https://doi.org/10.7840/kics.2023.48.3.295

통신 효율적 연합학습을 위한 오토인코더 기반 모델 압축 기법

이 도 윤◆, 이 훈°

Autoencoder-Based Model Compression Schemes for Federated Learning

Do-Yun Lee◆, Hoon Lee°

요 약

최근 지능형 통신 네트워크에 대한 논의가 본격화되면서 분산 인공지능 기술에 관한 연구가 주목받고 있다. 특히, 데이터를 직접 공유하지 않고, 분산된 기기들이 인공지능 모델을 원격으로 훈련하는 연합학습 알고리즘이 크게 각광받고 있다. 연합학습을 수행하기 위해서는 서버-클라이언트 간 모델 교환이 필수적인데, 고차원 인공지능 모델 파라미터를 직접 공유하는 것은 통신 시스템 관점에서 매우 큰 비용을 발생시킨다. 본 논문에서는 통신 효율적 연합학습 시스템을 구축하기 위한 모델 압축 기법을 개발한다. 오토인코더 체계를 활용하여 모델 파라미터의 비선형적인 압축-복원 과정을 실시한다. 제안하는 오토인코더 기법은 연합학습의 가중평균 연산을 추론 단계에 이식하여 서버에서 효과적으로 모델 파라미터를 취합하도록 한다. 이를 통해 기존 모델 압축 기법들에서 달성하지 못했던 수렴속도 및 확장성 개선을 이룩할 수 있다. 모의실험을 통해 제안하는 기법의 효율성을 검증하였다.

키워드 : 연합학습, FedAvg, 모델 압축, 오토인코더, 파라미터 공유
Key Words : Federated Learning, FedAvg, model compression, autoencoder, deep learning, parameter sharing

ABSTRACT

Edge intelligence has been an emerging key enabler of intelligent 6G networks. The federated learning (FL) algorithm has been regarded as a promising solution to realize remote and decentralized training processes of a number of artificial intelligence (AI) models distributed over multiple clients. For the FL system, it is essential to exchange AI model parameters among a server and clients, which incurs prohibitive communication cost. To overcome this challenge, this paper proposes an autoencoder approach to compress AI model parameters in the FL system. Inference steps of the proposed autoencoder are carefully designed such that the weighted averaging operations of the FL algorithm can be injected into the end-to-end compression-reconstruction process. Numerical results demonstrate the effectiveness of the proposed method over conventional schemes.

※ 이 논문은 부경대학교 자율창의학술연구비(2021년)에 의하여 연구되었음
◆ First Author : Pukyong National University, Department of Information and Communications Engineering, ehdbs8964@pukyong.ac.kr, 학생회원
° Corresponding Author : Pukyong National University, Department of Information and Communications Engineering, hlee@pknu.ac.kr, 정회원
논문번호 : 202212-297-A-RU, Received December 11, 2022; Revised January 20, 2023; Accepted January 30, 2023

Ⅰ. 서 론

최근 분산된 클라이언트의 인공지능 모델을 원격으로 최적화하는 연합학습 기법이 주목받고 있다[1-4]. 훈련 데이터를 직접 교환하지 않고, 서버가 개별 클라이언트의 지역 (local) 모델 파라미터를 취합한다. 데이터를 공유하지 않으므로 개인 정보 유출을 방지하고, 통신 효율적인 분산 훈련이 가능하다. 하지만 기계학습 문제의 난이도가 높아짐에 따라 요구되는 인공지능 모델이 매우 거대해졌으며, 연합학습에 필요한 통신 자원도 크게 증가하였다.

이러한 어려움을 타개하기 위해 통신 효율적 연합학습 기법들이 최근 연구되고 있다. 인공지능 모델 파라미터를 저차원 벡터로 압축하는 모델 압축 기술이 소개된 바 있다[5]. 행렬 분해 기술에 기반하여 각 클라이언트의 지역 모델 파라미터를 압축 행렬과 정보 행렬의 곱으로 분해한다. 정보 행렬은 무작위로 생성하여 서버와 클라이언트가 미리 생성하여 교환하지 않도록 설계하고, 저차원의 정보 행렬만을 훈련 변수로 취급하여 서버와 교환하여 통신 효율을 극대화한다. 또 다른 해결 방안으로 압축 센싱 (compressed sensing) 이론을 활용한 기법이 제안되었다[6,7]. 훈련에 필요한 gradient 벡터를 희소화하여 통신 비용을 절감하는 전략이 주로 채택되었다. 이를 위해 클라이언트는 지역 gradient 벡터를 희소 벡터로 근사한 후, 압축 행렬을 곱해 차원을 축소하여 전송한다. 서버에서는 복원 알고리즘을 통해 희소 벡터를 재구성한 후, 모든 클라이언트에서 수신한 정보를 취합하여 훈련을 수행한다.

기존 문헌에서는 대부분 선형 압축 과정을 고려하였다. 하지만 선형 연산만으로는 고차원 모델 파라미터의 비선형적인 은닉 특성을 인지하는 것은 불가능하다[8]. 압축 센싱 기반 기법들의 경우, 압축 행렬을 무작위하게 생성하므로 은닉 특성을 파악하기 어렵다. 따라서 시스템의 성능은 압축을 수행하지 않은 기존 연합학습 알고리즘보다 열화된다. 이를 위해 연합학습의 손실함수를 최소화하는 모델 희소화 방법이 제안되었다[9],[10]. 모델을 압축하면서도 은닉 특성을 효과적으로 추출하여, 특정 상황에서는 압축을 수행하지 않은 기존 연합학습 알고리즘의 성능을 상회한다[10]. 그러나, 파라미터의 희소화 위치 정보를 교환하기 위한 추가 비용이 발생한다.

모델 파라미터의 비선형 특성을 반영한 압축 기법이 최근 발표되었다[11-13]. 단순한 선형 압축 기법을 사용하지 않고, 비선형 연산 능력을 지닌 오토인코더 신경망을 활용한다. 이를 통해 고차원 모델 파라미터의 은닉 특성을 효과적으로 추출하여 높은 복원 정확도를 달성한다. 기존 오토인코더 기반 연합학습 시스템에서는 오토인코더를 학습하기 위해 압축 오차를 최소화하는 손실함수를 차용하였다. 하지만 이는 기존의 압축 센싱 방법들과 유사하게 단순 압축-복원만을 수행하므로, 연합학습 시스템에서 궁극적으로 목표로 하는 기계학습 문제의 성능을 개선할 수는 없다. 또한, 각 클라이언트-서버 쌍에서 서로 다른 오토인코더를 사용하므로 특정 데이터 집합 특성에 과적합되어 일반화 능력이 저하된다.

본 논문에서는 통신 효율적인 연합학습 시스템을 위한 오토인코더 체계를 개발한다. 오토인코더의 일반화 능력 증대를 위해 모든 클라이언트-서버 쌍이 동일한 오토인코더를 사용하는 파라미터 공유 (parameter sharing) 전략을 채택한다. 연합학습 알고리즘의 가중평균 연산을 오토인코더의 추론 과정에 포함하여, 훈련을 통해 오토인코더가 연합학습의 주요 메커니즘을 인지하도록 한다. 모의실험 결과를 통해 제안하는 기법의 효율성을 확인한다.

Ⅱ. 시스템 모델

본 논문에서는 K개의 클라이언트와 서버가 협력적으로 분산 인공지능 훈련을 수행하는 연합학습 시스템을 고려한다(그림 1 참고). 클라이언트 k의 개별 훈련 데이터 집합 D_k는 아래와 같이 정의된다.

$$D_k = \left\{ (x_k^{(i)}, y_k^{(i)}) : i = 1, \cdots, n_k \right\} \tag{1}$$

여기서 n_k는 클라이언트 k의 훈련 샘플 수, $(x_k^{(i)}, y_k^{(i)})$는 i번째 데이터 샘플, $x_k^{(i)}$는 모델의 입력, $y_k^{(i)}$는 레

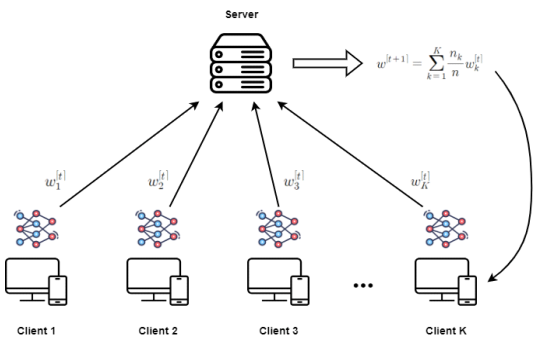

그림 1. 연합학습 시스템
Fig. 1. Federated learning system

이블을 의미한다. 서버는 클라이언트와 협력하여 N차원 전역 모델 파라미터 벡터 $w \in \mathbb{R}^N$를 최적화한다. 서버가 분산된 데이터 집합을 획득하여 중앙집중적 학습을 실시할 수 있으나, 통신 비용이 증가하고, 개인정보 유출 문제가 발생한다.

상기 문제점을 해결하기 위해 연합학습 시스템은 데이터 집합을 교환하지 않고 모델 파라미터 w를 서버와 클라이언트가 공유하여 훈련을 진행한다. 클라이언트 k의 손실함수 $L_k(w)$는 다음과 같다.

$$L_k(w) = \frac{1}{n_k} \sum_{(x,y)\in D_k} l(x,y;w) \tag{2}$$

이때 $l(x,y;w)$는 특정 데이터 샘플 (x,y)를 통해 계산된 손실함수를 의미한다. 예를 들어 regression 문제의 경우 아래의 mean-squared error (MSE)를 사용할 수 있다.

$$l(x,y;w) = \| f_w(x) - y \|^2 \tag{3}$$

이때 f_w는 파라미터 w를 지닌 연합학습 인공지능의 순전파 (forward-pass)를 의미한다. 연합학습 시스템에서 해결하고자 하는 문제는 아래와 같다.

$$\min_w \sum_{k=1}^{K} \frac{n_k}{n} L_k(w) \tag{4}$$

여기서 $n = \Sigma_{k=1}^{K} n_k$는 총 데이터 샘플의 수이다.

다양한 연합학습 알고리즘 중, 본 논문에서는 통신 네트워크 분야에서 가장 보편적으로 활용되는 FedAvg 기법[1]을 고려한다. FedAvg는 중앙서버와 클라이언트가 인공지능 모델 파라미터 w를 수차례 교환하며 반복적인 훈련을 수행한다. 서버-클라이언트 사이의 파라미터 교환 과정을 1회의 통신 round로 정의한다. 총 T회의 round 중 t번째 round의 모델 파라미터를 $w^{[t]}$로 표현한다. 매 round에서 각 클라이언트 k는 stochastic gradient descent (SGD) 기반의 지역 훈련을 총 E회의 epoch 만큼 수행한다. t번째 round의 e번째 ($e = 1,\cdots,E$) 지역 훈련 후 클라이언트 k의 모델 파라미터 $w_k^{[t,e]}$는 다음과 같이 얻어진다.

$$w_k^{[t,e]} = w_k^{[t,e-1]} - \eta \nabla L_k(w_k^{[t,e-1]}) \tag{5}$$

이때 η는 학습률 (learning rate), ∇는 gradient를 의미한다. 편의상 $w_k^{[t,0]} = w^{[t]}$로 가정한다. 빠른 연산을 위해 $\nabla L_k(w)$는 mini-batch 집합 $B_k \subset D_k$을 통해 근사할 수 있다.

$$\nabla L_k(w) \simeq \frac{1}{|B_k|} \sum_{(x,y)\in B_k} \nabla l(x,y;w) \tag{6}$$

지역 훈련을 수행한 후 클라이언트 k는 상향링크 통신 채널을 통해 지역 모델 파라미터 $w_k^{[t]} = w_k^{[t,E]}$를 서버로 송신한다. 모든 클라이언트로부터 지역 모델 파라미터를 수신한 후, 서버는 아래의 가중평균 연산으로 전역 모델 파라미터 $w^{[t+1]}$를 취합한다.

$$w^{[t+1]} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{[t]} \tag{7}$$

갱신된 전역 모델 파라미터 $w^{[t+1]}$는 하향링크 통신 채널을 통해 클라이언트들에게 배포되고, 각 클라이언트는 이를 초기값으로 사용하여 지역 훈련 (5)를 실시한다. 상기 절차가 총 T회의 round 동안 반복되고, 최종 전역 모델을 인공지능 추론에 활용한다.

FedAvg 알고리즘을 실시간으로 구동하기 위해서는 클라이언트-서버 간 반복적인 모델 파라미터 교환 절차가 필수적이다. 고난도의 인공지능 문제를 해결하려면 전역 모델 파라미터의 차원 N이 증가하므로, 각 round의 통신 비용이 폭발적으로 커진다. 또한, 효과적인 전역 모델을 얻기 위해 수백 회의 반복 훈련이 필요하므로 매우 긴 지연시간을 야기한다. 상기 문제들을 해결하기 위해 인공지능을 활용한 지역 모델 파라미터 압축 기법을 개발한다.

Ⅲ. 제안하는 오토인코더 기반 연합학습 시스템

본 절에서는 오토인코더를 이용한 통신 효율적 연합학습 기술을 제안한다. 오토인코더는 인코더 신경망 $g_\theta : \mathbb{R}^N \to \mathbb{R}^H$ 및 디코더 신경망 $h_\phi : \mathbb{R}^H \to \mathbb{R}^N$ 으로 구성된다. 여기서 θ와 ϕ는 각각 인코더 신경망과 디코더 신경망의 모델 파라미터이다. 입력 데이터 $w \in \mathbb{R}^N$의 압축을 수행하기 위해, 인코더 신경망 g_θ의 출력 차원 H는 입력 차원 N보다 작게 설정한다. 인코더 출력 $g_\theta(w)$는 입력 데이터를 압축한 신호로 사용한다. 디코더 신경망 h_ϕ은 압축 신호 $g_\theta(w)$를 통

해 입력 데이터 w를 복원한다. 즉, 디코더 신경망의 출력 $h_\phi(g_\theta(w))$은 w의 복원 예측 결과값으로 활용한다. 복원 오차를 최소화하기 위해 훈련 손실함수 $L(\theta,\phi)$로 다음과 같은 MSE를 사용할 수 있다.

$$L(\theta,\phi)=\mathbb{E}\left[\parallel w-h_\phi(g_\theta(w))\parallel^2\right] \tag{8}$$

3.1 기존 연구 결과

오토인코더를 통해 연합학습의 통신 비용을 줄이는 방법이 연구되었다[11],[12]. 인코더 신경망을 클라이언트에, 디코더 신경망을 서버에 설치하여 압축된 지역 파라미터를 서버로 전송한다. 클라이언트 k-서버 간 인코더 및 디코더 신경망을 각각 g_{θ_k}, h_{ϕ_k}로 표현하면, 클라이언트 k는 지역 모델 파라미터 $w_k^{[t]}\in\mathbb{R}^N$의 압축 신호 $z_k^{[t]}\in\mathbb{R}^H$를 생성한다.

$$z_k^{[t]}=g_{\theta_k}(w_k^{[t]}) \tag{9}$$

$z_k^{[t]}$를 수신한 후, 서버는 디코더 신경망 h_{ϕ_k}으로 클라이언트 k의 지역 파라미터 추정 $\hat{w}_k^{[t]}$을 계산한다.

$$\hat{w}_k^{[t]}=h_{\phi_k}(z_k^{[t]})=h_{\phi_k}(g_{\theta_k}(w_k^{[t]})) \tag{10}$$

복원 성능 향상을 위해 k번째 오토인코더의 훈련 손실함수 $L_{\mathrm{AE}}(\theta_k,\phi_k)$를 MSE 함수로 설정한다[6,7].

$$L_{\mathrm{AE}}(\theta_k,\phi_k)=\frac{1}{T}\sum_{t=1}^{T}\|w_k^{[t]}-h_{\phi_k}(g_{\theta_k}(w_k^{[t]}))\|^2 \tag{11}$$

기존 오토인코더 기법은 단점은 낮은 일반화 능력이다. 각 클라이언트가 서로 다른 오토인코더를 사용하므로, 훈련 상황에서 가정한 클라이언트 수와 실제 테스트 환경에서의 클라이언트 수가 다르면 직접 적용이 불가능하다. 이를 해결하기 위해 모든 가능한 클라이언트 수에 대해 미리 복수의 오토인코더를 훈련해 두는 것인데, 이는 훈련 복잡도 및 메모리 사용량 관점에서 매우 비합리적인 방법이다.

또한, k번째 오토인코더 g_{θ_k}, h_{ϕ_k}는 클라이언트 k의 데이터 집합 D_k만으로 최적화된다. 그러므로, 클라이언트에 분산된 데이터 집합의 사전 확률분포가 non-independent and identically distributed (iid) 특성을 지닌다면, 클라이언트 k의 오토인코더는 해당 데이터 분포에만 잘 동작하는 과적합 문제를 야기한다. 따라서, 실제 구현 상황에서 클라이언트가 수집한 데이터의 통계적 특성이 훈련 상황과 다르면 큰 성능 열화가 발생한다[10].

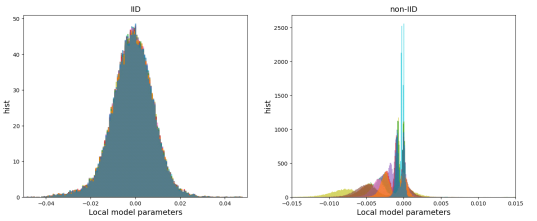

그림 2. 지역 모델 파라미터의 히스토그램
Fig. 2. Histogram of local model parameters

그림 2는 iid 및 non-iid 데이터 분포 상황에 FedAvg 알고리즘으로 최적화한 지역 모델 파라미터 분포도를 나타낸다. 이상적인 iid 상황에서는 모든 클라이언트의 데이터 집합이 동일한 통계적 특성을 보유하므로 최적화된 지역 모델 파라미터의 분포도가 모두 같다. 반면에, 현실적인 non-iid 데이터 분포 상황의 경우 각 클라이언트의 데이터 집합이 지닌 통계적 특성이 상이하다. 이에 따라 훈련된 지역 모델 파라미터의 분포 역시 크게 다른 것을 확인할 수 있다. 따라서, 특정 클라이언트만을 위해 훈련된 오토인코더로는 상이한 네트워크 환경 및 데이터 분포 상황에 바로 적용하는 것은 불가능하다.

3.2 제안하는 FedAvgAE 기법

오토인코더의 확장성을 개선하기 위해 파라미터 공유 전략을 채택한다. 이 기법은 합성곱 신경망에서 사용되었으며, 동일한 파라미터를 입력 데이터에 반복적으로 재사용하여 신경망의 일반화 능력을 개선할 수 있다[14]. 최근에는 그래프 신경망에서 접점의 연산을 동일한 신경망으로 모사하여 그래프 연결상태 및 크기에 대한 일반화 능력을 확보하였다[15].

그림 3과 같이 제안하는 기법은 모든 클라이언트-서버 쌍이 동일한 오토인코더를 사용한다.

$$g_\theta=g_{\theta_1}=\cdots=g_{\theta_K},\ h_\phi=h_{\phi_1}=\cdots=h_{\phi_K} \tag{12}$$

즉, 단일 인코더 신경망 g_θ을 모든 클라이언트에 설치하고, 단일 디코더 신경망 h_ϕ을 K개의 지역 모델파라미터 복원에 재사용한다. 훈련 과정에서 단일 인코더 신경망은 훈련 과정에서 모든 클라이언트들의 지역 모델 파라미터 $\left\{w_k^{[t]}:\forall k\right\}$를 관찰할 수 있다. 이러한 전략은 non-iid 데이터 집합의 상이한 통계적 특성을

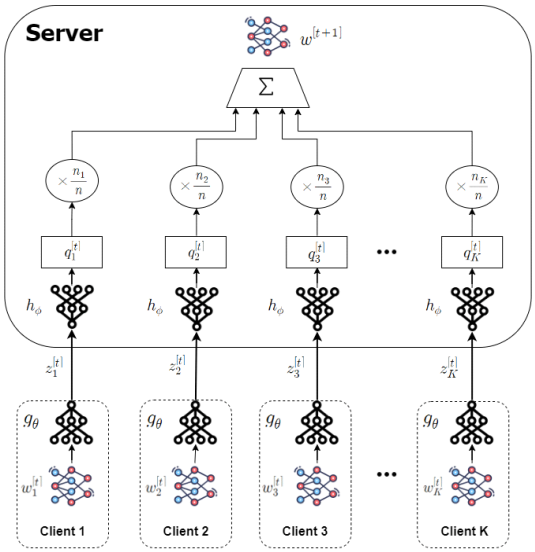

그림 3. 제안하는 FedAvgAE 기반 연합학습 시스템
Fig. 3. Proposed FedAvgAE-based federated learning system

인코더 신경망이 미리 경험하도록 하여 일반화 능력을 향상시킨다. 마찬가지로, 단일 디코더 신경망도 서로 다른 압축 코드 분포를 학습하여, 전혀 다른 테스트 환경에서도 적응이 가능하다.

기존 기법의 손실함수 (7)를 수정하여 파라미터 공유 전략을 적용한 오토인코더의 훈련 손실함수 $\tilde{L}_{\mathrm{AE}}(\theta,\phi)$를 다음과 같이 설정할 수 있다.

$$\tilde{L}_{\mathrm{AE}}(\theta,\phi)=\frac{1}{TK}\sum_{t=1}^{T}\sum_{k=1}^{K}\|w_k^{[t]}-h_\phi(g_\theta(w_k^{[t]}))\|^2 \quad (13)$$

기존 손실함수 (11)과 비교하면, 오토인코더를 재활용하기 위해 클라이언트에 대한 평균을 취한다.

하지만 위 손실함수는 여전히 한계를 가지고 있다. 디코더 신경망은 지역 모델 파라미터 $w_k^{[t]}$를 오차 없이 복원하는 것에만 집중한다. 그러나, FedAvg 알고리즘의 가중평균 연산 (7)을 고려하면 최적의 전역 모델 파라미터 $w^{[t]}$를 추출하기 위해 필요한 정보는 개별 지역 모델 파라미터 $\{w_k^{[t]}:\forall k\}$가 아닌, 그들의 가중평균임을 알 수 있다. 즉, 복원 오차를 줄이는 MSE 손실함수 (13)은 연합학습 시스템의 궁극적인 목표와는 차이가 있다.

이를 해결하기 위해 새로운 오토인코더 훈련 방법을 고안한다. 제안하는 기법의 핵심은 디코더 신경망이 가중평균 연산 (7)을 직접 수행하도록 디코더의 추론 과정을 개선하는 것이다. 먼저, 클라이언트는 인코딩 신경망을 통해 압축 신호 $z_k^{[t]}$를 생성한다.

$$z_k^{[t]}=g_\theta(w_k^{[t]}),\ \forall k \quad (14)$$

기존 기법과는 다르게, 서버는 디코더 신경망을 특정 클라이언트의 모델 파라미터 예측에 사용하지 않고, (7)을 위한 특징 추출기로 활용한다. 전역 파라미터의 예측 $\hat{w}^{[t+1]}$은 다음과 같이 계산된다.

$$\hat{w}^{[t+1]}=\sum_{k=1}^{K}\frac{n_k}{n}h_\phi(g_\theta(w_k^{[t]})) \quad (15)$$

즉, 디코딩 신경망은 FedAvg 알고리즘의 가중평균 연산 (7)을 직접 수행한다. 이러한 특징을 기반으로 제안하는 기법을 FedAvgAE 방법으로 명명한다.

최종 출력 (15)를 전역 모델 파라미터로 활용하기 위해 제안하는 FedAvgAE의 훈련 손실함수 $L_{\mathrm{FedAvgAE}}(\theta,\phi)$를 다음과 같이 설계한다.

$$\begin{aligned}L_{\mathrm{FedAvgAE}}(\theta,\phi)&=\frac{1}{T}\sum_{t=1}^{T}\|\,w^{[t]}-\hat{w}^{[t]}\,\|^2\\&=\frac{1}{T}\sum_{t=1}^{T}\left\|\sum_{k=1}^{K}\frac{n_k}{n}w_k^{[t]}-\sum_{k=1}^{K}\frac{n_k}{n}h_\phi(g_\theta(w_k^{[t]}))\right\|^2\end{aligned} \quad (16)$$

단순히 파라미터의 압축-복원 방법을 학습하는 기존 오토인코더 기법[11], [12]과는 다르게, 제안하는 FedAvgAE는 FedAvg 알고리즘의 내재적 의미를 오토인코더가 스스로 파악하도록 설계된다. 디코더가 각 클라이언트의 지역 모델 파라미터의 예측값이 아닌, 전역 모델 파라미터를 직접 예측하기 위한 은닉 정보로 활용된다. 따라서, FedAvgAE는 파라미터의 단순 압축에 과적합 되지 않아 일반화 능력을 개선할 수 있다. 디코더의 가중평균 연산 (15)는 그래프 신경망에서 서로 다른 정점의 정보를 취합할 때 사용하는 sum-pooling 연산의 일반화된 구조로 해석할 수 있다[15]. 서로 다른 인공지능의 출력을 합산하여 그래프 전반을 표현하는 은닉 정보로 사용한다. 연합학습 시스템을 서버-클라이언트로 구성되는 star 그래프 구조로 해석할 수 있다. 이때, FedAvgAE의 가중평균 동작은 graph attention[15]의 특수한 경우에 해당된다. 서버는 압축 신호를 디코더 신경망으로 전처리 후, 중요도 n_k/n의 비율만큼 만큼 취합하여 최종 전역 모델 파라미터를 생성한다. 그래프 기반의 FedAvgAE의 추론

과정 및 훈련 손실함수는 star 그래프의 노드 수, 즉, 클라이언트의 수 K, 및 non-iid 데이터 분포에 대한 일반화 능력을 증가시킨다.

제안하는 FedAvgAE 기법은 신경망으로 다른 신경망의 파라미터를 생성하는 하이퍼 네트워크 (hyper network) 체계의 일종으로 해석할 수 있다. 최근 하이퍼 네트워크 개념이 연합학습 시스템에 적용된 바 있다[16]. 거대 신경망을 활용하여 전역 모델 파라미터를 생성하면 인공지능의 성능이 기존 FedAvg 방법 대비 크게 개선된다는 사실이 입증되었다. 하지만, 기존 기법에서는 모델 파라미터의 압축을 고려하지 않아 통신 시스템 관점에서는 매우 비효율적이다. 이러한 결과에 기반하여, 제안하는 FedAvgAE 기법이 기존 FedAvg 기법의 성능을 개선할 수 있으며, 파라미터 압축을 통해 통신 효율성 역시 향상될 수 있을 것으로 예측된다.

3.3 FedAvgAE 기법의 훈련 및 구현 방법

Algorithm 1은 FedAvgAE의 훈련 과정을 나타낸다. 훈련 데이터 수집을 위해 FedAvg 알고리즘을 수행하여 각 통신 round t에 대한 지역 모델 파라미터 $w_k^{[t]}$ 및 가중치 $m_k = n_k/n$의 tuple 집합 $F^{[t]} = \{(w_k^{[t]}, m_k) : \forall k\}$을 획득한다. FedAvgAE의 추론 과정 (14), (15)은 round 순서 t에 무관하다. 따라서, 훈련을 위한 mini-batch 집합은 무작위로 선택한 round 집합 $\tau \subset [1, T]$의 tuple로 구성한다. 손실함수 (16)은 다음과 같이 mini-batch 집합을 통해 근사한다.

$$\tilde{L}_{\text{FedAvgAE}}(\theta,\phi) = \frac{1}{|\tau|}\sum_{t\in\tau} \left\| \sum_{(m,w)\in F^{[t]}} mw - \sum_{(m,w)\in F^{[t]}} m h_\phi(g_\theta(w)) \right\|^2 \quad (17)$$

순전파 과정을 통해 손실함수 (17)을 계산한 후, 역전파 알고리즘을 활용하여 FedAvgAE의 파라미터 θ와 ϕ를 mini-batch SGD 알고리즘으로 최적화한다. 상기 훈련 과정을 반복적으로 실시한다.

Algorithm 1. FedAvgAE 훈련 알고리즘

Initialize θ and ϕ.
repeat
  Randomly generate $\tau \subset [1, T]$.
  Calculate the loss function $\tilde{L}_{\text{FedAvgAE}}(\theta,\phi)$.
  Update the FedAvgAE parameters:
    $(\theta,\phi) \leftarrow (\theta,\phi) - \alpha \nabla \tilde{L}_{\text{FedAvgAE}}(\theta,\phi)$
until convergence

Algorithm 2. FedAvgAE 기반 연합학습

Initialize the global model parameter $w^{[1]}$
for each round $t = 1, \cdots, T$ **do**
  Server broadcasts $w^{[t]}$ to clients.
  Each client k updates its local model parameter $w_k^{[t]}$ from (5).
  Each client k sends the encoded signal $z_k^{[t]}$ in (14) to the server.
  Server recovers the global model parameter $w^{[t+1]}$ from (15).

Algorithm 2는 훈련된 FedAvgAE를 활용한 실시간 연합학습 과정을 나타낸다. 매 round의 시작마다 서버는 클라이언트들에게 현재 전역 모델 파라미터 $w^{[t]}$를 전송한다. 클라이언트 k는 수신한 전역 모델 파라미터를 초기값으로 지역 모델 파라미터의 훈련을 수행한다. 총 E회 epoch의 지역 훈련을 마친 후, 각 클라이언트는 훈련된 인코더 신경망 g_θ를 사용하여 지역 모델 파라미터 $w_k^{[t]}$를 압축 신호 $z_k^{[t]}$로 재구성한다. 이를 상향링크 통신으로 서버에게 전송하면, 서버는 디코더 신경망 h_ϕ을 기반으로 전역 모델 파라미터 $w^{[t+1]}$를 구성한다. 상기 과정을 T회의 round 동안 반복한다. FedAvgAE의 훈련은 사전에 수행되므로 Algorithm 1은 실시간 구현에 영향을 미치지 않는다. 연합학습 과정에서 서버와 클라이언트는 H차원의 압축 신호 $z_k^{[t]} \in \mathbb{R}^H$만을 교환한다. 따라서, 설계 변수 H를 모델 파라미터 $w \in \mathbb{R}^N$의 차원 N보다 작게 설정하면 기존 연합학습 기법 대비 통신 비용을 감소시킬 수 있다.

Ⅳ. 모의실험 결과

제안하는 기법의 성능을 분석하기 위해 모의실험 결과를 제공한다. K개의 클라이언트에 분산되어있는 FashionMNIST 분류 문제를 해결하는 연합학습 시스템을 고려한다. FedAvg 알고리즘의 모의실험 환경을 표 1에 정리하였다. FedAvg 알고리즘은 제안하는 FedAvgAE의 훈련 데이터 집합 $\{F^{[t]} : \forall t\}$를 획득하고, 훈련된 FedAvgAE의 성능을 검증하기 위해 활용된다. Non-iid 환경을 구축하기 위해 각 클라이언트의 집합에 소속된 각 클래스 샘플의 수가 Dirichlet 분포

표 1. FedAvg 모의실험 설정
Table 1. Simulation setup for FedAvg algorithm

Parameter	Settings
Number of Clients K	20
Rounds of FL T	200
Hidden Dimension H	420
Batch Size B	256
Local epoch E	4
Learning Rate η	iid: 3×10^{-5} non-iid: 7×10^{-5}

를 따르도록 설정한다. FedAvg 알고리즘으로 훈련하는 인공지능 모델은 2개의 합성곱 계층 및 1개의 완전연결 계층으로 구성한다. 합성곱 계층은 각각 32개, 64개의 3×3커널을 갖는다. 각 계층마다 batch normalization, ReLU, 그리고 max-pooling 연산을 진행한다. 연합학습 모델 파라미터 w의 차원은 $N=42058$로 결정된다.

모델 압축을 위한 FedAvgAE의 인코더 및 디코더 신경망은 각각 4개의 완전연결 계층으로 이루어져 있다. 인코더 신경망의 각 계층의 출력 차원은 42058, 4096, 2048, 1024, 그리고 압축 신호의 차원 H로 설정하였으며, 디코더 신경망은 반대의 구조를 갖는다. 배치 크기는 8로, 학습률은 iid 및 non-iid 환경에 각각 3×10^{-6} 및 10^{-5}로 설정하여 400회 반복하였다. FedAvgAE의 성능을 검증하는 테스트 단계에서는, 훈련 데이터를 수집하는 과정과는 다른 난수 발생 seed를 사용하여, 상이한 클라이언트 데이터 집합을 활용하도록 실험을 설계하였다.

제안하는 FedAvgAE 기법의 성능을 평가하기 위해 다음의 두 가지 비교 방법을 고려한다.

(1) FedAvg[1]: 압축을 수행하지 않고 모델 파라미터를 직접 교환하는 연합학습 시스템
(2) Conventional autoencoder (AE)[11]: 손실함수 (13)으로 오토인코더를 학습하여 연합학습의 지역 파라미터를 압축하는 기존 기법

통신 round에 따른 연합학습 모델의 정확도 성능 수렴성을 그림 4에 나타내었다. 훈련 및 테스트 과정에서 클라이언트의 수를 모두 $K=20$으로 고정하였다. 압축 신호의 차원은 $H=420$으로 설정하여 매 통신 round에서 약 97% 압축률을 달성한다. 이상적인 iid 및 현실적인 non-iid 환경에 대한 성능을 함께 나타내었다. 모든 상황에서 제안하는 기법이 다른 두 비교 기법 대비 더 빠른 수렴성을 보이며, 이러한 특성

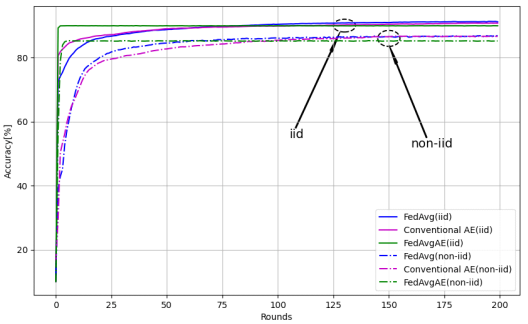

그림 4. 연합학습 정확도 성능 수렴성 ($K=20, H=420$)
Fig. 4. Convergence behavior of test accuracy performance ($K=20, H=420$)

은 non-iid 환경에서 두드러진다. 그에 비해 conventional AE 기법의 성능은 압축을 수행하지 않은 FedAvg 기법과 유사한 경향을 보인다. 이는 오토인코더가 단순 압축-복원 목표로 설계되었기 때문이다. 반면에, 제안하는 FedAvgAE는 FedAvg 알고리즘의 가중평균 연산 (7)을 추론 과정에서 직접 수행하여 연합학습 시스템이 궁극적으로 요구하는 맥락 정보 (context information)를 직접 생성한다. 따라서, 인코딩 신경망의 출력 $z_k^{[t]}$는 단순한 차원 축소 결과물이 아닌, 서버에서 전역 모델 파라미터를 통합할 때 요구하는 충분 통계량 (sufficient statistics)으로 동작하여 정확도 성능을 개선할 수 있다[16]. FedAvgAE의 훈련 과정에서 FedAvg 알고리즘의 모든 trajectory $\left\{w_k^{[t]}:\forall t\right\}$를 인코더 및 디코더 신경망이 사전에 관찰하여 최적 전역 모델 파라미터의 확률 분포도를 미리 학습할 수 있다. 이러한 특성들 덕분에 FedAvgAE가 연합학습의 수렴 지연시간을 단축할 수 있다. Non-iid 환경에서 85% 정확도 성능을 달성하기 위해 제안하는 기법은 약 5회 이내의 round 만에 정확도 성능이 빠르게 수렴하는 반면, 기존 기법들은 동일한 성능을 달성하는데 약 90회의 통신 round가 요구된다. FedAvgAE 기법은 기존 FedAvg 방법 대비 1% 이내의 정확도 성능 손실만으로도 매 통신 round에 요구되는 자원을 약 97% 절약한다. 이와 더불어 전체 분산 훈련 과정의 지연시간을 18배 감축한다. 결론적으로, 제안하는 방법은 모델 파라미터 압축 및 훈련시간 감소의 두 가지 측면에서 연합학습의 통신 비용을 낮출 수 있다.

압축 신호의 차원 H가 연합학습 시스템에 미치는 영향을 파악하기 위해 그림 5에 정확도 성능을 다양한 H값에 대해 평가한다. 연합학습 성능이 수렴하는

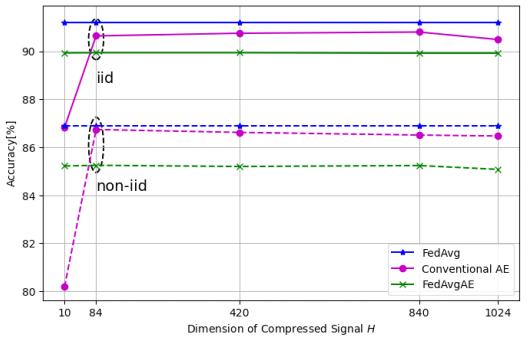

그림 5. 압축 차원에 따른 연합학습 정확도 성능 ($t=200$)
Fig. 5. Test accuracy performance with respect to the dimension of the compressed signal ($t=200$)

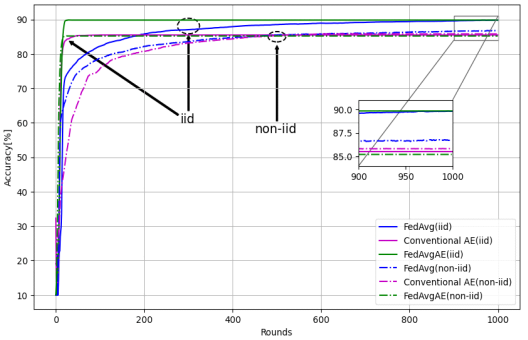

그림 6. 연합학습 정확도 성능 수렴성 ($K=400, H=420$)
Fig. 6. Convergence behavior of test accuracy performance ($K=400, H=420$)

200번째 round의 성능을 나타낸다. Conventional AE 기법의 경우 H를 증가시킴에 따라 정확도 성능이 비례하여 증가하였으며, 약 97%의 압축률을 달성하는 $H=420$ 이후로는 성능의 변화가 거의 없다. 제안하는 FedAvgAE 기법은 압축 차원 H에 큰 영향을 받지는 않지만, 수렴 성능이 FedAvg 기법 대비 소폭 감소한다. 그러나, 이러한 1% 이내의 성능 열화는 통신 효율성 개선에서 얻을 수 있는 이점과 비교하면 사소하다고 결론 내릴 수 있다.

클라이언트의 수 K에 대한 확장성을 평가하기 위해 그림 6에 연합학습 정확도 성능의 수렴성을 400개의 클라이언트가 존재하는 시스템에서 검증한다. 그림 4에서 수행한 모의실험과 동일하게 Conventional AE 및 FedAvgAE의 훈련은 $K=20$ 환경에서 진행되었다. 훈련된 오토인코더를 $K=400$ 상황에 바로 확장하여 검증 연합학습 시스템을 구성한다. 클라이언트의 수를 증가시킴에 따라, 각 클라이언트가 검증 연합학습 과정에서 관찰하는 데이터 집합은 훈련 데이터를 획득하는 상황과는 매우 다르다. 따라서, 본 모의실험을 통해 네트워크 크기 및 분산 데이터 집합에 대한 일반화 능력을 동시에 평가할 수 있다. 클라이언트가 많은 상황에서 제안하는 방법의 수렴 속도 및 정확도 성능이 모두 크게 개선되었으며, 이러한 현상은 non-iid 환경에서 두드러진다. 그에 비해 Conventional AE 기법은 non-iid 환경에서 더 천천히 수렴하므로 일반화 능력이 좋지 않다.

그림 7은 검증 환경에서의 클라이언트 수 K에 따른 연합학습의 정확도 성능을 평가한다. 모든 오토인코더 기반 기법들은 $K=20$ 환경에서 훈련되었다. 기존 FedAvg 및 Conventional AE 모두 클라이언트의 수가 증가하면 테스트 정확도가 감소한다. 클라이언트

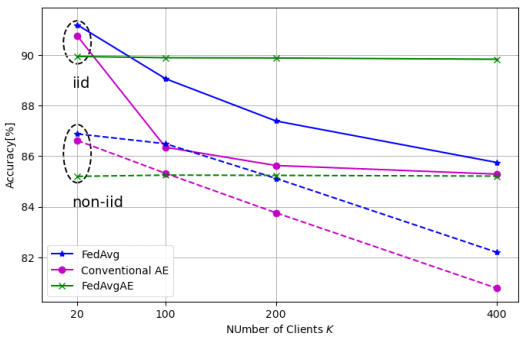

그림 7. 클라이언트 수에 따른 정확도 성능 ($H=420$, $t=200$)
Fig. 7. Test accuracy performance with respect to the number of clients ($H=420$, $t=200$)

가 많아짐에 따라 지역 데이터 집합이 더 잘게 분할되므로, 파편화된 지역 모델 파라미터를 정확히 취합하기 위해 더 많은 round가 필요하다. 그에 비해 FedAvgAE는 K와 관계없이 유사한 정확도 성능을 유지한다. 클라이언트의 수가 작은 환경 ($K\leq 100$)에서는 FedAvg 대비 소폭의 성능 열화가 관찰된다. 그러나, 많은 클라이언트가 존재하는 대규모 연합학습 네트워크 상황에서 제안하는 기법은 높은 정확도 성능을 보인다. 고차원 지역 파라미터를 저차원 압축 신호로 차원을 축소하는 과정에서 각 클라이언트의 중요 정보만을 취득할 수 있고, 이를 통해 서버는 파편화된 정보를 더 효과적으로 취합할 수 있다. 결론적으로, 제안하는 방법이 적절한 모델 압축을 통해 FedAvg의 확장성이 크게 개선함을 알 수 있다.

표 2는 다양한 연합학습 기법들의 수렴속도 성능을 나타낸다. 정확도 83% 달성에 필요한 통신 round 횟수를 iid 및 non-iid 상황에 대해 각각 표현하였다. FedAvg 알고리즘 대비 수렴에 필요한 round 수의 비

표 2. 연합학습 정확도 수렴속도 성능
Table 2. Convergence speed of test accuracy performance

IID			
K	*FedAvg*	*Coventional AE*	*FedAvgAE*
20	11	4 (0.36)	1 (0.09)
100	37	11 (0.30)	5 (0.14)
200	64	14 (0.22)	8 (0.13)
400	114	19 (0.17)	16 (0.14)
Non-IID			
K	*FedAvg*	*Coventional AE*	*FedAvgAE*
20	34	53 (1.56)	3 (0.09)
100	79	110 (1.40)	7 (0.09)
200	120	165 (1.40)	10 (0.08)
400	-	-	14

율을 괄호 안에 표기하였다. 수렴속도 비율이 1보다 작으면 FedAvg대비 빠르게 수렴하여 지연시간을 절감할 수 있고, 반대로 1보다 크면 느리게 수렴하여 더 많은 통신 비용이 발생한다. 모든 실험 환경에서 제안하는 기법이 FedAvg 및 Conventional AE 대비 더 빠르게 수렴하는 것을 확인할 수 있다. 이러한 우위는 non-iid 상황에서 두드러진다. 제안하는 기법은 여전히 빠른 수렴속도를 보이지만, Conventional AE는 오히려 더 많은 round를 필요로 한다. 특히 클라이언트의 수가 400개로 많은 상황에서는 기존 방법들은 목표 정확도 성능을 달성하지 못하지만, 제안하는 기법은 14회 만에 수렴한다. 이를 통해 FedAvgAE 기법이 비정형적인 클라이언트, 즉, non-iid 데이터 분포 상황에서 더 효과적으로 동작함을 알 수 있다. 따라서, FedAvgAE는 연합학습의 개인화 (personalization) 및 일반화 능력을 고취하는 기능을 내재적으로 학습한다고 결론지을 수 있다.

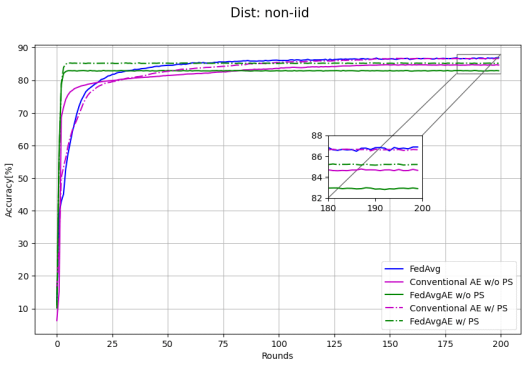

그림 8. 파라미터 공유 및 가중평균 동작의 영향 ($K=20$, $H=420$)
Fig. 8. Impact of parameter sharing and weighted averaging operations ($K=20$, $H=420$)

제안하는 FedAvgAE 기법의 주요 구성 요소는 오토인코더 파라미터 공유 (12)와 디코더의 가중평균 동작 (15)이다. 각 요인의 영향을 개별적으로 분석하기 위해 다음의 모의실험을 수행한다.

(1) FedAvgAE w/ PS: 파라미터 공유와 가중평균 동작을 모두 채택한 제안기법
(2) FedAvgAE w/o PS: 기법(1)에서 클라이언트마다 개별 오토인코더를 사용
(3) Conventional AE w/ PS: 오토인코더의 파라미터는 공유하지만 디코더의 가중평균 동작은 수행하지 않는 기존 기법
(4) Conventional AE w/o PS: 기법(3)에서 클라이언트마다 개별 오토인코더를 사용

기법(1)과 기법(3)의 성능을 비교하면 가중평균 연산 효과를 파악할 수 있다. 또한, 기법(2)와 기법(4)를 통해 파라미터 공유의 중요성을 알 수 있다.

상기 4개의 기법에 대한 non-iid 상황에서의 정확도 수렴성을 그림 8에 도시한다. 파라미터를 공유하면 제안하는 FedAvgAE 기법의 정확도 성능이 향상된다. 유사하게, Conventional AE 기법도 파라미터 공유를 통해 수렴 지점의 정확도 성능이 개선되지만, 수렴에 더 많은 round가 필요하다. “FedAvgAE w/ PS” 기법과 “Conventional AE w/ PS” 기법의 성능을 비교하면, 가중평균 연산 (15)를 통해 수렴속도를 크게 개선할 수 있음을 알 수 있다. 결론적으로, 파라미터 공유 및 디코더 가중평균 전략을 동시에 채택하는 FedAvgAE 기법을 활용하면 파라미터의 압축과 round 측면에서 모두 통신 비용을 크게 감축할 수 있다.

V. 결 론

본 논문은 FedAvg 알고리즘의 통신 오버헤드를 줄이기 위한 오토인코더 기반 모델 파라미터 압축 기법을 제안하였다. 클라이언트에 분산된 데이터 집합의 non-iid 특성 및 네트워크의 크기에 대한 확장성을 부여하기 위해 파라미터 공유 전략을 채택하였다. 또한, 서버의 디코더 신경망이 가중평균 연산을 직접 수행하도록 추론 과정 및 훈련 손실함수를 설계하였다. 모의실험을 통해 제안하는 FedAvgAE는 97%의 압축률을 달성하면서도 압축을 수행하지 않는 기존 FedAvg 알고리즘 대비 동일한 정확도 성능을 달성하기 위해 필요한 통신 지연시간을 약 18배 감축하였다. 제안하는 기법의 성능을 향상시키기 위해 오토인코더 최적

화 단계를 연합학습 알고리즘에 취합시켜 오토인코더와 연합학습 모델을 동시에 훈련하는 통합 알고리즘을 개발할 계획이다.

References

[1] H. B. McMahan, E. Moor, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTAT)*, 2017.

[2] S.-H. Park and H. Lee, "Completion time minimization of fog-RAN-assisted federated learning with rate-splitting transmission," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 10209-10214, Sep. 2022. (https://doi.org/10.1109/TVT.2022.3180747)

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50-60, May 2020. (https://doi.org/10.1109/MSP.2020.2975749)

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 12, pp. 1-19, Jan. 2019. (https://doi.org/10.1145/3298981)

[5] J. Konečný, H. B. McMahan, F. Yu, P. Richtárik, A. T. Suresh, and D. Bacon "Federated learning: Strategies for improving communication efficiency," *NIPS Wkshp. Private Multi-Party Mach. Learn.*, Barcelona, Spain, 2016.

[6] Y.-S. Jeon, M. M. Amiri, and N. Lee, "Communication-efficient federated learning over MIMO multiple access channels," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6547-6562, Oct. 2022. (https://doi.org/10.1109/TCOMM.2022.3198433)

[7] Y. Oh, N. Lee, Y.-S. Jeon, and H. Poor, "Communication-efficient federated learning via quantized compressed sensing," *IEEE Trans. Wireless Commun.*, to be published. (https://doi.org/10.1109/TWC.2022.3201207)

[8] C. M. Bishop, *Patten Recognition and Machine Learning*, Springer, 2006.

[9] S. M. Shah and V. K. N. Lau, "Model compression for communication efficient federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published. (https://doi.org/10.1109/TNNLS.2021.3131614)

[10] F. Staller, S. Wiedemann, K.-R. Muller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400-3413, Sep. 2020. (https://doi.org/10.1109/TNNLS.2019.2944481)

[11] S. Chandar, P. Chandran, R. Bhat, and A. Chakravarthi, "Communication optimization in large scale federated learning using autoencoder compressed weight updates," in *Proc. IJCAI*, 2021.

[12] Y. Yang, Y.-G. Hong, and J. Park, "Federated learning over wireless backhaul for distributed micro doppler radars: Deep learning aided gradient estimation," *IET Radar Sonar Navig.*, vol. 16, no. 5, May 2022. (https://doi.org/10.1049/rsn2.12227)

[13] D. Lee and H. Lee, "Communication-efficient federated learning with weighted-average autoencoder," in *Proc. KICS Summer Conf.* Jun. 2022.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[15] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4-24, Jan. 2021. (https://doi.org/10.1109/TNNLS.2020.2978386)

[16] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik, "Personalized federated learning using hypernetworks," in *Proc. ICML*, 2021.

이 도 윤 (Do-Yun Lee)

2022년 8월 : 부경대학교 정보통신공학과 학사
2022년 9월~현재 : 부경대학교 지능로봇공학과 석사 과정
<관심분야> 딥러닝, 연합학습

이 훈 (Hoon Lee)

2012년 2월 : 고려대학교 전기전자전파공학부 학사
2017년 2월 : 고려대학교 전기전자전파공학부 박사
2019년 3월~현재 : 부경대학교 부교수
<관심분야> 최적화, 기계학습
[ORCID:0000-0003-0753-8324]