



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2021년03월04일

(11) 등록번호 10-2223296

(24) 등록일자 2021년02월26일

(51) 국제특허분류(Int. Cl.)  
 G06K 9/62 (2006.01) G06K 9/46 (2006.01)  
 G06K 9/66 (2006.01)

(52) CPC특허분류  
 G06K 9/6267 (2013.01)  
 G06K 9/4628 (2013.01)

(21) 출원번호 10-2018-7029132

(22) 출원일자(국제) 2017년03월13일

심사청구일자 2020년03월13일

(85) 번역문제출일자 2018년10월08일

(65) 공개번호 10-2018-0117704

(43) 공개일자 2018년10월29일

(86) 국제출원번호 PCT/US2017/022206

(87) 국제공개번호 WO 2017/156547

국제공개일자 2017년09월14일

(30) 우선권주장

62/307,071 2016년03월11일 미국(US)

(56) 선행기술조사문헌

US20150036922 A1\*

US20150051840 A1\*

\*는 심사관에 의하여 인용된 문헌

(73) 특허권자

매직 립, 인코포레이티드

미국 플로리다 플랜타타운 웨스트 선라이즈 블러  
 마드 7500 (우: 33322)

(72) 발명자

라비노비치, 앤드류

미국 94131 캘리포니아 샌프란시스코 마뷰 웨이  
 157

바드리나라야난, 비제이

미국 94040 캘리포니아 마운틴 뷰 에스쿠엘라 애  
 비뉴 222 아파트먼트 172

(뒷면에 계속)

(74) 대리인

특허법인 남앤남

전체 청구항 수 : 총 33 항

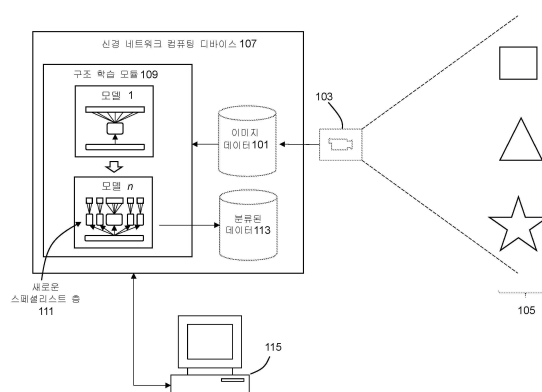
심사관 : 강현일

(54) 발명의 명칭 **콘볼루셔널 신경 네트워크들에서의 구조 학습**

### (57) 요약

본 개시내용은 네트워크가 해결하고자 하는 문제/데이터의 상관관계들(correlations)을 이용함으로써 신경 네트워크들의 구조 학습을 구현하기 위한 개선된 접근법을 제공한다. 하단 콘볼루셔널 층들로부터 완전히 연결된 층들까지 내내 정보 이득의 병목들을 발견하는 그리디 접근법이 설명된다. 단순히 아키텍처를 더 깊게 하기 보다는, 부가적인 컴퓨테이션 및 커패시터스는 그것이 필요할 곳에만 추가된다.

**대표도** - 도1



(52) CPC특허분류

**G06K 9/66** (2013.01)

(72) 발명자

**데톤, 다니엘**

미국 94107 캘리포니아 샌프란시스코 위스콘신 스트리트 955 유니트 4

**라젠드란, 스리비그네쉬**

미국 94043 캘리포니아 마운틴 뷰 이스트 미들필드 로드 100 아파트먼트 5씨

**리, 더글라스, 버트럼**

미국 94063 캘리포니아 레드우드 시티 베어 아일랜드 로드 630 유니트 105

**말리지에비크, 토마즈, 제이.**

미국 94041 캘리포니아 마운틴 뷰 뷰 스트리트 315

## 명세서

### 청구범위

#### 청구항 1

프로세서로 구현되는 방법으로서,

신경 네트워크를 생성하는 단계;

상기 신경 네트워크로부터 출력을 생성하는 단계;

상기 신경 네트워크로부터 저 성능 층을 식별하는 단계 - 상기 저 성능 층은 상기 신경 네트워크 내의 다른 층의 성능 보다 상대적으로 낮은 성능을 가짐 -;

상기 저 성능 층에 새로운 스페셜리스트 층(new specialist layer)을 삽입하는 단계; 및

상기 신경 네트워크의 최상부가 도달될 때까지 상기 식별하는 단계의 수행 및 상기 삽입하는 단계의 수행을 반복하는 단계를 포함하는,

프로세서로 구현되는 방법.

#### 청구항 2

제1항에 있어서,

업데이트된 모델을 획득하기 위하여 상기 신경 네트워크에 대한 모델을 업데이트하는 단계를 더 포함하고,

상기 업데이트된 모델은 상기 새로운 스페셜리스트 층 및 적어도 하나의 제너럴리스트(generalist) 층을 포함하는,

프로세서로 구현되는 방법.

#### 청구항 3

제1항에 있어서,

상기 새로운 스페셜리스트 층은 다른 스페셜리스트 층에 의해 처리되는 서브도메인과 구별되는 특정 데이터 서브도메인을 처리하도록 구성되는,

프로세서로 구현되는 방법.

#### 청구항 4

제1항에 있어서,

복수의 손실 층들이 상기 신경 네트워크에 추가되는,

프로세서로 구현되는 방법.

#### 청구항 5

제4항에 있어서,

상기 손실 층들 중 하나에서 예측들을 생성하는 단계; 및

상기 예측들을 텐서(tensor) T를 형성하는 하나 또는 그 초과인 혼동 행렬들로 변환하는 단계를 더 포함하는,

프로세서로 구현되는 방법.

#### 청구항 6

제5항에 있어서,

T의 구조는 깊이 및 폭 둘 모두의 관점에서 상기 신경 네트워크의 기존 구조를 수정 및 증강시키기 위해 분석되는,

프로세서로 구현되는 방법.

#### 청구항 7

제1항에 있어서,

상기 신경 네트워크는 수직 분할 및 수평 분할 둘 모두가 수행되는,

프로세서로 구현되는 방법.

#### 청구항 8

제7항에 있어서,

K-웨이 분기(K-way Bifurcation)가 상기 수평 분할을 구현하기 위해 수행되는,

프로세서로 구현되는 방법.

#### 청구항 9

제1항에 있어서,

상기 신경 네트워크의 각각의 층은 독립적으로 어드레싱되고, 상기 신경 네트워크의 소정의 층은 트레이닝 손실(training loss)에 대한 최상의 개선을 제공하는 상기 소정의 층을 분할하기 위해 그리디 선택(greedy choice)을 수행함으로써 분할이 수행되는,

프로세서로 구현되는 방법.

#### 청구항 10

제1항에 있어서,

올-오아-나씽 하이웨이 네트워크(all-or-nothing highway network)가 제거될 신경 네트워크 내의 층들을 식별하기 위해 사용되는,

프로세서로 구현되는 방법.

#### 청구항 11

제1항에 있어서,

상기 신경 네트워크는 가상 현실 또는 증강 현실 시스템을 위해 캡처된 이미지들을 분류하는 데 사용되는,

프로세서로 구현되는 방법.

#### 청구항 12

시스템으로서,

프로세서;

프로그래밍 가능 코드를 보유하기 위한 메모리를 포함하고,

상기 프로그래밍 가능 코드는,

신경 네트워크를 생성하는 단계;

상기 신경 네트워크로부터 출력을 생성하는 단계;

상기 신경 네트워크로부터 저 성능 층을 식별하는 단계 - 상기 저 성능 층은 상기 신경 네트워크 내의 다른 층의 성능 보다 상대적으로 낮은 성능을 가짐 -;

상기 저 성능 층에 새로운 스페셜리스트 층을 삽입하는 단계; 및

상기 신경 네트워크의 최상부가 도달될 때까지 상기 식별하는 단계의 수행 및 상기 삽입하는 단계의 수행을 반복하는 단계  
를 위한 명령들을 포함하는,  
시스템.

#### 청구항 13

제12항에 있어서,  
상기 프로세서는,  
업데이트된 모델을 획득하기 위하여 상기 신경 네트워크에 대한 모델을 업데이트하도록 구성되고,  
상기 업데이트된 모델은 상기 새로운 스페셜리스트 층 및 적어도 하나의 제너럴리스트 층을 포함하는,  
시스템.

#### 청구항 14

제12항에 있어서,  
상기 새로운 스페셜리스트 층은 다른 스페셜리스트 층에 의해 처리되는 서브도메인과 구별되는 특정 데이터 서브도메인을 처리하도록 구성되는,  
시스템.

#### 청구항 15

제12항에 있어서,  
상기 신경 네트워크는 복수의 손실 층들을 포함하는,  
시스템.

#### 청구항 16

제15항에 있어서,  
상기 프로세서는,  
상기 손실 층들 중 하나에서 예측들을 생성하고,  
상기 예측들을 텐서 T를 형성하는 하나 또는 그 이상의 혼동 행렬들로 변환하도록 구성되는,  
시스템.

#### 청구항 17

제16항에 있어서,  
T의 구조는 깊이 및 폭 둘 모두의 관점에서 상기 신경 네트워크의 기존 구조를 수정 및 증강시키기 위해 분석되는,  
시스템.

#### 청구항 18

제12항에 있어서,  
상기 신경 네트워크는 수직 분할 및 수평 분할 둘 모두가 수행되도록 구성되는,  
시스템.

#### 청구항 19

제18항에 있어서,  
상기 수평 분할은 K-웨이 분기를 사용하여 구현되는,  
시스템.

#### 청구항 20

제12항에 있어서,  
상기 프로세서는,  
상기 신경 네트워크의 각각의 층을 독립적으로 어드레싱하고,  
상기 신경 네트워크의 소정의 층으로 하여금, 트레이닝 손실에 대한 최상의 개선을 제공하는 상기 소정의 층을 분할하기 위해 그리디 선택을 수행함으로써 분할이 수행되도록 하도록 구성되는,  
시스템.

#### 청구항 21

제12항에 있어서,  
상기 프로세서는,  
제거될 신경 네트워크 내의 층들을 식별하기 위해 올-오아-나뿔 하이웨이 네트워크를 사용하도록 구성되는,  
시스템.

#### 청구항 22

제12항에 있어서,  
상기 신경 네트워크는 가상 현실 또는 증강 현실 시스템을 위해 캡처된 이미지들을 분류하도록 구성되는,  
시스템.

#### 청구항 23

컴퓨터 판독 가능 저장 매체로서,  
상기 컴퓨터 판독 가능 저장 매체는, 프로세서에 의해 실행될 때 상기 프로세서로 하여금, 방법을 실행하게 하는 명령들의 시퀀스를 저장하고 있고,  
상기 방법은,

신경 네트워크를 생성하는 단계;

상기 신경 네트워크로부터 출력을 생성하는 단계;

상기 신경 네트워크로부터 저 성능 층을 식별하는 단계 - 상기 저 성능 층은 상기 신경 네트워크 내의 다른 층의 성능 보다 상대적으로 낮은 성능을 가짐 -;

상기 저 성능 층에 새로운 스페셜리스트 층을 삽입하는 단계; 및

상기 신경 네트워크의 최상부가 도달될 때까지 상기 식별하는 단계의 수행 및 상기 삽입하는 단계의 수행을 반복하는 단계

를 포함하는,

컴퓨터 판독 가능 저장 매체.

#### 청구항 24

제23항에 있어서,  
상기 방법은,

업데이트된 모델을 획득하기 위하여 상기 신경 네트워크에 대한 모델을 업데이트하는 단계를 더 포함하고,

상기 업데이트된 모델은 상기 새로운 스페셜리스트 층 및 적어도 하나의 제너럴리스트 층을 포함하는, 컴퓨터 판독 가능 저장 매체.

#### 청구항 25

제23항에 있어서,

상기 방법에서,

상기 새로운 스페셜리스트 층은 다른 스페셜리스트 층에 의해 처리되는 서브도메인과 구별되는 특정 데이터 서브도메인을 처리하도록 구성되는,

컴퓨터 판독 가능 저장 매체.

#### 청구항 26

제23항에 있어서,

상기 방법에서,

복수의 손실 층들은 상기 신경 네트워크에 추가되는,

컴퓨터 판독 가능 저장 매체.

#### 청구항 27

제26항에 있어서,

상기 방법은,

상기 손실 층들 중 하나에서 예측들을 생성하는 단계; 및

상기 예측들을 텐서 T를 형성하는 하나 또는 그 초과 of 혼동 행렬들로 변환하는 단계를 더 포함하는,

컴퓨터 판독 가능 저장 매체.

#### 청구항 28

제27항에 있어서,

T의 구조는 깊이 및 폭 둘 모두의 관점에서 상기 신경 네트워크의 기존 구조를 수정 및 증강시키기 위해 상기 방법에서 분석되는,

컴퓨터 판독 가능 저장 매체.

#### 청구항 29

제23항에 있어서,

상기 방법에서,

상기 신경 네트워크는 수직 분할 및 수평 분할 둘 모두가 수행되는,

컴퓨터 판독 가능 저장 매체.

#### 청구항 30

제29항에 있어서,

K-웨이 분기는 상기 수평 분할을 구현하기 위해 상기 방법에서 수행되는,

컴퓨터 판독 가능 저장 매체.

### 청구항 31

제23항에 있어서,

상기 방법에서 상기 신경 네트워크의 각각의 층은 독립적으로 어드레싱되고,

상기 신경 네트워크의 소정의 층은 트레이닝 손실에 대한 최상의 개선을 제공하는 상기 소정의 층을 분할하기 위해 그리디 선택을 수행함으로써 분할이 수행되는,

컴퓨터 판독 가능 저장 매체.

### 청구항 32

제23항에 있어서,

울-오아-나쌍 하이웨이 네트워크가 제거될 신경 네트워크 내의 층들을 식별하기 위해 상기 방법에서 사용되는,

컴퓨터 판독 가능 저장 매체.

### 청구항 33

제23항에 있어서,

상기 신경 네트워크는 가상 현실 또는 증강 현실 시스템을 위해 캡처된 이미지들을 분류하기 위해 상기 방법에서 사용되는,

컴퓨터 판독 가능 저장 매체.

## 발명의 설명

### 기술 분야

[0001] 본 발명은 컴퓨팅 네트워크들에 관한 것으로, 보다 상세하게는, 데이터로부터 계층적 표현들을 학습(learn)하도록 구성된 신경 네트워크들에 관한 것이다.

### 배경 기술

[0002] 신경 네트워크들은 복잡한 컴퓨테이셔널 문제들을 해결하는 데 사용될 수 있는 생물학적 뇌 프로세싱의 신경 구조들을 본 받아서 느슨하게 모델링된 컴퓨테이셔널 접근법들에 관한 것이다. 신경 네트워크들은 대개, 층들의 세트로서 구조화되며, 각각의 층은 다양한 기능들을 포함하는 상호연결된 노드들의 세트를 포함한다. 가중화된 연결들은 다양한 분석 동작들을 수행하기 위해 네트워크 내에서 프로세싱되는 기능들을 구현한다. 학습 프로세스들은 네트워크들 및 네트워크 내의 커넥터들에 대한 연관된 가중치들을 구성 및 수정하는 데 사용될 수 있다. 커넥터 가중치들을 수정함으로써, 이는, 향후 분석 결과들을 개선하기 위해 네트워크가 과거 분석으로부터 시간이 지남에 따라 학습하도록 허용한다.

[0003] 신경 네트워크들은 임의의 적절한 유형의 데이터 분석을 수행하기 위해 사용될 수 있지만, 패턴 분석 및 분류와 같은 복잡한 분석 태스크들에 적용되기에 특히 적합하다. 따라서, 이러한 기술들의 직접적인 적용은 예를 들어, 디지털 이미징 디바이스들에 의해 캡처된 이미지 데이터로부터 특정 객체들 및 객체 클래스들의 인식 및 분류와 같은 머신 비전 기능(machine vision function)들을 구현하는데 적합하다.

[0004] 당 업계에 알려져 있는 수많은 유형들의 신경 네트워크들이 있다. 딥 신경 네트워크(deep neural network)들은, 분석 기능들을 수행하기 위해 다수의 비선형 프로세싱 층들의 캐스케이드(cascade)를 구현하도록 딥 러닝 기술들이 적용되는 신경 네트워크의 유형이다. 딥 러닝 알고리즘들은 쉘로 러닝(shallow learning) 알고리즘들보다 많은 층들을 통해 그의 입력들을 변형시킨다. 각각의 층에서, 신호는 파라미터들이 트레이닝을 통해 학습되는 프로세싱 유닛, 이를테면, 인공 뉴런(artificial neuron)에 의해 변형된다.

[0005] 콘볼루션 신경 네트워크(convolutional neural network)는 네트워크의 연결 패턴이 생물학적 시각 피질 기능(biological visual cortex functioning)에 의해 영감을 받은(inspire) 신경 네트워크의 유형이다. 시각 필드들은 네트워크를 통해 구성되며, 여기서 입력 자극에 대한 개별 인공 뉴런의 응답은 콘볼루션 연산(convolution operation)에 의해 수학적으로 근사될 수 있다.



- [0006] [0006] 콘볼루션 딥 신경 네트워크들은 알려진 기술로 구현되었다. LeNet(LeCun 등(1998)), AlexNet(Krizhevsky 등(2012)), GoogLeNet(Szegedy 등(2015)), 및 VGGNet(Simonyan & Zisserman(2015))는 모두 상이한 유형의 딥 신경 네트워크들을 구현하는 ConvNet 아키텍처들의 예들이다. 이러한 모델들은 상당히 상이하다(예를 들어, 깊이, 폭 및 활성화 기능들이 상이함). 그러나 이러한 모델들은 하나의 핵심적인 면에서 모두 동일한데, 각각의 모델은 당면한 문제에 대한 설계자의 인사이트(architects' insight)들을 구현하는 손 설계 구조(hand designed structure)이다.
- [0007] [0007] 이러한 네트워크들은 Gabor 필터 또는 그의 일부 표현들과 유사한 저-레벨 피쳐들을 학습하는 콘볼루션 층으로 시작하는, 비교적 간단한 레시피(recipe)를 따른다. 추후의 층들은 객체 부분들(얼굴들, 자동차들 등의 부분들)과 같은 더 높은-레벨 피쳐들을 인코딩한다. 마지막으로, 최상부에, 클래스들에 걸친 확률 분포를 리턴하는 층이 있다. 이 접근법은 라벨 공간에서, 트레이닝된 네트워크에 의해 생성된 출력에 일부 구조를 제공하지만, 이 구조는 이러한 네트워크들이 설계 및 트레이닝될 때 거의 활용되지 않는다는 이슈가 있다.
- [0008] [0008] 확률적 그래픽 모델들에서 구조 학습이 제안되었으며, 여기서 딥 콘볼루션 네트워크들에서 구조 학습(structure learning)을 위한 종래의 알고리즘은 통상적으로 2개의 카테고리들, 즉 넷들(nets)을 더 작아지게 하는 카테고리 및 넷들을 더 양호하게 하는 카테고리 중 하나에 속한다. 하나의 제안된 접근법은 다루기 힘든 미리 트레이닝된 네트워크들을 취하여 이들을, 더 작은 메모리 풋프린트를 갖고 이에 따라 더 적은 컴퓨터에서 실행될 자원들을 필요로 하는 네트워크들로 밀어 넣는 것(squeezing)에 초점을 맞춘다. 이 기술 클래스는, 목표들이 교사를 모방하는 학생 네트워크를 생성하는 것인 "교사-학생(teacher-student)" 패러다임에 따른다. 이는, 그것이 오라클(Oracle) 아키텍처 및 그의 학습된 가중치들 둘 모두로 시작할 필요가 있다는 것을 의미한다(학생의 트레이닝만이 추후에 발생함). 초대형 데이터셋들 상에서 스페셜리스트들의 앙상블을 정제(distilling)할 때, 컴퓨터에서 실행적으로 비싼 앙상블 트레이닝 단계가 먼저 수행되어야 한다.
- [0009] [0009] Feng 등의 "Learning the Structure of Deep Convolutional Networks"은 딥 모델의 구조의 앙상블들을 자동으로 학습하는 기술의 예이다. 이 접근법은 인디언 뷔페 프로세스(Indian Buffet Process)를 사용하여 구조를 식별하기 위한 새로운 콘볼루션 신경 네트워크 모델을 제안하며, 여기서 구조가 결정된 후에 네트워크의 보다 간결한 표현을 생성하기 위해 프루닝(pruning)이 수행된다. 그러나 이 접근법의 하나의 단점은 층들의 수가 정적인 채로 유지된다는 것이며; 여기서 정적인 수의 층들 내의 알려진 개별 층들만이 구조 학습 프로세스를 통해 더 복잡해지도록 또는 덜 복잡해지도록 증강된다. 따라서, 이 접근법은 구조를 최적화하는데 필요할 수 있는 어떠한 새로운 층들도 식별할 수 없다.
- [0010] [0010] 따라서, 콘볼루션 신경 네트워크들에 대한 구조 학습을 구현하기 위한 개선된 접근법에 대한 필요성이 존재한다.

### 발명의 내용

- [0011] [0011] 본 발명의 일부 실시예들은 신경 네트워크들에 대한 구조 학습을 구현하기 위한 개선된 접근법들에 관한 것이다. 접근법은 네트워크로 시작하고, 라벨링된 데이터를 갖는 문제를 네트워크에 제공하고, 그 후 이 네트워크에 의해 생성된 출력의 구조를 검토한다. 그 후, 네트워크의 아키텍처는 특정 문제에 대한 더 양호한 솔루션을 획득하도록 수정된다. 전문가들이 몹시 복잡하고 도메인-특정 네트워크 아키텍처들을 생각해 내게 하기 보다는, 이 접근법은 데이터가 특정 태스크에 대해 사용될 네트워크의 아키텍처를 구동하도록 허용한다.
- [0012] [0012] 일부 실시예들에 따라, 신경 네트워크는, (a) 그 구조에서 정보 이득 병목(information gain bottleneck)을 식별하고, (b) 병목을 완화시키기 위해 예측들의 구조를 적용하고, 마지막으로, (c) 스페셜리스트 경로들의 깊이를 결정함으로써 개선될 수 있다.
- [0013] [0013] 일부 실시예들은, 네트워크가 해결하고자 하는 문제/데이터의 상관관계들(correlations)을 이용함으로써 신경 네트워크들의 구조 학습을 구현하며, 여기서 그리디 접근법(greedy approach)이 하단 콘볼루션 층들로부터 완전히 연결된 층들까지 내내 정보 이득의 병목들을 발견하도록 수행된다. 일부 실시예들에서, 네트워크가 초기 시점에 생성되고, 예를 들어, 이미지 인식/객체 분류 태스크들을 수행하기 위해 지정된 태스크에 적용될 때, 네트워크로부터 출력 세트가 생성된다. 다음으로, 네트워크 모델 내의 다양한 층들은 모델 내의 최저 성능 층을 식별하도록 분석된다. 그 후, 부가적인 구조들이 모델의 성능을 개선하기 위해 모델에 주입된다. 특히, 모델의 성능을 증강시키기 위해 식별된 수직 포지션에서 모델에 새로운 스페셜리스트 층들이 삽입된다. 다수의 유형의 객체들에 대한 분류를 수행하기 위한 단 하나의 범용 경로만을 갖기보다는, 제1 새로운 스페셜리스트 층이 단지 제1 유형의 객체의 분류를 해결하기 위해 부가될 수 있고, 제2 새로운 스페셜리스트 층은 단

지 제2 유형의 객체의 분류를 해결하기 위해 부가될 수 있다. 이러한 액션을 취함으로써, 시간이 지남에 따라, 이들 스페셜리스트 컴포넌트들 각각은, 그 자신의 전용 전문 분야에 관해 고도로 정통하게 되는데, 그 이유는 스페셜리스트가 그 스페셜리스트 컴포넌트에 할당된 특정 서브도메인에 관한 대규모 세부사항 레벨들(extensive levels of detail)을 학습하도록 강제되기 때문이다. 이러한 방식으로, 모델은 네트워크의 다른 부분들에 비해 차선인 것으로 특별히 식별된 분류 영역들을 직접 해결할 새로운 층들을 부가함으로써 개선된다. 이와 동일한 프로세스가 모델의 잔여부를 통해 지속되어 수정 및/또는 증강되어야 하는 임의의 부가적인 층들을 식별한다.

[0014] 소정의 실시예들에서, "손실(loss)" 메커니즘(예를 들어, 손실 층, 손실 함수 및/또는 비용 함수)이 네트워크의 각각의 층에 포함된다. 단지 단일 최상부-손실 층을 갖는 대신에, 부가적인 손실 층들이 네트워크 내의 다른 층에 부가되며, 예를 들어, 여기서 딥 신경 네트워크는 피쳐 추출의 중간 및 최종 스테이지들에 다수의 손실 층들을 가지며, 여기서 각각의 손실 층은 그 깊이 지점까지의 네트워크의 성능을 측정한다. 예측들이 각각의 손실 층에서 생성될 수 있고 네트워크에 대한 모든 혼동 행렬들을 포함하는 텐서 T를 형성하도록 각각의 혼동 행렬로 변환될 수 있다. T 및 그의 엘리먼트들의 구조를 분석함으로써, 깊이 및 폭 둘 모두의 관점에서 네트워크의 기존 구조를 수정 및 증강시키는 것이 목표이다. 한편으로, 피쳐 공유를 최대화하고 컴퓨테이션을 감소시키고, 그럼에도 불구하고 다른 한편으로는 정확도를 증가시키기 위해, 기존 네트워크의 구조를 재구성하는 것이 목표이다. 이를 위해, 접근법은 그의 현재 성능에 따라 네트워크의 깊이는 물론, 폭을 파티셔닝한다. 따라서, 수직 분할은 일부 실시예들에서, 예를 들어, 상이한 층들 간의 내적을 컴퓨팅함으로써 수행된다. 아키텍처를 깊이 면에서 파티셔닝하기 위해, 일부 실시예들은 이웃 층들에서의 연속적인 손실 함수 평가들에 대응하는 이웃 서브공간들을 비교한다. 또한, 예를 들어, K-웨이 분기(K-way Bifurcation)를 수행함으로써 수평 분할이 수행된다. 특정 층에서의 네트워크의 성능을 개선하기 위해, 그의 구조(예를 들어, 완전 콘볼루션층)는 증강을 필요로 할 수 있다. 네트워크의 부분들은 일반적인 지식(제너럴리스트)에 초점을 맞추는 반면, 다른 부분들은 서로 간에 높은 유사성을 갖는 라벨들의 소형 서브세트들(스페셜리스트)에 집중한다. 층 i에 의해 달성된 지식은 네트워크의 제1 수평 파티셔닝을 수행하는 데 사용될 것이다. 네트워크의 최상부에 도달할 때까지 프로세싱이 (예를 들어, 재귀적 방식으로) 지속된다. 이 지점에서, 최종 모델이 컴퓨터 판독 가능 매체에 저장된다.

[0015] 일부 실시예들은 스페셜리스트들의 딥 러닝(deep learning)과 관련된다. 제너럴리스트의 구조는 일반적인 지식으로 잘 수행되는 것으로 알려져 있지만, 이와 동일한 구조가 스페셜리스트(스페셜리스트의 태스크는 보다 단순하거나 복잡한 표현을 요구할 수 있음)에서 잘 수행될 것임은 보장되지 않는다. 일부 실시예들은, 각각의 스페셜리스트의 구조가 데이터-주도 방식(data-driven manner)으로 깊이-별 분할(depth-wise splitting)을 통해 제너럴리스트의 구조로부터 벗어나도록 허용한다.

[0016] 이들 기술들의 부가적인 변동들이 대안적인 실시예들에서 적용될 수 있다. 예를 들어, 모든 각각의 분할들(수직 또는 수평)의 쌍에 대해, 네트워크는 주어진 경로에서 분류를 획득하도록 재트레이닝될 수 있다. 이를테면, 응집성 클러스터링(agglomerative clustering) 및/또는 분할에 의해, 이를 가속화하고 그리고/또는 이를 완전히 회피하기 위해 기술들이 소정의 실시예들에서 적용될 수 있다. 또한, 혼동 행렬( $C_i$ ) 및 그의 파티셔닝(K)를 고려하여, 응집성 클러스터링이  $C_i$ 의 K 부분들 각각 상에서 수행되어 추가의 분할들을 추정할 수 있다. 이는 비용  $X_0$ 로 이어진다. 비용  $X_0$ 는 높은 레벨의 네트워크에서 새로운 혼동 행렬들을 학습하는 지도된 그룹핑의 비용이다.  $X_0$ 는  $X_s + \text{Tau}$  이하며, 여기서 Tau는 클러스터링 에러의 상한이다.

[0017] 일부 실시예들에서, 변동들은 콘볼루션층 대 완전-연결(1x1 콘볼루션)과 관련하여 고려된다. 콘볼루션층들(심지어, 이를테면, 의미론적 구분(semantic segmentation)의 경우에, 완전 콘볼루션층들) 사이에 분할이 필요한 경우, 층의 선형 크기(이 경우 fc)를 변경하는 대신에, 디멘션의 깊이가 변경되어 클래스들의 수(이것은 FCN의 확장임)를 반영할 수 있다.

[0018] 경로 마다 붕괴 또는 부가 또는 수직 층들 사용하고, 라벨 공간의 함수로써 층의 크기를 변경하고, 그리고/또는 검출 및 RNN에 대한 확장(혼동들을 비교함으로써 동일한 방식으로 전개(unrolling)된)하여 추가의 변동들 및 실시예들이 생성될 수 있다.

[0019] 또 다른 실시예에서, 필요한 프로세싱 태스크들에 대해 더 적은 층들로도 충분하도록 하기 위해 네트워크에 너무 많은 층들이 있을 수 있는 때를 식별하기 위한 기술이 적용될 수 있다. 위에서 언급된 바와 같이, 네트워크에 깊이가 신뢰성있게 추가될 수 있고, 충분한 트레이닝 데이터가 주어지면, 성능의 개선이 확인될 수

있다. 그러나 성능 면에서 이러한 부가된 상승은 FLOP들 및 메모리 소비의 관점에서 비용이 들 수 있다. 일부 실시예들에서, 네트워크는 이 트레이드오프를 염두에 두고 네트워크의 주어진 컴퓨테이션 층이 이진 결정을 통해 사용되는지 여부를 학습하는 올-오아-나씽 하이웨이 네트워크(all-or-nothing highway network)의 사용과 관련하여 최적화된다. 주어진 컴퓨테이션 블록이 사용되는 경우, 페널티가 초래된다. 이 페널티 항목을 변동 시킴으로써, 타겟 아키텍처를 염두에 두고 학습 프로세스가 커스텀화(customize)될 수 있고: 임베디드 시스템은 클라우드-기반 시스템보다 훨씬 더 간결한(leaner) 아키텍처를 선호할 것이다.

[0020] 본 발명의 양상들, 목적들 및 이점들에 대한 추가의 세부사항들은 아래의 상세한 설명, 도면들 및 청구항들에서 설명된다. 위의 일반적인 설명 및 다음의 상세한 설명은 둘 모두 예시적이고 설명적이며, 본 발명의 범위에 관해 제한하는 것으로 의도되지 않는다.

### 도면의 간단한 설명

[0021] 도면들은 본 발명의 다양한 실시예들의 설계 및 활용을 예시한다. 도면들은 실체대로 그려진 것이 아니며 유사한 구조들 또는 기능들의 엘리먼트들은 도면들 전체에 걸쳐 유사한 참조 번호들로 표현된다는 것이 주의되어야 한다. 본 발명의 다양한 실시예들의 위에서 언급된 그리고 다른 이점들 및 목적들이 어떻게 달성되는지를 더 잘 인지하기 위해, 위에서 간략하게 설명한 본 발명들의 보다 상세한 설명이 첨부 도면들에서 예시되는 본 발명의 특정 실시예들을 참조하여 제공될 것이다. 이들 도면들이 단지 본 발명의 통상적인 실시예들을 도시할 뿐이며, 이에 따라 본 발명의 범위를 제한하는 것으로 간주되지 않는다는 것을 조건으로, 본 발명은 첨부된 도면들의 사용을 통해 부가적인 특이성 및 세부사항에 관해 설명되고 기술될 것이다.

[0022] 도 1은 신경 네트워크들에 대한 구조 학습을 구현하기 위해 본 발명의 일부 실시예들에서 사용될 수 있는 예시적인 시스템을 예시한다.

[0023] 도 2는 본 발명의 일부 실시예들에 따라, 신경 네트워크들에 대한 구조 학습을 구현하기 위한 접근법의 흐름도를 도시한다.

[0024] 도 3은 일부 실시예들에 따라, 신경 네트워크들에 대한 구조 학습을 구현하기 위한 접근법의 보다 상세한 흐름도를 예시한다.

[0025] 도 4a 내지 도 4f는 본 발명의 다양한 실시예들을 예시한다.

[0026] 도 5a 및 도 5b는 네트워크에 너무 많은 층들이 존재할 때를 식별하기 위한 접근법을 예시한다.

[0027] 도 6a 내지 도 6d는 다양한 실시예들에 대한 일반적인 AR 시스템 컴포넌트 옵션들을 예시한다.

[0028] 도 7은 본 발명의 일부 실시예들이 구현될 수 있는 컴퓨터화된 시스템을 도시한다.

### 발명을 실시하기 위한 구체적인 내용

[0022] 본 발명의 일부 실시예들은 신경 네트워크들에 대한 구조 학습을 구현하기 위한 개선된 접근법들에 관한 것이다. 접근법은 네트워크로 시작하고, 라벨링된 데이터를 갖는 문제를 네트워크에 제공하고, 그 후 이 네트워크에 의해 생성된 출력의 구조를 검토한다. 그 후, 네트워크의 아키텍처는 특정 문제에 대한 더 양호한 솔루션을 획득하도록 수정된다. 전문가들이 몹시 복잡한 도메인-특정 네트워크 아키텍처들을 생각해 내게 하기 보다는, 이 접근법은 데이터가 특정 태스크에 대해 사용될 네트워크 아키텍처를 구동하도록 허용한다.

[0023] 도 1은 신경 네트워크들에 대한 구조 학습을 구현하기 위해 본 발명의 일부 실시예들에서 사용될 수 있는 예시적인 시스템을 예시한다. 시스템은, 컴퓨팅 시스템(107 또는 115)과 인터페이스싱하고 이를 동작시켜 시스템을 제어하고 그리고/또는 이 시스템과 상호작용하는 하나 또는 그 초과와 사용자들을 포함할 수 있다. 시스템은 신경 네트워크 컴퓨팅 디바이스(107) 또는 사용자 컴퓨팅 디바이스(115)를 동작시키고, 이와 인터페이스하거나 이를 구현하는 데 사용될 수 있는 임의의 유형의 컴퓨팅 스테이션을 포함한다. 이러한 컴퓨팅 시스템들의 예들은, 예를 들어, 서버들, 워크스테이션들, 개인용 컴퓨터들, 또는 네트워킹된 또는 클라우드-기반 컴퓨팅 플랫폼에 연결된 원격 컴퓨팅 단말들을 포함한다. 컴퓨팅 시스템은 사용자가 시스템의 활동들에 관한 동작 제어를 제공하기 위한 하나 또는 그 초과와 입력 디바이스들, 이를테면, 포인팅 객체를 조작하기 위한 마우스 또는 키보드를 포함할 수 있다. 컴퓨팅 시스템은 또한, 컴퓨팅 시스템의 사용자들에 대한 분석 결과들 및/또는 제어 인터페이스들을 위한 디스플레이 디바이스 이를테면, 디스플레이 모니터와 연관될 수 있다.

[0024] 일부 실시예들에서, 시스템은 컴퓨터 비전 기능성을 구현하는 데 사용된다. 따라서, 시스템은 시스템

이 동작하는 환경에서 하나 또는 그 초과 객체들(105)에 대한 이미지 데이터(101)를 캡처하기 위해 카메라(103)와 같은 하나 또는 그 초과 이미지 캡처 디바이스들을 포함할 수 있다. 이미지 데이터(101) 및/또는 임의의 분석 결과들(예를 들어, 분류 출력 데이터(113))은 하나 또는 그 초과 컴퓨터 판독 가능 저장 매체들에 저장될 수 있다. 컴퓨터 판독 가능 저장 매체는 컴퓨터 판독 가능 저장 매체에 로케이팅된 데이터에 대한 준비된 액세스를 허용하는 하드웨어 및/또는 소프트웨어의 임의의 결합을 포함한다. 예를 들어, 컴퓨터 판독 가능 저장 매체는 운영 체제에 의해 동작 가능하게 관리되는 컴퓨터 메모리 및/또는 하드 드라이브 저장소, 및/또는 네트워크된 저장 디바이스의 원격 저장소, 이를테면, NAS(networked attached storage), SAN(storage area network) 또는 클라우드 저장소로서 구현될 수 있다. 컴퓨터 판독 가능 저장 매체는 또한 영구적인 저장소 및/또는 비-영구적인 저장소를 갖는 전자 데이터베이스 시스템으로서 구현될 수 있다.

[0025] [0032] 신경 네트워크 컴퓨팅 디바이스(107)는 원래의 모델(1)을 개선된 모델(n)로 수정하기 위한 구조 학습 모듈(109)을 포함하며, 여기서 모델(n)은 모델 내의 층들을 수정하기 위한 가능한 다수의 반복 프로세스들의 결과들이다. 모델(n)은 바람직하게는, 지식의 깊이 및 폭, 본질적으로 전문가의 혼합(mixture of experts)을 포함한다. 이 모델은 대강의(coarse) 카테고리들 간의 차이를 이해해야 하지만 동시에, 다양한 도메인들에 걸친 세분화된 클래스들의 차이를 이해해야 한다. 새로운 스페셜리스트 층들(111)은 이러한 목표들을 구현하기 위해 필요에 따라 모델에 추가된다. 이러한 시스템의 설계는 이들이 필요한 곳에만 자원들을 추가하는 제약에 의해 통제된다. 네트워크를 임의로 더 깊고 넓게 함으로써 네트워크를 단순히 확장시키는 것은, 컴퓨테이션 제약들로 인해 그대로 스케일링되지 않고, 이에 따라 본 접근법은 추가의 규칙화 트릭(regularization trick)들에 대한 필요성을 회피한다.

[0026] [0033] 도 2는 본 발명의 일부 실시예들에 따라, 신경 네트워크들에 대한 구조 학습을 구현하기 위한 접근법의 흐름도를 도시한다. 본 접근법은 네트워크가 해결하고자 하는 문제/데이터의 상관관계들(correlations)을 이용함으로써 신경 네트워크들의 구조 학습을 구현한다. 하단 콘볼루션층들로부터 완전히 연결된 층들까지 내내 정보 이득의 병목들을 발견하는 그리디 접근법(greedy approach)이 설명된다. 단순히 아키텍처를 임의로 더 깊게 하기 보다는, 부가적인 컴퓨테이션 및 커패시턴스는 그것이 필요할 곳에만 추가된다.

[0027] [0034] 131에서, 네트워크는 초기 시점에 생성된다. 임의의 적합한 접근법이 네트워크를 생성하는 데 사용될 수 있다. 예를 들어, 종래의 알렉스넷(Alexnet) 또는 구글넷(Googlenet) 접근법들이 네트워크를 생성하는 데 사용될 수 있다.

[0028] [0035] 다음으로, 133에서, 예를 들어, 이미지 인식/객체 분류 태스크들을 수행하기 위해 지정된 태스크에 적용될 때, 네트워크로부터 출력 세트가 생성된다. 예를 들어, 다수의 사람들 및 동물들이 환경 내에 있고, 할당된 태스크는 이미지 데이터를 분석하여 환경 내에서 관찰될 수 있는 상이한 사람들 및 동물들의 유형들을 분류하는 것이라고 가정한다. 모델의 각각의 층은 그 층 내에서 수행되는 활동들에 대한 소정의 출력들을 제공한다. 출력은 그에 대한 소정의 구조들을 가지며, 이는 분류 문제가 해결되는데 있어 클래스들 간의 관계들을 확인하기 위해 검토될 수 있다.

[0029] [0036] 135에서, 네트워크 모델 내의 다양한 층들은 모델 내의 최저 성능 층을 식별하도록 분석된다. 예를 들어, 층들(1 내지 3) 및 층들(5 내지 10) 각각이 10%의 분류 정확도 개선을 제공하지만, 층(4)은 1%의 개선만을 제공하는, 10개의 층들을 갖는 모델을 가정한다. 이 상황에서, 층(4)은 최저 성능 층으로서 식별될 것이다.

[0030] [0037] 다음으로, 137에서, 부가적인 구조들이 모델의 성능을 개선하기 위해 모델에 주입된다. 특히, 모델의 성능을 증강시키기 위해, 식별된 수직 포지션의 모델에 새로운 스페셜리스트 층(new specialist layer)들이 삽입된다.

[0031] [0038] 본 발명의 실시예의 이러한 양상을 설명하기 위해, 모델은 도 4a에 예시된 바와 같이 환경에서 사람들 및 동물들의 분류들을 수행하도록 의도된 것으로 가정한다. 여기서, 이미지 캡처 디바이스는 상이한 사람들(예를 들어, 여성(401), 남성(403) 및 어린이(405))의 이미지들을 캡처한다. 또한, 환경은 다수의 동물들(예를 들어, 고양이(407), 개(409) 및 쥐(411))을 포함한다. 또한, 기존 모델은 동물들(407, 409, 411)로부터 사람들(401, 403, 405)을 성공적으로 구별할 수 있지만, 상이한 사람들을 서로 구별하거나 상이한 유형의 동물들을 서로 구별하는 것은 더 어려운 것으로 보인다고 가정한다. 네트워크(예를 들어, 오라클 네트워크)로부터 학습될 수 있는 실제 구조를 검토하면, 네트워크는 행해지고 있는 예측들 간에 학습 종속성을 포함한다는 것이 명확하다. 그러나 종래의 딥-러닝 아키텍처 설계에서, 이는 활용되지 않는다. 이 구조를 훨씬 더 가까이에서 보면, 시스템은 실제로 시각적으로 서로 유사한 개념들을 학습하고 있다는 것이 자명하다. 도 4b를 참조하면, 클래스들에 관한 3D로 된 예시적인 산점도가, 다수의 그룹들로 클러스터링된, 완전히-트레이닝된 알렉스넷에 대



한 예측들의 예시적인 구조를 예시하기 위해 도시된다. 지점들 사이의 거리는 개념들 간의 시각적 유사성에 대응한다. 여기서, 사람 객체들에 대한 지점들의 제1 밀집 클러스터링 및 동물 객체들에 대한 지점들의 제2 밀집 클러스터링이 있다는 것을 알 수 있다. 이는, 한 사람을 다른 사람으로부터 또는 한 동물을 다른 동물로부터 구별하는데 있을 수 있는 모델에서의 어려움들을 야기할 수 있는 현상이다.

[0032] [0039] 이러한 상황에서, 본 발명의 일부 실시예들에서, 이러한 유형들의 객체들 모두에 대한 분류를 수행하기 위한 단 하나의 범용 경로만을 갖기보다는, 제1 새로운 스페셜리스트 층이 단지 사람들의 분류를 해결하기 위해 부가될 수 있고, 제2 새로운 스페셜리스트 층은 단지 동물들의 분류를 해결하기 위해 부가될 수 있다. 따라서, 하나의 스페셜리스트(사람 스페셜리스트 층)가 도 4b의 차트의 부분(413)에 대한 데이터를 처리하기 위해 할당되는 반면, 제2 스페셜리스트(동물 스페셜리스트 층)가 도 4b의 부분(415)에 대한 데이터를 처리하기 위해 할당될 것이다. 이러한 액션을 취함으로써, 시간이 지남에 따라, 이들 스페셜리스트 컴포넌트들 각각은, 그 자신의 전용 전문 분야에 관해 고도로 정통하게 되는데, 그 이유는 스페셜리스트가 그 스페셜리스트 컴포넌트에 할당된 특정 서브도메인에 관한 대규모 세부사항 레벨들(extensive levels of detail)을 학습하도록 강제되기 때문이다. 이러한 방식으로, 모델은 네트워크의 다른 부분들에 비해 차선인 것으로 특별히 식별된 분류 영역들을 직접 해결할 새로운 층들을 부가함으로써 개선된다.

[0033] [0040] 이와 동일한 프로세스가 모델의 잔여부를 통해 지속되어 수정 및/또는 증강되어야 하는 임의의 부가적인 층들을 식별한다. 따라서, 139에서, 프로세싱이 네트워크의 최상부에 도달했는지에 관한 결정이 내려진다. 만약 그렇다면, 모델은 141에서 완결된다. 만약 그렇지 않다면, 프로세스는 네트워크의 최상부가 도달될 때까지 프로세스를 지속하도록 133으로 되돌아간다.

[0034] [0041] 임의의 기성품의 콘볼루션 신경 네트워크의 아키텍처를 수정 및 개선하기 위해, 이 접근법이 취해질 수 있다. 본 개시내용의 본 접근법을 따름으로써, 임의의 신경 네트워크는, (a) 그 구조에서 정보 이득 병목(information gain bottleneck)을 식별하고, (b) 병목을 완화시키기 위해 예측들의 구조를 적용하고, 마지막으로, (c) 스페셜리스트 경로들의 깊이를 결정함으로써 개선될 수 있다.

[0035] [0042] 도 3은 일부 실시예들에 따라, 신경 네트워크들에 대한 구조 학습을 구현하기 위한 접근법의 보다 상세한 흐름도를 예시한다. 이 흐름의 목적들을 위해, 네트워크(예를 들어, 모놀리식(monolithic) 네트워크)가 알렉스넷 또는 구글넷과 같은 임의의 적절한 접근법에 따라 이미 생성되었다고 가정한다.

[0036] [0043] 151에서, "손실" 메커니즘(예를 들어, 손실 층, 손실 함수 및/또는 비용 함수)이 네트워크의 각각의 층에 포함된다. 손실 메커니즘은 이벤트 또는 값을, 신경 네트워크 내의 프로세싱과 연관된 비용 또는 에러 값의 표현에 맵핑하는 기능에 대응한다. 도 4c에 도시된 바와 같이, 단지 단일 최상부-레벨 손실 층(421)을 갖는 대신에, 부가적인 손실 층들(423)이 네트워크 내의 다른 층에 부가된다. 그러므로, 이 도면은, 피쳐 추출의 중간 및 최종 스테이지들에 다수의 손실 층들을 갖는 딥 신경 네트워크의 예를 도시하며, 여기서 각각의 손실 층은 그 깊이 지점까지의 네트워크의 성능을 측정한다. 목표는 네트워크 아키텍처를 증강 및 수정하여 태스크에 가장 잘 맞도록 그의 아키텍처를 수정함으로써 가능한 한 최상으로 주어진 문제를 해결하는 것임을 상기하라. 따라서, 이 접근법은 네트워크 전반에 걸친 다양한 손실 층들에서 형성된 예측들을 분석하고 이들 사이의 혼동에 기초하여 신경 활성화들(neuron activations)을 그룹핑한다.

[0037] [0044] 도 4d 및 도 4e에 예시된 바와 같이, 예측들이 각각의 손실 층에서 생성되고 (도 4d에 도시된 바와 같이) 각각의 혼동 행렬들로 변환되어, (도 4e에 도시된 바와 같이) 네트워크(예를 들어, 오라클 네트워크)에 대한 모든 혼동 행렬들을 포함하는 텐서(tensor) T를 형성한다. T 및 그의 엘리먼트들의 구조를 분석함으로써, 깊이 및 폭 둘 모두의 관점에서 네트워크의 기존 구조를 수정 및 증강시키는 것이 목표이다.

[0038] [0045] 설명하자면,  $C_i$ 를 클래스들 및 손실 층  $i$ 의 혼동 행렬이라고 하면,

$$A_i = C_i C_i^T \quad (1)$$

$$D(i, i) = \sum_{i=1}^I A(i, :) \quad (2)$$

$$L_i = D_i^{-1/2} A_i D_i^{1/2} \quad (3)$$

$$\hat{C}_i = \text{top}_{N_E}(\text{eig}(L_i)), \quad (4)$$

[0039] 그리고,

[0040] 여기서  $A_i$ 는 손실 층  $i$ 에서의 유사도 행렬(affinity matrix)이고,  $D_i$ 는 대각 행렬이고,  $L_i$ 는 그래프 라플라시안

(graph Laplacian)이고,  $\hat{C}_i$  는  $C_i$ 에 의해 생성된 유사도 행렬의 그래프 라플라시안의 선행 고유벡터에 의해 스패닝되는(spanned) 서브공간이다. 결과적으로, 텐서 :  $\hat{T} = \{\hat{C}_1, \hat{C}_2, \hat{C}_3, \dots, \hat{C}_n\}$ .

[0041] [0046] 한편으로, 피쳐 공유를 최대화하고 컴퓨테이션을 감소시키고, 그림에도 불구하고 다른 한편으로는 정확도를 증가시키기 위해, 기존 네트워크의 구조를 재구성하는 것이 목표이다. 이를 위해, 접근법은 그의 현재 성능에 따라 네트워크의 깊이는 물론, 폭을 파티셔닝한다.

[0042] [0047] 따라서, 153에서, 예를 들어, 상이한 층들 사이의 내적을 컴퓨팅함으로써 수직 분할이 수행된다. 아키텍처를 깊이 면에서 파티셔닝하기 위해, 일부 실시예들은 다음의 수식을 사용하여 이웃 층들에서의 연속적인 손실 함수 평가들에 대응하는 이웃 서브공간들을 비교한다 :

$$\Phi(i, i+1) = 1 - \frac{1}{N_E} \|\hat{C}_{i+1}^T \hat{C}_i\|_F^2, \quad (5)$$

[0043] [0044] 여기서,  $\hat{C}_i$  및  $\hat{C}_{i+1}$  는 레벨  $i$  및  $i+1$ 에서의 손실 함수들에 대한 혼동 행렬들의 근사 선행 고유벡터들을 나타내고,  $F$ 는 프로베니우스 노름(Frobenius norm)을 나타낸다. 공식적으로,  $\hat{C}_i$  및  $\hat{C}_{i+1}$ 는  $N_E$ -차원 서브 공간들을 나타내며,  $\Phi(i, i+1)$ 는 이들 사이의 정규화된 여각(normalized complement angle)이다. 이 측정( $\Phi$ )은 단지  $\hat{C}_i$  및  $\hat{C}_{i+1}$ 의 열들에 의해 스패닝되는 서브공간에 의존하고 이에 따라 고유벡터들의 회전들에 대해 불변이라는 것에 주의하는 것이 중요하다. 또한  $\Phi$ 는  $[0, 1]$  내에서 제한되며, 레벨들  $i$  및  $i+1$ 은,  $\Phi(i, i+1)$ 가 0에 근접하는 경우 구조 면에서 유사하고  $\Phi(i, i+1)$ 가 직교일 때 정확히 1이라고 간주된다. 스케일 공간의 레벨들 간의 완전한 유사성 관계를 구성하기 위해, 모든 이웃 손실 층들의 쌍들은  $\Phi$ 를 사용하여 비교된다. 설정된 유사성 관계들로 인해, 이제, 모듈리식 네트워크 아키텍처를 파티셔닝하는 문제를 해결하는 것이 가능하다.

[0045] [0049]  $\Phi$ 를  $i$  및  $i+1$ 의 모든 순차적 쌍들의 벡터라고 하자(여기서  $\phi_i = \Phi(i, i+1)$ 임). 0에 가장 근접한  $\phi_i$ 의 값들은 층들( $i$  및  $i+1$ ) 간의 최저 정보 이득을 표시한다. 따라서,  $\text{argmin}(\phi)$ 는 모듈리식 아키텍처의 최적 초기 분할이다. 아키텍처를 깊이 면에서 분할하는 것은 리던던시(redundancy)(0 정보 이득)의 지점들을 식별하면서 피쳐 공유를 가능하게 한다.

[0046] [0050] 155에서, 예를 들어, K-웨이 분기(K-way Bifurcation)를 수행함으로써 수평 분할이 수행된다. 특정 층에서의 네트워크의 성능을 개선하기 위해, 그의 구조(예를 들어, 완전 콘볼루션서널)는 증강을 필요로 할 수 있다. 네트워크의 부분들은 일반적인 지식(제너럴리스트(generalist))에 초점을 맞추는 반면, 다른 부분들은 서로 간에 높은 유사성을 갖는 라벨들의 소형 서브세트들(스페셜리스트)에 집중한다. 층  $i$ 에 의해 달성된 지식은 네트워크의 제1 수평 파티셔닝을 수행하는 데 사용될 것이다.

[0047] [0051] 공식적으로,  $C_i$ 가 주어지면, 위에서 개시된 바와 같이, 수식들(1, 2 및 3)에 따라  $L_i$ 를 컴퓨팅한다. 고유값(eigengap)은 새로운 경로들(스페셜리스트들)의 수를 결정하도록 그래프 라플라시안  $L_i$  선행 고유값들을 분석함으로써 결정된다. 원래의 데이터는  $L_i$ 의 최상부  $N$ 개의 선행 고유벡터들로 투영되고;  $N$ 에서, 데이터는  $k$ 개의 클래스들로 추가로 클러스터링되며, 여기서  $k$ 는 고유값과 동일하다. 이러한 투영 및 그룹핑의 예가 도 4b에 예시된다. 이러한 절차는 제1 분할 후에 네트워크(407)를 예시하는 도 4f에 도시된 바와 같은 아키텍처의 수정으로 이어질 것이다.

[0048] [0052] 제1 분할이 설정되면, 모든 새로운 경로들이 원래의 네트워크로서 취급된다. 분할 절차는 분할할 라벨이 더 이상 남아있지 않을 때까지, 또는 100% 정확도가 달성될 때까지 적용된다.

[0049] [0053] 157에서, 네트워크의 최상부에 도달할 때까지 위의 프로세싱이 (예를 들어, 재귀적 방식으로) 지속된다. 이 지점에서, 최종 모델이 컴퓨터 판독 가능 매체에 저장된다.

[0050] [0054] 본 개시내용의 이 부분은 스페셜리스트들의 딥 러닝과 관련된다. 제너럴리스트의 구조는 일반적인 지식으로 잘 수행되는 것으로 알려져 있지만, 이와 동일한 구조가 스페셜리스트(스페셜리스트의 태스크는 보다 단순하거나 복잡한 표현을 요구할 수 있음)에서 잘 수행될 것임은 보장되지 않는다. 일부 실시예들은, 각각의 스페셜리스트의 구조가 데이터-주도 방식(data-driven manner)으로 깊이-별 분할(depth-wise splitting)을 통해 제너럴리스트의 구조로부터 벗어나도록 허용한다.

[0051] [0055]  $L = \{L_1, L_2, \dots, L_n\}$ 가 추가의 분할을 위해 고려되는 완전-연결 층(fully-connected layer)들의 세트라고 한다. 출력(y)을 생성하는 L 내의 층( $L_i$ )을 고려한다. 이 층이  $y = \sigma(f(x))$ 로서 그의 입력에 적용하는 변형이 작성될 수 있으며, 여기서  $\sigma()$ 는 ReLU 및  $f(x) = Wx$ 와 같은 비-선형성을 적용하며, 여기서 W는 디멘션들( $M \times N$ )의 학습된 가중치 행렬이고 디멘션들( $N \times 1$ )을 갖는 x는 이 층에 입력된다. 분할을 수행하기 위해 접근법은  $y = \sigma_1(g(\sigma_2(h(x))))$ 로의  $L_i$ 의 변형을 분해하며, 여기서  $\sigma_1()$  및  $\sigma_2()$ 는 활성화 함수이고  $g(x) = W_1x$  및  $h(x) = W_2x$ 이며, 여기서  $W_1$ 은 디멘션들( $N \times N$ )을 갖고,  $W_2$ 는 디멘션들( $M \times N$ )을 갖는다. 접근법은 다음을 선택한다:

$$\sigma_1(x) = \sigma(x) \quad (6)$$

$$g(x) = U\sqrt{\Sigma}x \quad (7)$$

$$\sigma_2(x) = Ix \quad (8)$$

$$h(x) = \sqrt{\Sigma}V^Tx \quad (9)$$

[0052]

[0053] [0056] 여기서  $W = U\Sigma V^T$ 는 W의 SVD 인수분해이고, I는 아이덴티티 행렬이다. 이 변화를 통해, 층( $L_i$ )의 변형은 변하지 않는다.  $L_i$ 의 학습된 표현의 복잡성을 증가시키기 위해, ReLU와 같은 비선형 활성화 함수로서  $\sigma_2$ 이 세팅될 수 있다. 그러나 이 비선형성을 부가하는 것은  $L_i$ 의 학습된 표현에서의 급격한 변화를 야기하고 네트워크가 그의 학습 대부분을 처음부터 재시작하게 할 수 있다. 대신, PReLU 비선형성이 삽입되고 그의 단일 파라미터 a를 1이 되도록 초기화할 수 있으며, 이는 수식 8에서 I와 등가이다. 이는, 이 층에 새로운 비선형성을 도입하기 위한 원활한(smooth) 메커니즘을 스페셜리스트에 제공한다.

[0054] [0057] 층들의 세트 L이 주어지면, 위의 전략이 각각의 층  $L_i$ 에 독립적으로 적용될 수 있고, 트레이닝 손실에 관한 최상의 개선을 제공하는 분할을 그리디하게(greedily) 선택될 수 있다. 이 프로세스는 층들의 세트 ( $L_{\text{new}} = \{L_1, L_2, \dots, L_n, L_{n+1}\}$ )에 대해 재귀적으로 반복될 수 있다.

[0055] [0058] 이들 기술들의 부가적인 변동들이 대안적인 실시예들에서 적용될 수 있다. 예를 들어, 모든 각각의 분할들(수직 또는 수평)의 쌍에 대해, 네트워크는 주어진 경로에서 분류를 획득하도록 재트레이닝될 수 있다. 이를테면, 응집성 클러스터링(agglomerative clustering) 및/또는 분할에 의해, 이를 가속화하고 그리고/또는 이를 완전히 회피하기 위해 기술들이 소정의 실시예들에서 적용될 수 있다. 또한, 혼동 행렬( $C_i$ ) 및 그의 파티셔닝(K)이 주어지면, 응집성 클러스터링이  $C_i$ 의 K 부분들 각각 상에서 수행되어 추가의 분할들을 추정할 수 있다. 이는 비용  $X_u$ 로 이어진다. 비용  $X_s$ 는 높은 레벨의 네트워크에서 새로운 혼동 행렬들을 학습하는 지도된 그룹핑의 비용이다.  $X_u$ 는  $X_s + \text{Tau}$  이하이며, 여기서 Tau는 클러스터링 에러의 상한이다.

[0056] [0059] 일부 실시예들에서, 변동들은 콘볼루션 층 대 완전-연결( $1 \times 1$  콘볼루션)과 관련하여 고려된다. 콘볼루션 층들(이를테면, 의미론적 구분(semantic segmentation)의 경우에, 심지어 완전 콘볼루션 층들) 사이에 분할이 필요한 경우, 층의 선형 크기(이 경우 fc)를 변경하는 대신에, 디멘션의 깊이가 변경되어 클래스들의 수(이것은 FCN에 대한 확장임)를 반영할 수 있다.

[0057] [0060] 경로 마다 붕괴 또는 부가 또는 수직 층들 사용하고, 라벨 공간의 함수로써 층의 크기를 변경하고, 그

리고/또는 검출 및 RNN에 대한 확장(혼동들을 비교함으로써 동일한 방식으로 전개(unrolling)된)하여 추가의 변동들 및 실시예들이 생성될 수 있다.

[0058] [0061] 또 다른 실시예에서, 필요한 프로세싱 태스크들에 대해 더 적은 층들로도 충분하도록 하기 위해 네트워크에 너무 많은 층들이 있을 수 있는 때를 식별하기 위한 기술이 적용될 수 있다. 위에서 언급된 바와 같이, 네트워크에 깊이가 신뢰성있게 추가될 수 있고, 충분한 트레이닝 데이터가 주어지면, 성능의 개선이 확인될 수 있다. 그러나 성능 면에서 이러한 부가된 상승은 FLOP들 및 메모리 소비의 관점에서 비용이 들 수 있다. 일부 실시예들에서, 네트워크는 이 트레이드오프를 염두에 두고 네트워크의 주어진 컴퓨테이션 층이 이진 결정을 통해 사용되는지 여부를 학습하는 올-오아-나씽 하이웨이 네트워크(all-or-nothing highway network)의 사용과 관련하여 최적화된다. 주어진 컴퓨테이션 블록이 사용되는 경우, 페널티가 초래된다. 이 페널티 항목을 변동시킴으로써, 타겟 아키텍처를 염두에 두고 학습 프로세스가 커스텀화(customize)될 수 있고: 임베디드 시스템은 클라우드-기반 시스템보다 훨씬 더 간결한(leaner) 아키텍처를 선호할 것이다.

[0059] [0062] 이 실시예에 의해 해결되는 이슈는, 주어진 문제  $X$ 에 대해 네트워크에 어느 정도 깊이로 컴퓨테이션 블록이 주어져야 하는지를 결정하는 것이다. 올-오아-나씽 하이웨이 네트워크를 사용하는 접근법에 있어서, 하이웨이 네트워크들은 현재 컴퓨테이션 블록의 출력과의 혼합 이전에 이전 층으로부터의 스킵 연결(skip connection)이 어떻게 변형되어야 하는지를 학습하기 위해 혼합 행렬을 도입한다. 다음의 수식을 고려한다:

$$y = F(x, W_i) + W_s x \quad (10)$$

[0061] 잔여 네트워크들은 스킵 연결을 결합하도록 아이덴티티 맵핑을 사용하는데 있어 성공을 거둘 수 있다. 아이덴티티 맵핑이 덜 대표적(representative)이긴하지만, 그것은 최적화하기에 더 효율적이고 쉽다:

$$y = F(x, W_i) + x \quad (11)$$

[0063] [0064] 현재의 접근법은 컴퓨테이션 블록의 출력을 게이팅(gate)하는 단일 스칼라  $\alpha$ 에 의해 혼합 행렬을 대신 파라미터화한다(도 5a 참조):

$$y = \alpha F(x, W_i) + x \quad (12)$$

[0065]  $\alpha = 0$ 이면  $y = x$ 이고 입력은 단순히 출력에 전달된다.  $\alpha = 1$  일 때 (수식 12)는 (수식 10)이 되고 잔여 유닛은 컴퓨테이션을 위해 사용된다.

[0066] 도 5a는 올-오아-나씽 하이웨이 연결을 갖는 네트워크에 대한 차트(501)를 예시한다. 이 도면에서, 컴퓨테이션 블록은 입력을 공급받고 잔여 연결을 통해 추후에 결합된다(엘리먼트별 부가). 부가되기 전에, 컴퓨테이션 블록의 출력은 이 컴퓨테이션 블록의 사용에 불이익을 주는 학습된 파라미터  $\alpha$ 에 의해 스케일링된다. 이 손실은 아래에 설명된다.

[0067] 컴퓨테이션 블록을 사용할지 여부를 결정하기 위해 학습이 수행된다. 딥 네트워크에서 주어진 층의 거동을 제어하는  $\alpha$  파라미터에 우선순위를 부여하고 모델 파라미터들 및 그의 목적 함수와 공동으로 이 파라미터를 최적화하는 것이 바람직하다. 트레이닝 동안,  $\alpha$ 가 각각의 깊이에 대해 0 또는 1을 독립적으로 선택하는 이진 결정을 장려하는 것이 바람직하다. 컴퓨테이션 블록이 스킵되는 것으로 학습되는 경우, 그것은 추론 시간에 모델로부터 그 컴퓨테이션 블록을 단순히 제거할 수 있다.

[0068] 잔여 네트워크에서, 연속 층들은 일반적으로, 학습된 잔여 함수들이 작은 응답들을 갖는 작은 맵핑들을 가지며, 이는 아이덴티티 맵핑들이 합리적인 프리컨디셔닝(preconditioning)을 제공할 것을 제안한다. 이는 (수식 10)의 아이덴티티 맵과 아이덴티티 층 간(그리고 그 반대도 가능함)의 트랜지셔널이 목적 함수의 치명적인 변화를 야기하지 않아야 함을 제안한다. 따라서, 본 접근법은 다양한 깊이들에서 컴퓨테이션 블록의 출력을 게이팅하는,  $\alpha$  파라미터에 대한 조각별 원활한 손실 함수(piecewise smooth loss function)를 도입한다.

[0069] 또한, 상이한 시나리오들에 대해, 더 많은 컴퓨테이션을 사용하는 모델들에 더 높은 페널티가 할당되도록  $\alpha$  파라미터들에 대한 손실 함수를 파라미터화하는 것이 바람직하다. 스마트폰과 같은 가벼운 임베디드 플랫폼의 경우, 그것은 층의 선택에 대해 높은 페널티를 원할 수 있다. 클라우드 컴퓨팅 플랫폼의 경우에, 컴퓨테이션 블록의 사용에 대해 그러한 어떠한 페널티도 원하지 않을 수 있다. 이러한 기준들이 주어지면, 다음에 의해 파라미터화될 수 있는, 도 5b에 도시된 조각별 원활한 다항식/선형 함수를 사용할 수 있다.



```

if x < 0.:
    y = (np.absolute(x) * self.steepness)
elif x > 1.:
    y = (x - 1.) * self.steepness + self.peak*0.125
elif x < 0.5:
    y = -self.peak * (x**2. - x)
else:
    y = -self.peak/2. * (x**2. - x) + self.peak*0.125

```

[0070]

[0070] 도 5b에 도시된 피크의 다양한 선택들을 위해, 변동되는 사용 페널티가 모델에 주어진다.

[0072]

증강 현실 및 컴퓨팅 시스템 아키텍처(들)

[0073]

[0071] 위에서 설명된 기술들은, 특히 가상 현실 및 증강 현실 시스템들에 대한 머신 비전 애플리케이션들에 적용 가능하다. 본 발명의 신경 네트워크 분류 디바이스는, AR 시스템들과 독립적으로 구현될 수 있지만, 이하의 다수의 실시예들은 단지 예시 목적들을 위해 AR 시스템들과 관련하여 설명된다.

[0074]

[0072] 다양한 컴퓨터 시스템들에 대한 분류 및 인식을 위한 디바이스들, 방법들 및 시스템들이 개시된다. 일 실시예에서, 컴퓨터 시스템은 다양한 다른 컴퓨터 시스템들(예를 들어, 금융 컴퓨터 시스템들)과의 사용자 상호작용을 용이하게 하도록 구성된 머리-장착 시스템일 수 있다. 다른 실시예들에서, 컴퓨터 시스템은 사용자 금융 트랜잭션들을 용이하게 하도록 구성된 정적인 디바이스(예를 들어, 상업용 단말기 또는 ATM)일 수 있다. 다양한 실시예들은 AR 시스템(예를 들면, 머리-장착)의 맥락에서 아래에서 설명될 것이지만, 본원에서 개시된 실시예들은 임의의 기존의 및/또는 알려진 AR 시스템들과 독립적으로 사용될 수 있다는 것이 인지되어야 한다.

[0075]

[0073] 이제 도 6a 내지 도 6d를 참조하면, 일부 일반적인 AR 시스템 컴포넌트 옵션들이 다양한 실시예들에 따라 예시된다. 도 6a 내지 도 6d의 실시예들은 머리-장착 디스플레이들을 예시하지만, 동일한 컴포넌트들이 정적인 컴퓨터 시스템들에 또한 통합될 수 있고 도 6a 내지 도 6d는 제한하는 것으로 간주되어서는 안 된다는 것이 인지되어야 한다.

[0076]

[0074] 도 6a에 도시된 바와 같이, 머리-장착 디바이스 사용자(60)는 사용자(60)의 눈들 앞에 포지셔닝된 디스플레이 시스템(62)에 커플링된 프레임(64) 구조를 착용한 것으로 도시된다. 프레임(64)은 필요한 보안 레벨에 의존하여 하나 또는 그 초과 사용자 식별 특정 서브 시스템들에 영구적으로 또는 일시적으로 커플링될 수 있다. 스피커(66)는 도시된 구성에서 프레임(64)에 커플링될 수 있고 사용자(60)의 외이도에 인접하게 포지셔닝될 수 있다. 대안적인 실시예에서, 다른 스피커(도시되지 않음)는 사용자(60)의 다른 외이도에 인접하게 포지셔닝되어 입체 음향/성형 가능한 사운드 제어를 제공한다. 하나 또는 그 초과 실시예들에서, 사용자 식별 디바이스는 이를테면, 유선 리드 또는 무선 연결에 의해, 로컬 데이터 프로세싱 및 데이터 모듈(70)에 동작 가능하게 커플링되는 디스플레이(62)를 가질 수 있으며, 이 로컬 데이터 프로세싱 및 데이터 모듈(70)은 다양한 구성들로 장착될 수 있는데, 이를테면, 프레임(64)에 고정되게 부착되거나, 도 6b에 도시된 실시예에 도시된 바와 같이, 헬멧 또는 모자(80)에 고정되게 부착되거나, 헤드폰들에 임베딩되거나, 도 6c의 실시예에 도시된 바와 같이 백팩-스타일 구성으로 사용자(60)의 몸통(82)에 제거 가능하게 부착되거나, 또는 도 6d의 실시예에 도시된 바와 같이 벨트-커플링 스타일 구성으로 사용자(60)의 엉덩이(84)에 제거 가능하게 부착될 수 있다.

[0077]

[0075] 로컬 프로세싱 및 데이터 모듈(70)은 전력-효율적 프로세서 또는 제어기뿐만 아니라, 디지털 메모리 이를테면, 플래시 메모리를 포함할 수 있으며, 이들 둘 모두는 데이터의 프로세싱, 캐싱(caching) 및 저장을 보조하기 위해 활용될 수 있다. 데이터는 프레임(64)에 동작 가능하게 커플링될 수 있는 센서들 이를테면, 이미지 캡처 디바이스들(이를테면, 카메라들), 마이크로폰들, 관성 측정 유닛들, 가속도계들, 컴퍼스들, GPS 유닛들, 라디오 디바이스들 및/또는 자이로들로부터 캡처될 수 있다. 대안적으로 또는 부가적으로, 데이터는 원격 프로세싱 모듈(72) 및/또는 원격 데이터 리포지토리(74)를 이용하여 취득 및/또는 프로세싱될 수 있다(이는 가능하다면, 이러한 프로세싱 또는 리트리브 후에 디스플레이(62)로의 전달을 위한 것임). 로컬 프로세싱 및 데이터 모듈(70)은 이를테면, 유선 또는 무선 통신 링크들을 통하여, 원격 프로세싱 모듈(72) 및 원격 데이터 리포지토

리(74)에 동작 가능하게 커플링(76, 78)될 수 있어서, 이들 원격 모듈들(72, 74)은 서로 동작 가능하게 커플링되고 로컬 프로세싱 및 데이터 모듈(70)에 대한 자원들로서 이용 가능하다.

[0078] [0076] 일 실시예에서, 원격 프로세싱 모듈(72)은 데이터 및/또는 이미지 정보를 분석 및 프로세싱하도록 구성된 하나 또는 그 초과와 비교적 강력한 프로세서들 또는 제어기들을 포함할 수 있다. 일 실시예에서, 원격 데이터 리포지토리(74)는 "클라우드" 자원 구성에서 인터넷 또는 다른 네트워킹 구성을 통하여 이용 가능할 수 있는 비교적 대형-스케일 디지털 데이터 저장 설비를 포함할 수 있다. 일 실시예에서, 모든 데이터는 저장되고 모든 컴퓨테이션은 로컬 프로세싱 및 데이터 모듈에서 수행되어, 임의의 원격 모듈들로부터 완전히 자율적인 사용을 허용한다.

[0079] [0077] 일부 실시예들에서, 도 6a 내지 도 6d에 설명된 것들과 유사한 식별 디바이스들(또는 식별 애플리케이션들을 갖는 AR 시스템들)은 사용자의 눈들에 대한 고유한 액세스를 제공한다. 식별/AR 디바이스가 사용자의 눈들과 결정적으로(crucially) 상호작용하여 사용자가 3-D 가상 콘텐츠를 지각하도록 허용하고, 다수의 실시예들에서, 사용자의 눈들과 관련된 다양한 바이오메트릭스(biometrics)(예를 들어, 홍채 패턴들, 눈 이접운동, 눈 모션, 추상체들 및 간상체들의 패턴들, 안구 움직임들의 패턴들 등)를 추적한다는 것을 고려하면, 결과적인 추적된 데이터는 식별 애플리케이션들에 유리하게 사용될 수 있다. 따라서, 사용자의 눈들에 대한 이 전례 없는 액세스는 그 자체로 자연스럽게, 다양한 식별 애플리케이션들에 적합하다.

[0080] [0078] 도 7은 본 발명의 일 실시예를 구현하기에 적합한 예시적인 컴퓨팅 시스템(1400)의 블록도이다. 컴퓨터 시스템(1400)은 버스(1406) 또는 정보를 통신하기 위한 다른 통신 메커니즘을 포함하며, 이는 서브시스템들 및 디바이스들, 이를테면, 프로세서(1407), 시스템 메모리(1408)(예를 들어, RAM), 정적 저장 디바이스(1409)(예를 들어, ROM), 디스크 드라이브(1410)(예를 들어, 자기 또는 광학), 통신 인터페이스(1414)(예를 들어, 모뎀 또는 이더넷 카드), 디스플레이(1411)(예를 들어, CRT 또는 LCD), 입력 디바이스(1412)(예를 들어, 키보드), 및 커서 제어를 상호연결한다.

[0081] [0079] 본 발명의 일 실시예에 따라, 컴퓨터 시스템(1400)은, 프로세서(1407)가 시스템 메모리(1408)에 포함된 하나 또는 그 초과와 명령들의 하나 또는 그 초과와 시퀀스들을 실행함으로써 특정 동작들을 수행한다. 이러한 명령들은 정적 저장 디바이스(1409) 또는 디스크 드라이브(1410)와 같은 다른 컴퓨터 판독 가능/사용 가능 매체로부터 시스템 메모리(1408)로 판독될 수 있다. 대안적인 실시예들에서, 하드-와이어드(hard-wired) 회로는, 본 발명을 구현하도록 소프트웨어 명령들을 대신하여 또는 그와 결합하여 사용될 수 있다. 따라서, 본 발명의 실시예들은 하드웨어 회로 및/또는 소프트웨어의 임의의 특정 결합으로 제한되지 않는다. 일 실시예에서, "로직"이란 용어는 본 발명의 전부 또는 일부를 구현하는 데 사용되는, 소프트웨어 또는 하드웨어의 임의의 결합을 의미한다.

[0082] [0080] 본원에서 사용되는 바와 같은 "컴퓨터 판독 가능 매체" 또는 "컴퓨터 사용 가능 매체"라는 용어는 실행을 위해 프로세서(1407)에 명령들을 제공하는데 참여하는 임의의 매체를 지칭한다. 그러한 매체는, 비-휘발성 매체들 및 휘발성 매체들을 포함(그러나 이에 제한되지 않음)하는 다수의 형태들을 취할 수 있다. 비-휘발성 매체들은, 예를 들어, 광학 또는 자기 디스크들, 이를테면, 디스크 드라이브(1410)를 포함한다. 휘발성 매체들은 동적 메모리, 이를테면, 시스템 메모리(1408)를 포함한다.

[0083] [0081] 일반적인 형태들의 컴퓨터 판독 가능 매체들은, 예를 들어, 플로피 디스크, 플렉시블 디스크, 하드 디스크, 자기 테이프, 임의의 다른 자기 매체, CD-ROM, 임의의 다른 광학 매체, 펀치 카드들, 페이퍼 테이프, 홀(hole)들의 패턴들을 갖는 임의의 다른 물리적인 매체, RAM, PROM, EPROM, FLASH-EPROM, 임의의 다른 메모리 칩 또는 카트리지 또는 컴퓨터가 판독할 수 있는 임의의 다른 매체를 포함한다.

[0084] [0082] 본 발명의 실시예에서, 본 발명을 실행하기 위한 명령들의 시퀀스들의 실행은 단일 컴퓨터 시스템(1400)에 의해 수행된다. 본 발명의 다른 실시예들에 따라, 통신 링크(1415)(예를 들어, LAN, PTSN, 또는 무선 네트워크)에 의해 커플링되는 둘 또는 그 초과와 컴퓨터 시스템들(1400)은 서로 협력하여 본 발명을 실시하는데 필요한 명령들의 시퀀스를 수행할 수 있다.

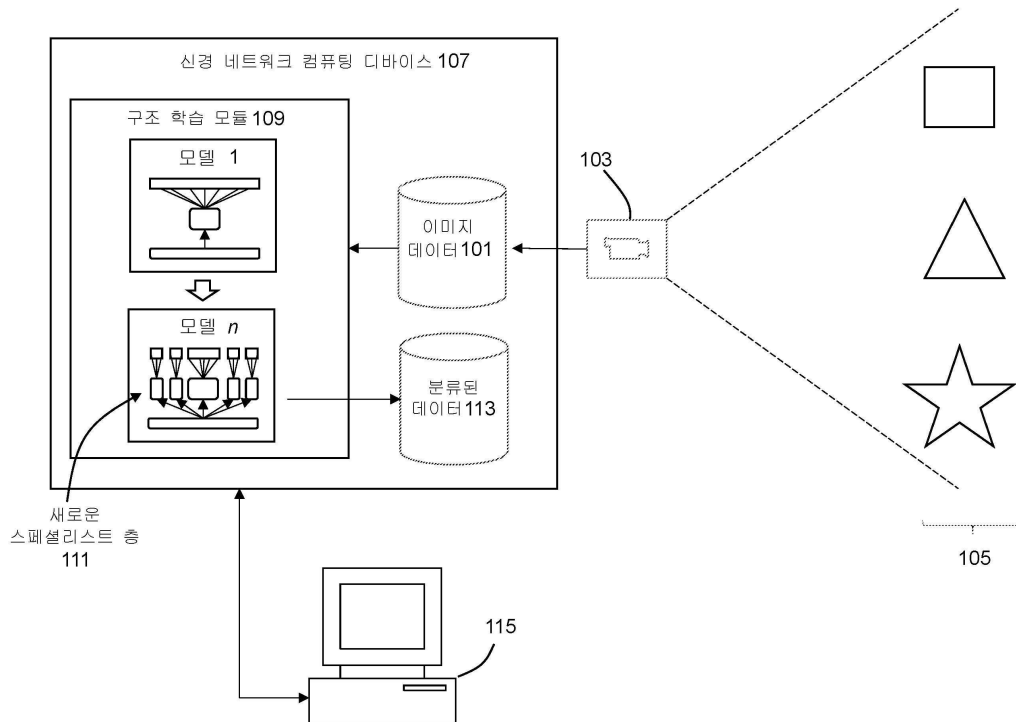
[0085] [0083] 컴퓨터 시스템(1400)은 통신 링크(1415) 및 통신 인터페이스(1414)를 통해 프로그램, 예를 들어, 애플리케이션 코드들을 포함하는 메시지들, 데이터 및 명령들을 송신 및 수신할 수 있다. 수신된 프로그램 코드는, 그것이 수신될 때 프로세서(1407)에 의해 실행될 수 있고 그리고/또는 추후 실행을 위해 디스크 드라이브(1410) 또는 다른 비-휘발성 저장소에 저장될 수 있다. 컴퓨터 시스템(1400)은 데이터 인터페이스(1433)를 통해 외부 저장 디바이스(1431) 상의 데이터베이스(1432)와 통신할 수 있다.

[0086]

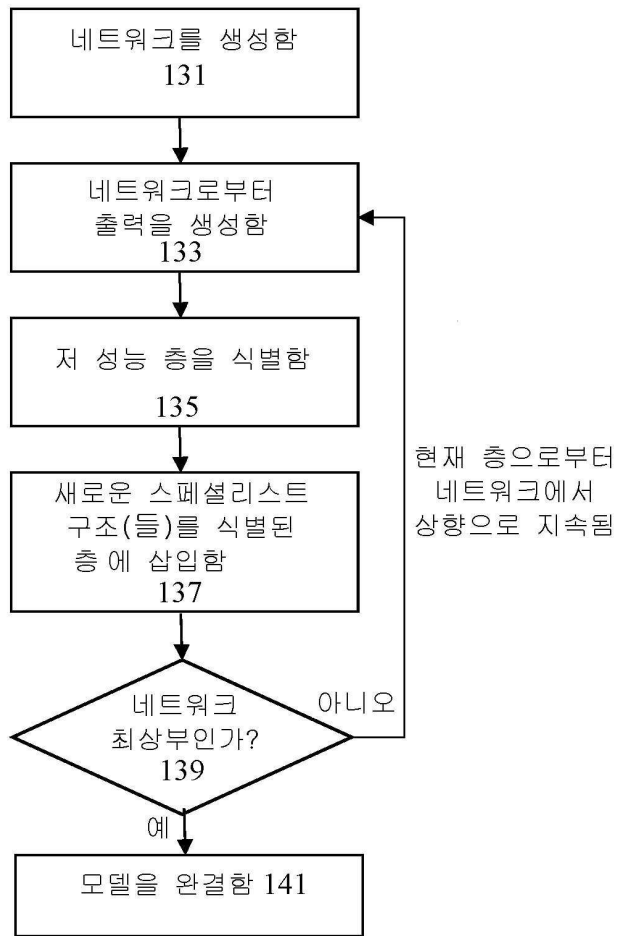
[0084] 위의 명세서에서, 본 발명은 본 발명의 특정 실시예들을 참조하여 설명되었다. 그러나, 본 발명의 더 넓은 사상 및 범위를 벗어나지 않으면서 다양한 수정들 및 변경들이 본 발명에 행해질 수 있다는 것은 명백할 것이다. 예를 들어, 위에서-설명된 프로세스 흐름들은, 프로세스 동작들의 특정한 순서를 참조하여 설명된다. 그러나, 설명된 프로세스 동작들 대부분의 순서는 본 발명의 범위 또는 동작에 영향을 주지 않으면서 변경될 수 있다. 따라서, 명세서 및 도면들은 제한적인 의미보다는 예시적인 의미로 간주될 것이다.

## 도면

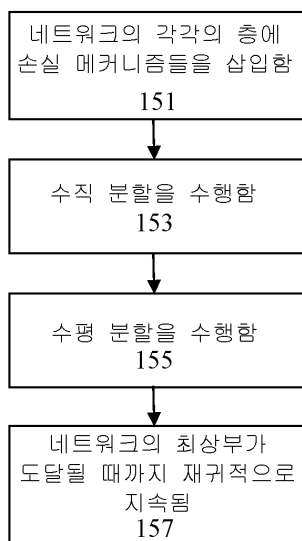
### 도면1



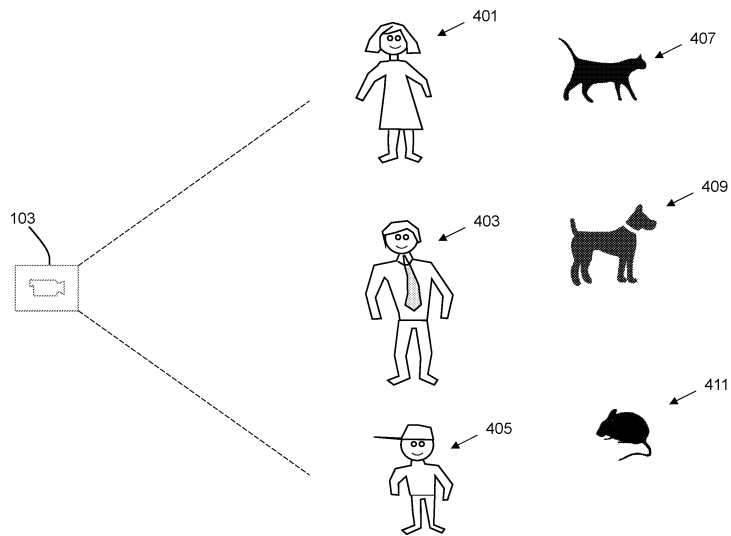
도면2



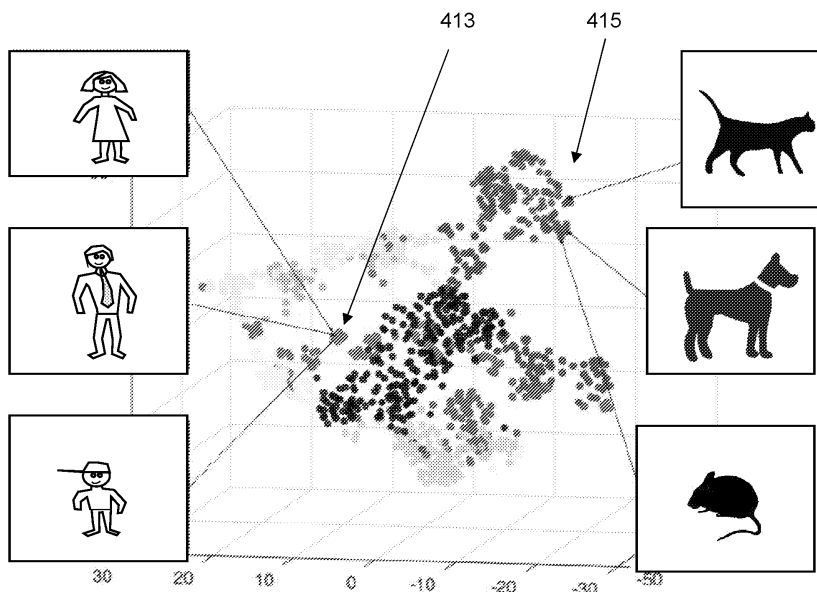
도면3



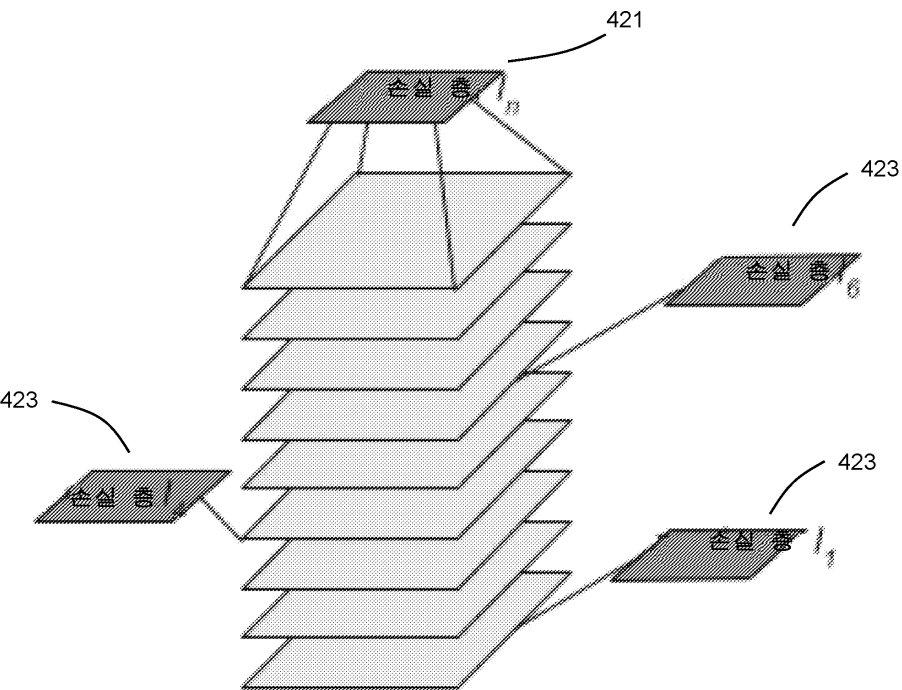
도면4a



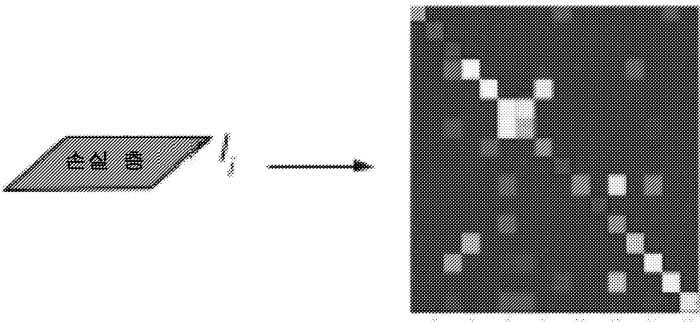
도면4b



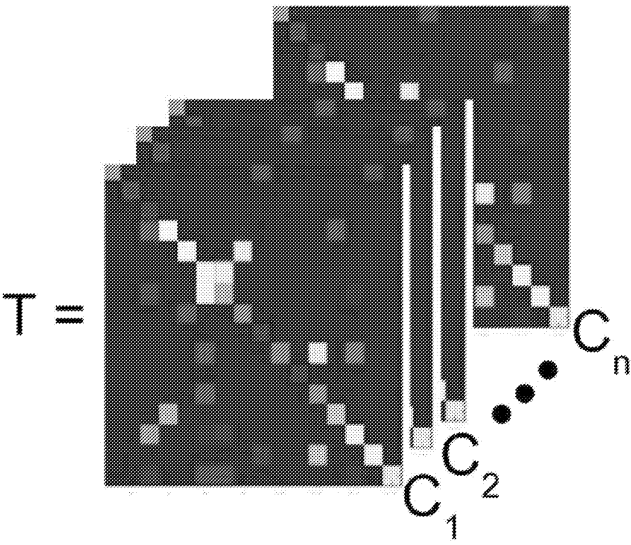
도면4c



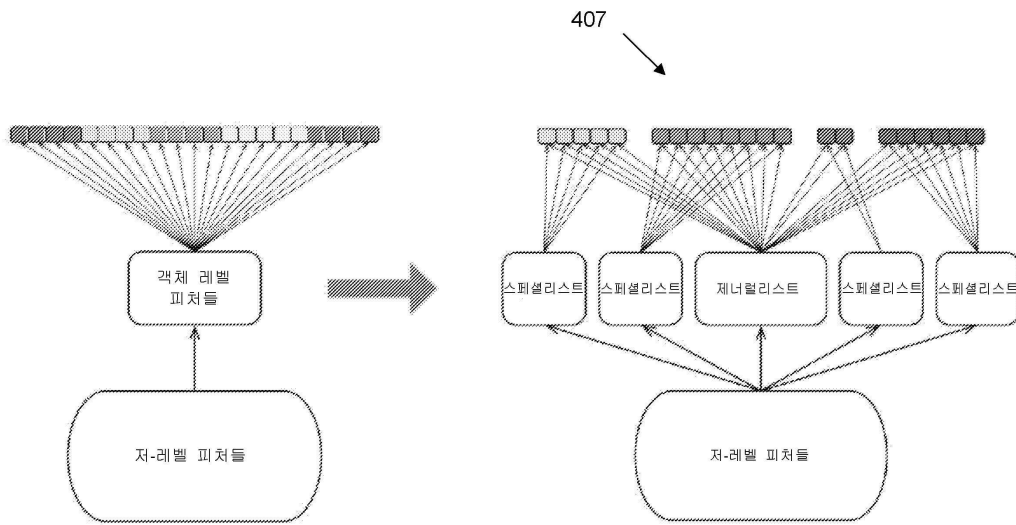
도면4d



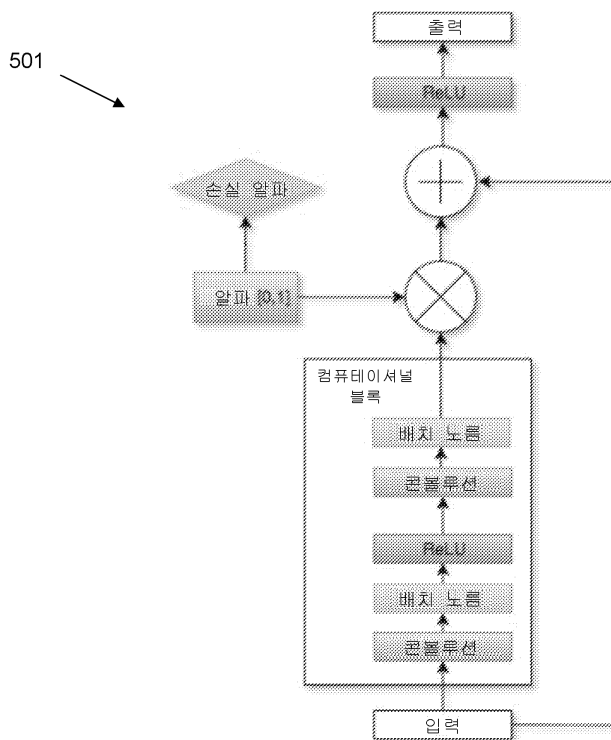
도면4e



도면4f

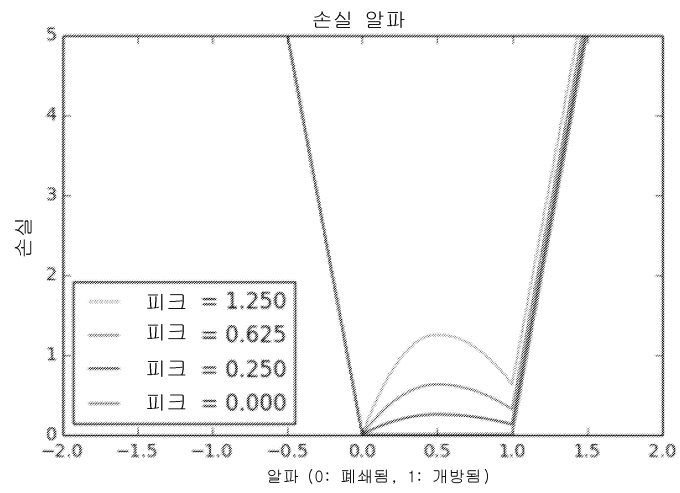


도면5a

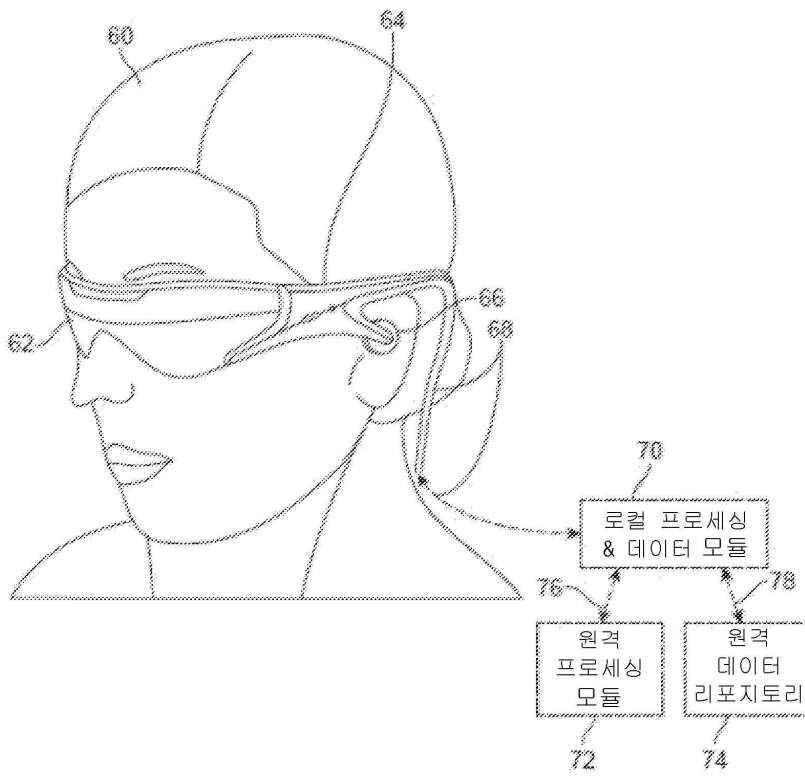




도면5b

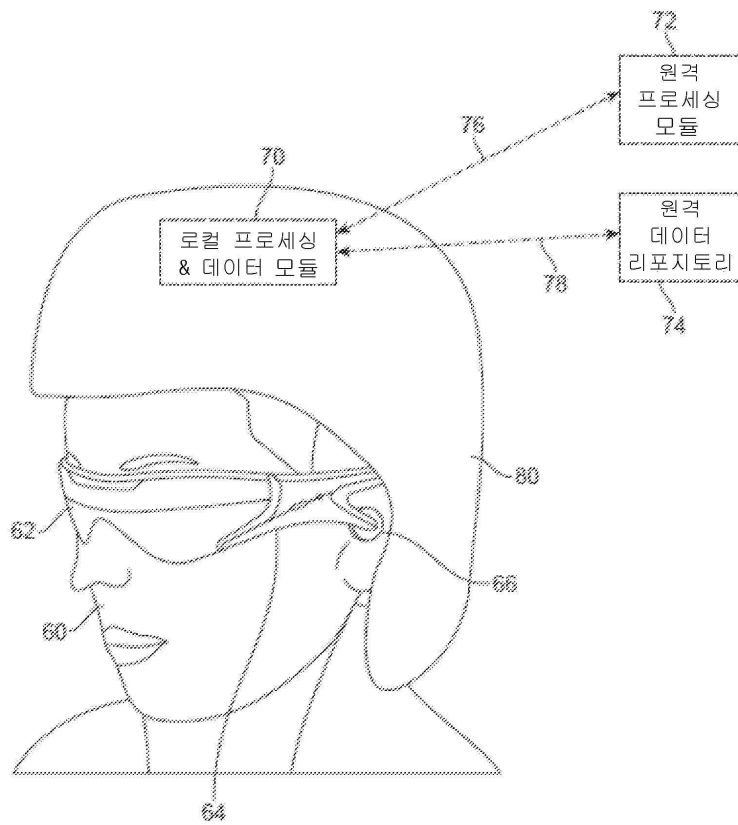


도면6a

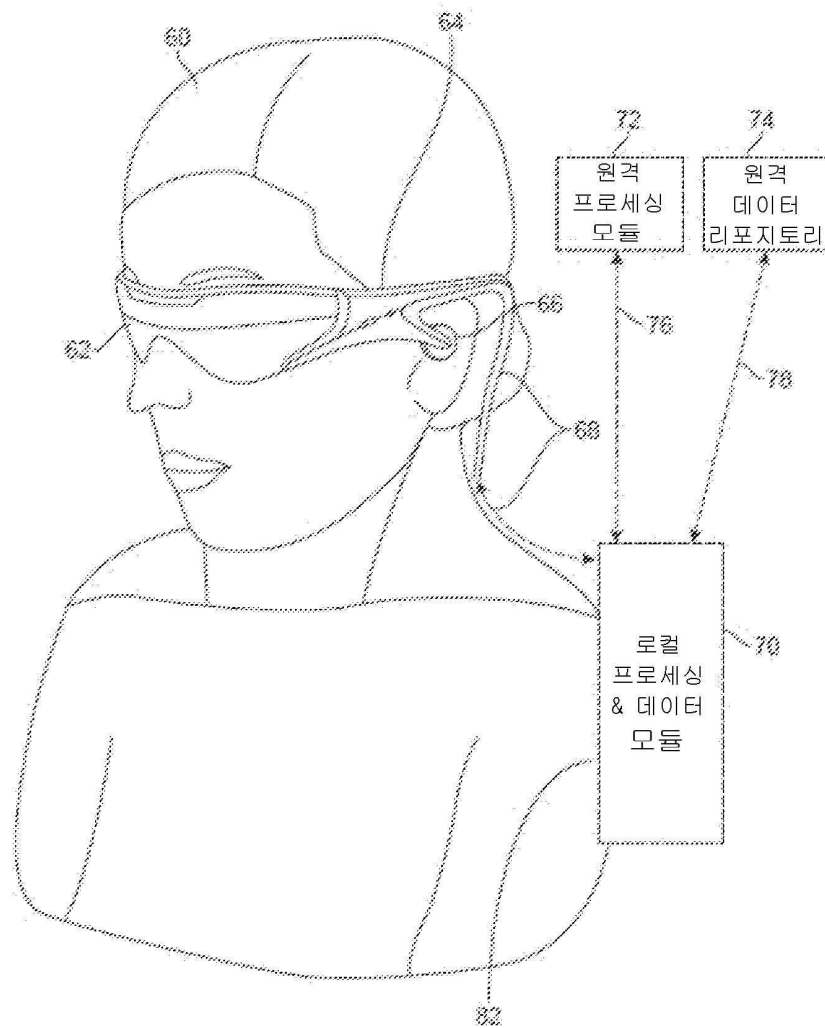




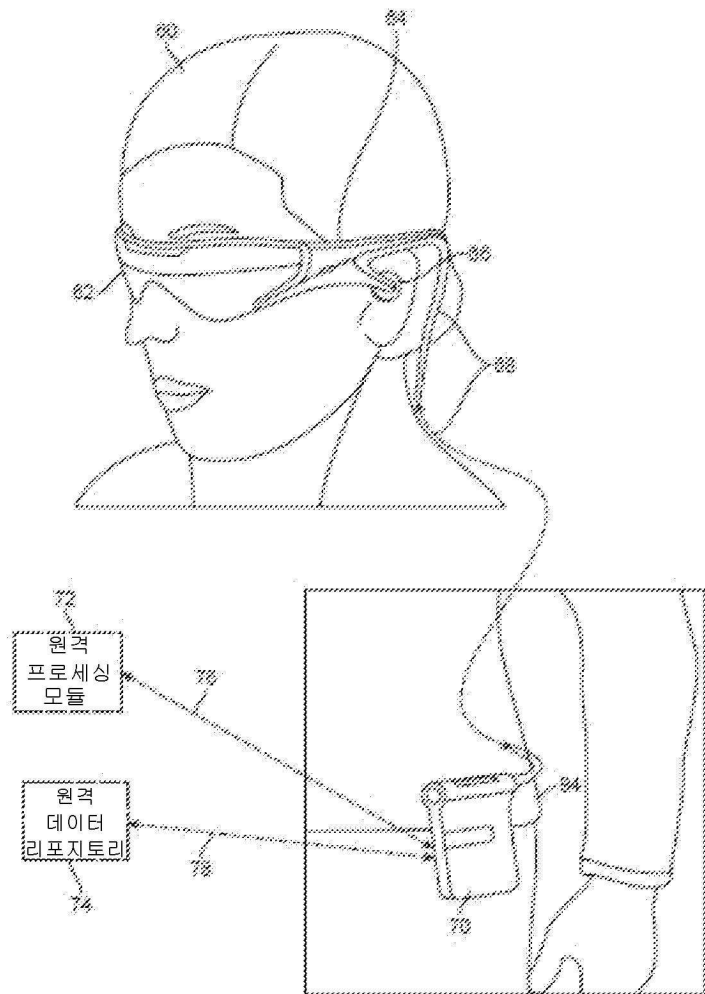
도면6b



도면6c



도면6d



도면7

