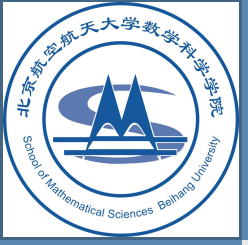
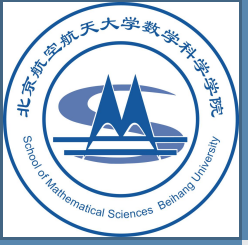
联邦学习中的优化问题

Optimizations in Federated Learning

WEN Hao

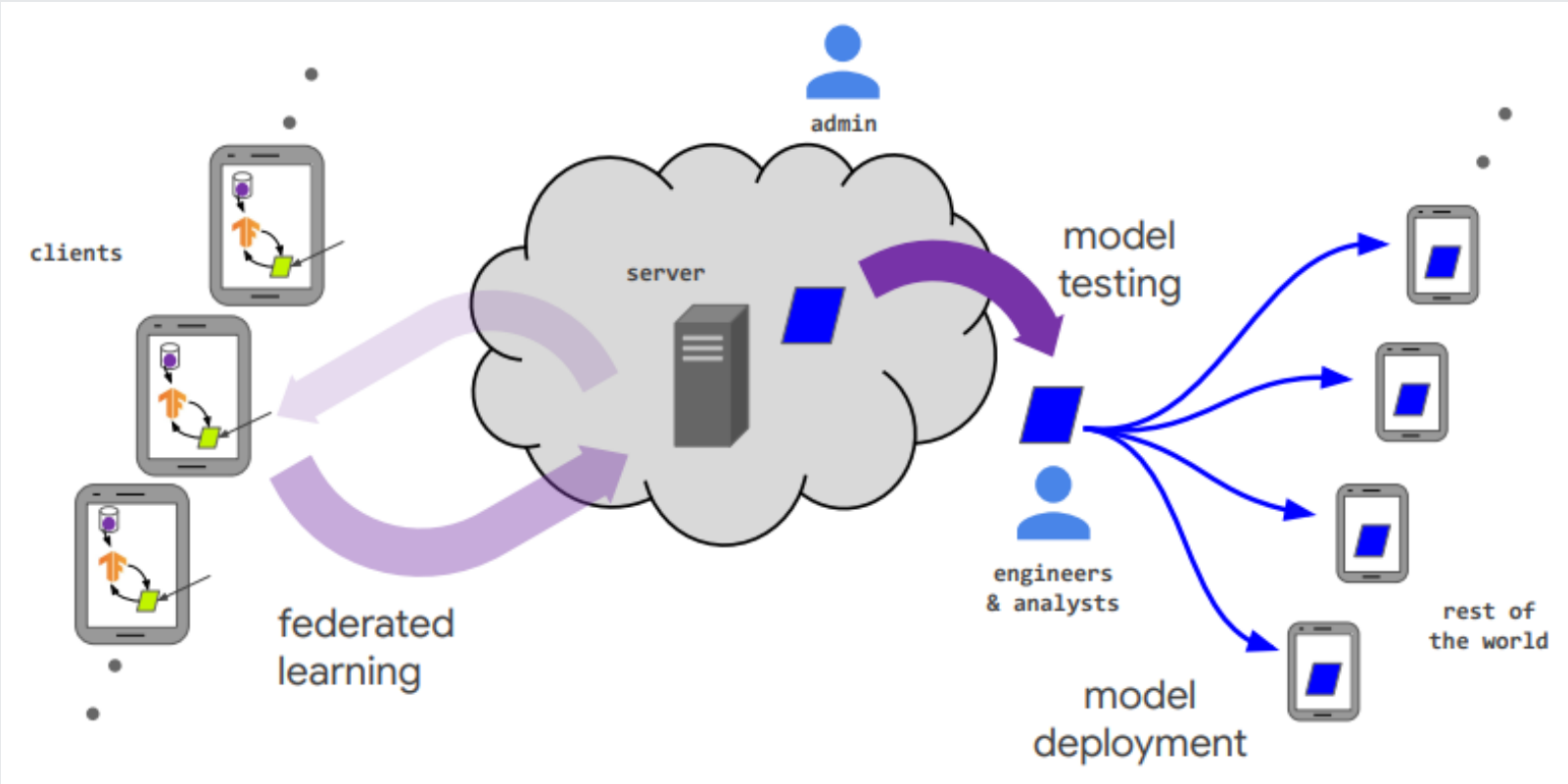
2023-02-04



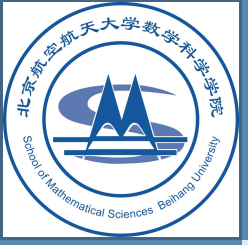


引言

联邦学习（Federated Learning）来源于机器（深度）学习模型分布式（Distributed）训练的需求



图片来源：[1] Kairouz et al., Advances and open problems in federated learning, 2019



引言

这种分布式训练的需求多是当多个数据拥有方想要联合他们各自的数据训练机器学习模型，由于涉及隐私和数据安全等法律问题，或是数据庞大且过于分散导致的可行性问题，而不能将数据集中到一起进行模型训练而产生的。随着越来越严格的数据隐私方面的法律法规的施行，这种需求会越来越大。

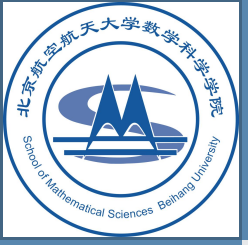
引言

这种分布式训练的需求多是当多个数据拥有方想要联合他们各自的数据训练机器学习模型，由于涉及隐私和数据安全等法律问题，或是数据庞大且过于分散导致的可行性问题，而不能将数据集中到一起进行模型训练而产生的。随着越来越严格的数据隐私方面的法律法规的施行，这种需求会越来越大。

一般来说，在联邦学习的框架下，数据拥有方在不用给出己方数据的情况下，也可进行模型训练得到公共的模型 M_{fed}，使得模型 M_{fed}，与将数据集中到一起进行训练能得到的模型 M，二者的预测值的偏差的期望能足够小。

$$\mathop{\mathbb{E}}_{z\sim\mathcal{D}}\|M_{fed}(z)-M(z)\|\leqslant\delta$$

注：以下将数据拥有方统称为“节点”



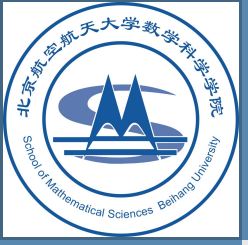
引言

Federated Learning 这个名词首次由 Google 的研究人员 McMahan 等人在文章 [2] *Communication-Efficient Learning of Deep Networks from Decentralized Data* (2016) 中提出。

引言

Federated Learning 这个名词首次由 Google 的研究人员 McMahan 等人在文章 [2] *Communication-Efficient Learning of Deep Networks from Decentralized Data* (2016) 中提出。

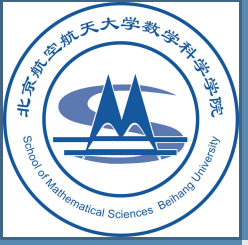
相关的分布式的模型训练（优化）方法则可以追溯到更早的时间，例如 Boyd 等人的著作 [3] *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers* (2010)



联邦学习的定义

综述文章 [1] Advances and open problems in federated learning (2019) 给联邦学习下过如下的定义

> *"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."*

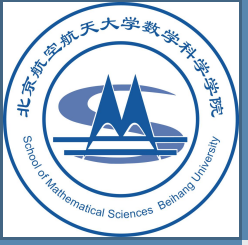


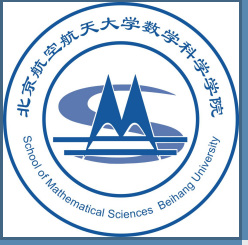
联邦学习研究的一些核心的问题

- **EE (Efficiency & Effectiveness)**
  - **Optimization**

联邦学习研究的一些核心的问题

- **EE (Efficiency & Effectiveness)**
  - **Optimization**
  - Compression
- Privacy & Security
  - Differential Privacy (DP)
  - Secure Multi-Party Computing (SMPC)
  - Trusted Execution Environment (TEE)
  - Homomorphic Encryption (HE)
- Applications
  - Medical
  - Recommendation
  - Finance
- etc.





问题描述

一般来说，联邦学习中我们考虑的是如下的优化问题

$$\text{minimize} \quad f(x) = \mathop{\mathbb{E}}_{i\sim\mathcal{P}}[f_i(x)]$$

$$\text{where} \quad f_i(x) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}_i}[\ell_i(x;z)]$$

这里的 $\mathcal{P}$ 为节点的分布，$\mathcal{D}_i$ 为节点 i 上的数据分布，ℓ_i 为损失函数。

问题描述

一般来说，联邦学习中我们考虑的是如下的优化问题

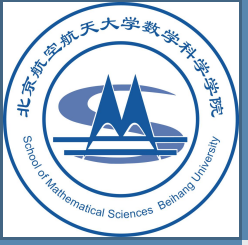
$$\text{minimize} \quad f(x) = \mathop{\mathbb{E}}_{i\sim\mathcal{P}}[f_i(x)]$$

$$\text{where} \quad f_i(x) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}_i}[\ell_i(x;z)]$$

这里的 $\mathcal{P}$ 为节点的分布，$\mathcal{D}_i$ 为节点 i 上的数据分布，ℓ_i 为损失函数。

或者更简单地，考虑如下的优化问题

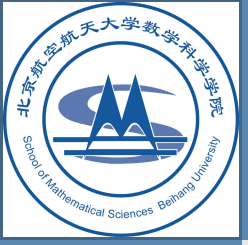
$$\text{minimize} \quad f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$$



问题描述

要注意的是，联邦学习中的“节点”（数据拥有方）意义比较宽泛，涵盖很多场景，例如

- 多家医院的服务器
- 多个移动设备（edge device）



问题描述

要注意的是，联邦学习中的“节点”（数据拥有方）意义比较宽泛，涵盖很多场景，例如

- 多家医院的服务器
- 多个移动设备（edge device）

前者一般被称作 cross-silo，后者一般被称作 cross-device。在 cross-device 的场景下，一般来说，通信效率才是整个系统的瓶颈所在，此外还需要考虑掉队者（stragglers）等问题。

数据分布

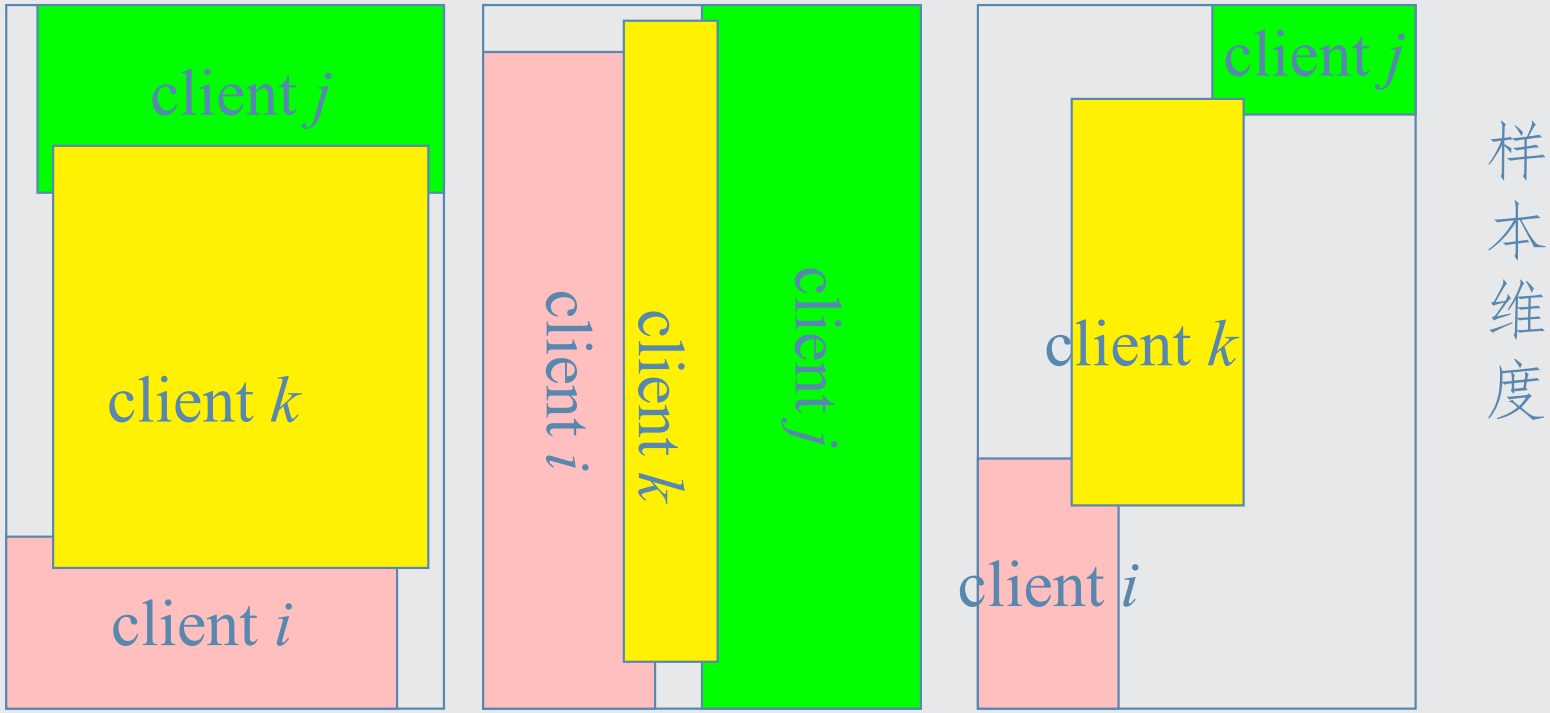
在真实场景下，各个节点上的数据的分布 $\mathcal{D}_i$ 一般不是独立同分布的（non-IID, 或称 heterogeneous）。这种数据分布的各向异性将联邦学习分为了 3 类

- 横向联邦学习：各节点的样本重叠度低，样本特征重叠度高
- 纵向联邦学习：各节点的样本重叠度高，样本特征重叠度低
- 迁移联邦学习：各节点的样本重叠度低，样本特征重叠度低

同一种算法（例如 SVM）在不同类型的联邦学习模式下，对应的优化问题的具体形式会稍有不同。

数据分布

横向联邦学习　纵向联邦学习　迁移联邦学习

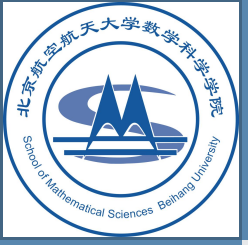


数据分布

non-IID 数据分布下，算法的收敛性分析相比 IID 数据下要更加困难，需要更多额外的假设，对节点之间的数据分布的不同性（dissimilarity）进行定量上的限制。

一般地，这种限制以 gradient variance 给出，例如 bounded inter-client gradient variance (BCGV):

$$\mathop{\mathbb{E}}_{i\sim\mathcal{P}}\|\nabla f_i(x)-\nabla f(x)\|_2^2\leqslant \text{const}\quad \text{for all } x$$

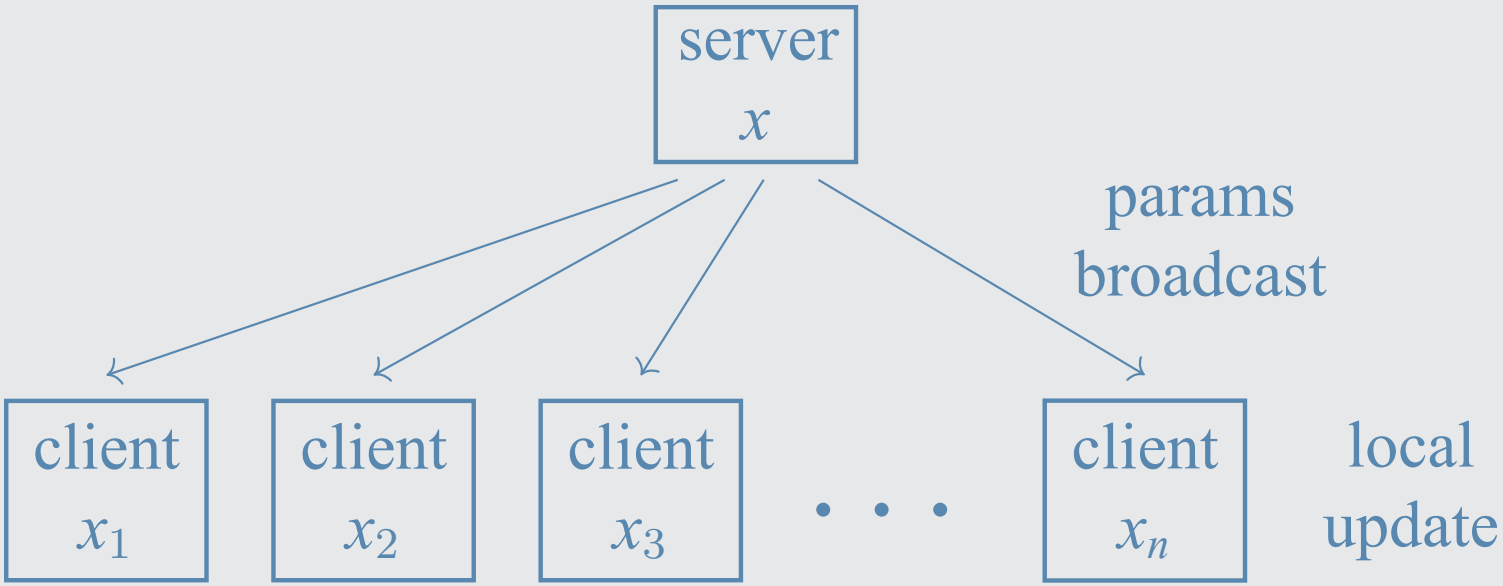


联邦学习的一般性框架（流程）

- client selection
- parameter broadcast
- **client computation (local parameter update)**
- parameter aggregation
- **server computation (global parameter update)**

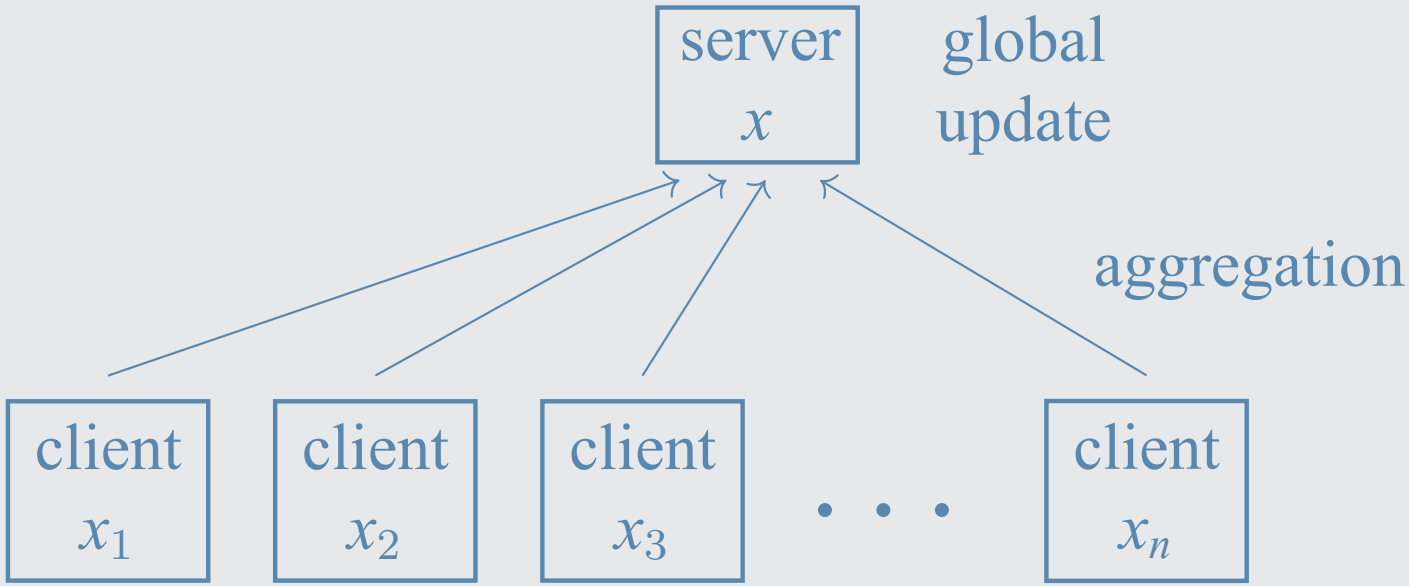
联邦学习的一般性框架（流程）

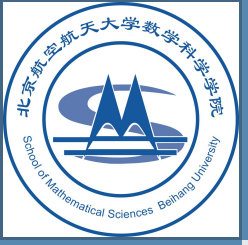
Broadcast and local update:



联邦学习的一般性框架（流程）

Aggregate and global update:





从 FedAvg 到 FedOpt

Google 研究人员 McMahan 等人在文章 [2](2016) 中考察了普通的 SGD 在分布式下的平凡推广 FedSGD，即在每次循环中，节点执行一次 full-batch gradient descent，并做出了进一步推广，提出了 FedAvg 算法。FedAvg 的具体做法就是在每次循环的 client local computation 中，执行多步 mini-batch SGD。这样，既降低了通信开销（communication-efficient），同时也在实验上观察到了模型效果的提升。

从 FedAvg 到 FedOpt

Algorithm 1: FedAvg

Server executes:

initialize parameters x_0, learning rate η, batch size B;

for *each round* $t = 0, 1, \cdots, T-1$ **do**

$S_t \leftarrow$ (random set of clients)

for *each client* $i \in \mathcal{S}_t$ **in parallel do**

$x_{i,t} \leftarrow \textbf{ClientUpdate}(i, x_t)$

$x_{t+1} \leftarrow \frac{1}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} x_{i,t}$

ClientUpdate(i, x): // on client i

$\mathcal{B} \leftarrow$ (split $\mathcal{P}_i$ into batches of size B)

for *local step* $k = 0, 1 \cdots, K-1$ **do**

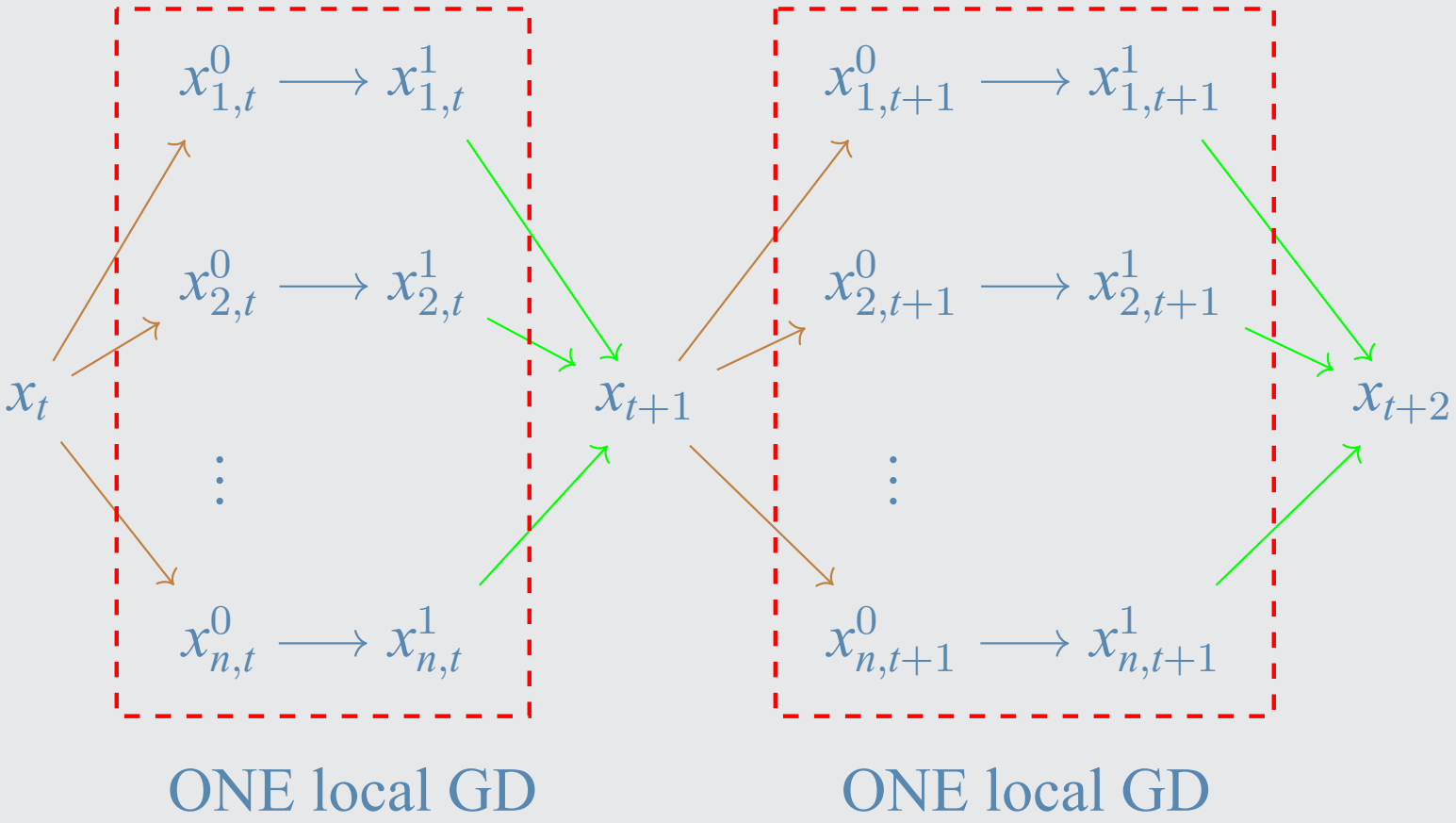
for *batch* $b \in \mathcal{B}$ **do**

$x \leftarrow x - \eta \nabla \ell_i(x; b)$

return x

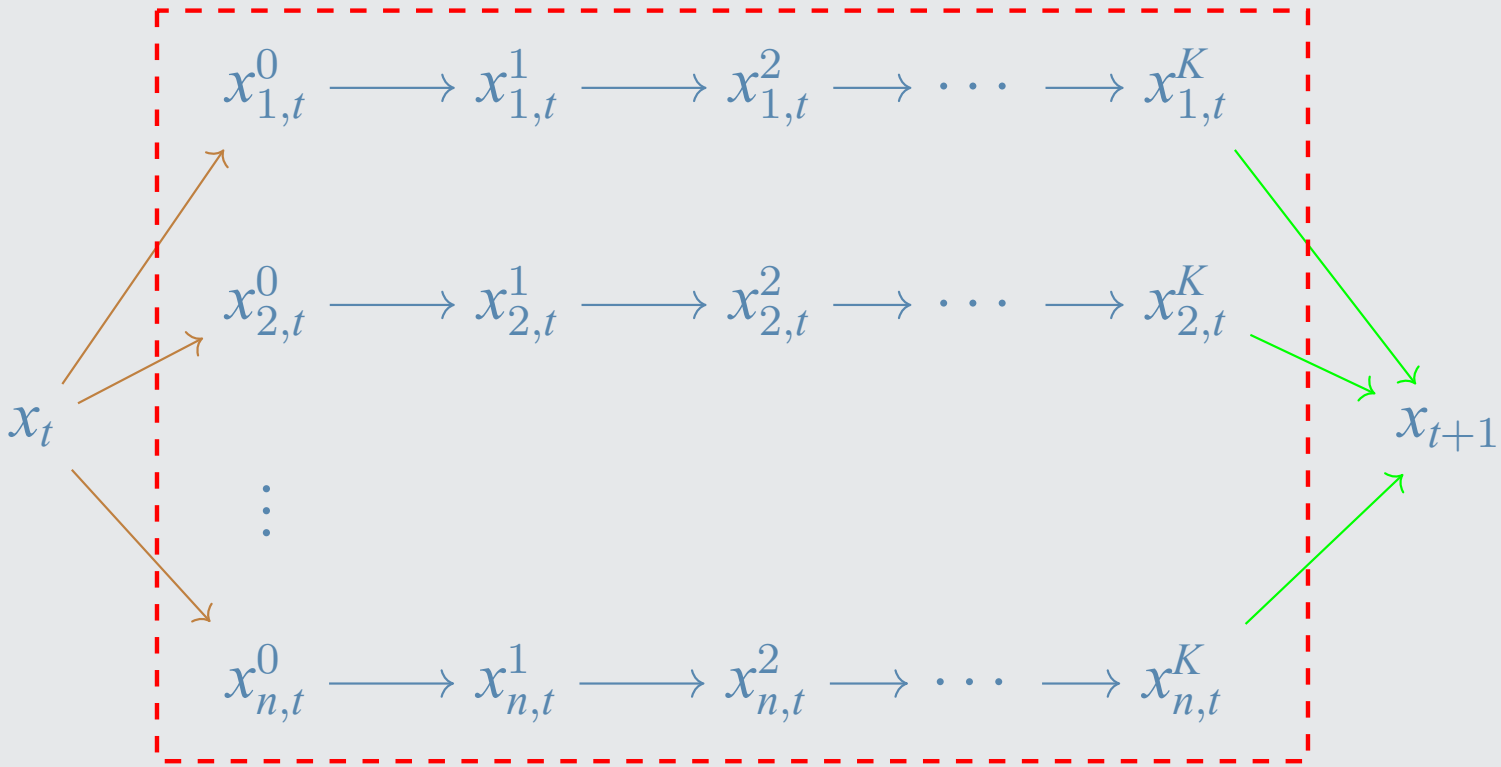
FedSGD – baseline

FedSGD: 在每个节点执行一次 full-batch GD 之后，即进行模型同步（平均）。

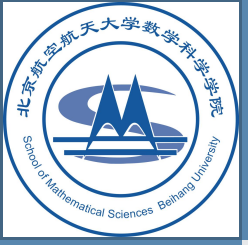


FedAvg

FedAvg: 每个节点执行 K 个 mini-batch SGD 之后，进行模型同步（平均）。在通信量大大降低的情况下，模型效果也得到了一定提升。



K local mini-batch SGD



FedOpt

McMahan 等人进一步在文章 [4](2020) 中，将 Adam 等自适应、（momentum）加速算法融入联邦学习中，提出了更一般的 FedOpt 框架。

FedOpt

Algorithm 2: FedOpt

Input: parameters x_0, methods **ServerOpt**, **ClientOpt**, learning rate (schedule) η_g, η_l

for *each round* $t = 0, 1, \cdots, T-1$ **do**

  $S_t \leftarrow$ (random set of clients)

  $x_{i,t}^0 \leftarrow x_t$

  for *each client* $i \in S_t$ **in parallel do**

    for *local step* $k = 0, 1, \cdots, K-1$ **do**

      Compute unbiased estimate $g_{i,t}^k$ of $\nabla f_i(x_{i,t}^k)$

      $x_{i,t}^{k+1} \leftarrow \textbf{ClientOpt}(x_{i,t}^k, g_{i,t}^k, \eta_l, t)$

    $\Delta_{i,t} \leftarrow x_{i,t}^K - x_t$

  $\Delta_t \leftarrow \text{aggregate}(\{\Delta_{i,t}\}_{i \in \mathcal{S}_t}) \quad (\text{e.g.} \frac{1}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} \Delta_{i,t})$

  $x_{t+1} \leftarrow \textbf{ServerOpt}(x_t, \Delta_t, \eta_g, t)$

FedOpt

一般来说，

- unbiased gradient estimate + **ClientOpt**:

  (local) mini-batch SGD,

  i.e. $x_{i,t}^{k+1} = x_{i,t}^{k} - \eta_l g_{i,t}^{k}$

- **ServerOpt**:

  Avg, Adagrad, Yogi, Adam, etc.

  后面几种算法还需要额外的一些超参数，例如 decay parameters $\beta_1, \beta_2 \in [0, 1)$ 以及状态量 momentum v_t。

FedAdagrad, FedAdam

Algorithm 3: FedAdagrad, FedAdam

for *each round* $t = 0, 1, \cdots, T-1$ **do**

  $S_t \leftarrow$ (random set of clients), $x_{i,t}^0 \leftarrow x_t$

  for *each client* $i \in S_t$ **in parallel do**

    for *local step* $k = 0, 1, \cdots, K-1$ **do**

      Compute unbiased estimate $g_{i,t}^k$ of $\nabla f_i(x_{i,t}^k)$

      $x_{i,t}^{k+1} \leftarrow x_{i,t}^k - \eta_l g_{i,t}^k$

  $\Delta_{i,t} \leftarrow x_{i,t}^K - x_t$

$\Delta_t \leftarrow \beta_1 \Delta_{t-1} + ((1-\beta_1)/|\mathcal{S}_t|) \sum_{i \in \mathcal{S}_t} \Delta_{i,t}$

$v_t \leftarrow v_{t-1} + \Delta_t^2$ **(FedAdagrad)**

$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2)\Delta_t^2$ **(FedAdam)**

$x_{t+1} \leftarrow x_t + \eta_g \Delta_t / (\sqrt{v_t} + \tau)$

Non-IID 数据分布下的特殊处理

对于 Non-IID 的数据分布，还有一些研究人员对于优化算法做了特殊的处理，例如

- proximal term (FedProx [5], 2018): 在 client local update 时，在目标函数上添加 proximal term

$$x_{i,t+1} \approx \arg\min_x f_i(x) + \frac{\mu}{2}\|x - x_t\|_2^2$$

- variance reduction (SCAFFOLD [6], 2019)：在 server 以及 client 额外的参数 c, c_i，与模型参数 x 一起更新。在进行 local mini-batch SGD 时，用这些参数对 gradient 进行修正：

$$x_{i,t}^{k+1} \leftarrow x_{i,t}^k - \eta_l(g_{i,t}^k - c_{i,t} + c_t)$$

收敛性

在假设损失函数 $\ell_i(x;z)$ 有 Lipschetz 光滑性条件

$$\|\nabla_x \ell_i(x;z) - \nabla_y \ell_i(y;z)\| \leqslant L \cdot \|x-y\|$$

节点内梯度方差（intra-client gradient variance）

$$\mathop{\mathbb{E}}_{z\sim\mathcal{D}_i} \|\nabla_x \ell_i(x;z) - \nabla f_i(x)\|^2 \leqslant \sigma^2$$

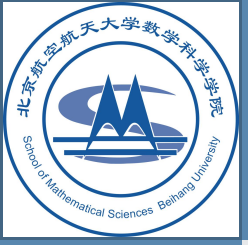
的限制条件下，我们考察误差

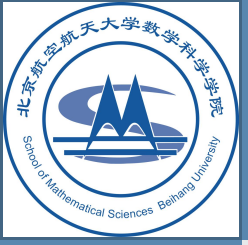
$$err = |\mathbb{E}[f(x_T)] - f(x^*)|$$

一般有（FedAvg）

$$err = \mathcal{O}\left(\frac{HN}{T} + \frac{\sigma}{\sqrt{TKN}}\right)$$

? TODO: more convergence results





Consensus and Sharing

更传统一些的处理分布式的机器学习模型训练的方法是（例如 [3]）将我们要优化的（简化的）问题

$$\text{minimize} \quad f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$$

转化成 Consensus 问题或是 Sharing 问题。

Consensus and Sharing

更传统一些的处理分布式的机器学习模型训练的方法是（例如 [3]）将我们要优化的（简化的）问题

$$\text{minimize} \quad f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$$

转化成 Consensus 问题或是 Sharing 问题。

例如以上问题引入一个新的变量 z，就可以转化为一个标准的 Consensus 问题

$$\begin{aligned} \text{minimize} \quad & \frac{1}{N}\sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} \quad & z = x_i, \quad i = 1, \cdots, N \end{aligned}$$

x_i 可以视作节点 i 上的模型参数，z 则可被视作 server 上的模型参数。

Consensus and Sharing

以上 Consensus 问题的 ADMM 算法为

- Client i local primal update

$$x_i^{k+1} = \arg\min_{x_i} \left\{ f_i(x_i) + \langle \lambda_i^k, x_i - z^k \rangle + \frac{\rho}{2} \|x_i - \bar{x}^k\|^2 \right\}$$

- Server aggregation

$$z^{k+1} = \frac{1}{N} \sum_{i=1}^{N} (x_i^{k+1} + \frac{1}{\rho} \lambda_i^k)$$

- Client i dual variable update

$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - z^{k+1})$$

Consensus and Sharing

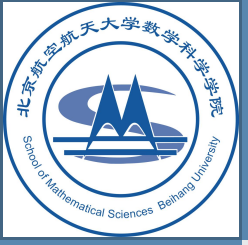
以上循环可以简化为

$$x_i^{k+1} = \arg\min_{x_i} \left\{ f_i(x_i) + \langle \lambda_i^k, x_i - \bar{x}^k \rangle + \frac{\rho}{2} \|x_i - \bar{x}^k\|^2 \right\}$$

$$z^{k+1} = \bar{x}^{k+1}$$

$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - \bar{x}^{k+1})$$

$\bar{x}$ 为对所有节点上的模型参数取均值。



Consensus and Sharing

一些传统的（线性）机器学习模型，进行垂直联邦学习，即当数据在所有节点上是垂直划分的（vertical splitting），不同的 client 上分布着同一批样本的不同特征时，我们还可以把要优化的问题表达为一个 Sharing 问题。

Consensus and Sharing

一些传统的（线性）机器学习模型，进行垂直联邦学习，即当数据在所有节点上是垂直划分的（vertical splitting），不同的 client 上分布着同一批样本的不同特征时，我们还可以把要优化的问题表达为一个 Sharing 问题。

例如线性问题

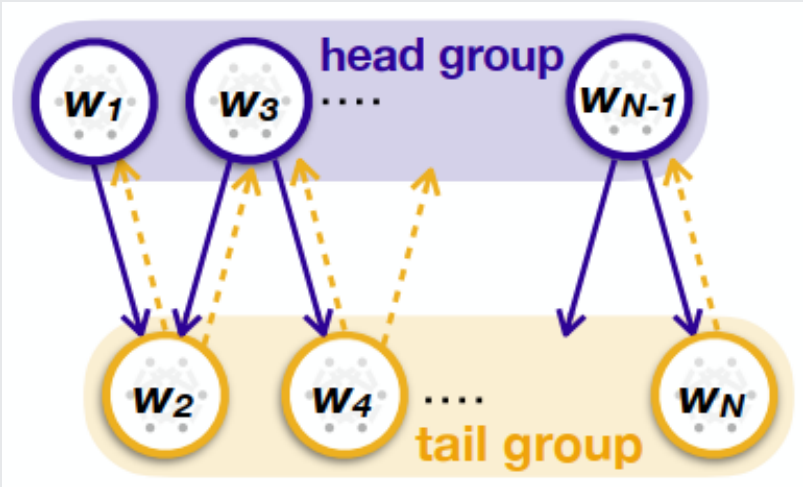
$$\text{minimize} \quad \ell(Ax-b)+r(x)$$

进行垂直划分 $A=(A_1,\cdots,A_N)$，$x=(x_1,\cdots,x_N)^T$，可以化为 Sharing 问题

$$\begin{aligned}\text{minimize} \quad & \sum_{i=1}^{N} r_i(x_i)+\ell\Big(\sum_{i=1}^{N} z_i-b\Big)\\ \text{subject to} \quad & A_i x_i=z_i\end{aligned}$$

GADMM

基于减少通信开销以及推广到去中心化学习的考虑，一些学者提出了 Group Alternating Direction Method of Multipliers (GADMM) 算法（[7], 2020）。他们考虑了一种链式的去中心化网络



图片来源：[8] Issaid et al., Communication Efficient Distributed Learning with Censored, Quantized, and Generalized Group ADMM, 2020

所有的节点被分为了 head group $\mathcal{N}_h$ 与 tail group $\mathcal{N}_t$，以此来进行交替方向优化。

GADMM

于是，要优化的问题变为了

$$\begin{aligned} \text{minimize} \quad & \frac{1}{N}\sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} \quad & x_i = x_{i+1}, \quad i = 1, \cdots, N-1 \end{aligned}$$

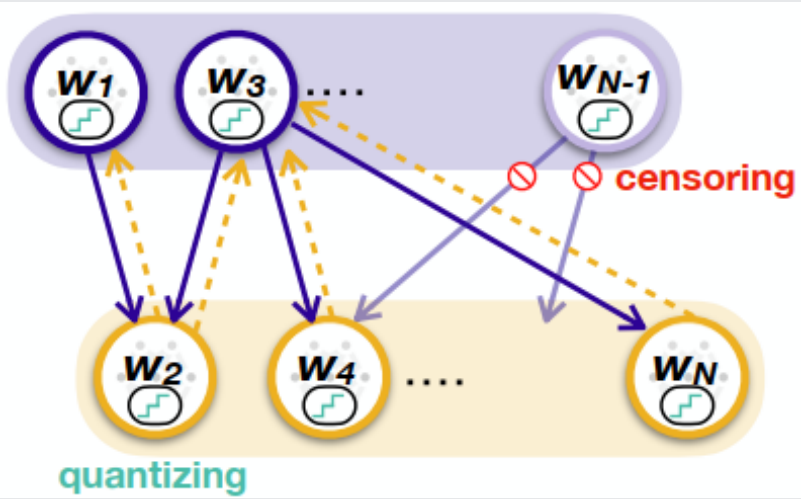
ADMM 循环中的 primal updates 为

$$x_i^{k+1} = \arg\min_{x_i}\{f_i(x_i) + \langle \lambda_{i-1}^k, x_{i-1}^k - x_i\rangle + \langle \lambda_i^k, x_i - x_{i+1}^k\rangle + \frac{\rho}{2}\|x_{i-1}^k - x_i\|^2 + \frac{\rho}{2}\|x_i - x_{i+1}^k\|^2\}, \quad i \in \mathcal{N}_h$$

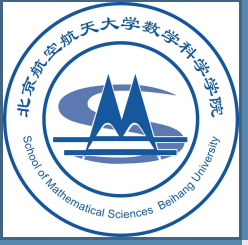
$$x_i^{k+1} = \arg\min_{x_i}\{f_i(x_i) + \langle \lambda_{i-1}^k, x_{i-1}^{k+1} - x_i\rangle + \langle \lambda_i^k, x_i - x_{i+1}^{k+1}\rangle + \frac{\rho}{2}\|x_{i-1}^{k+1} - x_i\|^2 + \frac{\rho}{2}\|x_i - x_{i+1}^{k+1}\|^2\}, \quad i \in \mathcal{N}_t$$

CQ-GGADMM

这一问题随后被推广到了更一般的网络结构上，即从链式的网络结构推广到了一般的二分网络（bipartite graph）上（需要注意的是，这个方案为了进一步降低通信开销，而在 local update 完成之后进一步对参数采取了 quantization 以及 censoring 等手段）。



图片来源：[8] Issaid et al., Communication Efficient Distributed Learning with Censored, Quantized, and Generalized Group ADMM, 2020



ADMM 与神经网络

待写。。。。

参考文献 I

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D' Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and Open Problems in Federated Learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

参考文献 II

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.

[3] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*.

Now Publishers Inc., 2011.

[4] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive Federated Optimization," in *International Conference on Learning Representations*, 2021.

参考文献 III

[5] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," in *Proceedings of Machine Learning and Systems* (I. Dhillon, D. Papailiopoulos, and V. Sze, eds.), vol. 2, pp. 429–450, 2020.

[6] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," in *International Conference on Machine Learning*, pp. 5132–5143, PMLR, 2020.

[7] A. Elgabli, J. Park, A. S. Bedi, M. Bennis, and V. Aggarwal, "GADMM: Fast and Communication Efficient Framework for Distributed Machine Learning," *Journal of Machine Learning Research*, vol. 21, no. 76, pp. 1–39, 2020.

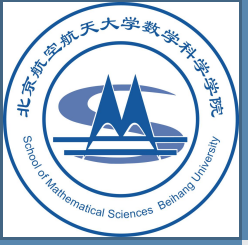
参考文献 IV

[8] C. B. Issaid, A. Elgabli, J. Park, and M. Bennis, "Communication Efficient Distributed Learning with Censored, Quantized, and Generalized Group ADMM," *arXiv preprint arXiv:2009.06459*, 2020.

[9] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated Multi-Task Learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4427–4437, 2017.

[10] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," in *International Conference on Machine Learning*, pp. 1126–1135, PMLR, 2017.

[11] F. Hanzely and P. Richtárik, "Federated Learning of a Mixture of Global and Local Models," *arXiv preprint arXiv:2002.05516*, 2020.



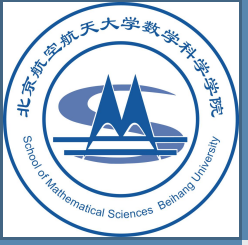
参考文献 V

[12] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized Federated Learning with Moreau Envelopes," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, (Red Hook, NY, USA), Curran Associates Inc., 2020.

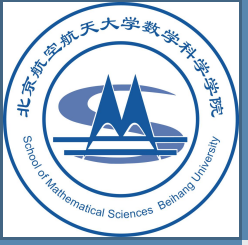
[13] S. Zhang, A. Choromanska, and Y. LeCun, "Deep Learning with Elastic Averaging SGD," in *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pp. 685–693, 2015.

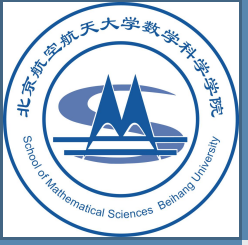
[14] C. T. Dinh, T. T. Vu, N. H. Tran, M. N. Dao, and H. Zhang, "FedU: A Unified Framework for Federated Multi-Task Learning with Laplacian Regularization," *arXiv preprint arXiv:2102.07148*, 2021.

[15] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence Analysis of Alternating Direction Method of Multipliers for a Family of Nonconvex Problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.



intentionally left blank





Personalization for FL

When does one need personalization?

— When data across clients are “enough” non-IID, which is more realistic.

Personalization for FL

When does one need personalization?

— When data across clients are "enough" non-IID, which is more realistic.

Means of personalization:

- Federated Multi-Task Learning (+ regularization / proximal term), e.g. [9]
- Model-Agnostic Meta Learning, e.g. [10]
- Local Fine-tuning.
- etc.

Personalization for FL

- Mixture of global and local [11]:

$$\text{minimize} \quad \sum_{i=1}^{N} f_i(x_i) + \frac{\lambda}{2} \sum_{i=1}^{N} \|x_i - \bar{x}\|^2$$

- pFedMe (bi-level) [12] (and similarly EASGD[13]):

$$\text{minimize} \quad \sum_{i=1}^{N} F_i(x),$$

$$\text{where} \quad F_i(x) = \min \left\{ f_i(x_i) + \frac{\lambda}{2} \|x_i - x\|^2 \right\}$$

- FedU [14]:

$$\text{minimize} \quad \sum_{i=1}^{N} f_i(x_i) + \frac{\lambda}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \|x_i - x_j\|^2$$

Mixture FL

The "weak" consensus problem (originally stated as "mixture" FL problem)

$$\text{minimize} \quad \sum_{i=1}^{N} f_i(x_i) + \frac{\lambda}{2} \sum_{i=1}^{N} \|x_i - \bar{x}\|^2$$

can be reformulated as constrained optimization problems

$$\text{minimize} \quad \sum_{i=1}^{N} f_i(x_i) + \frac{\lambda}{2} \sum_{i=1}^{N} \|x_i - z\|^2$$

$$\text{subject to} \quad Nz - \sum_{i=1}^{N} x_i = 0$$

or equivalently as the following problem,

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} f_i(x_i) + \frac{\lambda}{2} \sum_{i=1}^{N} \|x_i\|^2 - \frac{\lambda N}{2} \|z\|^2 \\ \text{subject to} \quad & Nz - \sum_{i=1}^{N} x_i = 0 \end{aligned}$$

which is a nonconvex sharing problem considered in [15] (Eq. (3.2)). Note the difference of between formulations of a sharing problem in [15] (Section 3) and in [3] (Section 7.3)

The algorithm "Flexible ADMM" proposed in [15] (Algorithm 4) updates x_i using Gauss-Seidel method, which is non-trivial (or impossible) for parallelization. On the other hand, Jacobi method seems to have no guarantee of convergence.

Mixture FL

Under certain assumptions, this problem is a (split?) DC (difference-of-convex) programming problem with linear constraints.

$$\text{minimize} \quad \sum_{i=1}^{N}\left(f_i(x_i)+\frac{\lambda}{2}\|x_i\|^2\right)-\lambda\frac{N}{2}\|z\|^2$$

$$\text{subject to} \quad Nz-\sum_{i=1}^{N}x_i=0$$

One writes $\widetilde{f}_i(x_i)=f_i(x_i)+\frac{\lambda}{2}\|x_i\|^2$, and $r(z)=\frac{N}{2}\|z\|^2$.

Mixture FL

The original unconstrained problem is studied in [11] using the so-called loopless local gradient descent (L2GD) method, with the assumptions that

- f_i are Lipschitz L-smooth

$$f(y) \leqslant f(x) + \langle \nabla f(x), (y-x) \rangle + \frac{L}{2}\|x-y\|^2$$

- f_i are μ-strongly convex

$$f(y) \geqslant f(x) + \langle \nabla f(x), (y-x) \rangle + \frac{\mu}{2}\|x-y\|^2$$

Looplessness is the one of the key contribution of [11], in which **inner (local) loops are replaced with probabilistic gradient updates**.

Mixture FL

Rewrite $\sum_{i=1}^{N} f_i(x_i) + \frac{\lambda}{2}\sum_{i=1}^{N}\|x_i - \bar{x}\|^2$ as $f(x) + \psi(x)$ with $x = (x_1, \cdots, x_N)$, the local step of L2GD at a client i is

$$x^{k+1} = x^k - \alpha G(x^k)$$

where

$$G(x^k) = \begin{cases} \dfrac{\nabla f(x^k)}{1-p} & \text{with probability } 1-p \\ \dfrac{\lambda \nabla \psi(x^k)}{p} & \text{with probability } p \end{cases}$$

Locally, one has

$$x_i^{k+1} = x_i^k - \beta \nabla f_i(x_i^k), \quad x_i^{k+1} = (1-\gamma)x_i^k + \gamma \bar{x}^k$$

with probabilities $1-p$ and p respectively.

Questions

1. Assumptions on the objective functions can be loosened?
2. DCA with linear constraints? (augmented) Lagrangian is

$$\mathcal{L}_\rho(x,z,y) = \sum_{i=1}^{N} \widetilde{f}_i(x_i) - \lambda r(z) + \langle y, Nz - \sum_{i=1}^{N} x_i \rangle + \boxed{\frac{\rho}{2}\|Nz - \sum_{i=1}^{N} x_i\|^2}$$

3. DCA (or stochastic, accelerated variants) can have better convergence?
4. more

Mixture FL — Research

Because of the existence of the boxed term $\frac{\rho}{2}\|Nz - \sum_{i=1}^{N} x_i\|^2$, the only choice to fit in the distributed settings is to update using the Jacobi method, as follows:

$$x_i^{k+1} = \arg\min_{x_i} \left\{ \widetilde{f}_i(x_i) - \langle y_i^k, x_i \rangle + \boxed{\frac{\rho}{2}\|Nz^k - \sum_{j\neq i}^{N} x_j^k - x_i\|^2} \right\}$$

$$z^{k+1} = \arg\min_{z} \left\{ \langle y^k, Nz \rangle + \frac{\rho}{2}\|Nz - \sum_{i=1}^{N} x_i^{k+1}\|^2 - \lambda r(z) \right\}$$

$$y^{k+1} = y^k + \beta(Nz^{k+1} - \sum_{i=1}^{N} x_i^{k+1})$$

Mixture FL — Research II

One can add more intermediate variables and reformulates the DC-like sharing problem as

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} \widetilde{f}_i(x_i) - \lambda r(z) \\ \text{subject to} \quad & x_i' = x_i, \quad i = 1, \cdots, N \\ & Nz = \sum_{i=1}^{N} x_i' \end{aligned}$$

Augmented Lagrangian of the above problem is

$$\sum_{i=1}^{N} \widetilde{f}_i(x_i) - \lambda r(z) + \sum_{i=1}^{N} \langle y_i, x_i' - x_i \rangle + \sum_{i=1}^{N} \frac{\rho_i}{2} \|x_i' - x_i\|^2$$

$$+ \langle y, Nz - \sum_{i=1}^{N} x_i' \rangle + \frac{\rho}{2} \left\| Nz - \sum_{i=1}^{N} x_i' \right\|^2$$

ADMM iterations of the above problem are

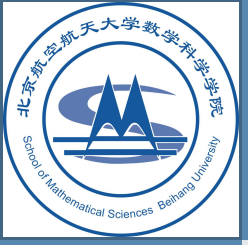
$$x_i^{k+1} = \arg\min_{x_i} \left\{ \widetilde{f}_i(x_i) + \frac{\rho_i}{2}\|x_i - (x_i')^k\|^2 - \langle y_i^k, x_i \rangle \right\}$$

$$(x_i')^{k+1} = \arg\min_{x_i'} \left\{ \langle y_i^k - y^k, x_i' \rangle + \frac{\rho_i}{2}\|x_i^{k+1} - x_i'\|^2 + \frac{\rho}{2}\|Nz^k - \sum_{j\neq i}^{N}(x_j')^k - x_i'\|^2 \right\}$$

$$z^{k+1} = \arg\min_{z} \left\{ \langle y^k, Nz \rangle + \frac{\rho}{2}\|Nz - \sum_{i=1}^{N}(x_i')^{k+1}\|^2 - \lambda r(z) \right\}$$

$$y_i^{k+1} = y_i^k + \beta((x_i')^{k+1} - x^{k+1})$$

$$y^{k+1} = y^k + \beta(Nz^{k+1} - \sum_{i=1}^{N}(x_i')^{k+1})$$



Mixture FL — Research III

to update....