Journal of Korea Society of Industrial Information Systems Vol.28 No.4, Aug. 2023 :21-34
http://dx.doi.org/10.9723/jksiis.2023.28.4.021 ISSN:1229-3741

K-비동기식 연합학습의 동적 윈도우 조절과 모델 안정성 향상 알고리즘+

(Dynamic Window Adjustment and Model Stability Improvement Algorithm for K-Asynchronous Federated Learning)

김 효 상[1], 김 태 준[2]*
(HyoSang Kim and Taejoon Kim)

요 약 연합학습은 동기식 연합학습과 비동기식 연합학습으로 구분된다. 그 중에서 비동기식 연합학습은 동기식 연합학습 보다 시간적인 이득이 있으나 좋은 모델 성능을 얻기 위한 도전 과제가 남아있다. 특히 non-IID 학습 데이터셋에서 성능열화 방지, 적절한 클라이언트 선택 및 오래된 그래디언트 정보 관리는 모델 성능 개선에 있어 중요하다. 본 논문에서는 K-비동기식 연합학습을 다루고 있으며 non-IID 데이터셋을 통해 학습한다. 또한 기존 방식이 선택할 클라이언트 수에 있어서 정적인 K개를 사용한 것과 달리 동적으로 K 값을 조절하는 알고리즘을 제안하여 학습 시간을 줄일 수 있었다. 추가적으로, 오래된 그래디언트를 다루는 방식을 활용해 모델 성능 개선을 이루었음을 보여준다. 마지막으로 강한 모델 안정성을 얻기 위해 모델 성능을 평가하는 방식을 활용하였다. 실험 결과를 통해 전체 알고리즘을 활용했을 때 학습 시간 단축, 모델 정확도 향상, 모델 안정성 향상의 이득을 얻을 수 있음을 보여준다.

핵심주제어: K-비동기식 연합학습, 오래된 그래디언트, 낙오된 클라이언트, 모델 안정성

Abstract Federated Learning is divided into synchronous federated learning and asynchronous federated learning. Asynchronous federated learning has a time advantage over synchronous federated learning, but asynchronous federated learning still has some challenges to obtain better performance. In particular, preventing performance degradation in non-IID training datasets, selecting appropriate clients, and managing stale gradient information are important for improving model performance. In this paper, we deal with K-asynchronous federated learning by using non-IID datasets. In addition, unlike traditional method using static K, we proposed an algorithm that adaptively adjusts K and we can reduce the learning time. Additionally, the we show that model performance is improved by using stale gradient handling method. Finally, we use a method of judging model performance to obtain strong model stability. Experiment results show that overall algorithm can obtain advantages of reducing training time, improving model accuracy, and improving model stability.

Keywords: K-Asycn FL, stale gradient, stragglers, model stability

* Corresponding Author: ktjcc@chungbuk.ac.kr
+ 이 논문은 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(RS-2023-00244014)
Manuscript received April 26, 2023 / revised May 24, 2023 / accepted August 16, 2023
1) 충북대학교 정보통신공학 전공, 제1저자
2) 충북대학교 정보통신공학부, 교신저자

1. 서 론

머신 러닝 기술이 발달함에 따라 AI와 관련된 기술이 다양한 분야에서 활용되고 있다. 또한 통신기술이 발달함에 따라 다양한 전자기기에서 데이터 생성이 가능해졌고 머신러닝 기술도 적용할 수 있게 되었다. 이렇게 분산된 데이터를 활용하면 머신러닝 모델 성능을 향상시킬 수 있다. 분산된 광대한 데이터를 기존의 머신러닝 방식으로 학습하기 위해서는 데이터를 중앙 서버로 전송해야 한다. 이렇게 데이터를 직접 전송하는 것은 개인정보의 유출을 야기할 수 있고 데이터 트래픽의 증가로 이어질 수 있다. 이는 데이터 정보 보안이 중요한 분야에서는 치명적인 단점으로 작용할 수 있다. 데이터를 직접적으로 중앙 서버로 전송하는 방식이 아닌 각각의 전자기기에서 학습이 이루어 진 후 지역 모델의 정보를 중앙 서버로 전송하는 방식인 연합학습이 연구되었다(Konečný et al., 2016; McMahan et al., 2017; Ammad-Ud-Din et al., 2019; Yang et al., 2019). 연합학습 중앙 서버에서는 지역 모델 정보들을 종합하게 되며 다양한 모델 종합 방법 관련 연구가 이루어지고 있다. 그 중에서도 가장 널리 활용되는 연합 평균(Federated Average:FedAvg) (McMahan et al., 2017)은 각각의 지역 모델 파라미터를 평균 내어 하나의 중앙 모델을 만드는 방식이다. 연합학습은 동기식과 비동기식으로 구분해볼 수 있다. 동기식 연합학습은 학습 반복이 시작되면 연합학습에 참여하는 모든 클라이언트가 연합학습 중앙 서버에서 중앙 모델을 새로 받아 학습을 진행한다. 동기식 방법을 활용하면 모든 클라이언트의 지역 모델 정보가 중앙 서버에 도착할 때까지 기다려야 하기 때문에 중심 모델 생성을 지연시키는 클라이언트들이 존재할 수 있으며 이를 낙오된 클라이언트(stragglers)라 한다(Tandon et al., 2017). 반면 비동기식 연합학습은 학습 반복이 시작되면 학습 중인 클라이언트가 중앙 모델을 새로 전송받지 않는다. 이에 따라 클라이언트 학습이 시작된 시점과 중앙 학습에 참여하는 시점에 차이가 발생한다. 비동기식 방법 중에서도 최초의 K개의 지역 모델 정보만 활용하는 K-비동기식 연합학습 방법이 있다(Dutta et al., 2018; Zhou et al., 2022; Wu et al., 2022). Async SGD(Zheng et al., 2017)를 활용하면 앞서 말한 것과 같이 낙오된 클라이언트 이슈에 강점이 있으나 학습 반복 당 1개의 그래디언트만 활용하기 때문에 학습된 모델의 성능이 좋지 않을 수 있다. K-비동기식 연합학습을 활용하면 매 학습 반복에서 다양한 클라이언트의 데이터를 활용할 수 있으며 학습 시간의 이득도 얻을 수 있게 된다.

비동기식 연합학습의 도전과제는 비독립 동일분포(Non-Independent Identically Distribution: non-IID) 데이터셋을 다루는 문제와 클라이언트 선택, 그리고 오래된 그래디언트(stale gradient)를 다루는 문제로 이어진다.

서로 다른 클라이언트에서 생성된 non-IID 데이터는 모델 성능을 나쁘게 만들 수 있으며 모델 수렴이 불가능하게 만들 수도 있다. 그러기에 non-IID에 강한 모델을 만들어 내는 것이 주된 도전 과제 중 하나로 자리잡았다(Zhou et al., 2022).

효율적인 클라이언트 선택은 중심 모델이 오버피팅 되는 것을 최소화 할 수 있다. Chen et al.(2021)은 경험을 통한 탐욕적 클라이언트 선택을 하며 학습 효율성에 따라 각 클라이언트의 순위를 매겨서 선택한다. Hao et al.(2020)는 우선순위 함수를 제안하여 각 노드에 대해 우선순위 값을 계산하고 이 값이 클수록 선택될 확률을 높게 설정하는 방법을 제시하였다. 우선순위 값은 컴퓨팅 파워와 통신 지연을 기반으로 계산된다. 또한 Hu et al.(2021)는 랜덤 클라이언트 선택 방법, 영향력이 큰 업데이트를 하는 클라이언트 선택 방법, 종합에 참여하는 빈도수에 따라 클라이언트를 선택하는 방법 등을 실험을 통하여 비교하였다.

다음으로 비동기식 연합학습에서 중요한 이슈로 오래된 그래디언트가 있다. 클라이언트가 지역 모델 업데이트에 활용하는 중앙 모델의 시점과, 지역 학습된 그래디언트가 중앙 서버의 모델 종합에 반영될 때까지의 시점 차이가 존재한다. 이러한 시점 차이로 인해 많이 오래된 그래디언트(high-stale gradient)는 현재 중앙 모

델과 이질적인 것으로 취급될 수 있으며 실제로 모델 학습에 악영향을 줄 수 있다. 이러한 오래된 그래디언트를 관리하기 위한 방법으로 중앙 모델을 종합 할 때 지역 그래디언트에 가중치를 부여하는 방법이 있다. Chen et al.(2020)은 지역 모델 업데이트 후 이전과 현재의 지역 그래디언트의 차이와 관련된 값이 계산되며 이 값이 모델에 종합될 때 학습 반복 시간에 따른 가중치가 부여되어 계산하였다. Shi et al.(2020)는 중앙 모델 업데이트에 참여한 클라이언트의 오래된 정도가 클수록 지역 그래디언트에 할당된 가중치가 작게 적용되도록 하였다. Chen et al.(2020)은 최근 업데이트된 지역 모델의 가중치를 높이기 위해 지역 그래디언트의 오래된 정도를 지수함수에 적용시켜 계수로 활용한다. 연합학습에서 학습 반복마다 참여하는 클라리언트가 달라지고 현재 중앙 모델과 유사한 그래디언트 즉, 가장 최근에 업데이트 된 모델의 가중치가 더 커야 한다는 개념으로 소개되었다. 그렇다고 해서 많이 오래된 그래디언트를 무시해서는 안된다. 많이 오래된 그래디언트가 학습에 좋은 영향을 줄 수 있으므로 이를 적절히 활용해야한다. 그래디언트의 진행 방향이 현재 시점과 비슷하다면 학습에 좋게 적용될 수 있다. 이와 관련하여 코사인 유사도를 활용한 유사도를 파악하여 오래된 그래디언트를 종합하는 방법을 Zhou et al.(2022)이 제안하였으며 오래된 정도에 따른 학습률 조정 또한 다루고 있다.

추가로 도전과제로 삼은 것이 적절한 K 크기의 설정이다. K값이 증가함에 따라 낙오된 클라이언트 이슈를 고려한 학습 시간이 점점 커지게 된다. 이에 따라 적절한 K의 설정은 중요한 이슈임에도 불구하고 이에 대한 연구가 진행되지 않았다. K-비동기식 연합학습에서 학습이 진행됨에 따라 K를 동적으로 조절하여 학습 시간을 감소시켰다. 학습이 충분히 안정화되면 K를 줄이더라도 모델 성능 변화에 차이가 없다. 따라서 모델 성능은 유지하되 학습 시간을 줄일 수 있다.

K-비동기식 연합학습에서 학습에 참여하는 전체 클라이언트 수 P와 학습 반복에 참여하는 클라이언트 수 K는 그래디언트의 오래된 정도에 영향을 준다. P에 비해 K의 크기가 작을수록 모델 학습이 진행됨에 따라 클라이언트의 오래된 정도가 매우 커지게 된다. 이는 오래되지 않은 그래디언트(low-stale gradient)만 활용하고 많이 오래된 그래디언트를 무시하는 경우로 이어질 수 있다. 본 논문에서 그래디언트의 오래된 정도를 제한하는 방법인 리모델(remodel) 방법으로 다양한 클라이언트의 그래디언트를 활용하여 모델 정확도 향상과 모델 안정성 이득을 얻을 수 있다.

또한 수많은 클라이언트 그리고 각기 다른 데이터로 학습된 지역 모델 정보가 종합되기 때문에 중앙 모델은 변동(fluctuation) 될 수 있다. 중앙 모델을 강력하게 안정적으로 만들기 위해 중앙 서버에서 모델 학습 후반부 성능을 평가한다. 기준치에 미달할 경우 이전 모델을 선택할 수 있게 하며 적은 수의 모델 보류를 이용하여 강력한 모델 안정성 이득을 얻을 수 있다.

2. 시스템 모델

2.1 K-비동기식 연합학습

K-비동기식 연합학습이란 비동기식 연합학습을 기본으로 하며, 학습 반복에 참여하는 클라이언트 수가 K인 학습 방법이다. 중앙 서버가 1개 존재하고 총 P개의 클라이언트가 학습에 참여한다. 매 반복마다 먼저 도착하는 K개의 클라이언트를 활용하며, 이것들을 통해서 중앙 서버의 모델을 업데이트한다.

비동기식 연합학습에서 중앙 서버의 모델을 업데이트 하는 식은 다음과 같이 표현된다(Zhou et al., 2022).

$$\boldsymbol{w}_{j+1} = \boldsymbol{w}_j - \frac{\eta}{K}\sum_{i=1}^{K} g\left(\boldsymbol{w}_{\tau(j,i)}, \boldsymbol{\xi}_{j,i}\right), \quad (1)$$

$g\left(w_{\tau(j,i)}, \xi_{j,i}\right)$은 j번째 학습 반복에서, i번째 클라이언트를 통해 얻어진 오래된 그래디언트를 의미한다. 또한 $\tau(j,i)$는 j번째 학습 반복에서

얻어진 클라이언트 i의 오래된 정도다. 이는 지역 모델 학습 전과 후의 시점 차이를 의미한다. $\boldsymbol{w_j}$는 j번째 학습 반복에서의 중앙 모델을 의미하며 $\boldsymbol{w_{\tau(j,i)}}$는 j번째 학습 반복에서 참여한 클라이언트 i이 활용했던 시점의 중앙 모델을 말한다. $\boldsymbol{\xi_{j,i}}=\{\xi_{j,i}^{(1)},\xi_{j,i}^{(2)},\ldots,\xi_{j,i}^{(m)}\}$은 j번째 학습 반복에서 얻어진 클라이언트 i의 그래디언트 갱신에 활용했던 m개의 데이터 샘플의 집합을 의미한다. η는 모델의 학습률이다.

이러한 기본 모델을 활용하게 되면, 모든 그래디언트가 동일한 가중치로 계산되기 때문에, 오래된 그래디언트와 non-IID의 영향을 고려하지 않아 비합리적인(unreasonable) 업데이트가 된다.

따라서 오래된 그래디언트와 non-IID 데이터를 모두 고려하여 각각의 클라이언트에 대해 가중치를 계산하여 활용하는 방식을 Zhou et al.(2022)가 적응형 학습률을 가지는 가중 K-비동기식 연합학습(weighted K-asynchronous FL with adaptive learning rate:WKAFL)을 제안하였다. 우리는 WKAFL 모델 업데이트 방식 중 일부를 채택하였으며 다음과 같이 표현된다.

$$\boldsymbol{w_{j+1}}=\boldsymbol{w_j}-\eta_j\sum_{i=1}^{K}p_{j,i}g(\boldsymbol{w_{\tau(j,i)}},\boldsymbol{\xi_{j,i}}), \quad (2)$$

$p_{j,i}$는 j번째 학습 반복에서 클라이언트 i의 가중치로 $\sum_{i}^{K}p_{j,i}=1$이며 지역 그래디언트 와 추정된 중앙 그래디언트사이의 유사 한 정도를 수치화 시켜 계산된다. 또한 학습률 η_j를 학습 반복 마다 그래디언트의 오래된 정도를 기반으로 조절한다.

Fig. 1은 ($P=5$, $K=2$)의 상황에서 이 논문에서 활용할 업데이트 방식인(2)을 설명한다. 학습이 시작되면 모든 클라이언트에게 중앙 모델을 전송하며, 각각의 클라이언트는 지역 학습을 시작한다. 각 학습 반복에서 중앙 서버에 최초 전송된 K개의 클라이언트를 이용하여 중앙 모델 종합을 진행한다. 이때, 가중치 $p_{j,i}$와 학습률 η_j가 새롭게 계산되어 학습에 적용된다. 중앙 모델 종합이 완료되면, 학습에 참여한 K개의 클라이언트에게 새로운 중앙 모델 정보를 전송한다.

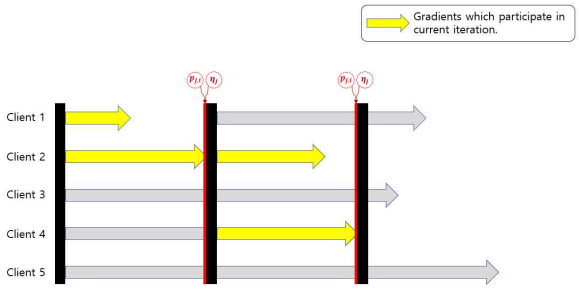


Fig. 1 WKAFL with K=2 and P=5.

2.2 모델 안정성

학습 과정에서 모델 안정성을 평가하기 위하여 지수적 가중 이동 평균(Exponentially Weighted Moving Average:EWMA)(Lucas et al., 1990)을 사용하였다. EWMA는 이전 값과 다음 값의 가중된 조합으로, 정확도에 적용하면 다음과 같이 표현된다.

$$E_{j+1}=(1-\delta_1)E_j+\delta_1 A_j\ , \quad (3)$$

(3)은 현재 모델 정확도의 예측된 값이다. 이렇게 예측된 값의 변화율 측정은 (4)와 같다.

$$D_{j+1}=(1-\delta_2)D_{j+1}+\delta_2|A_j-E_{j+1}|, \quad (4)$$

(4)는 본 논문에서 편차 정확도(DevAccuracy)라고 표현한다. (3),(4)에서 활용된 δ_1, δ_2는 0과 1 사이의 상수로, 현재 시점에 대한 가중치를 의미한다. δ가 작을수록 현재 시점에 반영되는 비중을 크게 하는 것이다. EWMA는 작은 변화를 감지하는 데 활용될 수 있으며, 학습이 진행됨에 따라 즉각적으로 대응할 수 있기 때문에 매우 유용하다. 이러한 장점을 이용하여 모델의 안정성을 평가하고 제안 알고리즘에 활용하기 위한 지표로 사용한다.

이 논문에서 활용된 기호들은 Table 1에 정리하였다.

Table 1 Notations

Notation	Description
P	Number of total clients
J_t	Total iterations number
K_j	Number of gradients needed at jth iteration
m	Mini-batch size
$l_{j,avg}$	Average loss of all clients at jth iteration
w_j	Global model at jth iteration
$\tau_{j,i}$	The staleness of the ith gradients at jth iteration
$p_{j,i}$	Weight of the ith gradients at jth iteration
η_j	Learning rate at jth iteration
CI_{all}	All clients participating in learning
CI_j	All clients participating in the jth iteration
M_{re}	Clients which will receive the global model.

3. K-비동기식 연합학습 성능 향상을 위한 알고리즘

3.1 주된 아이디어

학습 시간의 감소, 오래된 그래디언트 영향 감소, non-iid 데이터셋의 영향 감소, 및 최종 모델 정확도 증가를 목표로 리모델링과 모델 판단이 적용된 적응형 K-비동기식 연합학습(adapted K-async FL with remodeling and judgement: AKAFL-RJ)을 제안하며 이는 총 3가지 절차를 걸쳐 진행한다.

Dutta et al.(2018)의 설명과 같이 K-비동기식 연합학습에서 K에 따라 모델 수렴 시 에러와 런타임이 달라지며, 이 둘은 트레이드 오프(trade-off) 관계에 있다. 연합학습에서 학습이 느린 클라이언트 나 낙오되는 클라이언트는 런타임에 많은 영향을 끼치며, K-비동기식 연합학습에서 K의 값이 커짐에 따라 런타임이 지수적으로 급증함을 확인할 수 있다. 하지만 수렴했을 때 모델의 성능에는 장점이 있다. 이러한 에러-런타임의 트레이드 오프 관계에서, 동적으로 K를 조절하여 시간적으로도 이득을 보고, 모델 성능도 좋게 만들고자 적응형 K(adapted-K) 방법을 활용하고자 하며 이는 섹션 3.2.1에서 자세히 다루겠다.

K-비동기식 연합학습은 오래된 그래디언트를 다루는 것이 중요하다. Zhou et al.(2022)는 추정된 중앙 그래디언트를 계산하여 오래된 그래

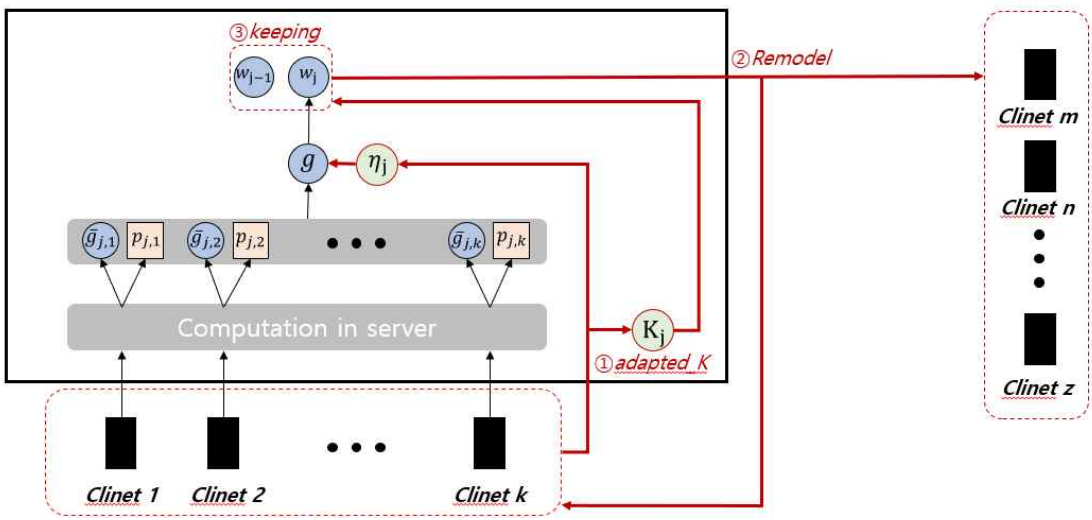


Fig. 2 AKAFL-RJ with three additional procedures.

디언트를 다룬다. j번째 학습 반복에 참여한 클라이언트의 오래된 정도에 따라 추정된 중앙 그래디언트에 반영되는 비율을 달리하며, 이렇게 찾아진 추정된 중앙 그래디언트와 각 클라이언트의 그래디언트 유사도를 측정한다. 그리고 유사도가 큰 그래디언트를 중앙 모델 종합에 크게 반영하는 방법을 활용하고 있다. 즉, 오래되지 않은 그래디언트가 현재 그래디언트와 비슷하다는 것을 기반으로 하지만, 많이 오래된 그래디언트도 반영하겠다는 생각이다. 이러한 상황에서 $\frac{K}{P}$의 비율이 작으면, 클라이언트의 참여가 후순위로 늦춰지게 되고 각각의 클라이언트가 가지는 지역 그래디언트의 오래된 정도가 커지고 크기 분포도 다양해진다.

따라서 오래되지 않은 그래디언트만 추정된 중앙 그래디언트 계산에 반영이 되면서 적절하지 못한 추정이 이루어질 수 있다. 그래디언트의 오래된 정도를 다루는 방법은 섹션 3.2.2에서 다루겠다.

모델이 수렴되면 모델 안정성를 높이는 것이 중요하다. 모델 업데이트에 적용되는 종합된 그래디언트의 크기가 작아지거나 학습률을 줄이면 모델의 변화를 줄여 모델 안정성을 높일 수 있다. 하지만 단순히 모델 변화의 크기만 줄이는 방법은 모델 성능의 향상 가능성을 제한한다는 단점이 있으며, 모델의 성능이 나빠졌을 때 회복되는 시간이 길어지는 단점도 있다. 좀더 완전한 모델 안정성을 얻기 위해서 non-IID의 영향으로부터 강한 학습이 필요하다. 중앙 모델의 일관성을 해치는 지역 그래디언트 그룹에 대한 평가를 할 것이며, 이러한 방법으로 모델 안정성 이득을 얻는 방법을 섹션 3.2.3에서 다루겠다.

3.2 AKAFL-RJ 알고리즘

AKAFL-RJ의 학습 과정을 Fig. 2에 설명하였다. 클라이언트로부터 얻은 그래디언트를 중앙 서버에서 하나의 그래디언트로 종합한다. 이를 통해 중앙 모델 $\boldsymbol{w_j}$를 만드는 일련의 과정을 거친다. 추가로 (①)에서 클라이언트로부터 받은 정보인 $l_{j,i}$를 활용해 다음 라운드의 K_j를 결정한다. (②)에서는 학습이 진행됨에 따라 클라이언트의 오래된 정도를 관찰하고 적정 수준을 넘어선 클라이언트에게 중앙 모델을 다시 전송한다. 마지막으로 (③)에서 모델 안정기에 접어들어 수렴을 기대해야 되는 상황이 되면, 현재 모델이 적절한 개선인지 확인을 하는 절차를 거친다.

자세한 학습 과정은 알고리즘 1과 같이 정리된다. 지역 학습이 완료된 클라이언트 K개를 선택하여 지역 그래디언트 정보를 얻게 된다. (8-12) 지역 그래디언트를 활용해 추정된 중앙 그래디언트를 계산하고(14-17), 지역 그래디언트와의 유사도를 계산하여 중앙 그래디언트 종합에 반영될 비율을 정한다(18-25). 이를 통해 중앙 모델 업데이트에 사용할 한 개의 그래디언트를 만들어내고(26), 학습률을 오래된 그래디언트를 기준으로 갱신한다 (27). 새로운 중앙 모델을 만들어내면(28), 이 모델이 적절한지에 대한 판단하여 진짜 모델을 결정하며(29-30), 이는 모델 학습 후반부에 모델이 안정화가 되었을 때 진행된다. 중앙 모델이 결정되면 K의 크기를 동적으로 조절하며(31), 많이 오래된 글로벌 모델을 가진 클라이언트들에게 중앙 모델을 다시 전송해 준다(32).

Algorithm 1: AKAFL-RJ	
Input: learning rate η_0 , initial number of gradients received by server K_0 , learning rate adjustment parameterb γ, remodel threshold ε_{th}, model maintain threshold ε_{dev}, weighted parameter β, DevAccuracy parameter δ_1, δ_2 , Remodel parameter A, B **Output**: optimal solution $\boldsymbol{w}^*$	
1	initialize model parameter $\boldsymbol{w_0}$ and iteration $j=1$
2	initialize the estimated gradient $\overline{\boldsymbol{g}}(\boldsymbol{w_0})=\boldsymbol{0}$
3	initialize maintain parameter $M_{st}=0$, $M=0$, $KP=False$, $acc_{est}=0$
4	broadcast $\boldsymbol{w_0}$ and j to $\boldsymbol{CI_{all}}$
5	**while** $j \rightarrow J_t$ **do**
6	$\boldsymbol{CI_j} \leftarrow empty\ list$
7	$\boldsymbol{\tau_j} \leftarrow empty\ list$

8 **for** $i=1 \rightarrow K_j$ **do**
9 receive loss value, the gradient and staleness from each client($l_i, g(w_{j,i}, \xi_{j,i}), \tau_{j,i}$)
10 CI_j.insert(CI_j)
11 τ_j.insert(τ_j)
12 **end for**
13 $l_{avg} = \frac{1}{K_j}\sum_{m=1}^{K_j} l_m$
14 $\tilde{g}(w_{j,i}, \xi_{j,i}) = g(w_{j,i}, \xi_{j,i}) + \alpha \bar{g}(w_{j-1})$
15 $a_j = \frac{e^{-\tau_j}}{2}$
16 $a = \sum_{i=1}^{K_j} a_{j,i}$
17 $\bar{g}(w_j) = \sum_{i=1}^{K_j} \frac{a_{j,i}}{a} \bar{g}(w_{j,i}, \xi_{j,i})$
18 **for** $i=1 \rightarrow K_j$ **do**
19 $s_{j,i} = \cos\langle \bar{g}(w_{j,i}, \xi_{j,i}), \bar{g}(w_j) \rangle$
20 **if** $s_{j,i} \geq s_{\min}$ then
21 $p'_{j,i} = \exp(\beta s_{j,i})$
22 **else**
23 $p'_{j,i} = 0$
24 **end for**
25 $p_j = \frac{p'_j}{\sum_{i=1}^{K} p'_{j,i}}$
26 $g(w_j) = \sum_{i=1}^{K} p_{j,i} \bar{g}(w_{j,i}, \xi_{j,i})$
27 $\eta_j \leftarrow \eta_0 \frac{1}{\tau_{\min}\gamma + 1}$
28 $w_{j+1} \leftarrow w_j - \eta_j g(w_j)$
29 $l_{avg} < \varepsilon$
30 $Judgment(\delta_1, \delta_2, w_j, w_{j+1}, j, D_j, \epsilon_{dev}, J_t, U_{st}, U, JU, acc_{est})$
31 $K_{j+1} \leftarrow Adapted_K(K_0, l_{avg}, A, B)$
32 $Remodel(\epsilon_{th}, CI_j, CI_{all}, w_j, j)$
33 $j \leftarrow j+1$
34 **end while**

3.2.1 모델 학습 속도 향상

K가 작을수록 최종 모델 에러가 커지지만 학습 시간은 줄어든다. 그러므로 학습의 초반에는 정교한 모델 수렴을 위해 상대적으로 큰 크기의 K가 필요하고 모델 학습이 진행됨에 따라 모델이 안정화되고 손실값(loss)가 작아지면 K값을 점점 작게 할 필요가 있다. 이렇게 하여 모델 수렴 직전까지의 성능을 향상시키고 안정화가 되었을 때, K를 낮추어 학습 시간의 단축 효과를 얻을 수 있는 장점이 있다.

Algorithm 2: Adapted-K

Input : K_0 , l_{avg}, A, B
Output : K

function $Adapted_K$, K_0 , l_{avg}
  If $l_{avg} > \varepsilon$ **then**
    $K = K_0$
  Else
    $K = (Ae^{l_{avg}} + B)K_0$
    K=integer(K)
  return
end function

이 방법은 현재 학습 반복에 참여한 클라이언트의 손실값의 평균을 낸 값을 사용하며 이 값이 기준 값(ε)보다 작을 때 지수적으로 감소하는 형상을 띄도록 함수를 설계하였다. 또한 최소한의 K값을 유지하는 방법을 통해 최소한의 K값의 제한을 줄 수도 있다. 이렇게 학습이 진행되었을 때 K_j의 변화 예시를 Fig. 3에 표시하였다.

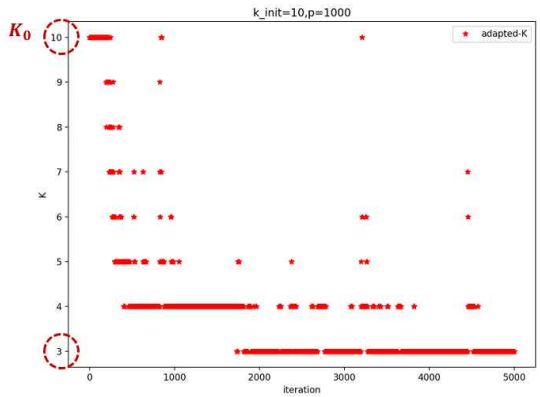
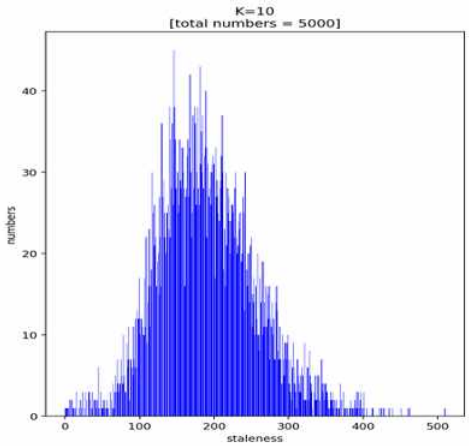


Fig. 3 K change with Iterations=5000, $K_0 = 10$, $P = 1000$.

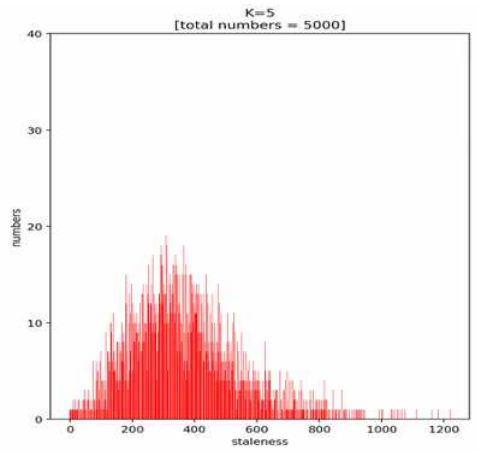
3.2.2 모델 안정성과 정확도 향상을 위한 staleness 조절

P와 K에 따라 지역 그래디언트의 오래된 정

도의 크기와 편차가 달라진다. 동일한 P에 대해 K가 작아질 경우 오래된 정도의 편차가 커지게 된다. 이는 모델 업데이트를 위한 중앙 그래디언트를 결정할 때 활용하는 추정된 중앙 그래디언트 계산 과정에서 문제가 발생한다. 그래디언트의 오래된 정도의 격차가 매우 커져 소수의 오래되지 않은 그래디언트의 값이 매우 큰 비율을 차지하면서 추정된 그래디언트가 결정된다. 이렇게 결정된 추정 값을 기반으로 추정된 중앙 그래디언트를 계산한다면 적절하지 않은 결과가 도출될 수 있다.



(a)



(b)

Fig. 4 K-async's staleness average frequency with P=2000 , total iterations=5000.

Fig. 4는 K가 작을수록 지역 그래이던트간 오래된 정도의 편차가 커짐을 보여준다. 또한 그래디언트의 오래된 정도의 크기가 전체적으로 커지기 때문에 학습이 진행됨에 따라 학습률이 매우 작은 결과를 만들어 낼 수 있다. Fig. 5은 K의 크기에 따라 학습률의 차이가 생김을 보여준다. K가 작은 경우 합습률이 매우 작은 상황이 나타나며 매우 소극적인 모델 학습이 이루어진다.

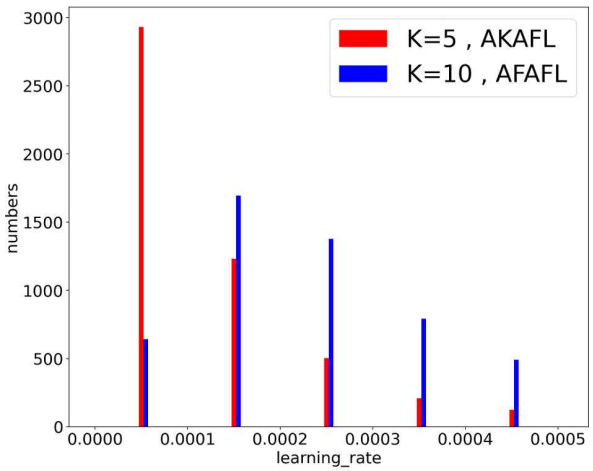


Fig. 5 The number of frequencies in the learning rate range with $K=5$, $K=10$.

이러한 그래디언트의 오래된 정도가 비대해지는 형상을 막기위해 리모델 방법을 활용한다.

Algorithm 3. Remodel

Input : ϵ_{th}, CI_j, CI_{all}, w_j, j

$M_{re} \leftarrow CI_j$

function $Remodel, \epsilon_{th}, CI_{all}, w_j, j$
  for $i = CI_{all,1st} \rightarrow CI_{all,end}$ **do**
    If $(\tau_i \rangle\ \varepsilon_{th})$ **then**
      M_{re}.insert(i)
  broadcast w_j **and** j **to** M_{re}.
end function

리모델 방법은 일정 시간 이상의 오래된 그래디언트를 가진 클라이언트 감시하고 정도를 벗어난 클라이언트($\tau_i \rangle\ \varepsilon_{th}$)에게 현재 중앙 모델 정보를 재전송해주는 방법을 의미한다. 이러한 방법을 활용함으로써 모델 정확도 향상과 모델 안정성에서 이득을 얻을 수 있다.

3.2.3 모델 판단을 통한 강한 모델 안정성

최종적으로 모델 수렴을 위해 모델 안정성에 대한 추가 보완이 필요하다. K를 조절하여 모델 학습의 후반부에 소수의 클라이언트만 활용하게 된다. 그러므로 클라이언트의 학습 데이터들이 non-IID 하면 모델의 정확도에 부정적인 영향을 미치게 된다. 따라서 좀 더 강한 모델 안정성이 요구된다.

강한 모델 안정성을 위해 섹션 2.2에서 언급한 EWMA를 사용한다. 모델 후반부에 loss 값이 매우 낮아 수렴의 근사치에 오면 모델 판단(judgement)을 위한 조건 계산이 활성화된다. 이어서 정확도 테스트를 통해 새롭게 만들어진 글로벌 모델의 성능을 평가하고 모델 정확도의 편차 정확도를 추정하며 편차 정확도는 안정성 지표로 활용한다.

Algorithm 4. Judgement

Input : $\gamma, \delta, \boldsymbol{w_j}, \boldsymbol{w_{j+1}}, j, D_j, \epsilon_{dev}$, $J_t, U_{st}, U, JU, acc_{est}$

$acc_j \leftarrow test(w_{j+1},\ D_{test})$
$acc_{est} = (1-\alpha)acc_{est} + (\alpha)acc_j$
$dev\ = (1-\beta)dev + (\beta)\left|acc_j - acc_{est}\right|$

if $dev < \varepsilon_{dev}$ **then**
  if $JU \equiv False$
    $U_{st} = j$
    $U = \dfrac{M_{st}}{J_t}$
    $JU = True$
  if $JU \equiv True$
    $M_{rate} = e^{-\frac{1}{JU_{st}}(J_t - U_{st})}$
    if $(J_{rate} > J)$ **then**
      $J_{rate} = J$
    if $(acc_{est} + J_{rate} < acc_j)$ **then**
      $\boldsymbol{w_{j+1}} \leftarrow \boldsymbol{w_j}$

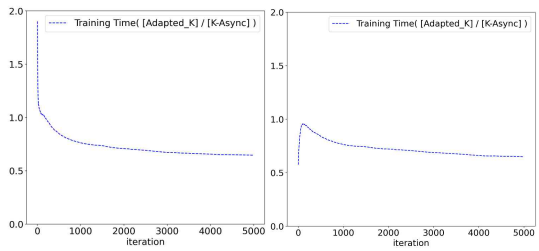
편차 정확도의 값이 기준치보다 낮으면 ($dev < \varepsilon_{dev}$) 모델 판단이 활성화된다. 시작지점에 대한 지표를 변수로 관리하며 이는 얼마나 엄격하게 안정성을 관리할 것인가로 직결된다. 시작되는 학습 반복 시점 j가 M_{st}이되며 시작점을 기준으로 학습이 진행됨에 따라 M_{rate}가 작아져 엄격한 판단이 이루어진다. 이러한 방법을 활용하여 모델 학습이 마무리되는 과정에서 적은 횟수의 모델 유지를 통해 강한 모델 안정성를 얻을 수 있음을 섹션 4.3에서 보여준다.

4. 실험 결과 및 분석

이 실험에서 간단한 합성곱 신경망(Convolution Neural Network:CNN) 모델을 활용한다. 2개의 5x5 합성곱 신경망를 활용하며 첫번째는 32 채널 두번째는 64채널을 가진다. 이 때 ReLU 활성화 함수와 2x2 최대 풀링을 활용한다. 이후 전결합층에서 ReLU 활성화 함수를 사용한다. 최종적으로 소프트맥스(softmax) 출력 계층을 가진다.

또한 이 실험에서 데이터세트는 EMNIST 데이터셋(Cohen et al, 2017)을 사용하였으며 non-IID한 경우로 활용하였다. 0~9까지의 라벨을 가지는 데이터 70,000개로 이루어진 데이터세트에서 각 클라이언트는 랜덤한 숫자 종류와 랜덤한 숫자의 양을 가지도록 분배하였다. 이때 각각의 클라이언트가 가지는 데이터는 중복될 수 있으며 70,000개 중 선택이 되지 않은 클라이언트가 있을 수도 있다.

4.1 적응형 K 비동기식 연합학습 (AKAFL)



(a) (b)

Fig. 6 Training with the stragglers effect applied with $K, K_0 = 10$, $P = 2000$.

(a) When adapted_K has a larger delay at the beginning of training.

(b) When adapted_K has a smaller delay at the beginning of training.

Fig. 6은 낙오된 클라이언트 가 있는 학습 환경에서 반복 당 시간을 측정해본 실험 결과이다. K와 K_j가 동일할 때는 지연의 정도와 상관없이 시간이 지날수록 1.0 으로 비율이 같아지는 것을 확인할 수 있다.(200 iteration 이전) 하지만 모델의 학습이 진행됨에 따라 손실값이 줄어들고 적응형-K가 동작하게 되면 점차 학습 시간이 줄어드는 이득을 얻을 수 있다. Fig. 7에서는 학습 반복을 기준으로 적응형-K의 모델 정확도를 확인할 수 있다. 적응형-K의 활용은 K-비동기식 방법을 사용한 것과 비슷하거나 그 이상의 성능을 관측할 수 있다.

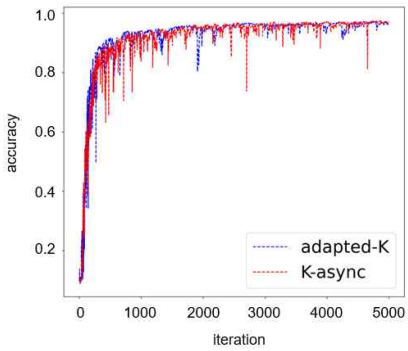


Fig. 7 K-Async and Adapted-K accuracy

4.2 리모델 방식 적용(AKAFL-R)

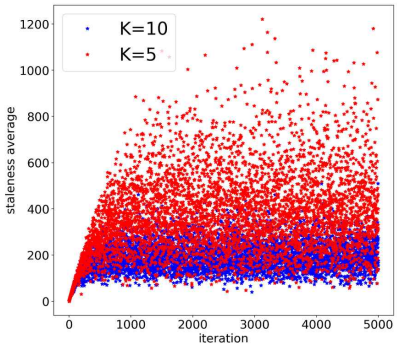
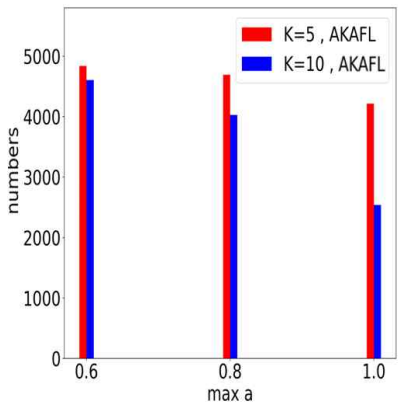
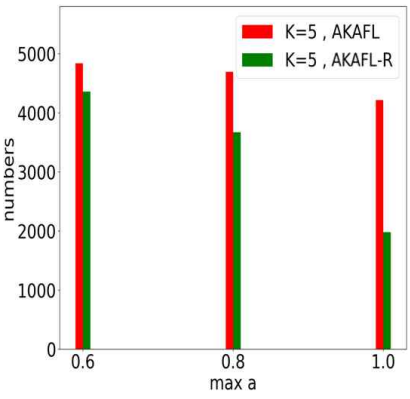


Fig. 8 Average staleness per iteration with K=5, K=10

$\frac{K}{P}$의 비율이 작아지면 (즉, 동일한 P에 대해 K가 작아지게 되면) 학습이 진행됨에 따라 후순위로 밀려나게 되는 클라이언트가 늘어난다. 섹션 3.2.2에서 언급했듯이 동일한 학습 반복에 참여하는 오래된 클라이언트의 편차가 커지고 Fig. 8에서 볼 수 있는 것처럼 그래디언트의 오래된 정도의 평균 크기도 매우 커지게 된다. 이는 중앙 그래디언트를 결정하기 위해 활용하는 추정된 중앙 그래디언트가 소수의 지역 그래디언트에 의해 결정될 수 있다는 것을 의미한다. Fig. 9 (a) 에서 K가 작을 때 단일 그래디언트(1.0에 해당)가 지배적으로 동작하는 비율이 80%에 다다른다. 이를 해결하기 위해 리모델 방법을 적용한 결과가 Fig. 9 (b)의 초록색 바이고 단순히 K의 개수를 올리는 것보다 그래디언트의 오래된 정도의 크기를 관리하는 것이 더 좋게 작용한다.



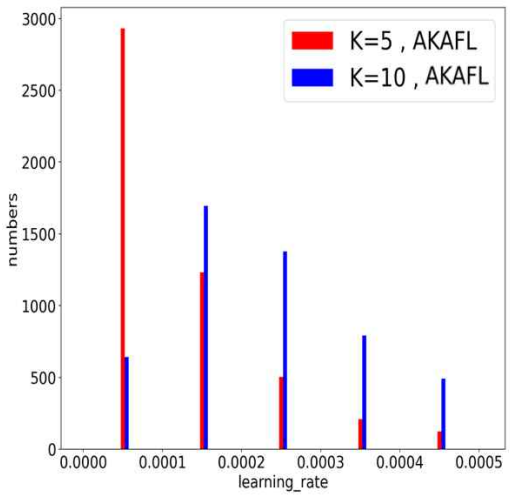
(a) K=5, K=10



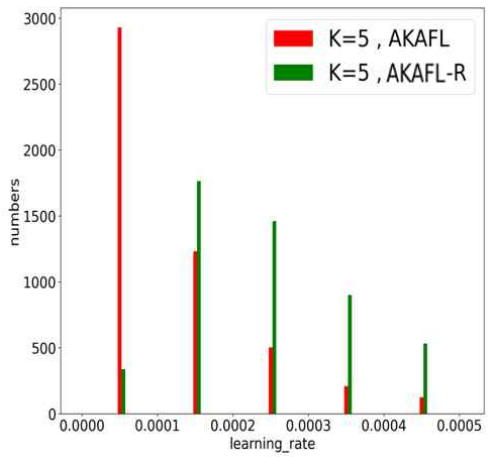
(b) K=5 , K=5 & remodel

Fig. 9 The number for which the gradient that accounts for the largest ratio when estimating the estimated gradient is greater than 0.6, 0.8, 1.0 with total iterations=5000.

또한 그래디언트가 많이 오래되면 학습률이 작아지는 반비례 관계에 있기 때문에 학습이 진행되는 동안 학습률이 전반적으로 작아지는 것을 확인할 수 있다. Fig. 10 (a)에서 K=5 일 때 학습률이 0에서 0.0001 사이를 차지하는 비중이 매우 크다는 것을 확인할 수 있다. 이는 소극적인 모델 학습으로 이어질 수 있다는 것을 의미한다. 이를 해결하기 위해 리모델 방법 적용을 하면 Fig. 10(b)에서 볼 수 있는 것과 같이 η_j =0.0001 미만의 경우가 절반 정도 줄어들고 전체적으로 적극적인 모델 학습을 유지할 수 있다.



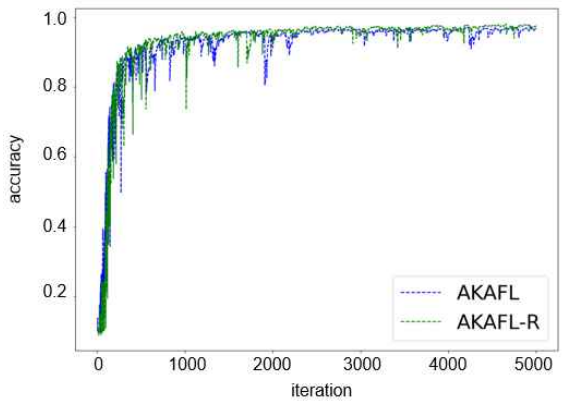
(a) K=5, K=10



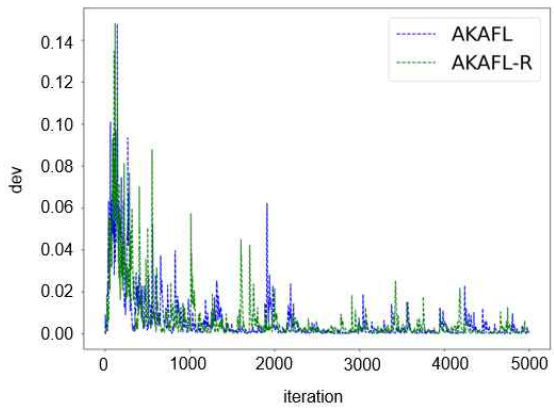
(b) K=5 , K=5 & remodel

Fig. 10 The frequency of the learning rate with $\eta_0 = 0.0005$, total iterations=5000.

AKAFL은 모델 학습 후반부에 K가 작기 때문에 모델 성능의 한계점을 확인할 수 있다. Fig. 11과 TABLE 2를 확인해보면 후반부 모델 정확도가 0.96 초중반에 머물러 있다. Fig. 10에서 확인한 바와 같이 $\frac{K}{P}$가 작아질수록 많이 오래된 그래디언트의 수가 많아져 학습이 진행되는 동안 학습률이 작아진다. 이에 따라 학습이 소극적으로 이루어지며 비동기식 연합학습(AFL 학습) 동안 중앙 모델 성능이 나빠졌을 때 회복하는 시간이 오래 걸리는 것을 1500번째 반복 근처, 2000 번째 반복 근처, 4000번째 반복 이후



(a)



(b)

Fig 11. (a) Accuracy, (b) DevAccuracy. ($common : P = 2000.$, $AKAFL : K = 10,$ $AKAFL-R : K_0 = 10$, $\epsilon_{th} = 100.$) Accuracy and DevAccuracy with the same P and initial K

등에서 찾아볼 수 있다. 반면 AKAFL-R의 경우 학습률의 크기가 매우 작아지는 빈도가 적기 때문에 학습간 나빠진 모델 성능을 회복하는 시간이 짧다. 또한 Fig. 9 에서 확인할 수 있는 것과 같이 그래디언트의 오래된 정도의 조절을 통해 여러 지역 그래디언트를 활용한 추정된 중앙 그래디언트를 만들어 냄으로써 모델 정확도 측면에서도 이득이 있음을 확인할 수 있다.

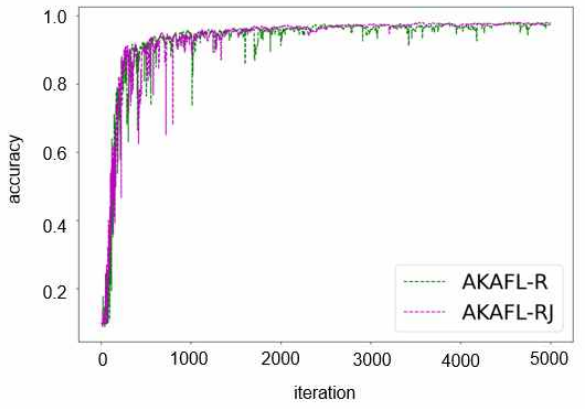
AKAFL의 모델 안정성은 학습률의 극소화의 영향으로 AFKAFL-R와 비슷한 것으로 보일 수 있다. 하지만 AKAFL은 낮은 학습률을 가지기 때문에 모델 갱신 시 작은 변동을 가져 낮은 편차 정확도 크기를 갖는다. 반면 AKAFL-R은 큰 학습률을 가지고 있어 학습 중에 성능이 크게 하락하거나 빠르게 회복되는 상황에서 큰 편차 정확도가 측정된다. 이 두 가지를 고려한다면 AKAFL의 모델 안정성이 AKAFL-R과 비슷하다고 보긴 힘들다.

Table 2 Prediction accuracy in the late model training with remodel algorithm.

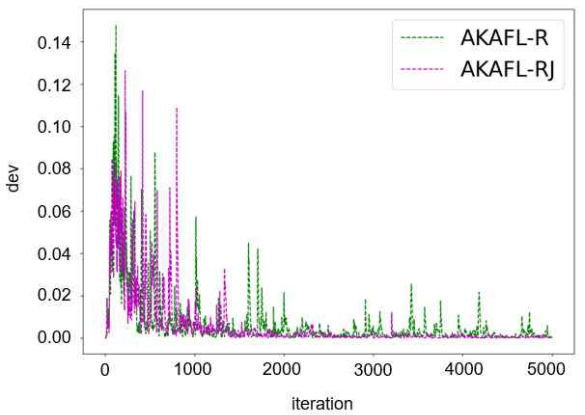
Accuracy average	last 1000 iterations	last 500 iterations
AKAFL	0.9624	0.9660
AKAFL-R	0.9716	0.9736
DevAccuracy average	last 1000 iterations	last 500 iterations
AKAFL	0.0020	0.0013
AKAFL-R	0.0019	0.0016

4.3 모델 판단 적용(AKAFL-RJ)

최종적으로 모델 정확도 향상과 모델 안정성을 위해 모델 판단 방법을 적용한 결과를 Fig. 12에서 확인할 수 있다. 모델 판단을 하기 전에는 모델 수렴 부분에서 크고 작은 변화가 존재한다. 이러한 상황이 누적되면 4000번째 반복 이후에서 보이는 것처럼 최대 1~2%까지 하락되는 현상을 확인할 수 있다. 반면 모델 유지 방법을 적용한 결과를 보면 2000번째 반복 이후부터 모델이 안정적으로 접어들며 기존에 문제 되던 모델 성능 하락 현상을 제거할 수 있다. 이에 따라 모델이 매우 안정된 상태가 유지된다. Fig. 12 실험에서 모델 유지가 되는 횟수는 138회로 전체 모델 반복 수에 비해 작은 비율을 차지하며 이러한 약간의 조정으로도 전체적인 모델 성능을 높일 수 있다는 것을 보여준다. Table 3을 살펴보면 최종 1000회 반복 및 500회 반복을 기준으로 측정한 모델 정확도 평균치도 약 0.85%p 이상 상승하였으며 특히 안정성 평가인 편차 정확도에서 -80% 가까이 되는 수치가 감소하는 이득을 얻을 수 있었다.



(a)



(b)

Fig. 12 (a) Accuracy, (b) DevAccuracy.
$common: K_0 = 10$, $\epsilon_{th} = 100$, $P = 2000$.
$AKAFL-RJ$: $M_{rate} = 0.001$
Accuracy and DevAccuracy with the same P and initial K

Table 3 Prediction accuracy in the late model training with judgement algorithm.

Accuracy average	last 1000 iterations	last 500 iterations
AKAFL-R	0.9716	0.9736
AKAFL-RJ	0.9765	0.9778
DevAccuracy average	last 1000 iterations	last 500 iterations
AKAFL-R	0.0019	0.0016
AKAFL-RJ	0.0006	0.0006

5. 결 론

이 논문에서 우리는 3가지 절차를 걸쳐 K-비동기식 연합학습 환경에서 최종적으로 모델 학습 시간 감소, 모델 정확도 향상, 모델 안정성 향상을 보여주었다. 이 과정에서 모델 학습이 진행되는 동안 K조절, 그래디언트의 오래된 정도의 관리가 이루어 졌다. 특히 모델 안정성을 평가할 때 편차 정확도를 활용하여 중앙 서버에서 관리하였고 이를 기준으로 삼아 직전 모델과 새로운 모델 사이의 선택이 이루어지게 하였다. 잘못된 방향으로 쉬지 않고 나아가는 것보다는 가끔씩 좋지 않은 모델을 과감히 버림으로써 모델 성능 향상을 할 수 있다는 것을 보여준다.

References

Ammad-Ud-Din, M., Ivannikova, E., Khan, S. A., Oyomno, W., Fu, Q., Tan, K. E., & Flanagan, A. (2019). Federated collaborative filtering for privacy-preserving personalized recommendation system. arXiv preprint arXiv:1901.09888.

Chen, Y., Ning, Y., Slawski, M., & Rangwala, H. (2020). Asynchronous Online Federated Learning for Edge Devices with Non-IID Data. In International Conference on Big Data. https://doi.org/10.1109/bigdata50022.2020.9378161.

Chen, Y., Sun, X., & Jin, Y. (2020). Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation. IEEE Transactions on Neural Networks and Learning Systems, 31(10), 4229 - 4238. https://doi.org/10.1109/tnnls.2019.2953131.

Chen, Z., Liao, W., Hua, K., Lu, C., & Yu, W. (2021). Towards asynchronous federated learning for heterogeneous edge-powered internet of things. Digital Communications and Networks, 7(3), 317 - 326. https://doi.org/10.1016/j.dcan.2021.04.001.

Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017, May). EMNIST: Extending MNIST to handwritten letters. In 2017 international joint conference on neural networks (IJCNN) (pp. 2921-2926). IEEE.

Dutta, S., Joshi, G., Ghosh, S., Dube, P., & Nagpurkar, P. (2018). Slow and Stale Gradients Can Win the Race: Error-Runtime Trade-offs in Distributed SGD. In International Conference on Artificial Intelligence and Statistics (pp. 803 - 812). http://proceedings.mlr.press/v84/dutta18a/dutta18a.pdf.

Hao, J., Zhao, Y., & Zhang, J. (2020). Time Efficient Federated Learning with Semi-asynchronous Communication. In International Conference on Parallel and Distributed Systems. https://doi.org/10.1109/icpads51040.2020.00030.

Hu, C., Chen, Z., & Larsson, E. G. (2021). Device Scheduling and Update Aggregation Policies for Asynchronous Federated Learning. In International Workshop on Signal Processing Advances in Wireless Communications. https://doi.org/10.1109/spawc51858.2021.9593194.

Konečný, J., McMahan, H. B., Yu, F. X.,

Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.

Lucas, J. M., & Saccucci, M. S. (1990). Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements. Technometrics, 32(1), 1 - 12. https://doi.org/10.1080/00401706.1990.10484583.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. a. Y. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In International Conference on Artificial Intelligence and Statistics (pp. 1273 - 1282). http://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf.

Shi, G., Li, L., Wang, J., Chen, W., Ye, K., & Xu, C. (2020). HySync: Hybrid Federated Learning with Effective Synchronization. In High Performance Computing and Communications. https://doi.org/10.1109/hpcc-smartcity-dss50907.2020.00080.

Tandon, R., Lei, Q., Dimakis, A. G., & Karampatziakis, N. (2017, July). Gradient coding: Avoiding stragglers in distributed learning. In International Conference on Machine Learning (pp. 3368-3376). PMLR.

Wang, Z., Zhang, Z., Tian, Y., Yang, Q., Shan, H., Wang, W., & Quek, T. Q. (2022). Asynchronous federated learning over wireless communication networks. IEEE Transactions on Wireless Communications, 21(9), 6961-6978.

Wu, X., & Wang, C. L. (2022, July). KAFL: Achieving High Training Efficiency for Fast-K Asynchronous Federated Learning. In 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS) (pp. 873-883). IEEE.

Xie, C., Koyejo, S., & Gupta, I. (2019). Asynchronous federated optimization. arXiv preprint arXiv:1903.03934.

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated Machine Learning. ACM Transactions on Intelligent Systems and Technology, 10(2), 1 - 19. https://doi.org/10.1145/3298981.

Zheng, S., Meng, Q., Wang, T., Chen, W., Yu, N., Ma, Z. M., & Liu, T. Y. (2017, July). Asynchronous stochastic gradient descent with delay compensation. In International Conference on Machine Learning (pp. 4120-4129). PMLR.

Zhou, Z., Li, Y., Ren, X., & Yang, S. (2022). Towards Efficient and Stable K-Asynchronous Federated Learning with Unbounded Stale Gradients on Non-IID Data. IEEE Transactions on Parallel and Distributed Systems, 33(12), 3291 - 3305. https://doi.org/10.1109/tpds.2022.3150579.

김 효 상 (HyoSang Kim)

- 충북대학교 정보통신공학부 공학사
- (현재) 충북대학교 전자정보대학 정보통신공학부 석사과정 재학
- 관심분야: 연합학습, 머신러닝, 컴퓨터 비전

김 태 준 (Taejoon Kim)

- 연세대학교 전자공학과 공학사
- 한국과학기술원 전기.전자공학과 공학박사
- (현재) 충북대학교 전자정보대학 정보통신공학부 교수
- 관심분야: 연합학습, 머신러닝, 네트워크 최적화