

SNN 기반 연합학습의 이미지 분류 성능 및 에너지 효율성 분석

김 동 균*, 임 재 한°, 이 현 종*, 임 연 섭*

Analysis on Performance and Energy Consumption in Image Classification with SNN-Based Federated Learning

Dong-gyun Kim*, Jae-han Lim°, Hyun-jong Lee*, Yeon-sup Lim*

요 약

연합학습은 신경망 전역 모델을 여러 개의 디바이스에서 공동으로 학습을 하는 방식으로, 기존의 중앙 집중화 학습에 비해 데이터 보안 및 통신 비용에 있어 이점을 가지고 있기 때문에 현재 많은 연구가 이루어지고 있다. 이러한 연합학습에 대한 연구는 대부분 Artificial Neural Network(ANN)을 전역 모델로 사용하는데, ANN은 에너지 소모가 크기 때문에 모바일 디바이스 및 임베디드 디바이스에서 사용을 하기에는 무리가 있다. 대신에 에너지 효율성이 높은 Spiking Neural Network(SNN)을 전역 모델로 사용한 연합학습에 대한 연구가 이루어지고 있다. SNN은 스파이크를 통해서 뉴런들 간에 정보를 전달하는 신경망으로, event-driven 방식으로 작동하기 때문에 ANN에 비해 에너지 효율성이 높다는 이점이 있다. 하지만 ANN을 연합학습에 사용한 연구 결과에 비해 SNN을 연합학습에 사용한 연구 결과는 매우 적으며, 이는 SNN 기반 연합학습을 여러 분야에 적용하기 힘들게 한다. 본 논문에서는 실험을 통해 분석한 결과 SNN 기반의 연합학습이 ANN 기반의 연합학습에 비해 에너지 소비량이 더 적다는 결과를 얻을 수 있었다.

키워드 : 연합학습, 스파이킹 뉴럴 네트워크, 2차원 이미지 분류

Key Words : Federated Learning, Spiking Neural Network, 2-dimensional image classification

ABSTRACT

Federated learning is a cooperative learning process of global neural network models with several devices. Many researchers focus on it due to its advantages in data privacy protection and low communication costs. Most previous studies on federated learning use conventional Artificial Neural Networks (ANNs) as global model. However, it is difficult to use ANNs in mobile and embedded devices because they consume a lot of energy. To apply federate learning to mobile and embedded devices successfully, it is important to use Spiking Neural Networks (SNNs) as global model due to their high energy efficiency. This is because SNNs deliver

※ 본 연구는 한국연구재단 신소자원천기술개발사업 (grant no. NRF-2021M3F3A2A01037962)와 한국연구재단 생애초기연구사업 (grant no. NRF-2022R1G1A1006174), 그리고 (grant no. 한국연구재단 기본연구사업 (grant no. RS-2023-00241628)의 지원을 받아 수행된 연구입니다.

• First Author : Kwangwoon University Department of Software, ehdrbs4935@kw.ac.kr, 학생회원

° Corresponding Author : Kwangwoon University Department of Software, ljhar@kw.ac.kr, 정회원

* Kwangwoon University Department of Software, hetzer44@kw.ac.kr

* SungShin Women's University Department of Convergence Security Engineering, ylim@sungshin.ac.kr, 정회원

논문번호 : 202401-016-B-RN, Received January 18, 2024; Revised March 11, 2024; Accepted April 2, 2024

data between neurons with spikes and the delivery process works in an event-driven manner, which is more energy efficient than that of ANNs. However, the studies on federated learning with SNNs are much less than the studies with ANNs. This problem makes the federated learning with SNNs difficult to be utilized in various applications. In this paper, we conduct simulation with diverse situations and find out that the federated learning with SNNs is more energy efficient than the federated learning with ANNs.

1. 서 론

현재 모바일 네트워크 환경에서 각각의 디바이스 별로 분류 또는 예측을 위해 신경망을 사용한다. 이때 신경망을 학습시키기 위해서는 훈련 데이터셋이 필요한데, 대부분의 경우 모바일 디바이스는 신경망 모델을 좋은 성능을 가질 수 있도록 충분히 학습을 시킬 만큼의 데이터를 보유하고 있지 않다. 따라서 각 디바이스에서 보유하고 있는 데이터를 하나의 서버로 전송하여 서버에서 신경망을 학습하는 방식이 등장하게 되었다. 이것을 중앙집중형 학습(Centralized Learning)이라고 한다. 중앙집중형 학습을 통해서 훈련 데이터의 개수가 증가하여 신경망의 성능은 높아졌지만, 디바이스가 서버로 데이터를 전송하는 과정에서 발생할 수 있는 보안에 대한 문제가 제기되었다. 이러한 보안 문제를 해결하기

위해서 등장한 것이 연합학습(Federated Learning)이다. 연합학습은 서버로 데이터를 전송하는 대신 각 디바이스에서 로컬 데이터셋의 학습을 통해 업데이트된 모델의 파라미터(가중치, 바이어스) 값을 전송한다.

연합학습에서 주로 사용하는 신경망으로는 Artificial Neural Network(ANN)이 있다. 연합학습을 주제로한 대부분의 기존 연구들은 기본적으로 ANN을 신경망 모델로 사용하고 있다(Federated ANN)^[1,2,3]. 하지만 ANN을 사용하였을 때의 문제는 임베디드 디바이스와 같이 연산 성능이 떨어지고 낮은 전력 소비를 요구하는 작은 디바이스는 ANN이 요구하는 성능을 충족하지 못한다는 것이다. 따라서 성능이 낮은 디바이스에서도 신경망 모델을 학습하기 위해서 연산 및 에너지 효율성을 높이는 방법들이 연구되었다.

그 방법 중 하나로 ANN 대신에 Spiking Neural Network(SNN)을 사용하는 방식(Federated SNN)이 제안되었다^[4]. ANN은 실수 값으로 데이터를 전달하기 때문에 뉴런으로 정보를 전달할 때 이전 뉴런의 데이터와 뉴런 사이의 시냅스 가중치를 곱하는 연산이 발생한다. 반면에 SNN은 스파이크 형태의 이산적인 값으로 데이터를 전달하기 때문에 스파이크 정보가 1일 때 뉴런 사이의 시냅스 가중치 값을 전달하기만 하면 된다. 이때 하드웨어에서 요구하는 곱셈 연산의 에너지 소비량에 비해 덧셈 연산의 에너지 소비량은 매우 적기 때문에 에너지 소비량이 크게 줄어들게 된다. 결과적으로 SNN은 ANN에 비해 더 적은 연산과 에너지를 요구하며 이는 한정된 리소스들을 가지는 임베디드 디바이스에서 ANN 보다 SNN이 적합함을 의미한다.

현재 연합학습에 대한 대부분의 연구는 분류 또는 예측 정확도의 성능을 중요시하고 있다. 이때 분류 또는 예측 정확도의 성능은 SNN보다 ANN이 더 좋기 때문에 ANN보다 SNN을 모바일 디바이스에 적용하는 것이 에너지 소비 측면에서 더 효율적임에도 불구하고 SNN을 적용한 연합학습에 대한 연구가 많이 이루어지지 않아 기존의 연구 결과가 매우 적다. 하지만 모바일 네트워크 환경에서는 정확도 성능만큼이나 에너지 효율성도 중요하기 때문에 SNN 기반 연합학습에 대한 연구가 필요하다. 또한 실제 모바일 네트워크 환경에서

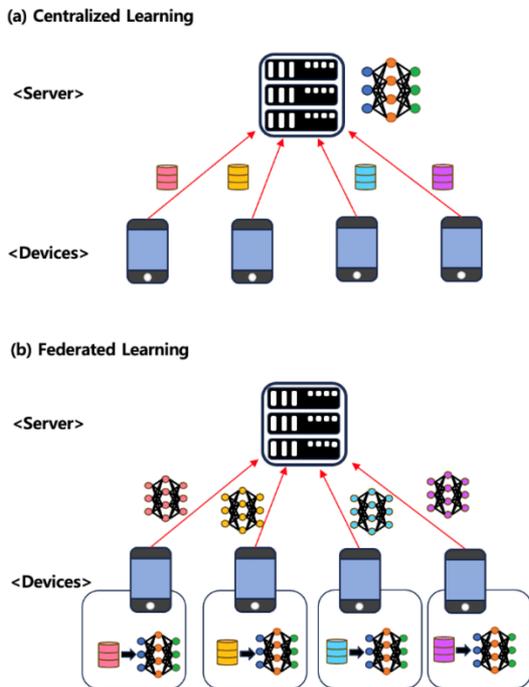


그림 1. (a) 중앙집중화 학습 과정, (b) 연합학습 과정
Fig. 1. (a) Centralized learning process, (b) Federated learning process

각 디바이스들은 서로 다른 분포의 데이터를 가지고 있을 수 있는데 이를 데이터 분포가 불균일하다고 한다. 데이터 분포의 불균일성에 따른 연합학습의 성능에 대한 분석은 기존의 많은 연구에서 다루고 있지만 SNN을 사용한 연합학습에서 데이터 불균일성의 영향을 분석한 연구는 적다⁵⁾. 따라서, 본 연구에서는 연합학습에서 ANN을 신경망으로 사용하였을 때와 SNN을 신경망으로 사용하였을 때의 분류 정확도 및 에너지 소비량에서의 성능 차이를 각각 비교하여 SNN을 사용하였을 때 발생하는 손실과 이득을 살펴보고자 하며, Federated ANN과 Federated SNN 각각에 대해서 데이터 분포의 불균일성에 따른 성능 변화 측정하여 비교 및 분석을 하고자 한다. 위의 내용을 정리하였을 때 본 연구의 의의는 다음과 같다.

- 동일한 구조의 ANN과 SNN을 각각 연합학습의 신경망으로 사용하였을 때, 이미지 분류 작업 각각에 대한 정확도를 구하고 성능을 비교하였다.
- 또한 ANN 기반의 연합 학습과 SNN 기반의 연합 학습 각각에 대해서 에너지 소비량을 측정하여 연합학습에서 SNN을 사용하였을 때의 에너지 소비량이 더 적다는 결과를 보였다.
- ANN 기반의 연합 학습과 SNN 기반의 연합 학습이 데이터 분포의 불균일성에 얼마나 저항성이 있는지 분석하였다.

본 논문의 구성의 순서는 다음과 같다. 2단락에서는 연합학습과 SNN에 대해 간략히 설명하며, 3단락에서는 연합학습의 활용 및 SNN 기반의 연합학습에 대한 기존의 연구를 소개한다. 4단락에서는 본 논문에서 SNN 기반의 연합학습을 구현한 내용을 설명하며, 5단락에서는 실험 설정 및 결과를 보여준다. 이 논문의 6단락은 결론으로 구성되어 있다.

II. 배경지식

2.1 연합학습

연합학습은 신경망 모델을 가지고 있는 다수의 디바이스들이 서로 동일한 신경망 모델을 가지고, 학습에 참여하는 모든 디바이스들이 연합하여 학습을 진행하여 파라미터를 업데이트 하는 방식을 의미한다. 연합학습의 등장 이전에 공동으로 학습을 하는 방식으로는 중앙집중형 학습(Centralized Learning)이 있다. 이 학습

방식은 다음과 같은 과정을 거쳐서 이루어진다. 먼저 각 디바이스가 보유하고 있는 데이터 원본을 서버로 전송한다. 다음으로 서버는 각 디바이스로부터 수집한 데이터들을 사용하여 전역 신경망 모델을 업데이트 한 후 업데이트된 모델을 각 디바이스에게 전송한다. 이 중앙집중형 학습 방식은 두 가지 문제점이 있다. 첫번째로 데이터의 보안 문제가 있다. 디바이스에서 서버로 데이터를 보내는 과정에서 제 3자에게 데이터가 유출될 위험이 존재한다. 이를 방지하기 위해 암호화와 복호화 기술을 사용한다 하더라도 데이터의 크기가 커짐에 따라서 요구되는 연산량도 많아지고 암호화, 복호화 키를 관리해야 한다는 문제가 있다.

두번째로 디바이스가 서버로 데이터를 보낼 때 발생하는 통신 오버헤드 문제가 있다. 디바이스는 네트워크를 통해 서버로 데이터를 보내게 되는데 이때 네트워크의 대역폭, 채널의 상태 등과 같은 여러 통신 요소의 제약들로 인해 한 번에 많은 데이터를 보내지 못한다. 때문에 데이터를 나누어서 여러 번 보내야 하는 상황이 발생하는데, 그로 인해 전체 학습에서의 시간 지연이 발생할 수 있다.

따라서 이러한 데이터 보안 및 통신 오버헤드 문제들을 해결하기 위해 등장한 공동 학습 방식이 연합학습이다. 연합학습은 다음과 같은 과정을 통해 학습이 이루어진다. 우선 서버에서 연합학습에 참여하는 디바이스가 공동으로 사용할 전역 모델을 초기화한다. 이 때 모델은 목적, 데이터셋, 목표 성능에 따라서 다양하게 구성될 수 있다. 다음으로 서버는 초기화된 모델을 각 디바이스에게 전송한다. 디바이스는 서버로부터 받은 모델을 디바이스가 보유하고 있는 데이터를 사용하여 학습을 통해 업데이트를 한다. 이렇게 모든 디바이스에서 업데이트된 모델의 파라미터(가중치, 바이어스 등)를 다시 서버로 보내고 서버에서는 수신한 파라미터들을 집계하여 전역 모델을 업데이트하게 된다. 이러한 과정을 목표로 하는 성능(분류 및 예측 정확도)에 도달할 때까지 반복한다. 이 때 디바이스에서 서버로 데이터 원본이 아닌 모델의 업데이트 값을 보내기 때문에 통신 비용이 크게 줄어들고, 데이터 유출을 방지할 수 있게 된다.

2.1.1 Federated Averaging

Federated Averaging이란, 서버가 디바이스로부터 받은 모델 업데이트를 집계할 때 사용하는 가장 기본적인 방식이다⁷⁾. 이 방식은 디바이스에서 업데이트된 파라미터 값들에 대해 평균을 내는 방식으로 수식으로 표현을 했을 때 아래의 식과 같이 표현할 수 있다.

$$w_{t+1} \leftarrow \sum_{k+1}^K \left(\frac{n_k}{n} w_t^k \right) \quad (1)$$

w 는 전역 모델의 파라미터 값, w^k 는 디바이스 k에서 업데이트된 파라미터 값, n 은 전체 디바이스에 포함된 데이터 샘플의 개수, n_k 는 디바이스 k에 포함된 데이터 샘플의 개수, t 는 현재 연합학습의 라운드 수, K 는 연합 학습에 참여하는 디바이스의 개수를 의미한다.

이러한 Federated Averaging 알고리즘을 사용하면 연합학습의 라운드 수가 증가함에 따라 모델의 파라미터 값이 수렴하고 전역 모델의 분류 또는 예측 정확도 또한 계속해서 증가하여 최종 정확도로 수렴하게 된다.

2.2 Spiking Neural Network

SNN은 동물의 뇌에서 정보를 전달하는 생물학적 과정을 모방하여 만든 인공신경망이다. 동물의 뇌에 있는 뉴런들은 서로 스파이크라고 하는 전기적 신호 및 화학적 신호를 생성하여 정보를 전달한다. 정보가 뉴런에 전달되면 뉴런의 막 전위 값이 높아지고 이 막 전위 값이 임계 전위보다 높아지면 전기적 신호 및 화학적 신호를 생성하여 그 다음 연결된 뉴런으로 전달하게 된다.

SNN은 이러한 과정을 모방하였는데, 전기적 신호인 스파이크를 0 또는 1의 이산적인 값으로 표현하여 뉴런 간에 전달하고, 값이 1인 스파이크가 들어오면 이전 뉴런과 다음 뉴런 사이의 시냅스 가중치 값을 더하여 다음 뉴런이 활성화되는 값을 높여준다. 그리고 이 활성화 값이 사전에 정의한 임계치보다 높아지면 뉴런에서 스파이크를 발화하여 그 다음 뉴런으로 전달을 하게 된다.

기존의 인공신경망은 데이터가 입력될 때마다 모든 시냅스에서 가중치의 곱 연산과 누적 연산이 이루어지는 반면, SNN은 스파이크가 발화될 때만 시냅스에서 가중치의 누적 연산만이 이루어진다. 때문에 SNN에서의 연산 비용이 크게 감소하여 에너지 효율성이 높아지게 된다. 따라서 SNN은 저전력 소모를 필요로 하는 뉴로모픽 하드웨어에서 많이 사용되고 있다^{8,9)}.

하지만 ANN은 뉴런 간에 정보를 전달할 때 데이터의 실수 값을 그대로 전달하는 반면, SNN은 실수 값의 데이터를 스파이크 형태의 이산적인 정보로 변환하고, LIF 뉴런 모델을 통해 다음 뉴런으로 스파이크를 전달하기 때문에 정보의 손실이 발생하게 된다. 따라서 SNN은 ANN에 비해 정확도 성능이 떨어진다는 문제가 있다.

2.2.1 스파이크 인코딩 방식

SNN에서는 스파이크로 정보를 전달하기 위해 입력

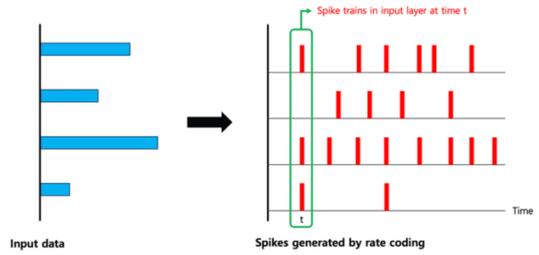


그림 2. Rate coding 방식의 스파이크 생성 과정
Fig. 2. Spike generation process of Rate coding

데이터를 스파이크로 인코딩하는 과정을 거치게 된다. 이때 자주 사용되는 인코딩 방식으로는 Rate coding 방식이 있다.¹⁰⁾

Rate coding 방식은 입력 데이터 값의 크기에 비례하여 생성되는 스파이크의 개수를 조절하는 방식으로, 값의 크기가 클수록 생성되는 스파이크의 개수가 더 많아진다. 그림 2에서 보이는 것 처럼 각 데이터 값 별로 사전에 정의한 스파이크 인코딩 단위 시간 내에 발생하는 스파이크의 개수를 구하고, 각 시간별로 생성된 스파이크 트레인을 입력 계층에 전달한다.

2.2.2 Leaky Integrate and Fire(LIF) 모델

LIF 모델은 동물의 뇌에서 뉴런의 활성화를 위한 막 전위 값이 시간이 지남에 따라서 감소하는 뉴런의 특성을 SNN에 적용하기 위해서 만든 모델이다¹¹⁾. LIF 모델을 SNN에 적용하면 그림 3에서와 같이 스파이크가 들어올 경우 뉴런의 막 전위 값이 시냅스 가중치의 크기 만큼 증가하였다가 스파이크가 들어오지 않으면 시간이 지남에 따라 감소한다. 때문에 뉴런이 활성화되기 위해서는 해당 뉴런과 연결된 이전 뉴런에서 스파이크를 연속적으로 생성하거나, 연결된 이전 뉴런 사이의 시냅스 가중치의 값이 커야 한다. 또한 뉴런이 활성화되어 스파이크가 출력되면 막 전위 값을 휴지 전위 값으로 떨어트려 스파이크가 들어오더라도 계속해서 활성화가 이루어지지 못하게 한다.

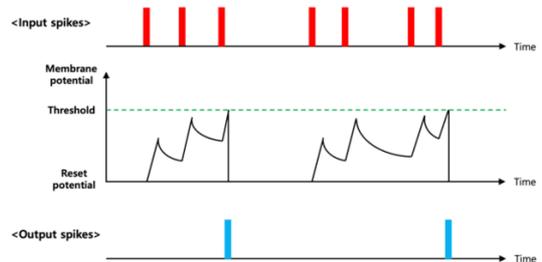


그림 3. LIF 모델의 동작 과정
Fig. 3. Process of LIF model operation

III. 기존 연구

3.1 연합학습의 활용

[1]에서는 도로 위에 있는 차량의 정확한 위치를 알아내기 위해 차량 및 위성 간의 통신을 할 때 각 차량의 위치 정보라는 개인적인 정보를 보호하기 위해 연합학습을 사용하였다.

[3]에서는 차량에 부착된 LIDAR, 카메라 등의 센서를 통해 수집된 정보로 도로에 있는 물체를 탐지하고 식별하기 위한 작업에서 차량 간 통신에서 발생할 수 있는 대역폭 사용량 및 통신 지연 속도 증가, 데이터 유출 문제를 방지하기 위해 연합학습을 사용하였다.

3.2 SNN 기반 연합학습의 최신 연구 성과 및 응용 사례

[12]에서는 SNN에 Batch Normalization Through Time(BNTT)라는 기술을 적용하여 연합학습의 과정을 통해 학습된 전역 모델의 테스트 정확도 성능이 ANN 보다 더 높아질 수 있음 보였다.

[13]에서는 디바이스 별 메모리 성능이 불균일한 상황에서 SNN을 신경망으로 사용하여 연합학습을 진행하는 것을 처음으로 고려하였으며, 각각의 디바이스별로 메모리 성능에 따라 다른 SNN 구조를 사용하는 방식을 제안하여 메모리 오버헤드를 줄였다.

[14]에서는 스마트 차량에서 표지판을 인식하는 작업에 SNN 기반 연합학습을 적용하였다. 표지판 인식은 차량에 부착된 센서에서 이루어지기 때문에 전력 소비를 최소화하는 것이 중요하고, 차량 간의 실시간 통신이 중요하기 때문에 에너지 소비량이 적고 빠른 추론을 특징으로 하는 SNN을 사용하였다.

IV. SNN 기반 연합학습 구현

4.1 신경망 구현

디바이스에서 사용하는 신경망 구조는 ANN과 SNN 모두 동일하며 그림 4와 같이 입력, 은닉, 출력 세 개의 계층으로 구성되어 있고 모든 계층의 뉴런이 연결되어 있는 완전 연결(Fully Connected) 구조이다. 각 계층의 뉴런 개수를 살펴보자면 입력 계층의 뉴런 개수는 784개이며 28 x 28 크기의 픽셀 값으로 이루어진 이미지 데이터에 대해 각각의 픽셀 값이 하나의 뉴런의 입력으로 들어갈 수 있도록 하였다. 은닉 계층의 뉴런 개수는 100개로 설정하였고, 출력 계층의 뉴런 개수는 10개이며 10개의 클래스를 가진 데이터셋에 대해서 각 클래스가 하나의 뉴런에 할당되도록 하였다.

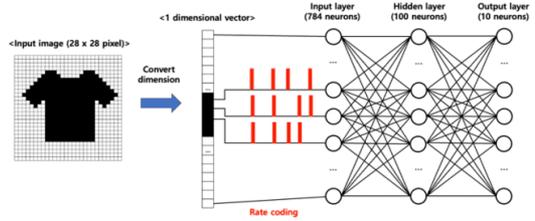


그림 4. 실험에서 사용한 신경망 구조 및 데이터 입력 과정
Fig. 4. Neural network architecture and data input process in experiments

신경망 구조를 구현함에 있어서 pytorch 모듈을 사용하였고, SNN의 특징인 LIF 구조 및 레이트 인코딩 방식을 구현할 때는 sntorch 모듈을 사용하였다.

4.2 연합학습 환경 구현

연합학습 환경을 구현함에 있어서 실제 서버와 디바이스 여러 대를 사용하는 대신에 그림 5와 같이 IDE(Integrated Development Environment)를 사용하여 데스크탑에서 가상의 서버 객체와 디바이스 객체를 생성하여 시뮬레이션을 하였다. 여기서 말하는 IDE란 소프트웨어 코드 편집, 빌드, 컴파일, 테스트 등과 같이 프로그램 개발을 위해 사용되는 여러 도구들이 통합되어 있는 소프트웨어 애플리케이션으로, 대표적으로 visual studio, eclipse, pycharm 등이 있다. 우리는 ANN 및 SNN 신경망 구현을 위해 pytorch 라이브러리 및 sntorch 모듈을 사용하였기 때문에 pycharm IDE를 사용하였다. 따라서 본 논문의 실험에서는 디바이스와 서버 간의 통신 시 발생하는 오버헤드는 고려하지 않는다. 연합학습의 각 라운드에는 모든 디바이스가 참여하며, 각 디바이스 당 보유하고 있는 로컬 훈련 데이터 샘플의 개수, 클래스의 개수 그리고 클래스 당 데이터 샘플의 개수는 모두 동일하다.

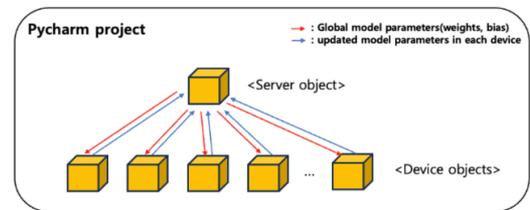


그림 5. 실험에서 사용한 연합학습 환경
Fig. 5. Federated learning environment in experiments

4.3 로컬 및 전역 모델 훈련 과정

연합학습 시작 전, 서버에서 신경망 모델을 초기화하여 모든 디바이스로 모델을 전송한다. 각 디바이스에서

는 서버로부터 전달받은 모델을 보유하고 있는 로컬 훈련 데이터 샘플을 사용하여 학습을 통해 업데이트한다. 이때 ANN 모델의 경우에는 데이터 샘플을 모델의 입력 계층으로 전달할 때 데이터 실수 값을 그대로 전달하고, SNN 모델의 경우 입력 데이터를 레이트 인코딩 방식을 사용하여 스파이크로 변환한 후 입력 계층으로 전달한다. 레이트 인코딩에서는 기존의 데이터를 이용하여 스파이크의 빈도 수를 조절하는 방식으로 정보를 표현하며 인코딩 과정에서 정보의 손실이 발생할 수 있으므로, 인코딩 과정을 여러 번 진행하여 스파이크를 여러 번 전달한다.

ANN과 SNN 모델 모두 지도 학습 방식을 사용하여 학습이 되며 업데이트 시에는 오차역전파 방식을 사용한다. 각 디바이스에서 업데이트된 매개변수(가중치, 편향)들은 서버로 전달되고, 서버에서는 모든 디바이스로부터 전달받은 매개 수들을 식(1)에 따라 계산을 하여 전역 모델의 파라미터로 업데이트를 한다.

각 연합학습의 라운드마다 위의 과정을 거쳐 업데이트된 전역 모델에서 동일한 테스트 데이터셋을 사용하여 테스트를 진행해서 정확도를 측정한다.

V. 실험

5.1 실험 설정

본 실험에서는 2차원 이미지 분류 작업을 진행하였고, 데이터셋은 MNIST 데이터셋과 함께 흔히 사용되는 Fashion-MNIST 데이터셋을 사용하였다. Fashion-MNIST 데이터셋은 10가지 종류의 옷 이미지를 가지고 있는 데이터셋이며, 60,000개의 훈련 데이터 샘플과 10,000개의 테스트 데이터 샘플로 구성되어 있다. MNIST를 사용하는 대신 Fashion-MNIST를 사용한 이유는 SNN 기반 연합학습의 실제 애플리케이션에 대한 적용 가능성을 확인하기 위해서는 단순한 숫자 형태의 이미지인 MNIST보다 좀 더 복잡한 형태의 이미지인 Fashion-MNIST를 사용하는 것이 적합하다고 생각했기 때문이다.

실험에서 사용된 디바이스의 개수는 10개이고, 각 디바이스 별로 할당된 훈련 데이터의 개수는 6,000개이다. 데이터의 불균일성을 다르게 하여 성능을 측정하기 위해 디바이스 별로 훈련 데이터셋에 포함된 클래스의 개수를 각각 10개, 7개, 4개, 3개로 설정하여 실험을 진행하였다. 연합학습은 총 100회 이루어지며, 연합학습의 라운드 당 각 디바이스에서는 로컬 모델을 훈련하기 위해 2,000개의 훈련 데이터 샘플을 사용하고, 로컬 훈련 에포크 수는 1회이다. 이때 에포크 수를 1회로 설

정한 이유는 각 디바이스 별로 훈련 데이터에 과적합이 되는 것을 방지하기 위함이다. 연합학습에 참여하는 디바이스의 가중치가 훈련 데이터에 대해서 과적합이 되어 있으면 서버에서 Federated Averaging 알고리즘을 사용하여 전역 모델을 업데이트 할 때 가중치의 형태가 학습되는 방향으로 수렴되는 것이 아니라 왜곡이 발생하게 된다. 따라서 과적합의 영향을 최소화하기 위해 에포크 수를 1회로 설정하였다. ANN과 SNN 모델에서 사용한 목적 함수는 각각 pytorch 모듈에 있는 크로스 엔트로피(cross-entropy) 함수와 snntorch 모듈에 있는 출력 스파이크 개수를 기반으로 오차를 측정하는 함수(mean squared error count loss)이고, 옵티마이저는 모두 Adam 옵티마이저를 사용하였다.

또한 디바이스 별 보유하고 있는 훈련 데이터에 포함된 클래스의 개수를 다르게 하여 데이터 분포의 불균일성을 설정하였다. 각 디바이스에 포함된 클래스의 개수가 적을수록 디바이스 별로 서로 겹치지 않는 다른 클래스를 가질 확률이 높아지므로 데이터 분포의 불균일성이 높아진다. 데이터 분포가 균일하다는 것은 각 데이터가 독립적이고 동일한 확률 분포를 가지고 있다는 의미에서 Independent and Identically Distributed(IID)하다고 하고, 불균일하다는 것은 이와 반대로 Non-IID하다고 한다.

5.2 실험 결과

먼저 디바이스의 데이터 분포가 균일한 상황(IID 정도가 크며 이는 모든 디바이스에 포함된 클래스의 개수가 10개인 상황을 의미한다.)에서 Federated ANN과 Federated SNN의 연합학습 라운드 별 테스트 정확도를 측정하였다. 측정 결과 그림 6과 같이, 데이터 분포가 IID한 상황에서는 Federated ANN과 Federated SNN에서 테스트 정확도 수렴 속도 및 성능 차이가 거의 나타나지 않음을 확인할 수 있다.

다음으로는 디바이스 데이터 분포를 다르게 했을 때 테스트 정확도를 측정하였다. 그림 7을 보면 디바이스 별 보유하고 있는 클래스의 개수가 적을수록 즉 데이터 분포의 불균일성(Non-IID의 정도)이 클수록 테스트 정확도의 수렴 속도가 느려지고 최종 수렴 정확도가 낮아지는 것을 볼 수 있다. 또한 이러한 성능 저하가 Federated ANN 보다 Federated SNN에서 더 많이 발생하는 것을 볼 수 있다. 성능 저하에 대한 차이를 명확하게 보여주기 위해 그림 5와 그림 6을 기반으로 하여 데이터 분포의 불균일성 정도에 따라 Federated ANN과 Federated SNN의 최종 수렴 정확도 값의 변화를 표로 나타내었다.

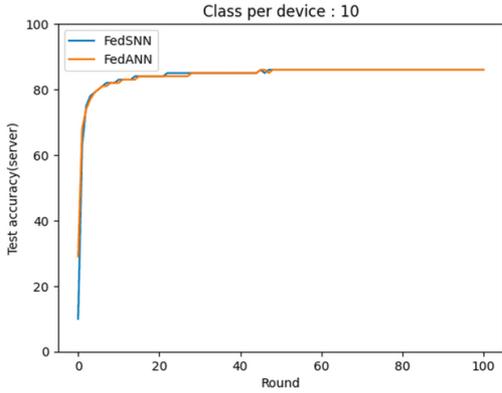


그림 6. 데이터 분포가 IID한 상황에서의 연합학습 라운드 별 테스트 정확도
 Fig. 6. Test accuracy across rounds of Federated Learning in case that data distributions of devices are IID

표 1. Federated ANN과 Federated SNN의 최종 수렴 정확도 비교
 Table 1. Final convergence accuracy comparison between Federated ANN and Federated SNN

Class per device	ANN	SNN (Difference from ANN)
10	86%	86% (0%)
7	85%	84% (1% ↓)
5	83%	80% (3% ↓)
3	82%	76% (6% ↓)

표 1의 내용을 보면 데이터 분포의 불균일성이 커짐에 따라서 Federated ANN과 Federated SNN의 최종 수렴 정확도의 값의 차이가 증가하는 것을 볼 수 있다. 이러한 현상이 나타나는 이유는 데이터 분포의 불균일성이 커질수록, 서버에서 가중치의 평균을 구했을 때 가중치의 형상이 왜곡되는 정도가 Federated ANN보다 Federated SNN에서 더 크기 때문이다.

이미지 데이터를 학습하는 신경망은 입력 계층과 은닉 계층 사이의 가중치가 학습에 사용된 이미지의 형태로 수렴이 되는 경향이 있는데¹⁵⁾, 이는 입력되는 이미지의 특징을 은닉 계층으로 잘 전달하기 위해서이다. 따라서 같은 이유로 테스트를 할 때도 정확도가 높게 나오기 위해서는 모델의 가중치의 형상이 이미지 형태와 유사하게 나타나야 한다.

그림 8과 그림 9는 각각 데이터 분포가 균일한 상황(디바이스 별 포함된 클래스 개수: 10개)과 데이터 분포가 불균일한 상황(디바이스 별 포함된 클래스 개수: 3개)에서 연합 학습에 참여한 모든 클라이언트들에 의해 업데이트된 모델의 입력 계층과 은닉 계층 사이의 가중치들을 서버에서 Federated averaging 알고리즘을 적용하여 구한 결과값을 시각화한 히트맵이다. 두 그림을 비교해보면, 데이터 분포의 불균일성이 증가함에 따라 가중치의 형상이 뚜렷한 형태의 이미지에서 흐릿한 형태의 이미지로 왜곡이 발생하는 것을 볼 수 있다. 이는 데이터 불균일성이 높아져 각 디바이스에서 서로 동일하게 보유하는 이미지가 적어지므로 디바이스 별로 가중치가 서로 다른 이미지 형태로 가중치가 수렴될 확률이 높아지기 때문이다.

또한 Federated ANN보다 Federated SNN에서 왜곡이 더 많이 일어나는 것을 볼 수 있다. SNN은 데이터를 스파이크로 변환하여 전달하는 과정에서 정보 손실이 발생하게 되는데, 그로 인해 디바이스에서 학습 시 ANN에 비해 가중치 형상이 뚜렷한 이미지 형태로 나타나지 않는다. 따라서 가중치의 평균 값을 구했을 때 왜곡이 더 심하게 발생하는 것이다.

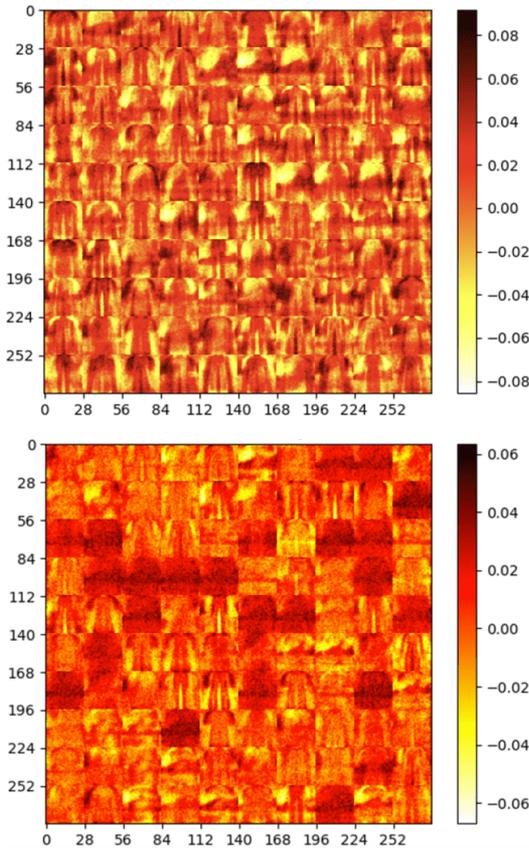


그림 7. 데이터 분포가 Non-IID한 상황에서의 연합학습 라운드 별 테스트 정확도 (디바이스 별 포함된 클래스 개수: 위(7개), 중간(4개), 아래(3개))
 Fig. 7. Test accuracy across rounds of Federated Learning in case that data distributions of devices are Non-IID (the number of class per device : top(7), center(4), bottom(3))

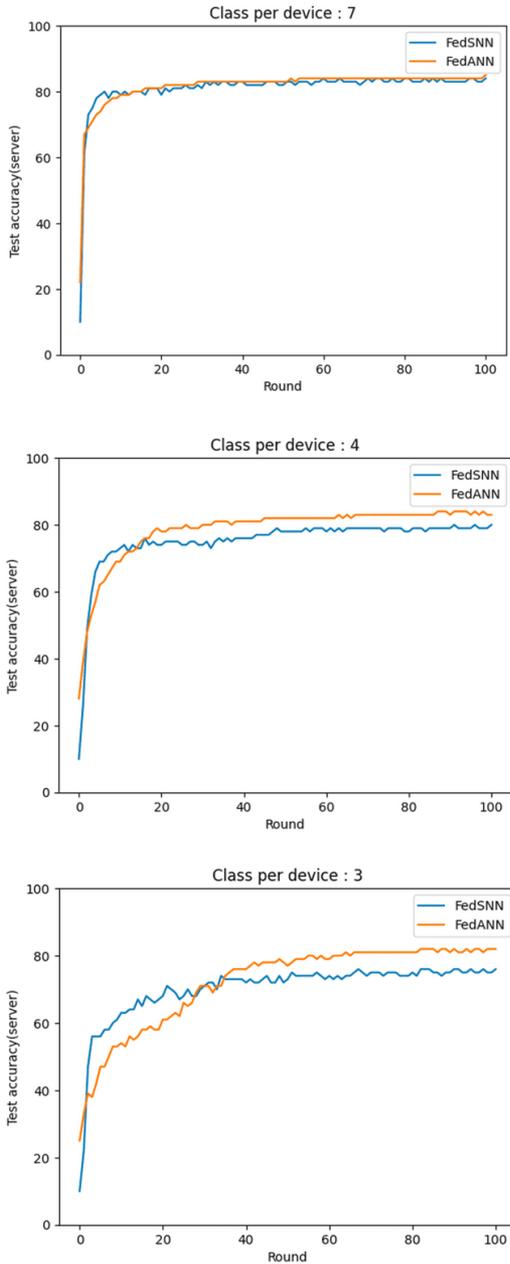


그림 8. 데이터 분포가 균일한 상황에서(디바이스에 포함된 클래스 개수: 10개), 모든 디바이스들의 입력 계층과 은닉 계층 사이의 가중치의 평균 값을 시각화한 히트맵: 위(Federated ANN), 아래(Federated SNN)
 Fig. 8. Heatmaps between input layer and hidden layer after averaging weights of all devices when data distribution is IID (the number of class per device: 10) : top(Federated ANN), bottom(Federated SNN)

위의 내용을 정리하자면, 데이터 분포의 불균일성이 증가할수록 Federated ANN보다 Federated SNN에서

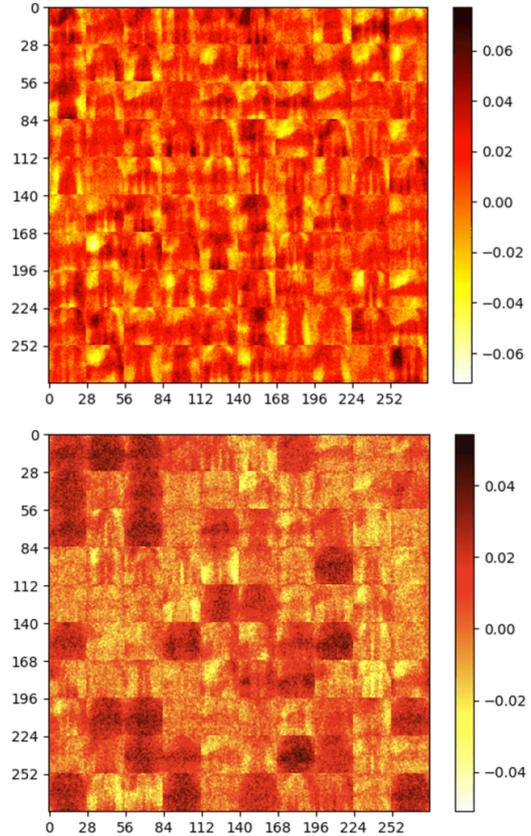


그림 9. 데이터 분포가 불균일한 상황에서(디바이스에 포함된 클래스 개수: 3개), 모든 디바이스들의 입력 계층과 은닉 계층 사이의 가중치의 평균 값을 시각화한 히트맵: 위(Federated ANN), 아래(Federated SNN)
 Fig. 9. Heatmaps between input layer and hidden layer after averaging weights of all devices when data distribution is Non-IID (the number of class per device: 3) : top(Federated ANN), bottom(Federated SNN)

입력 계층과 은닉 계층 사이의 가중치 형상의 왜곡 정도가 커지므로, Federated SNN에서 정확도가 더 많이 떨어지는 것이다.

다음으로는 Federated ANN과 Federated SNN의 에너지 소비량의 차이를 비교하였다. 비교를 위해 계산된 연산량은 연합학습에 참여하는 디바이스들의 덧셈, 곱셈 등의 연산을 총 합한 횟수를 의미한다. 연산량을 비교할 때는 한 라운드에서 발생하는 평균적인 연산량을 구하여 비교하였다. 연산량은 MAC(Multiply Accumulate Computation) 연산 방식을 기준으로 계산하였으며, 이 방식은 모델에서 뉴런 사이에 연결된 각 시냅스에서 발생하는 곱하기, 누적 연산의 횟수를 세는 방식이다¹⁶⁾. 이때 ANN의 경우 데이터 값이 뉴런에 전달이 되면 데이터는 해당 뉴런과 다음에 연결된 뉴런

사이의 시냅스 가중치가 곱해지고 곱해진 값은 다음 뉴런의 입력으로 들어갈 값에 누적이 된다.

반면 SNN의 경우 0 또는 1의 스파이크 값으로 전달이 되기 때문에 가중치와의 곱셈 연산을 수행할 필요없이 값이 전달될 때마다 다음 뉴런의 입력 값에 시냅스의 가중치 값을 누적하는 연산만 하면 된다. 이때 하드웨어에서 32비트 정수로 곱셈과 누적, MAC, AC 연산을 수행하는데 소비되는 에너지량의 추정치는 표 2에 기술된 값과 같다.

모델의 각 계층의 시냅스에서 발생하는 연산(ANN에서는 MAC연산, SNN에서는 AC 연산)의 횟수를 계산했을 때, 각 모델 별 연산 횟수는 다음과 같이 구해질 수 있다. 입력 계층의 뉴런 수를 I, 은닉 계층의 뉴런 수를 H, 출력 계층의 뉴런 수를 O라고 하면, ANN에서의 데이터 샘플 한 개 학습 시 발생하는 연산 횟수는 아래의 식 (2)와 같다.

$$(I \times H) + (H \times O) \quad (2)$$

SNN에서는 입력 데이터를 스파이크로 변환하여 여러 번 전달을 하고, 뉴런에서 스파이크가 발화할 때만 연산이 이루어지기 때문에 SNN에서 연산 횟수를 구하는 방식은 ANN과는 차이가 있다. 데이터를 스파이크로 변환하여 전달하는 횟수를 T라고 하고, 각 time step에서의 평균적인 스파이크의 발화율을 R이라고 했을 때 R1은 입력 계층에서 은닉 계층으로 전달되는 스파이크의 평균 개수를 의미하고, R2는 은닉 계층에서 출력 계층으로 전달되는 스파이크의 평균 개수를 의미한다. 이때 SNN에서의 연산 횟수는 다음 식 (3)과 같다.

$$(I \times H \times R_1 \times T) + (H \times O \times R_2 \times T) \quad (3)$$

이후 각 모델 별 에너지 소비량은 위에서 구한 각각의 연산 횟수에 E_{MAC} 와 E_{AC} 값을 곱해줌으로써 계산될 수 있다.

표 2. 곱셈 및 누적 연산에 대한 에너지 소비량
Table 2. Energy consumption of MAC operation

Operation	Estimated Energy (pJ)
32-bit Multiply (E_{Mult})	3.1
32-bit Add (E_{Add})	0.1
32-bit Multiply and Accumulate ($E_{MAC} = E_{Mult} + E_{Add}$)	3.2
32-bit Accumulate (E_{AC})	0.1

표 3. Federated ANN과 Federated SNN의 에너지 소비량 비교
Table 3. Energy consumption comparison between Federated ANN and Federated SNN

Class per device	ANN (μJ)	SNN (μJ) (Percent of ANN)
10	4922.8	2340.8 (47.55%)
7		2341.1 (47.55%)
5		2345.0 (47.63%)
3		2338.0 (47.49%)

따라서 Federated ANN과 Federated SNN에서 연합 학습의 한 라운드에 발생하는 에너지 소비량을 디바이스의 데이터 분포를 다르게 하여 구한 결과는 표 3과 같다.

표 3의 결과를 보면 데이터 분포의 불균일성에 상관없이 SNN의 에너지 소비량이 ANN의 에너지 소비량보다 50% 이상 감소한 것을 확인할 수 있다.

또한 SNN과 달리 ANN에서의 에너지 소비량이 디바이스에 포함된 클래스의 개수와 상관없이 모두 같은 이유는 ANN은 연산량을 구할 때 식 (2)에서처럼 고정된 입력, 은닉, 출력층의 뉴런의 개수만을 사용하기 때문이다.

VI. 결 론

본 논문에서는 2차원 데이터 분류 작업에서의 Federated ANN과 Federated SNN의 테스트 정확도 및 에너지 소비량을 실험을 통해 측정하고 비교하였다. 실험 결과 SNN의 모델 특성상 데이터 분포의 불균일성이 커짐에 따라서 ANN에 비해 테스트 정확도가 더 많이 감소되는 것을 볼 수 있었다. 하지만 에너지 소비량 측면에서는 데이터 분포의 불균일성에 관계없이 Federated SNN이 Federated ANN보다 더 적은 것을 볼 수 있었다.

위의 결과를 통해서 SNN 기반의 연합학습에는 데이터 분포가 불균일한 상황에서 발생할 수 있는 정확도 성능 저하 문제가 있지만, 에너지 소비량 측면에서는 확실한 강점이 있다는 것을 알 수 있다. 따라서 앞으로의 연구를 통해 SNN 기반의 연합학습에서 정확도 성능 저하 문제가 해결된다면 실제 모바일 네트워크에서 ANN 기반의 연합학습보다 사용 가능성이 더 높아질 것으로 생각된다.

이 논문의 실험에서 SNN은 ANN과의 공정한 비교를 위해 오차역전과 방식을 사용하여 가중치를 업데이트하였다. 하지만 오차역전과 방식보다 SNN에 적합한

가중치 업데이트 방식으로는 뉴런 전후 스파이크의 발화 타이밍을 기반으로 한 STDP(Spike-Timing-Dependent Plasticity)라는 방식이 있다^[7]. 따라서 SNN에 STDP 학습 방식을 적용하여 Federated SNN의 성능을 향상시킬 수 있을 것으로 예상되며 이후 연구를 통해 확인할 계획이다. 또한 실험에서 SNN의 스파이크 인코딩 방식으로 Rate coding 방식만을 사용하였다. Rate coding 외에도 Temporal coding, Burst coding, Phase coding 등 여러 스파이크 인코딩 방식을 적용하였을 때, Federated SNN의 성능을 측정하고 분석하여 Federated SNN에 적합한 스파이크 인코딩 방식에 대해서 추가로 연구를 해 볼 필요가 있다.

References

- [1] X. Kong, H. Gao, G. Shen, G. Duan, and S. K. Das, "FedVCP : A federated-learning-based cooperative positioning scheme for social internet of vehicles," *IEEE Trans. Computational Soc. Syst.*, vol. 9, no. 1, pp. 197-206, Feb. 2022. (<https://doi.org/10.1109/TCSS.2021.3062053>)
- [2] A. Li, J. Sun, B. Wang, L. Duan, S. Li, Y. Chen, and H. Li, "LotteryFL : Empower edge intelligence with personalized and communication-efficient federated learning," *IEEE/ACM SEC 2021*, pp. 68-79, San Jose, CA, USA, Dec. 2021. (<https://doi.org/10.1145/3453142.3492909>)
- [3] A. M. Elbir, B. Soner, S. Coleri, D. Gunduz, and M. Bennis, "Federated learning in vehicular networks," in *Proc. IEEE MeditCom*, pp. 72-77, Athens, Greece, Sep. 2022. (<https://doi.org/10.1109/MeditCom55741.2022.9928621>)
- [4] N. Skatchkovsky, H. R. Jang, and O. Simeone, "Federated neuromorphic learning of spiking neural networks for low-power edge intelligence," *ICASSP 2020-2020*, pp. 8524-8528, Barcelona, Spain, May 2020. (<https://doi.org/10.1109/ICASSP40776.2020.9053861>)
- [5] J. Y. Kim and W. H. Lee, "Analysis of performance of federated learning by considering local and global trainings in non-iid data environments," *J. Next-Generation Convergence Technol. Assoc.*, vol. 7, no. 1, pp. 21-31, Jan. 2023. (<https://doi.org/10.33097/JNCTA.2023.07.01.21>)
- [6] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data(2022)," Retrieved Jan. 6, 2024, from <https://arxiv.org/abs/1806.00582>. (<https://doi.org/10.48550/arXiv.1806.00582>)
- [7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. The 20th AISTATS 2017*, pp. 1273-1282, Fort Lauderdale, Florida, USA, Apr. 2017. (<https://doi.org/10.48550/arXiv.1602.05629>)
- [8] H. Chu, Y. Yan, L. Gan, H. Jia, L. Qian, Y. Huan, L. Zheng, and Z. Zou, "A neuromorphic processing system with spike-driven SNN processor for wearable ECG classification," *IEEE Trans. Biomedical Circuits and Syst.*, vol. 16, no. 4, pp. 511-523, Aug. 2022. (<https://doi.org/10.1109/TBCAS.2022.3189364>)
- [9] A. Balaji, A. Das, Y. Wu, K. Huynh, F. G. Dell'Anna, G. Indiveri, J. L. Krichmar, N. D. Dutt, S. Schaafsma, and F. Catthoor, "Mapping spiking neural networks to neuromorphic hardware," *IEEE Trans. VLSI Syst.*, vol. 28, no. 1, pp. 76-86, Jan. 2020. (<https://doi.org/10.1109/TVLSI.2019.2951493>)
- [10] W. Guo, M. E. Fouda, and A. E. A. Eltawil, "Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems," *Front Neurosci-Switz*, vol. 15, pp. 1-21, Mar. 2021. (<https://doi.org/10.3389/fnins.2021.638474>)
- [11] J. W. Han and M. Meyyappan, "Leaky integrate-and-fire biristor neuron," *IEEE Electron Device Lett.*, vol. 39, pp. 1457-1460, Sep. 2018. (<https://doi.org/10.1109/LED.2018.2856092>)
- [12] Y. Venkatesha, Y. G. Kim, L. Tassiulas, and

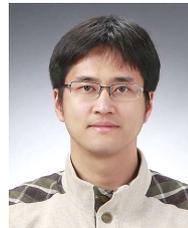
- P. Panda, "Federated learning with spiking neural networks," *IEEE. Trans. Signal Process.*, vol. 69, pp. 6183-6194, Oct. 2021. (<https://doi.org/10.1109/TSP.2021.31211632>)
- [13] S. A. Tumpa, S. Singh, M. F. F. Khan, M. T. Kandemir, V. Narayanan, and C. R. Das, "Federated learning with spiking neural networks in heterogeneous systems," *2023 IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, pp. 1-6, Foz do Iguacu, Brazil, Jun. 2023. (<https://doi.org/10.1109/ISVLSI59464.2023.10238618>)
- [14] K. Xie, Z. Zhang, B. Li, J. Kang, D. Niyato, S. Xie, and Y. Wu, "Efficient federated learning with spike neural networks for traffic sign recognition," *IEEE. Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9980-9992, Sep. 2022. (<https://doi.org/10.1109/TVT.2022.3178808>)
- [15] S. Nizampatnam, L. Zhang, R. Chandak, N. Katta, and B. Raman, "Invariant odor recognition with on-off neural ensembles," in *Proc. PNAS 2022*, vol. 119, no. 2, pp. 1-11, Jan. 2022. (<https://doi.org/10.1073/pnas.2023340118>)
- [16] P. Panda, S. A. Aketi, and K. Roy, "Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization," *Front. Neurosci.*, vol. 14, pp. 653-670, Jun. 2020. (<https://doi.org/10.3389/fnins.2020.00653>)
- [17] H. Markram, W. Gerstner, and P. J. Sjöström, "A history of spike-timing-dependent plasticity," *Front. Synaptic Neurosci.*, vol. 3, Aug. 2011. (<https://doi.org/10.3389/fnsyn.2011.00004>)

김 동 균 (Dong-gyun Kim)



2022년 2월 : 광운대학교 소프트웨어학부 공학사
 2023년 9월~현재 : 광운대학교 컴퓨터과학과 석사과정
 <관심분야> 연합학습, 기계학습, 스파이킹 뉴럴 네트워크
 [ORCID:0009-0008-5562-3981]

임 재 한 (Jae-han Lim)



2004년 2월 : 서울대학교 전기공학부 공학사
 2006년 2월 : 서울대학교 전기컴퓨터공학부 공학석사
 2014년 12월 : UCLA Electrical Engineering 공학박사

한국 전자통신연구원 선임연구원
 광운대학교 소프트웨어학부 부교수 (현)
 <관심분야> 스파이킹 뉴럴 네트워크, 차량용 네트워크, 자율주행, 모바일 컴퓨팅, 기계학습, 백스캐터 통신

이 현 중 (Hyun-jong Lee)



2024년 2월 : 광운대학교 소프트웨어학부 공학사
 <관심분야> 무선 신호 처리, 머신러닝, 뉴럴 네트워크
 [ORCID:0000-0001-8493-1002]

임 연 섭 (Yeon-sup Lim)



2007년 2월 : 서울대학교 컴퓨
터공학부 학사

2009년 2월 : 서울대학교 전기
컴퓨터공학부 석사

2017년 2월 : University of
Massachusetts Amherst,
Computer Science 박사

2017년 2월~2020년 2월 : IBM T. J. Watson
Research Center, Research Staff Member

2020년 3월~현재 : 성신여자대학교 융합보안공학과
조교수

<관심분야> 네트워크, 이동통신, 연합학습, 클라우
드 컴퓨팅