Introduction to Federated Learning

2022_Fall

x

Federated Learning

Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective.

Advances and Open Problems in Federated Learning,
https://arxiv.org/abs/1912.04977

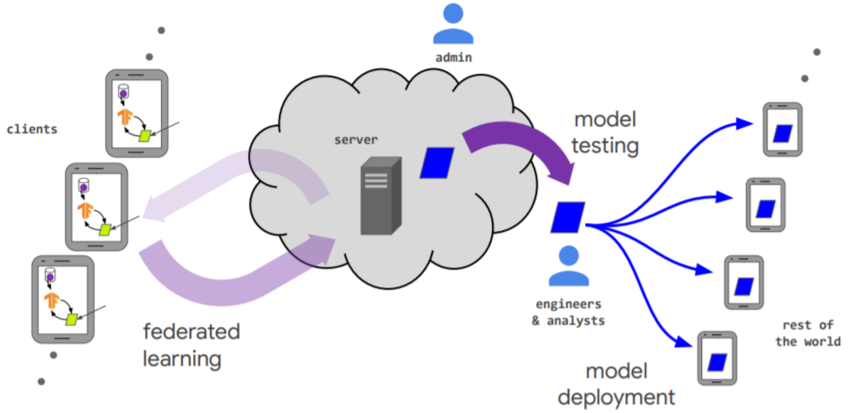
연합학습은 중앙 서버 또는 서비스 제공자의 관리 하에, 다수의 클라이언트/디바이스가 기계학습 문제를 해결하기 위해 협력하는 기술

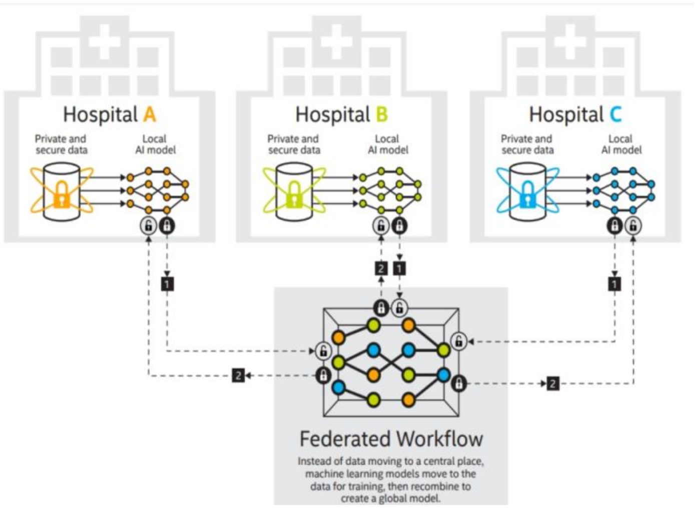
- 각 클라이언트/디바이스는 보유한/생산한 원시 데이터를 교환 또는 (중앙으로) 전송하지 않고, 로컬모델 학습에만 사용함으로써, 데이터 생산자의 프라이버시 보호
- 각 클라이언트/디바이스에서의 학습 결과는 (중앙의) 글로벌 모델 학습에 반영/기여. 'A fed B'학습의 성능은 'A+B'성능에 근사
- 데이터 생산자의 프라이버시 보호, 통신 오버헤드 감소

Federated Learning

▶ 개인 정보의 노출/침해 없이, 데이터를 확보/활용할 수 있는 연합학습 기술

- 인공지능 모델을 학습하기 위해서는 많은 양의 데이터가 필요하지만, 데이터 프라이버시 정책 등으로 인하여 (개인)데이터 수집/활용에 제약
- 기존에는 중앙 서버에 모든 데이터를 수집 후 학습하는 과정이 일반적으로, 프라이버시 침해 위험이 존재. 이를 개선하기 위해 각 디바이스에서 로컬 모델을 학습하고 이를 동기화하는 연합학습 기술 필요성 대두
- 연합학습 기술은 사용자 로컬 데이터에 직접 접근하지 않으면서 모든 사용자들의 정보를 반영한 글로벌 모델을 학습하여 이용할 수 있음





Federated Workflow

Instead of data moving to a central place, machine learning models move to the data for training, then recombine to create a global model.

연합학습 개요

- 연합학습은, 로컬 데이터 샘플을 보유하는 다수의 분산 에지 장치 또는 서버들이 원시 데이터를 교환/공유하지 않고 기계학습 문제를 해결하기 위해 협력하는 기술
- 각 로컬노드(클라이언트/디바이스)는 생산한/보유한 원시 데이터를 로컬모델 학습에만 사용함으로써, **데이터 생산자/제공자의 프라이버시를 보호하고, 데이터 소유/활용의 파편화 문제를 해결**
- 모든 로컬 데이터 세트가 하나의 서버에 업로드/공유 되는 전통적인 중앙집중식 기계학습 방식 혹은 로컬 데이터 샘플이 동일하게 분포 (identically distributed) 된다고 가정하는 전통적인 분산접근 방식과는 대비됨
- 연합학습은 데이터 소유/관리/활용의 파편화 문제를 해결하기 위한 사일로-교차(Cross-silo) 연합학습, 디바이스/서비스 사용자 데이터를 활용하기 위한 디바이스-교차(Cross-device) 연합학습으로 특징과 이슈를 구분

	분산학습 (Datacenter distributed learning)	사일로-교차 연합학습 (Cross-silo federated learning)	디바이스-교차 연합학습 (Cross-device federated learning)
환경	단일 크러스터 혹은 데이터센터가 대규모 데이터로 학습	서로 다른 기관(의료 혹은 금융) 혹은 지리적으로 분산되어 있는 데이터센터들이, 각자의 사일로 데이터를 학습	**클라이언트는 많은 수의 모바일 혹은 IoT 디바이스**
데이터 분산	데이터는 중앙에 저장되며, 클라이언트들은 데이터에 제한 없이 접근, 혼합	**데이터는 로컬에서 생성, 분산되어 있음. 각 클라이언트는 자신의 데이터를 저장하며 다른 클라이언트의 데이터를 읽을 수 없음. 데이터는 iid (independently or identically distributed) 하지 않음**	
오케스트레이션	중앙에서 데이터 관리와 학습을 관장	**중앙 오케스트레이션 서버/서비스 주도로 학습을 관장하지만, 원시 데이터에는 접근하지 않음**	
데이터 가용성	모든 클라이언트가 항상 가용		**일정 시간에, 일부 클라이언트만 가용**
분산 규모	1 - 1000 클라이언트	2 - 100 클라이언트	**10^{10} 까지 대규모**
주요 병목	Computation (연산량 및 연산속도)	연산 및 통신	**일반적으로 통신이 주된 병목**

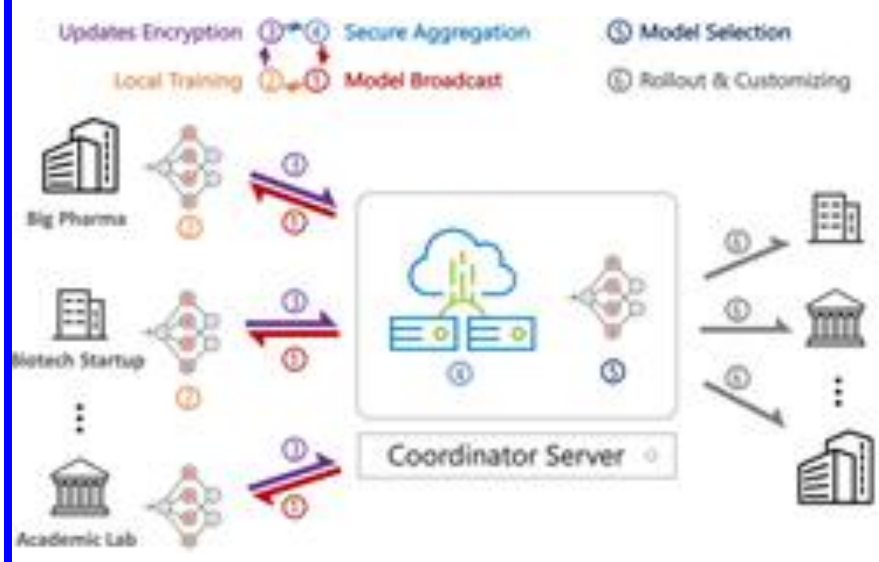
Advances and Open Problems in Federated Learning, https://arxiv.org/abs/1912.04977

Typical characteristics of federated learning settings vs. distributed learning in the datacenter

	Datacenter distributed learning	Cross-silo federated learning	Cross-device federated learning
Setting	Training a model on a large but “flat” dataset. Clients are compute nodes in a single cluster or datacenter.	Training a model on siloed data. Clients are different organizations (e.g. medical or financial) or geo-distributed datacenters.	The clients are a very large number of mobile or IoT devices
Data distribution	Data is centrally stored and can be shuffled and balanced across clients. Any client can read any part of the dataset.	**Data is generated locally and remains decentralized.** Each client stores its own data and cannot read the data of other clients. Data is not independently or identically distributed.	
Orchestration	Centrally orchestrated.	**A central orchestration server/service organizes the training**, but never sees raw data	
Wide-area communication	None (fully connected clients in one datacenter/cluster).	Hub-and-spoke topology, with the hub representing a coordinating service provider (typically without data) and the spokes connecting to clients.	
Data availability	All clients are almost always available.		Only a fraction of clients are available at any one time, often with diurnal or other variations.
Distribution scale	Typically 1 - 1000 clients.	Typically 2 - 100 clients.	Massively parallel, up to 1010 clients
Primary bottleneck	Computation is more often the bottleneck in the datacenter, where very fast networks can be assumed.	Might be computation or communication.	Communication is often the primary bottleneck, though it depends on the task. Generally, cross-device federated computations use wi-fi or slower connections.
……			

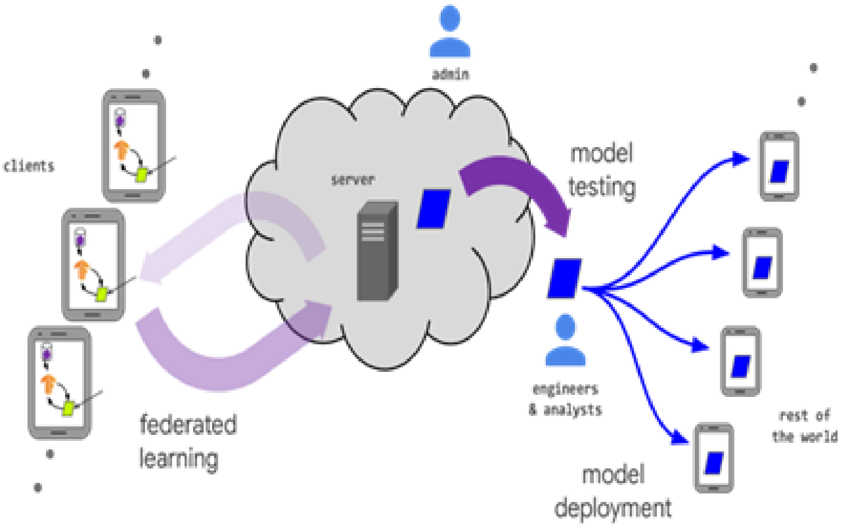
Advances and Open Problems in Federated Learning, https://arxiv.org/abs/1912.04977

연합학습 개요 : Cross-silo vs. Cross-device



사일로-교차 연합학습 (Cross-silo FL) :

- 서로 다른 기관 (의료 혹은 금융) 혹은 지리적으로 분산되어 있는 데이터센터들이, 각자의 사일로 데이터를 학습 : 2 - 100 clients
- 데이터/통계적 이질성, 디바이스/시스템적 이질성 문제 小
- 모든 클라이언트가 항상 가용



디바이스-교차 연합학습 (Cross-device FL) :

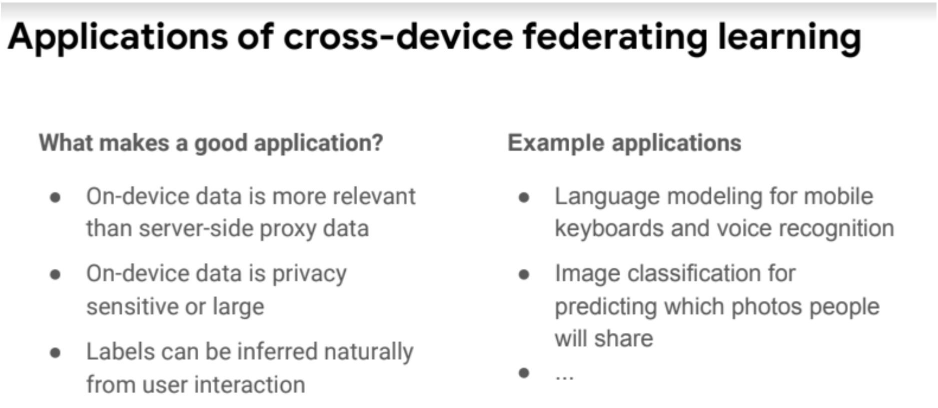
- 사용자의 개인 디바이스 (휴대폰, IoT) 가 개인 데이터를 학습 : Massive # of clients
- 데이터/통계적 이질성, 디바이스/시스템적 이질성 문제 大
- 일정 시간에 일부 클라이언트만 가용하고, straggler effect 대응 필요

* **통계적 이질성**: 다수의 다양한 사용자/디바이스, 동적 환경 및 시공간으로부터 수집된 데이터는 독립동일분포(iid: independent identically distributed) 조건을 만족하지 못하고 비균일/불균형의 특성을 지님

** **시스템적 이질성**: 연합학습에 참여/기여하는 디바이스의 성능과 기능 및 네트워크 환경이 다양하고, 디바이스의 추가, 변동이 지속적으로 발생

인공지능 기술청사진 2030 2차년도 보고서,
https://www.iitp.kr/kr/1/knowledge/openReference/view.it?ArticleIdx=5248&count=true

연합학습 개요



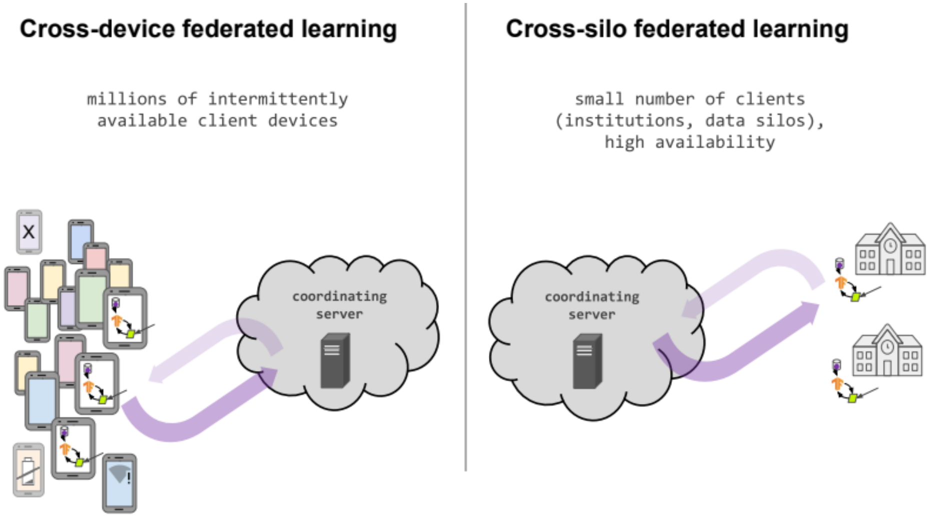
Applications of cross-device federating learning

What makes a good application?

- On-device data is more relevant than server-side proxy data
- On-device data is privacy sensitive or large
- Labels can be inferred naturally from user interaction

Example applications

- Language modeling for mobile keyboards and voice recognition
- Image classification for predicting which photos people will share
- ...

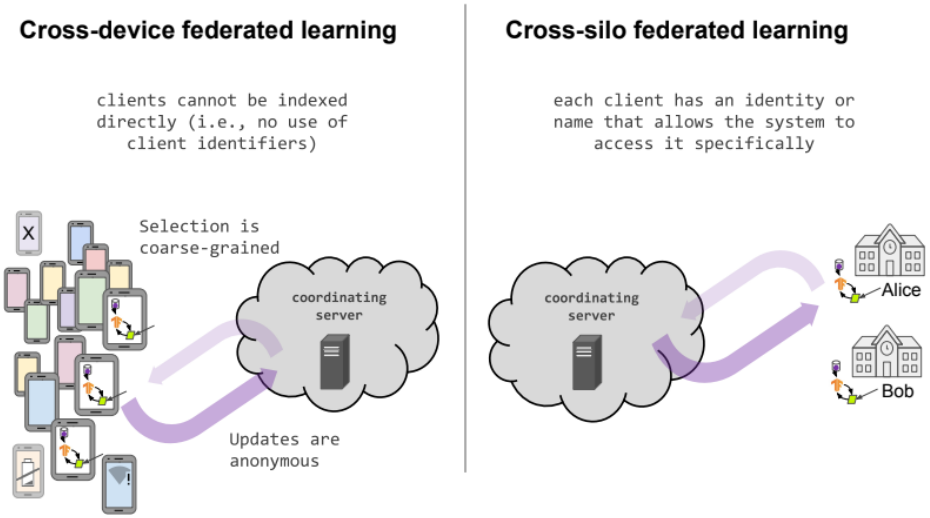


Cross-device federated learning

millions of intermittently available client devices

Cross-silo federated learning

small number of clients (institutions, data silos), high availability



Cross-device federated learning

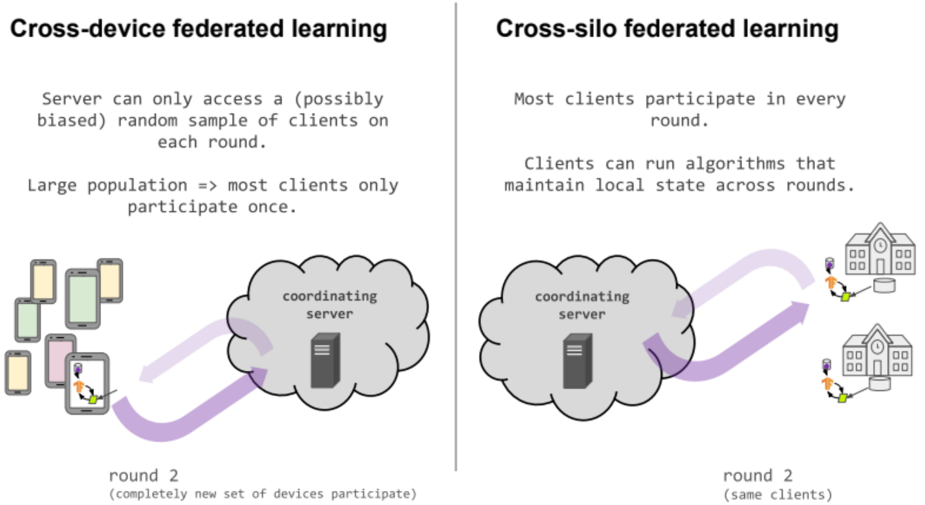
clients cannot be indexed directly (i.e., no use of client identifiers)

Selection is coarse-grained

Updates are anonymous

Cross-silo federated learning

each client has an identity or name that allows the system to access it specifically



Cross-device federated learning

Server can only access a (possibly biased) random sample of clients on each round.

Large population => most clients only participate once.

round 2
(completely new set of devices participate)

Cross-silo federated learning

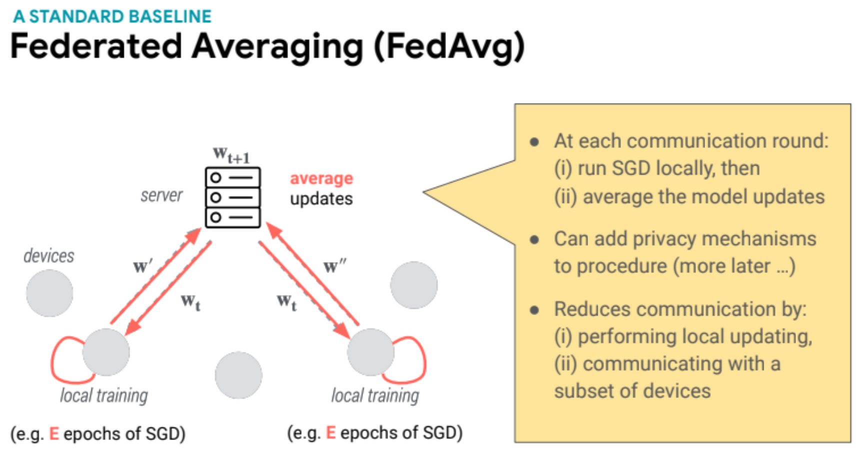
Most clients participate in every round.

Clients can run algorithms that maintain local state across rounds.

round 2
(same clients)

Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

연합학습 개요

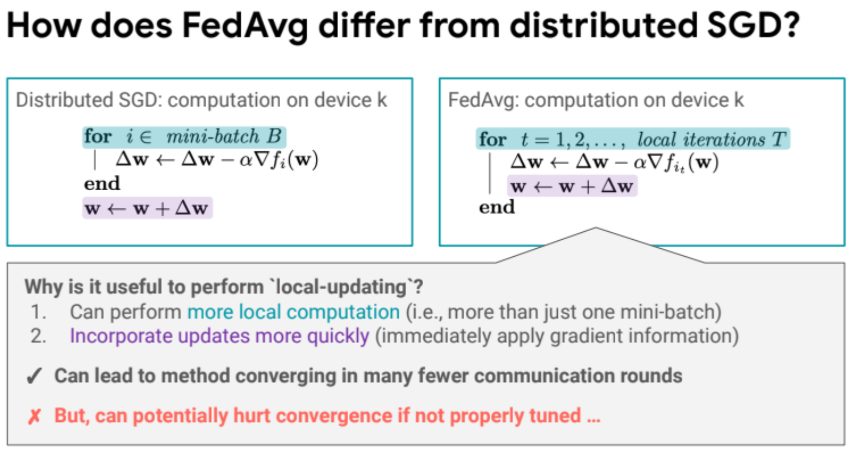


A STANDARD BASELINE

Federated Averaging (FedAvg)

(e.g. E epochs of SGD) (e.g. E epochs of SGD)

- At each communication round: (i) run SGD locally, then (ii) average the model updates
- Can add privacy mechanisms to procedure (more later ...)
- Reduces communication by: (i) performing local updating, (ii) communicating with a subset of devices



How does FedAvg differ from distributed SGD?

Distributed SGD: computation on device k

```
for $i \in$ mini-batch $B$
    $\Delta \mathbf{w} \leftarrow \Delta \mathbf{w} - \alpha \nabla f_i(\mathbf{w})$
end
$\mathbf{w} \leftarrow \mathbf{w} + \Delta \mathbf{w}$
```

FedAvg: computation on device k

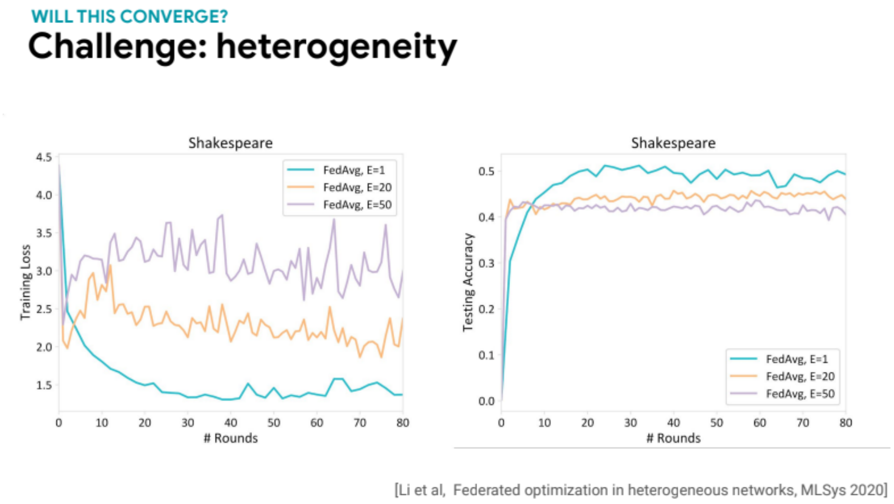
```
for $t = 1, 2, \ldots,$ local iterations $T$
    $\Delta \mathbf{w} \leftarrow \Delta \mathbf{w} - \alpha \nabla f_{i_t}(\mathbf{w})$
    $\mathbf{w} \leftarrow \mathbf{w} + \Delta \mathbf{w}$
end
```

Why is it useful to perform `local-updating`?

1. Can perform more local computation (i.e., more than just one mini-batch)
2. Incorporate updates more quickly (immediately apply gradient information)

✓ **Can lead to method converging in many fewer communication rounds**

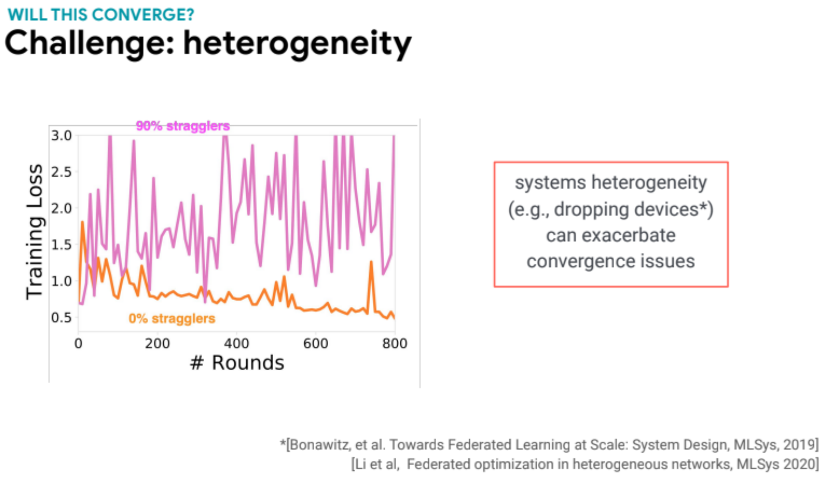
✗ **But, can potentially hurt convergence if not properly tuned ...**



WILL THIS CONVERGE?

Challenge: heterogeneity

[Li et al, Federated optimization in heterogeneous networks, MLSys 2020]



WILL THIS CONVERGE?

Challenge: heterogeneity

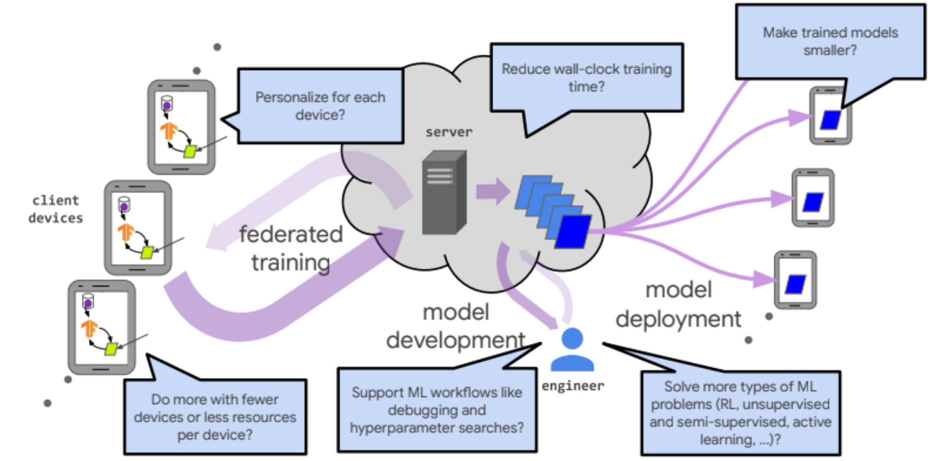
systems heterogeneity (e.g., dropping devices*) can exacerbate convergence issues

*[Bonawitz, et al. Towards Federated Learning at Scale: System Design, MLSys, 2019]
[Li et al, Federated optimization in heterogeneous networks, MLSys 2020]

Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

연합학습 개요

Improving efficiency and effectiveness



Personalize for each device?

Reduce wall-clock training time?

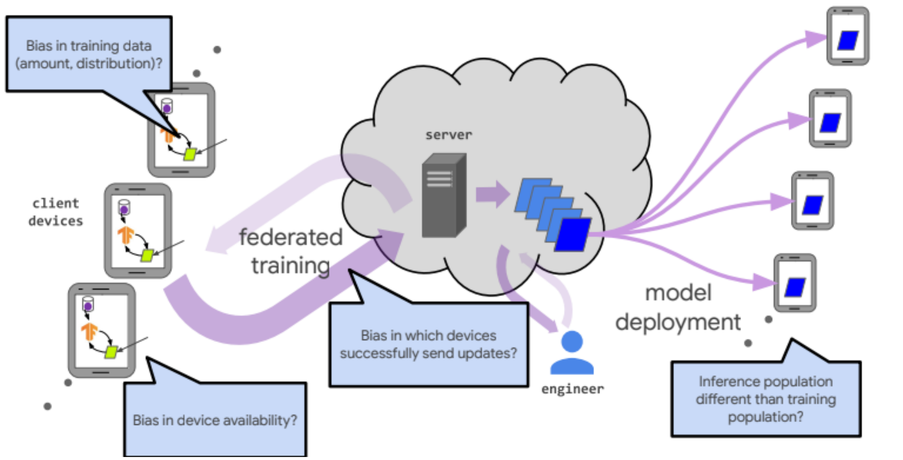
Make trained models smaller?

Do more with fewer devices or less resources per device?

Support ML workflows like debugging and hyperparameter searches?

Solve more types of ML problems (RL, unsupervised and semi-supervised, active learning, ...)?

Ensuring fairness and addressing sources of bias



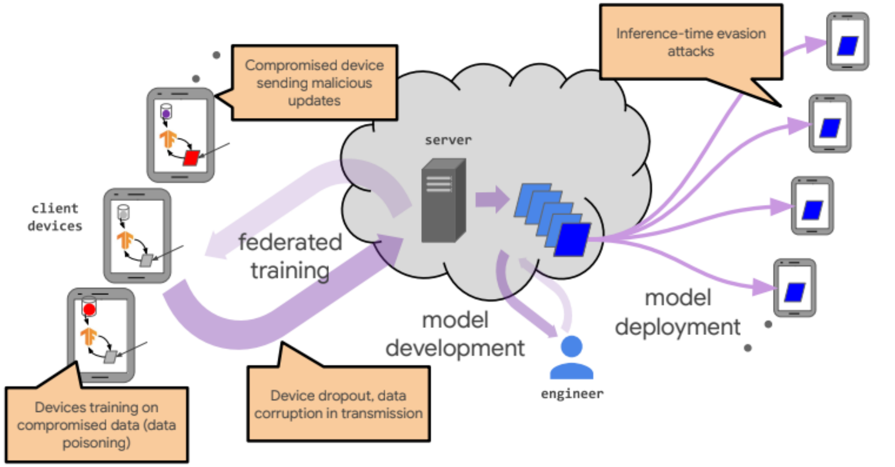
Bias in training data (amount, distribution)?

Bias in which devices successfully send updates?

Bias in device availability?

Inference population different than training population?

Robustness to attacks and failures



Compromised device sending malicious updates

Inference-time evasion attacks

Devices training on compromised data (data poisoning)

Device dropout, data corruption in transmission

Advances and Open Problems in Federated Learning

Peter Kairouz[7*] H. Brendan McMahan[7*] Brendan Avent[21] Aurélien Bellet[9]
Mehdi Bennis[19] Arjun Nitin Bhagoji[13] Keith Bonawitz[7] Zachary Charles[7]
Graham Cormode[23] Rachel Cummings[6] Rafael G.L. D'Oliveira[14]
Salim El Rouayheb[14] David Evans[22] Josh Gardner[24] Zachary Garrett[7]
Adrià Gascón[7] Badih Ghazi[7] Phillip B. Gibbons[2] Marco Gruteser[7,14]
Zaid Harchaoui[24] Chaoyang He[21] Lie He [4] Zhouyuan Huo [20]
Ben Hutchinson[7] Justin Hsu[25] Martin Jaggi[4] Tara Javidi[17] Gauri Joshi[2]
Mikhail Khodak[2] Jakub Konečný[7] Aleksandra Korolova[21] Farinaz Koushanfar[17]
Sanmi Koyejo[7,18] Tancrède Lepoint[7] Yang Liu[12] Prateek Mittal[13]
Mehryar Mohri[7] Richard Nock[1] Ayfer Özgür[15] Rasmus Pagh[7,10]
Mariana Raykova[7] Hang Qi[7] Daniel Ramage[7] Ramesh Raskar[11]
Dawn Song[16] Weikang Song[7] Sebastian U. Stich[4] Ziteng Sun[3]
Ananda Theertha Suresh[7] Florian Tramèr[15] Praneeth Vepakomma[11] Jianyu Wang[2]
Li Xiong[5] Zheng Xu[7] Qiang Yang[8] Felix X. Yu[7] Han Yu[12] Sen Zhao[7]

[1] Australian National University, [2] Carnegie Mellon University, [3] Cornell University,
[4] École Polytechnique Fédérale de Lausanne, [5] Emory University, [6] Georgia Institute of Technology,
[7] Google Research, [8] Hong Kong University of Science and Technology, [9] INRIA, [10] IT University of Copenhagen,
[11] Massachusetts Institute of Technology, [12] Nanyang Technological University, [13] Princeton University,
[14] Rutgers University, [15] Stanford University, [16] University of California Berkeley,
[17] University of California San Diego, [18] University of Illinois Urbana-Champaign, [19] University of Oulu,
[20] University of Pittsburgh, [21] University of Southern California, [22] University of Virginia,
[23] University of Warwick, [24] University of Washington, [25] University of Wisconsin–Madison

Abstract

Federated learning (FL) is a machine learning setting where many clients (e.g. mobile devices or whole organizations) collaboratively train a model under the orchestration of a central server (e.g. service provider), while keeping the training data decentralized. FL embodies the principles of focused data collection and minimization, and can mitigate many of the systemic privacy risks and costs resulting from traditional, centralized machine learning and data science approaches. Motivated by the explosive growth in FL research, this paper discusses recent advances and presents an extensive collection of open problems and challenges.

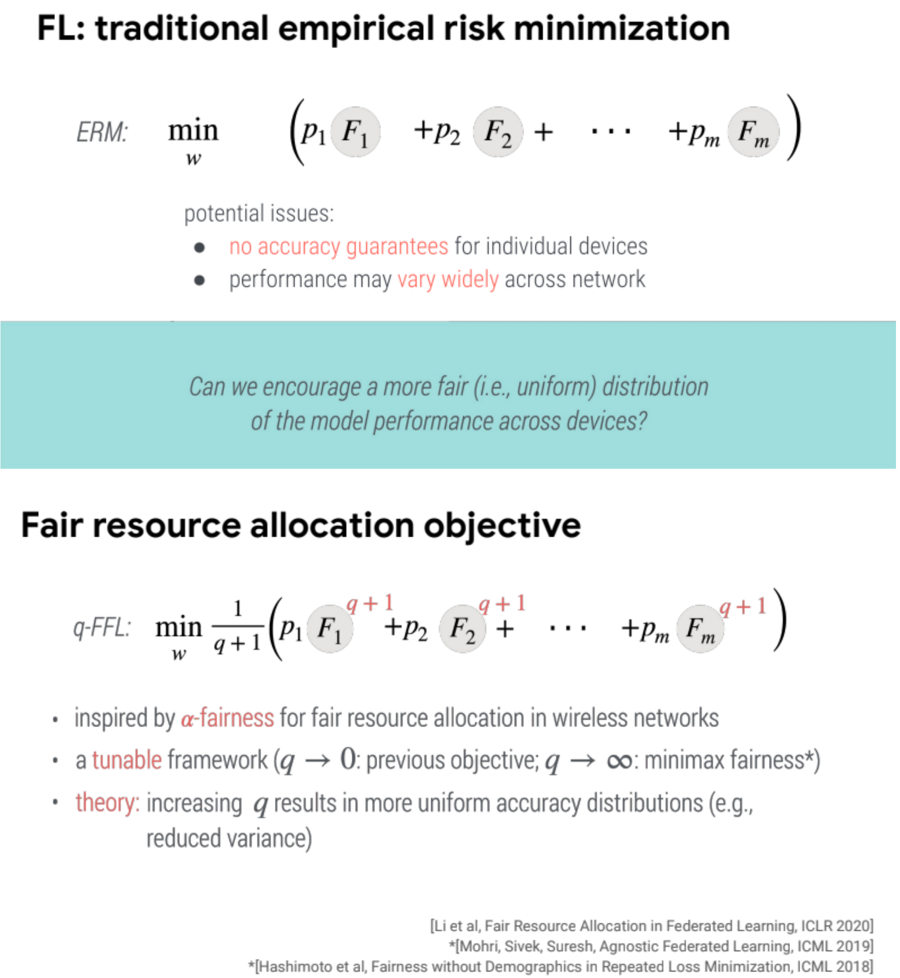
Advances and Open Problems in FL

58 authors from 25 top institutions

arxiv.org/abs/1912.04977

Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

연합학습 개요



FL: traditional empirical risk minimization

ERM: $\min_{w} \left(p_1 F_1 + p_2 F_2 + \cdots + p_m F_m \right)$

potential issues:

- no accuracy guarantees for individual devices
- performance may vary widely across network

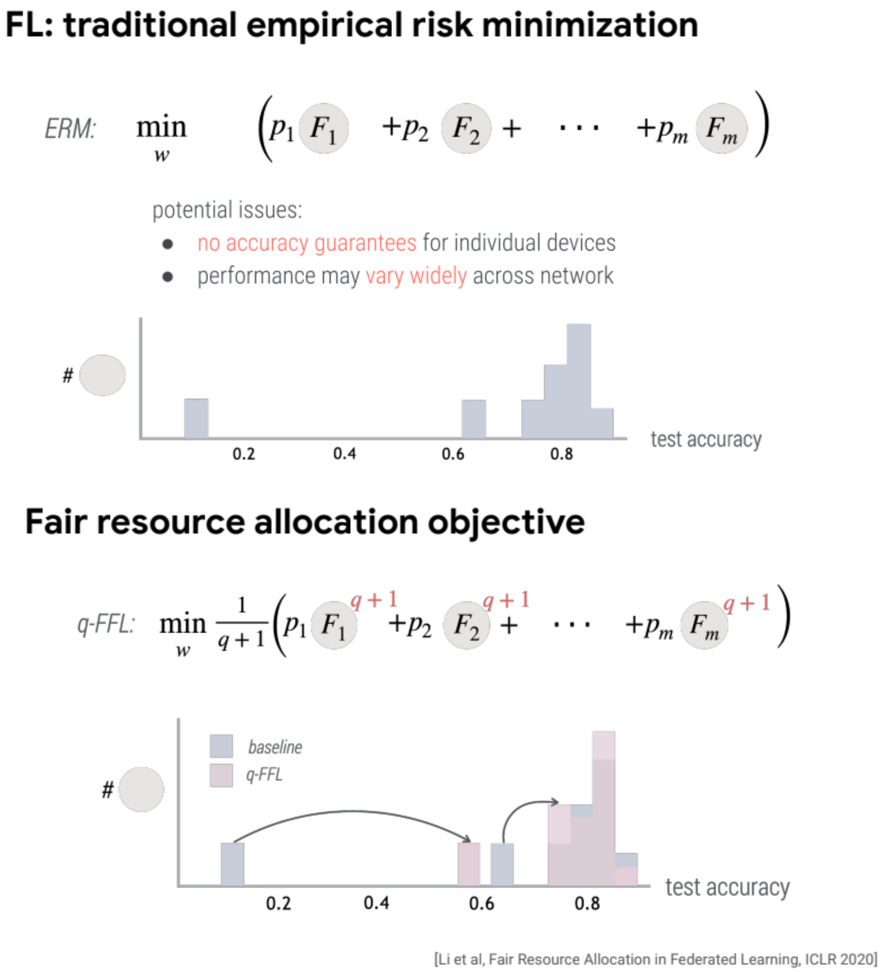
Can we encourage a more fair (i.e., uniform) distribution of the model performance across devices?

Fair resource allocation objective

q-FFL: $\min_{w} \frac{1}{q+1} \left(p_1 F_1^{q+1} + p_2 F_2^{q+1} + \cdots + p_m F_m^{q+1} \right)$

- inspired by α-fairness for fair resource allocation in wireless networks
- a tunable framework ($q \rightarrow 0$: previous objective; $q \rightarrow \infty$: minimax fairness*)
- theory: increasing q results in more uniform accuracy distributions (e.g., reduced variance)

[Li et al, Fair Resource Allocation in Federated Learning, ICLR 2020]
*[Mohri, Sivek, Suresh, Agnostic Federated Learning, ICML 2019]
*[Hashimoto et al, Fairness without Demographics in Repeated Loss Minimization, ICML 2018]



FL: traditional empirical risk minimization

ERM: $\min_{w} \left(p_1 F_1 + p_2 F_2 + \cdots + p_m F_m \right)$

potential issues:

- no accuracy guarantees for individual devices
- performance may vary widely across network

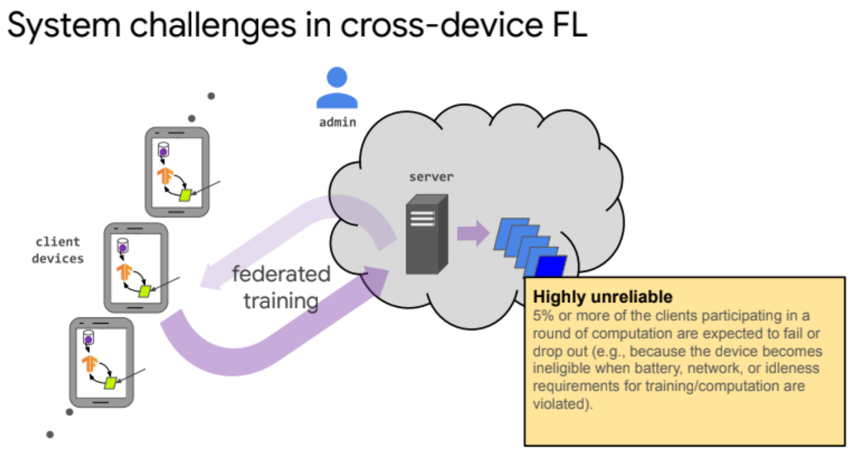
Fair resource allocation objective

q-FFL: $\min_{w} \frac{1}{q+1} \left(p_1 F_1^{q+1} + p_2 F_2^{q+1} + \cdots + p_m F_m^{q+1} \right)$

[Li et al, Fair Resource Allocation in Federated Learning, ICLR 2020]

Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

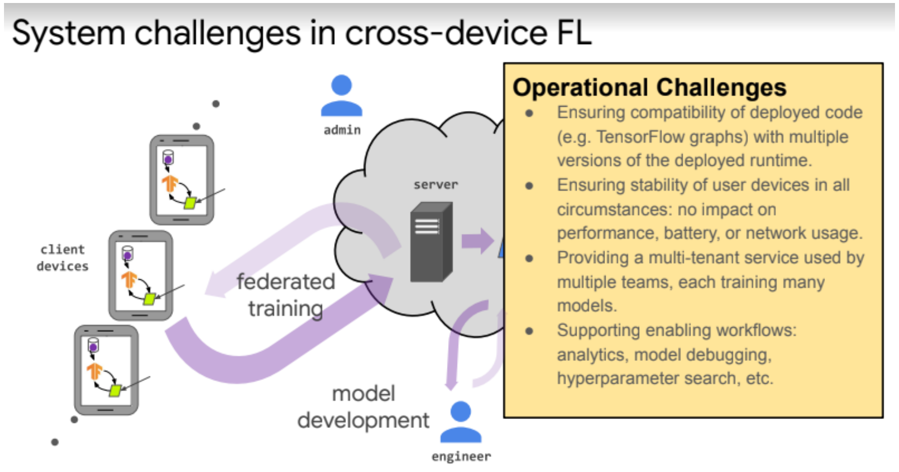
연합학습 개요



System challenges in cross-device FL

Highly unreliable

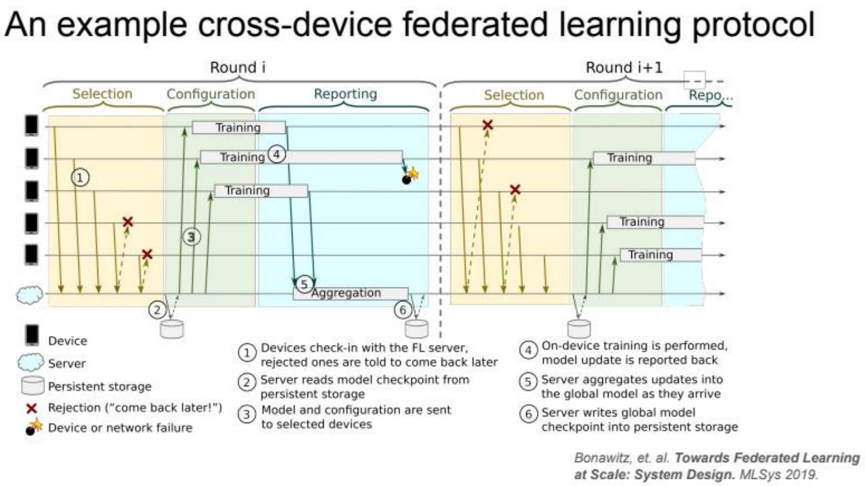
5% or more of the clients participating in a round of computation are expected to fail or drop out (e.g., because the device becomes ineligible when battery, network, or idleness requirements for training/computation are violated).



System challenges in cross-device FL

Operational Challenges

- Ensuring compatibility of deployed code (e.g. TensorFlow graphs) with multiple versions of the deployed runtime.
- Ensuring stability of user devices in all circumstances: no impact on performance, battery, or network usage.
- Providing a multi-tenant service used by multiple teams, each training many models.
- Supporting enabling workflows: analytics, model debugging, hyperparameter search, etc.

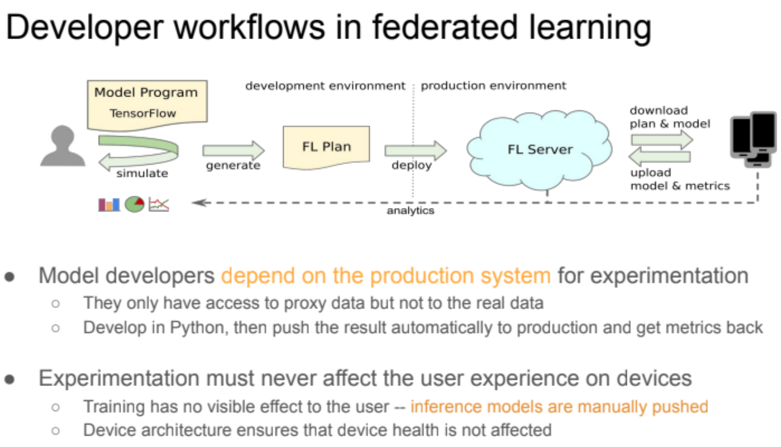


An example cross-device federated learning protocol

Device
Server
Persistent storage
Rejection ("come back later!")
Device or network failure

1. Devices check-in with the FL server, rejected ones are told to come back later
2. Server reads model checkpoint from persistent storage
3. Model and configuration are sent to selected devices
4. On-device training is performed, model update is reported back
5. Server aggregates updates into the global model as they arrive
6. Server writes global model checkpoint into persistent storage

Bonawitz, et. al. ***Towards Federated Learning at Scale: System Design.*** *MLSys 2019.*

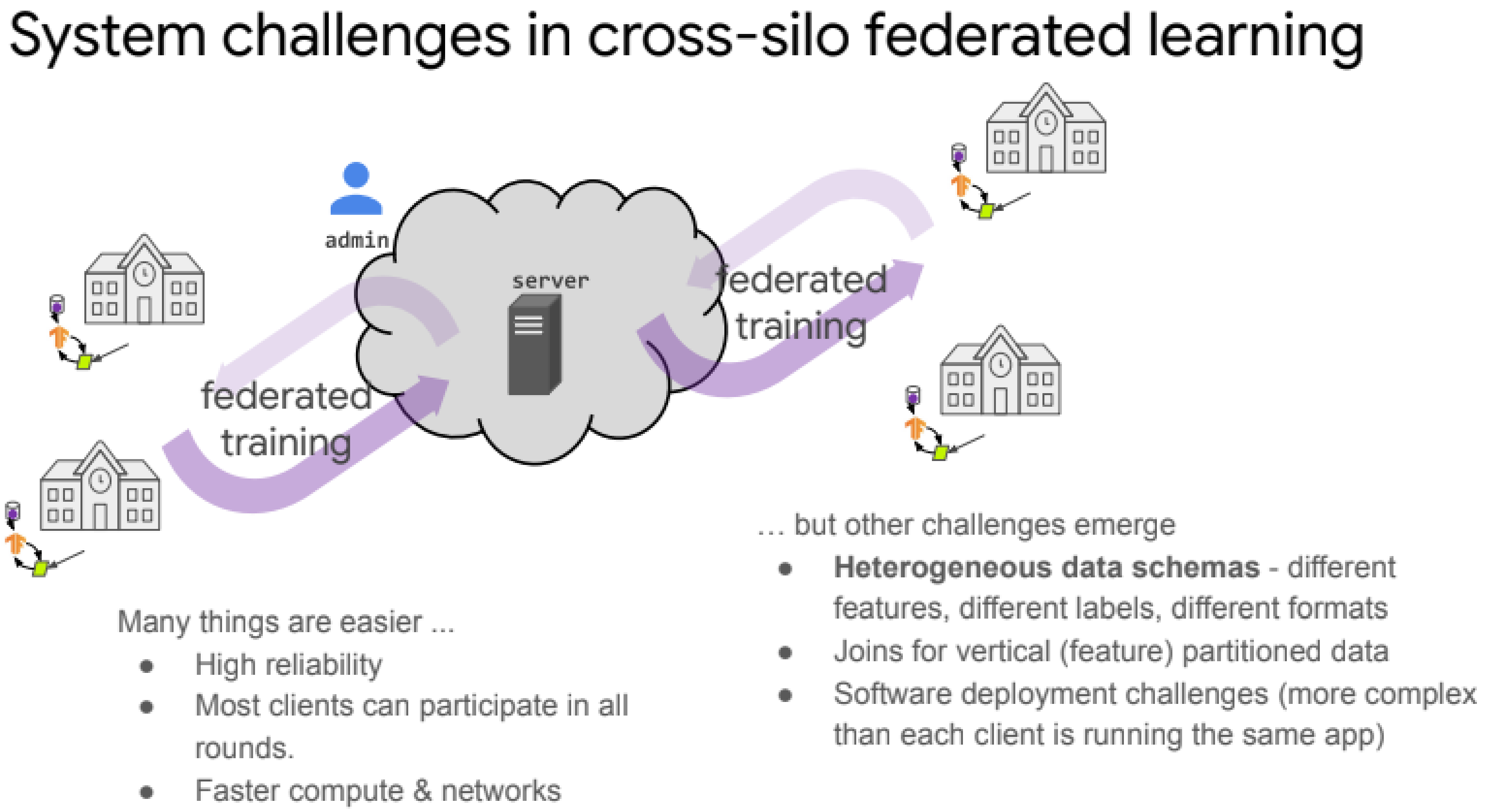


Developer workflows in federated learning

- Model developers depend on the production system for experimentation
  - They only have access to proxy data but not to the real data
  - Develop in Python, then push the result automatically to production and get metrics back
- Experimentation must never affect the user experience on devices
  - Training has no visible effect to the user -- inference models are manually pushed
  - Device architecture ensures that device health is not affected

Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

연합학습 개요



System challenges in cross-silo federated learning

Many things are easier ...

- High reliability
- Most clients can participate in all rounds.
- Faster compute & networks

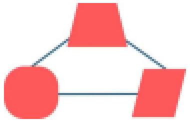
… but other challenges emerge

- **Heterogeneous data schemas** - different features, different labels, different formats
- Joins for vertical (feature) partitioned data
- Software deployment challenges (more complex than each client is running the same app)

Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

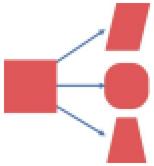
연합학습 개요

Approaches for personalization



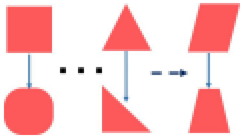
Multi-task learning

- Jointly learn shared, yet personalized models



Fine-tuning

- Learn a global model, then “fine-tune”/adapt it on local data
- See also: transfer learning, domain adaptation



Meta learning (initialization-based)

- Learn initialization over multiple tasks, then train locally

Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

Personalization for FL

*** 연합학습은 일반적으로 모든 디바이스 및 사용자에 공통으로 적용되는 글로벌모델을 학습하는 것을 목표로 하고 있으나, 동적인 디바이스 환경의 데이터 이질성 및 디바이스 이질성으로 인하여 **모든 디바이스에서 잘 동작하는 하나의 모델을 학습하기 어려우며, 개별 디바이스 및 사용자 관점에서 최적의 성능이 보장되지 않음.** 동적인 디바이스 환경에서 각 사용자 및 디바이스의 특징과 애플리케이션 요구사항을 최적 반영하기 위해서는, 글로벌 모델 뿐 만 아니라 **개인화·로컬 모델(locally adapted personalized model)의 성능을 최적화**할 수 있는 연합학습 기술 필요

Personalization 방식	특징
Adding User Context	▪ user clustering where similar clients are grouped together and a separate model is trained for each group.
Transfer Learning	▪ some or all parameters of a trained global model are re-learned on local data. ▪ To avoid the problem of catastrophic forgetting [21] [22], care must be taken to not retrain the model for too long on local data. A variant technique freezes the base layers of the global model and retrains only the top layers on local data. Transfer learning is also known as fine-tuning, and it integrates well into the typical federated learning lifecycle.
Multi-task Learning	▪ multiple related tasks are solved simultaneously allowing the model to exploit commonalities and differences across the tasks by learning them jointly
Meta-Learning	▪ **MAML builds an internal representation generally suitable for multiple tasks, so that fine tuning the top layers for a new task can produce good results. MAML proceeds in two connected stages: meta-training and meta-testing.** ➢ **Meta-training builds the global model on multiple tasks, and** ➢ **meta-testing adapts the global model individually for separate tasks.**
Knowledge Distillation	▪ extracting the knowledge of a large teacher network into a smaller student network by having the student mimic the teacher.
Base + Personalization Layers	▪ the base layers are trained centrally by Federated Averaging, and the top layers (also called personalization layers) are trained locally with a variant of gradient descent
Mixture of Global and Local Models	▪ Instead of learning a single global model, each device learns a mixture of the global model and its own local model.

Survey of Personalization Techniques for Federated Learning, https://arxiv.org/abs/2003.08673