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

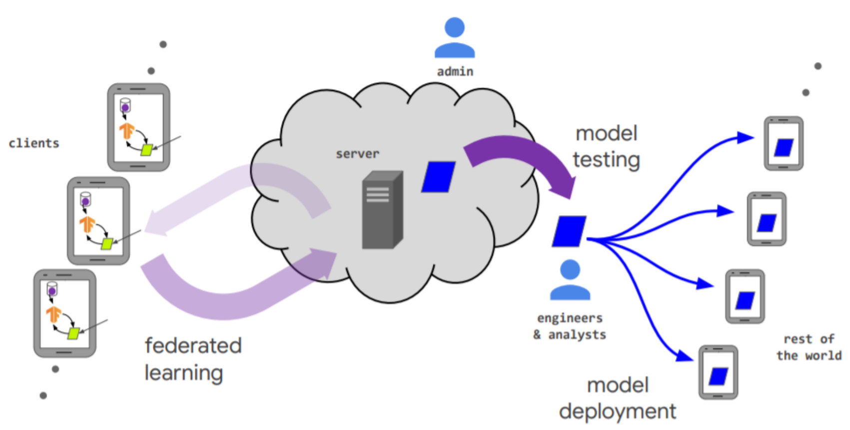
연합학습은 중앙 서버 또는 서비스 제공자의 관리 하에, 다수의 클라이언트/디바이스가 기계학습 문제를 해결하기 위해 협력하는 기술

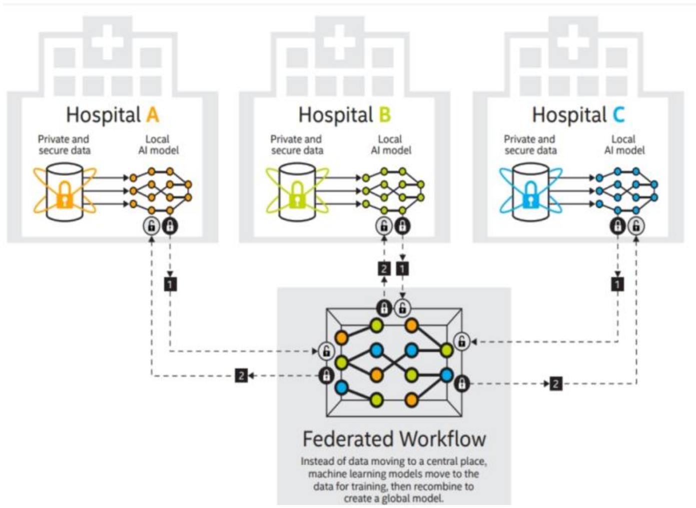
- 각 클라이언트/디바이스는 보유한/생산한 원시 데이터를 교환 또는 (중앙으로) 전송하지 않고, 로컬모델 학습에만 사용함으로써, 데이터 생산자의 프라이버시 보호
- 각 클라이언트/디바이스에서의 학습 결과는 (중앙의) 글로벌 모델 학습에 반영/기여. 'A fed B'학습의 성능은 'A+B'성능에 근사
- 데이터 생산자의 프라이버시 보호, 통신 오버헤드 감소

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





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**

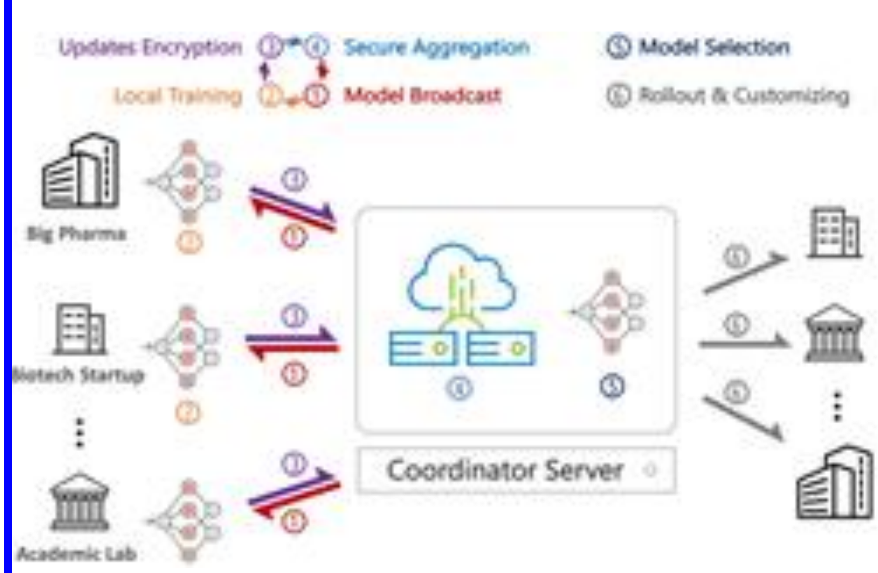
Advances and Open Problems in Federated Learning, https://arxiv.org/abs/1912.04977

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

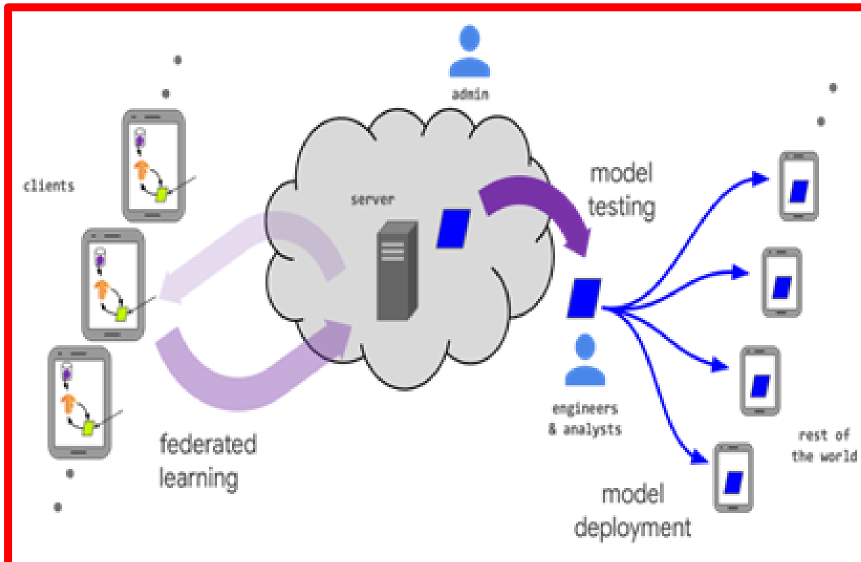
	Datacenter distributed learning	Cross-silo federated learning	Cross-device federated learning
Setting	Training a model on a large but "flat" dataset. Clients are compute nodes in a single cluster or datacenter.	Training a model on siloed data. Clients are different organizations (e.g. medical or financial) 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

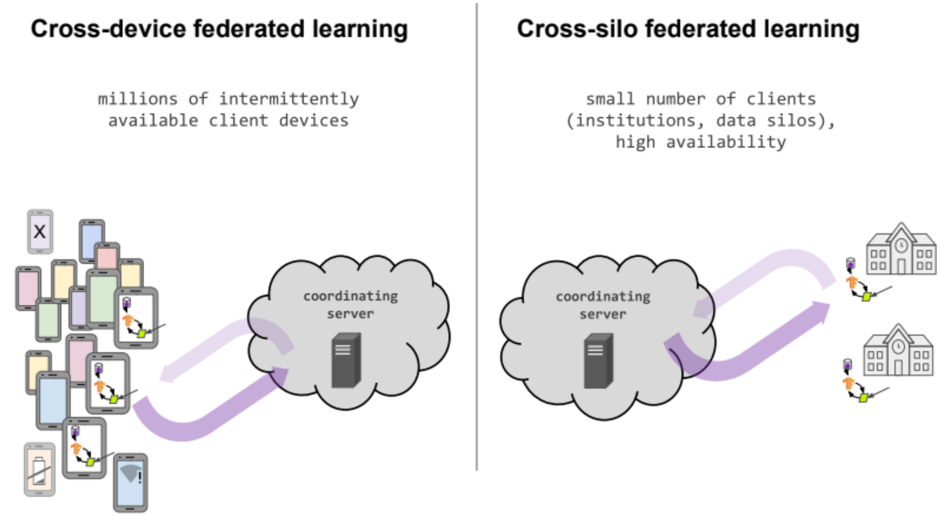
Applications of cross-device federating learning

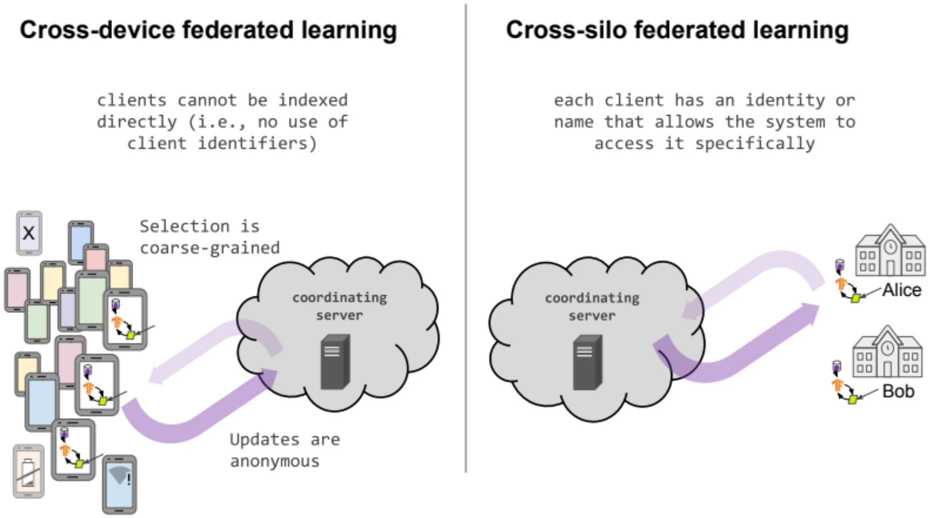
What makes a good application?

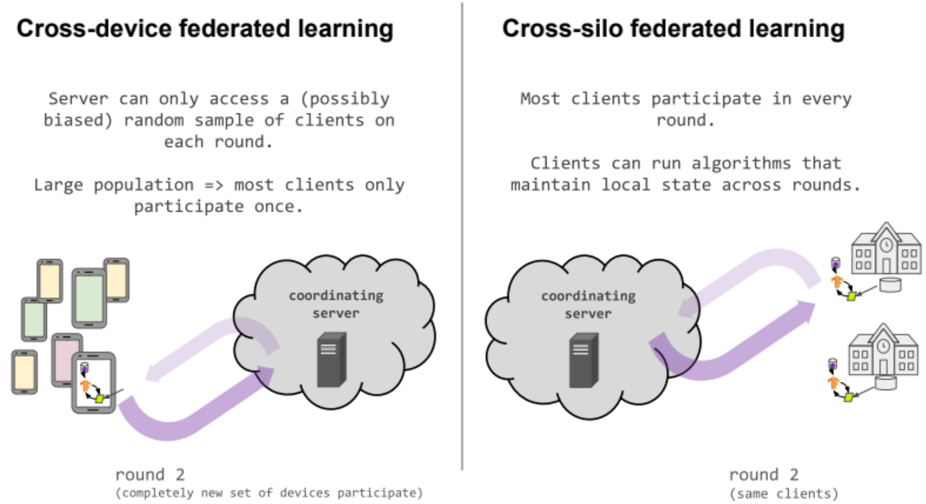
- On-device data is more relevant than server-side proxy data
- On-device data is privacy sensitive or large
- Labels can be inferred naturally from user interaction

Example applications

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





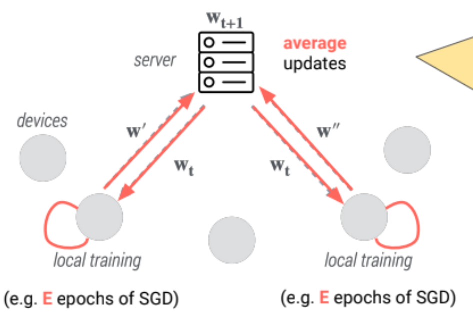


Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

연합학습 개요

A STANDARD BASELINE

Federated Averaging (FedAvg)



- 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`?

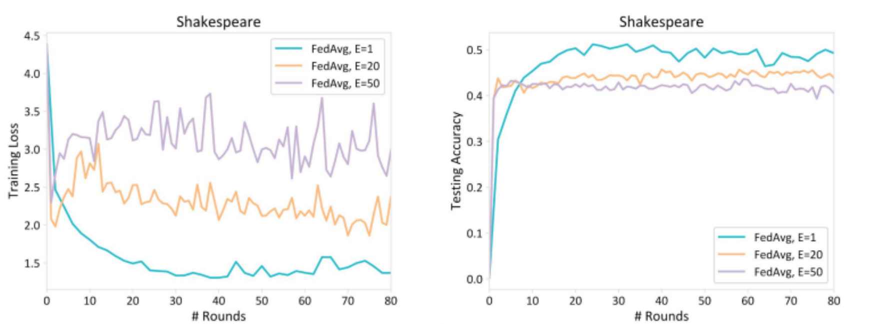
1. Can perform more local computation (i.e., more than just one mini-batch)
2. Incorporate updates more quickly (immediately apply gradient information)

✓ Can lead to method converging in many fewer communication rounds

✗ But, can potentially hurt convergence if not properly tuned …

WILL THIS CONVERGE?

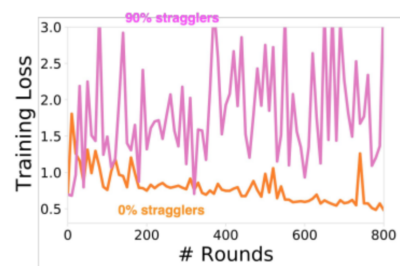
Challenge: heterogeneity



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

WILL THIS CONVERGE?

Challenge: heterogeneity

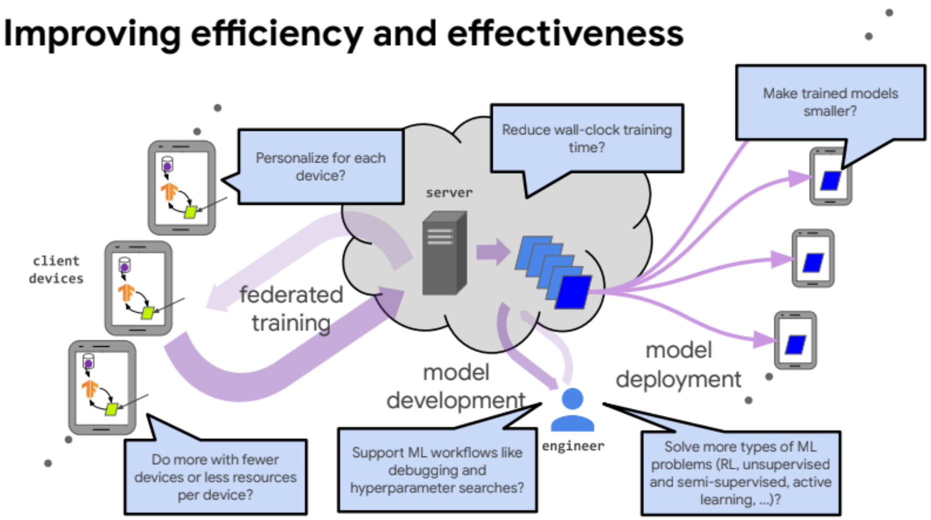


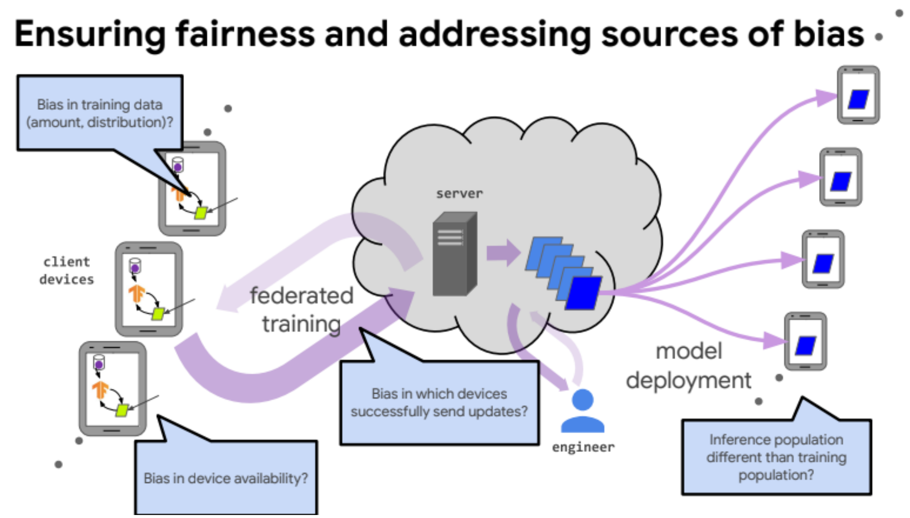
systems heterogeneity (e.g., dropping devices*) can exacerbate convergence issues

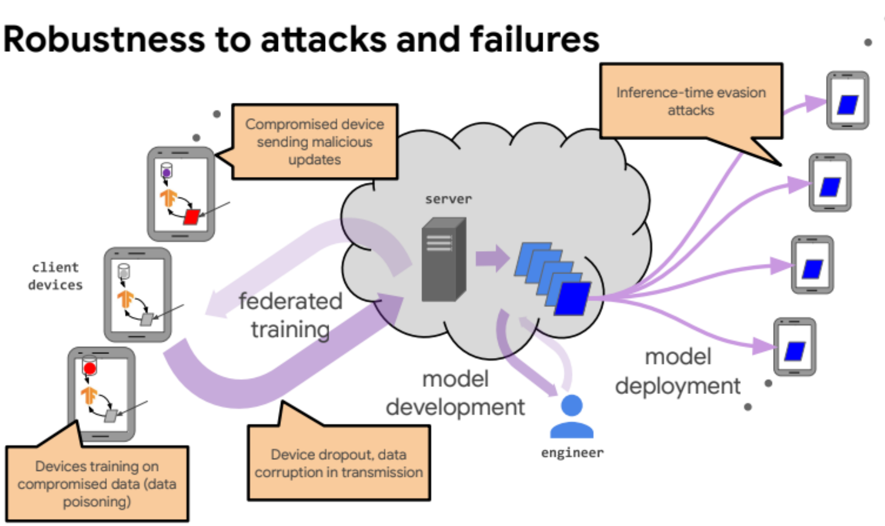
*[Bonawitz, et al. Towards Federated Learning at Scale: System Design, MLSys, 2019]
[Li et al, Federated optimization in heterogeneous networks, MLSys 2020]

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

연합학습 개요







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)

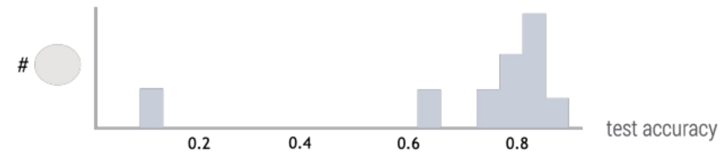
[Li et al, Fair Resource Allocation in Federated Learning, ICLR 2020]
*[Mohri, Sivek, Suresh, Agnostic Federated Learning, ICML 2019]
*[Hashimoto et al, Fairness without Demographics in Repeated Loss Minimization, ICML 2018]

FL: traditional empirical risk minimization

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

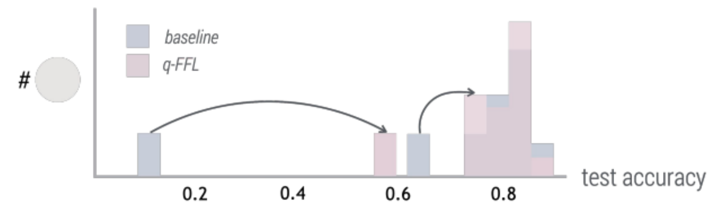
potential issues:

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



Fair resource allocation objective

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

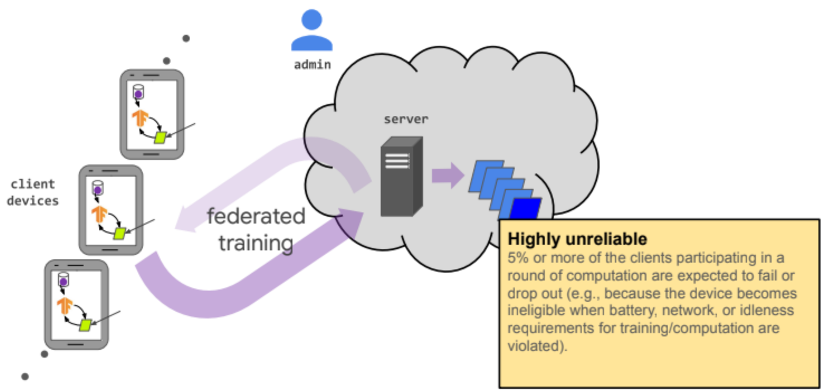


[Li et al, Fair Resource Allocation in Federated Learning, ICLR 2020]

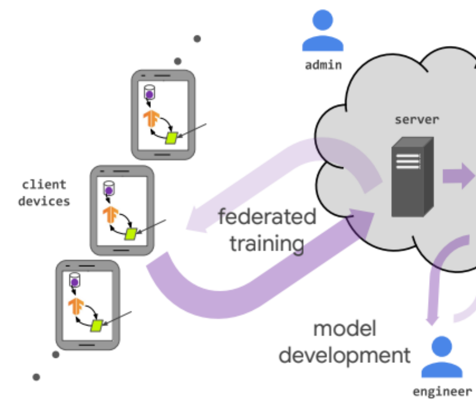
Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

연합학습 개요

System challenges in cross-device FL



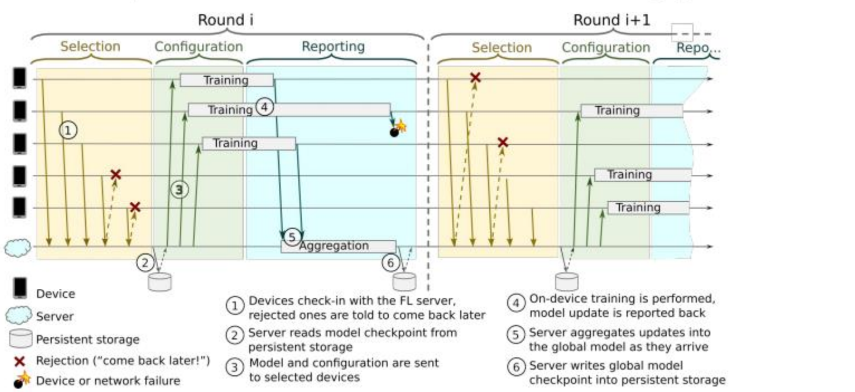
System challenges in cross-device FL



Operational Challenges

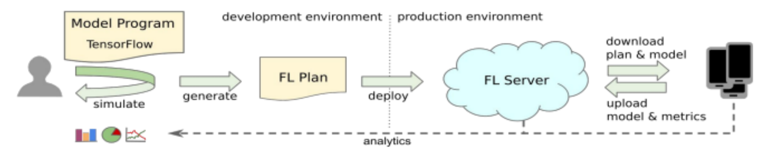
- Ensuring compatibility of deployed code (e.g. TensorFlow graphs) with multiple versions of the deployed runtime.
- Ensuring stability of user devices in all circumstances: no impact on performance, battery, or network usage.
- Providing a multi-tenant service used by multiple teams, each training many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

An example cross-device federated learning protocol



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

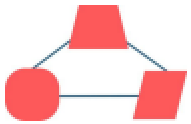
Federated Learning Tutorial@NeurIPS 2020, https://sites.google.com/view/fl-tutorial/

연합학습 개요

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

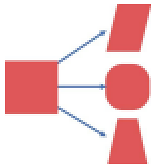
연합학습 개요

Approaches for personalization



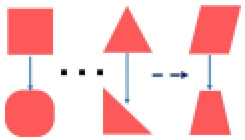
Multi-task learning

- 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

Open Platforms

2022_Fall

x

Flower

Flower, https://flower.dev/
A Friendly Federated Learning Framework, A unified approach to federated learning. Federate any workload, any ML framework, and any programming language.
Github: https://github.com/adap/flower
Slack: https://friendly-flower.slack.com/ssb/redirect
Summit: https://flower.dev/conf/flower-summit-2021

Flower: A Friendly Federated Learning Research Framework, https://arxiv.org/abs/2007.14390
Youtube: https://www.youtube.com/watch?v=t5WdERBPQfk&t=1s
On-device Federated Learning with Flower, https://arxiv.org/abs/2104.03042
Youtube: https://www.youtube.com/watch?v=QJEX5c0y1I8&t=2s

Flower use case

Scaling Flower with Multiprocessing, https://towardsdatascience.com/scaling-flower-with-multiprocessing-a0bc7b7aace0
Github: https://github.com/matturche/flower_scaling_example

Learn how to scale locally your Federated Learning experiments using the Flower framework and multiprocessing with PyTorch.

How to solve the issue:
This problem that you might have encountered, can be solved quite easily. Since the memory is not released until the process accessing it is released, then we simply need to encapsulate the part of our code that need to access the GPU in a sub-process, waiting for it to be terminated until we can continue to execute our program. Multiprocessing is the solution, and I will show you how to do it using PyTorch and Flower.

Differentially Private Federated Learning with Flower and Opacus,
https://towardsdatascience.com/differentially-private-federated-learning-with-flower-and-opacus-e14fb0d2d229
Github: https://github.com/matturche/flower_opacus_example

OpenFL

[intel]
OpenFL: An open-source framework for Federated Learning, https://arxiv.org/abs/2105.06413
Github: https://github.com/intel/openfl
YouTube: Federated Learning in Healthcare Use Cases | Intel Software,
https://www.youtube.com/watch?v=z5jJsvvfKbM
YouTube: SOSCON Russia 2021 [English]. Open리: Python library for Federated Learning. Olga Perepelkina, Intel, https://www.youtube.com/watch?v=Zso2oYsEgw0

인텔-펜실베니아大, 연합학습으로 환자 보안 유지하며 뇌종양 식별하는 AI 개발,
http://www.airtimes.kr/news/articleView.html?idxno=16331
https://www.apheris.com/blog-top7-open-source-frameworks-for-federated-learning

IBM Federated Learning, https://ibmfl.mybluemix.net/
Github: https://github.com/IBM/federated-learning-lib

OpenMined, https://www.openmined.org/
Github: https://github.com/OpenMined/PySyft

FedML: A Research Library and Benchmark for Federated Machine Learning, https://fedml.ai/
Github: https://github.com/FedML-AI

OpenFL

[intel] OpenFL: An open-source framework for Federated Learning, https://arxiv.org/abs/2105.06413

Abstract. Federated learning (FL) is a computational paradigm that enables organizations to collaborate on machine learning (ML) projects without sharing sensitive data, such as, patient records, financial data, or classified secrets.
Open Federated Learning (OpenFL) is an open-source framework for training ML algorithms using the data-private collaborative learning paradigm of FL. OpenFL works with training pipelines built with both TensorFlow and PyTorch, and can be easily extended to other ML and deep learning frameworks.
Here, we summarize the motivation and development characteristics of OpenFL, with the intention of facilitating its application to existing ML model training in a production environment.

Finally, we describe the first use of the OpenFL framework to train consensus ML models in a consortium of international healthcare organizations, as well as how it facilitates the first computational competition on FL.

Our ambition is that federations, such as the FeTS Initiative, will not serve as ad hoc collaborations for specific research efforts, but will serve as permanent networks for researchers in the healthcare, financial, industrial, and retail industries to more effectively train, deploy, monitor, and update their AI algorithms over time.

FeTS : https://www.fets.ai

OpenFL

[intel] OpenFL: An open-source framework for Federated Learning, https://arxiv.org/abs/2105.06413

3.2 First Computational Competition on Federated Learning

As the first challenge ever proposed for federated learning, the FeTS challenge 20212 intends to address these hurdles towards both the creation and the evaluation of tumor segmentation models. Specifically, the FeTS 2021 challenge uses clinically acquired, multi-institutional MRI scans from the BraTS 2020 challenge [18–20], as well as from various remote independent institutions included in the collaborative network of a real-world federation.

Federated Tumor Segmentation Challenge 2021, https://fets-ai.github.io/Challenge/

Compared to the BraTS challenge, the ultimate goal of the FeTS challenge is divided into the following two tasks:

1. Task 1 ("Federated Training") aims at effective weight aggregation methods for the creation of a consensus model given a pre-defined segmentation algorithm for training, while also (optionally) accounting for network outages.
2. Task 2 ("Federated Evaluation") aims at robust segmentation algorithms, given a pre-defined weight aggregation method, evaluated during the testing phase on unseen datasets from various remote independent institutions of the collaborative network of the fets.ai federation.

Nevermined

Nevermined, https://www.nevermined.io/

Nevermined is a data ecosystem solution that provides the capabilities of building bespoke networks where different entities can share and monetize their data and make an efficient and secure usage of it even with untrusted parties.

Github: https://github.com/nevermined-io
Docs: https://docs.nevermined.io/
Blog: https://docs.nevermined.io/Blog/

YouTube: Leveraging blockchain to unlock data for federated learning, https://www.youtube.com/watch?v=A0A9hSlPhKI
Description: This demo we will be using Flower and Nevermined with the goal of the model to train classify images given two different datasets.

Data Monetization for Enterprises, https://multimedia.getresponse.com/getresponse-ylcbE/documents/12e30a17-5321-49cd-a613-963451401b07.pdf

https://www.nevermined.io/solutions/details/data-sharing#federated-learning
https://www.nevermined.io/solutions/details/data-governance#incentives

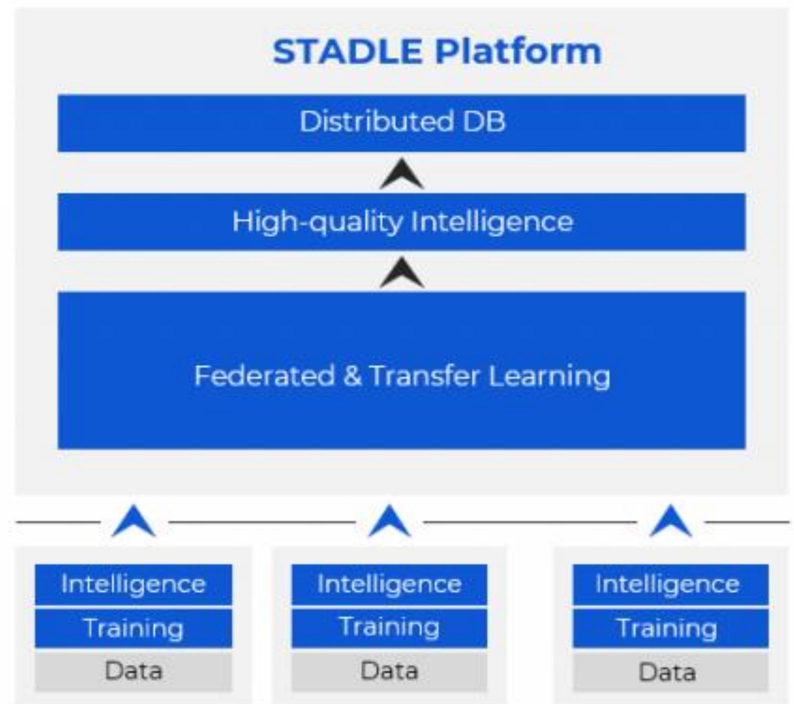
- One of the most powerful features of blockchain technology is the ability to issue incentives, rewards, and penalties for specific activity. In Bitcoin, the incentive manifests as block rewards for miners operating computers with the largest computational power. This ability to bake in incentives that promote participant behavior provides a fundamental shift in how digital ecosystems operate, including how they are monetized as well as governed.

STADLE

STADLE: https://www.stadle.ai/
TieSet Website: https://tie-set.com/
Doc/Install: https://stadle-documentation.readthedocs.io/en/latest/overview.html

Introduction to TieSet, https://www.youtube.com/watch?v=Nk9gdsFBKGs
YouTube: https://www.youtube.com/channel/UCv3NW3foNBRv12q-ymKa37A

Before (Typical Big Data Companies)
Must upload full data to generate AI model, can't work in offline mode.



After (TieSet)
Able to generate AI models on local devices, no need to upload data, works in offline mode.

STADLE

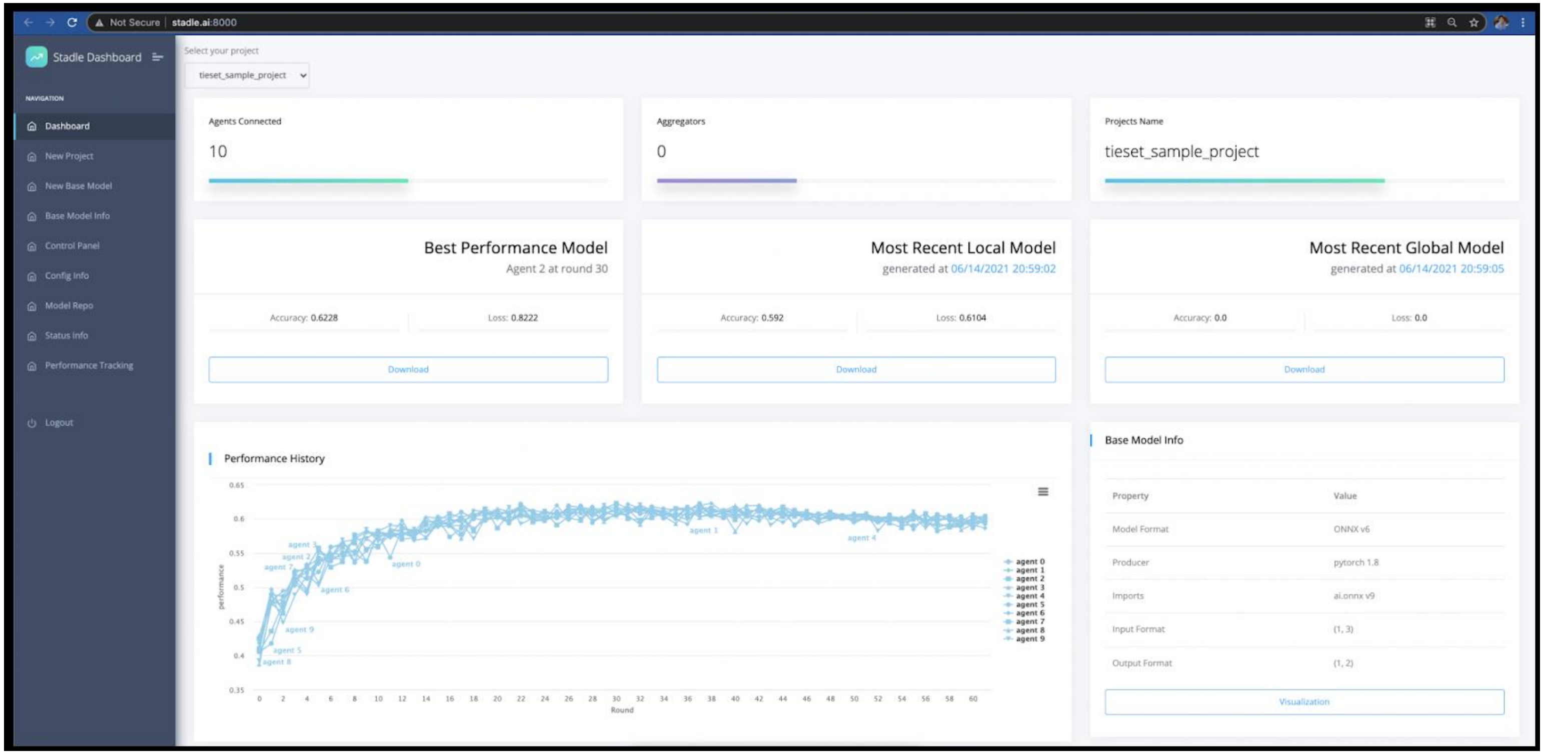
STADLE: https://www.stadle.ai/
TieSet Website: https://tie-set.com/
Doc/Install: https://stadle-documentation.readthedocs.io/en/latest/overview.html

Introduction to TieSet, https://www.youtube.com/watch?v=Nk9gdsFBKGs
YouTube: https://www.youtube.com/channel/UCv3NW3foNBRv12q-ymKa37A
TieSet Team 1 min intro, https://www.youtube.com/watch?v=vURWKP1jrv0

STADLE

STADLE: https://www.stadle.ai/
TieSet Website: https://tie-set.com/
Doc/Install: https://stadle-documentation.readthedocs.io/en/latest/overview.html

News: https://re-how.net/all/1380746/
Now that we have completed the development of the basic functions for commercialization, we have decided to carry out a private release. With this release, further functional improvements and commercialization of this platform To promote this, we are looking for a partner who can carry out both technical verification and verification verification.



PowerFlow

PowerFlow: https://integrate.ai/powerflow/
Design a federated learning system in seven steps, https://integrate.ai/blog/design-a-federated-learning-system-in-seven-steps-pftl/

Etc.

Federated Learning in Heterogeneous Environments,
https://www.youtube.com/watch?v=651VFm2vIhA

FL Applications

FL Applications

[Recommendation]

[IoT]
Federated Learning for Internet of Things: A Federated Learning Framework for On-device Anomaly Data Detection, https://arxiv.org/abs/2106.07976
Deep Anomaly Detection for Time-series Data in Industrial IoT: A Communication-Efficient On-device Federated Learning Approach, https://arxiv.org/abs/2007.09712
Privacy Preserving Federated Learning Solution for Security of Industrial Cyber Physical Systems, https://link.springer.com/chapter/10.1007/978-3-030-76613-9_11
A Survey on Federated Learning and its Applications for Accelerating Industrial Internet of Things, https://arxiv.org/abs/2104.10501

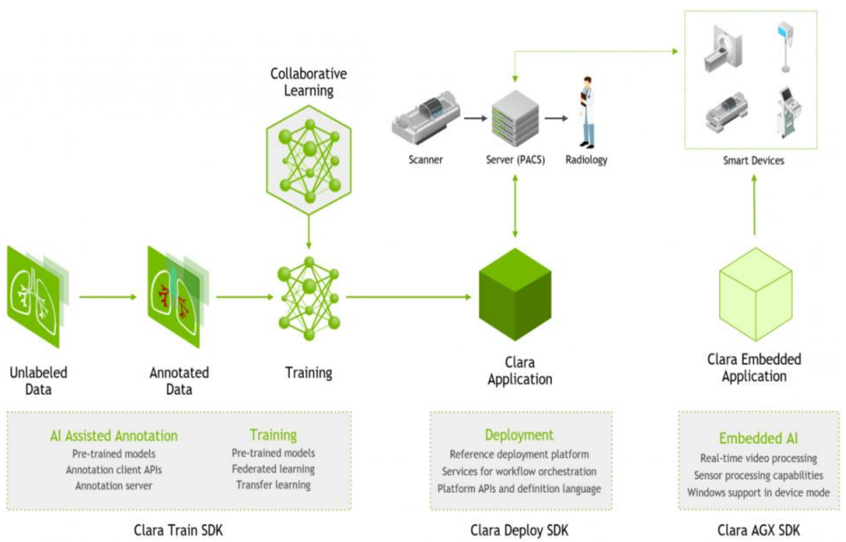
[Health]
FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare, https://arxiv.org/abs/1907.09173
Federated Learning for Healthcare Informatics, https://link.springer.com/article/10.1007/s41666-020-00082-4
The future of digital health with federated learning, https://www.nature.com/articles/s41746-020-00323-1
Privacy-first Health Research with Federated Learning, https://www.nature.com/articles/s41746-021-00489-2 https://research.google/pubs/pub50116/

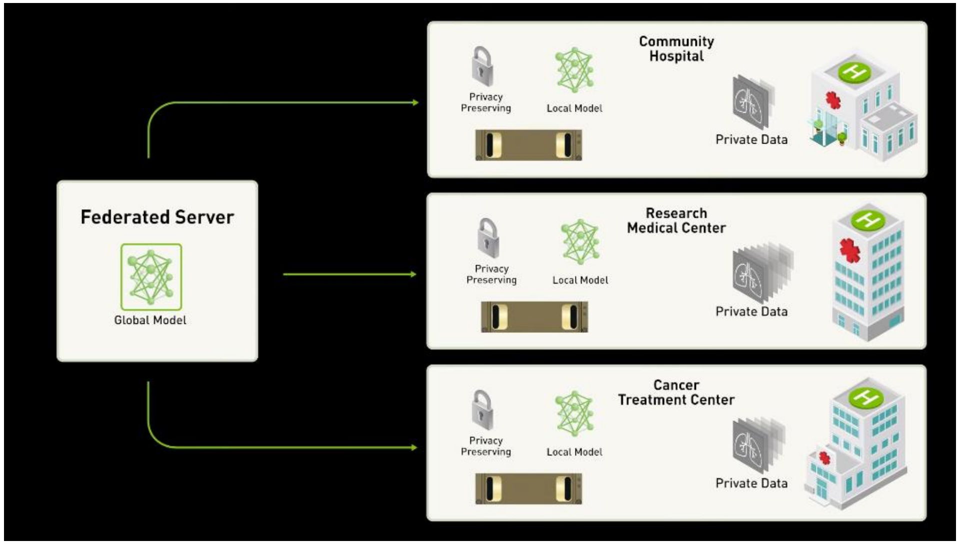
https://sooyongshin.wordpress.com/2020/11/22/federated-learning/
Reliability and Performance Assessment of Federated Learning on Clinical Benchmark Data, https://arxiv.org/abs/2005.11756
Federated Learning on Clinical Benchmark Data: Performance Assessment, https://www.jmir.org/2020/10/e20891

Federated Learning powered by NVIDIA Clara

- An Application Framework Optimized for Healthcare and Life Sciences Developers, https://developer.nvidia.com/clara
- Federated Learning powered by NVIDIA Clara, https://developer.nvidia.com/blog/federated-learning-clara/

- Transforming AI Healthcare with Federated Learning, https://news.developer.nvidia.com/transforming-ai-healthcare-with-federated-learning/
- https://www.nature.com/articles/s41746-020-00323-1





Google Health Studies

Advancing health research with Google Health Studies,
https://blog.google/technology/health/google-health-studies-app/
https://play.google.com/store/apps/details?id=com.google.android.apps.health.research.studies

COVID-19 has highlighted the importance of research in providing information about disease and treatments. However, it's challenging for researchers to recruit enough volunteers so that studies are representative of the general population. To make it easier for leading research institutions to connect with potential study participants, we're introducing the Google Health Studies app with the first study focused on respiratory illness.

Keeping participant data private, safe and secure
Studying respiratory illnesses
We've partnered with researchers from Harvard Medical School and Boston Children's Hospital for the first study, which will help scientists and public health communities better understand respiratory illnesses, including influenza and COVID-19.

- This Respiratory Health Study will be open to adults in the U.S., and will focus on identifying how these types of illnesses evolve in communities and differ across risk factors such as age, and activities such as travel.
- Study participants will use the Google Health Studies app to regularly self-report how they feel, what symptoms they may be experiencing, any preventative measures they've taken, and additional information such as COVID-19 or influenza test results. By taking part in this study, volunteers can represent their community in medical research, and contribute to global efforts to combat the COVID-19 pandemic.

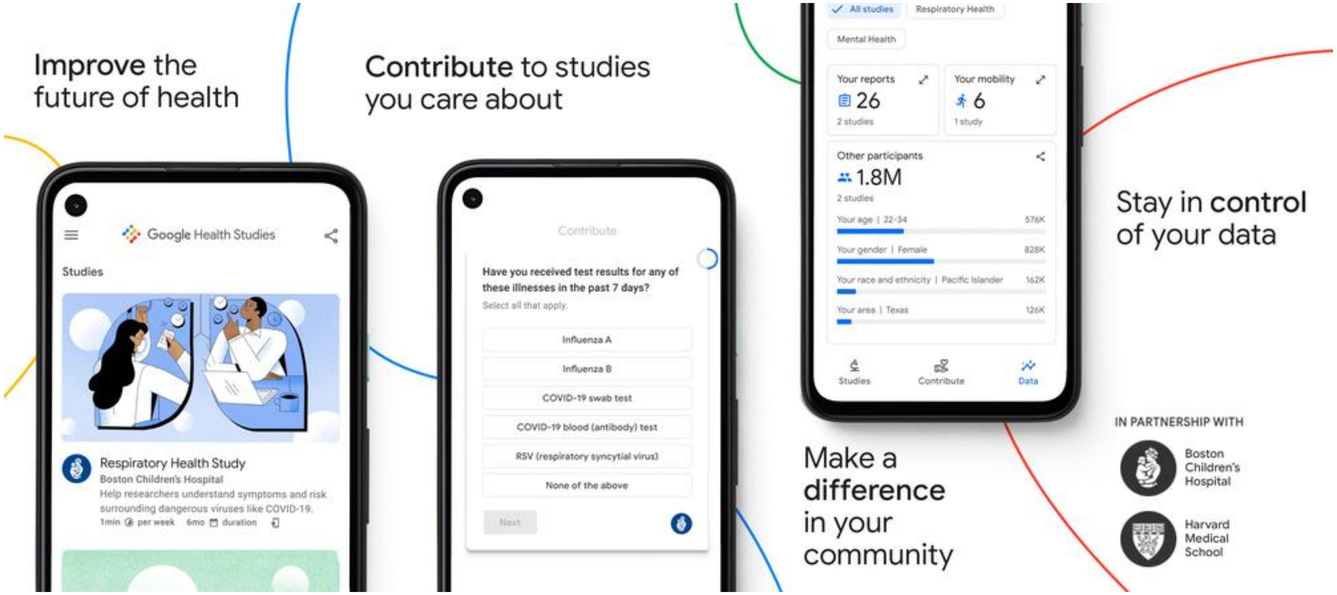
In collaboration with Google Research, this first study utilizes federated learning and analytics—a privacy technology that keeps a person's data stored on the device, while allowing researchers to discover aggregate insights based on encrypted, combined updates from many devices.

Google Health Studies : 연합학습 적용

Blog: https://blog.google/technology/health/google-health-studies-app/
App: https://play.google.com/store/apps/details?id=com.google.android.apps.health.research.studies

The Google Health Studies app is now available in the Google Play Store, and we're inviting people to download the app to join this initial study. We look forward to partnering with health researchers and to making it possible for more people to participate in these important studies.

… this first study utilizes federated learning and analytics—a privacy technology that keeps a person's data stored on the device, while allowing researchers to discover aggregate insights based on encrypted, combined updates from many devices. This means researchers in this study can examine trends to understand the link between mobility (such as the number of daily trips a person makes outside the home) and the spread of COVID-19, This same approach powers typing predictions on Gboard, without Google seeing what individuals type.



Google Health Studies : 연합학습 적용

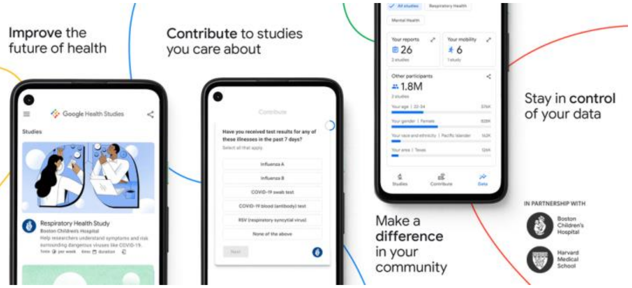
Blog: https://blog.google/technology/health/google-health-studies-app/
App: https://play.google.com/store/apps/details?id=com.google.android.apps.health.research.studies

Google Health Studies lets you securely contribute to health research studies with leading institutions, right from your phone. Volunteer for studies that matter to you and represent your community.

Simply download the app and enroll in a study.

Help researchers make advancements in medicine and healthcare:

- Self-report symptoms and other data
- Volunteer for multiple studies in one app
- Track your information with digital health reports
- Learn research findings from the studies you participate in



Help scientists better understand respiratory diseases.
The first study available is a respiratory health study conducted by Boston Children's Hospital and Harvard Medical School. If you participate in this study, you'll provide data to help researchers understand how demographics, health history, behavior, and mobility patterns contribute to the spread of respiratory illnesses. Upcoming studies will research mental health and diabetes.

You're in control of your data: In the respiratory health study, your personal information is kept on your device. Researchers only see aggregated study data combined from all participants. This allows researchers to collect the information needed to advance the study without seeing individual details.

Your input matters: Google Health Studies aims to create opportunities for more people to participate in health research. By contributing, you'll represent your community and start improving the future of health for everyone.

Google Health Studies

- https://health.google/for-everyone/health-studies/
- App: https://play.google.com/store/apps/details?id=com.google.android.apps.health.research.studies
- Blog: https://blog.google/technology/health/google-health-studies-app/

Benefit the public, in private

Protecting your information in the respiratory health study.

Your study data stays on your device

After joining a health study, you'll begin completing weekly surveys. At all times, your individual survey responses, location history and other personally identifiable data stays on your device.

Your device computes statistics based on your study data

During the study, your device receives different queries, computes and summarizes the results based on your individual study data, and encrypts these results for subsequent aggregation with federated analytics.

Participant data gets aggregated

Encrypted summaries from many devices are combined together, using the federated analytics technology. Google and study partners do not receive any individual study data about you.

Research that values your privacy

Combined insights are sent securely to the researchers conducting the study. You can safely contribute to health research knowing your personally identifiable study data will never be available to Google or third parties.

Google Health Studies

Paper: Privacy-first Health Research with Federated Learning, https://www.nature.com/articles/s41746-021-00489-2,
Patent: Privacy-First On-Device Federated Health Modeling and Intervention, https://patents.google.com/patent/US20210090750A1/en

We show—on a diverse set of single and multi-site health studies—that federated models can achieve similar accuracy, precision, and generalizability, and lead to the same interpretation as standard centralized statistical models while achieving considerably stronger privacy protections and without significantly raising computational costs.

At this point, however, only specific large homogenous units of federation, such as at the level of a healthcare system, have been studied in detail in prior work, and the focus has been on traditional classification tasks.

Specifically, health study data is typically non-IID—not independent and identically distributed—which is compounded by the fact that in the federated regime, individual data points are distributed across many devices that participate asynchronously.

This work's primary focus is on cross-device (cross-patient) settings, where the unit of federation is a single individual.

By contrast, in this work, we focus on those scenarios commonly found in epidemiological health studies, specifically studies with many participants, each of whom has relatively small amounts of non-IID, labeled data. The approach described here can be appropriate for health studies involving smartphone/wearable data and virtual clinical studies (also called decentralized clinical studies) that directly recruit individual research participants without relying on clinical sites for recruitment.

Google Health Studies

Paper: Privacy-first Health Research with Federated Learning, https://www.nature.com/articles/s41746-021-00489-2,

Study Topic	Sample Results	Comparison Metric	Traditional Centralized Model[a]	Federated Replications	
				Per-Patient	Per-Silo[b]
Heart Failure	1. Survival Prediction (full model) 2. Survival Prediction (with variable selection)	AUC	0.82 0.82	0.85 0.83	N/A
Diabetes	1. Diabetes prediction at 5-years	AUC	0.84	0.875	N/A
MIMIC-III	1. Inpatient mortality prediction	AUC	0.780± 0.012	0.777 ± 0.011	0.777 ± 0.014
SARS-CoV-2	1. CV2+ve in Female vs. Male 2. CV2+ve in Recent vs. Never Cancer	OR	0.35 (0.32–0.38) 1.88 (1.36–2.60)	0.35 (0.32–0.38) 1.99 (1.45–2.68)	0.35 (0.32–0.38) 2.07 (1.50–2.86)
Avian Influenza	1. Fatality with each day before hospitalization 2. Fatality in Indonesia vs. group of countries	OR	1.33 (1.11–1.60) 0.23 (0.04–1.27)	1.34 (1.12–1.61) 0.25 (0.05–1.37)	1.33 (1.11–1.60) 0.24 (0.04–1.33)
Bacteraemia	1. Relapse with line-associated infection source 2. Relapse with presence of immunosuppression	Coefficient	1.57 (SE: 0.45) 1.07 (SE: 0.41)	1.59 (SE: 0.23) 1.12 (SE: 0.30)	N/A
Azithromycin	1. Adverse events in azithromycin treated	Coefficient	−0.11 (SE: 0.09)	−0.29 (SE: 0.19)	N/A
Tuberculosis	1. Extrapulmonary TB in individuals with HIV	Coefficient	1.16 (SE: 0.09)	1.35 (SE: 0.08)	0.15 (SE: 0.07)[c]

Google Health Studies

Patent: Privacy-First On-Device Federated Health Modeling and Intervention, https://patents.google.com/patent/US20210090750A1/en

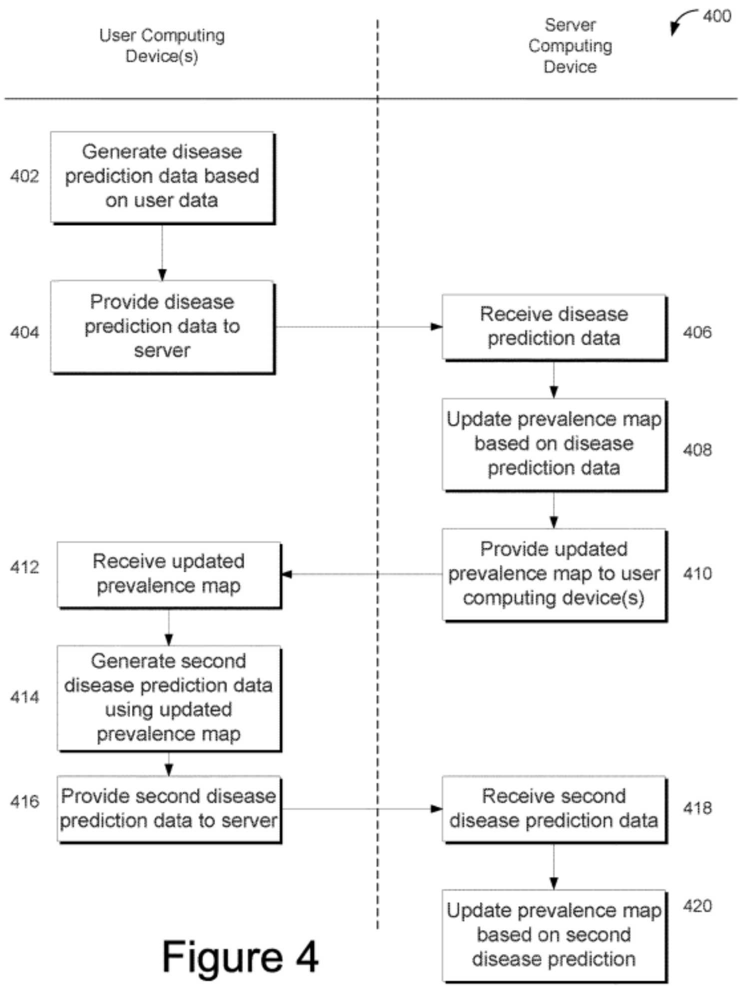


Figure 4

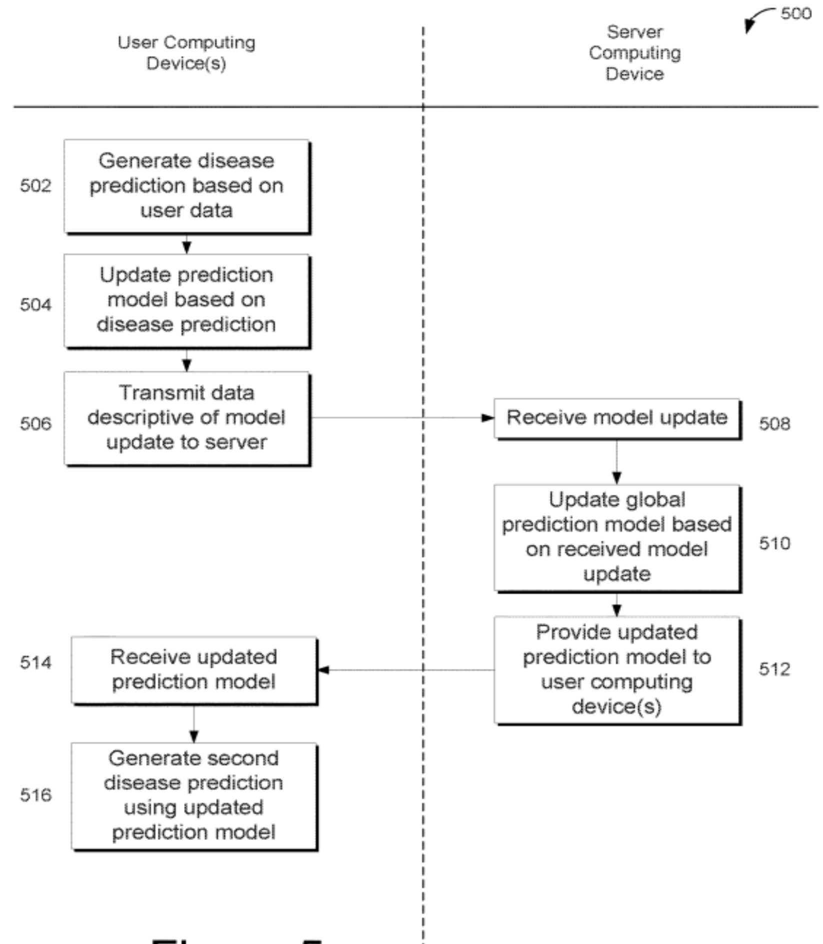


Figure 5

Google Health Studies

Wide Scale Monitoring for Acute Respiratory Infection Using a Mobile-Based Study Platform, https://clinicaltrials.gov/ct2/show/results/NCT04663776

- Sponsor: Boston Children's Hospital
- Collaborator: Google LLC.
- Information provided by (Responsible Party): John Brownstein, Boston Children's Hospital

Brief Summary:
This is a prospective observational study using a mobile study platform (app) that is designed for use on Android phones.

- Study participants will provide baseline demographic and medical information and report symptoms of respiratory infection on a weekly basis using the app.
- Participants will also report use of prevention techniques on the weekly survey.
- Mobility data will be collected passively using the sensors on the participant's smartphone, if the participant has granted the proper device permissions.
- The overall goals of the study are to track spread of coronavirus-like illness (CLI), influenza-like illness (ILI) and non-specific respiratory illness (NSRI) on a near-real time basis and identify specific behaviors associated with an increased or decreased risk of developing these conditions.

Study Population
The study population will be adult Android mobile device users who live within the United States.

MAML with FL

Federated Optimization

A Field Guide to Federated Optimization, https://arxiv.org/abs/2107.06917
Advances and Open Problems in Federated Learning, https://arxiv.org/abs/1912.04977

This paper provides recommendations and guidelines on formulating, designing, evaluating and analyzing federated optimization algorithms through concrete examples and practical implementation, with a focus on conducting effective simulations to infer real-world performance.

The goal of this work is not to survey the current literature, but to inspire researchers and practitioners to design federated learning algorithms that can be used in various practical applications.

Personalization and multi-task learning
In personalization, every client is allowed to have a different model that is adapted to their local data (i.e., a personalized model).

- One approach to learning a personalized model is to train a global model and use meta-learning to refine it and obtain personalized models [49, 78, 129, 158].

[49] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. arXiv preprint arXiv:1802.07876, 2018. FedMeta
[78] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. In Advances in Neural Information Processing Systems, 2020. Per-FedAvg
[129] Yihan Jiang, Jakub Koneˇcn´y, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. arXiv preprint arXiv:1909.12488, 2019. Personalized FedAvg

- Another line of work uses multi-task learning [70, 77, 101, 164, 232] to regularize local models towards the global average or towards some reference point. Section 7.5 provides additional detail and discussion of personalized models.

Federated Optimization

A Field Guide to Federated Optimization, https://arxiv.org/abs/2107.06917

7.5 Personalization

The idea that every client gets a good model for its own data not only improves the overall statistical performance, but also potentially improve fairness or robustness of private algorithms [164, 290].

Before discussing the specific personalization algorithms (e.g., multi-task learning, clustering, fine-tuning, and meta-learning), we give two general categories of personalization algorithms:

- Algorithms that require client-side state or identifier (Stateful) : A popular technique used in this category is multi-task learning.
- Algorithms that do not require client-side state or identifier (Stateless) : A popular technique used in this category is meta-learning.

- In contrast to the first category, algorithms in this category do not require the server to know the client's identifier; the clients also do not need to carry a state from the previous round to the next round. This makes these algorithms more attractive in the cross-device setting, where the population size is huge (e.g., millions of devices), only a small number of clients (e.g., a few hundreds) participate in each round, and a device usually participates only once during the entire training process.
- In the federated setting, every client can be treated as a different task, and the goal is to meta-learn a learning algorithm that can generalize to unseen clients.
- Splitting the entire model into a shared part and a local part is another natural approach to personalization.

FedMeta

Federated Meta-Learning with Fast Convergence and Efficient Communication, https://arxiv.org/abs/1802.07876

We show that meta-learning is a natural choice for federated setting and propose a novel federated meta-learning framework named FedMeta that incorporates the meta-learning algorithms with federated learning.

Initialization based meta-learning algorithms like MAML [5] are well known for rapid adaptation and good generalization to new tasks, which makes it particularly well-suited for federated setting where the decentralized training data is non-IID and highly personalized.

Our work bridges the meta-learning methodology and federated learning.
In meta-learning, a parameterized algorithm (or meta-learner) is slowly learned from a large number of tasks through a meta-training process, where a specific model is fast trained by the algorithm in each task.

- A task typically consists of a support set and a query set that are disjoint from each other. A task-specific model is trained on the support set and then tested on the query set, and the test results are used to update the algorithm.

By contrast, in federated meta-learning, an algorithm is maintained on the server, and is distributed to the clients for model training.

- In each episode of meta-training, a batch of sampled clients receives the parameters of the algorithm and performs model training. Test results on the query set are then uploaded to the server for algorithm update.

we apply FedMeta to an industrial recommendation task where each client has highly personalized records, and experimentally show that meta-learning algorithms achieve higher accuracies for recommendation tasks than federated or stand-alone recommendation approaches.

FedMeta

Federated Meta-Learning with Fast Convergence and Efficient Communication, https://arxiv.org/abs/1802.07876

2 Related Work

Initialization Based Meta-Learning. In meta-learning, the goal is to learn a model on a collection of tasks, such that it can solve new tasks with only a small number of samples [4]. As one promising direction to meta-learning, initialization based methods has recently demonstrated effectiveness by "learning to fine-tune".

- Another approach aims to learn a good model initialization [5, 12, 17, 16], such that the model has maximal performance on a new task with limited samples after a small number of gradient descents. All of the work mentioned above only explore the setting where the tasks have a unified form (e.g., 5-way 5-shot for image classification).
- In this work, we fill this gap by studying meta-learning algorithms on real-world federated datasets. We focus our attention on model initialization methods where the algorithms are model- and task-agnostic and can be deployed out of the box, as the tasks and models in the federated setting vary. To the best of our knowledge, our proposed framework is the first to explore the federated setting from the meta-learning perspective.

Federated Learning.

- Similar to [23], the federated meta-learning framework proposed by us treats each client as a task. Instead of training a global model that ingests all tasks, we aim to train a well-initialized model that can achieve rapid adaptation to new tasks.
- The intuition behind meta-learning algorithms is to extract and propagate internal transferable representations of prior tasks. As a result, they can prevent overfitting and improve generalization on new tasks, which shows the potential in handling the statistical and systematic challenges of federated setting.

FedMeta

Federated Meta-Learning with Fast Convergence and Efficient Communication, https://arxiv.org/abs/1802.07876

Algorithm 1: FedMeta with MAML and Meta-SGD

1 // Run on the server
2 **AlgorithmUpdate:**
3 Initialize θ for MAML, or initialize (θ, α) for Meta-SGD.
4 **for** each episode $t = 1, 2, \ldots$ **do**
5   Sample a set U_t of m clients, and distribute θ (for MAML) or (θ, α) (for Meta-SGD) to the sampled clients.
6   **for** each client $u \in U_t$ in parallel **do**
7   Get test loss $g_u \leftarrow$ ModelTrainingMAML(θ) or $g_u \leftarrow$ ModelTrainingMetaSGD(θ, α)
8   **end**
9   Update algorithm paramters $\theta \leftarrow \theta - \frac{\beta}{m} \sum_{u \in U_t} g_u$ for MAML or $(\theta, \alpha) \leftarrow (\theta, \alpha) - \frac{\beta}{m} \sum_{u \in U_t} g_u$ for Meta-SGD.
10 **end**

11 // Run on client u
12 **ModelTrainingMAML(θ):**
13 Sample support set D_S^u and query set D_Q^u
14 $\mathcal{L}_{D_S^u}(\theta) \leftarrow \frac{1}{|D_S^u|} \sum_{(x,y) \in D_S^u} \ell(f_\theta(x), y)$
15 $\theta_u \leftarrow \theta - \alpha \nabla \mathcal{L}_{D_S^u}(\theta)$
16 $\mathcal{L}_{D_Q^u}(\theta_u) \leftarrow \frac{1}{|D_Q^u|} \sum_{(x',y') \in D_Q^u} \ell(f_{\theta_u}(x'), y')$
17 $g_u \leftarrow \nabla_\theta \mathcal{L}_{D_Q^u}(\theta_u)$
18 Return g_u to server

ModelTrainingMetaSGD(θ, α):
Sample support set D_S^u and query set D_Q^u
$\mathcal{L}_{D_S^u}(\theta) \leftarrow \frac{1}{|D_S^u|} \sum_{(x,y) \in D_S^u} \ell(f_\theta(x), y)$
$\theta_u \leftarrow \theta - \alpha \circ \nabla \mathcal{L}_{D_S^u}(\theta)$
$\mathcal{L}_{D_Q^u}(\theta_u) \leftarrow \frac{1}{|D_Q^u|} \sum_{(x',y') \in D_Q^u} \ell(f_{\theta_u}(x'), y')$
$g_u \leftarrow \nabla_{(\theta,\alpha)} \mathcal{L}_{D_Q^u}(\theta_u)$
Return g_u to server

The algorithm Aϕ is in general parameterized, where its parameter ϕ is updated in the meta-training process using a collection of tasks.

- Line 13: A task T in meta-training consists of a support set DT S = {(xi , yi)} |DT S | i=1 and a query set DT Q = {(x 0 i , y0 i)} |DT Q| i=1 , both of which contain labeled data points.
- Line 14-15: The algorithm A trains a model f on the support set DT S and outputs parameter θT , which we call inner update
- Line 16-17: The model fθT is then evaluated on the query set DT Q, and some test loss LDT Q (θT) is computed to reflect the training ability of Aϕ.
- Line 9: Finally, Aϕ is updated to minimize the test loss, which we call outer update.
- Note that the support and query sets are disjoint to maximize the generalization ability of Aϕ.

FedMeta

Federated Meta-Learning with Fast Convergence and Efficient Communication, https://arxiv.org/abs/1802.07876

We incorporate meta-learning into the federated learning framework. The goal is to collaboratively meta-train an algorithm using data distributed among clients.

- Taking MAML as a running example, we aim to train an initialization for the model by using all clients' data together.
- Recall that MAML contains two levels of optimization: the inner loop to train task-specific models using the maintained initialization, and the outer loop to update the initialization with the tasks' test loss.
- In the federated setting, each client u retrieves the initialization θ from the server, trains the model using a support set Du S of data on device, and sends test loss LDu Q (θ) on a separate query set Du Q to the server.
- The server maintains the initialization, and updates it by collecting test losses from a mini batch of clients. The transmitted information in this process consists of the model parameter initialization (from server to clients) and test loss (from clients to server), and no data is required to be collected to the server.
- The algorithm is maintained in the AlgorithmUpdate procedure. In each round of update, the server calls ModelTrainingMAML or ModelTrainingMeta-SGD on a set of sampled clients to gather test losses. To deploy the model on client u after meta-training, the initialization θ is updated using the training set of u, and the obtained θu is used to make predictions.