An overview of Federated Learning

Definitions, methods and challenges

Jia Gu

Center for Statistical Sciences, Peking University

December 12, 2021

History

- The term **Federated Learning** was introduced in 2016 by McMahan et al. [McMahan et al., 2017] : "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."
- The remaining slides will be mainly based on two recent reviews on this topic: Li et al.[Li et al., 2020a] and Kairouz et al.[Kairouz and McMahan, 2021]. It is worth mentioning that the second paper was originated at the Workshop on Federated Learning and Analytics held June 17–18th, 2019, hosted at Google' s Seattle office.

Types of Federated Learning

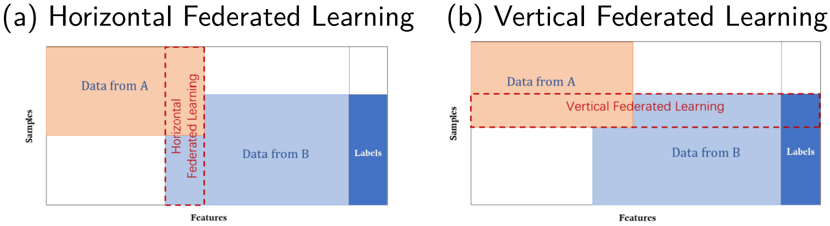


图 1: Main Types of Federated Learning. [Yang et al., 2019]

Problem Formulation

The canonical federated learning problem involves learning a *single, global* statistical model from data stored on tens to potentially millions of remote devices. In particular, we want to solve the following optimization problem:

$$\min_{w} \quad F(w), \text{ where } F(w) := \sum_{k=1}^{m} p_k F_k(w), \tag{1}$$

under the constraint that device-generated data is stored and processed locally, with only intermediate updates being communicated periodically with a central server. Here m is the number of devices, $p_k \geq 0$ and $\sum_{k=1}^{m} p_k = 1$, and F_k is the local objective function.

Core Challenges

In general, there are four core challenges in solving the distributed optimization problem posed in Equation (1), which make the federated setting distributed from other classical problems such as distributed learning:

1. Expensive Communication;
2. Systems Heterogeneity;
3. Statistical Heterogeneity;
4. Privacy Concerns.

1. Expensive Communication
2. Systems Heterogeneity
3. Statistical Heterogeneity
4. Privacy
5. Some Directions
6. Reference

Target

Communication is a critical bottleneck in federated learning. In fact, communication in the network can be slower than local computation by **many orders** of magnitude. Thus in order to fit a model to data generated by the devices in the federated network, it is necessary to develop **communication-efficient** methods which can

1. reduce the total number of communication rounds,
2. reduce the size of transmitted messages at each round.

The existing methods can be roughly grouped into

1. Local updating methods;
2. Compression schemes;
3. Decentralized training.

Local Updating

- For **convex** objects, distributed local updating **primal-dual** methods, have emerged as a popular way. The choice of primal or dual form depends on which form can be more easily decomposed into subproblems in the distributed setting.

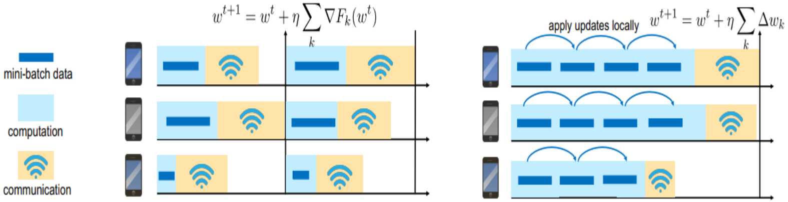


图 2: Comparison of schemes of distributed SGD (left) and Local Updating (right).

Compression Schemes

Model compression schemes such as **sparsification, subsampling and quantization** can significantly reduce the size of messages communicated at each round.

- Sparsification : communicating **low-precision** or **sparsified** versions (either by thresholding small entries or by random sampling) of the computed gradients. [Wang et al., 2018] reviewed the existing works and proposed a ATOMO framework which decomposes the gradient by some basis in an inner product space.
- Subsampling;
- Quantization: reducing the precison of data representation [Zhang et al., 2017].

Existing methods did not consider the low device participation in federated setting.

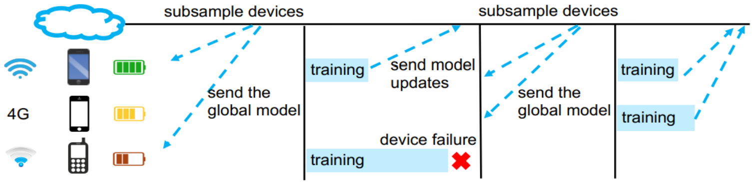
Decentralized Training

图 3: Star network (left) and decentralized network (right).

- Star network is dominant in federated learning.
- Related works either are restricted to the linear model (The CoLA framework under generalized linear model [He et al., 2018]), or require full device participation.

1. Expensive Communication
2. Systems Heterogeneity
3. Statistical Heterogeneity
4. Privacy
5. Some Directions
6. Reference

Systems Heterogeneity: description



- Asynchronous communication;
- Active device sampling;
- Fault tolerance.

Asynchronous Communication

While synchronous schemes are relatively simple and easy to derive theoretical guarantee, they are susceptible to **stragglers**.

- Typical asychronous scheme: the Asynchronous SGD "Hogwild!" [Niu et al., 2011] (In comparison to the parallel SGD [Zinkevich et al., 2010]);
- Existing works generally rely on **bounded-delay** assumptions, which can be unrealistic in federated settings, since here the delay may be on the order of hours to days, or completely unbounded.

Active Sampling

In federated networks, typically only **a small subset of devices** participate at each round of training.

- the most majority of federated methods are **passive**: they do not actively select which devices to participate.
- [Nishio and Yonetani, 2019] selected devices based on system resources (aiming to aggregate as many updates as possible) and [Kang et al., 2019] preferred higher-quality data.
- How to extend to dynamic (real-time) models instead of static models of the system?
- Can we actively sample devices based on some **statistical structure**?

Fault Tolerance

- Can be traced back to the classical Byzantine Generals Problem proposed by Leslie Lamport (the initial developer of the LaTeX);

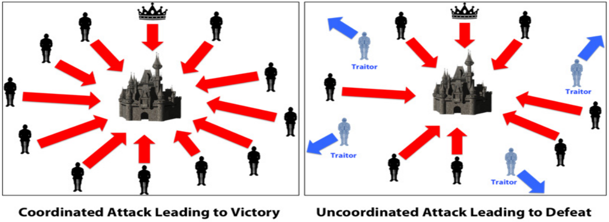
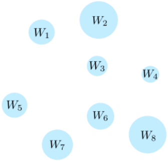
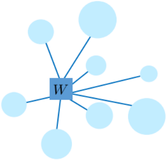


图 4: The Byzantine Generals Problem.

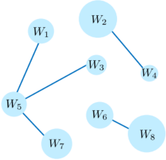
Modeling Heterogeneous Data



(a) Learn personalized models for each device; do not learn from peers.



(b) Learn a global model; learn from peers.



(c) Learn personalized models for each device; learn from peers.

Challenges arise when training federated models from data that is non-IID distributed across devices, especially in terms of

- modeling the data (depicted in the above figure);
- analyzing the convergence behavior of associated training procedures.

Multiple-Task Learning (MTL)

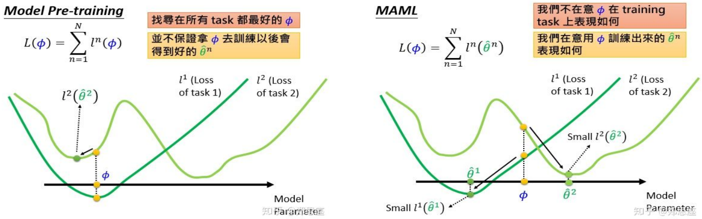
- MTL aims to leverage useful information contained in **multiple related tasks** to help improve the generalization performance of all the tasks. See [Caruana, 1997, Zhang and Yang, 2021].
- Typical Formulation:

$$\min_{W,\Omega} \left\{ \sum_{t=1}^{m} \sum_{i=1}^{n_t} l_t(w_t^T x_t^i, y_t^i) + \mathcal{R}(W, \Omega) \right\}. \tag{2}$$

  Example of $\mathcal{R}$: $\mathcal{R}(W, \Omega) = \lambda_1 tr(W \Omega W^T) + \lambda_2 \|W\|_F^2$.
  The matrix Ω models relationships amongst tasks, and is either **known a priori** or **estimated simultaneously learning task models**.
- In Federated Learning, [Smith et al., 2017] proposed a Mocha framework based on MTL.

Meta Learning (Learning-to-Learn, Few Shots Learning)

- Contrasting to the "data-hungery" traditional machine learning, when we perform a new task, we want to lean it **more efficiently and effectively.**
- [Finn et al., 2017] proposed the famous **MAML** framework.



Both strategies are **expensive to generalize to massive networks**.

Comparison of MTL and Meta-Learning

In summary, the difference between multi-task learning and meta-learning is:

- in multitask learning, your goal would be to try to solve all of the training tasks;
- whereas in meta-learning your goal is to use these training tasks in order to solve new tasks with a small amount of data, so the model could evaluate new tasks and quickly learn new tasks.
- see Stanford CS330: Multi-Task and Meta-Learning, 2019 lectured by Chelsea Finn. https://www.youtube.com/watch?v=0rZtSwNOTQo&list=PLoROMvodv4rMC6zfYmnD7UG3LVvwaITY5&index=2

Fairness

In practice, the learned model may become **biased towards devices with larger amounts of data.**

- Agnostic Federated Learning: For devices with **insufficient samples**, will federated learning outperforms purely local optimization? **No guarantee.** [Mohri et al., 2019] formulated the problem into a min-max problem, which aims to optimize the worst case and thus tends to give a conservative solution.
- q-Fair Federated Learning (q-FFL): [Li et al., 2020b] proposed q-FFL in which devices with higher loss are given higher relative weight to encourage less variance in the final accuracy distribution.

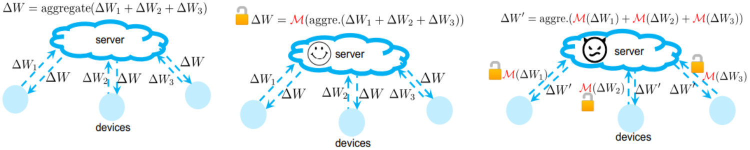
Main Strategies on Privacy-Preserving

- Differential Privacy: [Dwork et al., 2006] proposed the $\epsilon-$**privacy**. A randomized algorithm $\mathcal{A}$ satisfies $\epsilon-$privacy iff

$$P\{\mathcal{A}(D_1) \in S\} \leq exp(\epsilon)P\{\mathcal{A}(D_2) \in S\}$$

  for all subsets S and datasets D_1 and D_2 which differ on **a single element**.
- Homomorphic encryption: Compute on encrypted data.
- Secure Multiparty computation (SMC): Dating back to [Yao, 1982].

Privacy in Federated Learning



(a) Federated learning without additional privacy protection mechanisms.

(b) Global privacy, where a trusted server is assumed.

(c) Local privacy, where the central server might be malicious.

图 5: Privacy-enhancing mechanisms.

Current approaches may not be applicable to large-scale machine learning scenarios as they incur **substantial additional communication and computation costs**.

Some directions

- Communication: How to systematically analyze the trade-off between accuracy and communication for the Federated learning algorithms?
- Heterogeneity:
  1. Do simple diagnostics exist to quickly determine the level of heterogeneity in federated networks?
  2. How to better design the framework to encourage fairness among devices without compromising much efficiency?
- Unsupervised , semi-supervised problems or online learning problems in Federated Learning.
- Compression schemes with differential privacy guarantee.

[Caruana, 1997] Caruana, R. (1997).
Multitask learning.
Machine Learning, 28:41–75.

[Dwork et al., 2006] Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006).
Calibrating noise to sensitivity in private data analysis.
volume Vol. 3876, pages 265–284.

[Finn et al., 2017] Finn, C., Abbeel, P., and Levine, S. (2017).
Model-agnostic meta-learning for fast adaptation of deep networks.
In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

[He et al., 2018] He, L., Bian, Y. A., and Jaggi, M. (2018).
Cola: Decentralized linear learning.
Advances in Neural Information Processing Systems.

[Kairouz and McMahan, 2021] Kairouz, P. and McMahan, H. (2021).
Advances and open problems in federated learning.
Foundations and Trends in Machine Learning, 14:1–210.

[Kang et al., 2019] Kang, J., Xiong, Z., Niyato, D., Yu, H., Liang, Y.-C., and Kim, D. I. (2019).
Incentive design for efficient federated learning in mobile networks: A contract theory approach.
In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, pages 1–5.

[Li et al., 2020a] Li, T., Sahu, A., Talwalkar, A., and Smith, V. (2020a).
Federated learning: Challenges, methods, and future directions.
IEEE Signal Processing Magazine, 37:50–60.

[Li et al., 2020b] Li, T., Sanjabi, M., Beirami, A., and Smith, V. (2020b).
Fair resource allocation in federated learning.
In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.
OpenReview.net.

[McMahan et al., 2017] McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. (2017).
Communication-Efficient Learning of Deep Networks from Decentralized Data.
Proceedings of Machine Learning Research, 54:1273–1282.

[Mohri et al., 2019] Mohri, M., Sivek, G., and Suresh, A. T. (2019).
Agnostic federated learning.
Proceedings of Machine Learning Research, 97:4615–4625.

[Nishio and Yonetani, 2019] Nishio, T. and Yonetani, R. (2019).
Client selection for federated learning with heterogeneous resources in mobile edge.
In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7.

[Niu et al., 2011] Niu, F., Recht, B., Ré, C., and Wright, S. (2011).
HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent.
Advances in Neural Information Processing Systems, 24.

[Smith et al., 2017] Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. (2017).
Federated Multi-Task Learning.
Advances in Neural Information Processing Systems.

[Wang et al., 2018] Wang, H., Sievert, S., Charles, Z., Papailiopoulos, D., and Wright, S. (2018).
ATOMO: Communication-efficient Learning via Atomic Sparsification.
Advances in Neural Information Processing Systems.

[Yang et al., 2019] Yang, Q., Liu, Y., Chen, T., and Tong, Y. (2019).
Federated machine learning: Concept and applications.
ACM Transactions on Intelligent Systems and Technology, 10:1–19.

[Yao, 1982] Yao, A. C. (1982).
Protocols for secure computations.
In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 160–164.

[Zhang et al., 2017] Zhang, H., Li, J., Kara, K., Alistarh, D., Liu, J., and Zhang, C. (2017).
ZipML: Training linear models with end-to-end low precision, and a little bit of deep learning.
Proceedings of Machine Learning Research, 70:4035–4043.

[Zhang and Yang, 2021] Zhang, Y. and Yang, Q. (2021).
A survey on multi-task learning.
IEEE Transactions on Knowledge and Data Engineering, PP.

[Zinkevich et al., 2010] Zinkevich, M., Weimer, M., Smola, A., and Li, L. (2010).
Parallelized Stochastic Gradient Descent.
Advances in Neural Information Processing Systems, 23:2595–2603.

Thanks!