



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0149235
(43) 공개일자 2022년11월08일

(51) 국제특허분류(Int. Cl.)
G06N 20/00 (2019.01) G06N 3/04 (2006.01)
G06N 3/08 (2006.01)
(52) CPC특허분류
G06N 20/00 (2021.08)
G06N 3/04 (2013.01)
(21) 출원번호 10-2021-0056593
(22) 출원일자 2021년04월30일
심사청구일자 2021년04월30일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대 학교)
(72) 발명자
최동완
인천광역시 미추홀구 용정공원로 33, 108동 1303호(용현동, 인천 SK Sky VIEW)
최홍준
서울특별시 강서구 등촌로 163, 203동 603호(등촌동, 등촌아이파크아파트)
(74) 대리인
양성보

전체 청구항 수 : 총 7 항

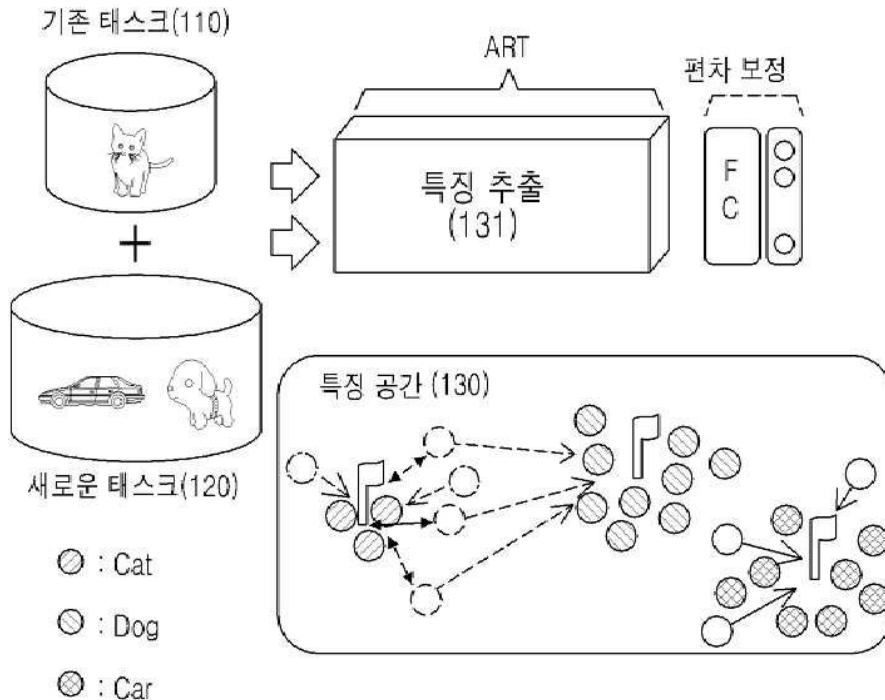
(54) 발명의 명칭 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 방법 및 장치

(57) 요약

클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 방법 및 장치가 제시된다. 본 발명에서 제안하는 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 방법은 특징 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는 클래스들 간의 구분을 위해 기존

(뒷면에 계속)

대표도 - 도1



학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성하는 단계, 새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을 기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특징 공간을 침범하지 않도록 거리를 두고 반발 학습을 수행하는 단계, 새로운 태스크에 포함되는 클래스들 각각의 중간 대푯값을 생성하는 단계 및 새로운 태스크의 클래스들 각각에 대하여 생성된 중간 대푯값을 기준으로, 새로운 태스크에 포함되는 클래스들의 데이터 각각은 자신이 속하는 클래스의 중간 대푯값과는 가까워지고, 자신이 속하지 않은 클래스의 중간 대푯값과는 멀어지도록 흡인 학습을 수행하는 단계를 포함한다.

(52) CPC특허분류

G06N 3/08 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1345318077
과제번호	2018R1D1A1B07049934
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	기본연구(교육부)
연구과제명	[Ezbaro] 미래형 무인 자동화 교통 서비스 실현을 위한 실시간 이동 데이터 분석 및
예측 기술	
기 여 율	30/100
과제수행기관명	인하대학교
연구기간	2021.03.01 ~ 2021.05.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126439
과제번호	2019-0-00240-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	방송통신산업기술개발사업
연구과제명	[Ezbaro] 자율협력형 스마트 임베디드 기기에서의 실시간 추론을 위한 딥러닝 모델
병합 및 분할 플랫폼 개발(3차년도)	
기 여 율	40/100
과제수행기관명	인하대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126440
과제번호	2019-0-00064-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	방송통신산업기술개발사업
연구과제명	[Ezbaro] 커넥티드 카를 위한 지능형 모바일 엣지 클라우드 솔루션 개발(3차년도)
기 여 율	20/100
과제수행기관명	인하대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126259
과제번호	2020-0-01389-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	인공지능융합연구센터지원사업(국고)
연구과제명	[Ezbaro][정부] 인공지능융합연구센터지원(2차년도)
기 여 율	10/100
과제수행기관명	인하대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

명세서

청구범위

청구항 1

데이터 불균형을 완화하기 위한 클래스 점진적 학습 방법에 있어서,

특정 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는 클래스들 간의 구분을 위해 기존 학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성하는 단계;

새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을 기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특정 공간을 침범하지 않도록 거리를 두고 반발 학습을 수행하는 단계;

새로운 태스크에 포함되는 클래스들 각각의 중간 대푯값을 생성하는 단계; 및

새로운 태스크의 클래스들 각각에 대하여 생성된 중간 대푯값을 기준으로, 새로운 태스크에 포함되는 클래스들의 데이터 각각은 자신이 속하는 클래스의 중간 대푯값과는 가까워지고, 자신이 속하지 않은 클래스의 중간 대푯값과는 멀어지도록 흡인 학습을 수행하는 단계

를 포함하는 데이터 불균형을 완화하기 위한 클래스 점진적 학습 방법.

청구항 2

제1항에 있어서,

상기 특정 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는 클래스들 간의 구분을 위해 기존 학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성하는 단계는,

기존 학습된 태스크에 포함되는 클래스들 각각에 대해 손실함수를 이용하여 특정 공간에서의 평균 값을 해당 클래스의 대푯값으로 생성하고 저장하는

데이터 불균형을 완화하기 위한 클래스 점진적 학습 방법.

청구항 3

제1항에 있어서,

상기 새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을 기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특정 공간을 침범하지 않도록 거리를 두고 반발 학습을 수행하는 단계는,

지식증류 손실 함수 및 크로스 엔트로피 함수를 포함하는 최종 손실함수를 이용하여 미리 정해진 최소한의 거리만큼 거리를 두고 반발 학습을 수행하는

데이터 불균형을 완화하기 위한 클래스 점진적 학습 방법.

청구항 4

제1항에 있어서,

상기 새로운 태스크에 포함되는 클래스들 각각의 중간 대푯값을 생성하는 단계는,

새로운 태스크에 포함되는 클래스들의 데이터가 자신이 속하는 클래스와는 가까워지도록 하고, 자신이 속하지 않은 클래스와는 멀어지도록 하는 중간 대푯값을 생성하고 저장하는

데이터 불균형을 완화하기 위한 클래스 점진적 학습 방법.

청구항 5

데이터 불균형을 완화하기 위한 클래스 점진적 학습 장치에 있어서,

특정 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는 클래스들 간의 구분을 위해 기존 학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성하고, 새로운 태스크에 포함되는 클래스들 각각의 중간 대푯값을 생성하는 대푯값 생성부;

새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을 기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특정 공간을 침범하지 않도록 거리를 두고 반발 학습을 수행하는 반발 학습 수행부; 및

새로운 태스크의 클래스들 각각에 대하여 생성된 중간 대푯값을 기준으로, 새로운 태스크에 포함되는 클래스들의 데이터 각각은 자신이 속하는 클래스의 중간 대푯값과는 가까워지고, 자신이 속하지 않은 클래스의 중간 대푯값과는 멀어지도록 흡인 학습을 수행하는 흡인 및 반발 학습 수행부

를 포함하는 데이터 불균형을 완화하기 위한 클래스 점진적 학습 장치.

청구항 6

제5항에 있어서,

상기 대푯값 생성부는,

기존 학습된 태스크에 포함되는 클래스들 각각에 대해 손실함수를 이용하여 특정 공간에서의 평균 값을 해당 클래스의 대푯값으로 생성하고 저장하고,

새로운 태스크에 포함되는 클래스들의 데이터가 자신이 속하는 클래스와는 가까워지도록 하고, 자신이 속하지 않은 클래스와는 멀어지도록 하는 중간 대푯값을 생성하고 저장하는

데이터 불균형을 완화하기 위한 클래스 점진적 학습 장치.

청구항 7

제5항에 있어서,

상기 반발 학습 수행부는,

지식증류 손실 함수 및 크로스 엔트로피 함수를 포함하는 최종 손실함수를 이용하여 미리 정해진 최소한의 거리만큼 거리를 두고 반발 학습을 수행하는

데이터 불균형을 완화하기 위한 클래스 점진적 학습 장치.

발명의 설명

기술 분야

[0001] 본 발명은 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 방법 및 장치에 관한 것이다.

배경 기술

[0002] 신경망 모델의 지속학습[1] 분야는 활발히 연구가 이루어지고 있는 분야 중 하나이다. 지속학습은 일반적인 신경망 모델이 학습할 모든 태스크에 대해 한 번에 학습하는 것과 다르게 순차적으로 들어오는 태스크를 학습하여 과거 태스크에 대한 정보 손실 없이 학습한 모든 태스크에 대해 잘 수행해 낼 수 있는 모델을 학습하는 것을 목표로 한다. 신경망 모델의 지속학습은 크게 태스크 점진적 학습(task incremental learning)과 클래스 점진적 학습(class incremental learning)[2]으로 나뉜다. 태스크 점진적 학습은 순차적으로 학습하는 태스크 각각에 대해 별개의 독립적인 분류기(classifier)를 학습하여 추론 시 태스크 정보를 알고 있다는 가정을 하는 반면 클래스 점진적 학습에서는 학습하는 태스크 모두에 대해 하나의 통합된 분류기를 학습하여 학습 시 태스크들 간의 구분도 할 수 있는 학습이 필요하다.

[0003] 신경망 모델의 지속학습에서 기존에 학습했던 과거 태스크의 정보 손실(catastrophic forgetting)[3]을 완화하기 위한 방법 중 하나로 종래기술[6]에서 처음 제안된 지식 증류 기법(knowledge distillation) 방법과 리허설

방법(rehearsal method)[4][5] 등이 같이 사용되는 방법이 기본으로 사용되고 있다. 먼저 지식 증류 기법은 과거 태스크에 대해 학습된 신경망 모델을 티처(teacher) 모델로 간주하고, 이 티처 모델의 예측 결과값을 이용하여 새로운 태스크를 학습하는 스튜던트(student) 모델이 과거 태스크에 대한 정보를 유지하면서 동시에 새로운 태스크를 학습할 수 있도록 하는 방법이다. 리허설 방법은 과거 태스크 데이터 일부를 보관 후 새로운 태스크 데이터와 같이 학습하는 방법이다. 이 두 가지의 방법을 같이 사용하는 경우 태스크 간의 관계를 학습해야 하는 클래스 점진적 학습에서 기본 방법으로 사용되고 있다. 하지만 리허설 방법의 경우 클래스 불균형이라는 불가피한 문제를 가지고 있다. 이것은 과거 태스크의 모든 데이터를 보관하는 것이 아니기 때문에 과거 태스크에 속하는 클래스의 데이터 숫자와 새로운 태스크에 속하는 클래스의 데이터 숫자가 큰 차이가 나는 문제점이 있다. 따라서 위 클래스 불균형 문제를 해결하고, 클래스 점진적 학습의 성능을 개선하고자 한다.

발명의 내용

해결하려는 과제

[0004] 본 발명이 이루고자 하는 기술적 과제는 특정 공간에서 각각의 클래스 별 데이터들이 자신이 속하는 클래스 대푯값과 가깝게 만들고, 자신이 속하지 않은 클래스의 대푯값으로부터 멀어지도록 학습하는 방식을 제안함으로써 클래스 불균형이 특징 추출기에 미치는 부정적인 영향을 제거하는 방법 및 장치를 제공하는데 있다.

과제의 해결 수단

[0005] 일 측면에 있어서, 본 발명에서 제안하는 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 방법은 특정 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는 클래스들 간의 구분을 위해 기존 학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성하는 단계, 새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을 기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특징 공간을 침범하지 않도록 거리를 두고 반발 학습을 수행하는 단계, 새로운 태스크에 포함되는 클래스들 각각의 중간 대푯값을 생성하는 단계 및 새로운 태스크의 클래스들 각각에 대하여 생성된 중간 대푯값을 기준으로, 새로운 태스크에 포함되는 클래스들의 데이터 각각은 자신이 속하는 클래스의 중간 대푯값과는 가까워지고, 자신이 속하지 않은 클래스의 중간 대푯값과는 멀어지도록 흡인 학습을 수행하는 단계를 포함한다.

[0006] 상기 특정 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는 클래스들 간의 구분을 위해 기존 학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성하는 단계는 기존 학습된 태스크에 포함되는 클래스들 각각에 대해 손실함수를 이용하여 특정 공간에서의 평균 값을 해당 클래스의 대푯값으로 생성하고 저장한다.

[0007] 상기 새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을 기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특징 공간을 침범하지 않도록 거리를 두고 반발 학습을 수행하는 단계는 지식증류 손실 함수 및 크로스 엔트로피 함수를 포함하는 최종 손실함수를 이용하여 미리 정해진 최소한의 거리만큼 거리를 두고 반발 학습을 수행한다.

[0008] 상기 새로운 태스크에 포함되는 클래스들 각각의 중간 대푯값을 생성하는 단계는 새로운 태스크에 포함되는 클래스들의 데이터가 자신이 속하는 클래스와는 가까워지도록 하고, 자신이 속하지 않은 클래스와는 멀어지도록 하는 중간 대푯값을 생성하고 저장한다.

[0009] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 장치는 특정 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는 클래스들 간의 구분을 위해 기존 학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성하고, 새로운 태스크에 포함되는 클래스들 각각의 중간 대푯값을 생성하는 대푯값 생성부, 새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을 기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특징 공간을 침범하지 않도록 거리를 두고 반발 학습을 수행하는 반발 학습 수행부 및 새로운 태스크의 클래스들 각각에 대하여 생성된 중간 대푯값을 기준으로, 새로운 태스크에 포함되는 클래스들의 데이터 각각은 자신이 속하는 클래스의 중간 대푯값과는 가까워지고, 자신이 속하지 않은 클래스의 중간 대푯값과는 멀어지도록 흡인 학습을 수행하는 흡인 및 반발 학습 수행부를 포함한다.

발명의 효과

[0010] 본 발명의 실시예들에 따르면 특징 추출기의 출력 공간인 특징 공간(feature space)에서 클래스 간 유사도가 높아져 구분이 힘들어지는 현상으로 인한 클래스 불균형을 감소시킬 수 있다. 다시 말해, 특징 공간에서 각각의 클래스 별 데이터들이 자신이 속하는 클래스 대푯값과 가깝게 만들고, 자신이 속하지 않은 클래스의 대푯값으로부터 멀어지도록 학습하는 방식을 제안함으로써 클래스 불균형이 특징 추출기에 미치는 부정적인 영향을 제거할 수 있다.

도면의 간단한 설명

[0011] 도 1은 본 발명의 일 실시예에 따른 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 과정을 설명하기 위한 개략도이다.

도 2는 본 발명의 일 실시예에 따른 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 방법을 설명하기 위한 흐름도이다.

도 3은 본 발명의 일 실시예에 따른 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 장치의 구성을 나타내는 도면이다.

도 4는 본 발명의 일 실시예에 따른 클래스 점진적 학습에서 각 단계마다의 정확도를 나타내는 그래프이다.

도 5는 본 발명의 일 실시예에 따른 특징 공간에서 클래스 간의 상관관계를 확인하기 위한 클래스 별 코사인 유사도를 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 발명에서 제안하고 있는 방법은 클래스 점진적 학습에 대한 문제를 다루고 있다. 본 발명은 심층 신경망 모델(deep neural network)이 지속학습(incremental learning)하는 상황에서 나타나는 클래스 불균형 문제를 해결하기 위해 특징 공간(feature space)에서 이전에 학습한 클래스와 새롭게 학습하고자 하는 클래스간 유사도를 낮춤으로써 효과적으로 클래스 불균형 문제를 극복하는 학습 방법을 제안한다. 본 발명은 클래스간 유사도를 낮추기 위해 클래스 별 대푯값을 생성하고, 클래스 내 각각의 데이터가 다른 클래스의 대푯값으로부터 멀어지고 자신이 속한 클래스의 대푯값으로부터 가깝게 위치하게 만드는 학습방법을 사용한다. 본 발명에서 제안하는 학습방법을 통해 지속학습 성능 평가를 실시한 결과, 특징 공간에서 이전에 학습한 데이터와 새롭게 학습한 데이터들의 클래스 간 유사도가 낮아져 더욱 효과적인 구분이 가능한 것을 확인할 수 있고, 전체 성능 또한 향상된 것을 확인할 수 있다.

[0013] 본 발명의 실시예에 따른 용어 '지속학습(continual learning)'은 과거 태스크 데이터의 보관 없이 순차적으로 들어오는 태스크에 대해서 학습하여, 학습한 모든 태스크에 대한 추론을 효과적으로 해낼 수 있는 것을 목표로 하는 학습을 의미한다.

[0014] 본 발명의 실시예에 따른 용어 '클래스 점진적 학습(class incremental learning)'은 일정한 수의 클래스를 태스크 단위로 하여, 모든 태스크에 대해 통합된 신경망 분류기(unified classifier)를 학습하는 지속학습 상황을 의미하고, 여기서 각각의 태스크 간의 관계를 학습해야 한다.

[0015] 본 발명의 실시예에 따른 용어 '과거 태스크(old task)'는 지속학습 하는 과정에서, 학습하고 있는 현재 시점 이전에 학습했던 모든 태스크들을 의미한다.

[0016] 본 발명의 실시예에 따른 용어 '새로운 태스크(new task)'는 지속학습 하는 과정에서, 현재 시점에 학습하는 태스크를 의미한다.

[0017] 본 발명의 실시예에 따른 용어 '클래스 불균형(Class Imbalance)'은 클래스 별 데이터 수의 차이가 나는 문제점을 의미한다.

[0018] 본 발명의 실시예에 따른 용어 '특징 공간(feature space)'은 합성곱 신경망(convolution neural network)에서 합성곱 레이어를 통과한 후의 n차원의 공간을 의미한다.

[0019] 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0021] 클래스 점진적 학습에서 클래스 불균형을 해결하고자 했던 종래기술들의 경우 대부분은 전 결합 레이어(fully connected layer)에서의 클래스 불균형 문제를 해결한다. 예를 들어 종래기술[8]은 클래스 불균형이 신경망 모

텔에 끼치는 영향을 전 결합 레이어에서의 각각의 클래스에 해당하는 가중치(weight)의 크기(norm)가 새로운 클래스와 기존 클래스의 차이가 나는 것을 실험을 통해 확인하고 가중치의 차이를 줄이는 방식으로 클래스 불균형 문제를 해결한다. 하지만 클래스 불균형은 전 결합 레이어뿐 아니라 심층 신경망 모델의 특징 추출기(feature extractor) 부분인 합성곱 레이어(convolutional layer)에도 부정적인 영향을 미친다. 특히 특징 추출기의 출력 공간인 특징 공간(feature space)에서 클래스간 유사도가 굉장히 높아져 구분이 힘들어지는 현상이 일어난다. 본 발명은 특징 공간에서 각각의 클래스 별 데이터들이 자신이 속하는 클래스의 대푯값과 가깝게 만들고, 자신이 속하지 않는 클래스의 대푯값으로부터 멀어지도록 학습하는 방식을 제안함으로써 클래스 불균형이 특징 추출기에 미치는 부정적인 영향을 제거하는 방법을 제안하고자 한다.

[0023] 도 1은 본 발명의 일 실시예에 따른 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 과정을 설명하기 위한 개략도이다.

[0024] 본 발명은 특징 공간(130)에서 새로운 태스크(120)에 속하는 클래스들과 기존에 학습한 태스크(110)에 속하는 클래스들의 간의 구분능력을 향상시키기 위한 학습 방법을 제안한다. 먼저, 첫 번째 태스크에 대한 학습이 완료 후, 학습된 기존 태스크(110)의 각 클래스들의 특징 공간(130)에서의 평균 값을 해당 클래스의 대푯값으로 지정한다. 이후, 새로운 태스크(120)가 들어오면 새로운 태스크(120)의 클래스들이 학습된 기존 태스크(110)의 클래스들의 특징 공간을 침범하지 않도록 거리를 두고 학습한다. 특징 공간에서의 새로운 태스크(120)에 포함되는 클래스들과 기존 학습된 태스크(110)에 포함되는 클래스들 간의 구분을 위해 기존 태스크(110) 및 새로운 태스크(120)에 대한 클래스들 각각의 대푯값, 다시 말해 특징을 추출할 수 있다(131). 나아가 어느 정도 거리를 두고 학습이 된 이후에는, 특징 공간(130)에서 다른 클래스와 거리를 두는 학습과 더불어 새로운 태스크(120)의 클래스 내의 데이터가 속하는 해당 클래스 대푯값에 가까워지도록 학습을 한다. 이를 통해 기존에 학습한 클래스들과 새롭게 학습하는 클래스들의 구분을 특징 공간(130)에서 조금 더 쉽게 할 수 있도록 모델은 학습된다.

[0026] 도 2는 본 발명의 일 실시예에 따른 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 방법을 설명하기 위한 흐름도이다.

[0027] 본 발명의 일 실시예에 따른 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 방법은 특징 공간에서 새로운 태스크에 속하는 클래스들과 기존에 학습한 태스크에 속하는 클래스들의 간의 구분능력을 향상시키기 위해 다음과 같은 방법으로 학습하고자 한다. 먼저, 첫 번째 태스크에 대한 학습이 완료 후 각 클래스들의 특징 공간에서의 평균 값을 해당 클래스의 대푯값으로 지정한다. 새로운 태스크가 들어오면 새로운 태스크의 클래스들이 기존 학습한 클래스들의 공간을 침범하지 않도록 거리를 두고 학습한다. 어느 정도 거리를 두고 학습이 된 이후에는, 특징 공간에서 다른 클래스와 거리를 두는 학습과 더불어 이제 자신이 속하는 클래스 대푯값에 가까워지도록 학습을 수행 한다. 이를 통해 기존에 학습한 클래스들과 새롭게 학습하는 클래스들의 구분을 특징 공간에서 조금 더 쉽게 할 수 있도록 모델이 학습된다.

[0028] 제안하는 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 방법은 특징 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는 클래스들 간의 구분을 위해 기존 학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성하는 단계(210), 새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을 기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특징 공간을 침범하지 않도록 거리를 두고 반발 학습을 수행하는 단계(220), 새로운 태스크에 포함되는 클래스들 각각의 중간 대푯값을 생성하는 단계(230) 및 새로운 태스크의 클래스들 각각에 대하여 생성된 중간 대푯값을 기준으로, 새로운 태스크에 포함되는 클래스들의 데이터 각각은 자신이 속하는 클래스의 중간 대푯값과는 가까워지고, 자신이 속하지 않은 클래스의 중간 대푯값과는 멀어지도록 흡인 학습을 수행하는 단계(240)를 포함한다.

[0029] 단계(210)에서, 특징 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는 클래스들 간의 구분을 위해 기존 학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성한다. 기존 학습된 태스크에 포함되는 클래스들 각각에 대해 손실함수를 이용하여 특징 공간에서의 평균 값을 해당 클래스의 대푯값으로 생성하고 이를 저장한다.

[0030] 더욱 상세하게는, 기존 태스크의 학습이 끝난 후 먼저 기존 학습된 태스크에 포함되는 클래스 별 대푯값을 생성한다. 대푯값은 특징 공간에서 각 클래스를 대표하는 하나의 값을 의미하고, 본 발명의 실시예에서는 평균을 사

용한다. 이 단계는 특징공간에서 데이터들이 다른 클래스들과 거리를 멀어지도록 할 때 클래스의 어떠한 기준으로부터 멀어지는 것을 결정하기 위해 클래스 별 대푯값을 생성하는 단계이다. 학습이 끝난 후 다음과 같은 손실 함수를 이용해 각 클래스 별 대푯값을 생성하고 저장한다:

$$\mathbf{f}_j^{(t)} = \begin{cases} \frac{1}{|C_j^{(t)}|} \sum_{\mathbf{x} \in C_j^{(t)}} \phi_t(\mathbf{x}) & \text{if } j \in T_t \\ \frac{1}{|E_j^{(t)}|} \sum_{\mathbf{x} \in E_j^{(t)}} \phi_t(\mathbf{x}) & \text{otherwise.} \end{cases}$$

여기서, 각 클래스 별 대푯값 $\mathbf{f}_j^{(t)}$ 는 새로운 태스크로 들어오는 시점 t 에서 새로운 태스크의 각 클래스 j 의 대푯값이다. $\phi_t(\mathbf{x})$ 는 데이터 $\mathbf{x}$ 를 t 시점에 학습한 모델에 합성곱 레이어를 통과시킨 후의 벡터이다. 즉, 전 결합 레이어로 입력될 입력 값이다. 새로운 태스크 T_t 의 클래스 들 중 C_t 에 속하는 데이터들과 이전 태스크까지의 속하는 클래스 집합 $E_j^{(t)}$ 에 속하는 클래스들 각각에 대해 평균 값을 구한다. 해당 클래스 별 대푯값들은 추후 학습을 위해 저장한다.

단계(220)에서, 새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을 기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특징 공간을 침범하지 않도록 거리를 두고 반발 학습을 수행한다. 지식중류 손실 함수 및 크로스 엔트로피 함수를 포함하는 최종 손실함수를 이용하여 미리 정해진 최소한의 거리만큼 거리를 두고 반발 학습을 수행한다.

새로운 태스크 데이터 D_t 는 특징 공간에서 기존 학습한 클래스들의 특징 공간을 침범하지 않도록 하기 위해 아래와 같은 수식을 이용해 학습이 된다:

$$\mathcal{L}_{flagh} = \sum_{\mathbf{x} \in D_t} \max(0, \alpha - \|\phi_t(\mathbf{x}) - \mathbf{f}_{j^*}^{(t-1)}\|_2)$$

위 수식에서 새로운 태스크에 속하는 데이터들 D_t 는 이전 시점에 학습이 끝나고 저장했던 플래그 $\mathbf{f}_{j^*}^{(t-1)}$ 로부터 거리가 멀어지도록 학습을 한다. 여기서 최소한의 거리를 α 만큼 벌리고자 한다. 해당 단계에서 사용되는 최종 손실함수는 다음과 같다:

$$\gamma \cdot \mathcal{L}_{flagh} + \mathcal{L}_{kd} + \mathcal{L}_{ce}$$

$\mathcal{L}_{flagh}$ 와 달리 지식중류 손실 함수 $\mathcal{L}_{kd}$ 와 크로스 엔트로피 함수 $\mathcal{L}_{ce}$ 는 이전에 학습한 클래스의 데이터 모두 사용한다. γ 는 하이퍼파라미터로 손실함수의 비율을 설정해주는 값이다.

단계(230)에서, 새로운 태스크에 포함되는 클래스들 각각의 중간 대푯값을 생성한다. 새로운 태스크에 포함되는 클래스들의 데이터가 자신이 속하는 클래스와는 가까워지도록 하고, 자신이 속하지 않은 클래스와는 멀어지도록 하는 중간 대푯값을 생성하고 저장한다.

앞선 손실함수를 통해 일정 에포크(epoch) 동안 학습을 한 후 새로운 태스크에 속하는 클래스들을 위한 대푯값을 생성한다. 새로운 클래스들을 위한 대푯값들을 생성하는 이유는 다른 클래스들과 멀어지는 학습을 하는 것뿐만이 아닌 자신이 속한 클래스들의 데이터들을 한 곳으로 가까워지도록 하는 학습을 위한 새로운 클래스들의 대푯값이 없기 때문이다. 새로운 클래스들의 대푯값을 설정하는 수식은 이전 클래스 별 대푯값을 생성하는 수식과 동일하다.

단계(240)에서, 새로운 태스크의 클래스들 각각에 대하여 생성된 중간 대푯값을 기준으로, 새로운 태스크에 포함되는 클래스들의 데이터 각각은 자신이 속하는 클래스의 중간 대푯값과는 가까워지고, 자신이 속하지 않은 클래스의 중간 대푯값과는 멀어지도록 흡인 학습을 수행한다.

중간 대푯값을 생성한 후 아래 수식을 통해 각각의 데이터들이 자신이 속한 클래스의 대푯값으로부터 거리를 가깝게 하고 자신이 속하지 않은 클래스의 대푯값으로부터 거리가 멀어지게 학습한다:

$$\begin{aligned}\mathcal{L}_{flag_t} = & \sum_{\mathbf{x} \in \mathcal{D}_t} \max(0, \alpha - \|\phi(\mathbf{x}) - \mathbf{f}_{j^*}^{(t-1)}\|_2 \\ & + \|\phi(\mathbf{x}) - \hat{\mathbf{f}}_{y(\mathbf{x})}^{(t)}\|_2) \\ & + \sum_{\mathbf{x}' \in \mathcal{M}_t} \max(0, \alpha - \|\phi(\mathbf{x}') - \hat{\mathbf{f}}_{k^*}^{(t)}\|_2 \\ & + \|\phi(\mathbf{x}') - \mathbf{f}_{y(\mathbf{x}')}^{(t-1)}\|_2),\end{aligned}$$

여기서, $\|\phi(\mathbf{x}) - \mathbf{f}_{j^*}^{(t-1)}\|_2$ 는 특징 공간에서 입력 데이터 $\mathbf{x}$ 에 대한 특징 벡터와 다른 클래스의 대푯값 $\mathbf{f}_{j^*}^{(t-1)}$ 로부
터의 거리를 의미한다. $\|\phi(\mathbf{x}) - \mathbf{f}_{j^*}^{(t)}\|_2$ 는 입력 데이터 $\mathbf{x}$ 에 대한 특징 벡터와 자신이 속한 대푯값 $\mathbf{f}_{j^*}^{(t)}$ 과의 거리
를 의미한다. 즉, $\|\phi(\mathbf{x}) - \mathbf{f}_{j^*}^{(t-1)}\|_2$ 의 거리가 $\|\phi(\mathbf{x}) - \mathbf{f}_{j^*}^{(t)}\|_2$ 보다 크게 학습을 함으로써 각각의 데이터는 자신이
속한 클래스 대푯값으로 가까워지는 동시에, 자신이 속하지 않은 클래스의 대푯값으로부터 구분되도록 위치할
수 있다.

또한, 본 발명에서는 입력 데이터에 따라 거리를 두고자 하는 대푯값의 클래스들을 다르게 설정할 수 있다. 예
를 들어, 입력 데이터가 새로운 태스크에 속하는 경우, 해당 입력 데이터가 거리를 두고자 하는 대푯값은 기존
학습한 태스크에 속하는 클래스 대푯값을 선택할 수 있다. 이와 반대로 입력 데이터가 기존에 학습한 태스크에
속하는 경우, 거리를 두고자 하는 대푯값은 새로운 클래스들의 대푯값 중 하나를 선택할 수 있다. 이를 통해 새
로운 클래스들과 기존에 학습한 클래스들의 구분을 특징 공간에서 용이하게 할 수 있고 궁극적으로는 점진적 학
습에서의 클래스 불균형 문제를 해결할 수 있다. 최종 수식은 아래와 같다:

$$\gamma \cdot \mathcal{L}_{flag_t} + \mathcal{L}_{kd} + \mathcal{L}_{ce}$$

도 3은 본 발명의 일 실시예에 따른 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식
학습 장치의 구성을 나타내는 도면이다.

제안하는 클래스 점진적 학습에서의 데이터 불균형을 완화하기 위한 흡인 및 반발식 학습 장치(300)는 대푯값
생성부(310), 반발 학습 수행부(320) 및 흡인 및 반발 학습 수행부(330)를 포함한다.

대푯값 생성부(310), 반발 학습 수행부(320) 및 흡인 및 반발 학습 수행부(330)는 도 2의 단계들(210~240)을 수
행하기 위해 구성될 수 있다.

대푯값 생성부(310)는 특징 공간에서의 새로운 태스크에 포함되는 클래스들과 기존 학습된 태스크에 포함되는
클래스들 간의 구분을 위해 기존 학습된 태스크에 포함되는 클래스들 각각의 대푯값을 생성하고, 새로운 태스크
에 포함되는 클래스들 각각의 중간 대푯값을 생성한다. 기존 학습된 태스크에 포함되는 클래스들 각각에 대해
손실함수를 이용하여 특징 공간에서의 평균 값을 해당 클래스의 대푯값으로 생성하고 이를 저장한다. 또한, 새
로운 태스크에 포함되는 클래스들의 데이터가 자신이 속하는 클래스와는 가까워지도록 하고, 자신이 속하지 않
은 클래스와는 멀어지도록 하는 중간 대푯값을 생성하고 저장한다.

반발 학습 수행부(320)는 새로운 태스크가 입력되면, 기존 태스크의 클래스들 각각에 대하여 생성된 대푯값을
기준으로 새로운 태스크에 포함되는 클래스들이 기존 태스크의 클래스들의 특징 공간을 침범하지 않도록 거리를
두고 반발 학습을 수행한다. 지식증류 손실 함수 및 크로스 엔트로피 함수를 포함하는 최종 손실함수를 이용하
여 미리 정해진 최소한의 거리만큼 거리를 두고 반발 학습을 수행한다.

흡인 및 반발 학습 수행부(330)는 새로운 태스크의 클래스들 각각에 대하여 생성된 중간 대푯값을 기준으로, 새
로운 태스크에 포함되는 클래스들의 데이터 각각은 자신이 속하는 클래스의 중간 대푯값과는 가까워지고, 자신
이 속하지 않은 클래스의 중간 대푯값과는 멀어지도록 흡인 학습을 수행한다.

[0054] 표 1은 본 발명에서 제안하는 흡인 및 반발학습을 이용한 지속학습 방법을 두 가지 데이터 셋에 대한 클래스 점진적 학습 문제에 대한 성능을 비교한 결과이다.

Methods	CIFAR-100				Tiny-ImageNet			
	5task (20class)	10task (10class)	20task (5class)	50task (2class)	5task (40class)	10task (20class)	20task (10class)	50task (4class)
rKD	57.73	57.00	55.63	59.08	43.74	44.07	42.55	41.75
iCaRL	57.63	57.60	55.68	58.87	44.23	43.98	42.83	41.12
EEIL	63.33	62.33	57.40	59.82	46.53	46.75	45.85	43.40
BiC	65.19	61.59	57.40	57.28	49.68	46.38	41.66	32.68
WA	65.26	63.91	61.83	60.77	49.73	49.28	45.19	38.13
CTL + WA	58.07	57.07	55.12	55.07	41.91	45.75	45.31	41.26
ART + WA (ours)	65.72	65.80	63.29	62.52	49.83	49.37	47.87	45.09

[0055]

클래스 점진적 학습에 대한 성능은 전체 데이터 셋을 일정한 수의 클래스로 구성된 태스크 단위로 나누고 이 태스크들을 순차적으로 학습하며 측정한 평균 정확도의 결과이다. 실험 결과 본 발명에서 제안한 흡인 및 반발학습이 기존 클래스 점진적 학습에서 제시된 최신 방법들과 비교해 우수한 성능을 보이는 것을 확인할 수 있다.

[0058] 도 4는 본 발명의 일 실시예에 따른 클래스 점진적 학습에서 각 단계마다의 정확도를 나타내는 그래프이다.

[0059] 도 4를 참조하면, 평균 정확도뿐만 아닌 매 단계에서의 정확도가 종래기술보다 높은 정확도를 가지는 것을 확인할 수 있다.

[0061] 도 5는 본 발명의 일 실시예에 따른 특징 공간에서 클래스 간의 상관관계를 확인하기 위한 클래스 별 코사인 유사도를 나타내는 도면이다.

[0062] 도 5를 참조하면, 각 블록의 색깔은 코사인 유사도의 정도를 나타내고, 붉은 색을 나타낼 수록 두 클래스의 코사인 유사도가 높은 것을 의미한다. 이는 특징 공간에서 유사한 위치에 있다는 것을 의미한다. 반대로 푸른 색을 나타낼 수록 각 클래스들의 코사인 유사도가 낮은 것을 의미한다. 도 5(a)는 종래기술[8]에서 제시한 학습방법을 이용한 후 측정한 클래스간 유사도이다. 도 5(b)는 본 발명에서 제안하는 흡인 및 반발 학습방법을 사용한 후의 클래스간 유사도이다. 따라서 이 실험 결과를 통해 실제로 본 발명에서 제시한 흡인 및 반발 학습 방법이 특징 공간에서 클래스간 유사도를 낮추고 클래스 불균형 해소에 큰 도움이 되는 것을 볼 수 있다.

[0064] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0065] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0066] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히

설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0067] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[0068] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

[0070] <참고문헌>

[0071] [1] Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* 113: 54-71.

[0072] [2] van de Ven, G. M.; and Tolias, A. S. 2019. Three scenarios for continual learning. *CoRR* abs/1904.07734.

[0073] [3] McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109-165. Elsevier

[0074] [4] Lopez-Paz, D.; and Ranzato, M. 2017. Gradient Episodic Memory for Continual Learning. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 4-9 December 2017, Long Beach, CA, USA, 6467-6476.

[0075] [5] Rebuffi, S.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, July 21-26, 2017, 5533-5542. IEEE Computer Society.

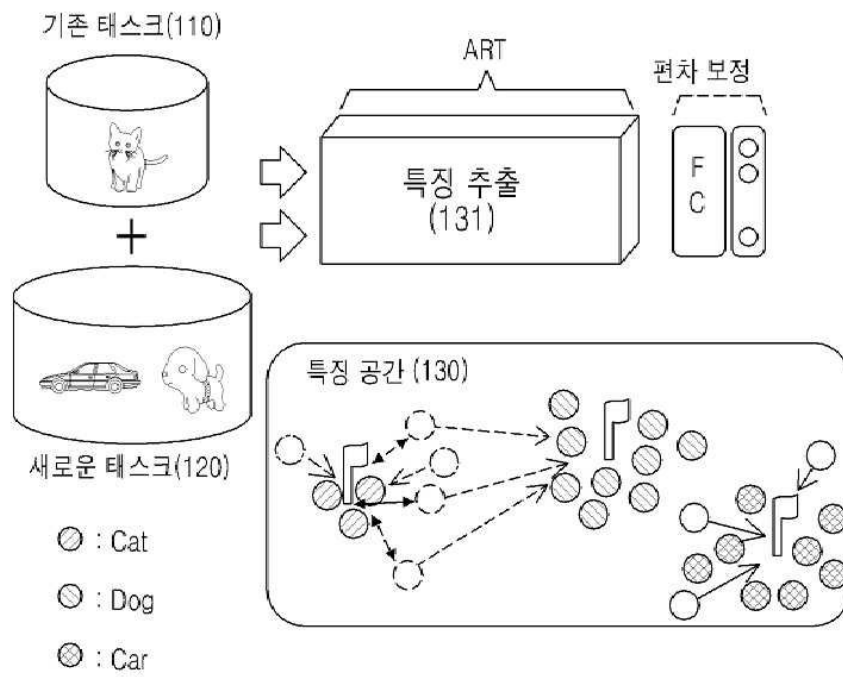
[0076] [6] Li, Z.; and Hoiem, D. 2018. Learning without Forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* 40(12): 2935-2947.

[0077] [7] Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR* abs/1503.02531.

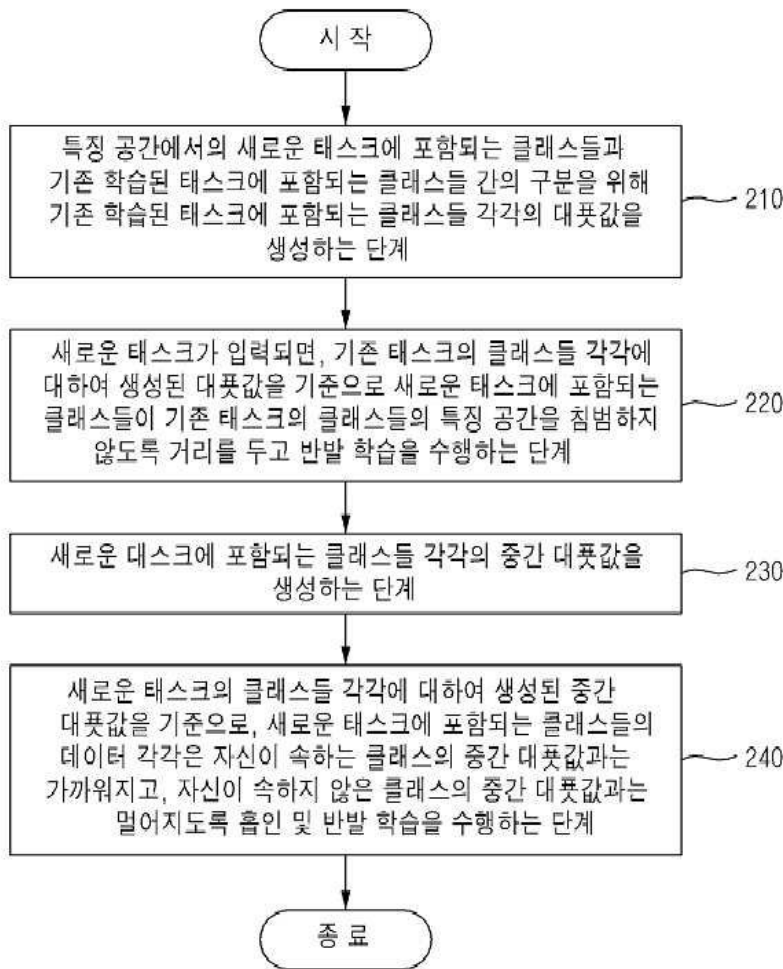
[0078] [8] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-TaoXia. Maintaining discrimination and fairness in class incremental learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, Seattle, WA, USA, June 13-19, 2020, pages 13205-13214. IEEE, 2020

도면

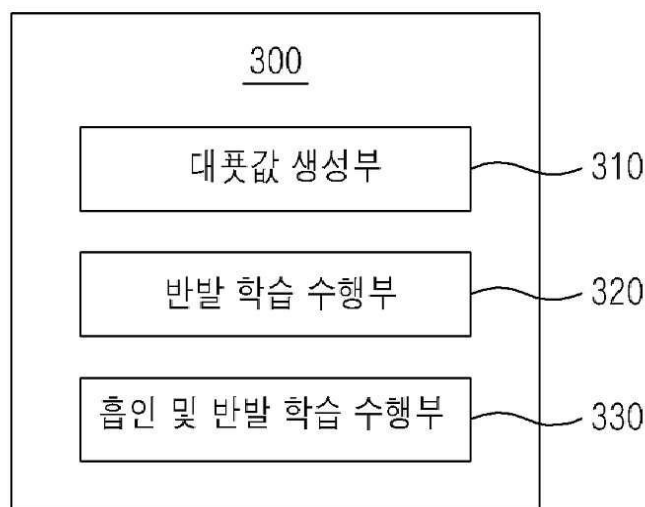
도면1



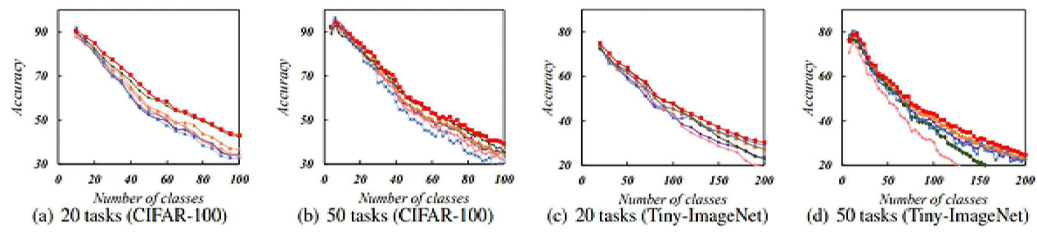
도면2



도면3



도면4



도면5

