

사물인터넷 환경에서 효과적인 분할 연합 학습을 위한 통신 오버헤드 최적화 연구 동향

정 한 길*, 김 중 욱*, 최 미 정^o

A Survey on Communication Overhead Optimization for Efficient Split Federated Learning in Internet of Things Environments

Han-Gil Jeong*, Jong-Wouk Kim*, Mi-Jung Choi^o

요 약

분할 연합 학습(Split Federated Learning)은 원시 데이터를 공유하지 않고도 자원 제약적인 장치들이 학습에 참여할 수 있도록 함으로써 IoT(Internet of Things) 환경에서 많은 주목을 받고 있다. IoT 장치들의 데이터를 활용하면 지능형 헬스케어, 제조, 도시, 교통 등 다양한 분야에서 지능형 서비스를 제공할 수 있다. 그러나 분할 연합 학습은 학습 과정에서 학습 관련 정보를 반복적으로 송수신하는 구조로 인해 통신 오버헤드를 발생시킨다. 이러한 오버헤드는 제한된 대역폭과 에너지 자원을 갖는 IoT 환경에서 학습 시간과 에너지 소모를 증가시키며, 결과적으로 실제 적용을 제약하는 요인으로 작용한다. 이에 따라 분할 연합 학습의 통신 오버헤드를 최적화하기 위한 연구가 진행되고 있다. 본 논문은 IoT 환경에서 분할 연합 학습을 적용하기 위해 고려해야 하는 주요 통신 오버헤드 문제를 소개하고, 각 연구가 다루는 세부적인 통신 오버헤드 문제를 중심으로 해결 전략을 체계적으로 비교한다.

Key Words : Internet of Things, Split Federated Learning, Communication Overhead Optimization

ABSTRACT

Split federated learning has attracted significant attention in Internet of Things (IoT) environments, as it enables resource-constrained devices to participate in training without sharing raw data. By utilizing data from IoT devices, intelligent services can be provided in healthcare, manufacturing, cities, and transportation. However, split federated learning incurs communication overhead due to the repeated exchange of training-related information during the training process. This overhead increases training time and energy consumption in IoT environments with limited bandwidth and energy resources, becoming a major constraint on practical deployment. Thus, recent studies on optimizing the communication overhead of split federated learning have been conducted. This paper introduces major communication overhead problems in split federated learning for IoT environments and systematically compares solution strategies for the specific problems addressed in each study.

※ 이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-지역지능화혁신인재양성사업의 지원(IITP-2026-RS-2023-00260267, 0.5)과 정부(대한민국 교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2024S1A5C3A01043273, 0.5).

• First Author : Kangwon National University Department of Data Science, jeong00400@kangwon.ac.kr

^o Corresponding Author : Kangwon National University Department of Computer Science and Kangwon National University IGP, in Medical Bigdata Convergence, mjchoi@kangwon.ac.kr

* Kangwon National University Department of Computer Science, jw.kim@kangwon.ac.kr

논문번호 : KNOM2026-01-009, Received July 28, 2026; Revised August 11, 2026; Accepted August 18, 2026

I. 서 론

최근 스마트폰, 웨어러블 기기, 드론, 그리고 스마트 센서와 같은 Internet of Things(IoT) 장치가 다양한 서비스 환경에 보급되면서, Deep Neural Network(DNN) 모델 학습에 활용할 수 있는 데이터가 지속적으로 생성되고 있다[1][2]. IoT 장치는 사용자의 행동, 생체 신호, 주변 환경, 설비 상태, 영상 및 음성 정보와 같은 다양한 데이터를 생성 및 수집하며, 이러한 데이터는 DNN 모델 기반 지능형 서비스를 구현하기 위한 주요 학습 자료로 활용된다[3]. 예를 들어, 지능형 헬스케어 서비스 분야에서는 웨어러블 기기 및 의료 IoT 장치의 데이터를 기반으로 사용자 건강 상태를 모니터링하거나 이상 징후를 탐지할 수 있다[1]. 지능형 제조 서비스 분야에서는 설비 센서 데이터를 활용하여 설비 고장을 예측하고 생산 품질을 관리할 수 있다[3][4]. 나아가 지능형 도시 및 지능형 교통 서비스 분야에서는 카메라, 차량, 도로 센서 등에서 생성되는 데이터를 활용하여 교통 혼잡 예측, 사고 감지와 같은 지능형 서비스를 제공할 수 있다[2][4].

일반적으로 DNN 모델 학습은 학습 데이터를 중앙 서버에 수집한 뒤, 해당 서버에서 모델을 학습하는 중앙 집중형 방식으로 수행된다[5]. 이는 높은 성능의 모델을 구축하기 위해서는 대규모 학습 데이터로부터 고수준 특징을 학습해야 하기 때문이다[6][7]. 그러나 IoT 환경에서 학습 데이터는 중앙 서버에 미리 저장되어 있는 것이 아니라, 여러 IoT 장치에서 분산되어 생성된다. 따라서 IoT 환경에서 중앙 집중형 학습 방식을 적용하기 위해서는 각 IoT 장치가 생성한 원시 데이터를 중앙 서버로 전송한 뒤, 서버에서 모델을 학습해야 한다[8]. 대규모 IoT 환경에서 생성되는 원시 데이터를 중앙 서버로 수집하는 과정은 높은 네트워크 대역폭을 요구하며, IoT 장치의 통신 지연과 에너지 소모를 증가시킬 수 있다[4][9]. 또한 원시 데이터는 사용자의 생체 정보, 얼굴 이미지, 위치 정보와 같은 민감한 정보를 포함할 수 있으므로, 이러한 데이터를 중앙 서버로 전송하는 과정에서 데이터 프라이버시 문제를 초래할 수 있다[8]. 따라서 IoT 환경에서는 원시 데이터를 중앙 서버로 직접 수집하지 않으면서도, 분산된 IoT 장치의 데이터를 활용하여 모델을 학습할 수 있는 방법이 필요하다.

이러한 필요성에 따라 연합학습(Federated Learning), 분할학습(Split Learning), 분할 연합 학습(Split Federated Learning)과 같은 분산 기계학습 방식이 연구되고 있다[10]. 세 방식 모두 분산된 장치들의 원시 데이터를 중앙 서버로 수집할 필요 없이 모델을 학습할 수 있다는 공통점을 가진다. 그러나 각 방식은 모델을 어느 위치에서 학습하는지, 서버와 클라이언트가 어떤 역할을 수행하는지, 그리고 학습 과정에서 모델 파라미터 또는 모델의 중간 연산 결과 중 무엇을 전송하는지에 따라 서로 다른 구조를 가진다[10].

연합학습은 여러 클라이언트가 자신의 원시 데이터를 외부로 공유하지 않고, 서버와 협력하여 하나의 전역 모델을 학습하는 분산 기계학습 방식이다[11]. 연합학습에서 서버는 현재 전역 모델을 여러 클라이언트에 전달하고, 각 클라이언트는 자신의 로컬 데이터로 모델을 학습한다. 이후 클라이언트는 원시 데이터가 아닌 학습된 모델 파라미터만을 서버로 전송하며, 서버는 여러 클라이언트의 모델 파라미터를 집계하여 새로운 전역 모델을 생성한다. 연합학습의 장점은 원시 데이터를 클라이언트 장치 내부에 유지하면서도, 여러 IoT 장치의 데이터를 전역 모델 학습에 활용할 수 있다는 점이다[12]. 그러나 연합학습에서는 각 클라이언트가 전체 모델의 파라미터를 보유하고 학습해야 하므로, 연산 능력, 메모리 그리고 에너지 자원이 제한된 IoT 장치에서는 복잡한 모델을 학습하는 데 부담이 발생할 수 있다[13].

분할학습은 모델을 클라이언트 측 모델과 서버 측 모델로 분할하고, 두 모델이 협력하여 하나의 모델을 학습하는 분산 기계학습 방식이다[14]. 분할학습에서 모델은 분할 계층인 cut layer를 기준으로 클라이언트 측 모델과 서버 측 모델로 분할된다. 클라이언트는 cut layer까지의 연산을 수행하고, 서버는 cut layer 이후의 연산을 수행한다. 학습 과정에서 클라이언트는 자신의 로컬 데이터를 클라이언트 측 모델에 입력하여 cut layer까지 forward propagation을 수행하고, 이 과정에서 생성된 모델의 중간 연산 결과인 activation을 서버로 전송한다. 서버는 수신한 activation을 서버 측 모델의 입력으로 사용하여 서버 측 모델의 forward propagation을 수행한다. 이후 서버는 backward propagation을 통해 서버 측 모델의 파라미터를 업데이트하고, 클라이언트 측 모델의 backward propagation에 필요한

gradient를 계산하여 클라이언트로 전송한다. 클라이언트는 수신한 gradient를 통해 클라이언트 측 모델 파라미터를 업데이트하며 학습을 마무리한다. 분할 학습의 장점은 클라이언트가 모델 전체가 아닌, cut layer까지만 학습함으로써 클라이언트 장치의 학습 부담을 줄일 수 있다는 점이다[14][15][16]. 그러나 일반적인 분할학습에서는 여러 클라이언트가 서버와 순차적으로 학습하기 때문에, 학습 과정에 참여하는 IoT 장치 수가 증가할수록 전체 학습 시간이 길어진다[17]. 이와 같은 학습 시간의 증가는 IoT 장치의 에너지를 소모시킬 수 있다[6].

분할 연합 학습은 분할학습의 모델 분할 구조와 연합학습의 병렬 학습 및 모델 집계 구조를 결합한 분산 기계학습 방식이다[18]. 분할 연합 학습에서는 일반적인 분할학습과 달리 여러 클라이언트가 클라이언트 측 모델을 병렬로 학습하고, 서버는 학습된 클라이언트 측 모델 파라미터를 집계하여 전역 클라이언트 측 모델을 생성한다. 연합학습과 분할학습의 장점을 결합함으로써 분할 연합 학습은 자원이 제한된 IoT 환경에서 IoT 장치 측의 학습 부담을 줄이고 전체 학습 시간을 줄일 수 있다는 장점을 가진다[10]. 최근에는 지능형 헬스케어, 차량 엣지 컴퓨팅, 지능형 도시, IoT 기반 이상 탐지 등 다양한 서비스 분야를 대상으로 분할 연합 학습을 적용하려는 연구가 활발히 진행되고 있다[4][10]. 이러한 연구 흐름은 자원 제약적인 IoT 환경에 분산 기계 학습을 적용함에 있어서 분할 연합 학습이 유망한 방식임을 보여준다[4][7][10].

이처럼 분할 연합 학습이 IoT 환경에서 유망한 분산 기계학습 방식으로 제시되고 있음에도, 이를 효과적으로 적용하기 위해서는 여러 도전 과제가 존재한다[4][10]. 그중 분할 연합 학습의 통신 오버헤드 문제는 제한된 네트워크 대역폭과 에너지 자원을 갖는 IoT 환경에서 전체 학습 시간을 지연시키고 에너지 소모를 증가시킬 수 있다[7]. 통신 오버헤드 문제는 학습 과정에 따라 크게 네 가지로 구분할 수 있다[10]. 첫 번째는 activation transmission 문제로, 이는 클라이언트가 생성한 activation을 서버로 전송하는 과정에서 발생하는 문제이다. 두 번째는 gradient transmission 문제로, 서버가 클라이언트 측 모델의 backward propagation을 위해 gradient를 클라이언트로 전송하는 과정에서 발생하는 문제이다. 세 번째는 model update 문제로, 전역 모델을 업데이트하기 위해 서버와 클라

이언트가 모델 파라미터를 송수신하는 과정에서 발생하는 문제이다. 네 번째는 control message 문제로, 클라이언트 선택, 모델 집계 시점 조율 등 서버와 클라이언트 간 학습 진행을 조율하기 위한 통신 과정에서 발생하는 문제이다[10]. 최근 이러한 통신 오버헤드 문제를 해결하고자 다양한 연구가 수행되고 있다. 또한, 분할 연합 학습의 통신 오버헤드 문제를 포함하여 여러 도전 과제에 대한 연구 동향 논문이 지속적으로 발표되고 있다[4][10][19][20].

기존 연구 동향 논문들은 분할 연합 학습의 방법, 응용 분야 및 도전 과제를 포괄적으로 정리하거나, 6G 네트워크와 같은 특정 응용 분야에 한정하여 분할 연합 학습을 다루었다[19][21]. 또한, 분할 연합 학습의 통신 오버헤드를 여러 문제 유형으로 구분하고, 이를 완화하기 위한 해결 전략을 분류하였다[10]. 세부적인 통신 오버헤드 문제에 따라 선택할 수 있는 해결 전략이 다양하므로, 해결 전략의 적용에는 세부 문제의 특성에 대한 고려가 필요하다. 그러나 각 연구가 다루는 세부적인 통신 오버헤드 문제를 중심으로 해결 전략을 체계적으로 비교한 연구는 부족하다.

본 논문에서는 IoT 환경에서 분할 연합 학습을 효과적으로 적용하기 위해 통신 오버헤드 문제를 중심으로 관련 연구 동향을 조사한다. 기존 연구 동향 논문에서는 통신 오버헤드를 네 가지 문제 유형으로 구분하였다[10]. 본 논문에서는 이 네 가지 문제 유형을 바탕으로 각 연구가 다루는 세부적인 통신 오버헤드 문제를 중심으로 해결 전략을 체계적으로 비교한다. 또한, 비교 결과를 바탕으로 IoT 환경에서 분할 연합 학습을 적용할 때 고려할 수 있는 해결 전략에 대해 논의한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 IoT 환경에 대해 간략히 설명하고, 연합학습, 분할 학습, 분할 연합 학습의 기본 구조와 학습 과정을 설명한다. 제 3장에서는 분할 연합 학습에서 발생하는 통신 오버헤드 문제를 activation transmission, gradient transmission, model update, control message로 구분하고, 각 문제에 대해 설명한다. 제 4장에서는 각 연구가 다루는 세부적인 통신 오버헤드 문제와 해결 전략을 체계적으로 정리한다. 마지막으로 제 5장에서는 본 논문의 내용을 요약하고, IoT 환경에서 분할 연합 학습을 적용할 때 고려할 수 있는 해결 전략에 대해 논의한다.

II. 배경지식

2.1 IoT 환경

IoT 환경의 핵심 제약 중 하나는 IoT 장치가 제한된 에너지 자원을 가진다는 점이다[22]. 특히 IoT 장치의 에너지는 데이터 처리 연산과 서버와의 데이터 송수신 과정에서 소모된다[23][24][25]. 따라서 IoT 환경에서 분산 기계학습을 효과적으로 적용하기 위해서는 IoT 장치의 모델 연산 비용과 통신 비용을 함께 고려해야 한다.

스마트폰, 웨어러블 기기, 드론과 같은 IoT 장치는 제한된 연산 능력과 메모리 용량을 가지고 있으며, 모델을 학습하기 위해서는 많은 연산량과 메모리 사용량을 요구한다[1]. 대표적인 예시로 512MB의 메모리와 Arm Cortex A7 CPU를 가지는 Hi3516E V200 SoC 이미지 처리 프로세서가 VGG-16 이미지 분류 모델을 효과적으로 학습하는 것은 불가능하다. VGG-16 모델은 528MB의 메모리 요구량과 1초당 16억 부동소수점 연산을 요구하기 때문이다[26]. 또한 IoT 장치가 모델을 수렴시키기 위해 많은 연산을 수행할 경우 학습 완료 시간이 증가하여 에너지 소모량이 커질 수 있다[6].

IoT 환경에서는 사용하는 통신 기술에 따라 가용 대역폭과 에너지 소모 특성이 달라질 수 있다. 예를 들어 Wi-Fi, Bluetooth, LoRaWAN과 같은 통신 기술은 서로 다른 데이터 전송률과 전력 소모 특성을 가진다[7][27]. 또한, IoT 환경에서 통신 상태는 고정적이지 않고 시간에 따라 변동될 수 있다. 특히 통신 상태가 좋지 않을 때는 IoT 장치의 데이터 전송 시간이 증가하고 에너지 소모량이 커질 수 있다[9]. 또한 IoT 장치의 에너지 소모량은 연산으로 인한 소모량보다 통신으로 인한 소모량이 더 크게 나타날 수 있으므로, 통신 오버헤드를 줄이는 것이 중요하다[7].

2.2 연합학습

연합학습은 분산 기계학습 프레임워크로, 구글의 키보드 패턴 예측 및 의료, 지능형 교통, 산업 자동화 분야의 다양한 사용 사례에 성공적으로 적용되었다[28][29][30]. 서버는 분산된 클라이언트들로부터 모델을 학습하기 위해 데이터를 수집함에 있어서 원시 데이터가 아니라 클라이언트의 모델 가중

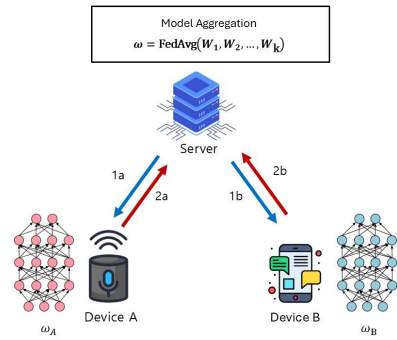


그림 1. 연합학습 과정

Fig. 1. Training process of Federated Learning

치를 집계한다[11]. 이를 통해 연합학습은 데이터 프라이버시를 보호하면서 여러 클라이언트와의 학습을 가능하게 한다[10].

그림 1은 장치 A와 장치 B가 전역 모델 학습에 참여하는 연합학습 과정을 보여준다. 먼저 서버는 전역 모델 ω 를 초기화한 후 장치 A와 장치 B에게 공유한다(그림 1.1a, 그림 1.1b). 그 다음, 각 장치들은 자신의 로컬 데이터를 이용하여 모델 가중치를 업데이트한다. 서버는 업데이트된 모델 가중치 ω_A , ω_B 를 집계한 후 전역 모델 ω 를 계산한다(그림 1.2a, 그림 1.2b). 이후 계산된 전역 모델 ω 를 다시 장치 A와 장치 B에게 공유하고 해당 과정을 모델이 수렴할 때까지 반복한다[20].

하지만 모델 전체를 학습하는 것은 많은 연산량을 요구한다. 특히, 자원 제약적인 IoT 장치에서는 모델을 수렴시키기 위해 많은 학습 시간이 필요하다[31]. 또한, 학습하는 모델의 파라미터 수가 많을수록 모델 가중치 크기가 커지므로 통신 오버헤드 문제가 발생한다[32].

2.3 분할학습

분할학습은 자원 제약적인 엣지 장치에서도 분산 기계학습을 가능하게 하는 방식으로서 등장한 프레임워크다[32]. 서버는 분산된 클라이언트들로부터 모델을 학습하기 위해 데이터를 수집함에 있어서 원시 데이터가 아니라 클라이언트의 중간 모델 연산 결과인 activation과 정답 label을 수신받는다[14]. 이를 통해 분할학습은 데이터 프라이버시를 보호하면서도 자원 제약적인 장치를 가진 클라이언트가 학습에 참여할 수 있도록 한다[10].

그림 2는 장치 A와 장치 B가 서버와 협력하여

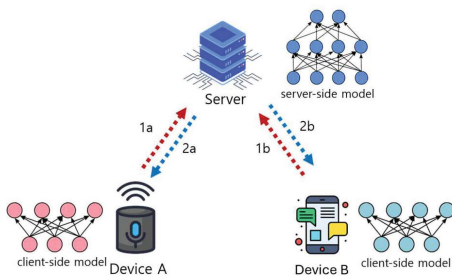


그림 2. 분할학습 과정
Fig. 2. Training process of Split Learning

모델을 학습하는 분할학습 과정을 보여준다. 먼저 모델은 특정 계층인 cut layer를 기준으로 클라이언트 측 모델과 서버 측 모델로 분할된다. 그 다음 각 장치들은 자신의 로컬 데이터를 자신의 클라이언트 측 모델에 입력하여 cut layer까지 forward propagation을 수행한다. 이 과정에서 생성된 activation은 정답 label과 함께 서버로 전송된다(그림 2.1a, 그림 2.1b). 서버는 수신한 activation을 서버 측 모델의 입력으로 사용하여 forward propagation을 이어서 수행하고, 수신한 정답 label을 기반으로 손실을 계산한다. 이후 backward propagation을 통해 서버 측 모델의 가중치를 업데이트하고 서버는 cut layer에 해당하는 gradient를 각 장치로 전송한다(그림 2.2a, 그림 2.2b). 장치 A와 장치 B는 수신한 gradient를 이용하여 클라이언트 측 모델의 backward propagation을 수행하고 모델 가중치를 업데이트한다. 이러한 과정은 모델이 수렴할 때까지 반복한다.

하지만 분할학습에서는 activation과 gradient를 빈번하게 송수신해야 하므로 통신 오버헤드가 발생한다[7]. 또한, 분할학습에서는 다수의 클라이언트가 서버와 순차적으로 학습을 수행하므로, 학습에 참여하는 클라이언트 수가 증가할수록 전체 학습 시간이 길어진다[17]. 이로 인해 학습 과정에 참여하는 IoT 장치들의 에너지 소모를 증가시킬 수 있다[6].

2.4 분할 연합 학습

분할 연합 학습은 분할학습의 모델 분할 구조와 연합학습의 모델 집계 구조를 결합한 분산 기계학습 프레임워크이다[18]. 이를 통해 분할

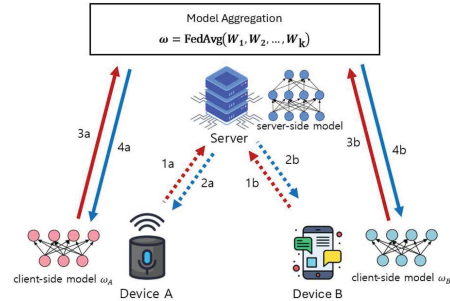


그림 3. 분할 연합 학습 과정
Fig. 3. Training process of Split Federated Learning

연합 학습은 데이터 프라이버시를 보호하면서도 자원 제약적인 장치를 가진 클라이언트들이 병렬적으로 학습에 참여할 수 있도록 한다[4][18]. 특히 일반적인 분할학습과 달리 여러 클라이언트가 순차적으로 학습하지 않고 병렬로 학습을 수행하므로, 클라이언트 수 증가에 따른 전체 학습 시간 증가를 완화할 수 있다[4].

그림 3은 장치 A와 장치 B가 협력하여 모델을 학습하는 분할 연합 학습 과정을 보여준다. 먼저 모델은 특정 계층인 cut layer를 기준으로 클라이언트 측 모델과 서버 측 모델로 분할된다. 그 다음, 각 장치들은 자신의 로컬 데이터를 클라이언트 측 모델에 입력하여 cut layer까지 forward propagation을 수행한다. 이 과정에서 생성된 activation은 정답 label과 함께 서버로 전송된다(그림 3.1a, 그림 3.1b). 서버는 수신한 activation을 서버 측 모델의 입력으로 사용하여 forward propagation을 이어서 수행하고, 수신한 정답 label을 기반으로 손실을 계산한다. 이후, backward propagation을 통해 서버 측 모델의 가중치를 업데이트하고 서버는 cut layer에 해당하는 gradient를 각 장치로 전송한다(그림 3.2a, 그림 3.2b). 장치 A와 장치 B는 수신한 gradient를 이용하여 클라이언트 측 모델의 backward propagation을 수행하고 모델 가중치를 업데이트한다. 마지막으로 일반적인 분할학습과 달리, 각 장치들은 업데이트된 클라이언트 측 모델 가중치를 서버로 전송한다(그림 3.4a, 그림 3.4b). 서버는 수신한 클라이언트 측 모델 가중치를 집계하여 새로운 전역 클라이언트 측 모델을 계산하고, 이를 다음 학습 라운드에서 장치 A와 장치 B에게 다시 공유한다(그림 3.3a, 그림 3.3b). 이러한 과정은 모

텔이 수렴할 때까지 반복한다[20].

하지만 분할 연합 학습에서는 일반적인 분할학습과 같이 클라이언트가 activation과 gradient를 빈번하게 송수신해야 한다. 또한 연합학습과 같이 클라이언트 측 모델 가중치도 주기적으로 송수신해야 한다[10]. 이로 인해 분할 연합 학습은 학습 과정에서 연합학습과 분할학습의 통신 오버헤드 문제를 모두 가진다. 또한, 다수의 기존 연구들이 안정적인 통신 상태를 가정하여 분할 연합 학습의 성능을 개선하려 한다[33]. 반면 IoT 환경은 불안정한 통신 상태를 가지므로 분할 연합 학습을 IoT 환경에 효과적으로 적용하기 위해서는 통신 오버헤드 문제를 고려하는 연구가 필요하다. 따라서, 본 논문에서는 통신 오버헤드 문제를 다루는 관련 연구들을 조사한다. 이를 위해 통신 오버헤드 문제가 구체적으로 무엇인지 설명한다.

III. 분할 연합 학습의 통신 오버헤드 문제

분할 연합 학습은 클라이언트와 서버가 협력하여 하나의 모델을 학습하는 구조이기 때문에, 학습 과정에서 여러 통신 과정이 반복적으로 발생한다. 특히 제한된 네트워크 대역폭과 에너지 자원을 갖는 IoT 환경에서는 이러한 통신 오버헤드가 전체 학습 시간을 증가시키고, IoT 장치의 에너지 소모를 높이는 요인으로 작용할 수 있다. Naseh et al.은 분할 연합 학습의 통신 오버헤드가 smashed data transmission, gradient transmission, model updates, control messages에서 발생한다고 설명한다[10]. 본 논문에서는 이를 기반으로 분할 연합 학습의 통신 오버헤드 문제를 activation transmission, gradient transmission, model update, control message의 네 가지 유형으로 구분하여 설명한다. 그림 4에서는 분할 연합 학습을 진행하면서 통신 오버헤드가 발생하는 과정을 보여준다.

- **Activation Transmission:** 클라이언트 장치가 cut layer까지 forward propagation을 수행한 뒤 생성한 activation을 서버로 전송하는 과정에서 발생한다. 분할학습 및 분할 연합 학습에서 activation은 서버 측 모델의 입력으로 사용되므로 서버가 forward propagation과 손실 계산을 이어서 수행하기 위해서는 클라이언트가 생성한 activation이 서버로 전송되어야 한다. 특히, 분할 연합 학습에서는 여러 클라이언트가 병렬적으로 학습에 참여하므로, 각 클라이언트가 미니

배치 단위로 생성한 activation이 서버로 빈번하게 전송된다. 업링크 통신에서 더 많은 에너지를 소모하는 IoT 환경 특성상, activation transmission은 에너지 소모량을 증가시키는 주요 원인으로 작용한다[7][34].

- **Gradient Transmission:** 서버가 cut layer에 해당하는 gradient를 클라이언트로 전송하는 과정에서 발생한다. 분할학습 및 분할 연합 학습에서 클라이언트 측 모델은 서버 측 모델에서 계산된 gradient를 수신해야 backward propagation을 이어서 수행할 수 있다. 특히, 분할 연합 학습에서는 여러 클라이언트가 병렬적으로 학습에 참여하므로, 각 클라이언트가 미니 배치 단위로 전송한 activation에 대응하는 gradient가 서버로부터 빈번하게 전송된다. 따라서 클라이언트는 서버의 gradient를 수신하기 위해 빈번하게 대기해야 하므로, 제한된 네트워크 대역폭을 가지는 IoT 환경에서는 gradient transmission이 전체 학습 시간을 증가시킬 수 있다[35]. 이는 gradient를 수신받는 과정과 함께 에너지 소모량을 증가시키는 주요 요인으로 작용한다[7].
- **Model Update:** 클라이언트 측 모델의 전역 모델을 생성하기 위해 클라이언트 측 모델 가중치를 서버와 송수신하는 과정에서 발생한다. 분할 연합 학습에서는 각 클라이언트가 activation과 gradient를 이용하여 클라이언트 측 모델을 업데이트한 뒤, 업데이트된 클라이언트 측 모델 가중치를 서버로 전송한다. 서버는 수신한 클라이언트 측 모델 가중치를 집계하여 새로운 클라이언트 측 모델의 전역 모델을 생성하고, 이를 다시 각 클라이언트에게 전송한다. 특히, 참여 클라이언트 수가 증가하거나 클라이언트 측 모델의 크기가 커질수록 서버가 수신하고 다시 전송해야 하는 모델 가중치의 통신량이 증가한다[36]. 따라서, 모든 클라이언트가 전역 모델 가중치를 수신하거나 서버가 클라이언트 측 모델 가중치를 모두 집계할 때까지 학습 라운드는 지연될 수 있다[38]. 이로 인해 제한된 네트워크 대역폭을 가지는 IoT 환경에서는 model update가 전체 학습 시간을 지연시키고, 모델 가중치 송수신 과정에서 IoT 장치의 에너지 소모량을 증가시키는 주요 요인으로 작용한다[2].
- **Control Message:** 서버와 클라이언트 간 학습 진행을 조율하기 위해 필요한 제어 정보를 송수신하는 과정에서 발생한다. Naseh et al.은

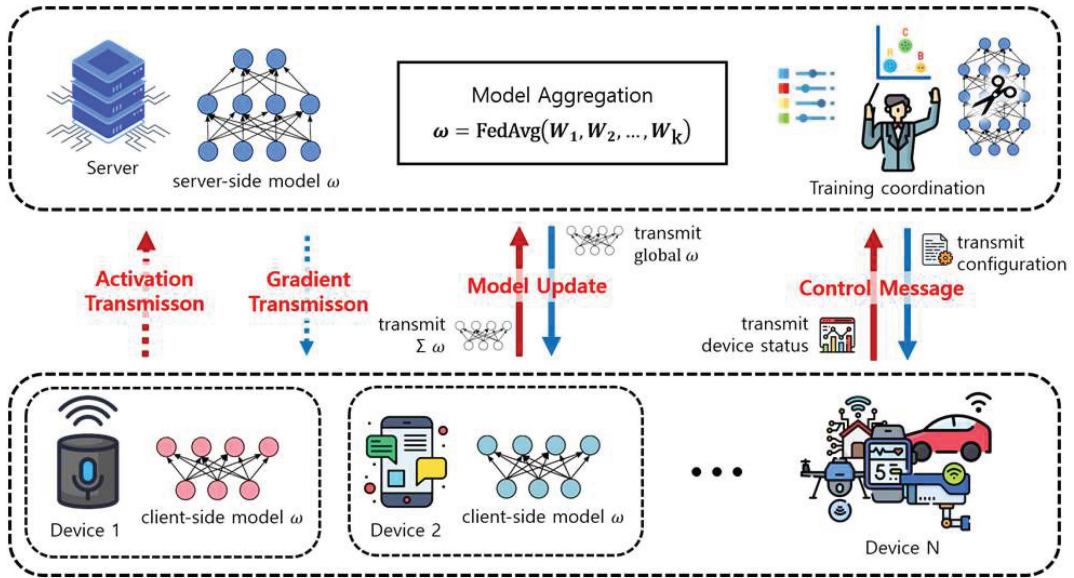


그림 4. 분할 연합 학습에서 통신 오버헤드가 발생하는 과정
Fig. 4. Process of communication overhead occurrence in split federated learning

control message를 분할 연합 학습의 통신 오버헤드 중 하나로 제시하고, 이를 학습 과정의 동기화 및 조율에 대한 오버헤드 문제로 설명한다 [10]. 분할 연합 학습에서는 학습 진행을 위해 참여 클라이언트 선택, cut layer 설정, 모델 집계 및 전역 모델 공유와 같은 제어 과정이 필요하다[38][39]. 따라서, 클라이언트 수가 증가하거나 클라이언트의 참여 상태가 자주 변하는 가변적인 IoT 환경에서는 클라이언트 선택, 자원 할당, 데이터 동기화와 같은 클라이언트 관리 과정이 복잡해질 수 있다. 이로 인해 모델 동기화와 학습 과정 조율에 어려움이 발생하여 전체 학습 시간이 지연될 수 있다[10][19].

IV. 통신 오버헤드 최적화 연구 동향

분할 연합 학습은 다양한 분야에서 높은 활용 가능성을 보이며, IoT 환경에도 이를 적용하려는 연구가 활발히 진행되고 있다. 그러나 분할 연합 학습을 IoT 환경에 효과적으로 적용하는 데 있어서 통신 오버헤드 문제가 핵심 제약으로 작용한다. 본 장에서는 이러한 통신 오버헤드 문제를 다룬 기존 연구들을 조사한다. 이를 위해 3장에서 구분한 통신 오버헤드 문제를 중심으로 기존 연구들을 분류하고, 각 연구의 해결 전략과 성능을 설명한다. 표 1에서

는 조사한 연구들을 통신 오버헤드 문제별로 구분하고, 각 문제에 대응하는 해결 전략과 각 연구에서 제시한 주요 성능 결과 및 정확도에 미치는 영향을 정리한다. 다만, 표 1에 정리한 주요 성능 결과는 각 연구에서 서로 다른 비교 대상, 데이터셋, 모델 및 평가 지표를 사용하여 산출되었으므로, 각 연구에서 제시한 주요 성능 결과를 직접 비교하기에는 한계가 있다.

Xing et al.은 에너지 하베스팅 기반 IoT 장치에서 activation 및 gradient 전송으로 인한 통신 에너지 소모를 줄이고자 하였다[7]. 이를 위해 제안 기법은 IoT 장치의 에너지 상태와 모델 훈련 손실 변화를 기반으로 activation 및 gradient 전송 여부를 선택적으로 결정한다. 구체적으로, IoT 장치는 현재 에포크(epoch)에서 모델 업데이트에 필요한 에너지가 충분하고 이전 업데이트 시점 대비 훈련 손실 변화가 특정 임계값 이상일 때 activation을 서버로 전송한다. 서버는 해당 activation을 이용하여 서버 측 학습을 수행한 뒤, IoT 장치가 backward propagation을 수행할 수 있도록 gradient를 전송한다. 반면, 위 조건을 만족하지 않아 클라이언트 측 모델이 업데이트되지 않는 경우에는 gradient 전송을 생략하고, 서버는 이전 에포크에서 저장한 activation을 재사용한다. 이를 통해 제안 기법은 activation 및 gradient 전송 횟수를 줄이고, 통신 에

표 1. 분할 연합 학습에서 통신 오버헤드 최적화 연구 비교

Table 1. Comparison of communication overhead optimization research in Split Federated Learning

Research	Problem	Solution strategy	Performance gain	Accuracy impact
[7]	activation transmission	selective transmission, caching and reuse	energy consumption ↓ 28.8 - 80.5%	maintained
	gradient transmission	selective transmission		
[38]	activation transmission	dynamic split	convergence time ↓ $7.7\times$	maintained
	gradient transmission			
	model update	aggregation interval adjust		
[39]	activation transmission	dynamic split	training time ↓, energy consumption ↓	maintained
	gradient transmission	dynamic split, gradient aggregation		
	model update	client scheduling		
	control message			
[40]	activation transmission	selective transmission	communication load ↓ 49.7 - 98.4%	2.6% - 7.3% ↓
	gradient transmission			
[41]	gradient transmission	gradient aggregation	training latency ↓, per-round latency ↓ 50%	maintained
[42]	activation transmission	dynamic split	training time ↓ 48%, energy consumption ↓ 62.8%	maximum 32% ↑
	gradient transmission			
	model update			
[43]	control message	client scheduling, client clustering	idle time ↓ 100ms	9% ↑
[44]	activation transmission	compression	communication load ↓ 45.9 - 95.9%	0.2% - 18.5% ↓
	gradient transmission	selective transmission		
[45]	activation transmission	compression	communication load ↓ $4\times$	maintained
	gradient transmission			
[46]	gradient transmission	gradient aggregation	training latency ↓, communication load ↓	maximum 5.8% ↑
	model update	aggregation interval adjust		

너지 소모를 감소시킨다. 실험 결과, 제안 기법은 모델 업데이트에 필요한 에너지가 충분할 때마다 업데이트를 수행하는 energy-aware selective updating 방식과 비교하여 정확도 감소를 0.5% 이내로 유지하였다[37]. 또한, 통신으로 인한 에너지 소모를 28.8%에서 80.5%까지 줄였다.

Lin et al.은 자원 제약 엣지 네트워크 환경에서 activation, gradient, 모델 가중치 전송으로 인해 발생하는 통신 오버헤드와 학습 지연을 줄이고자 하였다[38]. 이를 위해 제안 기법은 학습 수렴에 필요한 조건을 만족하면서 전체 학습 시간을 줄이도록 cut layer와 클라이언트 측 모델 집계 간격을 적응적으로 조정한다. 제안 기법은 각 엣지

장치의 연산 능력, 네트워크 처리량, 계층별 activation 및 gradient 크기, 모델 가중치 크기를 고려하여 cut layer와 클라이언트 측 모델 집계 간격을 결정한다. 이후 학습 수렴 조건을 만족하는 범위에서 전체 수렴 시간을 최소화하는 cut layer와 클라이언트 측 모델 집계 간격을 선택한다. 이를 통해 네트워크 및 연산 자원 상태에 따라 activation과 gradient의 전송 크기, 모델 가중치 전송 빈도를 함께 조절하여 통신 오버헤드와 학습 시간을 감소시킨다. 실험 결과, cut layer와 클라이언트 측 모델 집계 간격을 임의로 결정하는 기존 기법과 비교하여 정확도를 유지하면서, 수렴 속도를 최소 7.7배 향상시켰다.

Ao et al.은 자원 제약적인 무선 네트워크 환경에서 activation, gradient, 모델 가중치 전송과 동기화 및 조정 과정에서 발생하는 오버헤드로 인해 증가하는 학습 시간과 에너지 소모를 줄이고자 하였다[39]. 저자들은 이를 해결하고자 학습 지연과 에너지 소모의 가중합을 기준으로 cut layer와 학습 참여 클라이언트를 선택한다. cut layer 선택을 통해 클라이언트와 서버 간에 전송되는 activation 및 gradient 크기와 장치 측 연산량을 조절한다. 또한, 각 학습 라운드에서 클라이언트의 연산 성능과 무선 채널 상태를 고려하여 학습에 참여할 클라이언트를 선택함으로써, 학습 속도가 느린 클라이언트로 인해 서버와 다른 클라이언트가 다음 학습 단계로 넘어가지 못하고 대기하는 시간을 줄였다. 추가로, 저자들이 제안한 기법은 서버 측 backward propagation 이후 각 클라이언트의 gradient를 평균화하여 클라이언트로 전송하고, 연속으로 선택된 클라이언트에는 장치 측 모델을 다시 전송하지 않는다. 또한, 라운드 종료 후에는 채널 상태가 가장 좋은 클라이언트 하나만 모델 가중치를 서버로 전송한다. 이를 통해, activation 및 gradient 전송 크기, 모델 가중치 전송 빈도, 그리고 학습 과정에서 발생하는 대기 시간을 함께 줄여 학습 지연과 에너지 소모를 감소시킨다. 실험 결과, 저자들이 제안한 기법은 기존 연합학습 및 기존 순차적 분할학습 방식과 비슷한 정확도에 도달하면서, 동일 정확도 도달에 필요한 전체 학습 시간을 줄였다.

Mu et al.은 분할 연합 학습에서 activation 및 gradient 전송으로 인해 발생하는 통신 오버헤드를 줄이고자 하였다[40]. 이를 위해 저자들은 클라이언트 측 모델에 보조 네트워크를 추가하여, 클라이언트가 서버로부터 gradient를 수신하지 않고도 모델을 업데이트하는 방안을 제안했다. 클라이언트는 모든 미니배치마다 activation을 서버로 전송하지 않고, 미리 설정된 미니배치 간격마다 activation을 서버로 전송한다. 반면, activation을 전송하지 않는 미니배치에서는 보조 네트워크의 손실 함수를 기반으로 로컬 업데이트를 수행한다. 이를 통해 제안 기법은 activation 전송 횟수를 줄이고, gradient 전송을 제거하여 통신 오버헤드를 감소시킨다. 실험 결과, 제안 기법은 기존 분할 연합 학습과 비교하여, 약 2.6%에서 7.3%의 정확도 감소와 함께 통신량을 49.7%에서

98.4%까지 줄였다.

Lin et al.은 자원 제약적인 무선 엣지 네트워크 환경에서 gradient 전송으로 인해 발생하는 통신 오버헤드와 학습 지연을 줄이고자 하였다[41]. 이를 위해, 저자들은 서버 측 모델의 마지막 계층에서 일부 gradient를 먼저 집계한 뒤 backward propagation을 수행하는 방식을 제안했다. 서버는 집계 비율에 따라 각 엣지 장치의 미니배치 내 일부 gradient를 장치 간 평균화하고, 나머지 gradient는 개별적으로 유지한다. 이후, 집계된 gradient는 모든 엣지 장치로 전송하고, 집계되지 않은 gradient는 해당 엣지 장치로 각각 전송한다. 이를 통해 제안 기법은 backward propagation 과정에서 처리해야 하는 gradient 수와 엣지 장치로 전송해야 하는 gradient 크기를 줄여 통신 오버헤드와 학습 지연을 감소시킨다. 실험 결과, 제안 기법은 분할학습 및 분할 연합 학습과 유사한 정확도를 유지하면서, 목표 정확도 도달에 필요한 학습 시간을 줄였다.

Samikwa et al.은 서로 다른 IoT 기기를 사용하는 이기종 IoT 환경 및 데이터 분포가 다른 non-IID 환경에서 activation, gradient, 모델 가중치 전송으로 인해 발생하는 통신 오버헤드와 학습 지연을 줄이고 정확도 저하를 완화하는 방안을 제안 했다[42]. 저자들은 각 학습 라운드에서 네트워크 처리량과 모델의 계층별 연산 시간을 측정하고, 이를 기반으로 IoT 장치별 모델 분할 지점을 동적으로 결정한다. 또한, 장치가 보유한 데이터 분포의 유사성을 반영하기 위해, 클라이언트 측 모델 계층의 유사도를 기반으로 학습 참여 장치를 클러스터링한다. 모델 분할 지점이 달라지면 IoT 장치와 서버에서 각각 연산되는 모델의 계층 수가 달라진다. 이에 따라 서버로 전송해야 하는 activation 크기, IoT 장치로 전송해야 하는 gradient 크기, 그리고 모델 가중치 크기도 함께 달라진다. 이를 통해 제안 기법은 IoT 장치의 자원 상태에 따라 모델 분할 지점을 동적으로 결정하여 activation, gradient 및 모델 가중치 전송으로 인한 통신 오버헤드와 학습 시간을 줄인다. 또한, 유사한 데이터 특성을 가진 장치들을 함께 학습시켜 non-IID 환경에서의 정확도를 개선한다. 실험 결과, 제안 기법은 non-IID 환경에서 기존 분할 연합 학습 및 연합학습과 비교하여 정확도를 최대 32% 향상시키고, IoT 장치의 에너지 소모를 최대 62.8% 감소시키며, 학습 시간을 최대 48% 줄였다.

Wazze et al.은 자원 제약적인 IoT 환경에서 클라이언트의 자원 이질성과 무작위 클라이언트 선택으로 인해 발생하는 동기화 및 조정 오버헤드와 학습 지연을 줄이고자 하였다[43]. 제안 기법은 자원 상태가 유사한 클라이언트를 클러스터로 그룹화하고, 각 학습 라운드에서 자원 가용성과 보유한 데이터 특성을 고려하여 학습 참여 클라이언트를 선택한다. 자원에 따라 장치를 클러스터링할 때 자원이 부족한 장치들은 제외시킨다. 이후, 자원 사용량 예측 모델을 통해 각 클라이언트의 동적 자원 사용량을 추정하고, 유전 알고리즘을 이용하여 고유 사용자 수, 데이터 샘플 수, 선택 이력, 처리 성능 차이를 고려해 각 클러스터에서 학습에 참여할 클라이언트 집합을 선택한다. 이를 통해, 제안 기법은 자원 상태가 유사한 클라이언트들이 함께 학습에 참여하도록 하여 학습 속도가 느린 클라이언트로 인해 발생하는 대기 시간을 줄이고 학습 과정에서 발생하는 동기화 및 조정 오버헤드를 완화한다. 실험 결과, 무작위 클라이언트 선택을 사용하는 기존 클러스터 기반 분할 연합 학습보다 정확도를 약 9% 향상시켰으며, 평균 대기 시간을 약 100 ms 줄였다.

Ayad et al.은 IoT 환경에서 분할학습 적용 시 activation 및 gradient 전송으로 인해 발생하는 통신 오버헤드를 줄이고자 하였다[44]. 이를 위해 제안 기법은 오토인코더를 이용하여 activation을 압축하고 적응형 임계값을 기반으로 gradient 전송 여부를 선택적으로 결정한다. 클라이언트는 cut layer에서 생성된 activation을 인코더를 통해 더 작은 잠재 벡터로 변환한 뒤 서버로 전송하고, 서버는 디코더를 통해 이를 복원하여 서버 측 학습을 수행한다. 서버는 forward propagation 이후 계산된 손실이 임계값보다 큰 경우에만 gradient를 클라이언트로 전송하고, 손실이 임계값보다 작은 경우에는 gradient 전송을 생략한다. 이를 통해 제안 기법은 activation 전송 크기와 gradient 전송 횟수를 함께 줄여 통신 오버헤드를 감소시킨다. 실험 결과, 저자들은 약 0.2%에서 18.5%의 정확도 감소와 함께 통신량을 45.9%에서 95.9%까지 줄였다.

Mudvari et al.은 IoT 및 모바일 엣지 환경에서 분할학습 적용 시 activation 및 gradient 전송으로 인해 발생하는 통신 오버헤드와 학습 지연을 줄이고자 하였다[45]. 저자들은 압축-복원 모듈을 이용하여 cut layer에서 생성된 activation을 압축하고, 학습 과정에서 압축 수준을 적응적으로 조정하는

방안을 제시했다. 클라이언트는 cut layer까지 forward propagation을 수행한 뒤 생성된 activation을 압축하여 서버로 전송하고, 서버는 이를 복원한 뒤 서버 측 모델의 forward propagation과 backward propagation을 수행한다. 이후 서버는 앞서 적용된 activation의 압축 수준에 따라 크기가 압축된 gradient를 클라이언트로 전송한다. 이를 통해 학습 초기에는 높은 압축률을 적용하여 통신량을 줄이고, 학습 후반에는 압축 수준을 완화하여 정확도 저하를 줄인다. 따라서, 제안 기법은 activation 전송 크기와 gradient 전송 크기를 함께 줄여 통신 오버헤드와 학습 지연을 감소시킨다. 실험 결과, 일반적인 분할학습 방식과 비교하여 정확도 저하 없이 학습 과정의 통신량을 약 4배 줄였다.

Oh et al.은 분할 연합 학습에서 gradient 전송과 클라이언트 측 모델 집계로 인해 발생하는 통신 오버헤드와 학습 지연을 줄이고자 하였다[46]. 이를 위해 제안 기법은 서버에서 클라이언트로 전송되는 gradient를 평균화하고, 클라이언트 측 모델 집계 간격을 조정한다. 서버는 클라이언트별로 계산된 gradient와 여러 클라이언트의 평균 gradient를 클라이언트들에게 함께 전송한다. 이때, 클라이언트별로 서로 다른 gradient를 각각 전송하는 과정에서 발생하는 다운로드 통신 오버헤드를 줄인다. 또한, 클라이언트 측 모델 집계 간격을 조정하여 클라이언트가 모델 가중치를 서버로 업로드하는 과정에서 발생하는 업링크 통신 비용을 조절한다. 한편, 제안 기법은 클라이언트가 업로드한 activation을 서버에서 조합하여 서버 측 학습에 활용함으로써 정확도를 개선한다. 이를 통해 gradient 전송, 모델 집계 과정에서 발생하는 통신 오버헤드, 그리고 학습 지연을 감소시키고 정확도를 개선한다. 실험 결과, 기존 분할 연합 학습 기법과 비교하여 정확도를 최대 5.8% 향상했으며, gradient 전송과 모델 집계 과정에서 발생하는 통신량과 학습 지연을 줄였다.

V. 결 론

본 논문에서는 IoT 환경에서 분할 연합 학습을 적용할 때 핵심 제약으로 작용하는 통신 오버헤드 문제를 중심으로 관련 연구들을 조사하였다. 기존 연구 동향 논문은 통신 오버헤드를 smashed data transmission, gradient transmission, model update,

control message의 네 가지 문제 유형으로 구분하고, 이를 완화하기 위한 해결 전략을 분류하며 관련 기법이 정확도에 미치는 영향을 비교하였다. 본 논문은 동일한 네 가지 문제 유형을 바탕으로 각 연구가 다루는 세부적인 통신 오버헤드 문제와 해결 전략을 체계적으로 비교하였다. 또한, 각 연구의 주요 성능 결과와 정확도에 미치는 영향을 정리하였다. 다만, 각 연구는 서로 다른 항목을 평가하고 결과를 서로 다른 단위로 제시하므로, 해결 전략 간 성능을 직접 비교하기에는 한계가 있다.

관련 연구들은 선택적 전송, 압축, 동적 분할, gradient 집계, 클라이언트 선택 및 클러스터링, 모델 집계 간격 조정과 같은 다양한 전략을 제안하였다. 조사 결과, 기존 연구들은 주로 activation transmission 및 gradient transmission 문제 최적화에 집중되어 있었으며, model update와 control message 문제를 직접적으로 다루는 연구는 상대적으로 부족하였다. 따라서, IoT 환경에서 분할 연합 학습을 효과적으로 적용하기 위해서는 activation 및 gradient 전송뿐만 아니라, 모델 가중치 송수신과 학습 과정 조율에서 발생하는 통신 오버헤드를 함께 고려하는 연구가 필요하다.

향후 연구로는 model update와 control message 문제를 중심으로, 통신 오버헤드 최적화 방법을 연구할 계획이다. 또한, 실제 IoT 환경에서는 네트워크 상태와 장치의 자원 상태가 시간에 따라 변할 수 있으므로, 이를 반영하여 activation 전송, gradient 전송, 모델 가중치 송수신, 학습 과정 조율에 따른 통신 오버헤드를 함께 줄일 수 있는 방법을 연구하고자 한다. 이를 통해 자원 제약적인 IoT 환경에서 분할 연합 학습의 통신 효율성을 향상시키고, 실제 서비스 환경에서의 적용 가능성을 높이하고자 한다.

References

- [1] I. Siam, H. Ahn, L. Liu, S. Alam, H. Shen, Z. Cao, N. Shroff, B. Krishnamachari, M. Srivastava, and M. Zhang, "Artificial intelligence of things: A survey," *ACM Transactions on Sensor Networks*, Vol. 21, No. 1, Article 9, pp. 1 - 75, Jan. 2025.
- [2] E. Dritsas, M. Trigka, "Federated learning for iot: A survey of techniques, challenges, and applications," *Journal of Sensor and Actuator Networks*, Vol. 14, No. 1, pp. 1 - 25, Jan. 2025.
- [3] H. Fu, J. Rao, F. Deng, Y. Wang, B. Zhao, Z. Liu, H. Guan, P. H. Malinowski, and L. Xu, "AIoT: Artificial intelligence and the internet of things for monitoring and prognosis of systems and structures," *IEEE Transactions on Instrumentation and Measurement*, Vol. 74, Article 9700232, pp. 1 - 32, Jun. 2025.
- [4] Q. Duan, S. Hu, R. Deng, and Z. Lu, "Combined federated and split learning in edge computing for ubiquitous intelligence in internet of things: State-of-the-art and future directions," *Sensors*, Vol. 22, No. 16, pp. 1 - 32, Aug. 2022.
- [5] A. Arzovs, J. Judvaitis, K. Nesenbergs, and L. Selavo, "Distributed learning in the iot - Edge - Cloud continuum," *Machine Learning and Knowledge Extraction*, Vol. 6, No. 1, pp. 283 - 315, Feb. 2024.
- [6] Y. Gao, M. Kim, C. Thapa, S. Abuadbbba, Z. Zhang, S. A. Camtepe, H. Kim, and S. Nepal, "Evaluation and optimization of distributed machine learning techniques for Internet of Things," *IEEE Transactions on Computers*, Vol. 71, No. 10, pp. 2538 - 2552, Oct. 2022.
- [7] X. Chen, J. Li, and C. Chakrabarti, "Energy and loss-aware selective updating for splitfed learning with energy harvesting-powered devices," *Journal of Signal Processing Systems*, Vol. 94, No. 10, pp. 961 - 975, Oct. 2022.
- [8] B. Radović, M. Canini, and V. Pejović, "Review and comparative evaluation of resource-adaptive collaborative training for heterogeneous edge devices," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, Vol. 10, No. 1, pp. 1 - 35, Mar. 2025.
- [9] E. Samikwa, A. Di Maio, and T. Braun, "ARES: Adaptive resource-aware split learning for Internet of Things," *Computer Networks*, Vol. 218, Art. No. 109380, pp. 1 - 13, Dec. 2022.
- [10] D. Naseh, A. Bozorgchenani, S. S. Shinde, and D. Tarchi, "Unlocking distributed intelligence: A comprehensive survey on federated split learning's evolution, challenges, and future frontiers," *Computer Networks*, Vol. 285, pp. 1 - 25, Mar. 2025.

- Article 112402, pp. 1 – 32, May 2026.
- [11] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” *arXiv preprint arXiv:1610.05492*, pp. 1 – 10, Oct. 2016.
- [12] C. Thapa, M. A. P. Chamikara, and S. A. Camtepe, “Advancements of federated learning towards privacy preservation: From federated learning to split learning,” *arXiv preprint arXiv:2011.14818*, Nov. 2020.
- [13] J. Hu, Y. Liang, Y. Chen, G. Liu, W. Chen, and L. Duan, “Performance optimization of split federated learning in heterogeneous edge computing environments,” *IEEE Transactions on Industrial Informatics*, Vol. 22, No. 1, pp. 154 – 164, Jan. 2026.
- [14] O. Gupta and R. Raskar, “Distributed learning of deep neural network over multiple agents,” *Journal of Network and Computer Applications*, Vol. 116, No. 15, pp. 1 – 8, Aug. 2018.
- [15] M. Le, T. Huynh-The, T. Do-Duy, T. V. Vu, W. J. Hwang, and Q. V. Pham, “Applications of distributed machine learning for the internet-of-things: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, Vol. 27, No. 2, pp. 1053 – 1100, Apr. 2023.
- [16] B. Radović, M. Canini, S. Horváth, V. Pejović, and P. Vepakomma, “Towards a unified framework for split learning,” in *Proc. of the 5th Workshop on Machine Learning and Systems*, Rotterdam, Netherlands, pp. 183 – 191, Apr. 2025.
- [17] S. Oh, J. Park, P. Vepakomma, S. Baek, R. Raskar, M. Bennis, and S.-L. Kim, “LocFedMix-SL: Localize, federate, and mix for improved scalability, convergence, and latency in split learning,” in *Proc. of the ACM Web Conference 2022*, Lyon, France, pp. 3347 – 3357, Apr. 2022.
- [18] C. Thapa, M. A. P. Chamikara, S. Camtepe, and L. Sun, “SplitFed: When federated learning meets split learning,” in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 36, No. 8, Virtual, pp. 8485 – 8493, Jun. 2022.
- [19] G. S. Hukkeri, R. H. GOUDAR, G. M. DHANANJAYA, V. N. RATHOD, and S. ANKALAKI, “A comprehensive survey on split-fed learning: Methods, innovations, and future directions,” *IEEE Access*, Vol. 13, pp. 46312 – 46333, Mar. 2025.
- [20] W. Zhang, D. Zhan, H. Yu, L. Zhang, B. Zhao, X. Du, and Z. Tian, “Distributed machine learning for next-generation communication networks: A survey on privacy, fairness, efficiency, and trade-offs,” *Information Fusion*, Vol. 126, Article 103657, pp. 1 – 28, Feb. 2026.
- [21] H. HAFI, B. BRIK, P. A. FRANGOUDIS, A. KSENTINI, and M. BAGAA, “Split Federated Learning for 6G Enabled-Networks: Requirements, Challenges, and Future Directions,” *IEEE Access*, Vol. 12, pp. 9890 – 9930, Jan. 2024.
- [22] F. Samie, L. Bauer, and J. Henkel, “From cloud down to things: An overview of machine learning in internet of things,” *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 4921 – 4934, Jun. 2019.
- [23] S. Liu, B. Guo, C. Fang, Z. Wang, S. Luo, Z. Zhou, and Z. Yu, “Enabling resource-efficient AIoT system with cross-level optimization: A survey,” *IEEE Communications Surveys & Tutorials*, Vol. 26, No. 1, pp. 389 – 427, Sept. 2024.
- [24] I. Nikoloska and N. Zlatanov, “Data selection scheme for energy efficient supervised learning at IoT nodes,” *IEEE Communications Letters*, Vol. 25, No. 3, pp. 859 – 863, Mar. 2021.
- [25] H. Yin, Z. Wang, and N. K. Jha, “A hierarchical inference model for internet-of-things,” *IEEE Transactions on Multi-Scale Computing Systems*, Vol. 4, No. 3, pp. 260 – 271, Sept. 2018.
- [26] Y. Wang, C. Yang, S. Lan, L. Zhu, and Y. Zhang, “End-Edge-Cloud collaborative computing for deep learning: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, Vol. 26, No. 4, pp. 2647 – 2683,

- Apr. 2024.
- [27] A. P. Behera, P. Daubaris, I. Bravo, J. Gallego, R. Morabito, J. Widmer, and J. P. Champati, "Exploring the boundaries of on-device inference: When tiny falls short, go hierarchical," *IEEE Internet of Things Journal*, Vol. 12, No. 18, pp. 37456 – 37470, Sept. 2025.
 - [28] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," *arXiv preprint arXiv:1812.02903*, pp. 1 – 9, Dec. 2018.
 - [29] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, Vol. 14, No. 1-2, pp. 1 – 210, Jun. 2021.
 - [30] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, Vol. 68, No. 2, pp. 1146 – 1159, Feb. 2020.
 - [31] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen, "FedMask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking," in *Proc. of the 19th ACM Conference on Embedded Networked Sensor Systems*, Coimbra, Portugal, pp. 42 – 55, Nov. 2021.
 - [32] J. Park, S. Samarakoon, A. Elgabli, J. Kim, M. Bennis, and S.-Lyun Kim, "Communication-efficient and distributed learning over wireless networks: Principles and applications," *Proceedings of the IEEE*, vol. 109, No. 5, pp. 796 – 819, May 2021.
 - [33] S. Rahman, J. Arora, S. Pal, and C. Karmakar, "DR-SFL: Dynamic and resource-aware split federated learning for semantic-enabled 6G industrial iot systems," *IEEE Communications Standards Magazine*, Vol. 10, No. 1, pp. 1 – 8, Jun. 2026.
 - [34] J. Lee, M. Seif, J. Cho, and H. V. Poor, "Exploring the privacy-energy consumption tradeoff for split federated learning," *IEEE Network*, vol. 38, no. 6, pp. 388 – 395, Nov. 2024.
 - [35] D. J. Han, H. I. Bhatti, J. Lee, and J. Moon, "Accelerating federated learning with split learning on locally generated losses," in *Proc. of the ICML 2021 workshop on federated learning for user privacy and data confidentiality*, Vol. 2, No. 3, Virtual, pp. 1 – 9, Sept. 2021.
 - [36] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed comparison of communication efficiency of split learning and federated learning," *arXiv preprint arXiv:1909.09145*, pp. 1 – 5, Sept. 2019.
 - [37] X. Chen, J. Li, and C. Chakrabarti, "Communication and computation reduction for split learning using asynchronous training," in *Proc. of the 2021 IEEE Workshop on Signal Processing Systems*, Coimbra, Portugal, pp. 76 – 81, Oct. 2021.
 - [38] Z. Lin, G. Qu, W. Wei, X. Chen, and K. K. Leung, "AdaptSFL: Adaptive split federated learning in resource-constrained edge networks," *IEEE Transactions on Networking*, Vol. 33, No. 6, pp. 2993 – 3008, Dec. 2025.
 - [39] H. Ao, H. Tian, and W. Ni, "Federated split learning for edge intelligence in resource-constrained wireless networks," *IEEE Transactions on Consumer Electronics*, Vol. 71, No. 2, pp. 4451 – 4463, May 2025.
 - [40] Y. Mu and C. Shen, "Communication and storage efficient federated split learning," in *Proc. of the IEEE International Conference on Communications*, Rome, Italy, pp. 2976 – 2981, Oct. 2023.
 - [41] Z. Lin, G. Zhu, Y. Deng, X. Chen, Y. Gao, K. Huang, and Y. Fang, "Efficient parallel split learning over resource-constrained wireless edge networks," *IEEE Transactions on Mobile Computing*, Vol. 23, No. 10, pp. 9224 – 9239, Oct. 2024.
 - [42] E. Samikwa, A. Di Maio, and T. Braun, "DFL:

Dynamic federated split learning in heterogeneous iot,” *IEEE Transactions on Machine Learning in Communications and Networking*, Vol. 2, pp. 733 - 752, Jun. 2024.

- [43] M. Wazzeh, M. Arafah, H. Sami, H. Ould-Slimane, C. Talhi, A. Mourad, and H. Otrok, “CRSFL: Cluster-based resource-aware split federated learning for continuous authentication,” *Journal of Network and Computer Applications*, Vol. 231, Article 103987, pp. 1 - 17, Aug. 2024.
- [44] A. Ayad, M. Renner, and A. Schmeink, “Improving the communication and computation efficiency of split learning for iot applications,” in *Proc. of the 2021 IEEE Global Communications Conference*, Madrid, Spain, pp. 1 - 6, Dec. 2021.
- [45] A. Mudvari, A. Vainio, I. Ofeidis, S. Tarkoma, and L. Tassiulas, “Adaptive compression-aware split learning and inference for enhanced network efficiency,” *ACM Transactions on Internet Technology*, Vol. 24, No. 4, pp. 1 - 26, Nov. 2024.
- [46] S. Oh, H. Nam, J. Park, P. Vepakomma, R. Raskar, M. Bennis, and S. L. Kim, “Mix2SFL: Two-way mixup for scalable, accurate, and communication-efficient split federated learning,” *IEEE Transactions on Big Data*, Vol. 10, No. 3, pp. 238 - 248, Jun. 2024.

김 종 욱 (Jong-Wouk Kim)



2019년 2월 : 강원대학교 컴퓨터과학 졸업
 2021년 2월 : 강원대학교 컴퓨터과학 석사
 2021년 3월 - 현재 : 강원대학교 컴퓨터과학 박사과정
 <관심분야> 네트워크 보안, 악성코드 분석, 딥러닝

최 미 정 (Mi-Jung Choi)



1998년 2월 : 이화여자대학교 공학사 졸업
 2000년 2월 : 포항공과대학교 공학석사
 2004년 2월 : 포항공과대학교 공학박사
 2004년 - 2005년 : 프랑스 INRIA 연구소 박사 후 연구원
 2005년 - 2006년 : 캐나다 워터루대학 박사 후 연구원
 2008년 - 현재 : 강원대학교 컴퓨터학부 컴퓨터공학 전공 교수
 <관심분야> 네트워크 관리, 네트워크 보안

정 한 길 (Han-Gil Jeong)



2026년 2월 : 강원대학교 컴퓨터과학 졸업
 2026년 2월 - 현재 : 강원대학교 데이터사이언스학과 석사 과정
 <관심분야> 분할 연합 학습, 시스템 최적화