다. 새로운 네트워크 (720) 가 제 2 데이터 및 제 2 라벨들로 트레이닝된 이후, 새로운 네트워크 (720) 는 새로운 클래스들을 추가하고 및/또는 기존 클래스들의 바운더리들을 수정하기 위해 제 3 라벨들 및 제 3 데이터로 보강된 제 2 데이터 및 제 2 라벨들 상에서 더욱 트레이닝된다.

[0079] 도 7e는 원래의 네트워크 (710) 와 새로운 네트워크 (720) 의 비교를 도시한다. 도 7e에 도시된 바와 같이, 새로운 네트워크 (720) 는 원래의 네트워크 (710) 와 유사하다.

[0080] 도 8은 본 개시의 일 양태에 따른 러닝 트랜스퍼를 위한 방법 (800) 을 도시한다. 블록 (802) 에서, 제 2 데이터가 수신된다. 일 구성에서, 제 2 데이터는 라벨링되지 않는다. 추가로, 블록 (804) 에서, 제 1 네트워크는 제 2 데이터에 대한 제 2 라벨들을 생성한다. 일 구성에서, 제 1 네트워크는 제 1 데이터에 대한 제 1 라벨들 상에서 미리 트레이닝되어 있다. 더욱이, 제 2 라벨들은 제 2 네트워크를 트레이닝하기 위해 생성된다. 제 1 네트워크 및 제 2 네트워크는 동일한 디바이스 상에서 정의될 수 있거나 또는 상이한 디바이스들 상에서 정의될 수 있음에 유의해야 한다. 제 2 데이터는 제 1 데이터와 실질적으로 유사할 수 있다. 다른 예에서, 제 2 데이터는 제 1 데이터와 동일하지만, 제 1 라벨들은 이용 가능하거나 이용 가능하지 않을 수 있다.

[0081] 도 9는 본 개시의 일 양태에 따른 러닝 트랜스퍼를 위한 방법 (900) 을 도시한다. 블록 (902) 에서, 제 2 네트워크는 제 2 데이터를 사용하여 제 1 네트워크에 의해 생성된 제 2 라벨들을 수신한다. 일 구성에서, 제 2 데이터는 제 1 네트워크에 의한 라벨링 이전에 라벨링되지 않는다. 더욱이, 일 구성에서, 제 1 네트워크는 제 1 라벨들 및 제 1 데이터 상에서 미리 트레이닝된다. 또한, 블록 (904) 에서, 제 2 네트워크는 제 2 라벨들 및 제 2 데이터 상에서 트레이닝된다. 제 1 네트워크 및 제 2 네트워크는 동일한 디바이스 상에서 정의될 수 있거나 또는 상이한 디바이스들 상에서 정의될 수 있음에 유의해야 한다. 제 2 데이터는 제 1 데이터와 실질적으로 유사할 수 있다. 다른 예에서, 제 2 데이터는 제 1 데이터와 동일하지만, 제 1 라벨들은 이용 가능하거나 이용 가능하지 않을 수 있다.

[0082] 상술된 방법들의 여러 동작들은 대응하는 기능들을 수행할 수 있는 임의의 적합한 수단에 의해 수행될 수도 있다. 그 수단은 회로, 주문형 반도체 (ASIC), 또는 프로세서를 포함하지만, 이들에 제한되지 않는 여러 하드웨어 및/또는 소프트웨어 컴포넌트(들) 및/또는 모듈(들)을 포함할 수도 있다. 일반적으로, 도면들에 도시된 동작들이 존재하는 경우, 이들 동작들은 유사한 넘버링을 갖는 대응하는 카운터파트 수단-플러스-기능 컴포넌트들을 가질 수도 있다.

[0083] 본원에서 이용되는 바와 같이, 용어 "결정하기" 는 매우 다양한 액션들을 망라한다. 예를 들어, "결정하기" 는 산출하기, 연산하기, 프로세싱하기, 도출하기, 조사하기, 검색하기 (예를 들어, 테이블, 데이터베이스, 또는 다른 데이터 구조에서 검색하기), 확인하기 등을 포함할 수도 있다. 또한, "결정하기" 는 수신하기 (예를 들어, 정보 수신하기), 액세스하기 (예를 들어, 메모리 내의 데이터에 액세스하기) 등을 포함할 수도 있다. 또한, "결정하기" 는 해결하기, 선택하기, 고르기, 설정하기 등을 포함할 수도 있다.

[0084] 본원에서 이용되는 바와 같이, 아이템들의 리스트 "중 적어도 하나" 를 지칭하는 구절은 단일 구성부를 포함하여, 이러한 아이템들의 임의의 조합을 지칭한다. 예로서, "a, b, 또는 c 중의 적어도 하나" 는 a, b, c, a-b, a-c, b-c, 및 a-b-c 를 포함하고자 한다.

[0085] 본원 개시물과 연계하여 설명된 다양한 예증적인 논리 블록들, 모듈들, 및 회로들은 본원에서 개시된 기능들을 수행하도록 디자인된 범용 프로세서, 디지털 신호 프로세서 (DSP), 주문형 반도체 (ASIC), 필드 프로그램가능한 게이트 어레이 (FPGA) 또는 다른 프로그램가능한 로직 디바이스 (PLD), 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 컴포넌트들, 또는 이들의 임의의 조합에 의해 구현되거나 수행될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있으나, 대안으로, 프로세서는 임의의 상업적으로 이용가능한 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한 컴퓨팅 디바이스들의 조합, 예를 들어, DSP 와 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 연계한 하나 이상의 마이크로프로세서들, 또는 임의의 다른 그러한 구성으로 구현될 수도 있다.

[0086] 본 개시물과 연계하여 설명된 방법의 단계들 또는 알고리즘은 하드웨어에서, 프로세서에 의해 실행되는 소프트웨어 모듈에서, 또는 이들 양자의 조합에서 직접적으로 구현될 수도 있다. 소프트웨어 모듈은 공지된 임의의 형태의 저장 매체 내에 있을 수도 있다. 이용될 수도 저장 매체들의 일부 예들은, 랜덤 액세스 메모리 (random access memory; RAM), 판독 전용 메모리 (read only memory; ROM), 플래시 메모리, 소거가능한 프로그램가능 판독 전용 메모리 (erasable programmable read-only memory; EPROM), 전기적으로 소거가능한 프로그램

가능 판독 전용 메모리 (electrically erasable programmable read-only memory; EEPROM), 레지스터들, 하드 디스크, 이동식 디스크, CD-ROM 등을 포함한다. 소프트웨어 모듈은 단일 명령 또는 많은 명령들을 포함할 수도 있고, 상이한 프로그램들 사이에서 여러 상이한 코드 세그먼트들에 걸쳐, 그리고 다수의 저장 매체들에 걸쳐 분포될 수도 있다. 저장 매체는 프로세서에 연결되어, 프로세서가 저장 매체로부터 정보를 판독하거나 저장 매체에 정보를 기록할 수 있다. 대안에서, 저장 매체는 프로세서에 통합될 수도 있다.

[0087] 본원에 개시된 방법들은 설명된 방법을 달성하기 위한 하나 이상의 단계들 또는 액션들을 포함한다. 방법 단계들 및/또는 액션들은 청구항들의 범위를 벗어나지 않으면서 서로 상호 교환될 수도 있다. 다시 말해, 단계들 또는 액션들에 대한 특정 순서가 명시되지 않는 한, 특정 단계들 및/또는 액션들의 순서 및/또는 이용은 청구항들의 범위로부터 벗어남이 없이 수정될 수도 있다.

[0088] 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 하드웨어에서 구현된다면, 일 예시적인 하드웨어 구성은 디바이스에서의 프로세싱 시스템을 포함할 수도 있다. 프로세싱 시스템은 버스 아키텍처로 구현될 수도 있다. 버스는 프로세싱 시스템 및 전체 설계 제약들의 특정 애플리케이션들에 따라 임의의 개수의 상호연결하는 버스들 및 브리지들을 포함할 수도 있다. 버스는 프로세서, 머신-판독가능 매체들, 및 버스 인터페이스를 포함하여 다양한 회로들을 함께 링크할 수도 있다. 버스 인터페이스는 다른 것들 중에서 네트워크 어댑터를 버스를 통해 프로세싱 시스템에 연결하는데 이용될 수도 있다. 네트워크 어댑터는 신호 프로세싱 기능들을 구현하는데 이용될 수도 있다. 소정의 양태들에서, 사용자 인터페이스 (예를 들어, 키보드, 디스플레이, 마우스, 조이스틱 등) 가 또한 버스에 연결될 수도 있다. 버스는 또한 다양한 다른 회로들, 예컨대, 타이밍 소스들, 주변기기들, 전압 조절기들, 전력 관리 회로들 등을 링크할 수도 있으며, 이는 공지되어 있으므로, 더 이상 설명되지 않을 것이다.

[0089] 프로세서는 컴퓨터 판독가능 매체 상에 저장된 소프트웨어의 실행을 포함하여 버스 및 범용 프로세싱을 관리하는 역할을 할 수도 있다. 프로세서는 하나 이상의 범용 및/또는 특수-목적용 프로세서들로 구현될 수도 있다. 예들은 마이크로프로세서들, 마이크로제어기들, DSP 제어기들, 및 소프트웨어를 실행할 수 있는 다른 회로부를 포함한다. 소프트웨어는 소프트웨어, 펌웨어, 미들웨어, 마이크로코드, 하드웨어 서술 언어, 또는 다른 것으로 지칭되더라도, 명령들, 데이터, 또는 이들의 임의의 조합을 의미하는 것으로 광범위하게 해석될 수 있다. 머신-판독가능 매체들은, 예로서, 랜덤 액세스 메모리 (RAM), 플래시 메모리, 판독 전용 메모리 (ROM), 프로그램가능한 판독 전용 메모리 (PROM), 소거가능한 프로그램가능 판독 전용 메모리 (EPROM), 전기적으로 소거가능한 프로그램가능 판독 전용 메모리 (EEPROM), 레지스터들, 자기 디스크들, 광학 디스크들, 하드 드라이브들, 또는 임의의 다른 적합한 저장 매체, 또는 이들의 임의의 조합을 포함할 수도 있다. 머신-판독가능 매체들은 컴퓨터-프로그램 제품으로 구체화될 수도 있다. 컴퓨터-프로그램 제품은 패키징 재료들을 포함할 수도 있다.

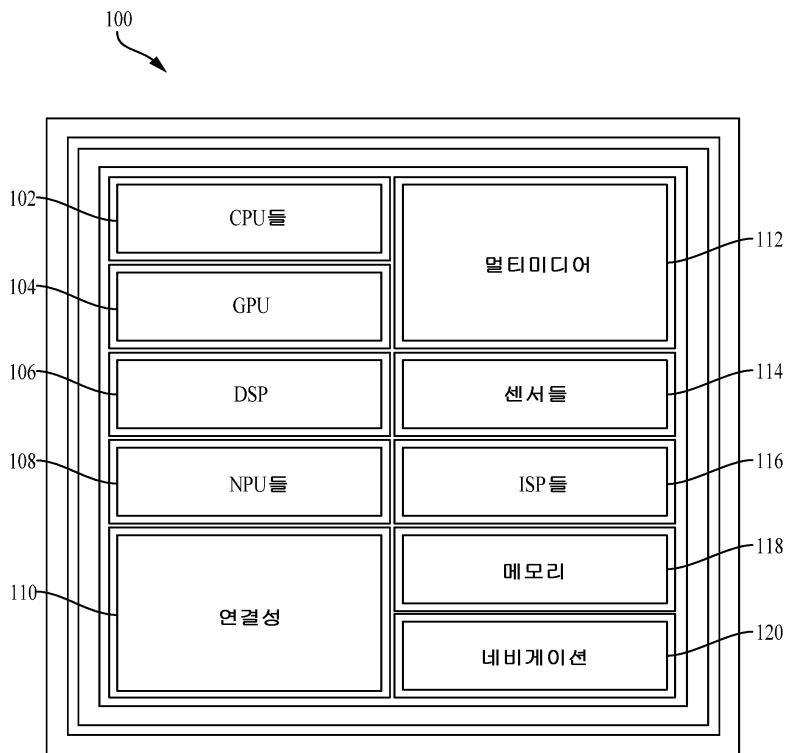
[0090] 하드웨어 구현에서, 머신-판독가능 매체들은 프로세서와 별개인 프로세싱 시스템의 일부일 수도 있다. 그러나, 머신-판독가능 매체들, 또는 이의 임의의 부분은 프로세싱 시스템의 외부에 있을 수도 있음을 당업자들은 쉽게 이해할 것이다. 예로서, 머신-판독가능 매체들은 송신 라인, 데이터에 의해 변조된 반송파, 및/또는 디바이스와 별도인 컴퓨터 제품 포함할 수도 있으며, 이 모두는 버스 인터페이스를 통해 프로세서에 의해 액세스가능하게 될 수도 있다. 대안으로, 또는 이에 더해, 머신-판독가능 매체들, 또는 이들의 임의의 부분은 프로세서에 통합될 수도 있으며, 그러한 경우에는 캐시 및/또는 범용 레지스터 파일들과 함께 있을 수도 있다. 논의된 다양한 컴포넌트들이 로컬 컴포넌트와 같이 특정 위치를 갖는 것으로 설명되었으나, 그것들은 또한 소정의 컴포넌트들이 분산 컴퓨팅 시스템의 일부로서 구성되는 것과 같이 다양한 방식으로 구성될 수도 있다.

[0091] 프로세싱 시스템은 프로세서 기능성을 제공하는 하나 이상의 마이크로프로세서들 및 적어도 일부분의 머신-판독가능 매체들을 제공하는 외부 메모리로 구현될 수도 있으며, 모두 외부 버스 아키텍처를 통해 다른 지원하는 회로부와 함께 링크된다. 대안으로, 프로세싱 시스템은 뉴런 모델들 및 본원에서 설명된 뉴런 시스템들의 모델들을 구현하기 위한 하나 이상의 뉴로모픽 프로세서들을 포함할 수도 있다. 다른 대안으로서, 프로세싱 시스템은 프로세서를 갖는 주문형 반도체 (ASIC), 버스 인터페이스, 사용자 인터페이스, 지원 회로부, 및 단일 칩 내에 통합되는 적어도 일부분의 머신-판독가능 매체들로, 또는 하나 이상의 필드 프로그램가능 게이트 어레이 (FPGA) 들, 프로그램가능 로직 디바이스 (PLD) 들, 제어기들, 상태 머신들, 게이트 로직, 이상 하드웨어 컴포넌트들, 또는 임의의 다른 적합한 회로부, 또는 본 개시물을 통해 설명된 다양한 기능성을 수행할 수 있는 회로들의 임의의 조합으로 구현될 수도 있다. 특정 응용 및 전체 시스템에 부과되는 전체 설계 제약들에 따라 본 개시물에 걸쳐 제시된 설명된 기능성을 가장 잘 구현하기 위한 방법을 당업자들은 인지할 것이다.

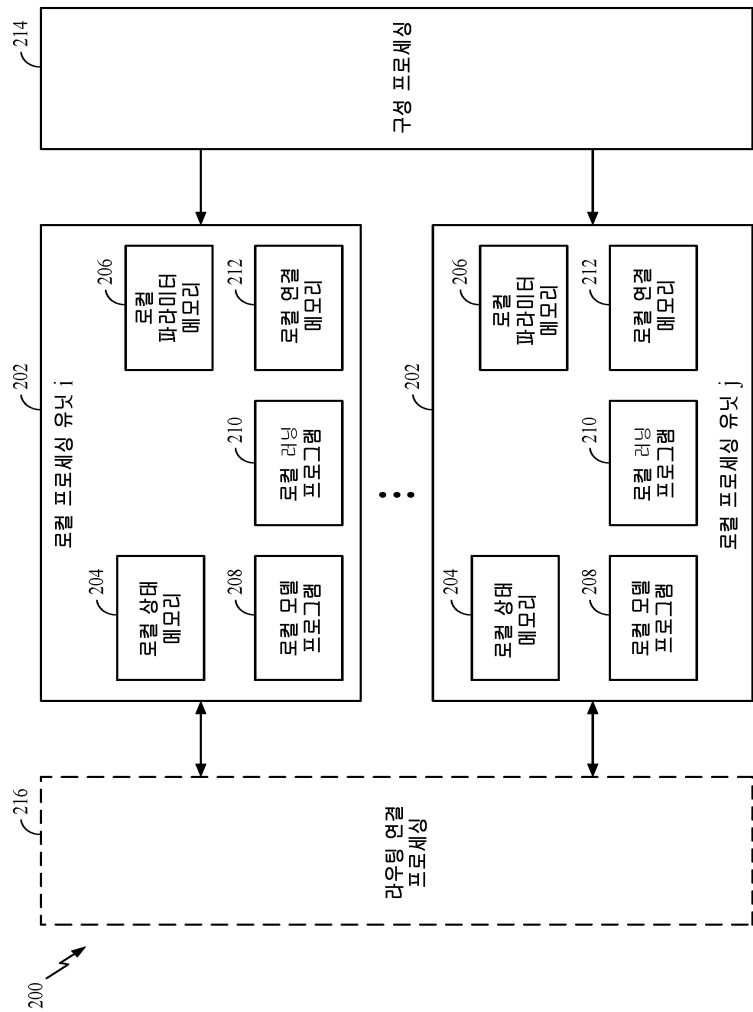
- [0092] 머신-판독가능 매체들은 다수의 소프트웨어 모듈들을 포함할 수도 있다. 소프트웨어 모듈들은, 프로세서에 의해 실행되는 경우, 프로세싱 시스템으로 하여금 다양한 기능들을 수행하게 하는 명령들을 포함한다. 소프트웨어 모듈들은 송신 모듈 및 수신 모듈을 포함할 수도 있다. 각각의 소프트웨어 모듈은 단일 저장 디바이스에 있을 수도 있거나 다수의 저장 디바이스들에 걸쳐 분산될 수도 있다. 예로서, 소프트웨어 모듈은 트리거링 이벤트가 발생하는 경우 하드웨어 드라이브로부터 RAM 으로 로딩될 수도 있다. 소프트웨어 모듈의 실행 중에, 프로세서는 액세스 속도를 증가시키기 위해 명령들의 일부를 캐시 내로 로딩할 수도 있다. 하나 이상의 캐시 라인들은 그러면 프로세서에 의한 실행을 위해 범용 레지스터 파일 내로 로딩될 수도 있다. 하기에서 소프트웨어 모듈의 기능성을 언급하는 경우, 그러한 기능성은 해당 소프트웨어 모듈로부터 명령들을 실행하는 경우 프로세서에 의해 구현된다는 것이 이해될 것이다. 또한, 본 개시의 양태들은 그러한 양태들을 구현하는 프로세서, 컴퓨터, 머신, 또는 다른 시스템의 기능에 대한 개선들을 야기한다.
- [0093] 소프트웨어로 구현된다면, 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 전송될 수도 있다. 컴퓨터-판독가능 매체들은 한 장소에서 다른 장소로 컴퓨터 프로그램의 전송을 가능하게 하는 임의의 매체를 포함하는 통신 매체들 및 컴퓨터 저장 매체들 양자 모두를 포함한다. 저장 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수도 있다. 비제한적인 예로서, 이러한 컴퓨터-판독가능 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기 스토리지 디바이스들, 또는 요구되는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 이송 또는 저장하기 위해 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 연결부는 컴퓨터-판독가능 매체라고 적절히 칭해진다. 예를 들어, 소프트웨어가 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (DSL), 또는 적외선 (IR), 무선, 및 마이크로파와 같은 무선 기술들을 사용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 전송된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 본원에서 사용된 디스크 (disk) 와 디스크 (disc) 는, 콤팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피디스크 및 블루레이<sup>®</sup> 디스크를 포함하며, 여기서 디스크 (disk) 는 통상 자기적으로 데이터를 재생하고, 디스크 (disc) 는 레이저를 이용하여 광학적으로 데이터를 재생한다. 따라서, 일부 양태들에서, 컴퓨터-판독가능 매체들은 비일시적 컴퓨터-판독가능 매체들 (예를 들어, 타입의 매체들) 을 포함할 수도 있다. 또한, 다른 양태들에 있어서, 컴퓨터-판독가능 매체들은 일시적 컴퓨터-판독가능 매체들 (예를 들어, 신호) 을 포함할 수도 있다. 위의 조합들도 컴퓨터-판독가능 매체들의 범위 내에 포함되어야 한다.
- [0094] 따라서, 소정의 양태들은 본원에 제시된 동작들을 수행하는 컴퓨터 프로그램 제품을 포함할 수도 있다. 예를 들어, 이러한 컴퓨터 프로그램 제품은 저장된 (및/또는 인코딩된) 명령들을 갖는 컴퓨터 판독가능 매체를 포함할 수도 있으며, 명령들은 본원에 설명된 동작들을 수행하기 위해 하나 이상의 프로세서들에 의해 실행가능할 수도 있다. 소정의 양태들에 있어서, 컴퓨터 프로그램 제품은 패키징 재료를 포함할 수도 있다.
- [0095] 또한, 본원에 설명된 방법들 및 기법들을 수행하는 모듈들 및/또는 다른 적절한 수단은 다운로드될 수도 있고/있거나, 그렇지 않으면 가능한 적용가능한 사용자 단말 및/또는 기지국에 의해 획득될 수도 있다. 예를 들어, 본원에서 설명된 방법들을 수행하기 위한 수단의 전송을 용이하게 하기 위한 서버에 디바이스가 연결될 수도 있다. 대안으로, 본원에 설명된 다양한 방법들이 저장 수단 (예를 들어, RAM, ROM, 물리적 콤팩트 디스크 (CD) 나 플로피 디스크와 같은 물리적 저장 매체 등) 을 통해 제공될 수도 있어, 사용자 단말 및/또는 기지국은 디바이스에 연결할 시에 또는 디바이스에 저장 수단을 제공할 시에 다양한 방법들을 획득할 수 있다. 또한, 본원에서 설명된 방법들 및 기술들을 디바이스에 제공하기 위해 임의의 다른 적절한 기술들이 활용될 수 있다.
- [0096] 청구항들은 위에서 예시된 정확한 구성 및 컴포넌트들로 제한되지 않는 것으로 이해되어야 한다. 청구항의 범위를 벗어나지 않으면서, 본원에서 설명된 시스템들, 방법들, 및 장치들의 배치, 동작 및 세부사항들에서 다양한 수정예들, 변경예들, 및 변형예들이 행해질 수도 있다.

도면

도면1

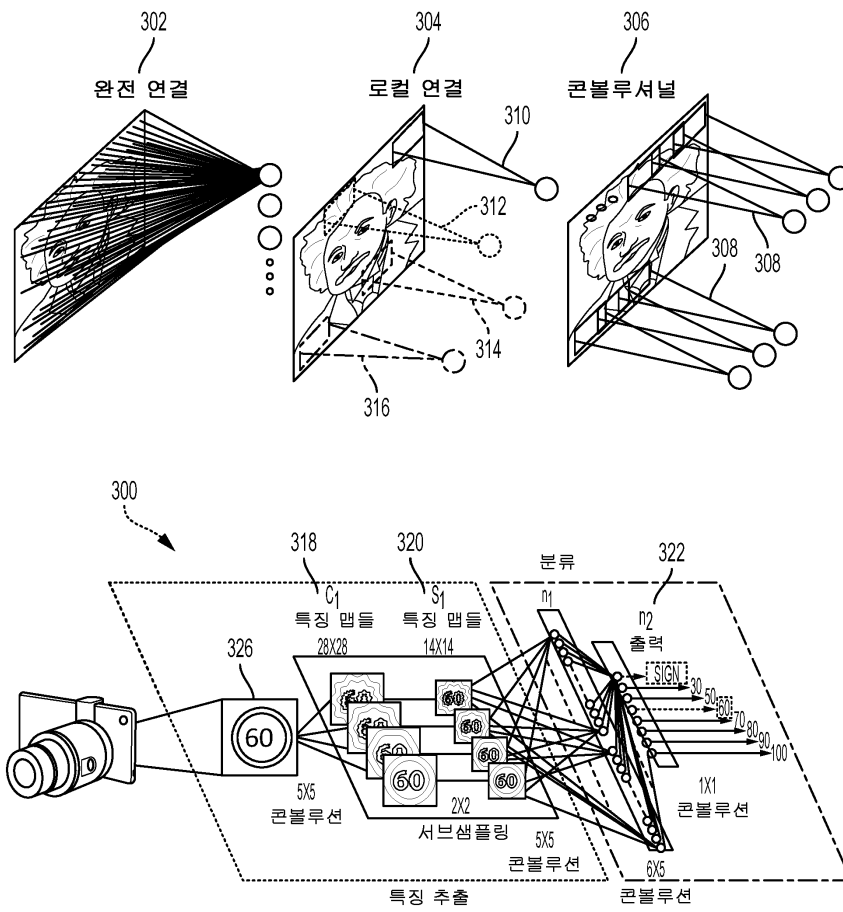


도면2

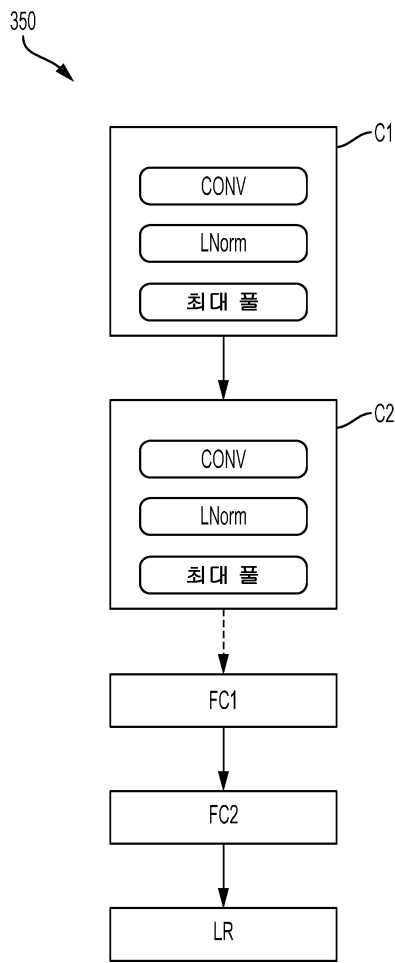




도면 3a

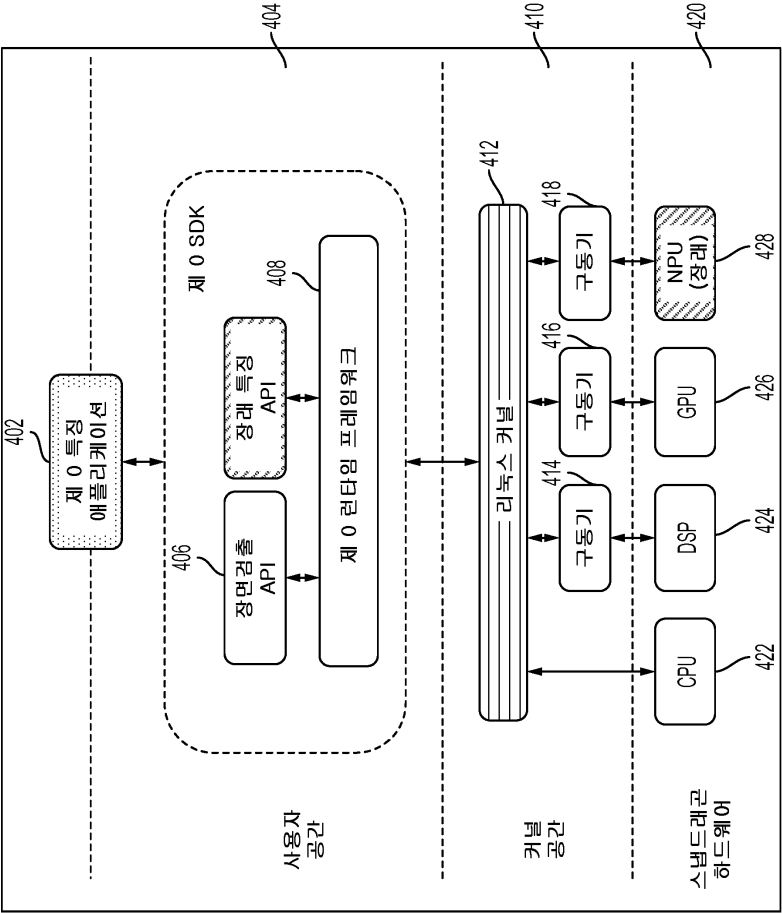


도면 3b

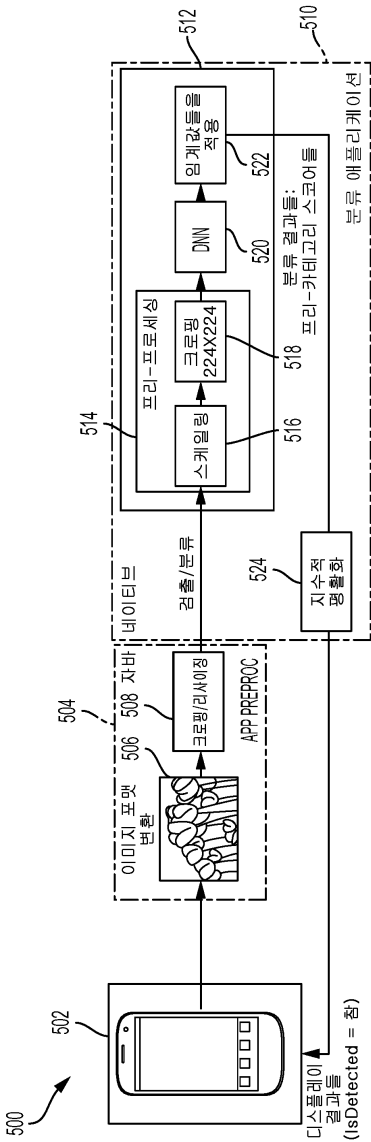




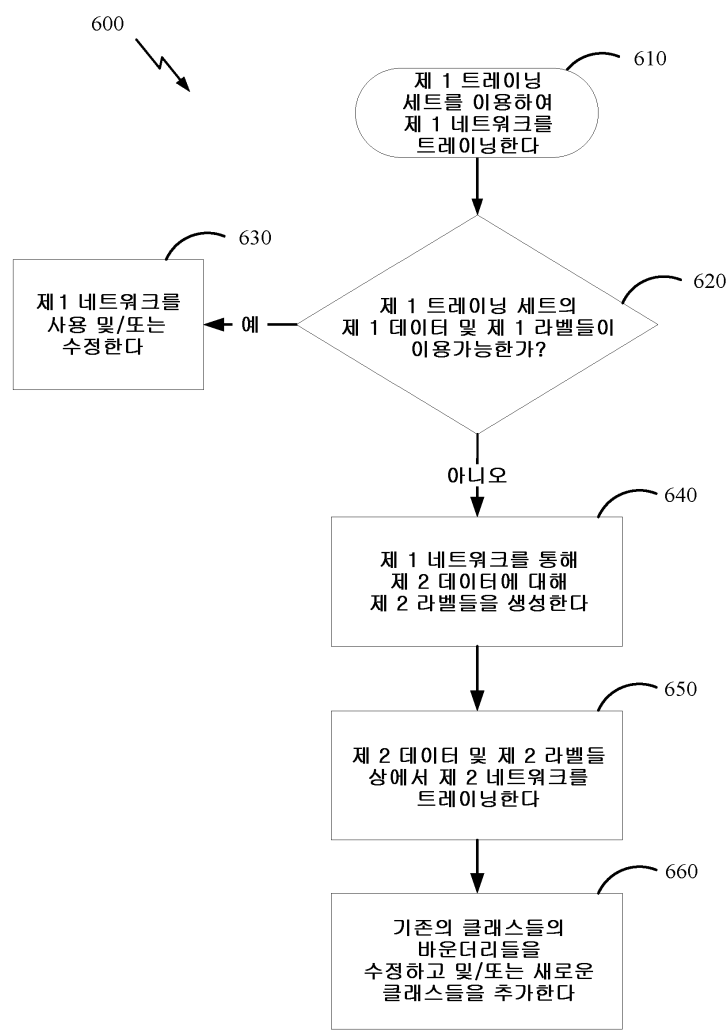
도면4



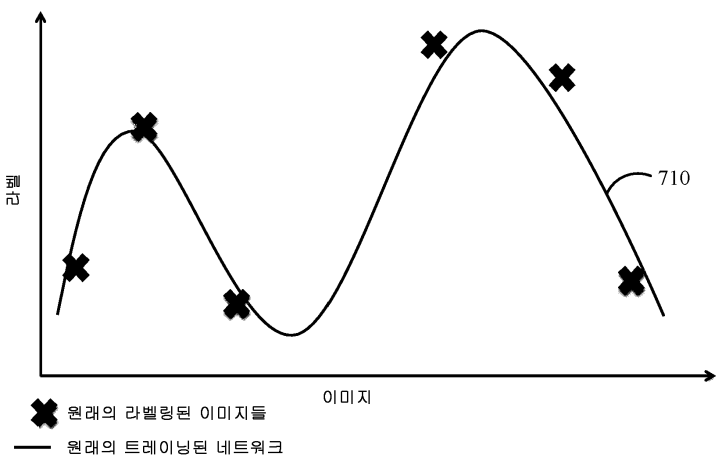
도면5



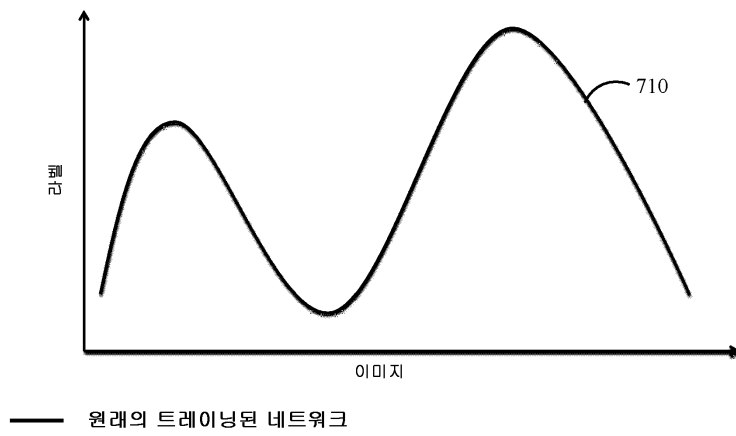
도면6



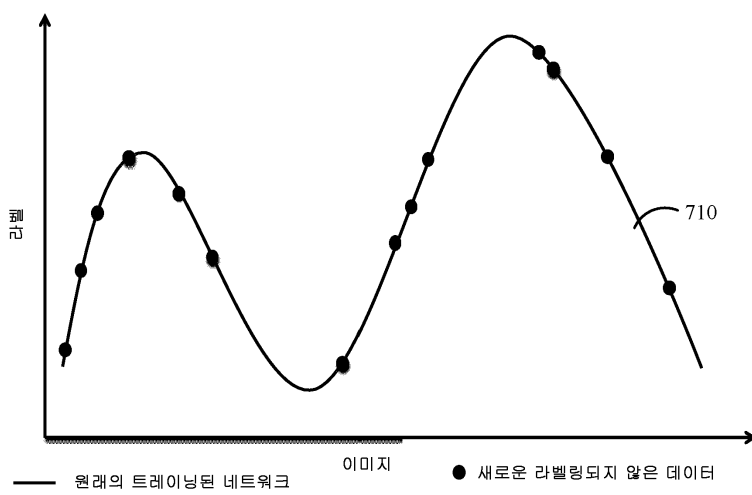
도면7a



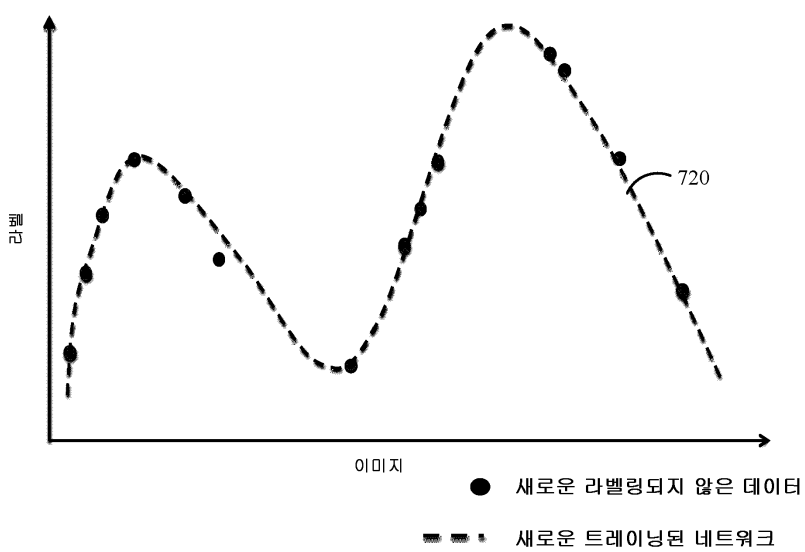
도면7b



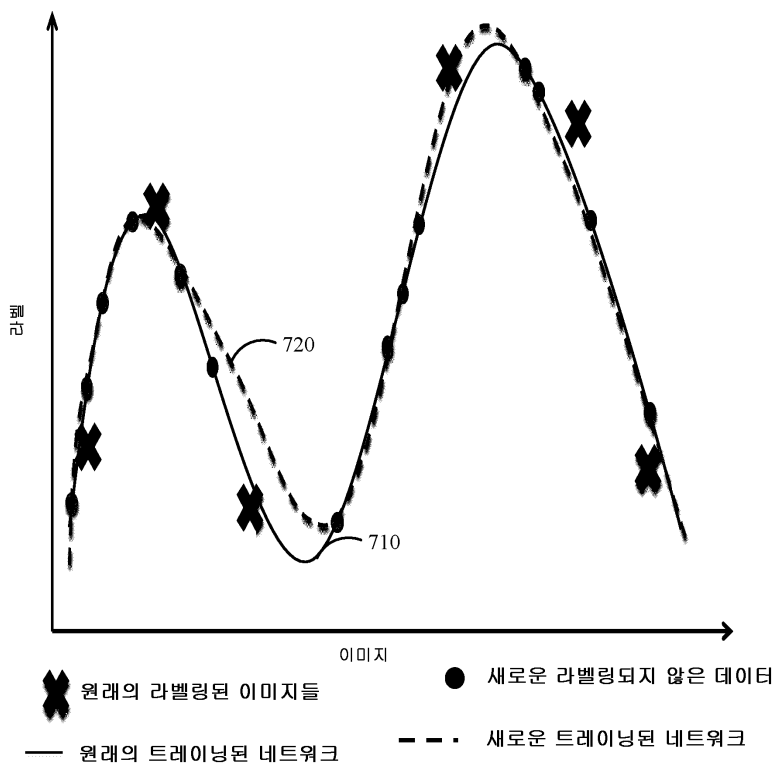
도면7c



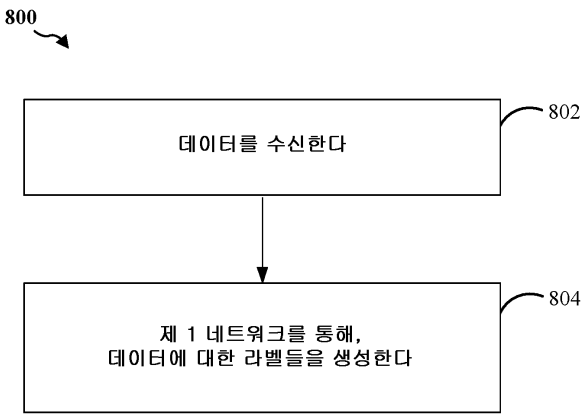
도면7d



도면7e



도면8



도면9

