Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

Problems of Personalization in Federated Learning

WEN Hao

Further updates will be done in other slides or notes.

2021-07-29

1 引言

2 联邦学习中的优化问题与算法

3 FedAvg

4 Personalization
- Model-Agnostic Meta Learning
- Federated Multi-Task Learning
- Mixture Federated Learning

5 Compression
- Naive Compression Methods
- Recent Development

1 引言

2 联邦学习中的优化问题与算法

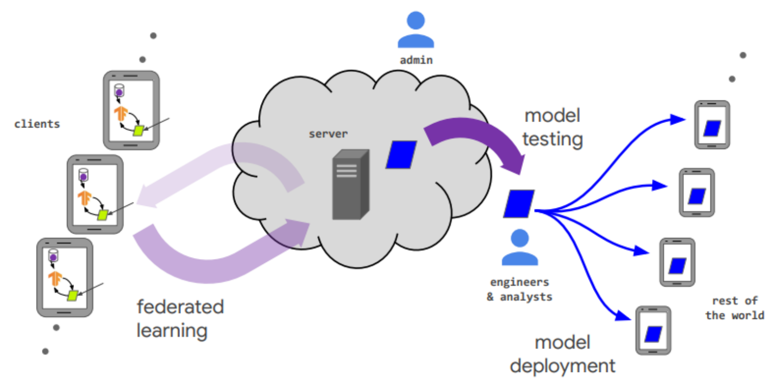
3 FedAvg

4 Personalization

5 Compression

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

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。

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

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。

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

注：以下将数据拥有方统称为“节点”

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**
  - Compression

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

- **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.

1 引言

2 联邦学习中的优化问题与算法

3 FedAvg

4 Personalization

5 Compression

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

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

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

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

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

Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

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

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

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

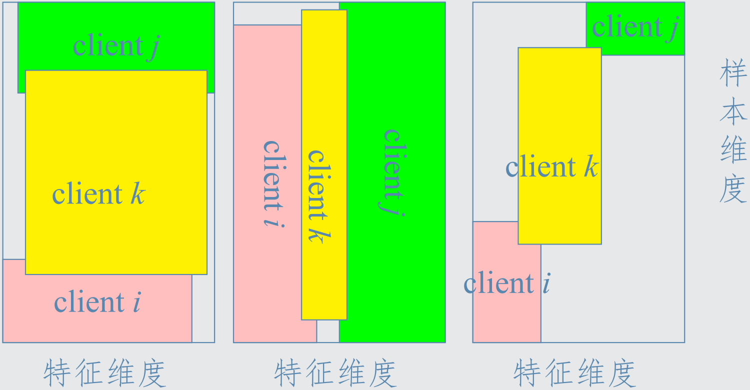
在真实场景下，各个节点上的数据的分布 $\mathcal{D}_i$ 一般不是独立同分布的（non-IID, 或称 heterogeneous）。这种数据分布的各向异性将联邦学习分为了 3 类

- 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):

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

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)**

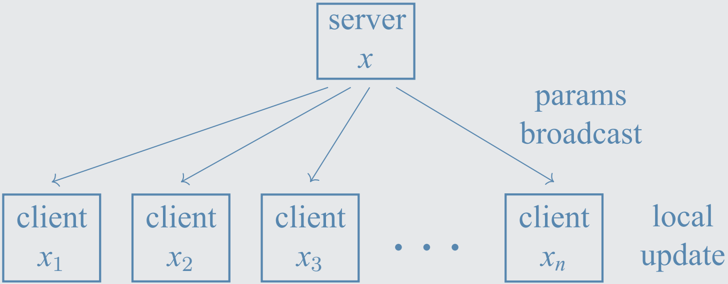
有人 [2] 把以上称为所谓的 “computation then aggregation” (CTA) protocol

[2] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, “FedPD: A Federated Learning Framework With Adaptivity to Non-IID Data,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 6055–6070, 2021

联邦学习的一般性框架（流程）

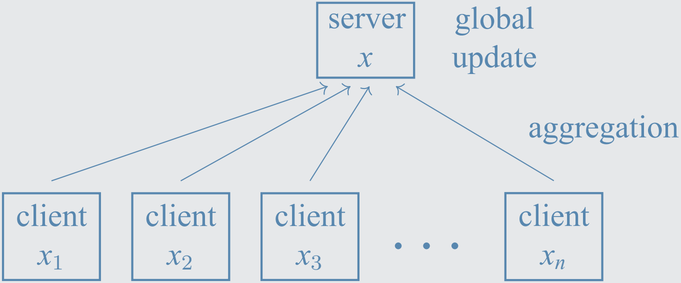
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

Broadcast and local update:



联邦学习的一般性框架（流程）

Aggregate and global update:



Another protocol

Transmit parameters $\Rightarrow$ Transmit gradients

[3]L. Zhu, Z. Liu, and S. Han, "Deep Leakage from Gradients," *Advances in Neural Information Processing Systems*, vol. 32, pp. 14774–14784, 2019

Another protocol

Transmit parameters $\Rightarrow$ Transmit gradients

风险：data leakage ([3])。

MWE: Consider the simplest model $y = ax + b$, updated using data point $(\widehat{x}, \widehat{y})$, with MSE loss

$$f = \text{loss} = (y - \widehat{y})^2 = (a\widehat{x} + b - \widehat{y})^2,$$

One has

$$\frac{\partial f}{\partial a} = 2\widehat{x}(a\widehat{x} + b - \widehat{y})$$
$$\frac{\partial f}{\partial b} = 2(a\widehat{x} + b - \widehat{y})$$

Knowing the gradients, $\widehat{x}, \widehat{y}$ can be easily recovered.

[3]L. Zhu, Z. Liu, and S. Han, "Deep Leakage from Gradients," *Advances in Neural Information Processing Systems*, vol. 32, pp. 14774–14784, 2019

1 引言

2 联邦学习中的优化问题与算法

3 FedAvg

4 Personalization

5 Compression

从 FedAvg 到 FedOpt

Google 研究人员 McMahan 等人在文章 [4] 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 SGD，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。随后，McMahan 等人进一步在文章 [5] 中，将 Adam 等自适应、加速算法融入联邦学习中，提出了更一般的 FedOpt 框架。

[4]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

[5]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

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**

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

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

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**

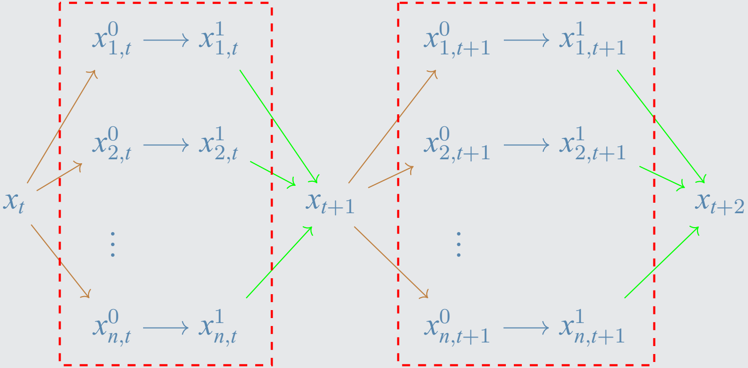
for *batch* $b \in \mathcal{B}$ **do**

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

return x

FedSGD – baseline

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



ONE local (S)GD　　　ONE local (S)GD

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。

$$
\begin{array}{ccccccccccc}
 & x_{1,t}^0 & \longrightarrow & x_{1,t}^1 & \longrightarrow & x_{1,t}^2 & \longrightarrow & \cdots & \longrightarrow & x_{1,t}^K & \\
 & x_{2,t}^0 & \longrightarrow & x_{2,t}^1 & \longrightarrow & x_{2,t}^2 & \longrightarrow & \cdots & \longrightarrow & x_{2,t}^K & \\
x_t & & & & & & & & & & x_{t+1} \\
 & \vdots & & & & & & & & & \\
 & x_{n,t}^0 & \longrightarrow & x_{n,t}^1 & \longrightarrow & x_{n,t}^2 & \longrightarrow & \cdots & \longrightarrow & x_{n,t}^K &
\end{array}
$$

K local mini-batch SGD

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

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 $\nabla f_i(x_{i,t}^k)$

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

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

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

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

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

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, $\boxed{\text{Adagrad, Yogi, Adam}}$, etc.

FedOpt

Algorithm 3: FedAdagrad, FedAdam

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

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

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)**

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

Convergence

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

Algorithm	CVX	BGD	NEC	ALS	RC (T)	LS (Q)	SC
FedAvg [Li et al 19]	μSC	(G,0)	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1)$	$\boldsymbol{\mathcal{O}(1/\epsilon)}$
FedAvg [Karimireddy et al 20]	μSC	(G,D)	✓	×	$\mathcal{O}(1/\sqrt{\epsilon})$	$\mathcal{O}(1/\sqrt{\epsilon})$	$\boldsymbol{\mathcal{O}(1/\epsilon)}$
FedSplit [Pathak-Wainwright 20]	μSC	-	✓	✓	$\boldsymbol{\mathcal{O}(\log(1/\epsilon))}$	$\mathcal{O}(1/\epsilon)^*$	$\mathcal{O}(\log(1/\epsilon)/\epsilon)$
Local-GD [Khaled et al 20]	C	-	✓	×	$\mathcal{O}(1/\epsilon^{1.5})$	$\mathcal{O}(1)$	$\mathcal{O}(M/\epsilon^{1.5})$
FedAvg [Karimireddy et al 20]	NC	(G,D)	✓	×	$\mathcal{O}(1/\epsilon^{1.5})$	$\mathcal{O}(1/\sqrt{\epsilon})$	$\mathcal{O}(1/\epsilon^2)$
VRL-SGD [Liang et al 20]	NC	-	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$
F-SVRG [Cen et al 19]	NC	-	×	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1)$	$\mathcal{O}(M/\epsilon)$
FedPD (Ours)	NC	-	✓	✓	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon)^*$	$\mathcal{O}(1/\epsilon^2)$
FedPD (Ours)	NC	(G,1)	✓	✓	$\boldsymbol{\mathcal{O}((1-p)/\epsilon)}$	$\mathcal{O}(1/\epsilon)^*$	$\mathcal{O}((1-p)/\epsilon^2)$
FedPD(VR) (Ours)	NC	-	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1)$	$\mathcal{O}(M+\sqrt{M}/\epsilon)$

CVX (Convexity), μSC (μ-strongly convex), C (convex), NC (non-convex); BGD (bounded gradient dissimilarity); NEC (no extra communication); ALS (arbitrary local solver); RC (round of communication); LS (local steps); SC (sample complexity); p is a function of ϵ/G; * assume local solvers are SGD.

上图来自洪明毅老师在第六十一期运筹千里纵横论坛所作的报告。

[6]P. Khanduri, P. Sharma, H. Yang, M. Hong, J. Liu, K. Rajawat, and P. K. Varshney, “STEM: A Stochastic Two-Sided Momentum Algorithm Achieving Near-Optimal Sample and Communication Complexities for Federated Learning,” 2021

Convergence

Algorithm	CVX	BGD	NEC	ALS	RC (T)	LS (Q)	SC
FedAvg [Li et al 19]	μSC	(G,0)	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1)$	$\boldsymbol{\mathcal{O}(1/\epsilon)}$
FedAvg [Karimireddy et al 20]	μSC	(G,D)	✓	×	$\mathcal{O}(1/\sqrt{\epsilon})$	$\mathcal{O}(1/\sqrt{\epsilon})$	$\boldsymbol{\mathcal{O}(1/\epsilon)}$
FedSplit [Pathak-Wainwright 20]	μSC	-	✓	✓	$\boldsymbol{\mathcal{O}(\log(1/\epsilon))}$	$\mathcal{O}(1/\epsilon)^*$	$\mathcal{O}(\log(1/\epsilon)/\epsilon)$
Local-GD [Khaled et al 20]	C	-	✓	×	$\mathcal{O}(1/\epsilon^{1.5})$	$\mathcal{O}(1)$	$\mathcal{O}(M/\epsilon^{1.5})$
FedAvg [Karimireddy et al 20]	NC	(G,D)	✓	×	$\mathcal{O}(1/\epsilon^{1.5})$	$\mathcal{O}(1/\sqrt{\epsilon})$	$\mathcal{O}(1/\epsilon^2)$
VRL-SGD [Liang et al 20]	NC	-	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$
F-SVRG [Cen et al 19]	NC	-	×	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1)$	$\mathcal{O}(M/\epsilon)$
FedPD (Ours)	NC	-	✓	✓	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon)^*$	$\mathcal{O}(1/\epsilon^2)$
FedPD (Ours)	NC	(G,1)	✓	✓	$\boldsymbol{\mathcal{O}((1-p)/\epsilon)}$	$\mathcal{O}(1/\epsilon)^*$	$\mathcal{O}((1-p)/\epsilon^2)$
FedPD(VR) (Ours)	NC	-	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1)$	$\mathcal{O}(M+\sqrt{M}/\epsilon)$

CVX (Convexity), μSC (μ-strongly convex), C (convex), NC (non-convex); BGD (bounded gradient dissimilarity); NEC (no extra communication); ALS (arbitrary local solver); RC (round of communication); LS (local steps); SC (sample complexity); p is a function of ϵ/G; * assume local solvers are SGD.

上图来自洪明毅老师在第六十一期运筹千里纵横论坛所作的报告。

洪明毅老师最近的工作 [6]，在进行 local update 的时候也用了 momentum 加速，进一步扩展了 FedOpt。

[6]P. Khanduri, P. Sharma, H. Yang, M. Hong, J. Liu, K. Rajawat, and P. K. Varshney, “STEM: A Stochastic Two-Sided Momentum Algorithm Achieving Near-Optimal Sample and Communication Complexities for Federated Learning,” 2021

1 引言

2 联邦学习中的优化问题与算法

3 FedAvg

4 Personalization

5 Compression

Personalization for FL

Introduction

Optim in FL

FedAvg

Personalization

MAML

FMTL

Mixture FL

Compression

Naive Compression Methods

Recent Development

What is model personalization for FL

— different models (parameters) for different clients

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

What is model personalization for FL

— different models (parameters) for different clients

When does one need personalization?

— When data across clients are “enough” non-IID and clients do not generally have enough training data, which is more realistic.

Personalization for FL

What is model personalization for FL

— different models (parameters) for different clients

When does one need personalization?

— When data across clients are "enough" non-IID and clients do not generally have enough training data, which is more realistic.

Means of personalization:

- Local Fine-tuning.
- Model-Agnostic Meta Learning, e.g. [7]
- Federated Multi-Task Learning (+ regularization / proximal term), e.g. [8]
- etc.

Model-Agnostic Meta Learning

"The goal of meta-learning is to train a model on a **variety of learning tasks***, such that it can solve new learning tasks using only a small number of training samples." – [7]*

i.e. over a distribution of learning tasks $p(\mathcal{T})$, where

$$\mathcal{T} = \{\mathcal{L}(\{(x_t, a_t)\}), q(x_1), q(x_{t+1}|x_t, a_t), H\}$$

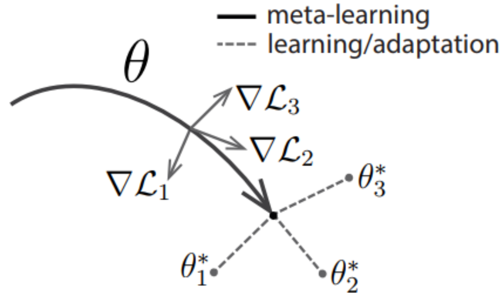
with

$$\begin{aligned}
(x_t, a_t) &: && \text{data points} \\
\mathcal{L} &: && \text{loss function} \\
q(x_1) &: && \text{initial distribution} \\
q(x_{t+1}|x_t, a_t) &: && \text{transition} \\
H &: && \text{episode length}
\end{aligned}$$

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

Intuition of MAML

Some internal representations are more transferrable than others. MAML should encourage the emergence of such general-purpose representations via searching for model parameters that are sensitive to changes in the task.

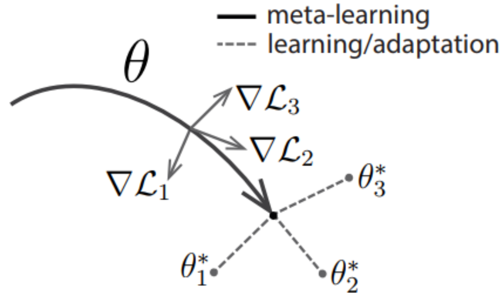


Model-Agnostic Meta Learning – Formulation

Mathematically, MAML can be formulated as a (bi-level?) optimization problem

$$\text{minimize} \quad \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$$

$$\text{where} \quad \theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$$



Model-Agnostic Meta Learning – Algorithm

Algorithm 4: MAML[7]

Require: $p(\mathcal{T})$ distribution over tasks

Require: α, β step size hyper-params

randomly initialize model params θ

while *not done* **do**

  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$

  for *all* $\mathcal{T}_i$ **do**

    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ w.r.t. K samples

    Compute adapted parameters with gradient descent $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$

  Update $\theta \leftarrow \theta - \beta \nabla_\theta \boxed{\sum_{\mathcal{T}_i \sim p(\mathcal{T})}} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$

Model-Agnostic Meta Learning – Algorithm

Algorithm 5: MAML[7]

Require: $p(\mathcal{T})$ distribution over tasks
Require: α, β step size hyper-params
randomly initialize model params θ
while *not done* **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for *all* $\mathcal{T}_i$ **do**
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ w.r.t. K samples
    Compute adapted parameters with gradient descent $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \boxed{\sum_{\mathcal{T}_i \sim p(\mathcal{T})}} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$

"Extragradient method"!

Model-Agnostic Meta Learning – Applications

In deep learning, a very commonly used architecture is as follows:

$$\text{input} \rightarrow \boxed{\text{CNN (encoder)}} (\rightarrow \text{attn}) \rightarrow \text{task specific module}$$

tasks can be one or more of

- classification (global pooling + linear)
- sequence labelling (linear)
- segmentation (upsample)
- object detection
- etc.

or many sub-tasks of the above (current main concern for meta-learning).

MAML forces the feature extractor (or called encoder, etc.) to capture general-purpose internal representations (features).

Federated Multi-Task Learning

- pFedMe (bi-level) [9] (and similarly EASGD[10]):

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

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

- FedU [11]:

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

[9]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

[10]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

[11]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

pFedMe – Formulation

pFedMe (Personalized Federated Learning with Moreau Envelopes (or proximity operator)) is formulated as the following bi-level optimization problem in [9]

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

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

which is equivalent to

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

[9]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

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

Algorithm 6: pFedMe[9]

Input: $T, R, S, \lambda, \eta, \beta, x^0$

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

  Server sends x^t to all clients

  for $i = 1, \cdots, N$ **do**

    $x_{i,0}^t = x^t$

    for $r = 0, \cdots, R-1$ **do**

      Sample a fresh mini-batch $\mathcal{D}_i$, and find an approximate $x_i(x_{i,r}^t)$ to the problem $\min\{\ell_i(x_i; \mathcal{D}_i) + \frac{\lambda}{2}\|x_i - x_{i,r}^t\|^2\}$

      Local update $x_{i,r+1}^t = x_{i,r}^t - \eta\lambda(x_{i,r}^t - x_i(x_{i,r}^t))$

    Server uniformly samples a subset of clients $\mathcal{S}^t$, each of which sends the local $x_{i,R}^t$ to the server

  Sever update $x^{t+1} = (1-\beta)x^t + \beta \sum \frac{x_{i,R}^t}{\#\mathcal{S}^t}$

Personalization

WEN Hao

Introduction

Optim in FL

FedAvg

Personalization

MAML

FMTL

Mixture FL

Compression

Naive Compression Methods

Recent Development

pFedMe observations

- global model x converges (if converges) to the average of local models, which can be inferred from

$$x^* = \min_x \left\{ \sum_{i=1}^N \left(f_i(x_i) + \frac{\lambda}{2}\|x_i - x\|^2 \right) \right\} = \frac{1}{N}\sum_{i=1}^N x_i$$

- local updates are not "totally local", i.e. the loop $r = 0, \cdots, R$ computes the "global objective" $\min\{F_i(x)\}$ locally, to reduce communication.
- pFedMe = Method 1 proposed by Caihua Chen, i.e. outer line search of x, inner solving Prox
- What is Method 2 (randomization?) proposed by Caihua Chen?

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,

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

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

which is a nonconvex sharing problem considered in [12] (Eq. (3.2)). Note the difference of between formulations of a sharing problem in [12] (Section 3) and in [13] (Section 7.3)

[12]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

[13]S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Publishers Inc., 2011

Mixture FL

The algorithm “Flexible ADMM” proposed in [12] (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.

[12]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

Mixture FL

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

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

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 [14] 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 [14], in which **inner (local) loops are replaced with probabilistic gradient updates**.

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

Mixture FL

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

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.

Mixture FL — ADMM

Questions

1. Assumptions on the objective functions can be loosened or not?
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 — ADMM

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 — ADMM

Let $u_i = Nz - \sum_{j \neq i}^{N} x_j$, the updates can be done as follows:

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

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

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

$$u_i^{k+1} = Nz^{k+1} - \sum_{j \neq i}^{N} x_j^{k+1}$$

It should be noted that u_i^{k+1} are computed in the server and broadcast to the clients.

TODO: More analysis on this update pattern

Huawei Noah FL Workshop

7月30日下午 算法Session				
时间	时长	议题	主讲人	单位
14:30-14:35	5	欢迎致辞	Reporter	华为-诺亚
14:35-15:05	30	基于最大化相关性的个性化联邦学习	Reporter	华为-诺亚
15:05-15:35	30	联邦学习在语音唤醒中的应用	Reporter	华为-诺亚
15:35-16:05	30	诺亚纵向联邦学习框架	Reporter	华为-诺亚
16:05-16:35	30	多目标优化联邦学习	Reporter/胡泽欧	华为-诺亚（加拿大）/ 滑铁卢大学

pFedMac[15] modified the objective of pFedMe to

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

or equivalently

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

TODO: analyze pFedMac and propose possible improvements

[15]Y. Li, X. Liu, X. Zhang, Y. Shao, Q. Wang, and Y. Geng, "Personalized Federated Learning via Maximizing Correlation with Sparse and Hierarchical Extensions," *arXiv preprint arXiv:2107.05330*, 2021

More on FL Personalization - Huawei Noah FL Workshop

- FedPHP (to add ref. latter)
- FedMGDA+[16] and Pareto optimality

[16]Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu, “FedMGDA+: Federated Learning Meets Multi-Objective Optimization,” *arXiv preprint arXiv:2006.11489*, 2020

1 引言

2 联邦学习中的优化问题与算法

3 FedAvg

4 Personalization

5 Compression

Compression in Federated Learning

For federated learning, especially in the cross-device scenario, one of the main bottleneck communication cost can be reduced using

- compression
- lazy aggregation (censoring)
- etc.

Compression in Federated Learning

For federated learning, especially in the cross-device scenario, one of the main bottleneck communication cost can be reduced using

- compression
- lazy aggregation (censoring)
- etc.

The technique of compression mainly consists of

- (randomized) quantization
- sparsification

or their combination.

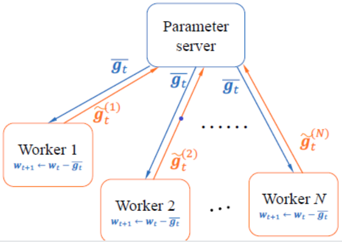
Deterministic Compression

compression can be naively done via fixed reduction of precision (fixed bit of quantization) of parameters and/or gradients, e.g. half precision (`float32` $\rightarrow$ `float16`) or mixed precision.

This is the common practice for acceleration of ordinary (non-distributed) model training process. e.g. the PyTorch Post on mixed precision training.

TernGrad

One extreme case of compression is to take the sign of each coordinate of the stochastic gradient vector, which makes it binary (1-bit, ± 1) or ternary ($\{-1, 0, +1\}$).



$\tilde{g}_t^{(i)}$ is the ternarized gradient

$$g_t^{(i)} = \|g_t^{(i)}\|_\infty \cdot \mathrm{sign}(g_t^{(i)}) \odot \boxed{b_t}$$

where b_t is a random binary vector satisfying some Bernoulli distribution $Be(|g_{t,k}^{(i)}|/s_t)$

Similar algorithms include 1-bit SGD [17], signSGD [18]

[17] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-Bit Stochastic Gradient Descent and Application to Data-Parallel Distributed Training of Speech DNNs," in *Interspeech 2014*, 9 2014

[18] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed Optimisation for Non-Convex Problems," in *International Conference on Machine Learning*, pp. 560–569, PMLR, 2018

QSGD

More generally, in QSGD [19], randomized quantization (called "low-precision quantizer" in [20]) is performed on gradients v via

$$Q_s(v) = \|v\|_2 \cdot \operatorname{sign}(v) \odot \boxed{\xi(v, s)},$$

where the i-th element in vector $\xi(v, s)$ is defined by

$$\xi_i(v, s) = \begin{cases} (\ell + 1)/s, & \text{with prob. } (|v_i|/\|v\|_2)s - \ell \\ \ell/s, & \text{otherwise} \end{cases}$$

s controls the number of quantization levels, and ℓ (should be ℓ_i) be s.t. $|v_i|/\|v\|_2 \in [\ell/s, (\ell+1)/s]$.

[19] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1709–1720, 2017

[20] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Distributed Learning with Compressed Gradients," *arXiv preprint arXiv:1806.06573*, 2018

DCGD

DCGD [20] generalized such operators Q_s into an abstract concept

Definition (Unbiased Random Quantizer (URQ))

A mapping $Q : \mathbb{R}^d \to \mathbb{R}^d$ is called an unbiased random quantizer if $\forall v \in \mathbb{R}^d$,

- $\mathrm{supp}(Q(v)) \subseteq \mathrm{supp}(v)$
- $\mathbb{E}[Q(v)] = v$
- $\mathbb{E}[\|Q(v)\|_2^2] \leqslant \alpha \|v\|_2^2$ *for some finite positive* α

And perhaps with more useful properties like

- sparsity: $\mathbb{E}[\|Q(v)\|_0] \leqslant \mathrm{const}$
- sign preserving: $Q(v)_i \cdot v_i \geqslant 0$

[20] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Distributed Learning with Compressed Gradients," *arXiv preprint arXiv:1806.06573*, 2018

Examples of URQs

Despite the ternary quantizer and low-precision quantizer, one has [21]

Random-k sparsification

$$\mathcal{C}(v) = \frac{d}{k}(v \odot \xi_k)$$

where $\xi_k \in \{0,1\}^d$ is a uniformly random binary vector with k nonzero entries, $v \in \mathbb{R}^d$.

[21] Z. Li, D. Kovalev, X. Qian, and P. Richtarik, "Acceleration for Compressed Gradient Descent in Distributed and Federated Optimization," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 5895–5904, PMLR, 7 2020

Examples of URQs

Despite the ternary quantizer and low-precision quantizer, one has [21]

(p, s)-quantization

$$\mathcal{C}_{p,s}(v) = \operatorname{sign}(v) \cdot \|v\|_p \cdot \frac{1}{s}\xi(v, s)$$

where $\xi(v, s)$ is a random vector with i-th element

$$\xi_i(v, s) = \begin{cases} \ell_i + 1, & \text{with prob. } (|v_i|/\|v\|_2)s - \ell_i \\ \ell_i, & \text{otherwise} \end{cases}$$

and ℓ_i be s.t. $|v_i|/\|v\|_2 \in [\ell_i/s, (\ell_i + 1)/s]$

[21] Z. Li, D. Kovalev, X. Qian, and P. Richtarik, "Acceleration for Compressed Gradient Descent in Distributed and Federated Optimization," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 5895–5904, PMLR, 7 2020

Implementations of Quantizers

One can refer to https://github.com/burlachenkok/marina for code and examples of various compressors, e.g. in files

- linear_model_with_non_convex_loss/compressors.py
- neural_nets_experiments/compressors.py

or this simple jupyter notebook

(A)DIANA

The main contribution of (A)DIANA [21, 22] is that, instead of quantizing the gradients, the difference of gradient updates, i.e. instead of

$$\widetilde{g}_t^{(i)} = Q(g_t^{(i)}) = Q(\nabla f_i(x_t))$$

one performs

$$\begin{cases} \widetilde{g}_t^{(i)} = h_t^{(i)} + Q(\nabla f_i(x_t) - h_t^{(i)}) \\ h_{t+1}^{(i)} = h_t^{(i)} + \alpha Q(\nabla f_i(x_t) - h_t^{(i)}) \end{cases}$$

$h^{(i)}$ are "memory" maintained locally, whose average is maintained in the central server.

[21] Z. Li, D. Kovalev, X. Qian, and P. Richtarik, "Acceleration for Compressed Gradient Descent in Distributed and Federated Optimization," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 5895–5904, PMLR, 7 2020

[22] K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik, "Distributed Learning with Compressed Gradient Differences," *arXiv preprint arXiv:1901.09269*, 2019

(A)DIANA

Another key point (feature) of (A)DIANA is the combination with acceleration (and variance reduction):

Algorithm 1 `DIANA` (n nodes)

input learning rates $\alpha > 0$ and $\{\gamma^k\}_{k\geq 0}$, initial vectors $x^0, h_1^0, \ldots, h_n^0 \in \mathbb{R}^d$ and $h^0 = \frac{1}{n}\sum_{i=1}^n h_i^0$, quantization parameter $p \geq 1$, sizes of blocks $\{d_l\}_{l=1}^m$, momentum parameter $0 \leq \beta < 1$

$v^0 = \nabla f(x^0)$
for $k = 0, 1, \ldots$ **do**
  Broadcast x^k to all workers
  for $i = 1, \ldots, n$ in parallel **do**
    Sample g_i^k such that $\mathbf{E}[g_i^k \mid x^k] = \nabla f_i(x^k)$ and let $\Delta_i^k = g_i^k - h_i^k$
    Sample $\hat{\Delta}_i^k \sim \mathrm{Quant}_p(\Delta_i^k, \{d_l\}_{l=1}^m)$ and let $h_i^{k+1} = h_i^k + \alpha\hat{\Delta}_i^k$ and $\hat{g}_i^k = h_i^k + \hat{\Delta}_i^k$
  end for
  $\hat{\Delta}^k = \frac{1}{n}\sum_{i=1}^n \hat{\Delta}_i^k$; $\hat{g}^k = \frac{1}{n}\sum_{i=1}^n \hat{g}_i^k = h^k + \hat{\Delta}^k$; $v^k = \beta v^{k-1} + \hat{g}^k$
  $x^{k+1} = \mathrm{prox}_{\gamma^k R}\left(x^k - \gamma^k v^k\right)$; $h^{k+1} = \frac{1}{n}\sum_{i=1}^n h_i^{k+1} = h^k + \alpha\hat{\Delta}^k$
end for

Note the “Quant” operator is a so-called “block-quantizer” or “bucket-quantizer”[19] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, “QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 1709–1720, 2017

(A)DIANA

Another key point (feature) of (A)DIANA is the combination with acceleration (and variance reduction):

Algorithm 2 Accelerated DIANA (ADIANA)

Input: initial point x^0, $\{h_i^0\}_{i=1}^n$, $h^0 = \frac{1}{n}\sum_{i=1}^n h_i^0$, parameters $\eta, \theta_1, \theta_2, \alpha, \beta, \gamma, p$

$z^0 = y^0 = w^0 = x^0$

for $k = 0, 1, 2, \ldots$ **do**

    $x^k = \theta_1 z^k + \theta_2 w^k + (1 - \theta_1 - \theta_2) y^k$

    for all machines $i = 1, 2, \ldots, n$ **do in parallel**

        Compress shifted local gradient $\mathcal{C}_i^k(\nabla f_i(x^k) - h_i^k)$ and send to the server

        Update local shift $h_i^{k+1} = h_i^k + \alpha \mathcal{C}_i^k(\nabla f_i(w^k) - h_i^k)$

    end for

    Aggregate received compressed gradient information

$$g^k = \frac{1}{n}\sum_{i=1}^n \mathcal{C}_i^k(\nabla f_i(x^k) - h_i^k) + h^k$$

$$h^{k+1} = h^k + \alpha \frac{1}{n}\sum_{i=1}^n \mathcal{C}_i^k(\nabla f_i(w^k) - h_i^k)$$

    Perform update step

$$y^{k+1} = \mathrm{prox}_{\eta\psi}(x^k - \eta g^k)$$

    $z^{k+1} = \beta z^k + (1-\beta) x^k + \frac{\gamma}{\eta}(y^{k+1} - x^k)$

$$w^{k+1} = \begin{cases} y^k, & \text{with probability } p \\ w^k, & \text{with probability } 1-p \end{cases}$$

end for

Personalization
WEN Hao
Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

MARINA [23] replaced the unbiased compressor by a biased one, via replacing

$$\begin{cases} \tilde{g}_t^{(i)} = h_t^{(i)} + Q(\nabla f_i(x_t) - h_t^{(i)}) \\ h_{t+1}^{(i)} = h_t^{(i)} + \alpha Q(\nabla f_i(x_t) - h_t^{(i)}) \end{cases}$$

by

$$\tilde{g}_t^{(i)} = \begin{cases} \nabla f_i(x_t), & \text{with prob. } p \\ \tilde{g}_{t-1}^{(i)} + Q(\nabla f_i(x_t) - \nabla f_i(x_{t-1})), & \text{with prob. } 1-p \end{cases}$$

for some small p.

MARINA

MARINA [23] replaced the unbiased compressor by a biased one, via replacing

$$\begin{cases} \widetilde{g}_t^{(i)} = h_t^{(i)} + Q(\nabla f_i(x_t) - h_t^{(i)}) \\ h_{t+1}^{(i)} = h_t^{(i)} + \alpha Q(\nabla f_i(x_t) - h_t^{(i)}) \end{cases}$$

by

$$\widetilde{g}_t^{(i)} = \begin{cases} \nabla f_i(x_t), & \text{with prob. } p \\ \widetilde{g}_{t-1}^{(i)} + Q(\nabla f_i(x_t) - \nabla f_i(x_{t-1})), & \text{with prob. } 1-p \end{cases}$$

for some small p.

As claimed by the authors, their intuition come from the rare (?) phenomenon in stochastic optimization that

> *"the bias of the stochastic gradient helps to achieve better complexity"*

Personalization

WEN Hao

Introduction

Optim in FL

FedAvg

Personalization

MAML

FMTL

Mixture FL

Compression

Naive Compression Methods

Recent Development

The basic MARINA algorithm is as follows:

Algorithm 1 MARINA

1: **Input:** starting point x^0, stepsize γ, probability $p \in (0, 1]$, number of iterations K
2: Initialize $g^0 = \nabla f(x^0)$
3: **for** $k = 0, 1, \ldots, K-1$ **do**
4: Sample $c_k \sim \text{Be}(p)$
5: Broadcast g^k to all workers
6: **for** $i = 1, \ldots, n$ in parallel **do**
7: $x^{k+1} = x^k - \gamma g^k$
8: Set $g_i^{k+1} = \nabla f_i(x^{k+1})$ if $c_k = 1$, and $g_i^{k+1} = g^k + \mathcal{Q}\left(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right)$ otherwise
9: **end for**
10: $g^{k+1} = \frac{1}{n}\sum_{i=1}^{n} g_i^{k+1}$
11: **end for**
12: **Return:** $\hat{x}^K$ chosen uniformly at random from $\{x^k\}_{k=0}^{K-1}$

References 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.

References II

[2] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, “FedPD: A Federated Learning Framework With Adaptivity to Non-IID Data,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 6055–6070, 2021.

[3] L. Zhu, Z. Liu, and S. Han, “Deep Leakage from Gradients,” *Advances in Neural Information Processing Systems*, vol. 32, pp. 14774–14784, 2019.

[4] 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.

[5] 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.

References III

[6] P. Khanduri, P. Sharma, H. Yang, M. Hong, J. Liu, K. Rajawat, and P. K. Varshney, “STEM: A Stochastic Two-Sided Momentum Algorithm Achieving Near-Optimal Sample and Communication Complexities for Federated Learning,” 2021.

[7] 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.

[8] 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.

[9] 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.

References IV

[10] 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.

[11] 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.

[12] 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.

References V

[13] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*.
Now Publishers Inc., 2011.

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

[15] Y. Li, X. Liu, X. Zhang, Y. Shao, Q. Wang, and Y. Geng, “Personalized Federated Learning via Maximizing Correlation with Sparse and Hierarchical Extensions,” *arXiv preprint arXiv:2107.05330*, 2021.

[16] Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu, “FedMGDA+: Federated Learning Meets Multi-Objective Optimization,” *arXiv preprint arXiv:2006.11489*, 2020.

References VI

[17] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-Bit Stochastic Gradient Descent and Application to Data-Parallel Distributed Training of Speech DNNs," in *Interspeech 2014*, 9 2014.

[18] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed Optimisation for Non-Convex Problems," in *International Conference on Machine Learning*, pp. 560–569, PMLR, 2018.

[19] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1709–1720, 2017.

[20] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Distributed Learning with Compressed Gradients," *arXiv preprint arXiv:1806.06573*, 2018.

References VII

[21] Z. Li, D. Kovalev, X. Qian, and P. Richtarik, “Acceleration for Compressed Gradient Descent in Distributed and Federated Optimization,” in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 5895–5904, PMLR, 7 2020.

[22] K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik, “Distributed Learning with Compressed Gradient Differences,” *arXiv preprint arXiv:1901.09269*, 2019.

[23] E. Gorbunov, K. Burlachenko, Z. Li, and P. Richtárik, “MARINA: Faster Non-Convex Distributed Learning with Compression,” *arXiv preprint arXiv:2102.07845*, 2021.

Introduction
Optim in FL
FedAvg
Personalization
MAML
FMTL
Mixture FL
Compression
Naive Compression Methods
Recent Development

The End

谢谢！

以上内容可以在 https://github.com/wenh06/fl_seminar 找到。