



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0110894
(43) 공개일자 2023년07월25일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2023.01) G06N 3/04 (2023.01)
(52) CPC특허분류
G06N 3/082 (2023.01)
G06N 3/045 (2023.01)
(21) 출원번호 10-2022-0006397
(22) 출원일자 2022년01월17일
심사청구일자 2022년01월17일

(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)
이건호
인천광역시 남동구 독곡로16번길 22-19, 304호(서창동, 하은빌라)
김성웅
서울특별시 강남구 삼성로 212, 22동 1209호(대치동, 은마아파트)
(74) 대리인
양성보

전체 청구항 수 : 총 8 항

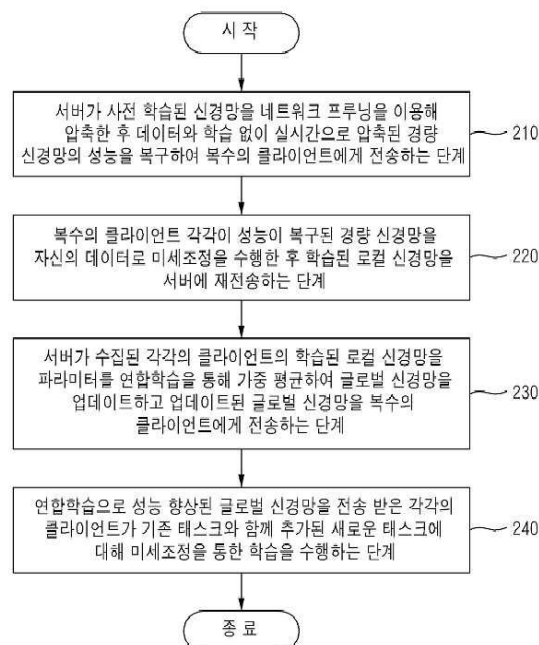
(54) 발명의 명칭 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 방법 및 시스템

(57) 요약

개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 방법 및 시스템이 제시된다. 본 발명에서 제안하는 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 방법은 서버가 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송하는 단계

(뒷면에 계속)

대표도 - 도2



성능을 복구하여 복수의 클라이언트에게 전송하는 단계, 상기 복수의 클라이언트 각각이 상기 성능이 복구된 경량 신경망을 자신의 데이터로 미세조정을 수행한 후 학습된 로컬 신경망을 서버에 재전송하는 단계, 상기 서버가 상기 수집된 각각의 클라이언트의 학습된 로컬 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송하는 단계 및 상기 연합학습으로 성능 향상된 글로벌 신경망을 전송 받은 각각의 클라이언트가 기존 태스크와 함께 추가된 새로운 태스크에 대해 미세조정을 통한 학습을 수행하는 단계를 포함한다.

(52) CPC특허분류

G06N 3/084 (2023.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 1711135511
과제번호 2021R1F1A1060160
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 기본연구

연구과제명 [Ezbaro] 질의 가능한 인공지능: 상호작용형 XAI 서비스를 실현하기 위한 질의 기반 딥러닝 모델 해석 기술

기 여 율 50/100
과제수행기관명 인하대학교
연구기간 2021.06.01 ~ 2022.02.28

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 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

서버가 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송하는 단계;

상기 복수의 클라이언트 각각이 상기 성능이 복구된 경량 신경망을 자신의 데이터로 미세조정을 수행한 후 학습된 로컬 신경망을 서버에 재전송하는 단계;

상기 서버가 상기 수집된 각각의 클라이언트의 학습된 로컬 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송하는 단계; 및

상기 연합학습으로 성능 향상된 글로벌 신경망을 전송 받은 각각의 클라이언트가 기존 태스크와 함께 추가된 새로운 태스크에 대해 미세조정을 통한 학습을 수행하는 단계

를 포함하는 합성곱 신경망의 경량화 방법.

청구항 2

제1항에 있어서,

상기 서버가 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송하는 단계는,

상기 프루닝으로 압축된 경량 신경망의 성능을 복구하기 위해 상기 경량 신경망의 제거되는 필터와 보존되는 필터 간의 선형관계를 가정하고 배치 정규화와 활성화 함수에 따른 손실함수를 구하는 단계;

상기 손실함수를 최소화하는 계수들을 구하는 단계; 및

제거되는 필터로 인해 연쇄적으로 제거되는 다음 레이어의 필터의 채널에 상기 계수를 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 손실 정보를 전달하여 상기 프루닝으로 압축된 경량 신경망의 성능을 복구하는 단계

를 포함하는 합성곱 신경망의 경량화 방법.

청구항 3

제1항에 있어서,

상기 서버가 상기 수집된 각각의 클라이언트의 학습된 경량 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송하는 단계는,

상기 연합학습에 참여하는 클라이언트의 목적함수의 합을 최소화하기 위해 각각의 클라이언트의 로컬 파라미터를 가중 평균하여 글로벌 신경망을 생성하고, 상기 연합학습을 통한 글로벌 신경망을 복수의 라운드로 학습하는 합성곱 신경망의 경량화 방법.

청구항 4

제3항에 있어서,

상기 연합학습은 서버가 전체 클라이언트 중 일부의 클라이언트를 선택하여 신경망을 배포하고, 서버로부터 배포된 신경망에 대하여 해당 클라이언트가 자신의 데이터로 상기 신경망을 학습한 후, 서버가 클라이언트의 학습된 로컬 신경망을 취합하여 새로운 글로벌 신경망을 생성하는

합성곱 신경망의 경량화 방법.

청구항 5

사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송하는 손실 함수 계산부; 및

상기 복수의 클라이언트 각각이 상기 성능이 복구된 경량 신경망을 자신의 데이터로 미세조정을 수행한 후 학습된 로컬 신경망을 상기 복수의 클라이언트로부터 수신 받아 상기 수집된 각각의 클라이언트의 학습된 로컬 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송하는 연합 학습부

를 포함하는 합성곱 신경망의 경량화 시스템.

청구항 6

제5항에 있어서,

상기 손실 함수 계산부는,

상기 프루닝으로 압축된 경량 신경망의 성능을 복구하기 위해 상기 경량 신경망의 제거되는 필터와 보존되는 필터 간의 선형관계를 가정하고 배치 정규화와 활성화 함수에 따른 손실함수를 구하고, 상기 손실함수를 최소화하는 계수들을 구하고, 제거되는 필터로 인해 연속적으로 제거되는 다음 레이어의 필터의 채널에 상기 계수를 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 손실 정보를 전달하여 상기 프루닝으로 압축된 경량 신경망의 성능을 복구하는

합성곱 신경망의 경량화 시스템.

청구항 7

제5항에 있어서,

상기 연합 학습부는,

상기 연합학습에 참여하는 클라이언트의 목적함수의 합을 최소화하기 위해 각각의 클라이언트의 로컬 파라미터를 가중 평균하여 글로벌 신경망을 생성하고, 상기 연합학습을 통한 글로벌 신경망을 복수의 라운드로 학습하는

합성곱 신경망의 경량화 시스템.

청구항 8

제7항에 있어서,

상기 연합 학습부는,

전체 클라이언트 중 일부의 클라이언트를 선택하여 신경망을 배포하고, 배포된 신경망에 대하여 해당 클라이언트가 자신의 데이터로 상기 신경망을 학습한 후, 클라이언트의 학습된 로컬 신경망을 취합하여 새로운 글로벌 신경망을 생성하는

합성곱 신경망의 경량화 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 방법 및 시스템에 관한 것이다.

배경 기술

[0002] 모델 경량화에 관한 연구는 컴퓨팅 자원이 제한된 환경에서 심층 신경망을 사용하기 위해 활발히 연구되고 있다. 모델 경량화 기법 중 하나인 네트워크 프루닝은 신경망 내 불필요한 파라미터를 프루닝 기준(pruning criterion)을 이용해 제거함으로써 신경망을 압축한다. 이때 네트워크 프루닝을 이용한 압축된 신경망은 압축률이 높아짐에 따라 추론에 필요한 파라미터가 제거되어 성능 하락이 발생하고 프루닝 이후 하락된 성능은 미세조정을 통해 복구된다.

[0003] 하지만 미세조정은 수많은 데이터와 컴퓨팅 자원이 필요하다는 점에서 자원이 제한된 엣지 디바이스와 같은 환경에서 수행되기 어려우며, 최근 클라이언트의 개인 정보 보호가 중요해지고 데이터 서버 유지 비용이 증가함에 따라 데이터를 저장하고 수집하는 데 제약이 따르기 때문에 미세조정을 수행하는 데 어려움이 있다.

[0004] 미세조정 없이 프루닝이 이후 하락된 성능을 복구하는 기존 연구들[3,4]은 제거되는 파라미터와 가장 유사한 보존되는 파라미터를 선택하고 선택된 파라미터를 보존되는 파라미터에 상속함으로써 압축된 신경망의 미세조정 과정 없이 복구하는 방법을 제안한다. 하지만 이렇게 유사도 기반 복구 기법들은 제거되는 파라미터와 유사한 파라미터가 부재할 경우 성능 복구가 제한적이라는 단점이 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명이 이루고자 하는 기술적 과제는 일차적인 성능 복구를 위해 합성곱 신경망의 제거되는 필터와 보존되는 필터들 간의 선형관계를 가정하고 배치 정규화와 활성화 함수를 고려한 손실함수를 새롭게 정의하며, 이러한 손실함수를 최소화하는 계수들을 구하고, 제거되는 필터로 인해 연쇄적으로 제거되는 다음 레이어의 필터의 채널에 해당 계수를 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 사전 학습된 신경망과 프루닝 된 신경망과의 재구성 오류(reconstruction error)를 최소화하는 방법 및 시스템을 제안한다. 이후 보다 근본적인 성능 복구를 위해 여러 클라이언트들의 신경망을 공유하는 연합 학습(Federated Learning)에 기반한 미세조정을 수행함으로써 개인정보를 포함하는 데이터를 전혀 공유하지 않고도 높은 성능의 압축된 신경망을 획득하고자 한다.

과제의 해결 수단

[0006] 일 측면에 있어서, 본 발명에서 제안하는 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 방법은 서버가 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송하는 단계, 상기 복수의 클라이언트 각각이 상기 성능이 복구된 경량 신경망을 자신의 데이터로 미세조정을 수행한 후 학습된 로컬 신경망을 서버에 재전송하는 단계, 상기 서버가 상기 수집된 각각의 클라이언트의 학습된 로컬 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송하는 단계 및 상기 연합학습으로 성능 향상된 글로벌 신경망을 전송 받은 각각의 클라이언트가 기존 태스크와 함께 추가된 새로운 태스크에 대해 미세조정을 통한 학습을 수행하는 단계를 포함한다.

[0007] 상기 서버가 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송하는 단계는 상기 프루닝으로 압축된 경량 신경망의 성능을 복구하기 위해 상기 경량 신경망의 제거되는 필터와 보존되는 필터 간의 선형관계를 가정하고 배치 정규화와 활성화 함수에 따른 손실함수를 구하는 단계, 상기 손실함수를 최소화하는 계수들을 구하는 단계 및 제거되는 필터로 인해 연쇄적으로 제거되는 다음 레이어의 필터의 채널에 상기 계수를 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 손실 정보를 전달하여 상기 프루닝으로 압축된 경량 신경망의 성능을 복구하는 단계를 포함한다.

[0008] 상기 서버가 상기 수집된 각각의 클라이언트의 학습된 경량 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송하는 단계는 상기 연합학습에 참여하는 클라이언트의 목적함수의 합을 최소화하기 위해 각각의 클라이언트의 로컬 파라미터를 가중 평균하여 글로벌 신경망을 생성하고, 상기 연합학습을 통한 글로벌 신경망을 복수의 라운드로 학습한다.

[0009] 본 발명의 실시예에 따른 연합학습은 서버가 전체 클라이언트 중 일부의 클라이언트를 선택하여 신경망을 배포하고, 서버로부터 배포된 신경망에 대하여 해당 클라이언트가 자신의 데이터로 상기 신경망을 학습한 후, 서버가 클라이언트의 학습된 로컬 신경망을 취합하여 새로운 글로벌 신경망을 생성한다.

[0010] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 시스템은 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송하는 손실 함수 계산부 및 상기 복수의 클라이언트 각각이 상기 성능이 복구된 경량 신경망을 자신의 데이터로 미세조정을 수행한 후 학습된 로컬 신경망을 상기 복수의 클라이언트로부터 수신 받아 상기 수집된 각각의 클라이언트의 학습된 로컬 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송하는 단계를 포함한다.

언트에게 전송하는 연합 학습부를 포함한다.

발명의 효과

[0011] 본 발명의 실시예들에 따르면 일차적인 성능 복구를 위해 합성곱 신경망의 제거되는 필터와 보존되는 필터들 간의 선형관계를 가정하고 배치 정규화와 활성화 함수를 고려한 손실함수를 새롭게 정의하며, 이러한 손실함수를 최소화하는 계수들을 구하고, 제거되는 필터로 인해 연쇄적으로 제거되는 다음 레이어의 필터의 채널에 해당 계수를 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 사전 학습된 신경망과 프루닝 된 신경망과의 재구성 오류(reconstruction error)를 최소화할 수 있다. 또한, 보다 근본적인 성능 복구를 위해 여러 클라이언트들의 신경망을 공유하는 연합 학습(Federated Learning)에 기반한 미세조정을 수행함으로써 개인정보를 포함하는 데이터를 전혀 공유하지 않고도 높은 성능의 압축된 신경망을 획득할 수 있다.

도면의 간단한 설명

[0012] 도 1은 본 발명의 일 실시예에 따른 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 과정을 설명하기 위한 개략도이다.

도 2는 본 발명의 일 실시예에 따른 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 방법을 설명하기 위한 흐름도이다.

도 3은 본 발명의 일 실시예에 따른 학습 없이 프루닝으로 압축된 신경망의 성능 복구 과정을 설명하기 위한 흐름도이다.

도 4는 본 발명의 일 실시예에 따른 신경망의 성능 복구 과정을 설명하기 위한 도면이다.

도 5는 본 발명의 일 실시예에 따른 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 시스템의 구성을 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0013] 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0015] 가장 대표적인 신경망 경량화 기법 중 하나인 네트워크 프루닝(network pruning)은 압축률이 높아짐에 따라 추론에 필수적인 파라미터가 제거되어 성능 하락이 발생한다. 이렇게 프루닝 이후 하락된 성능은 미세조정을 통해 복구된다. 하지만, 최근 클라이언트의 개인정보 보호가 중요해지고 데이터 서버 유지 비용이 증가함에 따라 데이터를 저장하고 수집하는 데 제약이 따르기 때문에 미세조정을 수행하기 어려운 경우가 많다.

[0016] 네트워크 프루닝[1, 2]은 신경망 내 불필요한 필터를 프루닝 기준으로 제거함으로써 합성곱 신경망의 크기를 줄이고 추론 속도를 가속화한다. 네트워크 프루닝을 이용한 압축된 신경망의 하락된 성능은 미세조정 과정을 통해 복구된다. 하지만 미세조정을 수행하는 데 있어 충분한 시간과 데이터, 컴퓨팅 자원이 필요로 한다. 이러한 자원 제한 없이 프루닝 이후 성능을 복구하는 기존 연구들[3, 4]은 제거되는 파라미터와 제거되지 않은 가장 유사한 파라미터를 찾아 스케일을 곱해 더함으로써 성능을 복구한다.

[0017] 연합학습 연합학습[5]은 프라이버시를 보장하는 분산 신경망 학습 방식으로, 각 클라이언트의 데이터를 전송하지 않고 클라이언트별로 학습된 신경망을 서버에 전송한다. 서버는 각 클라이언트로부터 전송 받은 신경망을 취합하여 하나의 글로벌 신경망을 업데이트하는데 사용한다. 연합학습의 주요 문제점은 데이터 이질성 문제로, 각 클라이언트에 존재하는 학습 데이터는 클라이언트의 패턴에 따라 각기 다른 데이터 분포로 형성되고 이로 인해 이질적인 분포에서 학습된 신경망을 취합한 글로벌 신경망에서는 성능이 저하되고 느리게 수렴하는 문제점이 발생한다. 이러한 문제를 해결하기 위해 제안된 FedAvg[5]는 로컬 SGD 업데이트된 클라이언트 신경망을 취합해 글로벌 신경망을 업데이트한다.

[0019] 도 1은 본 발명의 일 실시예에 따른 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 과정을 설명하기 위한 개략도이다.

[0020] 본 발명에서는 프루닝으로 압축된 합성곱 신경망의 하락된 성능을 서버에서의 미세조정 과정 없이 복구하는 방법을 제안한다. 일차적으로 데이터와 추가학습 없이 서버(110)에서 사전 학습된 신경망(111)을 이용하여 초기

압축된 모델을 복구하고, 복구된 경량 신경망(112)을 복수의 클라이언트(120)에게 전송한다. 복수의 클라이언트(120) 각각이 복구된 경량 신경망을 자신의 데이터로 미세조정을 수행한 후 학습된 로컬 신경망을 서버에 재전송한다. 이후 서버로의 데이터 전송 없이 복수의 클라이언트(120)의 신경망을 공유하는 연합 학습(Federated Learning)을 수행함으로써 성능을 더욱 향상시킨다.

[0021] 더욱 상세하게는 일차적인 성능 복구를 위해 합성곱 신경망의 제거되는 필터와 보존되는 필터들 간의 선형관계를 가정하고 배치 정규화와 활성화 함수를 고려한 손실함수를 새롭게 정의한다. 이후 이러한 손실함수를 최소화하는 계수들을 구하고, 제거되는 필터로 인해 연쇄적으로 제거되는 다음 레이어의 필터의 채널에 해당 계수를 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 사전 학습된 신경망과 프루닝 된 신경망과의 재구성 오류(reconstruction error)를 최소화하는 방법을 제안한다. 이후 보다 근본적인 성능 복구를 위해 여러 클라이언트들의 신경망을 공유하는 연합 학습(Federated Learning)에 기반한 미세조정을 수행함으로써 개인정보를 포함하는 데이터를 전혀 공유하지 않고도 높은 성능의 압축된 신경망을 획득할 수 있는 방법을 제안한다.

[0022] ResNet-34를 이용한 이미지넷 실험을 통해, 본 발명에서 제안하는 성능 복구 기법을 활용한 경량 신경망의 성능이 프루닝 직후의 성능인 정확도 17%보다 최대 47.10% 더 높은 64.5%의 정확도를 달성할 수 있음을 확인하였다.

[0024] 도 2는 본 발명의 일 실시예에 따른 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 방법을 설명하기 위한 흐름도이다.

[0025] 제안하는 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 방법은 서버가 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송하는 단계(210), 상기 복수의 클라이언트 각각이 상기 성능이 복구된 경량 신경망을 자신의 데이터로 미세조정을 수행한 후 학습된 로컬 신경망을 서버에 재전송하는 단계(220), 상기 서버가 상기 수집된 각각의 클라이언트의 학습된 로컬 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송하는 단계(230) 및 상기 연합학습으로 성능 향상된 글로벌 신경망을 전송 받은 각각의 클라이언트가 기존 태스크와 함께 추가된 새로운 태스크에 대해 미세조정을 통한 학습을 수행하는 단계(240)를 포함한다.

[0026] 먼저 단계(210)에서, 서버가 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송한다.

[0028] 도 3은 본 발명의 일 실시예에 따른 학습 없이 프루닝으로 압축된 신경망의 성능 복구 과정을 설명하기 위한 흐름도이다.

[0029] 도 1의 서버가 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송하는 단계(210)는 상기 프루닝으로 압축된 경량 신경망의 성능을 복구하기 위해 상기 경량 신경망의 제거되는 필터와 보존되는 필터 간의 선형관계를 가정하고 배치 정규화와 활성화 함수에 따른 손실함수를 구하는 단계(310), 상기 손실함수를 최소화하는 계수들을 구하는 단계(320) 및 제거되는 필터로 인해 연쇄적으로 제거되는 다음 레이어의 필터의 채널에 상기 계수를 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 손실 정보를 전달하여 상기 프루닝으로 압축된 경량 신경망의 성능을 복구하는 단계(330)를 포함한다.

[0030] 단계(310)에서, 프루닝으로 압축된 경량 신경망의 성능을 복구하기 위해 상기 경량 신경망의 제거되는 필터와 보존되는 필터 간의 선형관계를 가정하고 배치 정규화와 활성화 함수에 따른 손실함수를 구한다.

[0032] 도 4는 본 발명의 일 실시예에 따른 신경망의 성능 복구 과정을 설명하기 위한 도면이다.

[0033] 본 발명의 실시예에 따르면, 제거되는 필터가 복수의 필터에 정보를 분할하여 전달함으로써 데이터와 학습 없이 성능을 복구하는 기법을 제안한다. 이를 위해 도 4와 같이 l 번째 레이어의 m 개의 필터 중 제거되는 j 번째 필터(w_j^l)와 보존되는 필터들간의 선형관계를 식(1)과 같이 가정한다:

[0034] **Linear assumption** : $W_j^l \approx s_1 W_1^l + s_2 W_2^l + \dots + s_m W_m^l$ 식(1)

[0035] 이후 사전 학습된 신경망의 출력($Z^{(l+1)}$)와 프루닝으로 압축된 신경망의 복구된 출력($\hat{Z}^{(l+1)}$) 재구성 오류(reconstruction error)는 식(2)와 같다:

$$|Z^{(l+1)} - \hat{Z}^{(l+1)}| = |A^{(l)} \odot (W^{(l+1)} - W^{(l+1)} \times_2 S^T)|, \text{ where } A^{(l)} = F(N(Z^{(l)}), S \in \mathbb{R}^2 \quad \text{식(2)}$$

[0037] 여기서, F와 N은 각각 활성화 함수와 배치 정규화 레이어를 의미하며 S는 2차원 행렬로 l 번째 레이어에서 제거되는 필터로 인해 연속적으로 다음 레이어의 필터의 채널이 제거된 것을 2-모드 곱(mode product) 연산으로 나타낸 것이다. 배치 정규화와 활성화 함수 레이어를 고려한 재구성 오류는 다음과 같이 정의한다:

$$|Z^{(l+1)} - \hat{Z}^{(l+1)}| = \left| \frac{\gamma_j}{\sigma_j} (A^{(l-1)} \odot \mathcal{E}) + \mathcal{B} + \mathcal{R} \right|, \text{ where } \mathcal{E} = W_j^l - \sum_{k=1, k \neq j}^m s_k W_k^l, \mathcal{B} = \frac{\gamma_j}{\sigma_j} \left\{ \sum_{k=1, k \neq j}^m s_k \frac{\gamma_j \gamma_k}{\sigma_j \sigma_k} \left(\mu_k - \frac{\gamma_k}{\sigma_k} \beta_k \right) - \mu_j + \frac{\gamma_j}{\sigma_j} \beta_j \right\}, \mathcal{R} = \sum_{k=1, k \neq j}^m s_k \min(0, N(Z_k^{(l)})) - \min(0, N(Z_j^{(l)})) \quad \text{식(3)}$$

[0039] 여기서, $\mathcal{E}$ 는 잔차 오류이며, $\mathcal{B}$ 는 배치정규화 오류, $\mathcal{R}$ 은 활성화 함수 오류이다. 그리고 γ, σ 는 배치정규화 파라미터를 의미한다. 재구성 오류를 데이터 없이 최소화하기 위해 각각의 잔차, 배치 정규화, 활성화 함수 오류를 고려한 손실함수를 다음과 같이 정의한다:

$$L_{re} = ||\mathcal{E}||_2^2 + \lambda_1 ||\mathcal{B}||_2^2 + \lambda_2 ||s||_2^2 \quad \text{식(4)}$$

[0041] 여기서, λ_1, λ_2 는 배치 정규화 오류와 활성화 오류를 조절하는 파라미터이다.

[0042] 단계(320)에서, 손실함수를 최소화하는 계수들을 구한다.

[0043] 재구성 오류를 데이터와 학습 없이 최소화하기 위해, 식(4) 손실함수를 최소화하는 계수 S를 닫힌 형태(closed form)를 이용하여 구한다.

[0044] 단계(330)에서, 제거되는 필터로 인해 연속적으로 제거되는 다음 레이어의 필터의 채널에 상기 계수를 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 손실 정보를 전달하여 상기 프루닝으로 압축된 경량 신경망의 성능을 복구한다.

[0045] 도 4의 아래 그림과 같이 제거되는 l 번째 레이어의 j 번째 필터로 인해 연속적으로 제거되는 $l+1$ 번째 레이어의 필터 채널에 해당 계수를 적절히 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 손실 정보를 전달하여, 네트워크 프루닝으로 압축된 신경망의 하락된 성능을 복구한다.

[0046] 다시 도 2의 순서도를 참조하면, 단계(220)에서 복수의 클라이언트 각각이 상기 성능이 복구된 경량 신경망을 자신의 데이터로 미세조정을 수행한 후 학습된 로컬 신경망을 서버에 재전송한다.

[0047] 단계(230)에서, 서버가 상기 수집된 각각의 클라이언트의 학습된 로컬 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송한다.

[0048] 복구된 경량 신경망은 데이터를 전혀 사용하지 않았기에, 아주 높은 정확도를 달성하기에는 한계점이 존재한다. 따라서, 개인정보를 보장하면서도 압축된 신경망의 성능을 더욱 높이기 위해 본 발명에서는 연합학습에 기반한 미세조정을 제안한다. 다시 말해, 서버의 입장에서는 여전히 데이터를 전혀 사용하지 않고도 전체 성능을 향상시킬 수 있는 것이다. 연합학습은 학습에 참여하는 클라이언트들의 목적함수의 합을 최소화하는 것을 목표로 한다. 이를 수식으로 나타내면 식(5)와 같다:

$$\min_{\theta} F(\theta) = \sum_{i=1}^M \frac{|\mathcal{D}_i|}{|\mathcal{D}|} F_i(\theta),$$

$$\text{where } F_i(\theta) = \frac{1}{|\mathcal{D}_i|} \sum_{x \in \mathcal{D}_i} \ell(\theta; x)$$

식(5)

신경망 파라미터를 $\theta \in R^d$, 연합학습에 참여하는 전체 클라이언트의 수를 M , F_i 는 i^{th} 클라이언트의 로컬 목적 함수를 의미한다. 클라이언트들의 학습 데이터 집합을 $\mathcal{D}_i$, 연합학습에서의 전체 데이터 집합을 $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_M\}$ 이라 정의할 때, i^{th} 클라이언트의 가중치는 자신의 학습 데이터 수에 비례한 값 $\frac{|\mathcal{D}_i|}{|\mathcal{D}|}$ 으로 설정된다. 즉, 클라이언트들의 로컬 파라미터를 가중 평균하여 글로벌 신경망을 구한다.

본 발명의 실시예에 따른 연합학습의 글로벌 신경망은 총 T번의 라운드로 학습되며, 하나의 라운드는 다음과 같이 진행된다. 첫 번째로, 서버가 전체 클라이언트 중 일부를 선택해 신경망을 배포한다. 두 번째로, 선택된 클라이언트는 자신의 데이터로 신경망을 학습한 후 서버에 전송한다. 마지막으로, 서버는 클라이언트의 학습된 로컬 신경망을 취합해 새로운 글로벌 신경망을 형성한다.

단계(240)에서, 연합학습으로 성능 향상된 글로벌 신경망을 전송 받은 각각의 클라이언트가 기존 태스크와 함께 추가된 새로운 태스크에 대해 미세조정을 통한 학습을 수행한다.

예를 들어, 클라이언트에 적은 수의 데이터를 가진 새로운 태스크가 추가되는 경우, 연합학습으로 성능 향상된 경량 신경망을 전송 받은 클라이언트는 기존 태스크와 함께 추가된 새로운 태스크에 대해 균형 잡힌 미세조정 방법으로 신경망을 학습할 수 있다.

도 5는 본 발명의 일 실시예에 따른 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 시스템의 구성을 나타내는 도면이다.

제안하는 개인정보 데이터 공유 없이 신경망을 압축하기 위한 연합학습 기반 미세조정 시스템의 서버(500)는 손실 함수 계산부(510) 및 연합 학습부(520)를 포함한다.

본 발명의 실시예에 따른 손실 함수 계산부(510)는 사전 학습된 신경망을 네트워크 프루닝을 이용해 압축한 후 데이터와 학습 없이 실시간으로 압축된 경량 신경망의 성능을 복구하여 복수의 클라이언트에게 전송한다.

본 발명의 실시예에 따른 손실 함수 계산부(510)는 상기 프루닝으로 압축된 경량 신경망의 성능을 복구하기 위해 상기 경량 신경망의 제거되는 필터와 보존되는 필터 간의 선형관계를 가정하고 배치 정규화와 활성화 함수에 따른 손실함수를 구한다. 이후, 상기 손실함수를 최소화하는 계수들을 구하고, 제거되는 필터로 인해 연쇄적으로 제거되는 다음 레이어의 필터의 채널에 상기 계수를 곱하여 나머지 보존되는 필터들의 채널에 더함으로써 손실 정보를 전달하여 상기 프루닝으로 압축된 경량 신경망의 성능을 복구한다.

본 발명의 실시예에 따른 연합 학습부(520)는 복수의 클라이언트 각각이 상기 성능이 복구된 경량 신경망을 자신의 데이터로 미세조정을 수행한 후 학습된 로컬 신경망을 상기 복수의 클라이언트로부터 수신 받아 상기 수집된 각각의 클라이언트의 학습된 로컬 신경망의 파라미터를 연합학습을 통해 가중 평균하여 글로벌 신경망을 업데이트하고 업데이트된 글로벌 신경망을 복수의 클라이언트에게 전송한다.

본 발명의 실시예에 따른 연합 학습부(520)는 연합학습에 참여하는 클라이언트의 목적함수의 합을 최소화하기 위해 각각의 클라이언트의 로컬 파라미터를 가중 평균하여 글로벌 신경망을 생성하고, 상기 연합학습을 통한 글로벌 신경망을 복수의 라운드로 학습한다.

본 발명의 실시예에 따른 연합학습은 전체 클라이언트 중 일부의 클라이언트를 선택하여 신경망을 배포하고, 배포된 신경망에 대하여 해당 클라이언트가 자신의 데이터로 상기 신경망을 학습한 후, 클라이언트의 학습된 로컬 신경망을 취합하여 새로운 글로벌 신경망을 생성한다.

본 발명에서 사용한 이미지넷[8] 데이터셋의 학습, 검증 데이터는 각각 약 120만개, 5만개의 이미지로 구성되어 있고 클래스의 수는 총 1000개이다. 그리고 사전 학습된 신경망에서 사용된 적이 없는(unseen) 적은 수의 데이터 셋은 오픈 데이터셋[7]을 사용하였다. 실험에서 사용된 신경망은 ResNet[9], MobileNet-V2[10]이다.

본 발명의 실험 환경은 클라이언트는 NVIDIA Jetson TX2 (8GB)와 서버는 NVIDIA TITAN RTX를 사용하여 수행하였

다.

[0064] 실험에서 사용한 압축률(compression ratio)은 원본 신경망을 기준으로 압축된 FLOPs 비율이다. 연합학습 환경은 전체 100개의 클라이언트가 참여하고, 각 클라이언트가 이미지넷 1000개 클래스 중 10개 클래스 데이터만 갖도록 데이터를 할당한다. 프루닝된 신경망의 업데이트는 총 50라운드를 진행하고, 한 라운드당 10개의 클라이언트가 랜덤으로 선택된다. 그리고 종래기술[6]과 같이 적은 수의 데이터로 구성된 새로운 태스크가 클라이언트에 들어온 경우에 데이터셋의 구성은 5개 클래스와 클래스 당 10개 이미지(5-way 10-shot)이다. 이 때 클라이언트의 합성곱 신경망의 특징 추출기(feature extractor)는 고정되고, 새로운 (novel) 클래스 수에 맞춰 분류기(classifier)를 확장한 분류기는 기존 클래스에 대해 언더샘플링하여 새로운 클래스와 균형 잡힌 데이터셋을 이용한 미세조정을 수행하여 학습된다.

[0065] 표 1은 프루닝만 수행한 성능과 프루닝 이후 성능을 복구한 신경망의 성능을 비교한 결과와 이 두 신경망을 연합학습으로 업데이트를 수행한 비교 결과이다.

[0066] <표 1>

Compression ratio		Rounds					
(FLOPs)		0		10		50	
Model (Acc, FLOPs)	Method	FedPress	Prune	FedPress	Prune	FedPress	Prune
ResNet34	71% (2.59G)	48.38	17.02	63.43	60.57	64.50	63.25
(73.27%, 3.66G)	52% (1.90G)	9.45	0.88	49.36	43.32	53.53	51.01
MobileNet-V2	96% (201M)	52.72	39.54	65.8	64.87	65.84	65.48
(69.97% 209M)	92% (193M)	9.25	1.95	58.11	58.92	61.08	61.53

[0067]

[0068] 0번째 라운드는 프루닝 이후 성능 복구를 수행한 결과를 나타낸다.

[0069] FedPress를 적용한 ResNet-34 신경망의 복구된 정확도는 압축률 71%, 52%에서, 프루닝을 수행하고 연합학습을 수행한 신경망보다 각각 31.36%, 8.57% 더 높았다. 실험을 통해, 서버가 학습에 사용할 데이터가 없는 상황에서 제안된 방법은 프루닝으로 압축된 신경망을 실시간으로 복구하여, 더 높은 성능의 신경망을 생성함을 확인하였다. 또한, 10번째 라운드와 50번째 라운드의 실험 결과를 통해, 본 발명에서 제안한 성능 복구된 신경망이 기존 프루닝만 수행한 신경망보다 더 빠른 수렴속도를 보여, FedPress가 서버와 클라이언트 간의 통신이 더 적게 필요한 효율적인 시스템임을 확인하였다. 그리고 프루닝을 수행하지 않고 초기화된 신경망을 연합학습을 통해 처음부터 학습(from-scratch)한다면, 500라운드에서 최고 정확도를 10.78%를 달성함을 확인하였다. (표 1)의 실험을 통해, 개인 프라이버시를 보장하는 상황에서, FedPress는 사전 학습된 신경망을 실시간적으로 압축하고 복구하며, 복구된 경량 신경망의 성능은 연합학습을 이용하여 성능이 더욱 개선된 경량 신경망을 생성함을 확인했다.

[0070] 표 2는 사전 학습된 MobileNet-V2의 특정 태스크에 전문화된 정확도와 네트워크 프루닝으로 압축된 신경망을 제안한 기법을 통해 복구하고 난 후 클라이언트 자신의 데이터로 학습한 성능을 비교한 것이다.

[0071] <표 2>

	Acc	FLOPs	Size(M)
Oracle model	94.4	209	2.6
FedPress	94.2	193	2.1

[0072]

[0073] 클라이언트의 개인 데이터 분포에 학습된 전문화된 신경망은 사전 학습된 신경망보다 약 20% 더 작아졌지만, 성능 차이는 0.2%에 불과하다.

[0074] 표 3은 클라이언트가 적은 수의 데이터를 가지는 새로운 태스크를 MobileNet-V2에 대해 추론한 결과이다.

[0075] <표 3>

Before	Unseen	
Specialized Performance (Base)	Base	Novel
94.2	89.5	75.1

[0076]

[0077] 이때 기존 태스크의 정확도 하락은 4.7%에 불과하고, 새로운 태스크의 정확도는 75.1%를 달성하였다. 이를 통해, 연합학습을 이용한 신경망이 새로운 태스크에 빠르게 적응하는 것을 확인하였다.

[0078] 표 4는 본 발명에서 제안한 방법과 최신 프루닝 기법인 NPPM[5]와의 성능을 MobileNet-V2에 대해 비교한 결과이다.

[0079] <표 4>

	Acc	FLOPs(M)
FedPress	65.84	201
NPPM	72.02	221

[0080]

[0081] 이를 통해 학습 데이터를 사용해서 200 epochs 미세조정을 수행한 NPPM[5]보다 성능이 6.2% 성능 낮지만 20M더 적은 FLOPs를 보임을 확인하였다. 최신 연구와의 비교를 통해, 프라이버시를 보존하기 위해 데이터를 전혀 공유하지 않는 본 발명의 압축 방법이 모든 데이터를 충분히 보유한 상태에서 압축을 수행하는 방법에 비해 어느 정도 경쟁력 있는 성능을 보임을 확인하였다.

[0083] 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.

[0084] 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, 가상장치(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.

[0085] 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.

[0086] 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.

[0087] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

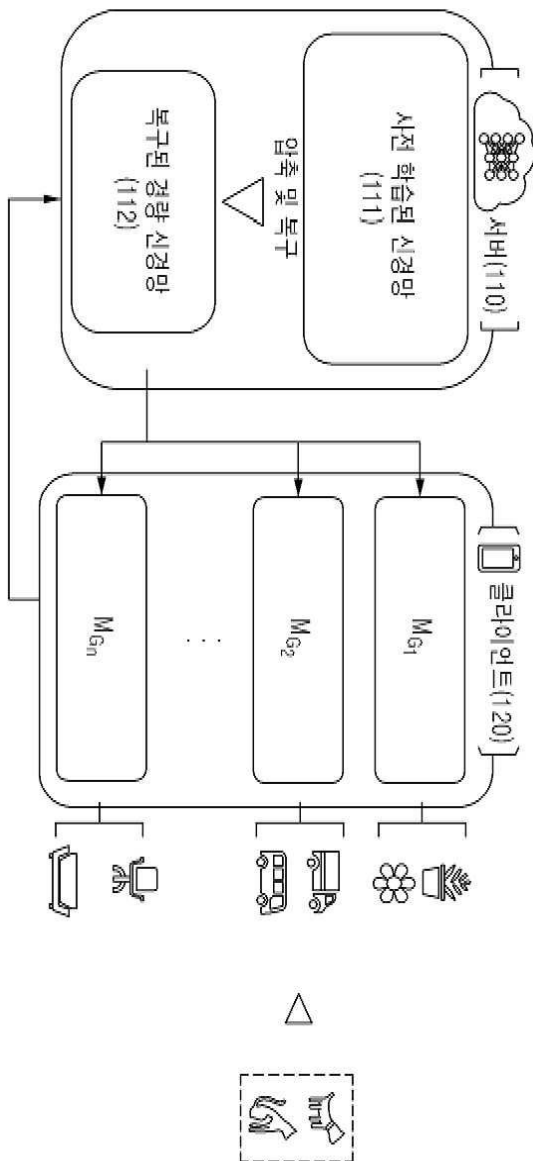
다.

[0089] <참고문헌>

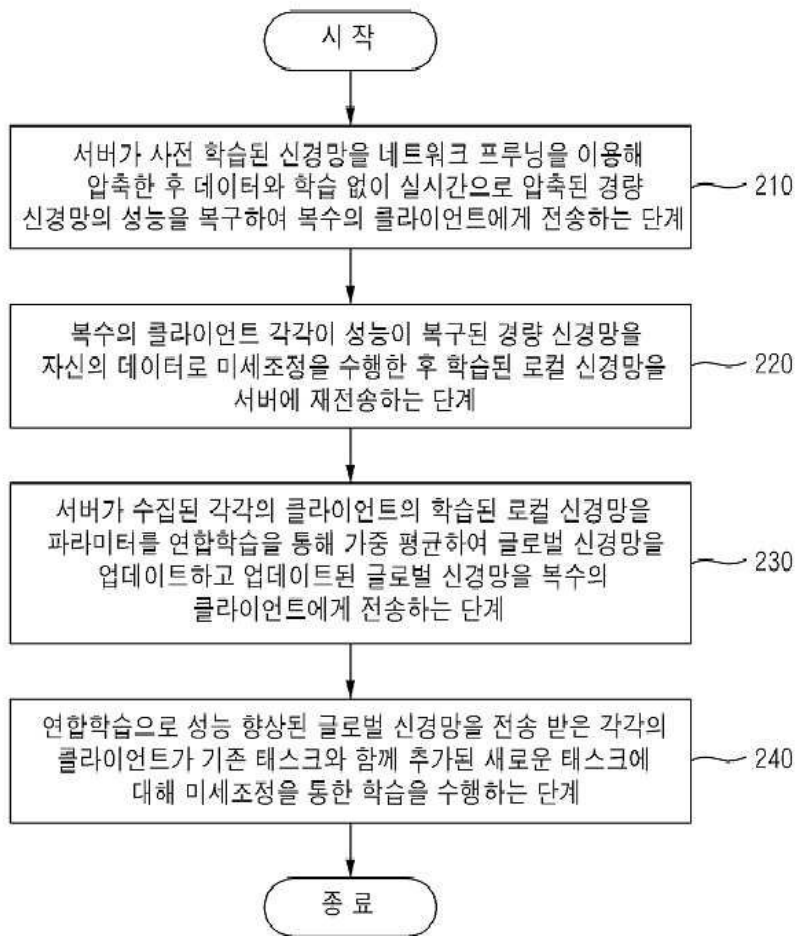
- [0090] [1] He, Y., et al. "Soft filter pruning for accelerating deep convolutional neural networks", In IJCAI, 2018.
- [0091] [2] Gao, et al. "Network Pruning via Performance Maximization", In CVPR, 2021.
- [0092] [3] Srinivas et al. "Data-free Parameter Pruning for Deep Neural Networks", In BMVC, 2015
- [0093] [4] Kim et al. "Neuron Merging: Compensating for Pruned Neurons", In NeurIPS, 2020.
- [0094] [5] B. McMahan et al. "Communication-efficient learning of deep networks from decentralized data''. In AISTATS, 2017
- [0095] [6] Wei-Yu Chen et al. "A closer look at few-shot classification." In ICLR, 2019.
- [0096] [7] A. Kuznetsova et al. "The open images dataset v4," Int. J. Comput. Vis., pp. 1-26, 2020.
- [0097] [8] Russakovsky et al. "Imagenet large scale visual recognition challenge." In International journal of computer vision, 2015.
- [0098] [9] He, Kaiming, et al. "Deep residual learning for image recognition." In CVPR, 2016.
- [0099] [10] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." In CVPR, 2018.

도면

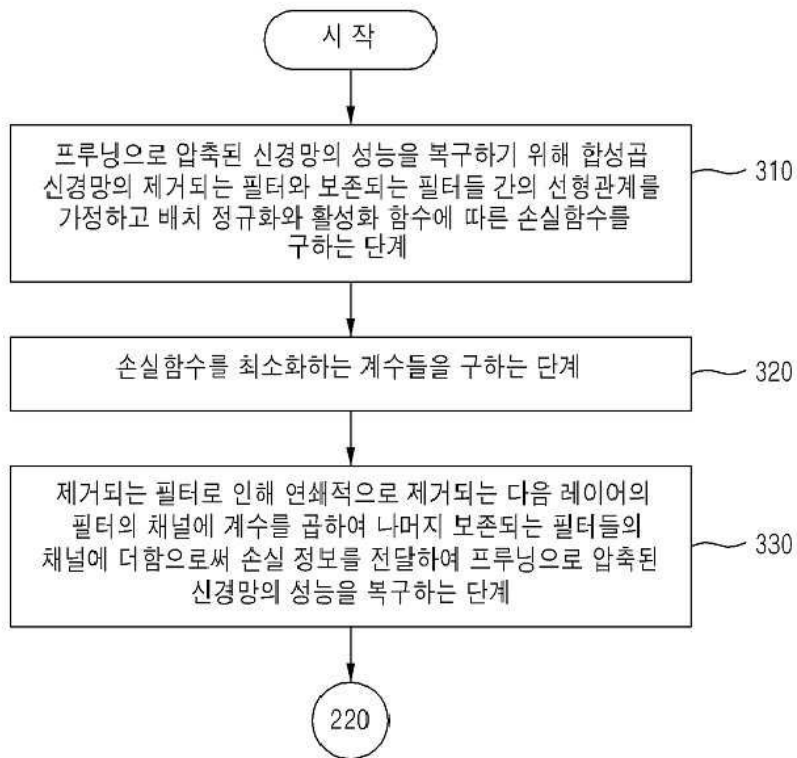
도면1



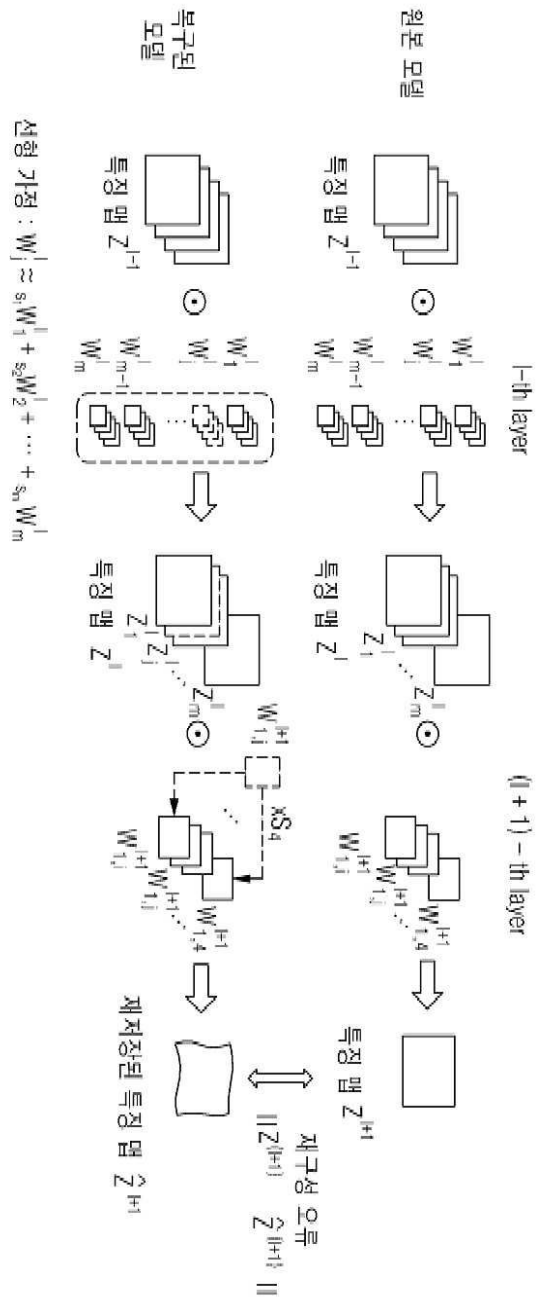
도면2



도면3



도면4



도면5

