Federated Learning

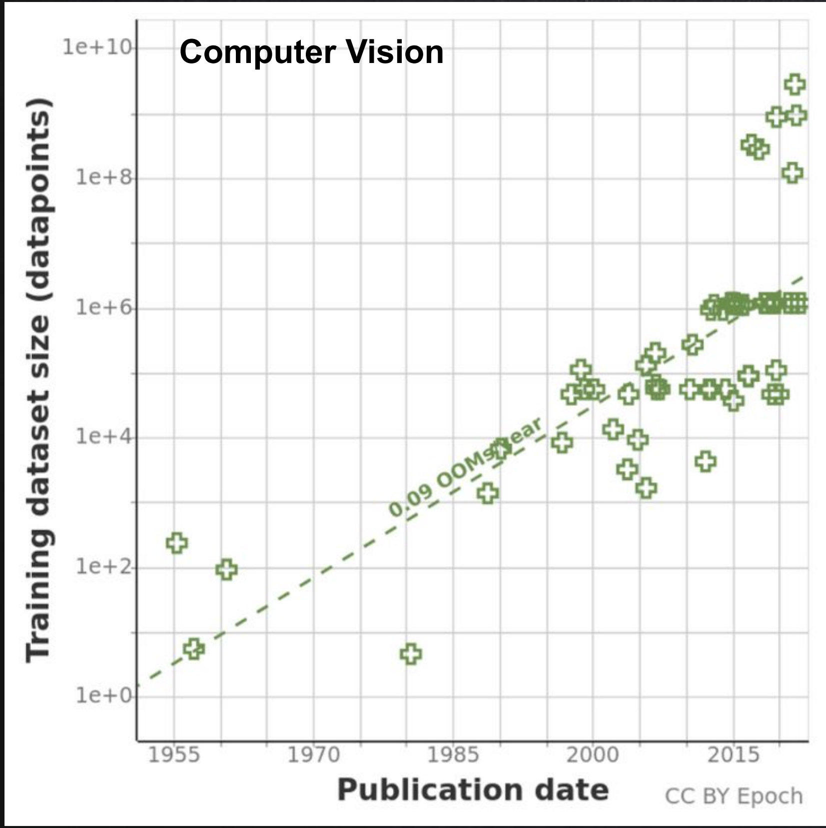
All for One and One for All

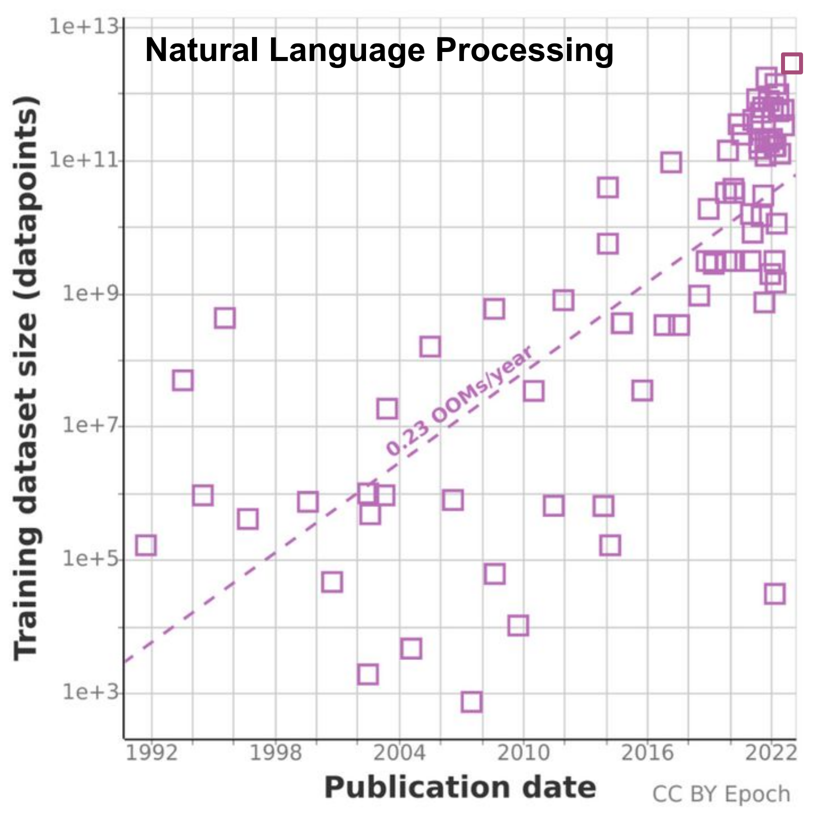
Christoph Lampert

Institute of Science and Technology Austria

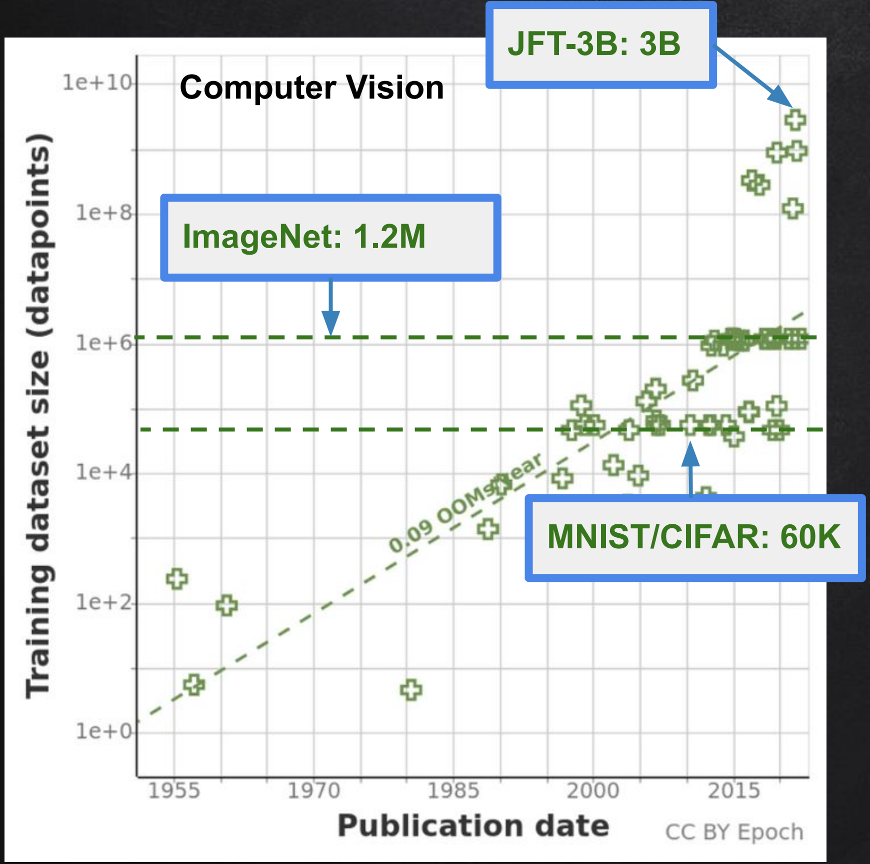
September 21, 2023

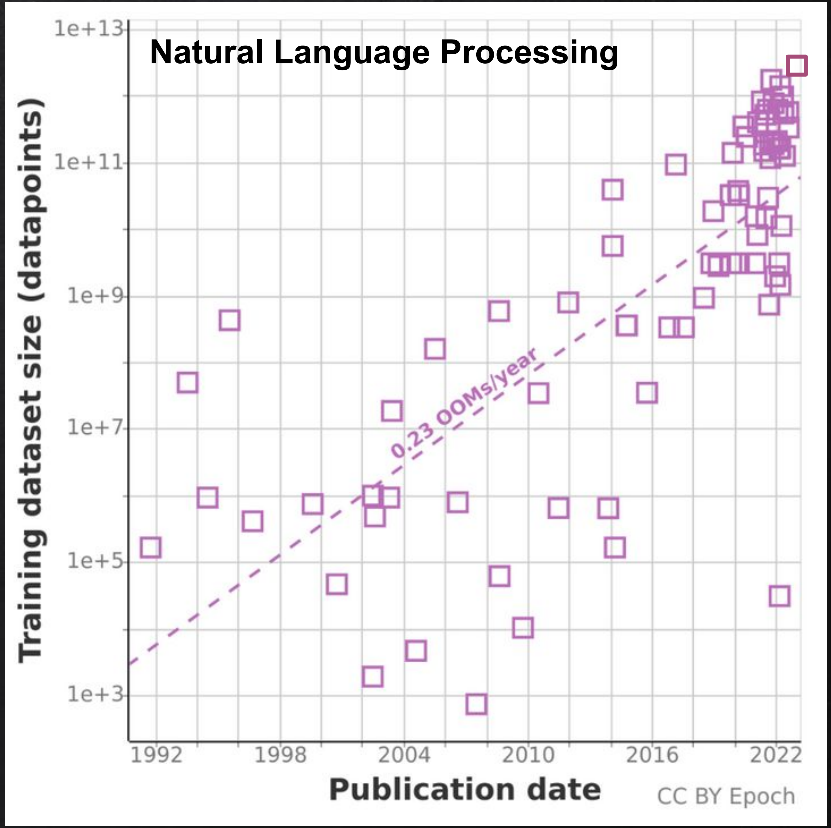
Learning on Big Data



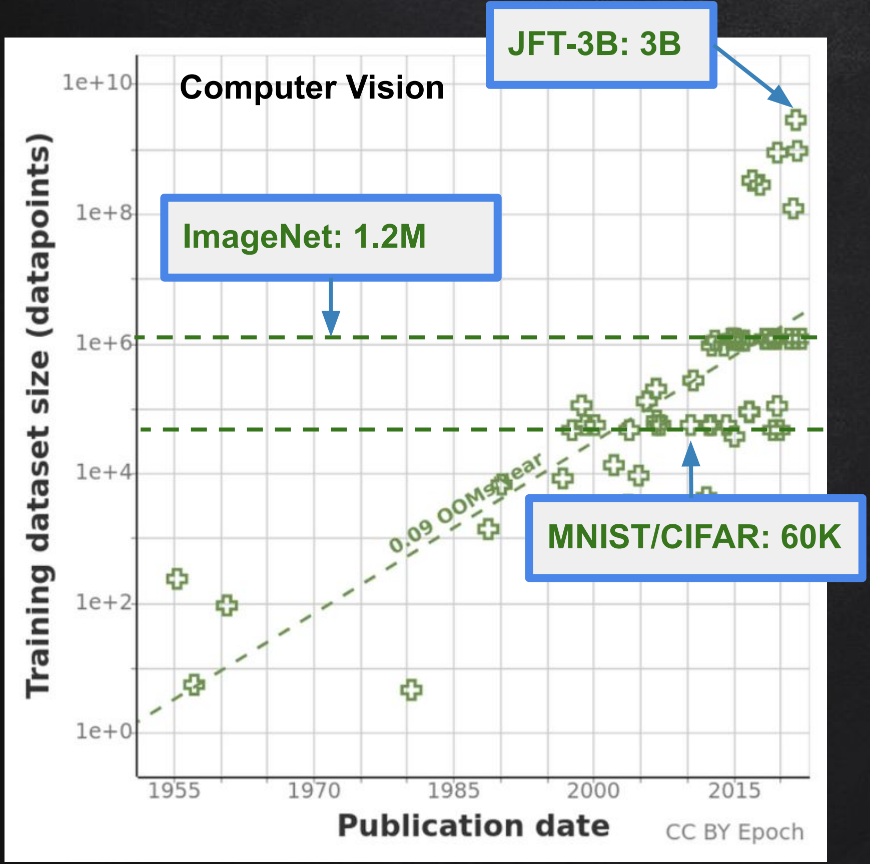


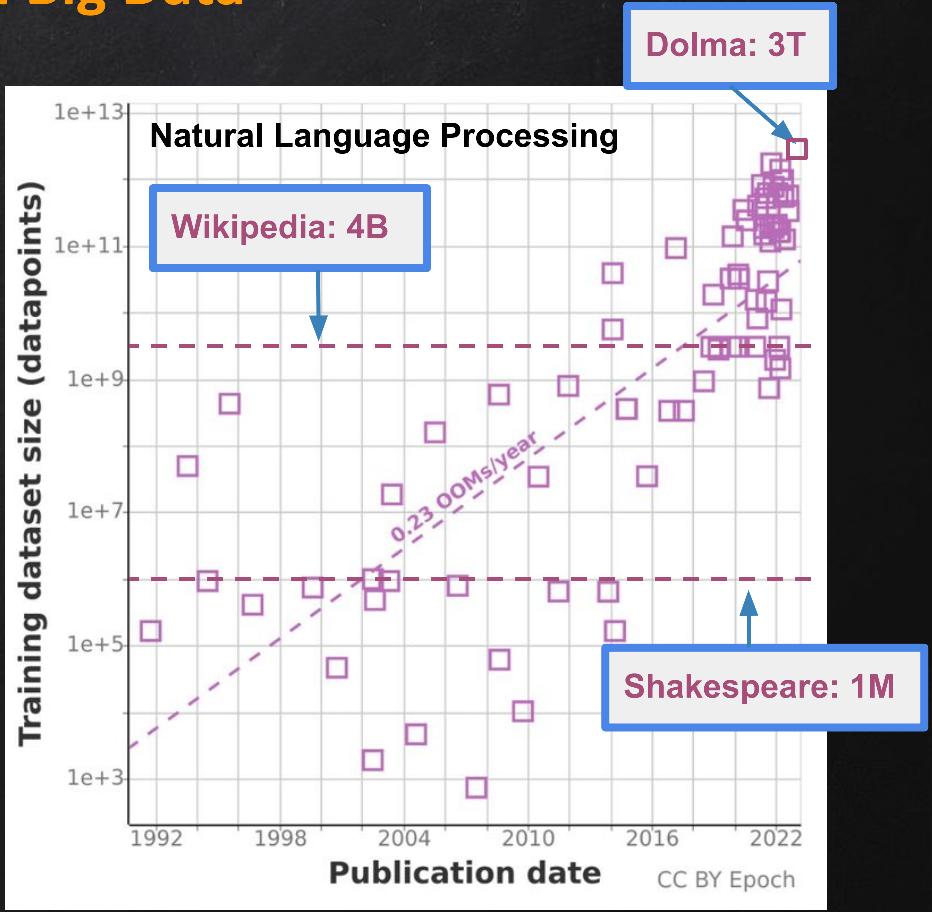
Learning on Big Data





Learning on Big Data

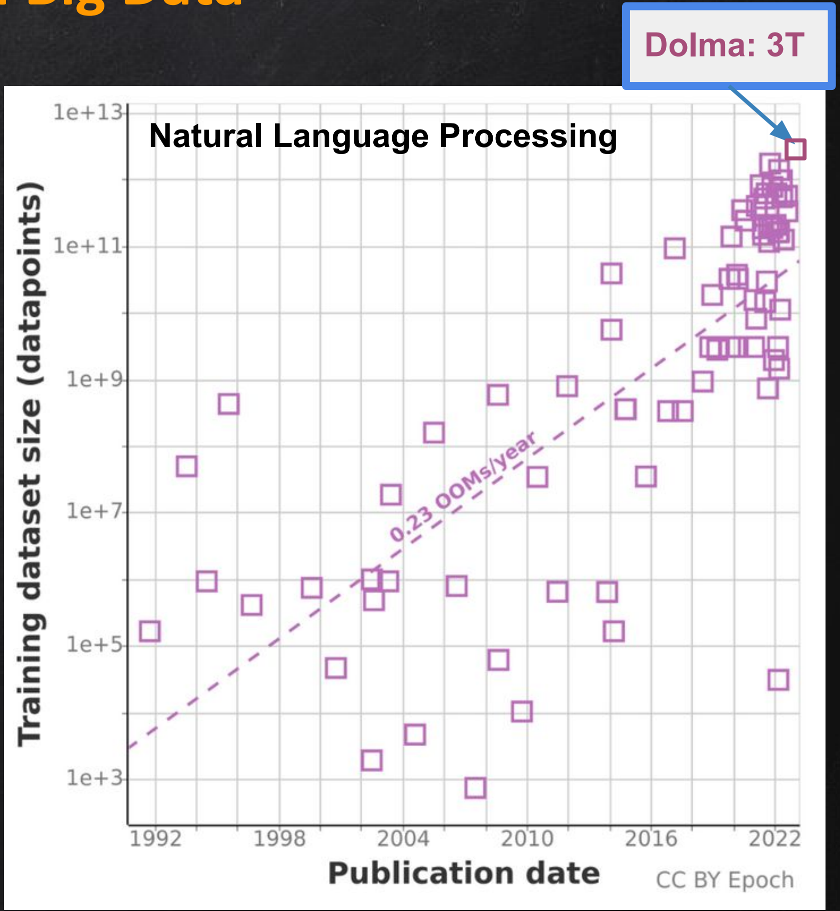




Learning on Big Data

Dolma: An Open Corpus of 3 Trillion Tokens for Language Model Pretraining Research

Subset: *Source*	*Kind*	*Gzip files (GB)*	Size: *Documents (millions)*	*Tokens (billions)*
Common Crawl 24 shards, 2020-05 to 2023-06	web	4,197	4,600	2,415
C4 [24] [8]	web	302	364	175
peS2o [27]	academic	150	38.8	57
The Stack [16]	code	675	236	430
Project Gutenberg	books	6.6	0.052	4.8
Wikipedia, Wikibooks (*en, simple*)	encyclopedic	5.8	6.1	3.6
Total		**5,334**	**5,245**	**3,084**



Learning on Big Data

Dolma: An Open Corpus of 3 Trillion Tokens for Language Model Pretraining Research

Subset *Source*	*Kind*	**Size** *Gzip files (GB)*	*Documents (millions)*	*Tokens (billions)*
Common Crawl 24 shards, 2020-05 to 2023-06	web	4,197	4,600	2,415
C4 [24] [8]	web	302	364	175
peS2o [27]	academic	150	38.8	57
The Stack [16]	code	675	236	430
Project Gutenberg	books	6.6	0.052	4.8
Wikipedia, **Wikibooks** (*en*, *simple*)	encyclopedic	5.8	6.1	3.6
Total		**5,334**	**5,245**	**3,084**

Will we run out of data soon?

Natural Language:

- 350 billion emails sent per day
- an average email has 400 words

→ 160 trillion tokens per day,
60 000 trillion tokens per year

Computer Vision:

- 1.8 trillion photos taken per year

Learning on User Data?

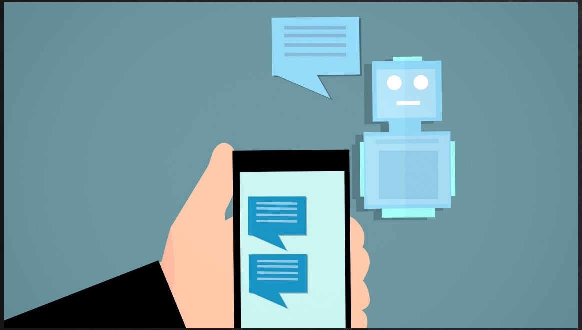


Image: CC0 Public Domain

Personal Assistants

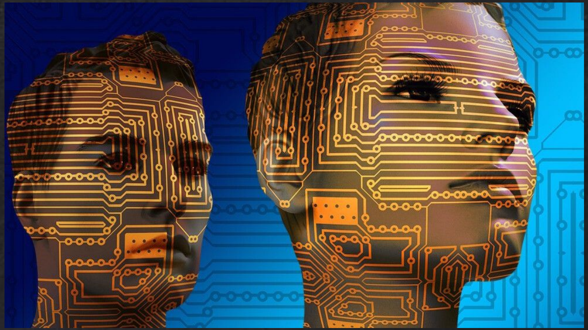


Image: Gerd Altmann (pixabay) CC0

Personal Healthcare

Image by Dllu under CC BY-SA 4.0

Autonomous Driving

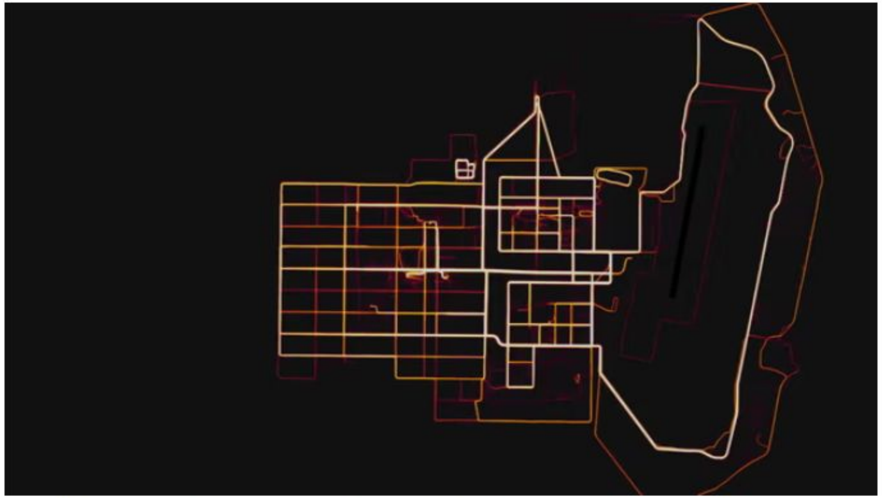
Image NASA/Norman Kuring

Sustainability


Learning o User Data?
Image: CC0 Public Domain
age: Gerd Altmann (pixabay) CC0
P rsonal ssi ants
PRIVACY!
Image NASA/Norman Kuring
Autonomous Driving
Sustainability

Fitness tracking app Strava gives away location of secret US army bases

Data about exercise routes shared online by soldiers can be used to pinpoint overseas facilities



A military base in Helmand Province, Afghanistan with route taken by joggers highlighted by Strava. Photograph: Strava Heatmap

ChatGPT banned in Italy over privacy concerns

1 April

OpenAI launched ChatGPT last November

Privacy. It's not just a good idea. It's the law!

$1 000 000 000 000 Question

Can we train machine learning models without the data owners having to give away their data?

Google Research Philosophy Research Areas Publications People Resources

BLOG ›

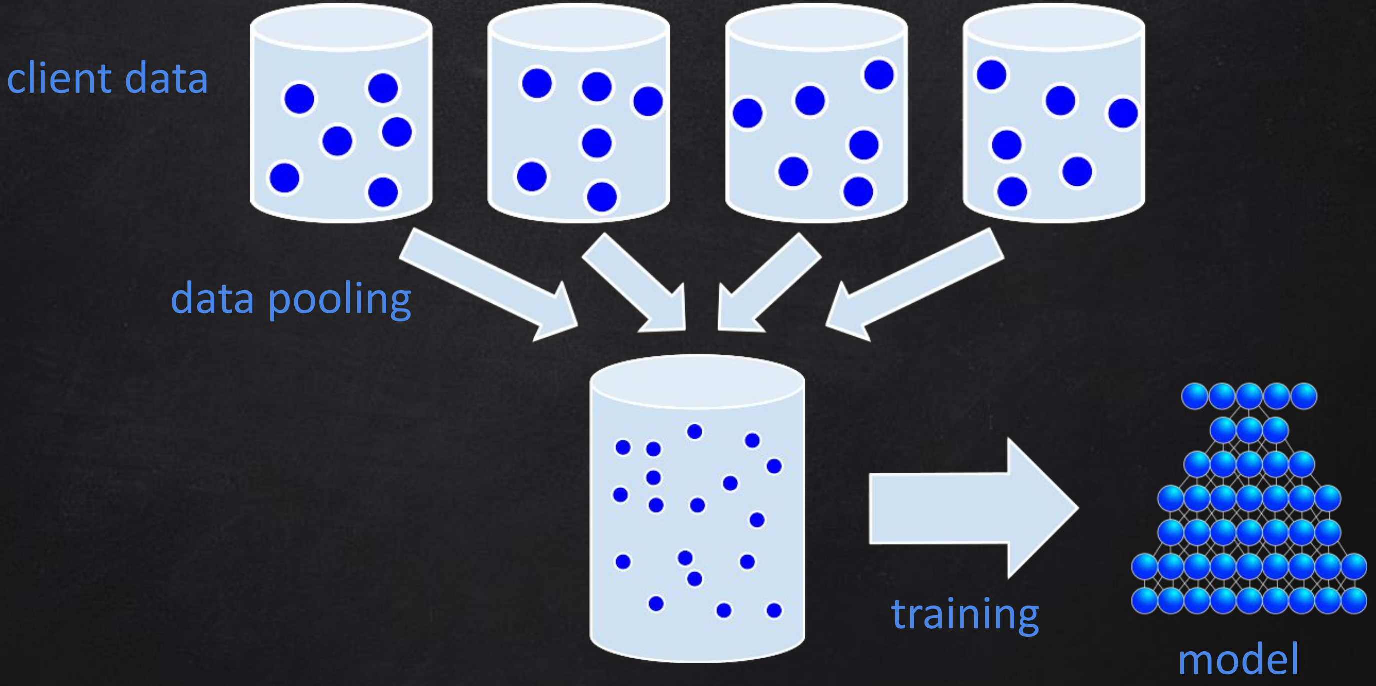
Federated Learning: Collaborative Machine Learning without Centralized Training Data

THURSDAY, APRIL 06, 2017

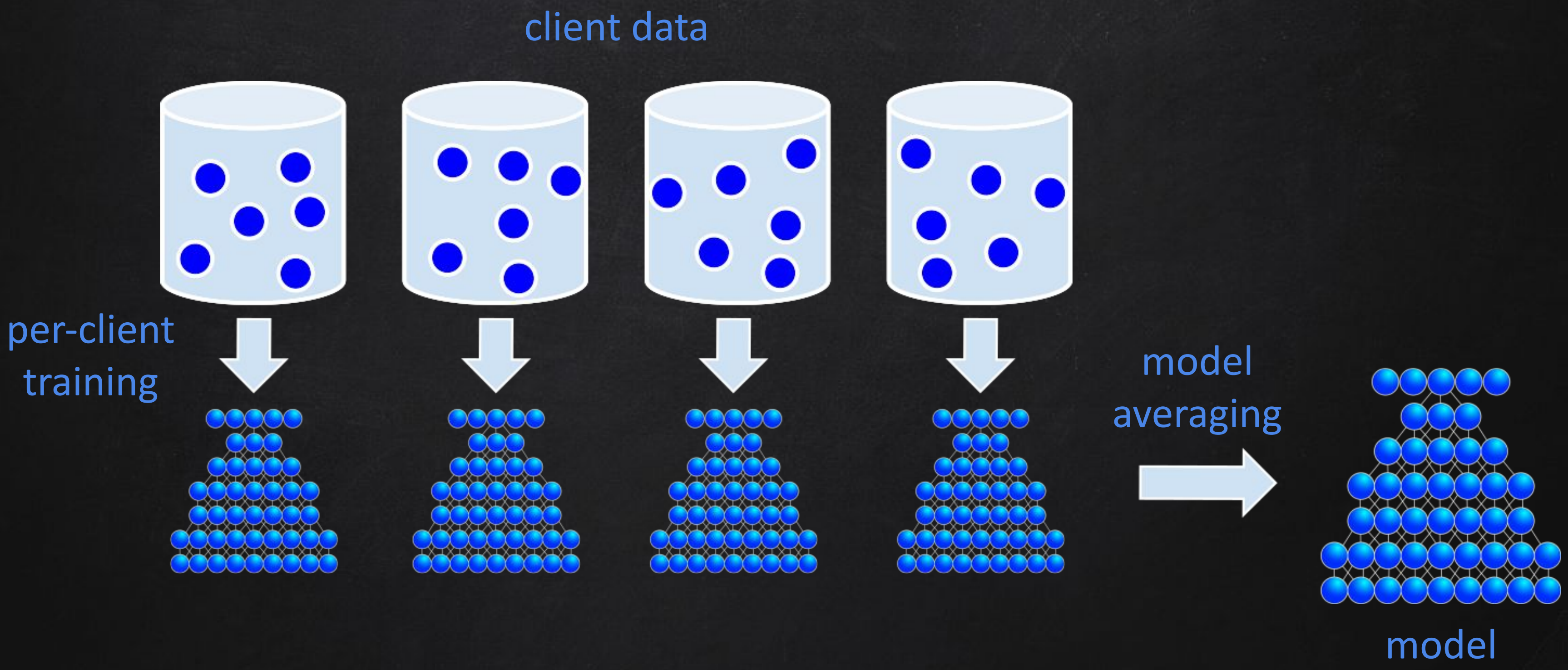
Posted by Brendan McMahan and Daniel Ramage, Research Scientists

https://blog.research.google/2017/04/federated-learning-collaborative.html

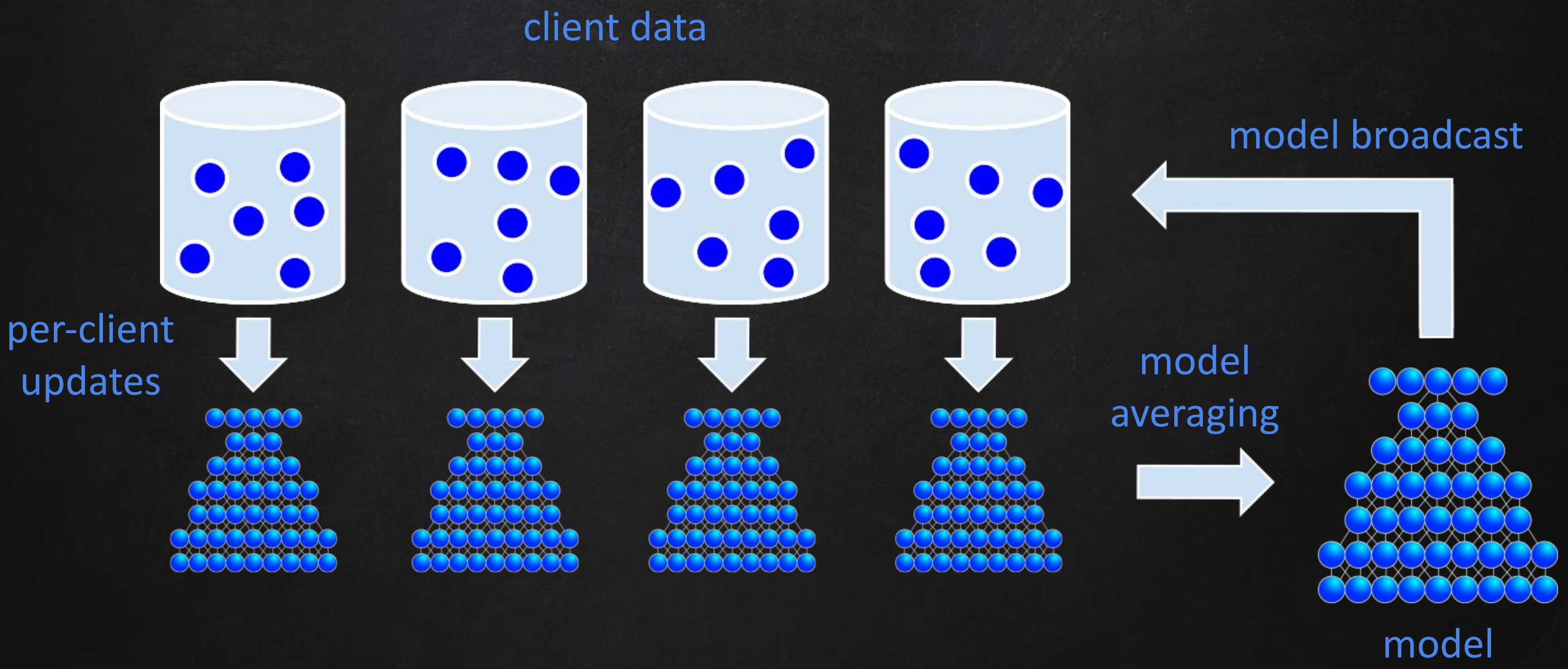
Centralized Learning



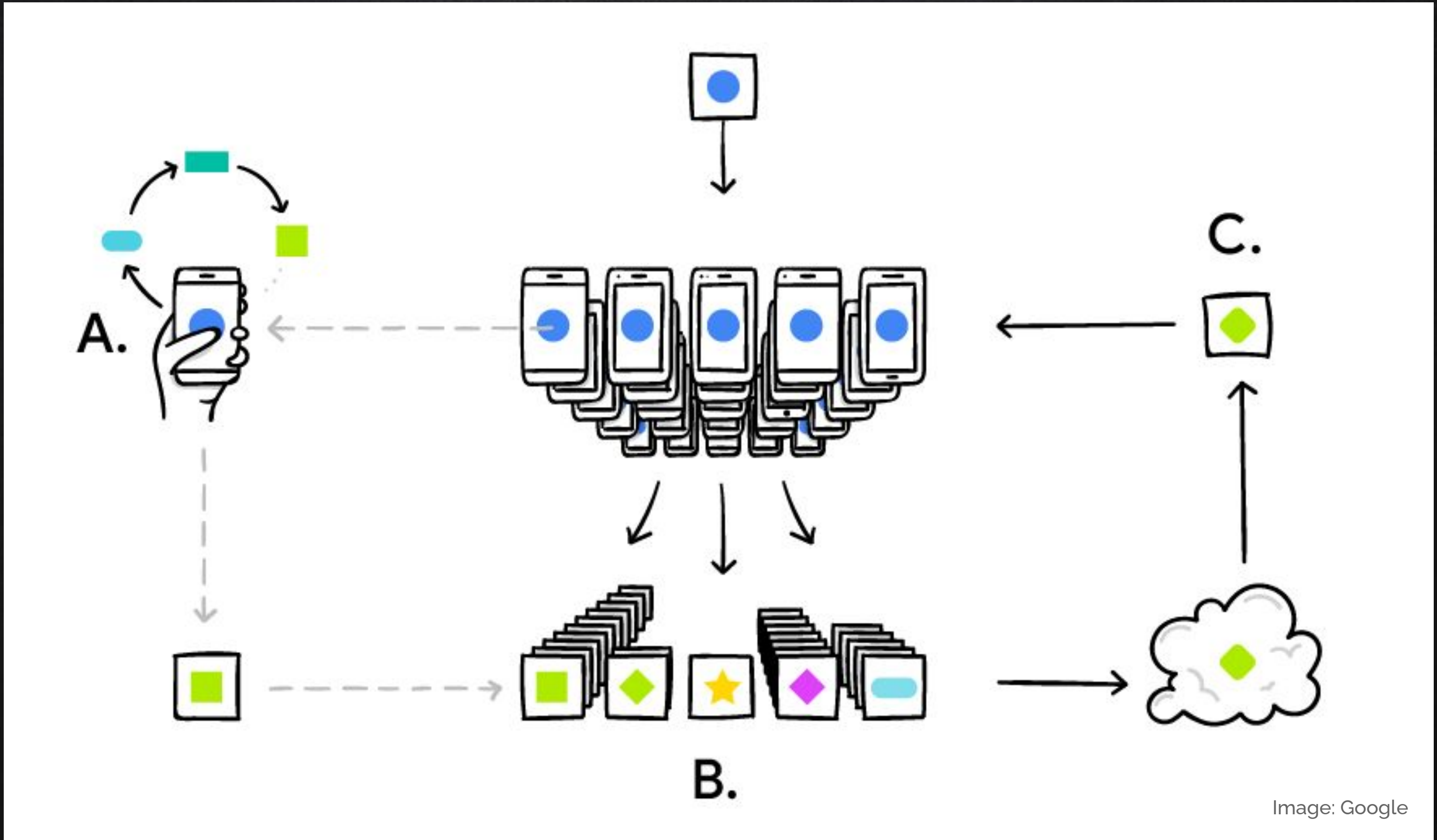
Decentralized Learning



Federated Learning

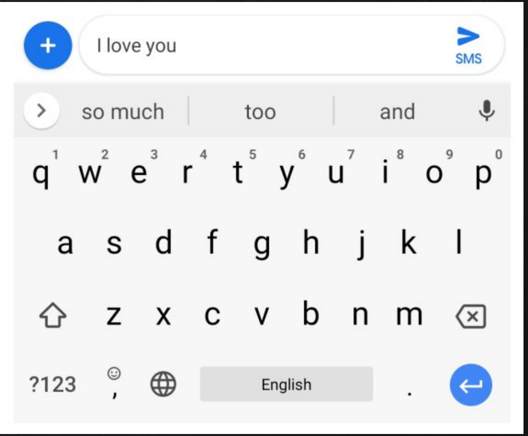


Federated Learning



Federated Learning

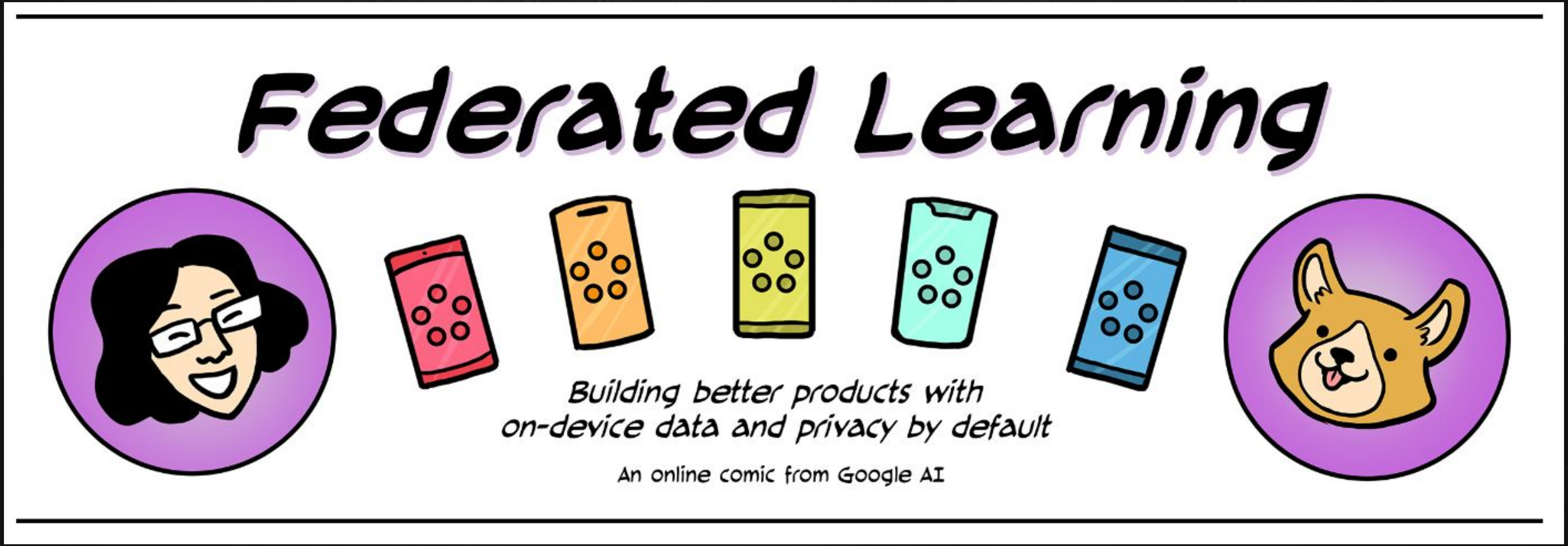
in-cloud auto-complete



Gboard

on-device next-word-prediction

[Hard et al, "Federated learning for mobile keyboard prediction", arXiv:1811.03604]



federated.withgoogle.com

Federated Learning - Considerations

- Efficacy
  - quality of learned models
- Efficiency
  - computational
  - communication
  - energy
- Robustness
  - clients can drop out any time, new clients might appear
  - clients are heterogeneous in hardware and data distributions
- Privacy
  - how well is the user data protected?
- Real-World Applications

Federated Learning - Considerations

- Efficacy
  - quality of learned models
- Efficiency
  - computational
  - communication
  - energy
- Robustness
  - clients can drop out any time, new clients might appear
  - clients are heterogeneous in hardware and data distributions
- Privacy
  - how well is the user data protected?
- Real-World Applications

Federated Learning - Efficiency

Simplest FL Algorithm: FedSGD [McMahan et al, AISTATS 2017]

```
1) server sends model to all clients
2) each client perform one step of SGD using their own data
3) each client sends updated model to server
4) server computes average over client models
5) goto 1)
```

Observation:

- equivalent to ordinary SGD on all data combined
- extremely inefficient in terms of communication cost

Federated Learning - Efficiency

Most popular FL Algorithm: FedAvg [McMahan et al, AISTATS 2017]

```
1) server sends model to all clients
2) each client perform K steps of SGD using their own data
3) each client sends updated model to server
4) server computes average over client models
5) goto 1)
```

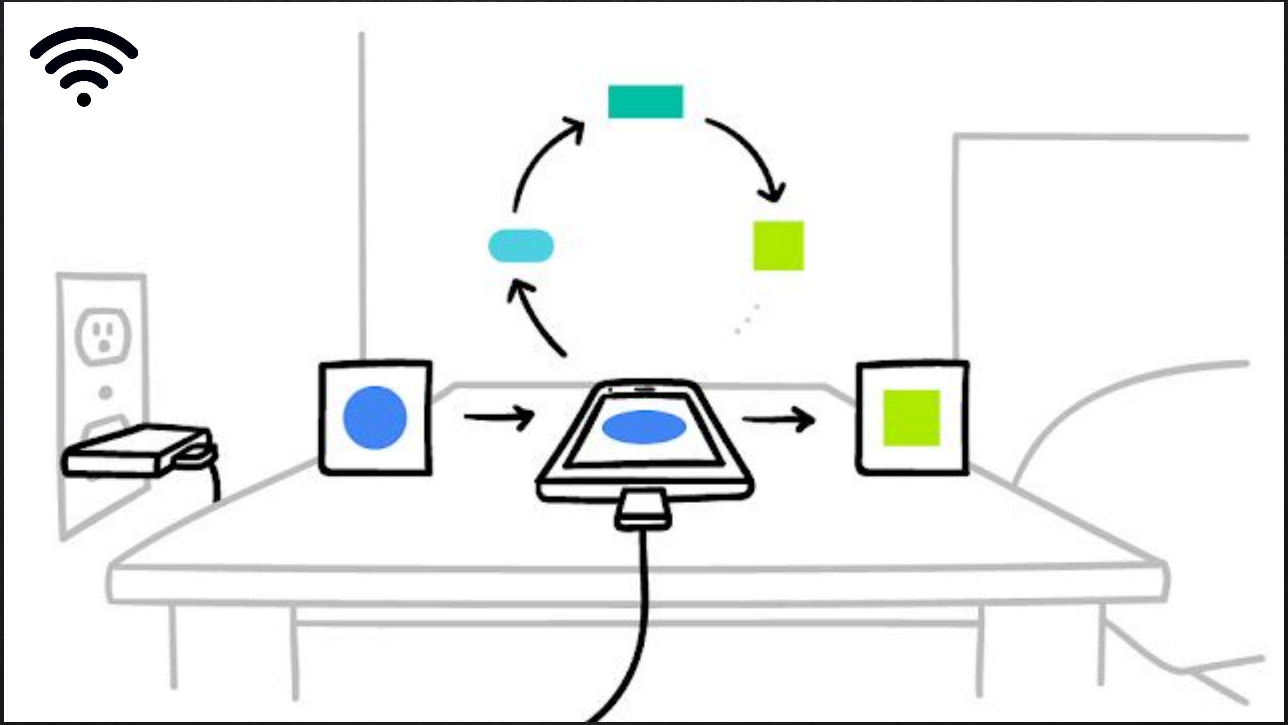
Observation: K trades off computational and communication efficiency

- small K: fast convergence, many communication rounds needed (K=1 $\longrightarrow$ FedSGD)
- large K: slow or no convergence, fewer communication rounds needed

Federated Learning - Considerations

- Efficacy
  - quality of learned models
- Efficiency
  - computational
  - communication
  - energy
- Robustness
  - clients can drop out any time, new devices might appear
  - clients are heterogeneous in hardware and data distributions
- Privacy
  - how well is the user data protected?
- Real-World Applications

Federated Learning - Energy

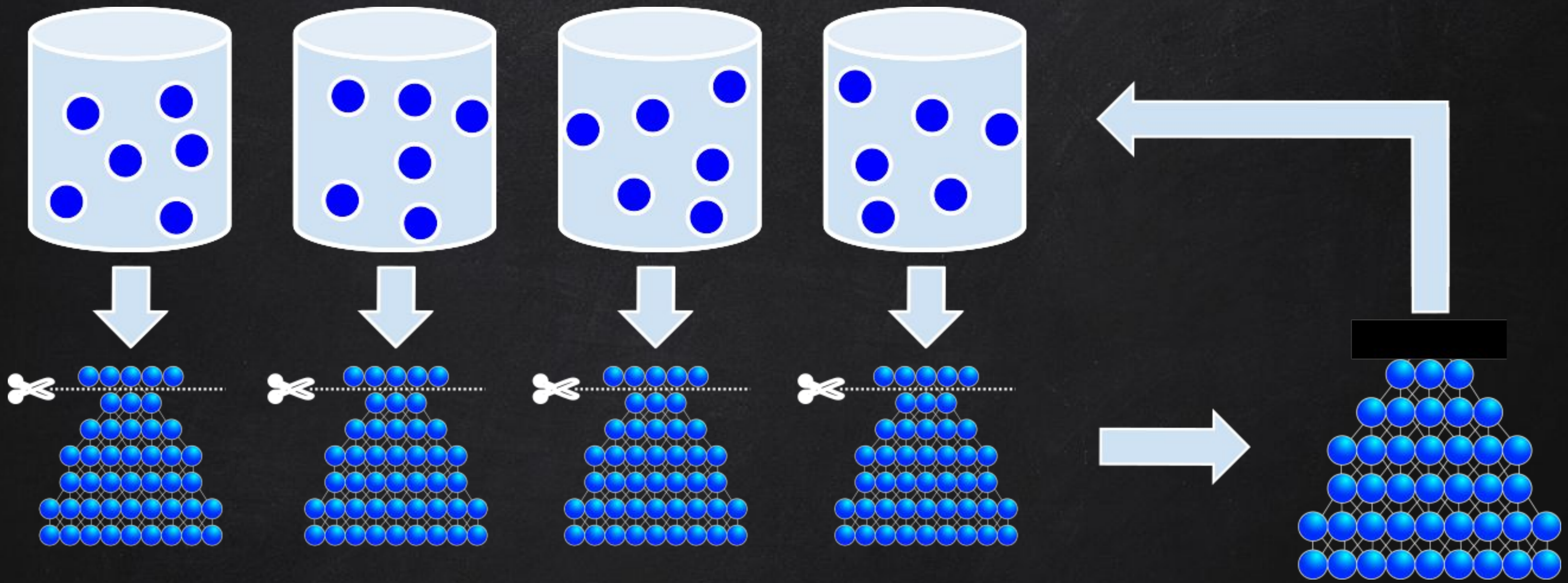


mobile devices: train only when plugged in and connected to WiFi

Federated Learning - Considerations

- Efficacy
  - quality of learned models
- Efficiency
  - computational
  - communication
  - energy
- Robustness
  - clients can drop out any time, new devices might appear
  - clients are heterogeneous in hardware and data distributions
- Privacy
  - how well is the user data protected?
- Real-World Applications

Federated Learning - Personalization



Each client learns its own model, e.g.:

- feature representation network is shared with all others
- prediction heads are specific to each client

[Arivazhagan et al, “Federated Learning with Personalization Layers”, arXiv:1912.00818]

Federated Learning - Considerations

- Efficacy
  - quality of learned models
- Efficiency
  - computational
  - communication
  - energy
- Robustness
  - clients can drop out any time, new clients might appear
  - clients are heterogeneous in hardware and data distributions
- Privacy
  - how well is the user data protected?
- Real-World Applications

Federated Learning - Privacy

How much does the central server learn about each clients' data?

- each client sends their updated model to the server receives
  $\longrightarrow$ the server knows which client made which updates (averaged gradients)

Deep Leakage from Gradients

[Zhu et al, NeurIPS 2019]

Inverting Gradients - How easy is it to break privacy in federated learning?

[Geiping et al, NeurIPS 2020]

Reconstructing Training Data from Model Gradient, Provably

[Wang et al, AISTATS 2023]

Observation: server would not need the individual clients' updates, only their average.

Excurse: Secure Aggregation

Can one compute the sum of multiple values without learning the actual values?

Yes, with cryptography!

(all operations mod 32)

	client A	client B	client C	
original data	1 2 3 4	-1 0 -1 0	15 0 0 0	$\sum$ = 15 2 2 4
shared (random) secret	3 29 23 19	10 16 7 22	19 19 2 23	$\sum$ = 0 0 0 0
encrypted data	4 31 26 23	9 16 6 22	2 19 2 23	

server: 4 31 26 23 + 9 16 6 22 + 2 19 2 23 = 15 2 2 4

Actually, no server needed. Clients can also privately compute averages themselves.

[Bonawitz et al, “Practical Secure Aggregation for Privacy-Preserving Machine Learning”, CCS 2017]

Federated Learning - Privacy

How much can others learn about the training data from the model itself?

- deep learning models often memorize training data,
- model weights/output contain information about original training data

Membership Inference Attacks Against Machine Learning Models [Shokri et al, IEEE SP 2017]

Exploiting Unintended Feature Leakage in Collaborative Learning* [Melis et al, IEEE SP 2019]

Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning [Nasr et al, IEEE SP 2019]

Excurse: Membership Attacks

Given a model, find out if a certain example was used to train it or not?

Can we provably prevent this? Yes, with differential privacy!

A (randomized) learning algorithm $\mathcal{L}$ is called ε-differentially private, if

$$p(\mathcal{L}(S)) \leq e^{\epsilon} \cdot p(\mathcal{L}(S'))$$

for all training sets S, S' that differ in only a single element.

For small ε, influence of individual training examples vanishes in algorithms randomness.

[Abadi et al, "Deep Learning with Differential Privacy", CCS 2016]

Excurse: Membership Attacks

Given a model, find out if a certain example was used to train it or not?

Can we provably prevent this? Yes, with differential privacy!

A (randomized) learning algorithm $\mathcal{L}$ is called ε-differentially private, if

$$p(\mathcal{L}(S)) \leq e^{\epsilon} \cdot p(\mathcal{L}(S'))$$

for all training sets S, S' that differ in only a single element.

For small ε, influence of individual training examples vanishes in algorithms randomness.

Mechanisms to increase privacy of learning algorithms:

- adding noise to intermediate calculations (noisy gradients: DP-SGD)
- data subsampling and aggregation

Challenge: ensure that accuracy stays high!

[Abadi et al, "Deep Learning with Differential Privacy", CCS 2016]

Excurse: Membership Attacks

Given a model, find out if a certain example was used to train it or not?

Can we provably prevent this? Yes, with differential privacy!

A (randomized) learning algorithm $\mathcal{L}$ is called ε-differentially private, if

$$p(\mathcal{L}(S)) \leq e^{\epsilon} \cdot p(\mathcal{L}(S'))$$

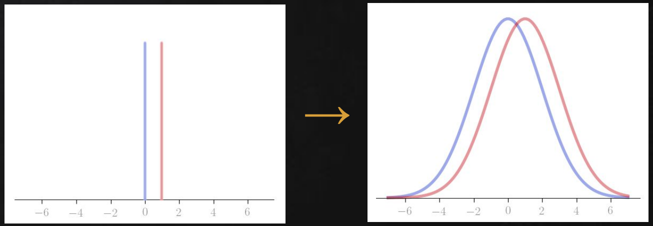
for all training sets S, S' that differ in only a single element.

For small ε, influence of individual training examples vanishes in algorithms randomness.

Mechanisms to increase privacy of learning algorithms:

- adding noise to intermediate calculations
- data subsampling and aggregation

Challenge: ensure that accuracy stays high!



[Abadi et al, "Deep Learning with Differential Privacy", CCS 2016]

Federated Learning - Considerations

- Efficacy
  - quality of learned models
- Efficiency
  - computational
  - communication
  - energy
- Robustness
  - clients can drop out any time, new clients might appear
  - clients are heterogeneous in hardware and data distributions
- Privacy
  - how well is the user data protected?
- Real-World Applications

Federated Learning - Application Scenarios

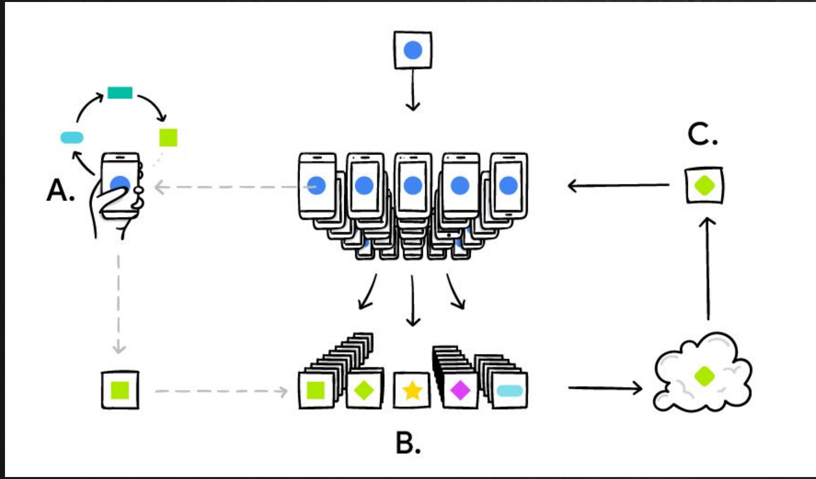


Image: Google

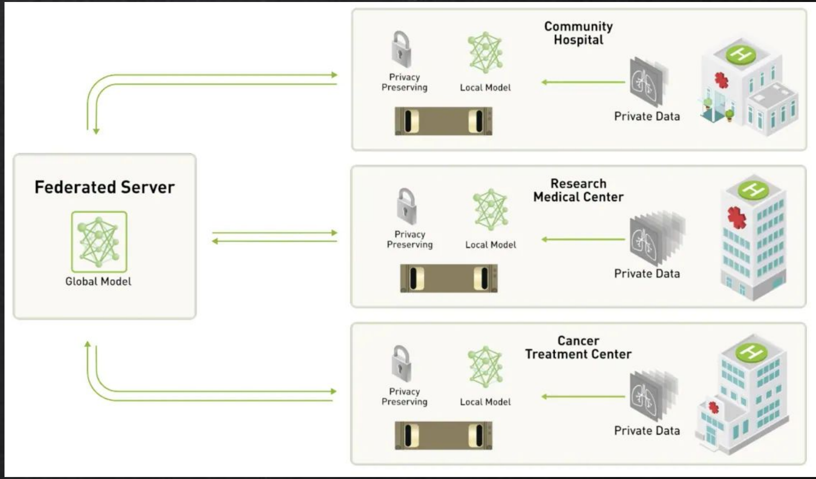


Image: NVIDIA

Cross-Device Federated Learning

(many clients, little data per client)

- next word prediction (Gboard)
- speech recognition
- personalized health
- autonomous driving

Cross-Silo Federated Learning

(few clients, a lot of data per client)

- healthcare
- predictive maintenance
- finance
- autonomous driving

Federated Learning -- Software Frameworks

FEDLAB

fedlab.readthedocs.io

Federated

tensorflow.org/federated

FedJAX

github.com/google/fedjax

Flower

flower.dev

⏩ 2023

Federated Learning at ISTA

Efficiency:

- more efficient distribution of models/updates: model compression, quantization, learning-to-learn

Beyond standard supervised learning:

- continual learning, semi-supervised learning, …

Privacy:

- multi-party computation, differential privacy

Theory:

- guarantees on convergence and/or generalization

Trustworthiness:

- how to protect the model against dishonest or biased clients?

Multi-agent Learning:

- how to incentivize clients to remain honest? $\longrightarrow$ Nikola Konstantinov (INSAIT, Sofia)

Federated Learning at ISTA

Efficiency:

- more efficient distribution of models/updates: model compression, quantization, learning-to-learn

Beyond standard supervised learning:

- continual learning, semi-supervised learning, …

Privacy:

- multi-party computation, differential privacy

Theory:

- guarantees on convergence and/or generalization

Trustworthiness:

- how to protect the model against dishonest or biased clients?

Multi-agent learning:

- how to incentivize clients to remain honest? ⟶ Nikola Konstantinov (INSAIT, Sofia)

Jonathan Scott, Hossein Zakerinia, CHL

“PeFLL: A Lifelong Learning Approach to Personalized Federated Learning”

arXiv:2306.05515

Jonathan Scott

Hossein Zakerinia

Reminder: Personalized Federated Learning

A new client connects to the network and requests a personalized model

1) the server sends the model to the client

2) the client trains/finetunes using its own data (typically multiple epochs of SGD)

Observation:

- high latency: on-client training required before model is available
- inefficient: the client has to do all the computational work

Idea of PeFLL:

- reduce latency by avoiding multi-step optimization
- offload computation from the client to the server
- allow smaller client models by avoiding one-fits-all approach

Background: Learning-to-Learn

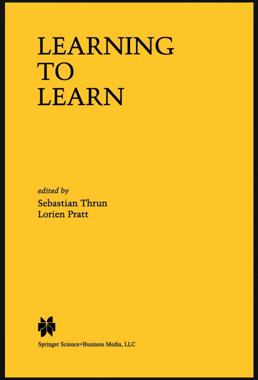
Abstract view of learning a model:

Standard learning:

- algorithm is fixed procedure: SGD on some loss function

Learning-to-learn:

- parametrize the learning algorithm and learn it



Excurse: Permutation Invariant Functions

How to parametrize a learning algorithm? We want a function $f : \mathcal{X} \to \mathcal{Y}$

- input: dataset $S = (z_1, \dots, z_m)$ output: model parameters $\theta \in \mathbb{R}^d$
- f should be *permutation invariant*: order of elements in S does not matter

Theorem 2 *A function $f(X)$ operating on a set X having elements from a countable universe, is a valid set function,* i.e., ***invariant*** *to the permutation of instances in X, iff it can be decomposed in the form $\rho\left(\sum_{x \in X} \phi(x)\right)$, for suitable transformations ϕ and ρ.*

[Zaheer et al. "Deep Sets", NeuIPS 2017]

[Jonathan Scott, Hossein Zakerinia, Christoph H. Lampert. "PeFLL: A Lifelong Learning Approach to Personalized Federated Learning", arXiv:2306.05515]

Excurse: Permutation Invariant Functions

How to parametrize a learning algorithm? We want a function $f : \mathcal{X} \to \mathcal{Y}$

- input: dataset $S = (z_1, \dots, z_m)$ output: model parameters $\theta \in \mathbb{R}^d$
- f should be *permutation invariant*: order of elements in S does not matter

Theorem 2 *A function $f(X)$ operating on a set X having elements from a countable universe, is a valid set function,* i.e., ***invariant*** *to the permutation of instances in X, iff it can be decomposed in the form $\rho\left(\sum_{x \in X} \phi(x)\right)$, for suitable transformations ϕ and ρ.*

[Zaheer et al. "Deep Sets", NeuIPS 2017]

PeFLL: $f(S; \eta_v, \eta_h) = \rho\left(\frac{1}{m}\sum_i \phi(z_i; \eta_v)\, ; \,\eta_h\right)$

- ϕ data embedding network (small) $\longrightarrow$ $\frac{1}{m}\sum_i \phi(z_i; \eta_v)$ acts as client descriptor
- ρ hyper-network (large): predict model from client descriptor

[Jonathan Scott, Hossein Zakerinia, Christoph H. Lampert. "PeFLL: A Lifelong Learning Approach to Personalized Federated Learning", arXiv:2306.05515]

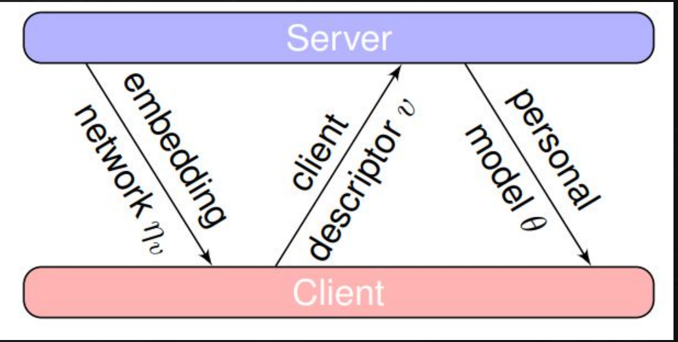
PeFLL - Prediction Phase

A new client connects to the network and requests a personalized model

1) the server sends the data embedding model to the client
2) the client encodes (some of) its data and averages the result
3) the client sends the resulting descriptor vector to the server
4) the server evaluates the hypernetwork with the client descriptor as input
5) the server send the resulting personalized model parameters to the client

Observation:

- the server performs most of the computation
- low latency:
  - three communication steps in total
  - no iterative optimization



[Jonathan Scott, Hossein Zakerinia, Christoph H. Lampert. "PeFLL: A Lifelong Learning Approach to Personalized Federated Learning", arXiv:2306.05515]

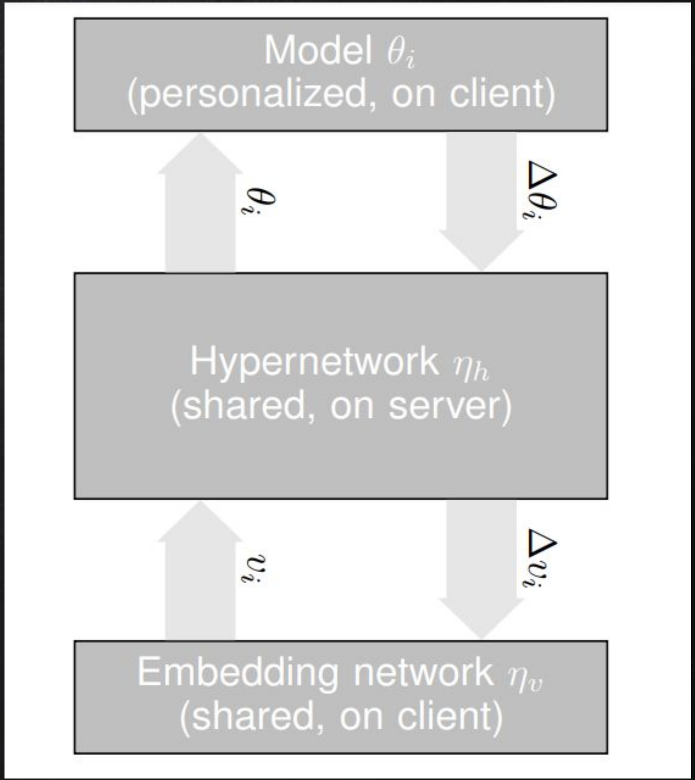
PeFLL - Training Phase

End-to-end (meta-)learning problem:

- each client computes the loss of its personalized model
- some regularizers suggested by theory ⟶ next slides

Train via SGD, just taking care to adhere to federated principle:

- data does not leave clients
- no heavy optimization on the client
- no large amount of data transferred between server and clients



[Jonathan Scott, Hossein Zakerinia, Christoph H. Lampert. "PeFLL: A Lifelong Learning Approach to Personalized Federated Learning", arXiv:2306.05515]

PeFLL - Convergence Guarantees

Does PeFLL the training procedure converge? Yes!

Theorem 3.1. *Under standard smoothness and boundedness assumptions (see appendix), PeFLL's optimization after T steps fulfills*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\,\|\nabla F(\eta_t)\|^2 \leq \frac{(F(\eta_0)-F_*)}{\sqrt{cT}} + \frac{L(6\sigma_1^2+4k\gamma_G^2)}{k\sqrt{cT}} + \frac{224cL_1^2b_1^2b_2^2}{T} + \frac{8b_1^2\sigma_2^2}{b} + \frac{14L_1^2b_2^2\sigma_3^2}{b}, \tag{2}$$

where F is the PeFLL objective (1), *which is lower bounded by F_*. η_0 are the parameter values at initialization, $\eta_1,\ldots,\eta_T$ are the intermediate parameter values. L, L_1 are smoothness parameters of F and the local models. b_1, b_2 are bounds on the norms of the gradients of the local model and the hypernetwork, respectively. σ_1 is a bound on the variance of stochastic gradients of local models, and σ_2, σ_3 are bounds on the variance due to the clients generating models with data batches of size b instead of their whole training set. γ_G is a bound on the dissimilarity of clients, c is the number of clients participating at each round, and k is the number of local SGD steps performed by the clients.*

[Jonathan Scott, Hossein Zakerinia, Christoph H. Lampert. "PeFLL: A Lifelong Learning Approach to Personalized Federated Learning", arXiv:2306.05515]

PeFLL - Generalization Guarantees

Will the models that PeFLL predicts for the future clients actually work? Yes!

Theorem 4.2. *For all $\delta > 0$ the following statement holds with probability at least $1-\delta$ over the clients. For all parameter vectors, $\eta = (\eta_h, \eta_v)$:*

$$\mathop{\mathbb{E}}_{(D,S)\sim\mathcal{T}} \mathop{\mathbb{E}}_{(x,y)\sim D} \mathop{\mathbb{E}}_{\substack{\bar\eta_h\sim\mathcal{Q}_h\\ \bar\eta_v\sim\mathcal{Q}_v}} \ell(x,y,h(v(S;\bar\eta_v);\bar\eta_h)) \leq \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{(x,y)\in S_i}\mathop{\mathbb{E}}_{\substack{\bar\eta_h\sim\mathcal{Q}_h\\ \bar\eta_v\sim\mathcal{Q}_v}} \ell(x,y,h(v(S_i;\bar\eta_v);\bar\eta_h))$$

$$+\sqrt{\frac{\frac{1}{2\alpha_h}\|\eta_h\|^2+\frac{1}{2\alpha_v}\|\eta_v\|^2+\log(\frac{2\sqrt{n}}{\delta})}{2n}}+\mathop{\mathbb{E}}_{\substack{\bar\eta_h\sim\mathcal{Q}_h\\ \bar\eta_v\sim\mathcal{Q}_v}}\sqrt{\frac{\frac{1}{2\alpha_\theta}\sum_{i=1}^{n}\|h(v(S_i;\bar\eta_v);\bar\eta_h)\|^2+\log(\frac{8mn}{\delta})+1}{2mn}}.$$

[Jonathan Scott, Hossein Zakerinia, Christoph H. Lampert. "PeFLL: A Lifelong Learning Approach to Personalized Federated Learning", arXiv:2306.05515]

PeFLL - Experimental Setup

Standard Benchmarks (in academia):

- FEMNIST (clients are writers), CIFAR10/100 (clients are created synthetically)

Simulated federated setting:

- set of clients split into two groups: *"training clients"* and *"test clients"*
- per-client datasets split into *"training points"* and *"test points"*
- train PeFLL using only *training points* of *training clients*

How well will models produced by PeFLL work in the future?

1) for clients that participated in training: evaluate on test data of training clients
2) for new (previously unseen) clients: evaluate on test data of test clients

[Jonathan Scott, Hossein Zakerinia, Christoph H. Lampert. "PeFLL: A Lifelong Learning Approach to Personalized Federated Learning", arXiv:2306.05515]

PeFLL - Results

	FEMNIST
#trn.clients	3237
Local	62.2 ± 0.1
FedAvg	82.1 ± 0.2
Per-FedAvg	82.7 ± 0.9
FedRep	83.6 ± 0.8
pFedMe	85.9 ± 0.8
kNN-Per	85.2 ± 0.3
pFedHN	83.8 ± 0.3
PeFLL	$\mathbf{90.1 \pm 0.1}$

accuracy on clients seen
during training (test data)

	FEMNIST
#trn.clients	3237
FedAvg	81.9 ± 0.4
Per-FedAvg	81.1 ± 1.5
FedRep	82.8 ± 0.7
pFedMe	86.1 ± 0.4
kNN-Per	84.6 ± 0.6
pFedHN	82.5 ± 0.1
PeFLL	$\mathbf{90.7 \pm 0.2}$

accuracy on clients
not seen during training

- clear improvements over prior methods, especially if the number of clients is large
- comparable quality on training clients and on new clients → good generalization
- other datasets, ablation studies, etc., in manuscript

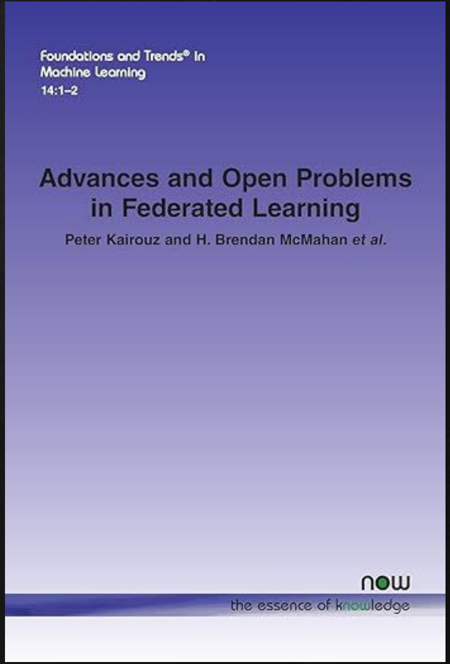
[Jonathan Scott, Hossein Zakerinia, Christoph H. Lampert. "PeFLL: A Lifelong Learning Approach to Personalized Federated Learning", arXiv:2306.05515]

Summary

Federated Learning: multiple clients learn a common model
- model parameters are exchanged between clients
- actual data never leaves the client

Relatively recent learning paradigm:
- high potential for privacy-preserving learning
- high commercial interest
- many challenges and open research questions
- connections to several other disciplines
    - distributed systems
    - cryptography
    - information theory

Further reading:



THANK YOU!