(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0065883
(43) 공개일자 2023년05월12일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2023.01) *G06F 17/16* (2006.01)
G06N 3/04 (2023.01)
(52) CPC특허분류
G06N 3/082 (2023.01)
G06F 17/16 (2013.01)
(21) 출원번호 10-2022-0109174
(22) 출원일자 2022년08월30일
심사청구일자 2022년08월30일
(30) 우선권주장
1020210151683 2021년11월05일 대한민국(KR)

(71) 출원인
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)
(72) 발명자
오태현
경상북도 포항시 남구 청암로 77(지곡동)
남현우
경상북도 포항시 남구 청암로 77(지곡동)
문예빈
경상북도 포항시 남구 청암로 77(지곡동)
(74) 대리인
특허법인(유한)아이시스

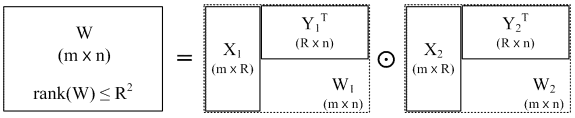
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **풀-랭크 축소 파라미터화를 이용한 데이터 경량화 방법 및 데이터 처리 장치**

(57) 요 약

데이터 처리 장치는 벡터 행렬로 표현되는 타깃 데이터 및 데이터 경량화를 수행하도록 제어하는 명령어들을 저장하는 메모리 및 타깃 파라미터 행렬 W 구성이 가능한 로-랭크 행렬 W_1 및 로-랭크 W_2를 결정하는 프로세서를 포함한다. 상기 타깃 파라미터 행렬 W는 상기 W_1 와 상기 W_2 사이의 하다마드 곱으로 구성된다.

대 표 도 - 도2



(52) CPC특허분류
G06N 3/04 (2023.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711128750
과제번호 2021R1C1C1006799
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 개인기초연구(과기정통부)(R&D)
연구과제명 에어로졸에 의한 시각외란을 투과 가능한 다중분광 비디오 모션 증폭
기 여 율 1/2
과제수행기관명 포항공과대학교
연구기간 2021.03.01 ~ 2022.02.28
이 발명을 지원한 국가연구개발사업
과제고유번호 1711125943
과제번호 2019-0-01906-003
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성(R&D)
연구과제명 인공지능대학원지원(포항공과대학교)
기 여 율 1/2
과제수행기관명 포항공과대학교 산학협력단
연구기간 2021.01.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

프로세서 및 상기 프로세서가 데이터 경량화를 수행하도록 제어하는 명령어를 저장하는 메모리를 포함하는 장치가 아래 과정을 수행하되,

상기 메모리가 벡터 행렬로 표현되는 데이터를 입력받는 단계; 및

상기 프로세서가 타깃 파라미터 행렬 W 구성이 가능한 로-랭크 행렬 W_1 및 W_2를 결정하는 단계를 포함하되,

상기 타깃 파라미터 행렬 W는 상기 W_1 와 상기 W_2 사이의 하다마드 곱으로 구성되고,

상기 W_1은 $m \times r_1$ 크기의 X_1 행렬 및 $n \times r_1$ 크기의 Y_1 행렬의 내적으로 형성되고, 상기 W_2는 $m \times r_2$ 크기의 X_2 행렬 및 $n \times r_2$ 크기의 Y_2 행렬의 내적으로 형성되는 데이터 경량화 방법. (여기서, m, n, r_1, r_2는 자연수임)

청구항 2

제1항에 있어서,

상기 데이터는 신경망의 컨볼루션 계층의 파라미터들 또는 신경망의 전연결 계층의 파라미터들을 포함하는 데이터 경량화 방법.

청구항 3

제1항에 있어서,

상기 프로세서는 $r_1 \times r_2 \geq \min(m,n)$를 만족하도록 상기 W_1 및 상기 W_2를 결정하는 데이터 경량화 방법.

청구항 4

제1항에 있어서,

상기 프로세서는 $r_1 = r_2 = R$ 및 $R^2 \geq \min(m,n)$을 만족하도록 상기 W_1 및 상기 W_2를 결정하는 데이터 경량화 방법. (여기서 R은 자연수)

청구항 5

제1항에 있어서,

상기 데이터는 신경망 계층의 텐서이고, 상기 텐서의 크기는 $R \times R \times k_3 \times k_4$이고, 상기 X_1 및 상기 X_2의 크기가 각각 $k_1 \times R$이고, 상기 Y_1 및 상기 Y_2의 크기가 각각 $k_2 \times R$인 데이터 경량화 방법. (여기서 R 및 k는 자연수)

청구항 6

프로세서 및 상기 프로세서가 신경망 모델을 학습하도록 제어하는 명령어를 저장하는 메모리를 포함하는 장치가 상기 신경망 모델의 계층들 중 청구항 1항 내지 청구항 5항 중 어느 하나의 항에 따른 데이터 경량화 방법으로 경량화된 타깃 계층에 대하여 아래 과정을 수행하되,

상기 메모리가 상기 타깃 계층의 타깃 파라미터 행렬을 구성하는 로-랭크 행렬 W_1 및 W_2를 입력받는 단계;

상기 프로세서가 상기 로-랭크 행렬 W_1 및 상기 W_2에 대한 하다마드 곱을 수행하여 상기 타깃 파라미터 행렬 W를 구성하는 단계; 및

상기 프로세서가 학습 데이터로부터 추출되는 특징을 이용하여 상기 타깃 파라미터 행렬 W를 갱신하는 단계를

포함하고,

상기 타깃 계층은 컨볼루션 계층들 중 어느 하나 또는 전연결 계층들 중 어느 하나인 신경망 모델 학습 방법.

청구항 7

프로세서 및 상기 프로세서가 신경망 모델을 이용한 추론 과정을 제어하는 명령어를 저장하는 메모리를 포함하는 장치가 상기 신경망 모델의 계층들 중 청구항 1항 내지 청구항 5항 중 어느 하나의 항에 따른 데이터 경량화 방법으로 경량화된 타깃 계층에 대하여 아래 과정을 수행하되,

상기 메모리가 상기 타깃 계층의 타깃 파라미터 행렬을 구성하는 로-랭크 행렬 W_1 및 W_2를 입력받는 단계;

상기 프로세서가 상기 로-랭크 행렬 W_1 및 상기 W_2에 대한 하다마드 곱을 수행하여 상기 타깃 파라미터 행렬 W를 구성하는 단계; 및

상기 프로세서가 입력 데이터 또는 상기 타깃 계층의 이전 계층에서 전달되는 특징을 상기 타깃 계층에 입력하여 추론을 진행하는 단계를 포함하고,

상기 타깃 계층은 컨볼루션 계층들 중 어느 하나 또는 전연결 계층들 중 어느 하나인 신경망 모델을 이용한 추론 방법.

청구항 8

프로세서 및 상기 프로세서가 전역 데이터 및 기기 종속적 특성을 갖는 지역 데이터를 이용하여 동작하는 명령어를 저장하는 메모리를 포함하는 장치가 아래 과정을 수행하되,

상기 메모리가 청구항 1항 내지 청구항 5항 중 어느 하나의 항에 따른 데이터 경량화 방법으로 경량화되는 과정에서 결정되는 상기 전역 데이터에 대한 로-랭크 행렬 W_1 및 상기 지역 데이터에 대한 로-랭크 행렬 W_2를 입력받는 단계;

상기 프로세서가 상기 W_1 와 상기 W_2에 대한 하다마드 곱으로 타깃 파라미터 행렬 W를 생성하는 단계; 및

상기 프로세서가 상기 W를 이용하여 신경망 모델 학습 또는 신경망 모델을 이용한 추론을 수행하는 단계를 포함하되,

상기 전역 데이터에 대한 로-랭크 행렬 W_1은 상기 장치가 아닌 다른 적어도 하나의 장치에서 전역 데이터를 이용하여 학습된 결과로 산출되는 데이터 개인화 방법.

청구항 9

벡터 행렬로 표현되는 타깃 데이터 및 데이터 경량화를 수행하도록 제어하는 명령어들을 저장하는 메모리; 및

타깃 파라미터 행렬 W 구성이 가능한 로-랭크 행렬 W_1 및 로-랭크 W_2를 결정하는 프로세서를 포함하되,

상기 타깃 파라미터 행렬 W는 상기 W_1 와 상기 W_2 사이의 하다마드 곱으로 구성되고,

상기 W_1은 $m \times r_1$ 크기의 X_1 행렬 및 $n \times r_1$ 크기의 Y_1 행렬의 내적으로 형성되고, 상기 W_2는 $m \times r_2$ 크기의 X_2 행렬 및 $n \times r_2$ 크기의 Y_2 행렬의 내적으로 형성되는 데이터 처리 장치. (여기서, m, n, r_1, r_2는 자연수임)

청구항 10

제8항에 있어서,

상기 프로세서는 $r_1 \times r_2 \geq \min(m,n)$를 만족하도록 상기 W_1 및 상기 W_2를 결정하는 데이터 처리 장치.

청구항 11

제8항에 있어서,

상기 프로세서는 $r_1 = r_2 = R$ 및 $R^2 \geq \min(m,n)$을 만족하도록 상기 W_1 및 상기 W_2를 결정하는 데이터 처리 장치.

청구항 12

제8항에 있어서,

상기 데이터는 신경망 계층의 텐서에 해당하고, 상기 텐서의 크기는 R ×R ×k_3 × k_4이고, 상기 X_1 및 상기 X_2의 크기가 각각 k_1×R이고, 상기 Y_1 및 상기 Y_2의 크기가 각각 k_2×R인 데이터 경량화를 수행하는 데이터 처리 장치.

발명의 설명

기 술 분 야

[0001] 이하 설명하는 기술은 데이터 경량화 기법에 관한 것이다. 특히, 이하 설명하는 기술은 폴-랭크 축소 파라미터화를 이용하여 압축된 데이터 내지 신경망 계층을 산출하는 기법에 관한 것이다.

배 경 기 술

[0002] 디지털 데이터는 신호 처리, 통신, AI(artificial intelligence) 등 다양한 분야에서 활용된다. 디지털 데이터 분야에서 데이터 경량화는 처리 속도 향상 및 통신 비용 감소에 있어서 중요한 이슈이다.

선행기술문헌

특허문헌

[0003] (특허문헌 0001) 한국공개특허 제10-2021-0066754호

발명의 내용

해결하려는 과제

[0004] 종래 데이터 경량화 기법은 모델을 양자화(quantization)하여 모델 크기를 축소하거나, 로-랭크 파라미터화(low-rank parameterization)으로 파라미터들의 개수를 줄이는 방식이었다. 그러나 종래 기술은 데이터의 양이 줄어들지만 동시에 유의한 정보가 손실되는 문제가 있었다.

[0005] 이하 설명하는 기술은 벡터 행렬로 표현되는 데이터의 파라미터들의 개수를 줄이면서도 최대 랭크(rank)를 유지하는 기법을 제공하고자 한다.

과제의 해결 수단

[0006] 데이터 경량화 방법은 프로세서 및 상기 프로세서가 데이터 경량화를 수행하도록 제어하는 명령어를 저장하는 메모리를 포함하는 장치가 아래 과정을 수행하되, 상기 메모리가 벡터 행렬로 표현되는 데이터를 입력받는 단계 및 상기 프로세서가 타깃 파라미터 행렬 W 구성이 가능한 로-랭크 행렬 W1 및 W2를 결정하는 단계를 포함한다.

[0007] 상기 타깃 파라미터 행렬 W는 상기 W1 와 상기 W2 사이의 하다마드 곱으로 구성되고, 상기 W1은 m×r1 크기의 X1 행렬 및 n×r1 크기의 Y1 행렬의 내적으로 형성되고, 상기 W2는 m×r2 크기의 X2 행렬 및 n×r2 크기의 Y2 행렬의 내적으로 형성된다. 여기서, m, n, r1, r2는 자연수이다.

[0008] 다른 측면에서 프로세서 및 상기 프로세서가 신경망 모델을 이용한 추론 과정을 제어하는 명령어를 저장하는 메모리를 포함하는 장치가 상기 신경망 모델의 계층들 중 상기 데이터 경량화 방법으로 경량화된 타깃 계층에 대하여 다음 과정을 수행하되, 신경망 모델을 이용한 추론 방법은 상기 메모리가 상기 타깃 계층의 타깃 파라미터 행렬을 구성하는 로-랭크 행렬 W_1 및 W_2를 입력받는 단계, 상기 프로세서가 상기 로-랭크 행렬 W_1 및 상기 W_2에 대한 하다마드 곱을 수행하여 상기 타깃 파라미터 행렬 W를 구성하는 단계 및 상기 프로세서가 입력 데이터 또는 상기 타깃 계층의 이전 계층에서 전달되는 특징을 상기 타깃 계층에 입력하여 추론을 진행하는 단계를 포함한다.

[0009] 또 다른 측면에서 프로세서 및 상기 프로세서가 전역 데이터 및 기기 종속적 특성을 갖는 지역 데이터를 이용하여 동작하는 명령어를 저장하는 메모리를 포함하는 장치가 아래 과정을 수행하되, 데이터 개인화 방법은 상기 메모리가 청구항 1항 내지 청구항 5항 중 어느 하나의 항에 따른 데이터 경량화 방법으로 경량화되는 과정에서 결정되는 상기 전역 데이터에 대한 로-랭크 행렬 W_1 및 상기 지역 데이터에 대한 로-랭크 행렬 W_2를 입력받는 단계, 상기 프로세서가 상기 W_1 와 상기 W_2에 대한 하다마드 곱으로 타깃 파라미터 행렬 W를 생성하는 단계 및 상기 프로세서가 상기 W를 이용하여 신경망 모델 학습 또는 신경망 모델을 이용한 추론을 수행하는 단계를 포함한다. 상기 전역 데이터에 대한 로-랭크 행렬 W_1은 상기 장치가 아닌 다른 적어도 하나의 장치에서 전역 데이터를 이용하여 학습된 결과로 산출된다.

발명의 효과

[0010] 이하 설명하는 기술은 파라미터들의 개수가 줄어들면서도 이론적으로 풀-랭크를 달성할 수 있다. 따라서, 이하 설명하는 기술은 신경망 학습이나 추론과 같은 애플리케이션에서 모델의 표현력(model capacity)을 유지하면서도 통신량을 줄인다. 나아가 이하 설명하는 기술은 경량화 과정에서 도출되는 개별 행렬들을 전역적 정보와 지역적(개인적) 정보로 구분하여 데이터 개인화에 활용할 수도 있다.

도면의 간단한 설명

[0011] 도 1은 일반적인 로-랭크 파라미터화에 대한 예이다.

도 2는 풀-랭크 축소 파라미터화에 대한 예이다.

도 3은 풀-랭크 축소 파라미터화가 적용된 연합 학습에 대한 예이다.

도 4는 풀-랭크 축소 파라미터화를 이용한 모델 개인화 기법에 대한 예이다.

도 5는 풀-랭크 축소 파라미터화를 이용한 모델 개인화 기법의 성능을 검증한 결과이다.

도 6은 풀-랭크 축소 파라미터화가 적용된 분산 학습에 대한 예이다.

도 7은 풀-랭크 축소 파라미터화가 적용된 신경망 추론 기법에 대한 예이다.

도 8은 풀-랭크 축소 파라미터화가 적용된 데이터 압축 기법에 대한 예이다.

도 9는 풀-랭크 축소 파라미터화를 수행하는 데이터 처리 장치에 대한 예이다.

도 10은 경량화된 데이터를 이용하여 일정한 애플리케이션을 수행하는 클라이언트에 대한 예이다.

발명을 실시하기 위한 구체적인 내용

[0012] 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0013] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0014] 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설명된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.

[0015] 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이

하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

[0016] 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

[0018] 이하 설명하는 기술은 특정 데이터 구조 내지 모델에서 파라미터들을 줄이는 경량화 기술이다. 설명의 편의를 위하여 이하 설명은 신경망 네트워크에서의 모델 압축을 중심으로 설명한다. 다만, 이하 설명하는 기술은 학습 모델 계층의 파라미터들을 줄이는 애플리케이션으로 제한되지는 않는다. 나아가, 이하 설명하는 기술이 적용될 수 있는 다양한 애플리케이션에 대해서도 후술한다.

[0020] 먼저, 일반적인 로-랭크 파라미터화에 대하여 설명한다.

[0021] 도 1은 일반적인 로-랭크 파라미터화에 대한 예이다.

[0022] 로-랭크 파라미터화는 어떤 디지털 데이터나 모델을 구성하는 파라미터들의 개수를 줄이는 기술이다. 로-랭크 파라미터화는 파라미터들의 개수를 줄여 데이터나 모델의 크기를 줄일 수 있다.

[0023] 로-랭크 분해(low-rank decomposition)는 인코딩되는 정보의 손실을 최소화하면서 파라미터들의 개수를 줄이는 기술이다. 로-랭크 파라미터화는 신경망에서 사전 학습된(pre-trained) 모델의 압축에 해당하는 행렬 분해(matrix decomposition)에 활용될 수 있다. 행렬 분해는 컨볼루션 계층들(convolution layers)의 커널 및 전연결 계층들(Fully connected layers, 이하 FC 계층) 등에 적용가능하다.

[0024] 학습된 파라미터 행렬 $\mathbf{W} \in \mathbb{R}^{m \times n}$이 있다고 가정한다. 해당 파라미터 행렬에 대한 최적 랭크 r 추정 과정은 $\arg\min_{\widetilde{\mathbf{W}}} ||\mathbf{W} - \widetilde{\mathbf{W}}||_F$으로 표현될 수 있다. 여기서 $\widetilde{\mathbf{W}} = \mathbf{X}\mathbf{Y}^\top$이다. $\mathbf{X} \in \mathbb{R}^{m \times r}$및 $\mathbf{Y} \in \mathbb{R}^{n \times r}$이고, r ≪min(m,n)이다. ┬는 전치행렬을 말한다. 로-랭크 파라미터화는 파라미터들의 개수(행렬의 복잡도)를 O(mn)에서 O((m+n)r)로 감소시킨다. 이때 최적의 r은 특이값 분해(Single Value Decomposition)로 찾을 수 있다.

[0025] 도 1을 살펴보면, 로-랭크 파라미터화는 랭크 1의 행렬들 2R 개를 합한 결과에 해당한다. 도 1은 로-랭크 파라미터화에 대한 하나의 예이다. 도 1은 후술하는 새로운 파라미터화 기법의 설명의 용의하게 하기 위하여 편의상 2R인 경우를 예시한 것이다. 로-랭크 파라미터화의 랭크는 rank(W) ≤ 2R의 값을 갖는다. 일반적인 로-랭크 파라미터화는 로-랭크 제한(low-rank constraints)때문에 표현력(expressiveness)이 제한된다. 예컨대, 신경망에 행렬 분해를 적용하면 신경망 모델은 낮은 성능을 보일 수 있다.

[0027] 종래 로-랭크 파라미터화는 전술한 바와 같이 랭크 제한이라는 한계를 갖는다. 이하 설명하는 기술은 모델의 파라미터들을 줄이면서도 로-랭크 제한이 없는 경량화 기술이다. 로-랭크 제한이 없는 파라미터화라는 의미에서 이하 설명하는 기술을 풀-랭크 축소 파라미터화(Full-rank Reduced Parameterization)라고 명명한다.

[0029] 도 2는 풀-랭크 축소 파라미터화에 대한 예이다. 풀-랭크 축소 파라미터화는 2개의 로-랭크 내적 행렬들(inner matrices)의 하다마드 곱(Hadamard product)에 해당한다. 풀-랭크 축소 파라미터화는 $\mathbf{W} = \mathbf{W}_1 \odot \mathbf{W}_2 = (\mathbf{X}_1\mathbf{Y}_1^\top) \odot (\mathbf{X}_2\mathbf{Y}_2^\top)$로 정의된다. ⊙는 하다마드 곱을 의미한다. 풀-랭크 축소 파라미터화의 랭크는 rank(W) ≤ R^2의 값을 갖는다. R^2이기 때문에, 풀-랭크 축소 파라미터화는 R을 작게 설정하여 파라미터를 적게 쓰더라도 로-랭크 제한이 없다.

[0030] 이하 데이터 처리 장치가 풀-랭크 축소 파라미터화를 수행한다고 설명한다. 데이터 처리 장치는 PC, 스마트기기, 서버, 데이터 처리 프로그램이 임베딩된 칩셋 등 다양한 형태로 구현될 수 있다.

[0032] 이하 풀-랭크 축소 파라미터화가 파라미터들의 개수를 최소화하면서 랭크 제한이 없는 특성(풀-랭크)을 갖는다는 것을 증명한다.

[0034] 명제 1: $\mathbf{X}_1 \in \mathbb{R}^{m\times r_1}$, $\mathbf{X}_2 \in \mathbb{R}^{m\times r_2}$, $\mathbf{Y}_1 \in \mathbb{R}^{n\times r_1}$, $\mathbf{Y}_2 \in \mathbb{R}^{n\times r_2}$, r_1 및 $r_2 \leq \min(m,n)$에서, 구성된 행렬이 $\mathbf{W} := (\mathbf{X}_1\mathbf{Y}_1^\top) \odot (\mathbf{X}_2\mathbf{Y}_2^\top)$이면, $\text{rank}(W) \leq r_1 r_2$이다.

[0035] 명제 1을 증명한다. $\mathbf{X}_1\mathbf{Y}_1^\top$ 및 $\mathbf{X}_2\mathbf{Y}_2^\top$는 랭크 -1 행렬의 합 $\mathbf{X}_i\mathbf{Y}_i^\top = \sum_{j=1}^{j=r_i} \mathbf{x}_{ij}\mathbf{y}_{ij}^\top$으로 표현될 수 있다. x_{ij} 및 y_{ij}는 각각 X_i 및 Y_i의 j번째 컬럼 벡터이다. 여기서 $i \in \{1,2\}$이다. 이 경우 W는 아래 수학식 1과 같이 표현될 수 있다.

수학식 1

[0037] $$\mathbf{W} = \mathbf{X}_1\mathbf{Y}_1^\top \odot \mathbf{X}_2\mathbf{Y}_2^\top = \sum_{j=1}^{j=r_1} \mathbf{x}_{1j}\mathbf{y}_{1j}^\top \odot \sum_{j=1}^{j=r_2} \mathbf{x}_{2j}\mathbf{y}_{2j}^\top = \sum_{k=1}^{k=r_1}\sum_{j=1}^{j=r_2} (\mathbf{x}_{1k}\mathbf{y}_{1k}^\top) \odot (\mathbf{x}_{2j}\mathbf{y}_{2j}^\top)$$

[0038] W는 $r_1 r_2$ 개의 랭크 1인 행렬의 합산에 해당한다. W는 $r_1 \times r_2$ 개의 행렬로 구성되기에 서로 독립인 벡터들이 최대 $r_1 \times r_2$ 개이다. 따라서 rank(W)는 최대 $r_1 r_2$ 의 랭크를 갖는다.

[0040] 명제 1은 풀-랭크 축소 파라미터화가 두 개의 내적 로-랭크 행렬들 W_1 및 W_2의 하다마드 곱을 통해 높은 랭크 행렬을 산출한다는 의미이다. 만약, $r_1 r_2 \geq \min(m,n)$를 만족하는 로-랭크 r_1 및 r_2 를 선택한다면, 구성된 행렬은 풀-랭크를 달성할 수 있다.

[0042] 명제 2: $R \in \mathbb{N}$(자연수)이면, $r_1 = r_2 = R$은 아래 수학식 2에 따른 기준의 고유 최적 선택이고, 최적값은 $2R(m+n)$이다.

수학식 2

[0043] $$\arg\min_{r_1,r_2\in\mathbb{N}} (r_1+r_2)(m+n) \quad \text{s.t.} \quad r_1 r_2 \geq R^2$$

[0045] 명제 2를 증명한다. 산술-기하 평균 부등식 및 주어진 제한 조건을 사용하여 아래 수학식 3과 같은 결과를 얻을 수 있다.

수학식 3

[0047] $$(r_1+r_2)(m+n) \geq 2\sqrt{r_1 r_2}(m+n) \geq 2R(m+n)$$

[0048] 상기 산술-기하 평균 부등식에서 $r_1 = r_2 = R$인 경우에만 동일한 값(=)을 갖는다.

[0050] 명제 2는 구성된 행렬 R^2의 랭크 제한을 사용한 풀-랭크 축소 파라미터화가 가중 파라미터들의 개수가 최소임을 나타낸다. 즉, 명제 2는 R^2로 만들 수 있는 r_1 및 r_2가 많지만, 두 개의 로-랭크 행렬의 랭크를 같게 설정하면 ($r_1 = r_2$) 최대 랭크(R^2)를 달성하면서 파라미터들의 개수가 최소임을 나타낸다. 명제 2는 하이퍼 파라미터들을 설정하는 효율적인 방법을 제시한다. 즉, $r_1 = r_2 = R$ 및 $R^2 \geq \min(m,n)$로 설정하면, 풀-랭크 축소 파라미터화는 제곱의 성질로 R을 작게 설정해도 되기 때문에, 로-랭크 제한 조건 없이 통상적인 경우(naive)보다 훨씬 적은 파라미터들 개수($2R(m+n) \ll mn$)를 보장할 수 있다.

[0052] 나아가, 동일한 파라미터들의 개수가 주어진다면 풀-랭크 축소 파라미터화의 rank(W)는 도 2와 같이 종래 로-랭크 파라미터화보다 제곱 계수만큼 높은 값을 갖는다.

[0054] 이하 풀-랭크 축소 파라미터화가 적용 가능한 다양한 애플리케이션에 대하여 설명한다. 물론, 풀-랭크 축소 파라미터화는 모델이나 데이터 구조의 경량화를 위한 것으로 아래 설명하는 실시예뿐만 아니라 다른 다양한 분야에도 적용될 수 있다.

[0056] 연구자는 전술한 풀-랭크 축소 파라미터화를 연합 학습(Federated Learning)에 적용하여 그 성능을 검증하였다.

[0057] 명제 1은 컨볼루션 계층의 텐서에도 적용할 수 있다. 연구자는 텐서 커널을 행렬 $\mathbb{R}^{O\times I\times K_1\times K_2} \rightarrow \mathbb{R}^{O\times (IK_1K_2)}$와 같이 재구성하였다. 여기서, O는 출력 채널들, I는 입력 채널들, K_1 및 K_2는 커널 크기를 의미한다. 즉, 풀-랭크 축소 파라미터화는 크기 $I \times K_1 \times K_2$의 기본 필터들을 확장한다. 모델 개발자는 내적 랭크 r_1 및 r_2를 각각 변경하여 파라미터들의 개수를 조절할 수 있다.

[0058] 데이터 처리 장치는 전술한 명제 1에 따라 내적 랭크 r_1 및 r_2를 설정하여 풀-랭크를 달성할 수 있다. 또한, 데이터 처리 장치는 전술한 명제 2에 따라 내적 랭크 r_1 및 r_2를 설정하여 최소의 파라미터들을 이용하여 최대 랭크를 달성할 수도 잇다.

[0059] 또한, 연구자는 풀-랭크 축소 파라미터화를 아래와 같이 텐서 구조에 추가로 확장하였다. 텐서를 파라미터화할 수 있는 방법은 다양하며, 연구자는 전술한 바와 같이 재구성(reshape)하고 텐서 구조를 그대로 사용하는 방법을 제안한다. 다만 아래 설명하는 텐서 구조 확장은 풀-랭크 축소 파라미터화에 대한 하나의 실시예이다.

[0061] 명제 3: $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{R}^{R\times R\times k_3\times k_4}$, $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^{k_1\times R}$, $\mathbf{Y}_1, \mathbf{Y}_2 \in \mathbb{R}^{k_2\times R}$, $R \leq \min(k_1,k_2)$이라면, 컨볼루션 커널은 $\mathcal{W} := (\mathcal{T}_1 \times_1 \mathbf{X}_1 \times_2 \mathbf{Y}_1) \odot (\mathcal{T}_2 \times_1 \mathbf{X}_2 \times_2 \mathbf{Y}_2)$와 같이 표현된다. 이때 커널은 $\operatorname{rank}(\mathcal{W}^{(1)}) = \operatorname{rank}(\mathcal{W}^{(2)}) \leq R^2$을 만족한다. T_1 및 T_2는 텐서이다. k는 자연수이다.

[0062] 명제 3을 증명한다. 첫 번째 및 두 번째 언폴딩된 텐서는 아래 수학식 4와 같이 표현될 수 있다.

수학식 4

[0064] $$\mathcal{W}^{(1)} = (\mathbf{X}_1\mathcal{T}_1^{(1)}(\mathbf{I}^{(4)} \otimes \mathbf{I}^{(3)} \otimes \mathbf{Y}_1)^\top) \odot (\mathbf{X}_2\mathcal{T}_2^{(1)}(\mathbf{I}^{(4)} \otimes \mathbf{I}^{(3)} \otimes \mathbf{Y}_2)^\top),$$
$$\mathcal{W}^{(2)} = (\mathbf{Y}_1\mathcal{T}_1^{(2)}(\mathbf{I}^{(4)} \otimes \mathbf{I}^{(3)} \otimes \mathbf{X}_1)^\top) \odot (\mathbf{Y}_2\mathcal{T}_2^{(2)}(\mathbf{I}^{(4)} \otimes \mathbf{I}^{(3)} \otimes \mathbf{X}_2)^\top),$$

[0065] $\mathbf{I}^{(3)} \in \mathbb{R}^{k_3 \times k_3}$ 및 $\mathbf{I}^{(4)} \in \mathbb{R}^{k_4 \times k_4}$는 단위 행렬(identity matrix)이다. $\otimes$는 크로네커 곱(Kronecker product)이다. $\mathcal{W}^{(1)}$ 및 $\mathcal{W}^{(2)}$는 행렬이기 때문에 수학식 1을 적용할 수 있고, 그 결과 $\text{rank}(\mathcal{W}^{(1)}) = \text{rank}(\mathcal{W}^{(2)}) \leq R^2$임을 알 수 있다.

[0066] 명제 3은 명제 1의 확장이라고 할 수 있다. 나아가, 풀-랭크 축소 파라미터화는 컨볼루션 계층의 재구성(reshaping) 없이 컨볼루션 계층 설계에 사용될 수 있다. 따라서, 랭크 관점에서 보면 풀-랭크 축소 파라미터화는 로-랭크 제한이 없어 성능을 유지할 수 있다.

[0068] 연합 학습은 프라이버시에 민감한 지역 데이터를 중앙 서버에서 처리하지 않고, 모델 학습을 클라이언트들에 분산하여 협력적으로 처리하는 방식이다. 클라이언트들은 주로 스마트폰, IoT 기기 등이다.

[0069] 도 3은 풀-랭크 축소 파라미터화가 적용된 연합 학습에 대한 예이다. 도 3은 풀-랭크 축소 파라미터화가 적용된 연합 학습 시스템(100)에 대한 예이다.

[0070] 먼저, 데이터 처리 장치는 풀-랭크 축소 파라미터화를 이용한 신경망 경량화 과정을 수행한다. 데이터 처리 장치는 컨볼루션 계층들 및/또는 전연결 계층들을 경량화할 수 있다. 데이터 처리 장치는 컨볼루션 계층들 중 적어도 일부 계층들의 파라미터들을 줄일 수 있다. 또한, 데이터 처리 장치는 전연결 계층들 중 적어도 일부 계층들의 파라미터들을 줄일 수도 있다. 데이터 처리 장치는 풀-랭크 축소 파라미터화을 통해 신경망을 압축한다. 이때 데이터 처리 장치는 전술한 명제 1 내지 명제 2에 따라 압축정도(=줄어드는 파라미터 개수)를 조절할 수 있다. 한편, 데이터 처리 장치는 모델 개발에 사용한 컴퓨터 장치 또는 도 3의 서버(110)일 수도 있다.

[0071] 서버(110)는 연합 학습에 사용할 전역 신경망 모델을 구축하거나, 수신한다. 이때 신경망 모델은 풀-랭크 축소 파라미터화를 통해 경량화된 계층으로 구성된 모델이다.

[0072] 서버(110)는 기본적으로 학습 스케쥴을 관리하고, 학습 과정을 제어한다. 서버(110)는 경량화된 전역 신경망 모델을 클라이언트들(120)에 배포한다. 클라이언트들(120)은 각각 전역 신경망 모델을 다운로드하고, 각자의 지역(local) 데이터를 이용하여 전역 신경망 모델에 대한 지역적 학습을 수행한다. 클라이언트들(120)은 서버로부터 수신한 파라미터들에 하다마드 곱을 적용하여 파라미터 행렬을 구성하고, 구성한 파라미터 행렬을 기준으로 학습을 수행할 수 있다.

[0073] 서버(110)는 전역 신경망 모델 학습을 위한 클라이언트들을 샘플링(선택)한다. 선택된 클라이언트들은 서버(110)로부터 전역 모델을 다운로드받고, 자신이 지역적으로 학습한 신경망 (계층)을 서버(110)에 업로드한다. 서버(110)는 샘플링한 클라이언트들로부터 수신한 학습된 신경망들을 통합(aggregation)한다.

[0074] 한편, 연합 학습은 다양한 알고리즘 중 어느 하나가 적용될 수 있다. 전술한 풀-랭크 축소 파라미터화는 학습 알고리즘과 관계 없이 신경망 계층을 사전 분해(decomposition)하는 것이다. 따라서, 풀-랭크 축소 파라미터화는 다양한 학습 모델이나 학습 알고리즘에 적용될 수 있다.

[0075] 결국, 다운로드 및 업로드에 필요한 모델 크기 자체가 줄어들기 때문에 풀-랭크 축소 파라미터화가 적용된 연합 학습은 서버와 클라이언트 사이의 통신 비용이 저감된다. 또한, 풀-랭크 축소 파라미터화가 적용된 연합 학습은 한정된 자원을 갖는 클라이언트의 한계를 완화하는데 도움이 된다.

[0077] 아래 표 1은 신경망 모델의 계층에서 풀-랭크 축소 파라미터화의 성능을 실험한 결과이다. 연구자는 VGG16 모델을 사용하였고, CIFAR-10, CIFAR-100 및 CINIC-10 데이터 세트를 사용하였다. 표 1은 파라미터 변경이 없는 원본 모델(Original), 로-랭크 파라미터화가 적용된 모델(Low-rank) 및 풀-랭크 축소 파라미터화가 적용된 모델(FedPara)에 대한 결과를 도시한다. 표 1은 파라미터 경량화 방식(paramerization)에 따른 파라미터들의 개수(#

Params), 최대 랭크(Maximal Rank) 등을 표시한 예이다. 표 1은 신경망 모델의 전연결 계층(FC layer) 및 컨볼루션 계층(Convolutional Layer)에서의 결과를 각각 나타낸다. 표 1에서 전연결 계층 및 컨볼루션 계층의 가중치는 각각 $\mathbb{R}^{m\times n}$ 및 $\mathbb{R}^{O\times I\times K_1\times K_2}$으로 가정하였다. 컨볼루션 계층의 랭크는 첫 번째 언폴딩 텐서의 랭크이다. 샘플은 m = n = O = I = 256, $K_1 = K_2 = 3$, R = 16인 예이다.

표 1

[0079]

Layer	Parameterization	# Params.	Maximal Rank	Example [# Params. / Rank]
FC Layer	Original	mn	$\min(m,n)$	66 K / 256
	Low-rank	$2R(m+n)$	$2R$	16 K / 32
	FedPara	$2R(m+n)$	R^2	16 K / 256
Convolutional Layer	Original	OIK_1K_2	$\min(O, IK_1K_2)$	590 K / 256
	Low-rank	$2R(O+I+RK_1K_2)$	$2R$	21 K / 32
	FedPara (Proposition 1)	$2R(O+IK_1K_2)$	R^2	82 K / 256
	FedPara (Proposition 3)	$2R(O+I+RK_1K_2)$	R^2	21 K / 256

[0081] 표 1을 살펴보면, Low-rank는 파라미터들의 개수는 줄지만 최대 랭크가 낮고, FedPara는 파라미터 개수가 줄고 R을 잘 선택하면 Original과 같은 랭크를 달성함을 알 수 있다. 컨볼루션 계층의 경우 명제 1을 적용한 FedPara는 풀-랭크를 유지하면서 파라미터들의 개수가 Low-rank보다 다소 높지만, 명제 3을 적용한 Fedpara는 Low-rank와 동일하게 파라미터들의 개수를 줄이면서도 높은 랭크를 달성함을 알 수 있다.

[0083] 전술한 풀-랭크 축소 파라미터화를 이용한 연합 학습 과정은 아래 표 2의 알고리즘 1과 같이 정리할 수 있다. 아래 알고리즘 1은 특정 계층에서 풀-랭크 축소 파라미터화를 적용하기 위한 행렬이 결정된 상태라고 가정한다.

표 2

[0085]

Algorithm 1

Input: rounds T, parameters $\mathbf{X}_1, \mathbf{X}_2, \mathbf{Y}_1, \mathbf{Y}_2$

for $t = 1, 2, \ldots, T$ **do**
  Sample the subset S of clients;
  Transmit $\mathbf{X}_1, \mathbf{X}_2, \mathbf{Y}_1, \mathbf{Y}_2$ to clients;
  for $c \in S$ **do**
    $\mathbf{W} = (\mathbf{X}_1\mathbf{Y}_1^\top) \odot (\mathbf{X}_2\mathbf{Y}_2^\top)$;
    Optimizer($\mathbf{W}$);
    Upload $\mathbf{X}_1, \mathbf{X}_2, \mathbf{Y}_1, \mathbf{Y}_2$;
  end
  Aggregate $\mathbf{X}_1, \mathbf{X}_2, \mathbf{Y}_1, \mathbf{Y}_2$;
end

[0087] 알고리즘 1을 간략하게 설명한다. 서버는 경량화된 신경망을 샘플링한 S개의 클라이언트들에 전달한다. 서버는 파라미터 X_1, X_2, Y_1 및 Y_2를 각 클라이언트에 전달한다.

[0088] S개의 클라이언트들은 각각 자신의 지역 데이터를 이용하여 신경망 학습 과정을 수행한다. 이때, 클라이언트는 도 2에서 설명한 바와 같이 파라미터들이 구성하는 내적 행렬 W_1 및 W_2에 하다마드 곱을 연산하여 파라미터 행렬 W를 구성한다. 클라이언트는 파라미터 행렬 W를 기준으로 학습을 수행한다. 지역 학습이 완료되면 클라이언트는 학습된 파라미터 X_1, X_2, Y_1 및 Y_2를 서버에 업로드한다. 그리고, 서버는 모든 클라이언트들로부터 전달받은 학습

된 파라미터를 통합하여 모델을 구축한다.

[0090] 연합 학습에 참여하는 클라이언트는 기기의 데이터가 서로 다른 개인적 특징을 갖기 때문에, 일반적인 학습에 사용되는 IID (independent and identically distributed)가정이 성립하지 않는다.

[0092] 이때 어느 하나의 클라이언트가 연합 학습으로 구축된 모델을 사용하면 해당 기기에서 추론 성능이 낮을 수 있다. 모델 개인화(Personalization)는 클라이언트별(사용자별) 맞춤형 결과를 산출하기 위한 기법이다. 종래 개인화 기법은 모델의 출력단의 계층을 개인화하여 사용자 데이터에 맞는 결과를 생성하도록 하였다. 종래 개인화 기법은 출력단을 제외한 계층이 개인화되지 않는 한계가 있었다.

[0093] 도 4는 풀-랭크 축소 파라미터화를 이용한 모델 개인화 기법에 대한 예이다. 도 4는 도 2의 풀-랭크 축소 파라미터화의 특징적 구조를 이용한 개인화 기법에 해당한다. 도 4는 연합 학습 시스템(200)을 예로 설명한다. 도 4를 살펴보면 풀-랭크 축소 파라미터화로 압축된 모델 계층들은 전역 파라미터들과 지역 파라미터들로 구성된다. 하나의 계층에서 파라미터 행렬 W는 전역 가중치 W_1와 지역 가중치 W_2 사이의 하다마드 곱으로 표현될 수 있다.

[0094] 서버(210)는 사전에 연합 학습을 위한 신경망 모델을 클라이언트들에 배포하였고, 클라이언트들은 각각의 지역 데이터로 신경망 모델을 학습한다고 전제한다. 도 4는 설명의 편의를 위하여 하나의 클라이언트(220)를 도시하였다. 클라이언트(220)는 학습 과정에서 지역 데이터를 사용하여 신경망 계층의 파라미터 행렬을 학습한다. 이때 파라미터 행렬 W는 전역 가중치 W_1와 지역 가중치 W_2 사이의 하다마드 곱으로 산출된 행렬이다. 클라이언트(220)는 학습이 종료되면 W_1만을 서버에 전달하고, W_2는 내부에 보유한다. 이후 서버(210)가 신경망 파라미터를 통합하여 완성된 전역 가중치를 클라이언트(220)에 전달할 수도 있다.

[0095] 클라이언트(220)에서 최종 W는 클라이언트의 개인 가중치와 전역 가중치의 합으로 표현할 수 있다. $W = W_1 \odot W_2 + W_1 = W_{per} + W_{glo}$, $W_{per} = W_1 \odot W_2$, $W_{glo} = W_1$이다. 이와 같은 모델을 사용하면 클라이언트(220)는 전역적으로는 연합 학습에 의한 높은 추론 성능을 갖고, 지역적으로는 자신의 개인 특성에 맞는 결과를 산출할 수 있다.

[0096] 한편, 도 4는 연합 학습 시스템을 전제로 설명하였지만, 도 4의 개인화 기법은 신경망외에도 다른 데이터 또는 특징을 처리하는 장치에서도 활용될 수 있다.

[0098] 연구자는 풀-랭크 축소 파라미터화를 이용한 모델 개인화 기법에 대한 성능을 검증하였다. 도 5는 풀-랭크 축소 파라미터화를 이용한 모델 개인화 기법의 성능을 검증한 결과이다. 도 5는 신경망 모델의 정확도(accuracy)에 대한 예이다. 연구자는 VGG16에 기반하여 영상 데이터를 분류하는 모델을 구축하였다. 도 6에서 'Local'은 연합 학습 기반 모델이 아닌 개별 모델, 'FedAvg'는 종래 대표적인 연합 학습 모델, 'FedPer'은 종래 대표적인 개인화 모델, 'pFedPara'는 풀-랭크 축소 파라미터화 기반의 개인화 모델을 의미한다. pFedPara가 도 4에서 설명한 개인화 기법으로 구축한 모델을 의미한다.

[0099] 도 5는 3개의 시나리오에 대한 결과를 나타낸다. 연구자는 FEMNIST를 각 클라이언트에 분배한 후 검증을 하였다. 도 5(A)는 Non-IID(not independent and identically distributed) 설정된 학습 데이터 FEMNIST의 local data를 100% 이용하여 지역(local)에서 학습한 경우와 비교한 결과이다. 충분한 학습 데이터를 사용하였기 때문에 local의 경우가 FedAvg 및 Fedper보다 높은 정확도를 보였다. 그러나 pFedPara가 local보다도 높은 정확도를 나타냈다. 도 5(B)는 Non-IID 설정된 학습 데이터 FEMNIST를 20% 이용하여 로컬에서 학습한 경우와 비교한 결과이다. 즉, 학습 데이터가 부족한 경우에 해당한다. 학습 데이터가 부족하였기 때문에, local은 FedAvg보다도 낮은 정확도를 보였다. FerPer은 FedAvg보다 낮은 정확도를 보였는데 이는 로컬 데이터가 부족한 경우 성능이 다소 떨어지는 것을 나타낸다. 그러나 pFedPara는 도 5(B)의 환경에서도 가장 높은 정확도를 보였다. 도 5(C)는 highly-skew Non-IID 설정으로 학습 데이터 MNIST를 100% 이용하여 로컬에서 학습한 경우와 비교한 결과이다. FedAvg는 다른 모델에 비하여 현저하게 낮은 정확도를 나타냈는데 이는 매우 편향된 데이터 분포를 갖는 경우 정확도가 낮아짐을 나타낸다. 다른 모델은 모두 비교적 높은 정확도를 보였다. 도 5의 결과를 살펴보면, pFedPara는 모든 경우에 거의 가장 높은 성능을 보였다. pFedPara는 연합 학습이 아닌 단일 기기에서 학습된 모델 보다도 높은 성능을 보였다.

[0101] 도 6은 풀-랭크 축소 파라미터화가 적용된 분산 학습에 대한 예이다. 도 6은 풀-랭크 축소 파라미터화가 적용된 분산 학습 시스템(300)에 대한 예이다.

[0102] 분산 학습 시스템(300)은 복수의 GPU들(320)을 이용하여 병렬적으로 학습을 수행한다. 중앙 제어 장치(310)는 학습을 수행하는 GPU와는 다른 연산 장치일 수도 있고, GPU들 중 어느 하나일 수도 있다. 분산 학습 시스템(300)은 복수의 GPU들을 이용하여 모델을 분할하거나 데이터를 분할하여 학습을 수행한다. 초거대 신경망 모델은 파라미터들이 매우 많기 때문에, 해당 모델에 대한 분산 학습 과정은 그레디언트(gradient)를 공유과정에서 네트워크 병목 현상이 발생할 수 있다.

[0103] 먼저, 데이터 처리 장치는 풀-랭크 축소 파라미터화를 이용한 신경망 경량화 과정을 수행한다. 데이터 처리 장치는 컨볼루션 계층들 및/또는 전연결 계층들을 경량화할 수 있다. 데이터 처리 장치는 컨볼루션 계층들 중 적어도 일부 계층들의 파라미터들을 줄일 수 있다. 또한, 데이터 처리 장치는 전연결 계층들 중 적어도 일부 계층들의 파라미터들을 줄일 수도 있다. 데이터 처리 장치는 풀-랭크 축소 파라미터화를 통해 신경망을 압축한다. 이때 데이터 처리 장치는 전술한 명제 1 내지 명제 3에 따라 압축정도(=줄어드는 파라미터 개수)를 조절할 수 있다. 한편, 데이터 처리 장치는 모델 개발에 사용한 컴퓨터 장치 또는 도 6의 중앙 제어 장치(310)일 수 있다.

[0104] 다수의 GPU들(320)은 각각 학습에 사용할 신경망 모델과 데이터를 갖고 있다고 전제한다. 다수의 GPU들(320) 각각은 샘플링된 데이터를 기준으로 할당된 신경망 학습을 수행한다.

[0105] 그레디언트 통합 과정은 하나의 중앙 제어 장치(310)가 모든 그레디언트를 통합하는 방식 또는 전체 GPU들(320)이 링(ring) 형태로 그레디언트를 전달하면서 공유하는 방식이 있다. 도 6은 중앙 제어 장치(=파라미터 서버)가 그레디언트를 통합하는 예를 도시한다. GPU들(320)은 풀-랭크 축소 파라미터화로 경량화된 신경망 계층의 파라미터 행렬 내지 파라미터를 중앙 제어 장치(310)에 전달한다. 도 6과 달리 GPU들(320)이 그레디언트를 공유하는 경우에도 어느 하나의 GPU는 다른 GPU에 자신이 업데이트한 파라미터 행렬 내지 파라미터를 전달한다. 따라서, 풀-랭크 축소 파라미터화가 적용된 분산 학습은 네트워크 통신양이 줄어들어 통신 비용이 줄어든다.

[0107] 도 7은 풀-랭크 축소 파라미터화가 적용된 신경망 추론 기법에 대한 예이다. 도 7은 풀-랭크 축소 파라미터화가 적용된 신경망 모델을 이용하는 신경망 추론 시스템(400)에 대한 예이다. 신경망 추론 시스템(400)은 제한된 메모리 용량을 갖는 하드웨어일 수 있다. 신경망 추론 시스템(400)은 신경망 계층을 한 번에 읽어와서 추론하기 어려울 수 있다. 신경망 추론 시스템(400)은 메모리 크기에 맞게 신경망의 파라미터를 읽어와서 데이터를 처리하고 다음 계층의 데이터를 처리하는 과정을 반복해야 한다. 신경망 추론 시스템(400)은 저장장치(410)에서 메모리(420)까지 파라미터들을 읽어오는 통신 시간이 필요하다. 신경망 추론 시스템(400)은 연산처리장치(420)에서 수행하는 연산시간 보다 통신 시간이 더 많이 소요될 수 있다. 이 경우 풀-랭크 축소 파라미터화를 통하여 신경망 모델을 경량화하면 통신 비용을 줄어들어 신경망 추론 시스템(400)의 추론 성능이 향상될 수 있다.

[0108] 먼저, 데이터 처리 장치는 풀-랭크 축소 파라미터화를 이용한 신경망 경량화 과정을 수행한다. 데이터 처리 장치는 컨볼루션 계층들 및/또는 전연결 계층들을 경량화할 수 있다. 데이터 처리 장치는 컨볼루션 계층들 중 적어도 일부 계층들의 파라미터들을 줄일 수 있다. 또한, 데이터 처리 장치는 전연결 계층들 중 적어도 일부 계층들의 파라미터들을 줄일 수도 있다. 데이터 처리 장치는 풀-랭크 축소 파라미터화를 통해 신경망을 압축한다. 이때 데이터 처리 장치는 전술한 명제 1 내지 명제 3에 따라 압축정도(=줄어드는 파라미터 개수)를 조절할 수 있다. 한편, 데이터 처리 장치는 모델 개발에 사용한 컴퓨터 장치일 수 있다.

[0109] 저장장치(410)는 풀-랭크 축소 파라미터화로 행렬 분해된 신경망 계층 데이터를 저장한다. 메모리(420)는 저장장치(410)로부터 파라미터 행렬 W를 구성할 수 있는 파라미터 W_1와 파라미터 W_2를 읽어온다. 연산처리장치(420)는 파라미터 W_1와 파라미터 W_2에 대한 하다마드 곱으로 파라미터 행렬 W를 구하여 해당 계층을 구성한다. 연산처리장치(420)는 구성한 계층을 이용하여 추론 과정을 수행할 수 있다.

[0111] 도 8은 풀-랭크 축소 파라미터화가 적용된 데이터 압축 기법에 대한 예이다. 도 8은 풀-랭크 축소 파라미터화가 적용된 데이터 압축 시스템(500)에 대한 예이다. 데이터 압축 시스템(500)은 사진, 비디오, 음향 등을 압축하는 장치일 수 있다.

[0112] 저장장치(510)는 디지털 콘텐츠의 로(raw) 데이터를 저장한다.

[0113] 연산 처리 장치(520)는 기본적으로 저장하고자하는 품질에 따라 표준화된 압축방식으로 데이터를 1차적으로 압축한다. 압축 방식은 JPEG, MPEG, HEVC 등 콘텐츠 종류 및 코딩 프로토콜에 따라 다양할 수 있다.

[0114] 이때 1차 압축된 데이터의 구조는 일종의 벡터로 볼 수 있다. 따라서, 압축된 데이터 구조에 전술한 풀-랭크 축소 파라미터화를 적용할 수 있다. 연산 처리 장치(520)가 1차 압축된 데이터를 풀-랭크 축소 파라미터화로 경량화하여 데이터 양을 줄일 수 있다.

[0116] 도 9는 풀-랭크 축소 파라미터화를 수행하는 데이터 처리 장치(600)에 대한 예이다. 데이터 처리 장치(600)는 데이터 또는 신경망 모델을 경량화하는 장치이다. 데이터 처리 장치(600)는 PC, 서버, 프로그램이 임베딩된 칩셋, 스마트 기기 등 다양한 형태로 구현될 수 있다.

[0117] 데이터 처리 장치(600)는 저장장치(610), 메모리(620), 연산장치(630), 인터페이스 장치(640) 및 통신장치(650)를 포함할 수 있다.

[0118] 저장장치(610)는 경량화 대상인 데이터 구조 또는 신경망 모델을 저장할 수 있다.

[0119] 저장장치(610)는 풀-랭크 축소 파라미터화를 위한 프로그램 내지 코드(명령어, instructions)를 저장할 수 있다.

[0120] 저장장치(610)는 경량화된 데이터, 신경망 계층 또는 파라미터 행렬들을 저장할 수 있다.

[0121] 메모리(620)는 데이터 처리 장치(600)가 풀-랭크 축소 파라미터화를 수행하는 과정에서 생성되는 데이터 및 정보 등을 저장할 수 있다.

[0122] 메모리(620)는 풀-랭크 축소 파라미터화를 수행하면서 저장장치(610)로부터 필요한 신경망 모델, 프로그램 코드 내지 명령어 등을 읽어오게 된다.

[0123] 인터페이스 장치(640)는 외부로부터 일정한 명령 및 데이터를 입력받는 장치이다. 인터페이스 장치(640)는 물리적으로 연결된 입력 장치 또는 외부 저장장치로부터 초기 데이터 또는 신경망 모델을 입력받을 수 있다. 인터페이스 장치(640)는 경량화된 데이터, 신경망 계층 또는 파라미터 행렬들을 다른 객체에 전달할 수도 있다.

[0124] 통신장치(650)는 유선 또는 무선 네트워크를 통해 일정한 정보를 수신하고 전송하는 구성을 의미한다. 통신장치(650)는 외부 객체로부터 초기 데이터 또는 신경망 모델을 수신할 수 있다. 통신장치(650)는 경량화된 데이터, 신경망 계층 또는 파라미터 행렬들을 사용자 단말, 서비스 서버 등과 같은 외부 객체에 송신할 수도 있다.

[0125] 이하 풀-랭크 축소 파라미터화는 도 2, 도 3 등에서 설명한 내용과 수식을 전제로 설명한다.

[0126] 연산 장치(630)는 메모리(620)에 저장된 프로그램 코드(instructions)를 실행하면서 풀-랭크 축소 파라미터화를 수행하게 된다.

[0127] 연산 장치(630)는 도 2에서 설명한 풀-랭크 축소 파라미터화에서 내적 행렬 W_1 및 W_2를 구한다.

[0128] 연산 장치(630)는 전술한 바와 같이 목표하는 압축 정도에 따라 파라미터들의 개수를 조절할 수 있다.

[0129] 연산 장치(630)는 전술한 명제 1 내지 명제 3에 따라 압축정도를 조절할 수 있다. 연산 장치(630)는 도 2와 같은 행렬 구조에서 명제 1(수학식 1)에 따라 $r_1 r_2 \geq \min(m,n)$를 만족하는 로-랭크 r_1 및 r_2 를 선택할 수 있다. 이 경우 논리적으로 해당 파라미터 행렬 W는 풀-랭크를 달성할 수 있다.

[0130] 연산 장치(630)는 도 2와 같은 행렬 구조에서 명제 2(수학식 2 및 수학식 3)에 따라 $r_1 = r_2 = R$ 및 $R^2 \geq \min(m,n)$로 설정하면 최소의 파라미터들을 이용하여 최대 랭크를 달성할 수 있다.

[0131] 연산 장치(630)는 신경망의 컨볼루션 계층에 대하여 명제 3(수학식 1 및 수학식 4)을 만족하는 행렬을 구성하면 컨볼루션 계층 재구성 없이 최소 파라미터들로 풀-랭크를 달성할 수 있다. 이때 행렬 X_1 및 X_2의 크기가 $k_1 \times R$이고, 행렬 행렬 Y_1 및 Y_2의 크기가 $k_2 \times R$이고, T_1 및 T_2 크기는 $R \times R \times k_3 \times k_4$이다.

[0132] 연산 장치(630)는 풀-랭크 축소 파라미터화를 통해 신경망 계층의 파라미터들, 벡터로 표현되는 데이터 구조의

파라미터들 및 행렬 구조로 표현되는 데이터의 파라미터들을 경량화할 수 있다.

[0133] 연산 장치(430)는 데이터를 처리하고, 일정한 연산을 처리하는 프로세서, AP, 프로그램이 임베디드된 칩과 같은 장치일 수 있다.

[0135] 도 10은 경량화된 데이터를 이용하여 일정한 애플리케이션을 수행하는 클라이언트(700)에 대한 예이다. 클라이언트(700)는 연합 학습의 클라이언트, 개인화된 신경망 모델을 이용하는 클라이언트, 분산 학습의 GPU, 신경망 모델을 이용하여 추론하는 AI(인공지능) 프로세서, 차량의 ECU, 디지털 데이터를 압축하는 인코딩 장치 등 다양한 형태로 구현될 수 있다.

[0136] 클라이언트(700)는 저장장치(710), 메모리(720), 연산장치(730) 및 인터페이스 장치(740)를 포함할 수 있다. 나아가, 클라이언트(700)는 통신 장치(750)를 더 포함할 수도 있다.

[0137] 저장장치(710)는 풀-랭크 축소 파라미터화로 경량화된 신경망 모델을 저장할 수 있다. 즉, 저장장치(710)는 전술한 데이터 처리 장치(600)가 결정한 파라미터 행렬을 저장할 수 있다.

[0138] 저장장치(710)는 신경망 모델 학습을 위한 학습 데이터를 저장할 수 있다.

[0139] 저장장치(710)는 신경망 모델을 이용한 추론을 위한 프로그램 내지 코드(명령어)를 저장할 수 있다.

[0140] 메모리(720)는 클라이언트(700)가 신경망 모델을 이용한 추론 과정, 데이터 압축 과정 등에서 생성되는 데이터 및 정보 등을 저장할 수 있다.

[0141] 메모리(720)는 저장장치(710)로부터 필요한 프로그램 코드 내지 명령어를 읽어온다.

[0142] 인터페이스 장치(740)는 외부로부터 일정한 명령 및 데이터를 입력받는 장치이다. 인터페이스 장치(740)는 물리적으로 연결된 입력 장치 또는 외부 저장장치로부터 경량화된 신경망 모델을 입력받을 수 있다. 인터페이스 장치(740)는 학습된 신경망 모델을 위한 입력 데이터를 입력받을 수 있다.

[0143] 인터페이스 장치(740)는 신경망 모델을 이용하여 학습된 파라미터, 신경망 모델을 이용하여 추론된 결과 등을 외부 객체에 전달할 수도 있다.

[0144] 통신장치(750)는 유선 또는 무선 네트워크를 통해 일정한 정보를 수신하고 전송하는 구성을 의미한다. 통신장치(750)는 외부 객체로부터 경량화된 신경망 모델을 수신할 수 있다. 통신장치(750)는 학습된 신경망 모델을 위한 입력 데이터를 수신할 수 있다. 또한, 통신장치(750)는 신경망 모델을 이용하여 학습된 파라미터, 신경망 모델을 이용하여 추론된 결과 등을 사용자 단말, 서비스 서버 등과 같은 외부 객체에 송신할 수도 있다.

[0145] 연산 장치(630)는 메모리(620)에 저장된 프로그램 코드(instructions)를 실행하면서 신경망 학습, 학습된 신경망을 이용한 추론, 데이터 압축 등의 과정을 수행한다.

[0146] 연산 장치(730)는 풀-랭크 축소 파라미터화에서 결정된 내적 행렬 W_1 및 W_2에 대한 하다마드 곱을 수행하여 파라미터 행렬을 구성할 수 있다.

[0147] 연산 장치(730)는 학습 데이터를 이용하여 학습 과정을 진행하면서 구성된 파라미터 행렬의 파라미터를 갱신할 수 있다. 연합 학습의 클라이언트 또는 분산 학습의 GPU가 이와 같은 동작을 수행한다.

[0148] 연산 장치(730)는 학습된 신경망 모델을 이용하여 추론을 수행할 수도 있다. 도 7에서 설명한 바와 같이 연산 장치(730)는 행렬 분해된 내적 행렬 W_1 및 W_2에 대한 하다마드 곱을 수행하여 신경망 계층을 구성하고, 구성된 계층에 입력 데이터 또는 이전 계층의 출력을 입력하여 추론을 수행할 수 있다.

[0149] 연산 장치(730)는 데이터를 처리하고, 일정한 연산을 처리하는 프로세서, AP, 프로그램이 임베디드된 칩과 같은 장치일 수 있다.

[0151] 클라이언트(700)가 풀-랭크 축소 파라미터화를 이용한 모델 개인화를 수행하는 장치일 수도 있다.

[0152] 저장장치(710)는 신경망 모델 개인화 및 개인화된 모델을 이용하여 추론을하는 과정을 수행하는 프로그램 코드 내지 명령어를 저장할 수 있다.

[0153] 메모리(720)는 데이터 개인화 내지 모델 개인화 과정에 필요한 명령어를 저장장치(710)로부터 읽어올 수 있다.

[0154] 연산 장치(730)는 메모리(720)에 저장된 명령어를 실행하면서 데이터 개인화, 신경망 모델 개인화 내지 개인화된 모델을 이용한 추론 과정을 수행한다.

[0155] 파라미터 행렬 W는 전역 가중치 W_1와 지역 가중치 W_2 사이의 하다마드 곱으로 산출된 행렬이다. 연산 장치(730)는 자신의 학습 데이터를 이용하여 파라미터 행렬 W의 가중치를 갱신한다. 학습이 종료되면, 인터페이스 장치(740) 또는 통신 장치(750)가 갱신된 전역 가중치 W_1만을 서버에 전달한다. 인터페이스 장치(740) 또는 통신 장치(750)는 서버로부터 특정 계층에 대한 그레이디언트 통합 결과인 전역 가중치 W'_1을 입력받을 수 있다. 저장 장치(710)는 특정 계층에 대한 전역 가중치 W'_1와 지역 가중치 W_2 를 저장한다. 이후 연산 장치(730)는 특정 계층에 대한 전역 가중치 W'_1와 지역 가중치 W_2에 대한 하다마드 곱을 수행하여 파라미터 행렬 W'를 생성한다. 연산 장치(730)는 자신의 입력 데이터 또는 이전 계층의 출력을 상기 특정 계층에 입력하여 추론을 수행한다.

[0157] 또한, 상술한 바와 같은 풀-랭크 축소 파라미터화 방법, 풀-랭크 축소 파라미터 기반 개인화 방법, 연합 학습 방법, 분산 학습 방법, 신경망 기반 추론 방법 및 데이터 압축 방법은 컴퓨터에서 실행될 수 있는 실행가능한 알고리즘을 포함하는 프로그램(또는 애플리케이션)으로 구현될 수 있다. 상기 프로그램은 일시적 또는 비일시적 판독 가능 매체(non-transitory computer readable medium)에 저장되어 제공될 수 있다.

[0158] 비일시적 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 구체적으로는, 상술한 다양한 어플리케이션 또는 프로그램들은 CD, DVD, 하드 디스크, 블루레이 디스크, USB, 메모리카드, ROM(read-only memory), PROM (programmable read only memory), EPROM(Erasable PROM, EPROM) 또는 EEPROM(Electrically EPROM) 또는 플래시 메모리 등과 같은 비일시적 판독 가능 매체에 저장되어 제공될 수 있다.

[0159] 일시적 판독 가능 매체는 스태틱 램(Static RAM, SRAM), 다이내믹 램(Dynamic RAM, DRAM), 싱크로너스 디램(Synchronous DRAM, SDRAM), 2배속 SDRAM(Double Data Rate SDRAM, DDR SDRAM), 증강형 SDRAM(Enhanced SDRAM, ESDRAM), 동기화 DRAM(Synclink DRAM, SLDRAM) 및 직접 램버스 램(Direct Rambus RAM, DRRAM) 과 같은 다양한 RAM을 의미한다.

[0160] 전술한 다양한 애플리케이션에 대한 프로그램 내지 명령어는 다음과 같다. 이하 설명은 도 2 내지 도 3 등에서 설명한 경량화 과정 및 구조를 전제로 한다.

[0161] 저장 매체 내지 메모리는 데이터 경량화를 수행하도록 제어하는 명령어를 저장할 수 있다. 명령어는 메모리가 벡터 행렬로 표현되는 데이터를 입력받는 단계 및 프로세서가 타깃 파라미터 행렬 W 구성이 가능한 로-랭크 행렬 W_1 및 W_2를 결정하는 단계를 제어하는 코드를 포함할 수 있다. 타깃 파라미터 행렬 W는 상기 W_1 와 상기 W_2 사이의 하다마드 곱으로 구성되고, 상기 W_1은 $m \times r_1$ 크기의 X_1 행렬 및 $n \times r_1$ 크기의 Y_1 행렬의 내적으로 형성되고, 상기 W_2는 $m \times r_2$ 크기의 X_2 행렬 및 $n \times r_2$ 크기의 Y_2 행렬의 내적으로 형성된다. 명령어는 프로세서가 $r_1 \times r_2 \geq \min(m,n)$를 만족하도록 W_1 및 W_2를 결정하는 과정을 제어하는 코드를 포함할 수 있다. 명령어는 프로세서가 $r_1 = r_2 = R$ 및 $R^2 \geq \min(m,n)$을 만족하도록 W_1 및 W_2를 결정하는 과정을 제어하는 코드를 포함할 수 있다. 신경망 계층의 텐서에 대하여 명령어는 프로세서가 X_1 및 X_2의 크기가 각각 $k_1 \times R$이고, Y_1 및 Y_2의 크기가 각각 $k_2 \times R$인 경우, $R \leq \min(k_1,k_2)$을 만족하도록 W_1 및 W_2를 결정하는 과정을 제어하는 코드를 포함할 수 있다.

[0162] 저장 매체 내지 메모리는 경량화된 신경망 모델을 이용한 신경망 학습을 제어하는 명령어를 저장할 수 있다. 이때 신경망 학습은 연합 학습, 분산 학습 등일 수 있다. 전술한 데이터 경량화 과정을 통해 신경망 계층 중 타깃 계층이 경량화된 경우를 가정한다. 타깃 계층은 컨볼루션 계층들 중 적어도 일부(전체 포함) 또는/ 및 전연결 계층들 중 적어도 일부(전체 포함)일 수 있다. 명령어는 메모리가 타깃 계층의 타깃 파라미터 행렬을 구성하는 로-랭크 행렬 W_1 및 W_2를 입력받는 단계, 프로세서가 로-랭크 행렬 W_1 및 W_2에 대한 하다마드 곱을 수행하여 상기 타깃 파라미터 행렬 W를 구성하는 단계 및 프로세서가 학습 데이터로부터 추출되는 특징을 이용하여 타깃 파라미터 행렬 W를 갱신하는 단계를 제어하는 코드를 포함할 수 있다.

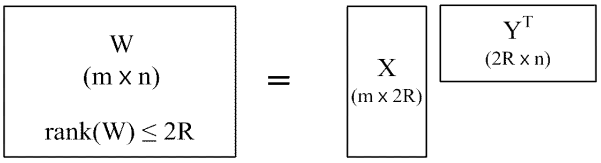
[0163] 저장 매체 내지 메모리는 경량화된 신경망 모델을 이용한 추론을 제어하는 명령어를 저장할 수 있다. 이때 신경망은 다양한 구조 중 어느 하나일 수 있다. 전술한 데이터 경량화 과정을 통해 신경망 계층 중 타깃 계층이 경량화된 경우를 가정한다. 타깃 계층은 컨볼루션 계층들 중 적어도 일부(전체 포함) 또는/ 및 전연결 계층들 중 적어도 일부(전체 포함)일 수 있다. 명령어는 메모리가 타깃 계층의 타깃 파라미터 행렬을 구성하는 로-랭크 행렬 W_1 및 W_2를 입력받는 단계, 프로세서가 로-랭크 행렬 W_1 및 W_1에 대한 하다마드 곱을 수행하여 타깃 파라미터 행렬 W를 구성하는 단계; 및 프로세서가 입력 데이터 또는 타깃 계층의 이전 계층에서 전달되는 특징을 상기 타깃 계층에 입력하여 추론을 진행하는 단계를 제어하는 코드를 포함할 수 있다.

[0164] 저장 매체 내지 메모리는 경량화된 데이터 내지 신경망 모델을 이용한 개인화를 제어하는 명령어를 저장할 수 있다. 이때 신경망은 다양한 구조 중 어느 하나일 수 있다. 경량화된 데이터는 전역적 데이터와 지역적(개인적) 데이터를 포함할 수 있다. 명령어는 메모리가 전술한 데이터 경량화 방법으로 경량화되는 과정에서 결정되는 전역 데이터에 대한 로-랭크 행렬 W_1 및 지역 데이터에 대한 로-랭크 행렬 W_2를 입력받는 단계, 프로세서가 W_1 와 W_2에 대한 하다마드 곱으로 타깃 파라미터 행렬 W를 생성하는 단계 및 상기 프로세서가 상기 W를 이용하여 신경망 모델 학습 또는 신경망 모델을 이용한 추론을 수행하는 단계를 제어하는 코드를 포함할 수 있다. 이때 전역 데이터에 대한 로-랭크 행렬 W_1은 상기 장치가 아닌 다른 적어도 하나의 장치에서 전역 데이터를 이용하여 학습된 결과로 산출된다. 또한, 명령어는 전역 데이터에 해당하는 행렬을 구성하는 로-랭크 내적 행렬들을 서버와 같은 외부 객체에 전달하고, 해당 외부 객체로부터 수신한 갱신된 전역 데이터인 행렬 W_1을 수신하는 과정을 제어하는 코드를 포함할 수 있다.

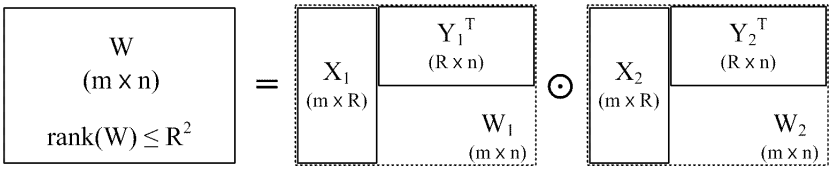
[0165] 본 실시예 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시례는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

도면

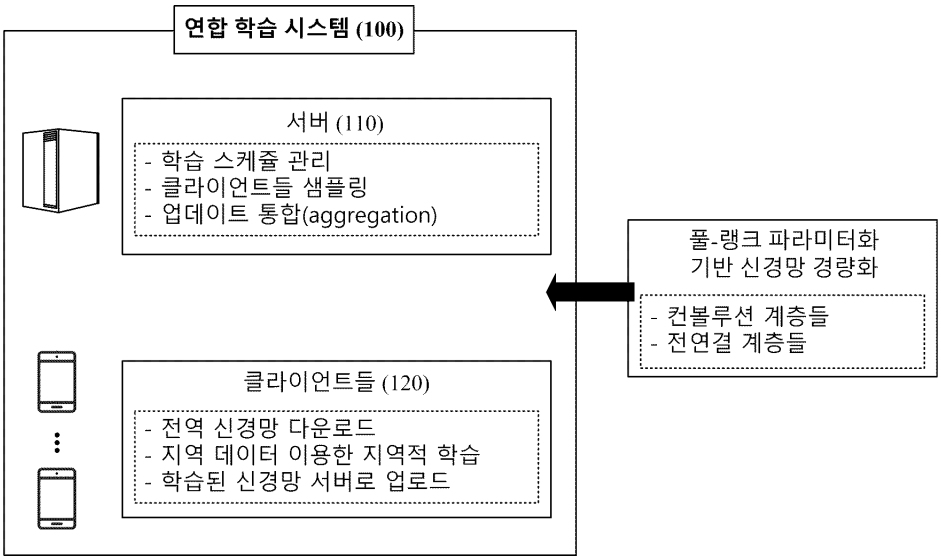
도면1



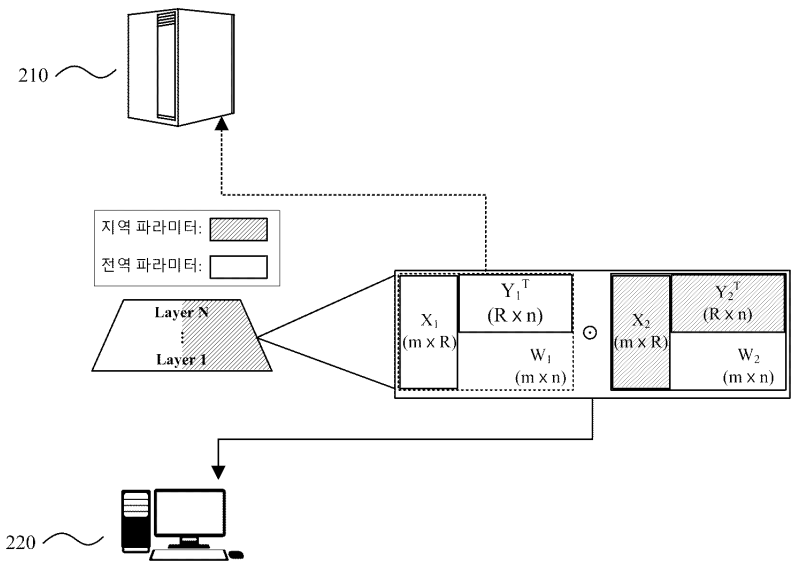
도면2



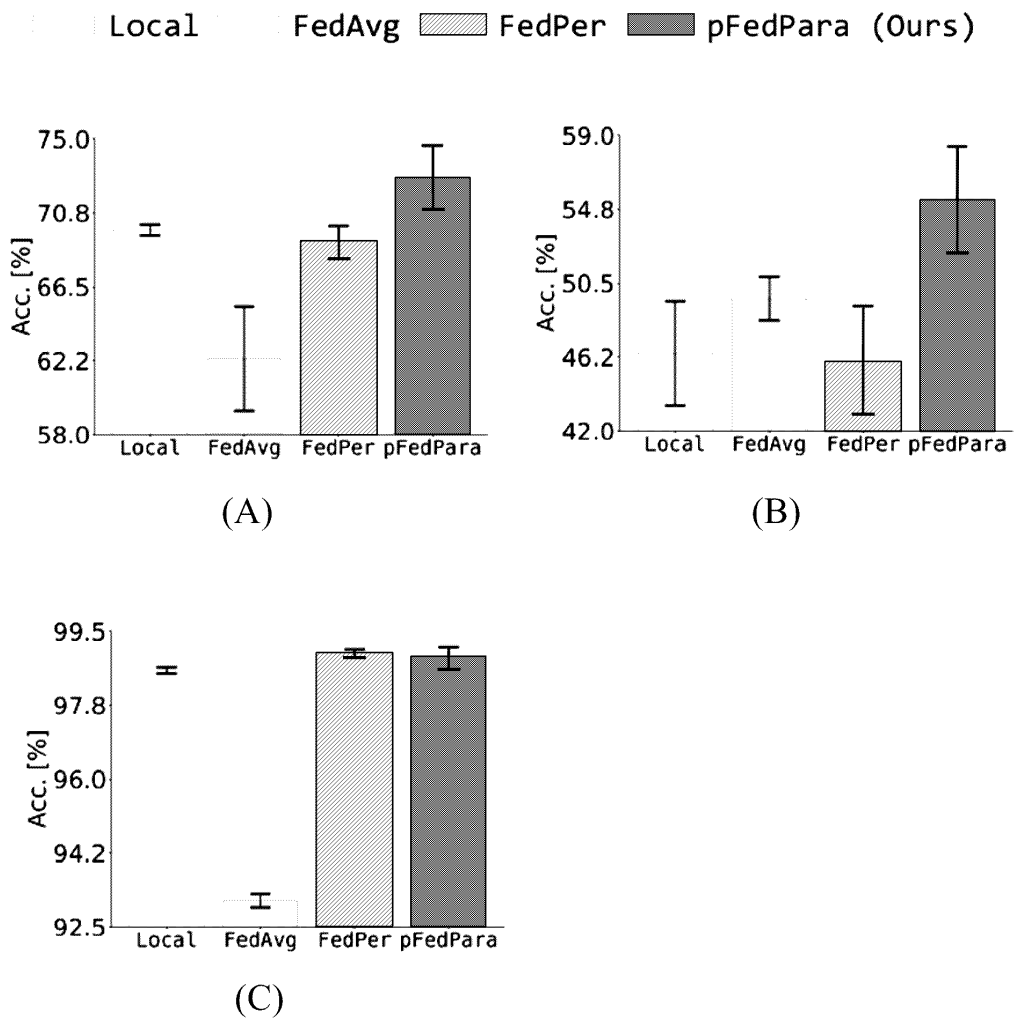
도면3


연합 학습 시스템 (100)
서버 (110)
- 학습 스케쥴 관리
- 클라이언트들 샘플링
- 업데이트 통합(aggregation)
풀-랭크 파라미터화
기반 신경망 경량화
- 컨볼루션 계층들
- 전연결 계층들
클라이언트들 (120)
- 전역 신경망 다운로드
- 지역 데이터 이용한 지역적 학습
- 학습된 신경망 서버로 업로드

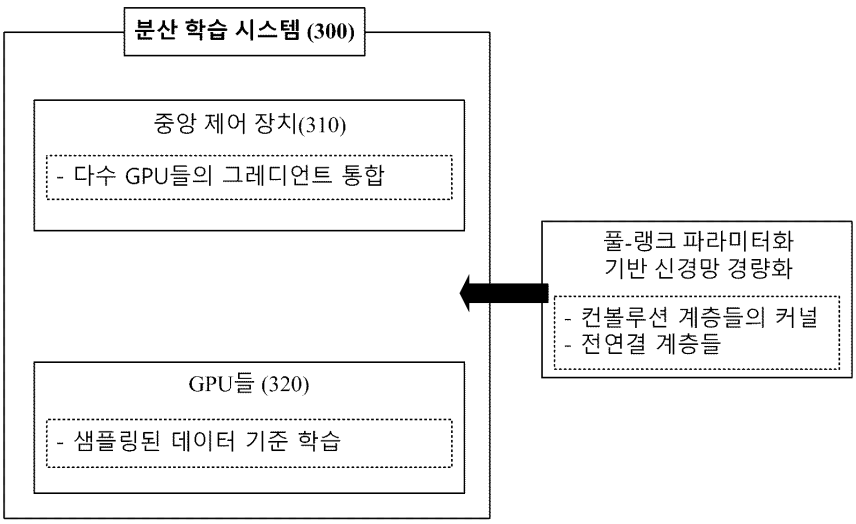
도면4


200
210
지역 파라미터:
전역 파라미터:
Layer N
Layer 1
X1 (m x R)
Y1T (R x n)
W1 (m x n)
X2 (m x R)
Y2T (R x n)
W2 (m x n)
220

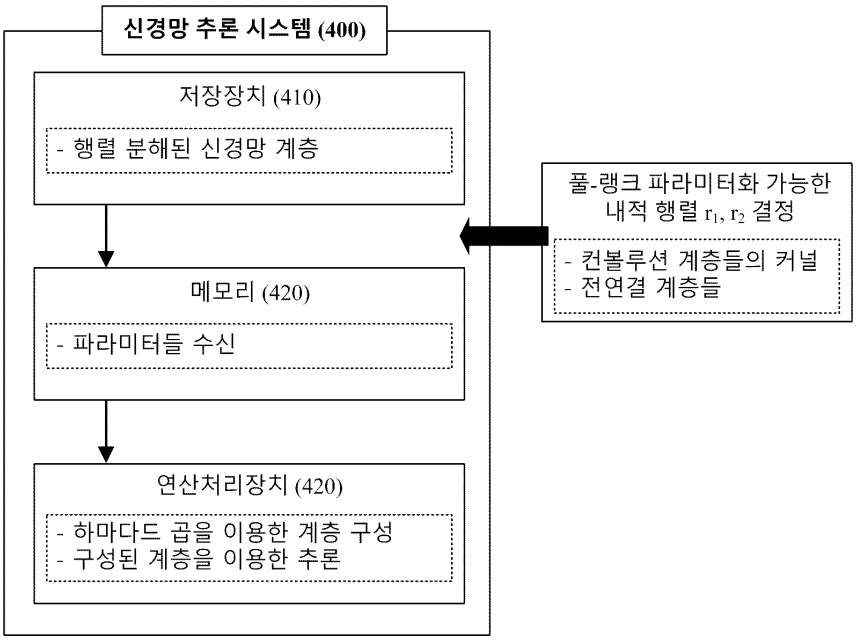
도면5


Local
FedAvg
FedPer
pFedPara (Ours)
Acc. [%]
75.0
70.8
66.5
62.2
58.0
Local
FedAvg
FedPer
pFedPara
(A)
59.0
54.8
50.5
46.2
42.0
(B)
99.5
97.8
96.0
94.2
92.5
(C)

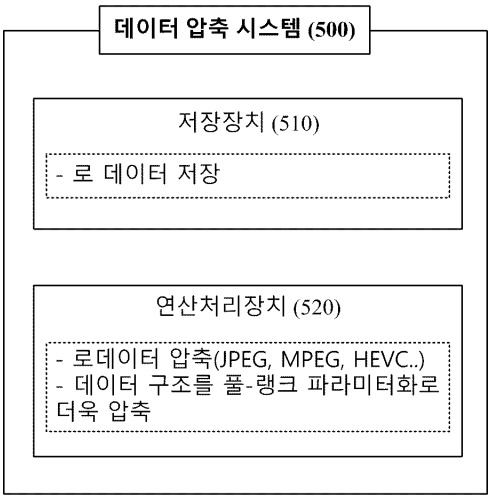
도면6


분산 학습 시스템 (300)
중앙 제어 장치(310)
- 다수 GPU들의 그레디언트 통합
GPU들 (320)
- 샘플링된 데이터 기준 학습
풀-랭크 파라미터화
기반 신경망 경량화
- 컨볼루션 계층들의 커널
- 전연결 계층들

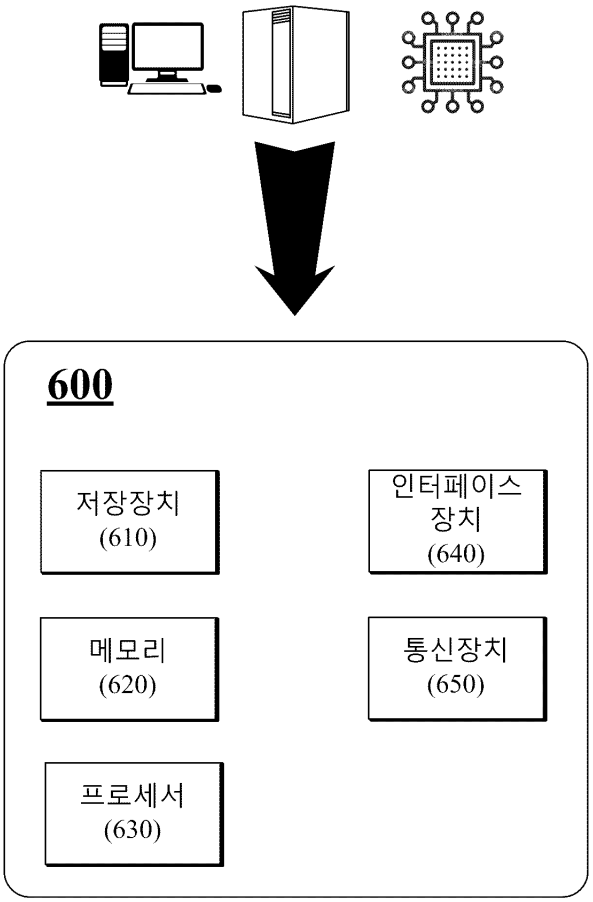
도면7


신경망 추론 시스템 (400)
저장장치 (410)
- 행렬 분해된 신경망 계층
메모리 (420)
- 파라미터들 수신
연산처리장치 (420)
- 하마다드 곱을 이용한 계층 구성
- 구성된 계층을 이용한 추론
풀-랭크 파라미터화 가능한 내적 행렬 r_1, r_2 결정
- 컨볼루션 계층들의 커널
- 전연결 계층들

도면8


데이터 압축 시스템 (500)
저장장치 (510)
- 로 데이터 저장
연산처리장치 (520)
- 로데이터 압축(JPEG, MPEG, HEVC..)
- 데이터 구조를 풀-랭크 파라미터화로 더욱 압축

도면9



도면10