CUAI CV 스터디 4팀

2024.11.26

발표자 : 나상현

스터디원 소개 및 만남 인증

스터디원 1 : 김대현

스터디원 2 : 김태환

스터디원 3 : 나상현

스터디원 4 : 박서윤

발표 내용

Communication-Efficient Learning of Deep Networks from Decentralized Data

H. Brendan McMahan **Elder Moore** **Daniel Ramage** **Seth Hampson** **Blaise Agüera y Arcas**
Google, Inc., 651 N 34th St., Seattle, WA 98103 USA

Abstract

Modern mobile devices have access to a wealth of data suitable for learning models, which in turn can greatly improve the user experience on the device. For example, language models can improve speech recognition and text entry, and image models can automatically select good photos. However, this rich data is often privacy sensitive, large in quantity, or both, which may preclude logging to the data center and training there using conventional approaches. We advocate an alternative that leaves the training data distributed on the mobile devices, and learns a shared model by aggregating locally-computed updates. We term this decentralized approach *Federated Learning*.

We present a practical method for the federated learning of deep networks based on iterative model averaging, and conduct an extensive empirical evaluation, considering five different model architectures and four datasets. These experiments demonstrate the approach is robust to the unbalanced and non-IID data distributions that are a defining characteristic of this setting. Communication costs are the principal constraint, and we show a reduction in required communication rounds by 10–100× as compared to synchronized stochastic gradient descent.

1 Introduction

Increasingly, phones and tablets are the primary computing devices for many people [30, 2]. The powerful sensors on these devices (including cameras, microphones, and GPS), combined with the fact they are frequently carried, means they have access to an unprecedented amount of data, much of it private in nature. Models learned on such data hold the promise of greatly improving usability by powering more intelligent applications, but the sensitive nature of the data means there are risks and responsibilities to storing it in a centralized location.

We investigate a learning technique that allows users to collectively reap the benefits of shared models trained from this rich data, without the need to centrally store it. We term our approach *Federated Learning*, since the learning task is solved by a loose federation of participating devices (which we refer to as *clients*) which are coordinated by a central *server*. Each client has a local training dataset which is never uploaded to the server. Instead, each client computes an update to the current global model maintained by the server, and only this update is communicated. This is a direct application of the principle of *focused collection* or *data minimization* proposed by the 2012 White House report on privacy of consumer data [39]. Since these updates are specific to improving the current model, there is no reason to store them once they have been applied.

A principal advantage of this approach is the decoupling of model training from the need for direct access to the raw training data. Clearly, some trust of the server coordinating the training is still required. However, for applications where the training objective can be specified on the basis of data available on each client, federated learning can significantly reduce privacy and security risks by limiting the attack surface to only the device, rather than the device and the cloud.

Our primary contributions are 1) the identification of the problem of training on decentralized data from mobile devices as an important research direction; 2) the selection of a straightforward and practical algorithm that can be applied to this setting; and 3) an extensive empirical evaluation of the proposed approach. More concretely, we introduce the `FederatedAveraging` algorithm, which combines local stochastic gradient descent (SGD) on each client with a server that performs model averaging. We perform extensive experiments on this algorithm, demonstrating it is robust to unbalanced and non-IID data distributions, and can reduce the rounds of communication needed to train a deep network on decentralized data by orders of magnitude.

Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

Communication-Efficient Learning of Deep Networks from Decentralized Data

McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." . PMLR, 2017.

(인용 : 19448회)

배경

모바일 장치의 확산
더 많은 사람들이 휴대폰과 태블릿을 주요 컴퓨팅 장치로 활용함
-> 해당 기기들의 센서에서 수집할 수 있는 방대한 데이터 존재

그러나 데이터 활용에 한계 존재:

- 개인정보와 관련됨
- 데이터 양이 너무 많음
- 데이터가 불균형한 분포를 가짐 (Non-IID)

해결책

" Federated Learning "

학습 데이터를 모바일 기기에 분산시킨 상태로 두고, 로컬에서 계산된 업데이트를 집계하여 공유 모델을 학습하는 방식

원시 훈련 데이터에 대한 직접적인 접근 없이 모델 훈련 가능

클라이언트 간에 데이터를 공유하지 않으며, 개인정보 보호와 보안 위험 감소

Federated Learning

이 방식의 효율성과 범용성을 확인하기 위해서 모델 아키텍처 5가지, 데이터셋 4가지에 대해 실험

주요 제약 사항 :
네트워크 분야에서 관심을 가지는 주제이다 보니, 통신 라운드, 즉 통신비용을 줄이는 것

우수성 :
불균형하고 Non-IID(Independent and Identically Distributed)한 데이터 분포에 robust함
분산 데이터에서 심층 네트워크를 훈련할 때 통신 라운드를 수십 배 줄일 수 있음을 입증

FederatedAveraging 알고리즘
각 클라이언트에서 Local SGD을 수행하고, 서버는 모델 averaging을 수행함

Federated Learning의 특성

저자의 자랑 :

- 데이터센터에서 사용하는 프록시 데이터에 비해 실생활 데이터를 활용할 수 있어 정확하다
- 사용자 상호작용 방식에 따라 자동 라벨링 가능
- 데이터센터에 데이터가 기록되지 않으므로 개인정보가 안전함 (데이터 최소화 원칙)

예시로, 이미지 데이터 (사용자가 찍은 사진이므로 개인정보), 텍스트 데이터 (키보드로 입력한 모든 것, 비밀번호, 채팅 내역 등등)

Federated Optimization이 필요한 이유

Non-IID한 데이터 분포
- 각 클라이언트의 훈련 데이터는 사용자의 모바일 기기 사용 방식에 따라 달라짐
- 특정 사용자의 로컬 데이터셋은 전체 사용자 집단의 분포를 대표하지 않음

불균형(Unbalanced)
- 일부 사용자는 다른 사용자들보다 서비스나 앱을 더 많이 사용함
- 로컬 훈련 데이터의 양이 다르게 나타남

대규모 분산(Massively Distributed)
- 최적화에 참여하는 클라이언트 수가 매우 많음

제한된 통신(Limited Communication)
- 모바일 기기는 자주 오프라인 상태이거나, 느리거나 비용이 많이 드는 연결을 사용

Federated Optimization의 실용적 문제

클라이언트 데이터셋의 변화
- 데이터의 추가 및 삭제로 인한 데이터셋의 변화

클라이언트 가용성 문제
- 지역 데이터 분포와 복잡하게 연관된 클라이언트 가용성 문제
  (미국 영어 사용자들의 휴대폰은 영국 영어 사용자들 휴대폰과 다른 시간대에 충전될 것)

비정상적인 클라이언트 응답
- 응답하지 않거나 손상된 업데이트를 보내는 클라이언트

Federated Learning에서의 최적화 문제

FL에서는 다음과 같은 형태의 **최적화 문제**를 다룹니다:

$$\min_{w \in \mathbb{R}^d} f(w), \quad \text{where} \quad f(w) = \frac{1}{n} \sum_{i=1}^{n} f$$

w: 최적화하고자 하는 모델 파라미터 벡터 (예: 신경망의 가중치)
$f(w)$: 전체 데이터셋에서의 손실 함수(목표 함수)
$f_i(w)$: 각 데이터 포인트 (x_i, y_i)에 대한 손실

즉, 전체 데이터셋의 평균 손실을 최소화하는 모델을 학습하려는 것

FL에서는 여러 클라이언트에 따라 데이터가 분산되어 있음을 감안한다면?

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w), \qquad F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k}$$

Federated Learning에서의 최적화 문제

IID (Independent and Identically Distributed):

- 데이터가 클라이언트 간에 무작위로 균등하게 배분된 경우
- 클라이언트 손실 $F_k(w)$의 기대값이 전체 손실 $f(w)$와 동일

$$\mathbb{E}_{\mathcal{P}_k}[F_k(w)] = f(w)$$

Non-IID:

- 각 클라이언트가 특이 데이터 분포를 따를 경우, $F_k(w)$가 전체 $f(w)$를 잘 근사하지 못함

  → 데이터가 클라이언트마다 편향된 분포를 가진다면 학습에 지장 발생

Data Center Optimization 와 Fed. Optimization의 차이점

데이터 센터 최적화

- 통신 비용이 상대적으로 작음
- 계산 비용이 주요 고려 사항
- GPU 사용으로 계산 비용을 낮추는 데 중점

연합 최적화

- 통신 비용이 주요 제약 조건
- 업로드 대역폭이 1 MB/s 이하로 제한됨
- 클라이언트는 특정 조건 하 자발적 참여
  (충전 중, 무제한 Wi-Fi에 연결된 경우 etc.)
- 각 클라이언트는 소수의 업데이트 라운드에만 참여

Fed. Optimization 에서의 목표와 접근 방식

목표
모델 학습에 필요한 통신 라운드 수를 줄이기 위해 추가 계산을 활용

두 가지 주요 방법
1. 병렬성 증가
2. 각 통신 라운드 사이에 더 많은 클라이언트를 독립적으로 활용

각 클라이언트에서의 계산 증가
단순한 그래디언트 계산 대신, 각 클라이언트가 더 복잡한 계산을 수행하여 성능 극대화

실험 결과
클라이언트 간 병렬성이 최소 수준을 넘은 이후, 클라이언트당 계산량 증가가 속도 향상의 주요 성능 개선 요인임을 확인

FederatedAveraging 알고리즘

딥러닝에서의 확률적 경사 하강법(SGD)
- 최근의 딥러닝 성공 사례는 대부분 SGD의 변형을 활용
- 대부분의 발전 케이스들은 모델 구조(및 손실 함수)를 SGD에 맞게 조정한 것

Fed. Optimization을 위한 SGD 기반 알고리즘 개발
- 연합 최적화 문제에 SGD를 자연스럽게 적용
- 한 라운드의 통신마다 Single Batch 그래디언트 계산을 수행

FedSGD (Federated SGD) 알고리즘

일반적인 FedSGD 구현
- C = 1 (전체 배치 경사 하강법)과 고정된 학습률 η를 가정
- 각 클라이언트 k는 로컬 데이터에서 평균 그래디언트 g_k를 계산
- 중앙 서버는 이 그래디언트를 합산하여 업데이트 적용:

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k$$

가중평균으로 구현
각 클라이언트가 로컬에서 하나의 그래디언트 계산 후, 서버가 가중 평균을 취함

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$$

(여기서 $w_{t+1}^k \leftarrow w_t - \eta g_k$)

FederatedAveraging (FedAvg) 알고리즘

로컬 업데이트 후 평균화
- 각 클라이언트는 로컬 업데이트를 반복적으로 수행하고, 그 후 평균화를 진행
- FedAvg는 이 방식으로 클라이언트 내에서 여러 번의 계산을 수행

$$w^k \leftarrow w^k - \eta \nabla F_k(w^k)$$

주요 매개변수
- C: 각 라운드에 계산을 수행하는 클라이언트 비율
- E: 각 라운드에서 클라이언트가 로컬 데이터셋에서 수행하는 훈련 횟수
- B: 클라이언트 업데이트에 사용되는 로컬 미니배치 크기

특별한 경우
- $B = \infty$는 전체 로컬 데이터셋을 단일 미니배치로 처리하는 경우
- $E = 1, B = \infty$는 FedSGD에 정확히 대응됨

Model Averaging의 문제

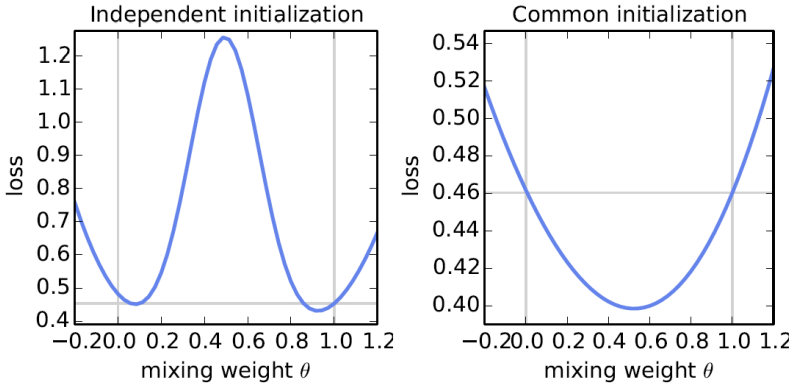
비볼록(Non-convex) 목표에서 평균화의 문제

- Non-convex 한 목표를 가지는 Parameter Space에서는 모델 성능이 낮을 수 있음
  (최적해가 여러 개 존재할 수 있기 때문에 그냥 나쁜 값을 정하게 됨)

Goodfellow et al.의 접근법

- 서로 다른 초기 조건의 두 MNIST 모델의 매개변수를 평균화할 때 발생하는 문제를 실험적으로 확인
- 모델 w와 w'를 비중 $\theta w + (1-\theta)w'$로 평균화
  (여기서 $\theta \in [-0.2, 1.2]$)

$\theta = 0.5$ 근처에서 결합하면
전체 손실이 모델 각각의 손실보다 낮을 수 있음



FedAvg 알고리즘

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:

initialize w_0

for each round $t = 1, 2, \dots$ **do**

$m \leftarrow \max(C \cdot K, 1)$

$S_t \leftarrow$ (random set of m clients)

for each client $k \in S_t$ **in parallel do**

$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*

$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)

for each local epoch i from 1 to E **do**

for batch $b \in \mathcal{B}$ **do**

$w \leftarrow w - \eta \nabla \ell(w; b)$

return w to server

FedAvg 성능 평가

이미지 분류와 언어 모델링 작업에서 FedAvg의 성능 평가

모바일 장치의 사용성 향상을 목표로 좋은 모델을 찾는 것을 목표로

이미지 분류 : MNIST 데이터셋
언어 모델링 : Shakespeare의 "The Complete Works" 데이터셋

FedAvg 이미지 분류 성능 평가

MNIST CNN:
2개의 5x5 컨볼루션 층, 512 유닛의 완전 연결층을 가진 모델(1,663,370개 매개변수)

MNIST CNN, 99% ACCURACY					
CNN	E	B	u	IID	NON-IID
FEDSGD	1	∞	1	626	483
FEDAVG	5	∞	5	179 (3.5×)	1000 (0.5×)
FEDAVG	1	50	12	65 (9.6×)	600 (0.8×)
FEDAVG	20	∞	20	234 (2.7×)	672 (0.7×)
FEDAVG	1	10	60	34 (18.4×)	350 (1.4×)
FEDAVG	5	50	60	29 (21.6×)	334 (1.4×)
FEDAVG	20	50	240	32 (19.6×)	426 (1.1×)
FEDAVG	5	10	300	20 (31.3×)	229 (2.1×)
FEDAVG	20	10	1200	18 (34.8×)	173 (2.8×)

FedAvg 이미지 분류 성능 평가

LSTM :
256 노드를 가진 2개의 LSTM 층을 통과한 후 소프트맥스 출력층에서 각 문자를 예측

SHAKESPEARE LSTM, 54% ACCURACY

LSTM	E	B	u	IID	NON-IID
FEDSGD	1	∞	1.0	2488	3906
FEDAVG	1	50	1.5	1635 (1.5×)	549 (7.1×)
FEDAVG	5	∞	5.0	613 (4.1×)	597 (6.5×)
FEDAVG	1	10	7.4	460 (5.4×)	164 (23.8×)
FEDAVG	5	50	7.4	401 (6.2×)	152 (25.7×)
FEDAVG	5	10	37.1	192 (13.0×)	41 (95.3×)

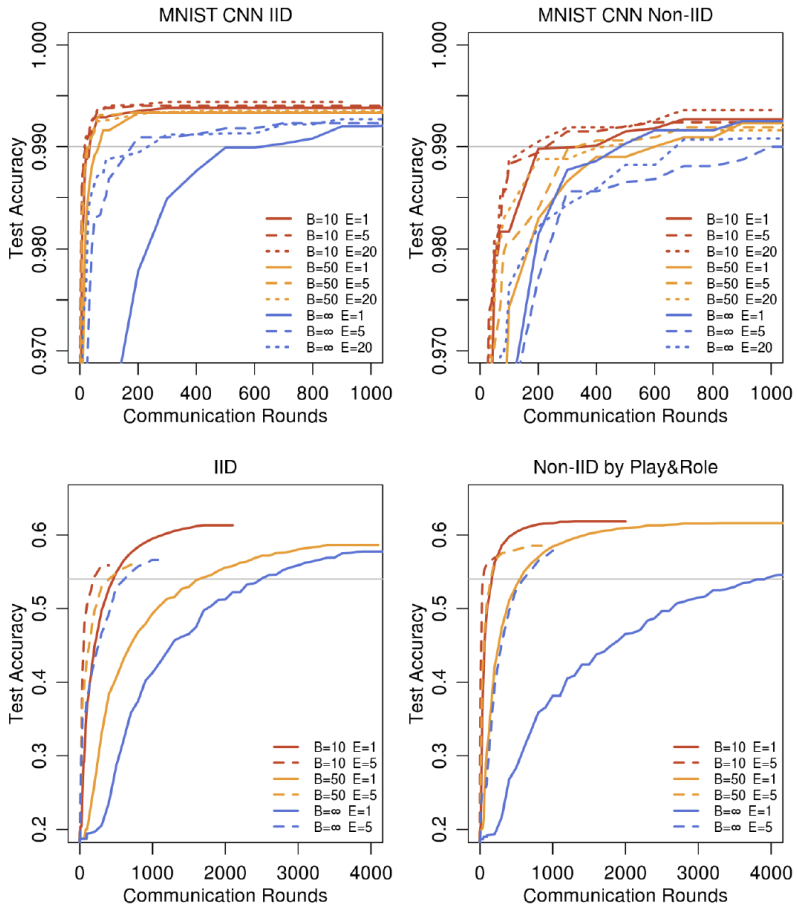
성능 추이 관찰

클라이언트 병렬 처리 증가:

$B = \infty$:
클라이언트의 모든 데이터를 한 번에 처리하는 경우, C 증가로 인한 개선 효과가 작음

$B = 10$ (소규모 배치):
특히 Non-IID 데이터에서 $C \geq 0.1$일 때 성능이 눈에 띄게 개선됨

$C = 0.1$로 고정:
계산 효율성과 수렴 속도 간의 균형이 가장 적절하다고 판단



성능 추이 관찰

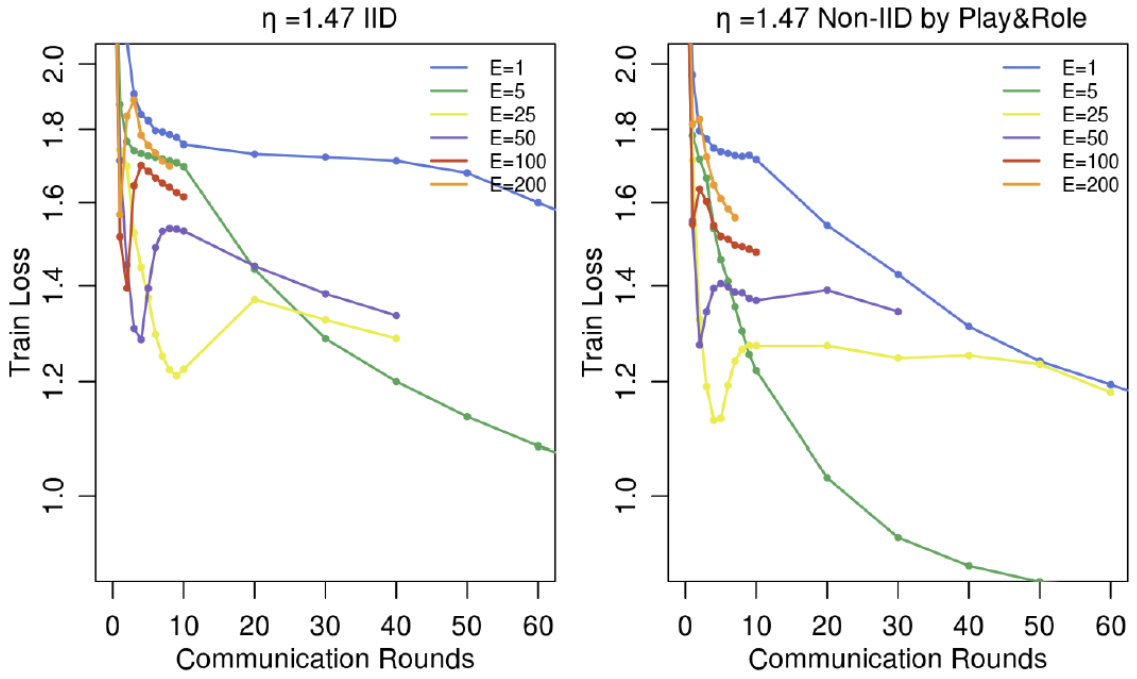


Figure 3: The effect of training for many local epochs (large E) between averaging steps, fixing $B = 10$ and $C = 0.1$ for the Shakespeare LSTM with a fixed learning rate $\eta = 1.47$.

성능 추이 관찰

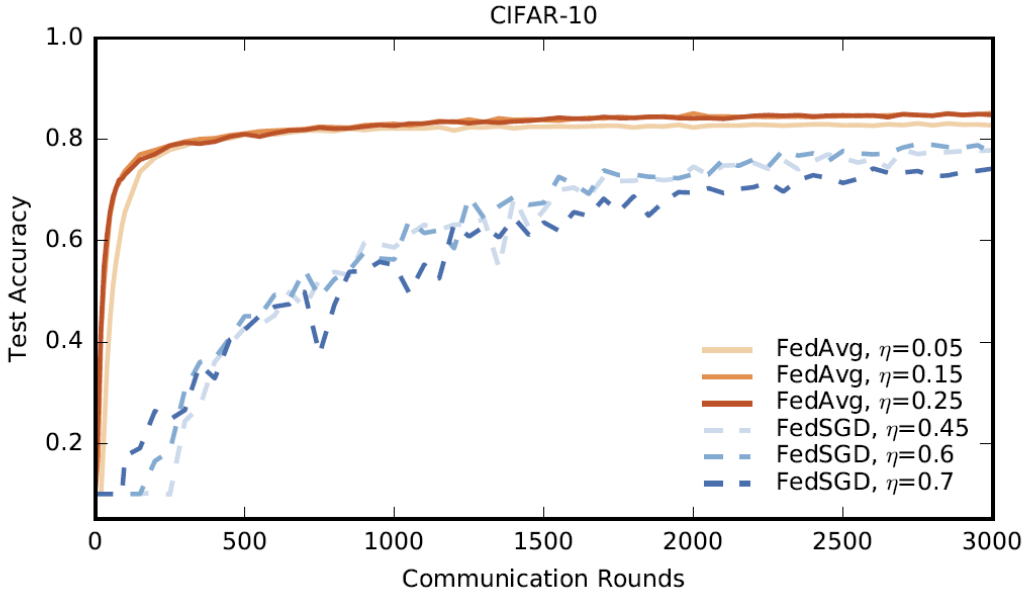


Figure 4: Test accuracy versus communication for the CIFAR10 experiments. `FedSGD` uses a learning-rate decay of 0.9934 per round; `FedAvg` uses $B = 50$, learning-rate decay of 0.99 per round, and $E = 5$.

성능 추이 관찰

Table 3: Number of rounds and speedup relative to baseline SGD to reach a target test-set accuracy on CIFAR10. SGD used a minibatch size of 100. `FedSGD` and `FedAvg` used $C = 0.1$, with `FedAvg` using $E = 5$ and $B = 50$.

ACC.	80%	82%	85%
SGD	18000 (—)	31000 (—)	99000 (—)
FEDSGD	3750 (4.8×)	6600 (4.7×)	N/A (—)
FEDAVG	280 (64.3×)	630 (49.2×)	2000 (49.5×)

성능 추이 관찰

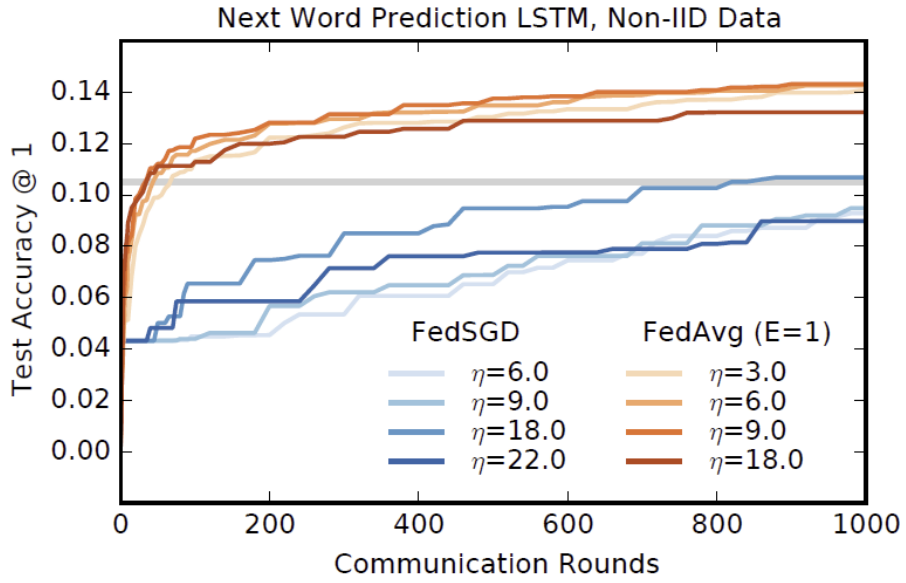


Figure 5: Monotonic learning curves for the large-scale language model word LSTM.

결론

연합 학습의 실용성
- 실험 결과, FedAvg는 적은 통신 라운드로 고품질 모델을 효과적으로 학습할 수 있음
- 다양한 모델 아키텍처에서 성능 확인:
  - 다층 퍼셉트론
  - 두 종류의 컨볼루션 신경망
  - 2층 문자 수준 LSTM
  - 대규모 단어 수준 LSTM

개선된 개인정보 보호
- 차등 개인정보 보호와 안전한 다자간 계산 또는 이들의 조합을 통해 더 강력한 개인정보 보호 보장을 탐구
- 이러한 기술들은 FedAvg와 같은 동기식 알고리즘과 자연스럽게 결합 가능

연구의 흥미로운 방향
- 개인정보 보호 및 보안 강화
- 연합 학습의 실용적 적용을 위한 추가 연구

감사합니다