[FIFAI 2024] Introduction to Federated Learning (Day 2)

I- FedAvg (2016)

Communication-Efficient Learning of Deep Networks from Decentralized Data

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

Abstract

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

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

1 Introduction

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

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

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

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

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

II- FedProx (2020)

Federated Optimization in Heterogeneous Networks

Tian Li [1] **Anit Kumar Sahu** [2] **Manzil Zaheer** [3] **Maziar Sanjabi** [4] **Ameet Talwalkar** [1 5] **Virginia Smith** [1]

Abstract

Federated Learning is a distributed learning paradigm with two key challenges that differentiate it from traditional distributed optimization: (1) significant variability in terms of the systems characteristics on each device in the network (systems heterogeneity), and (2) non-identically distributed data across the network (statistical heterogeneity). In this work, we introduce a framework, `FedProx`, to tackle heterogeneity in federated networks. `FedProx` can be viewed as a generalization and re-parametrization of `FedAvg`, the current state-of-the-art method for federated learning. While this re-parameterization makes only minor modifications to the method itself, these modifications have important ramifications both in theory and in practice. Theoretically, we provide convergence guarantees for our framework when learning over data from non-identical distributions (statistical heterogeneity), and while adhering to device-level systems constraints by allowing each participating device to perform a variable amount of work (systems heterogeneity). Practically, we demonstrate that `FedProx` allows for more robust convergence than `FedAvg` across a suite of realistic federated datasets. In particular, in highly heterogeneous settings, `FedProx` demonstrates significantly more stable and accurate convergence behavior relative to `FedAvg`—improving absolute test accuracy by 22% on average.

1 Introduction

Federated learning has emerged as an attractive paradigm for distributing training of machine learning models in networks of remote devices. While there is a wealth of work on distributed optimization in the context of machine learning, two key challenges distinguish federated learning from traditional distributed optimization: high degrees of *systems and statistical heterogeneity*[1] (McMahan et al., 2017; Li et al., 2019).

In an attempt to handle heterogeneity and tackle high communication costs, optimization methods that allow for local updating and low participation are a popular approach for federated learning (McMahan et al., 2017; Smith et al., 2017). In particular, `FedAvg` (McMahan et al., 2017) is an iterative method that has emerged as the de facto optimization method in the federated setting. At each iteration, `FedAvg` first locally performs E epochs of stochastic gradient descent (SGD) on K devices—where E is a small constant and K is a small fraction of the total devices in the network. The devices then communicate their model updates to a central server, where they are averaged.

While `FedAvg` has demonstrated empirical success in heterogeneous settings, it does not fully address the underlying challenges associated with heterogeneity. In the context of systems heterogeneity, `FedAvg` does not allow participating devices to perform variable amounts of local work based on their underlying systems constraints; instead it is common to simply drop devices that fail to compute E epochs within a specified time window (Bonawitz et al., 2019). From a statistical perspective, `FedAvg` has been shown to diverge empirically in settings where the data is non-identically distributed across devices (e.g., McMahan et al., 2017, Sec 3). Unfortunately, `FedAvg` is difficult to analyze theoretically in such realistic scenarios and thus lacks convergence guarantees to characterize its behavior (see Section 2 for additional details).

In this work, we propose `FedProx`, a federated optimization algorithm that addresses the challenges of heterogeneity both theoretically and empirically. A key insight we have in developing `FedProx` is that an interplay exists between systems and statistical heterogeneity in federated learning. Indeed, both dropping stragglers (as in `FedAvg`) or naively incorporating partial information from stragglers (as in `FedProx` with the proximal term set to 0) implicitly increases statistical heterogeneity and can adversely impact

[1]Carnegie Mellon University [2]Bosch Center for Artificial Intelligence [3]Goolge Research [4]Facebook AI [5]Determined AI. Correspondence to: Tian Li <tianli@cmu.edu>.

Proceedings of the 3rd MLSys Conference, Austin, TX, USA, 2020.

[1]Privacy is a third key challenge in the federated setting. While not the focus of this work, standard privacy-preserving approaches such as differential privacy and secure multiparty communication can naturally be combined with the methods proposed herein—particularly since our framework proposes only lightweight algorithmic modifications to prior work.

III- FedALA (2023)

The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

FedALA: Adaptive Local Aggregation for Personalized Federated Learning

Jianqing Zhang[1], **Yang Hua**[2], **Hao Wang**[3], **Tao Song**[1], **Zhengui Xue**[1], **Ruhui Ma**[1*], **Haibing Guan**[1]
[1]Shanghai Jiao Tong University
[2]Queen's University Belfast
[3]Louisiana State University
{tsingz, songt333, zhenguixue, ruhuima, hbguan}@sjtu.edu.cn, Y.Hua@qub.ac.uk, haowang@lsu.edu

Abstract

A key challenge in federated learning (FL) is the statistical heterogeneity that impairs the generalization of the global model on each client. To address this, we propose a method ***Fed**erated learning with **A**daptive **L**ocal **A**ggregation* (**FedALA**) by capturing the desired information in the global model for client models in personalized FL. The key component of FedALA is an *Adaptive Local Aggregation* (**ALA**) module, which can adaptively aggregate the downloaded global model and local model towards the local objective on each client to initialize the local model before training in each iteration. To evaluate the effectiveness of FedALA, we conduct extensive experiments with five benchmark datasets in computer vision and natural language processing domains. FedALA outperforms eleven state-of-the-art baselines by up to 3.27% in test accuracy. Furthermore, we also apply ALA module to other federated learning methods and achieve up to 24.19% improvement in test accuracy. Code is available at https://github.com/TsingZ0/FedALA.

Introduction

Federated learning (FL) can leverage distributed user data while preserving privacy by iteratively downloading models, training models locally on the clients, uploading models, and aggregating models on the server. A key challenge in FL is statistical heterogeneity, *e.g.*, the not independent and identically distributed (Non-IID) and unbalanced data across clients. This kind of data makes it hard to obtain a global model that generalizes to each client (McMahan et al. 2017; Reisizadeh et al. 2020; T Dinh, Tran, and Nguyen 2020).

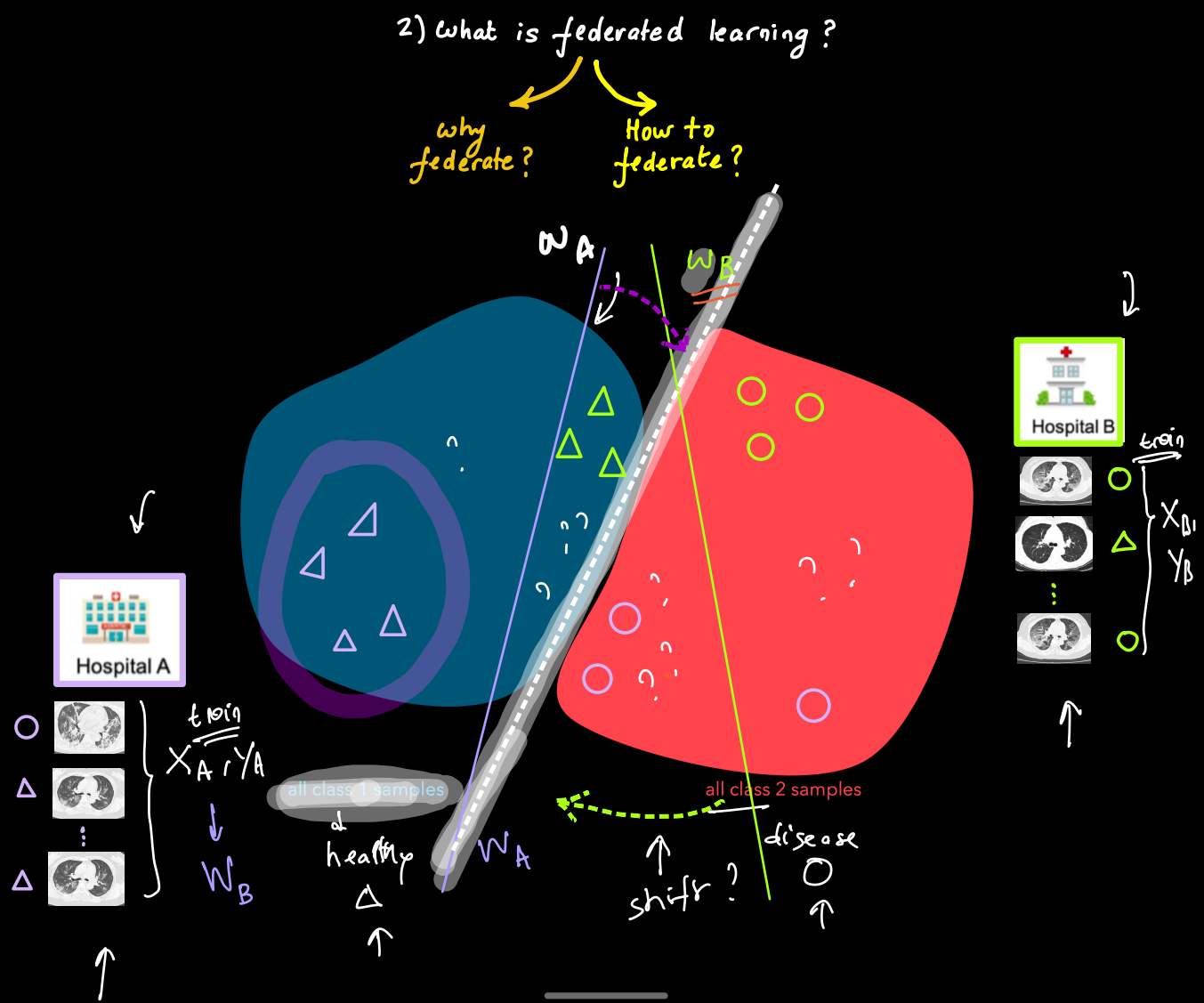
Personalized FL (pFL) methods have been proposed to tackle statistical heterogeneity in FL. Unlike traditional FL that seeks a high-quality global model via distributed training across clients, *e.g.*, FedAvg (McMahan et al. 2017), pFL methods are proposed to prioritize the training of a local model for each client. Recent pFL studies on model aggregation on the server can be classified into three categories: (1) methods that learn a single global model and fine-tune it, including Per-FedAvg (Fallah, Mokhtari, and Ozdaglar 2020) and FedRep (Collins et al. 2021), (2) methods that learn additional personalized models, including pFedMe (T Dinh, Tran, and Nguyen 2020) and Ditto (Li et al. 2021a), and (3) methods that learn local models with personalized (local) aggregation, including FedAMP (Huang et al. 2021), FedPHP (Li et al. 2021b), FedFomo (Zhang et al. 2020), APPLE (Luo and Wu 2021) and PartialFed (Sun et al. 2021).

pFL methods in Category (1) and (2) take all the information in the global model for local initialization, *i.e.*, initializing the local model before local training in each iteration. However, only the desired information that improves the quality of the local model is beneficial for the client. The global model has poor generalization ability since it has desired and undesired information for an individual client simultaneously. Thus, pFL methods in Category (3) intend to capture the desired information in the global model through personalized aggregation.

However, pFL methods in Category (3) still have shortcomings. FedAMP/FedPHP performs personalized aggregation on the server/clients without considering the local objective. FedFomo/APPLE downloads other client models and locally aggregates them with the approximated/learned aggregating weights on each client. All the parameters in one client model are assigned with the same weight, *i.e.*, model-level weight. Besides, downloading client models among clients causes high communication overhead in each iteration and also has privacy concerns since the data from other clients can be recovered through these client models (Zhu, Liu, and Han 2019). In addition, FedFomo/APPLE also requires feeding data forward in the downloaded client models to obtain the aggregating weights, which introduces additional computation overhead. PartialFed locally learns aggregation strategies to select the parameters in the global model or the local model. Still, the layer-level and binary selection cannot precisely capture the desired information in the global model. Furthermore, PartialFed uses non-overlapping samples to learn the local model and the strategy, so it can hardly learn a strategy that fully satisfies the local objective. Due to the significant modification of the learning process in FedAvg, the personalized aggregation process in these methods cannot be directly applied to most existing FL methods.

To precisely capture the desired information in the downloaded global model for each client without additional communication overhead in each iteration, we propose a novel pFL method ***Fed**erated learning with **A**daptive **L**ocal*

*Corresponding author.

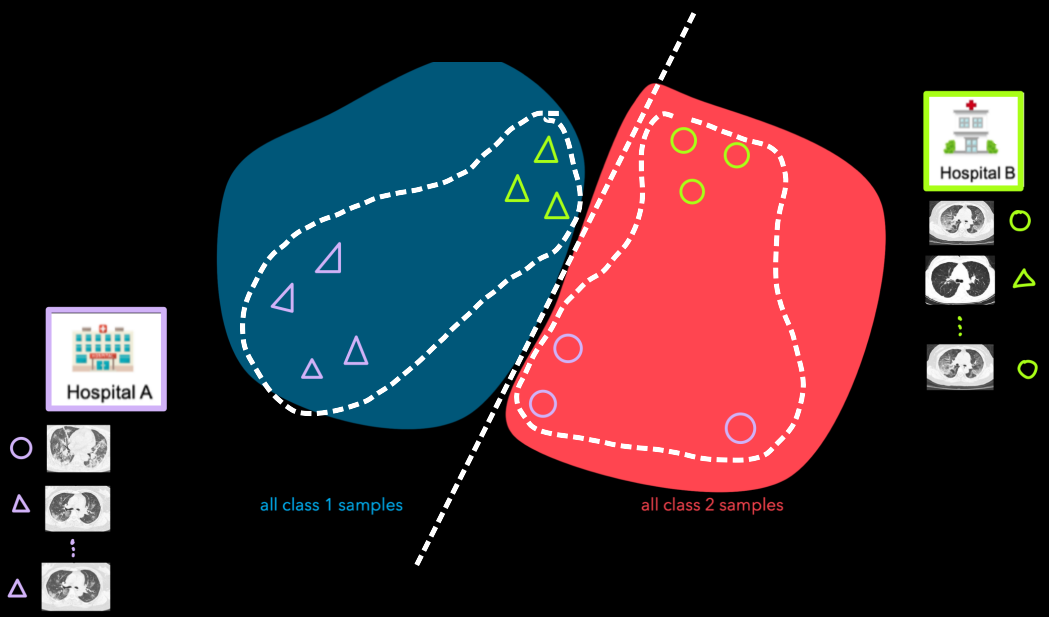


☹ Modern ML algorithms (classification, regression) are **data & computation hungry.**

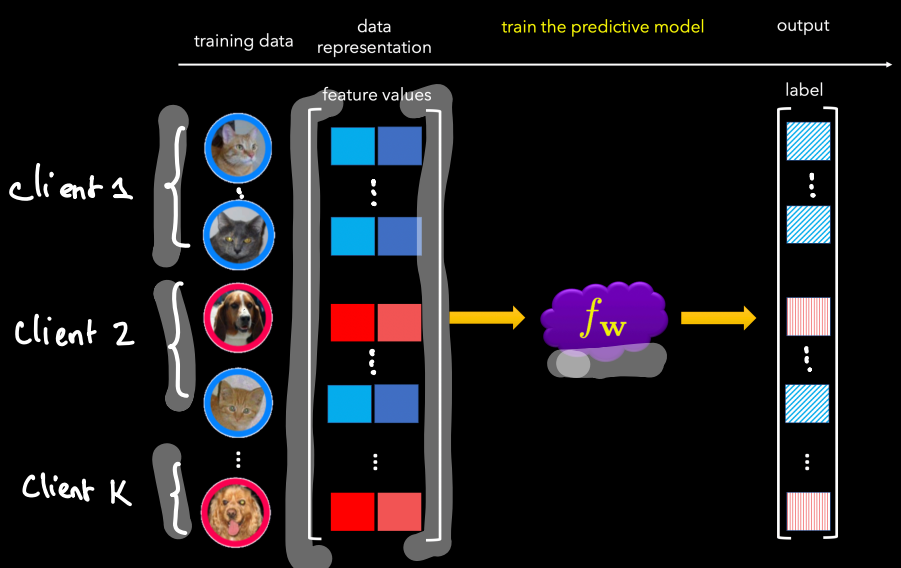
☹ The performance of an ML model is dictated by the data it is trained on (size, distribution, etc)

→ How can we design powerful & generalizable ML models ?

Solution 1 Build a single model using centralized data.



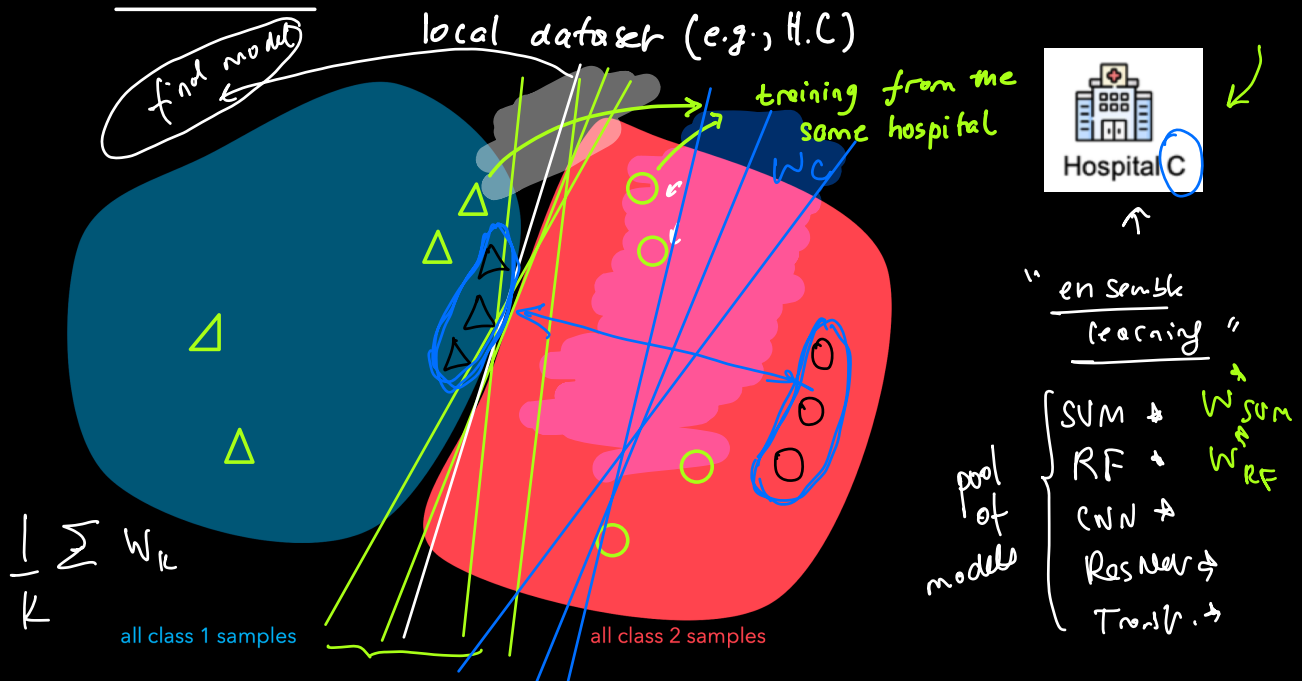
* The data is uploaded into a single device where a model is trained.



Concerns about data security and privacy
Such a data "sharing" process is not straightforward.
large-scale datasets are computationally expensive to upload.
(e.g.) on mobile phones & apps)
Intrusive and difficult if data is spread across multiple devices or clients.

→ How can we design powerful & generalizable ML models ?

Solution 2 Build an "ensemble" model on a single local dataset (e.g., H.C)



$\frac{1}{K}\sum w_k$

"ensemble learning"

pool of models { SVM → w_{SVM}, RF → w_{RF}, CNN →, ResNet →, Transf. → }

single model (classifier)

$\mathbf{x}$ → f_w → $\mathbf{y}$

Input data — Output data

$[X_c, y_c]$

ensemble

1) average models

2) average predictions

$f_{w^*_{SVM}}(x_{test}) = \hat{y}^{SVM}_{test}$

$\vdots$

$f_{w^*_{RF}}(x_{test}) = \hat{y}^{RF}_{test}$

"0"

☺ boost the model's performance

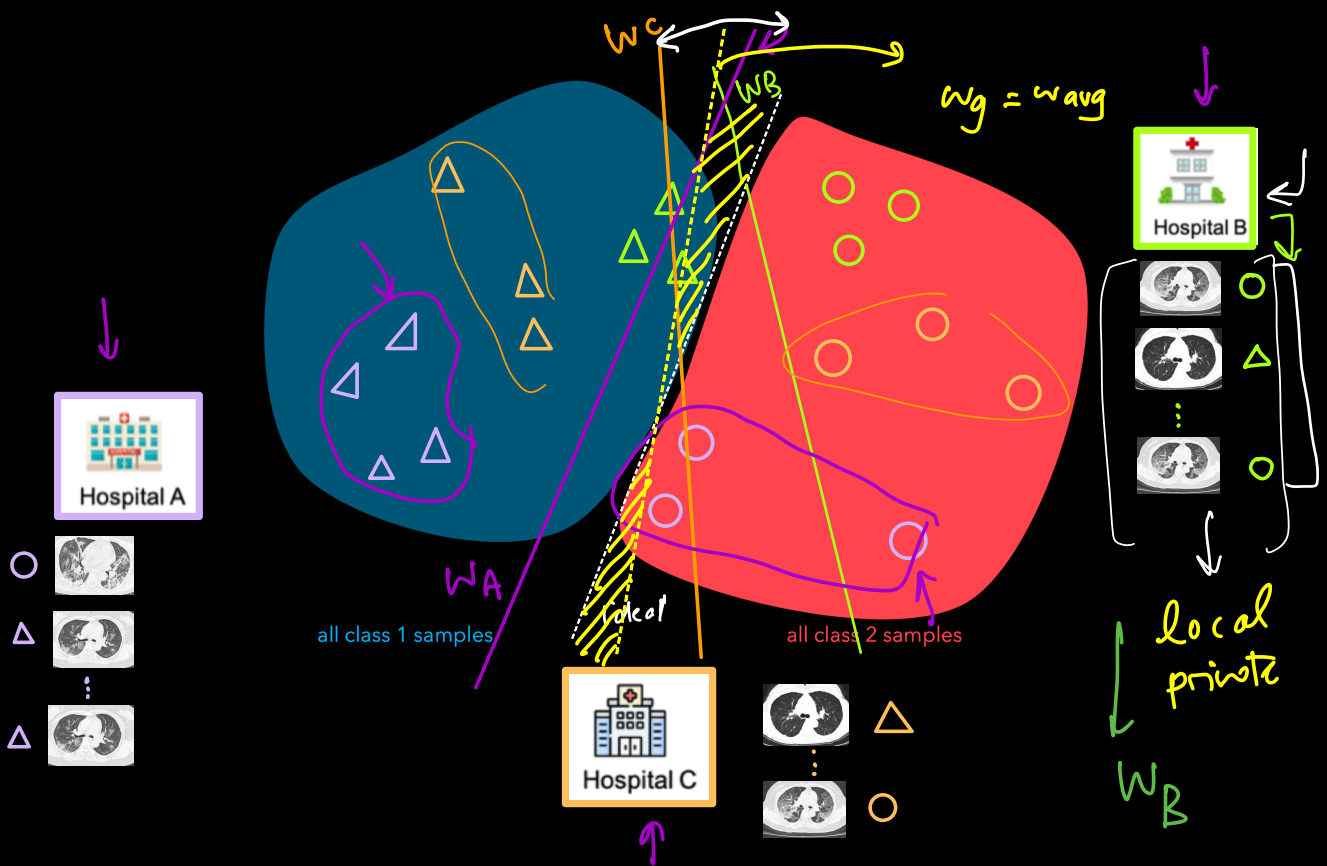
☹ high computations (many models to train on a single device)
training only benefits from a single dataset (here Hospital C)

Distribution

where your points are located?

→ How can we design powerful & generalizable ML models ?

Solution 3 Data-preserving federated learning



Goal : boost the local performance of a model of particular client / hospital without any data sharing.

Simple but creative Google paper by McMahon (2016)

Communication-Efficient Learning of Deep Networks from Decentralized Data

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

https://arxiv.org › cs

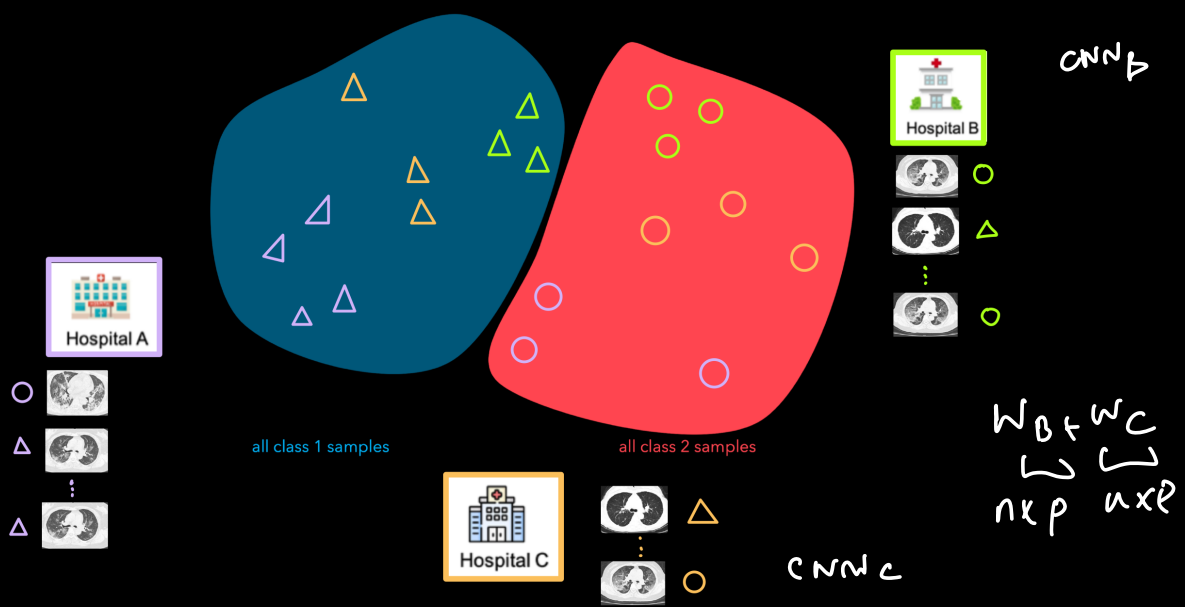
Communication-Efficient Learning of Deep Networks - arXiv

by HB McMahan · 2016 · Cited by 4039 — [Submitted on 17 Feb **2016** (v1), last revised 28 Feb 2017 (this version, v3)] ... Submission history. From: Hugh Brendan **McMahan**...

Goal : boost the local performance of a model of particular client / hospital without any data sharing.

THINK!

- Sol° 1 → centralized (one model on many datasets)
- Sol° 2 → ensemble (many models on a single dataset)
- Sol° 3 → federated (many models on many datasets)

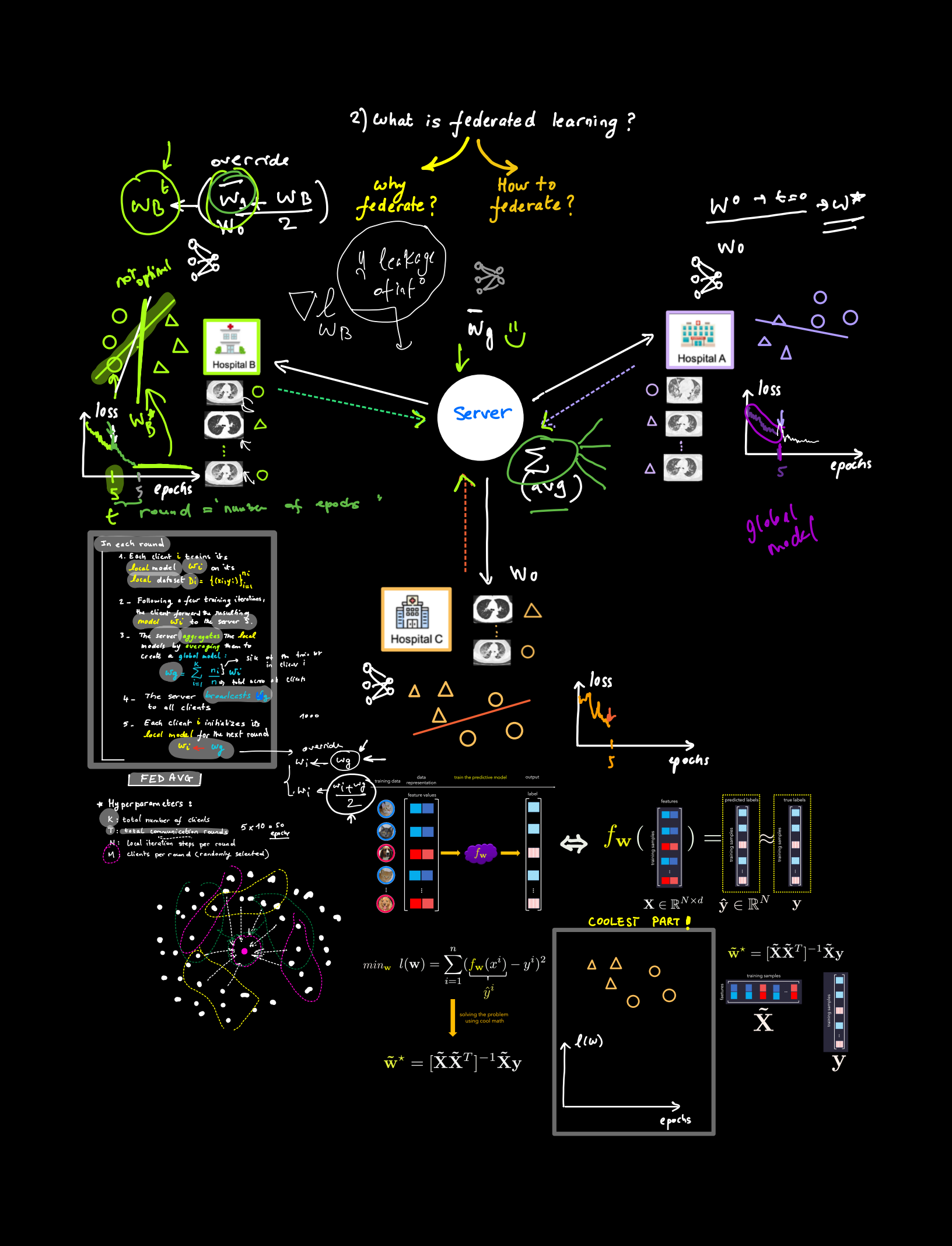


CNN B

$W_B + W_C$
↳ nxp ↳ nxp

CNN C

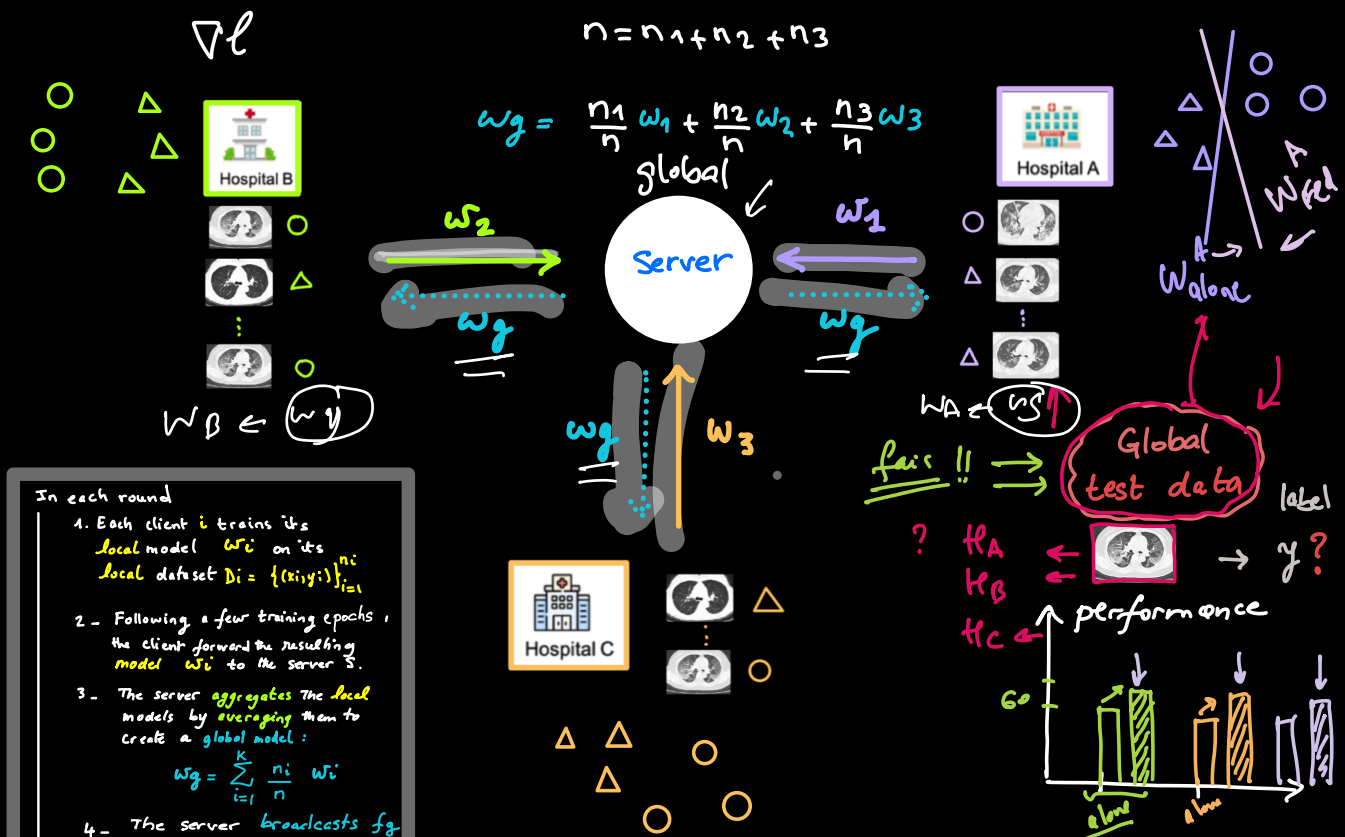
What's the trick here ?


2) what is federated learning ?
why federate ?
How to federate ?
override
not optimal
Hospital B
Hospital A
Hospital C
Server
loss
epochs
round = number of epochs
global model
In each round
1. Each client i trains its local model on its local dataset
2. Following a few training iterations, the client forward the resulting model to the server S.
3. The server aggregates the local models by averaging them to create a global model :
4. The server broadcasts to all clients
5. Each client i initializes its local model for the next round
FED AVG
* Hyperparameters :
K : total number of clients
T : total communication rounds
N : local iteration steps per round
M : clients per round (randomly selected)
5 x 10 = 50 epochs
1000
training data
data representation
train the predictive model
output
feature values
label
features
predicted labels
true labels
training samples
COOLEST PART !
solving the problem using cool math
epochs

2) What is federated learning ?

Why federate ?

How to federate ?



In each round

1. Each client i trains its *local* model w_i on its *local* dataset $D_i = \{(x_i, y_i)\}_{i=1}^{n_i}$
2. Following a few training epochs, the client forward the resulting model w_i to the server S.
3. The server aggregates the *local* models by *averaging* them to create a *global model* :

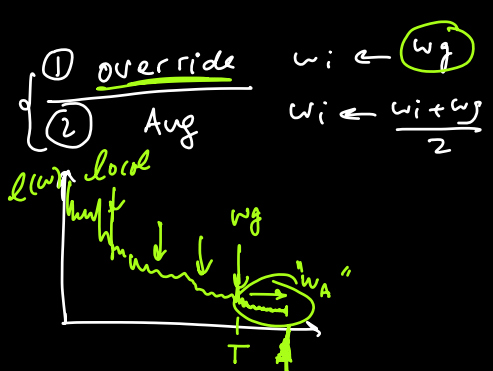
$$w_g = \sum_{i=1}^{K} \frac{n_i}{n} w_i$$

4. The server broadcasts f_g to all clients
5. Each client i initializes its *local model* for the next round $w_i \leftarrow w_g$

FED AVG

* Hyperparameters :

- K : total number of clients
- T : total communication rounds
- N : local epochs per round
- M : clients per round (randomly selected)



$$w^{t+1} = w^t - \eta \nabla \ell(w^t)$$

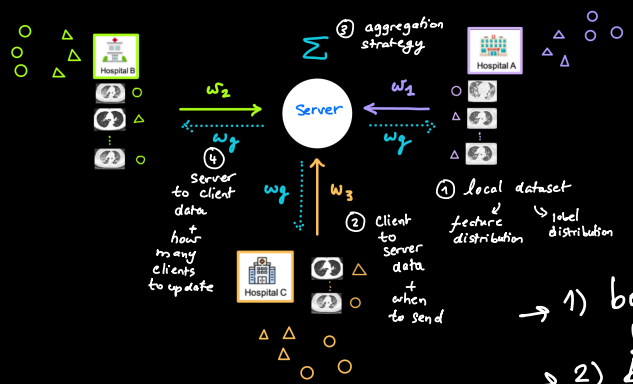
3) Exciting challenges & future AI problems TO SOLVE!

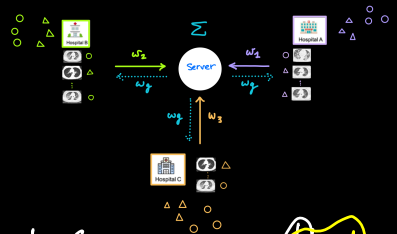
① formalize your FL problem

② Describe a potential solution

③ Design your experiments & evaluate

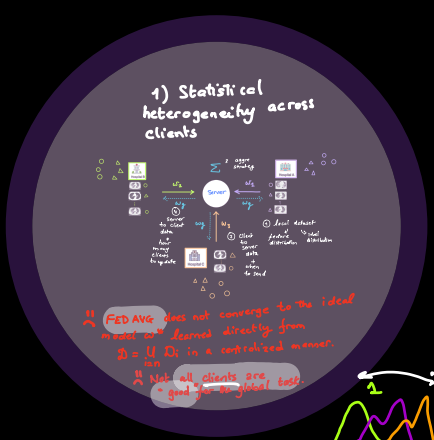
Each FL component = challenge

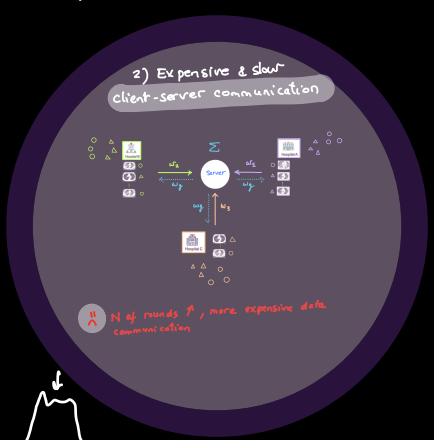


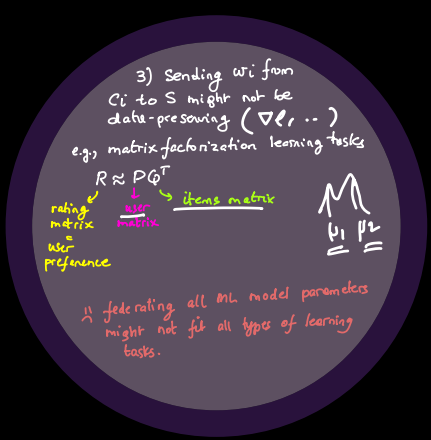


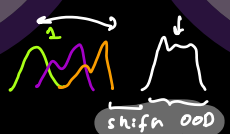
→ 1) boost the local model on local data

→ 2) boost the local model on global data









design new solutions

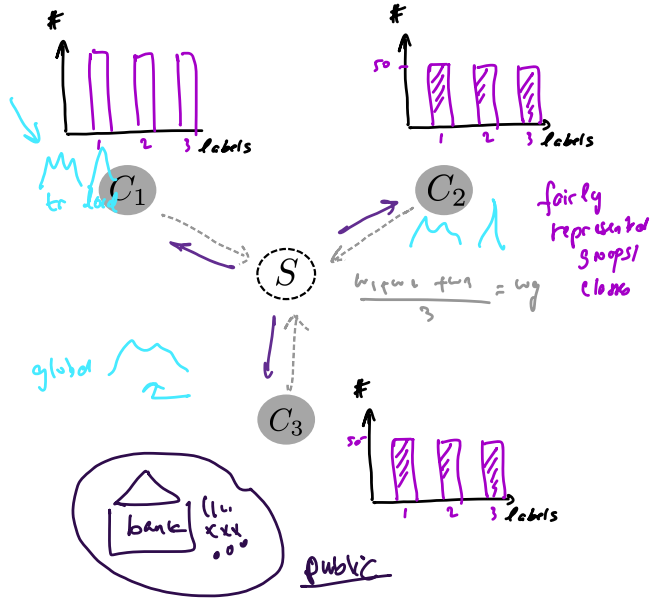
design new solutions

design new solutions

[FIFAI 2024] Introduction to Federated Learning (Day 2)

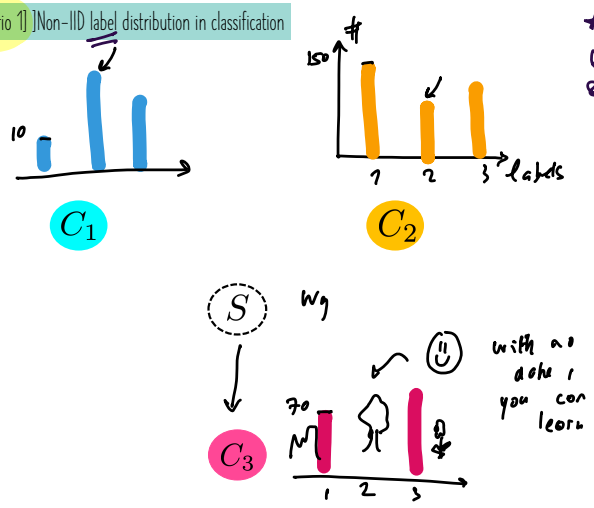
I- FedAvg (2016) [1]

a) IID label distribution across clients in classification

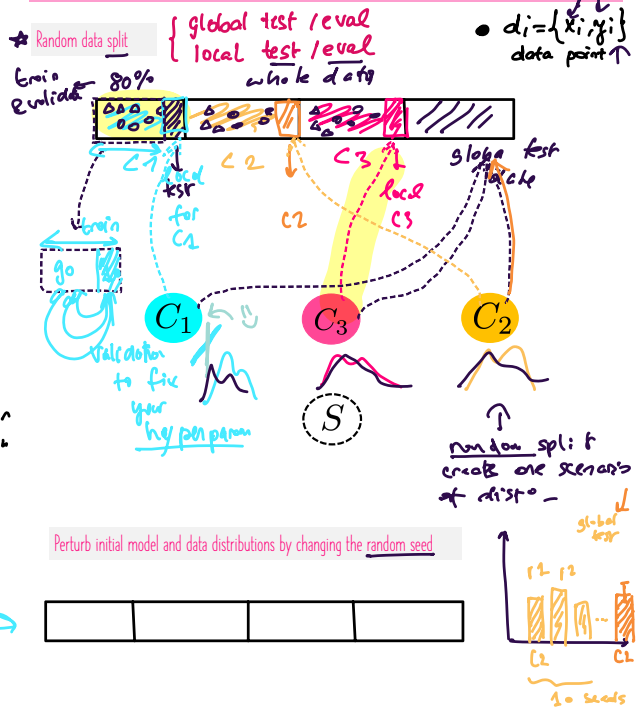


b) Non-IID scenario

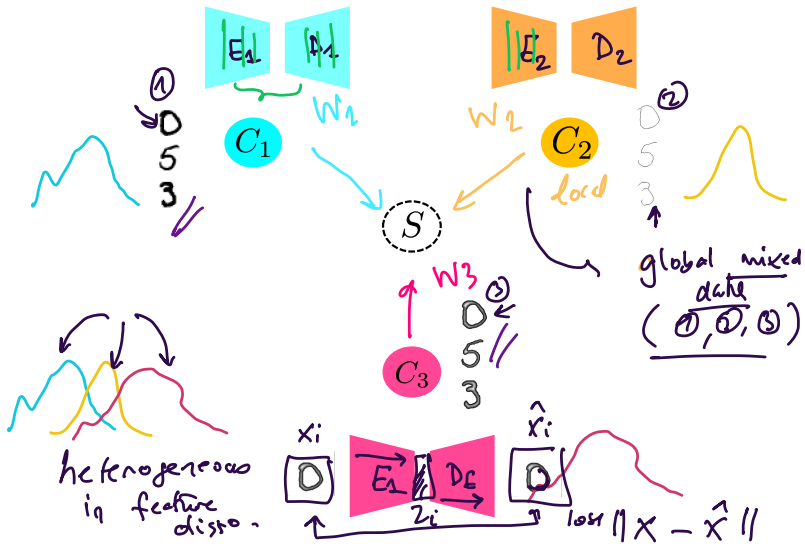
[Scenario 1] Non-IID label distribution in classification



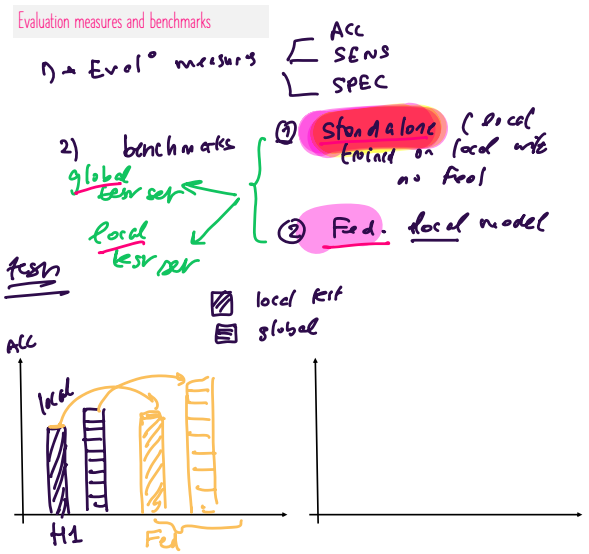
FL experiment setup for evaluation



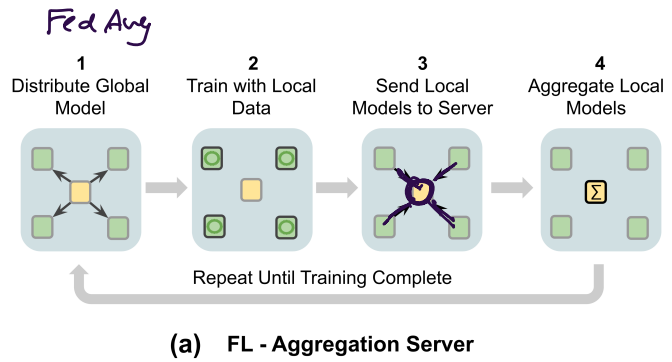
[Scenario 2] Non-IID feature distribution in generative learning



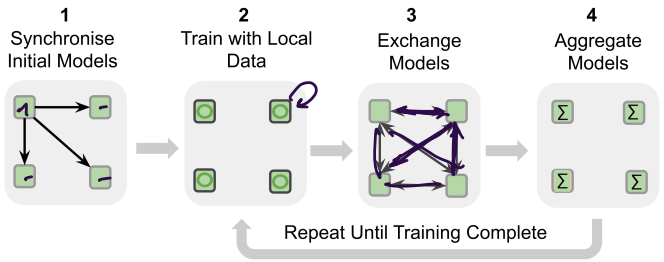
Evaluation measures and benchmarks



Diverse learning paradigms: centralized versus federated [2]



(a) FL - Aggregation Server



(b) FL - Peer to Peer

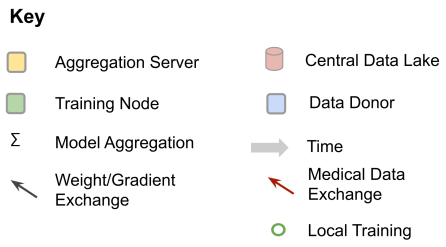


Image source: https://www.nature.com/articles/s41746-020-00323-1

Personal reflections and notes

[FIFAI 2024] Introduction to Federated Learning (Day 2)

II- FedProx [3]: A boost to fairness in federated learning

Algorithm 1 Federated Averaging (`FedAvg`)

Input: $K, T, \eta, E, w^0, N, p_k, k = 1, \cdots, N$
for $t = 0, \cdots, T-1$ **do**
  Server selects a subset S_t of K devices at random (each device k is chosen with probability p_k)
  Server sends w^t to all chosen devices
  Each device $k \in S_t$ updates w^t for E epochs of SGD on F_k with step-size η to obtain w_k^{t+1}
  Each device $k \in S_t$ sends w_k^{t+1} back to the server
  Server aggregates the w's as $w^{t+1} = \frac{1}{K}\sum_{k \in S_t} w_k^{t+1}$
end for

Algorithm 2 `FedProx` (Proposed Framework)

Input: $K, T, \mu, \gamma, w^0, N, p_k, k = 1, \cdots, N$
for $t = 0, \cdots, T-1$ **do**
  Server selects a subset S_t of K devices at random (each device k is chosen with probability p_k)
  Server sends w^t to all chosen devices
  Each chosen device $k \in S_t$ finds a w_k^{t+1} which is a γ_k^t-inexact minimizer of: $w_k^{t+1} \approx \arg\min_w h_k(w; w^t) = F_k(w) + \frac{\mu}{2}\|w - w^t\|^2$
  Each device $k \in S_t$ sends w_k^{t+1} back to the server
  Server aggregates the w's as $w^{t+1} = \frac{1}{K}\sum_{k \in S_t} w_k^{t+1}$
end for

⊗ Cons:

Statistical heterogeneity: is common with data being non-identically distributed between devices.

Existing solutions: There are also heuristic approaches that aim to tackle statistical heterogeneity by sharing the local device data or server-side proxy data
→ However, these methods may be unrealistic: in addition to imposing burdens on network bandwidth, sending local data to the server violates the key privacy assumption of federated learning
→ Sending globally-shared proxy data to all devices requires effort to carefully generate or collect such auxiliary data.

System heterogeneity: is also a critical concern in federated networks. The storage, computational, and communication capabilities of each device in federated networks may differ due to variability in hardware (CPU, memory), network connectivity (3G, 4G, 5G, wifi), and power (battery level).

Existing solutions: ignore the more constrained devices (stragglers) failing to complete a certain amount of training (unfair!)
→ However, this can have negative effects on convergence as it limits the number of effective devices contributing to training,
→ May induce bias in the device sampling procedure if the dropped devices have specific data characteristics.

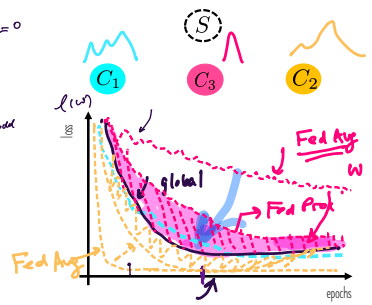
Proposed solution (FedProx)

Challenge 1 to solve: With dissimilar (heterogeneous) local objectives F_k, a larger number of local epochs may lead each device towards the optima of its local objective as opposed to the global objective-potentially hurting convergence or even causing the method to diverge.
Challenge 2 to solve: In federated networks with heterogeneous systems resources, setting the number of local epochs to be high may increase the risk that devices do not complete training within a given communication round and must therefore drop out of the procedure.
Proposed solution: merge local models by accounting for their heterogeneity and tolerating partial work → allowing for variable amounts of work to be performed locally across devices based on their available systems resources, and then aggregate the partial solutions sent from the stragglers (as compared to dropping these devices) (fairer!)

local updates. In particular, instead of just minimizing the local function $F_k(\cdot)$, device k uses its local solver of choice to approximately minimize the following objective h_k:

$$\min_w h_k(w; w^t) = F_k(w) + \frac{\mu}{2}\|w - w^t\|^2. \quad (2)$$

The proximal term is beneficial in two aspects: (1) It addresses the issue of statistical heterogeneity by restricting the local updates to be closer to the initial (global) model without any need to manually set the number of local epochs. (2) It allows for safely incorporating variable amounts of local work resulting from systems heterogeneity. We summarize the steps of `FedProx` in Algorithm 2.



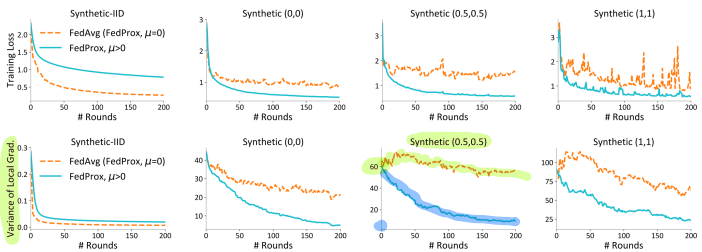
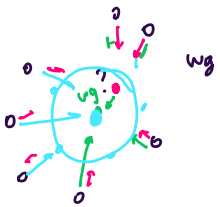


Figure 2. Effect of data heterogeneity on convergence. We remove the effects of systems heterogeneity by forcing each device to run the same amount of epochs. In this setting, `FedProx` with $\mu = 0$ reduces to `FedAvg`. (1) Top row: We show training loss (see results on testing accuracy in Appendix C.3, Figure 6) on four synthetic datasets whose statistical heterogeneity increases from left to right. Note that the method with $\mu = 0$ corresponds to `FedAvg`. Increasing heterogeneity leads to worse convergence, but setting $\mu > 0$ can help to combat this. (2) Bottom row: We show the corresponding dissimilarity measurement (variance of gradients) of the four synthetic datasets. This metric captures statistical heterogeneity and is consistent with training loss — smaller dissimilarity indicates better convergence.



Personal reflections and notes

III- FedALA [4]

FedALA: Adaptive Local Aggregation for Personalized Federated Learning

Jianqing Zhang[1] Yang Hua[2] Hao Wang[3]

Tao Song[1] Zhengui Xue[1] Ruhui Ma[1] Haibing Guan[1]

[1] 上海交通大学 SHANGHAI JIAO TONG UNIVERSITY [2] QUEEN'S UNIVERSITY BELFAST [3] LSU

Source link: https://github.com/TsingZ0/FedALA/blob/main/FedALAOral.pdf

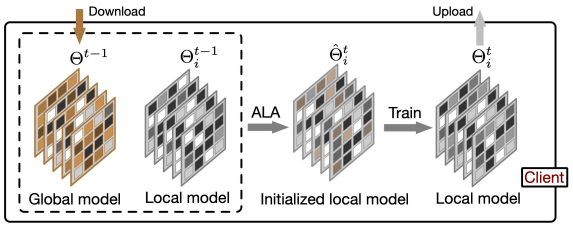


Figure 1: Local learning process on client i in the t-th iteration. Specifically, client i downloads the global model from the server, locally aggregates it with the old local model by ALA module for local initialization, trains the local model, and finally uploads the trained local model to the server.

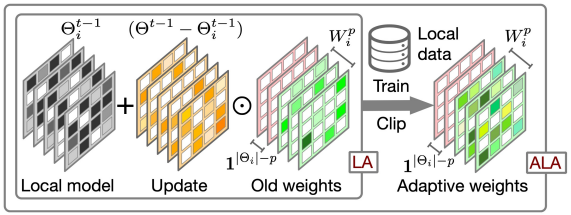


Figure 2: The learning process in ALA. LA denotes "local aggregation". Here, we consider a five-layer model and set $p = 3$. The lighter the color, the larger the value.

Personal reflections and notes

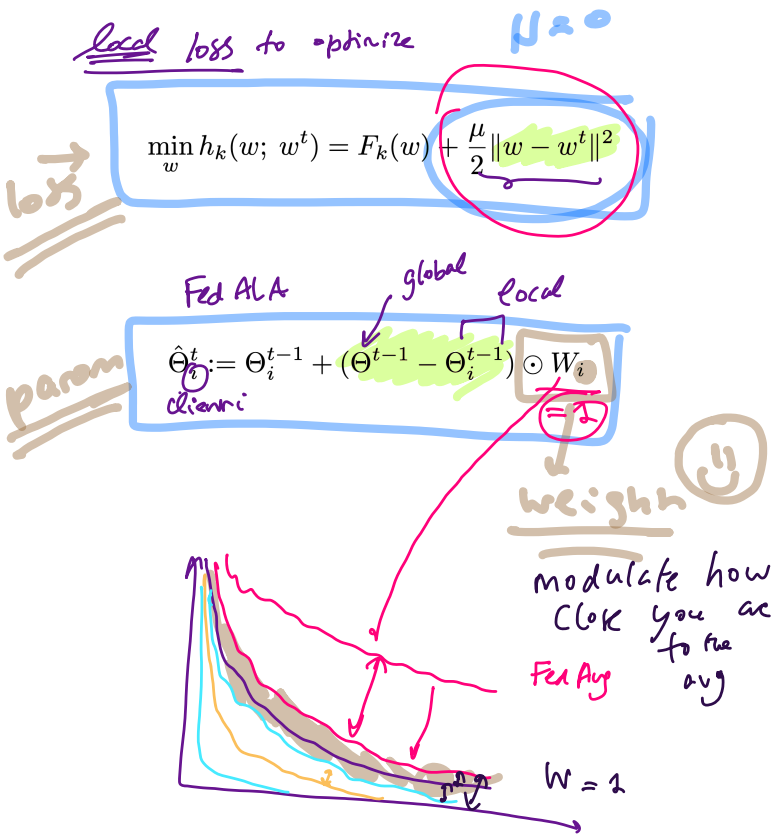
[FIFAI 2024] Introduction to Federated Learning (Day 2)

Federated learning

FedProx vs FedALA

2020 2023

Key idea: Regularize/control the local model using the global model (how far am I from the average)?
Stay personalized (retain local information) without diverging too much from the global/universal model.



Personal reflections and notes

References:

1. FedAvg: McMahan et al., 2017. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR. https://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf
2. Other federation topologies: Rieke et al., 2020. The future of digital health with federated learning. NPJ digital medicine, 3(1), pp.1-7. https://www.nature.com/articles/s41746-020-00323-1
3. FedProx: Li et al., 2020. Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems, 2, pp.429-450. https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf; https://github.com/litian96/FedProx
4. FedALA: Zhang et al., 2023. FedALA: Adaptive local aggregation for personalized federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 9, pp. 11237-11244). https://dl.acm.org/doi/10.1609/aaai.v37i9.26330, https://github.com/TsingZ0/FedALA