Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心


# 联邦学习的参数更新方法

硕士研究生　崔成钢

2021年09月25日

# 内容提要

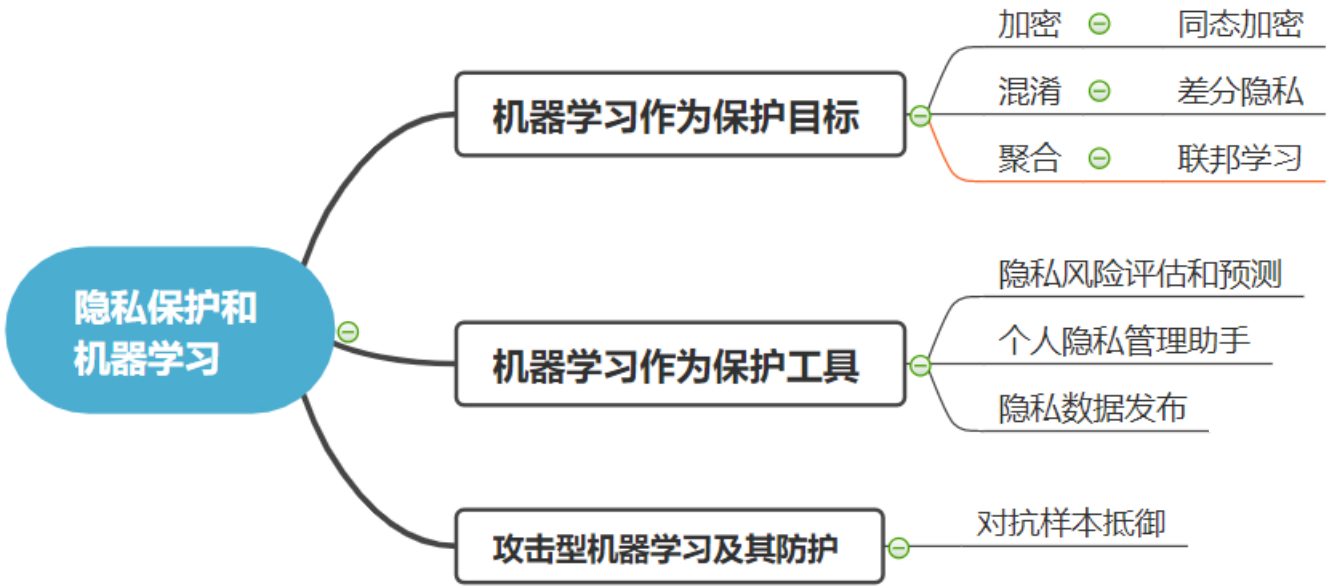
- 背景简介
- 基本概念
  - 联邦学习
  - 非独立同分布
- 算法原理
  - FedAvg
  - FedProx
- 应用总结
- 参考文献

# 预期收获

- 预期收获
  - 1. 了解隐私保护和机器学习的基本关系
  - 2. 了解联邦学习及其分类
  - 3. 了解联邦学习中的参数传递方法
  - 4. 了解应用领域和发展方向等

# 背景简介

- 隐私保护和机器学习



# 背景简介

- 提出背景
  - 在传统的机器学习中，模型的效率和准确性依赖于集中式服务器的计算能力和训练数据，即用户数据存储在中央服务器上，用于培训和测试过程，以便最终开发全面的ML模型
  - 基于集中式ML方法的挑战
    - 计算能力
    - 计算时间
    - 数据的安全性和隐私性被忽视

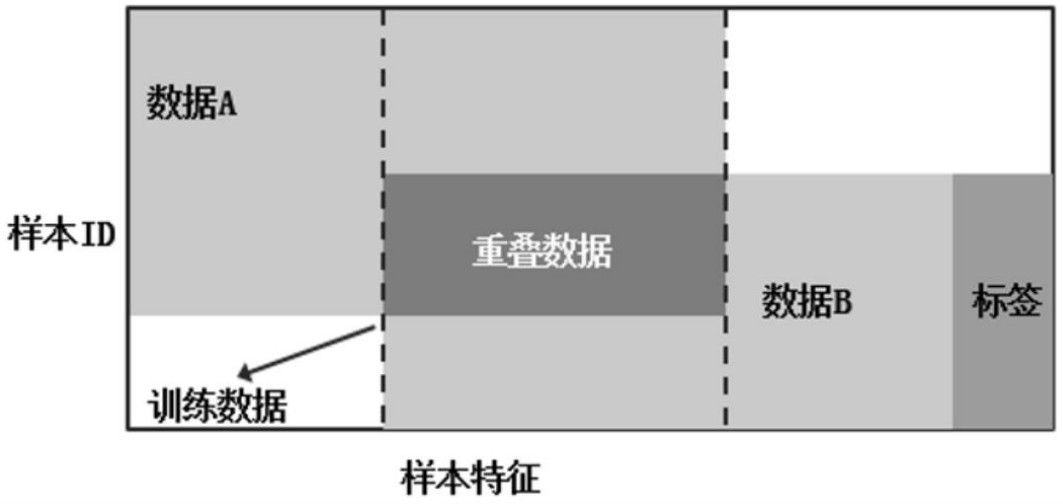
# 基本概念

- 联邦学习（ Federated Learning ）
  - 联邦学习思想
    - 是一种机器学习设定，其中许多客户端（例如，移动设备或整个组织）在中央服务器（例如，服务提供商）的协调下共同训练模型，同时保持训练数据的去中心化及分散性
- 联邦学习分类
  - 按照应用场景区分：跨孤岛、跨设备

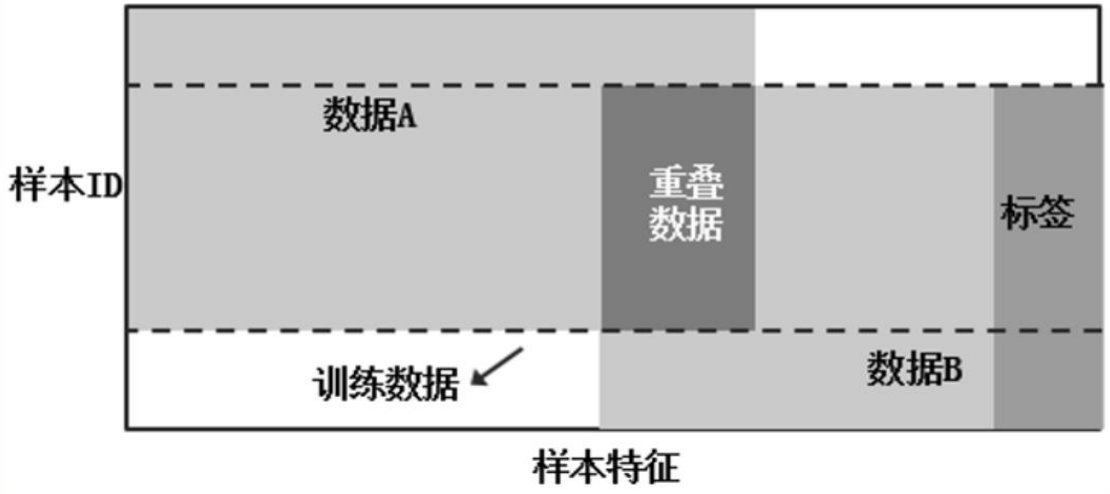
| | 跨孤岛 | 跨设备 |
|---|---|---|
| 例子 | 医疗机构 | 手机端应用 |
| 节点数量 | 1~100 | 1~10^10 |
| 节点状态 | 节点几乎稳定运行 | 大部分节点不在线 |
| 主要瓶颈 | 计算瓶颈和通信瓶颈 | 传输速度和设备状态 |

# 基本概念

- 联邦学习分类
  - 按照参与各方数据分布区分：横向联邦、纵向联邦
    - 横向联邦：参与者特征重叠较多，本身ID重叠较少。比如业务相同但是分布在不同地区的两家企业
    - 纵向联邦：本身ID重叠较多，特征重叠较少。比如同一个地区不同业务的两家企业




特征重叠较多




本身ID重叠较多

# 基本概念

- 非独立同分布（Non-IID）
  - 特征：
    - 不同客户端数据分布不同：
      - 特征分布倾斜：不同人的笔迹不同
      - 标签分布倾斜：企鹅只在南极、北极熊只在北极
      - 标签相同特征不同：概念飘移
      - 特征相同标签不同：点头表示Yes / No
    - 数量不平衡
    - 数据偏移：训练集测试集不同分布
    - 非独立：可用节点大多在某一时间或地点

# 算法原理

# 算法原理 FedAvg

| | |
|---|---|
| T | 原始数据不交互的前提下联合训练模型 |
| I | 参与方各自数据 |
| P | 1、选择客户端传播模型数据<br>2、客户端根据模型SGD计算参数<br>3、平均化聚合参数<br>4、共享模型 |
| O | 更新后的全局模型 |

| | |
|---|---|
| P | 将局部模型整合成为全局模型 |
| C | 参与方有标签和重叠的特征属性（符合横向联邦） |
| D | 参与方数据非独立同分布及不平衡 |
| L | JMLR 2017 |

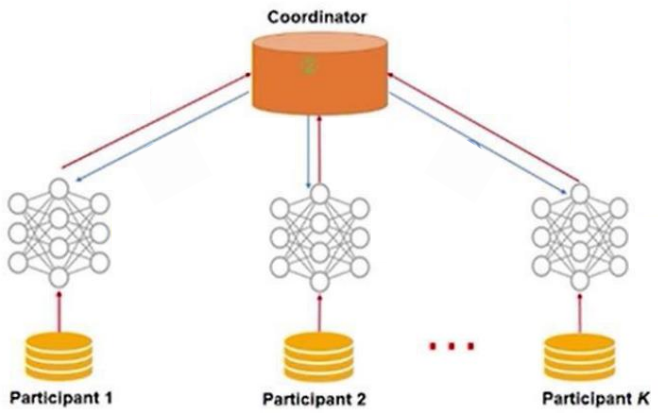
# 算法原理 FedAvg

- 算法流程：
  - 初始化权重$w_0$
  - 进行第t轮模型融合，从所有客户端中随机选取$S_t$个客户端
  - 基于$w_0$在每个客户端上进行E轮训练，得到新模型参数$w_{t+1}^k$
  - 对k个模型取平均得到$w_{t+1}$
  - 依次往复，进行下一轮模型融合

**Algorithm 1** `FederatedAveraging`. The $K$ clients are indexed by $k$; $B$ is the local minibatch size, $E$ is the number of local epochs, and $\eta$ is the learning rate.

**Server executes:**
initialize $w_0$
**for** each round $t = 1, 2, \dots$ **do**
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ (random set of $m$ clients)
  **for** each client $k \in S_t$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

**ClientUpdate**$(k, w)$: *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size $B$)
**for** each local epoch $i$ from 1 to $E$ **do**
  **for** batch $b \in \mathcal{B}$ **do**
    $w \leftarrow w - \eta \nabla \ell(w; b)$
return $w$ to server



# 算法原理 FedAvg

- 超参数
  - C：客户端数量的占比，相当于在K个客户端中提取个数的比例， C=0表示只有一个客户端
  - B：客户端训练数据的最小批量大小
  - E：客户端训练数据每一轮迭代次数

- FedSGD（baseline）
  - 每轮随机选择的客户端进行一次SGD梯度计算，得到k个梯度后更新服务器上的总梯度
  - 当B = ∞，E = 1 时FedAVG和FedSGD等价

- 数据集
  - MNIST
    - 模型：2个隐藏层的感知机（2NN）、CNN
    - 数据：
      - 独立同分布（IID）：随机打乱，逐个分配
      - 非独立同分布（Non-IID）：0-9数据排序，逐个分配
  - Shakespeare
    - 模型：LSTM
    - 数据：
      - 独立同分布（IID）：随机打乱，逐个分配
      - 非独立同分布（Non-IID）：按照角色人物分类，逐个分配

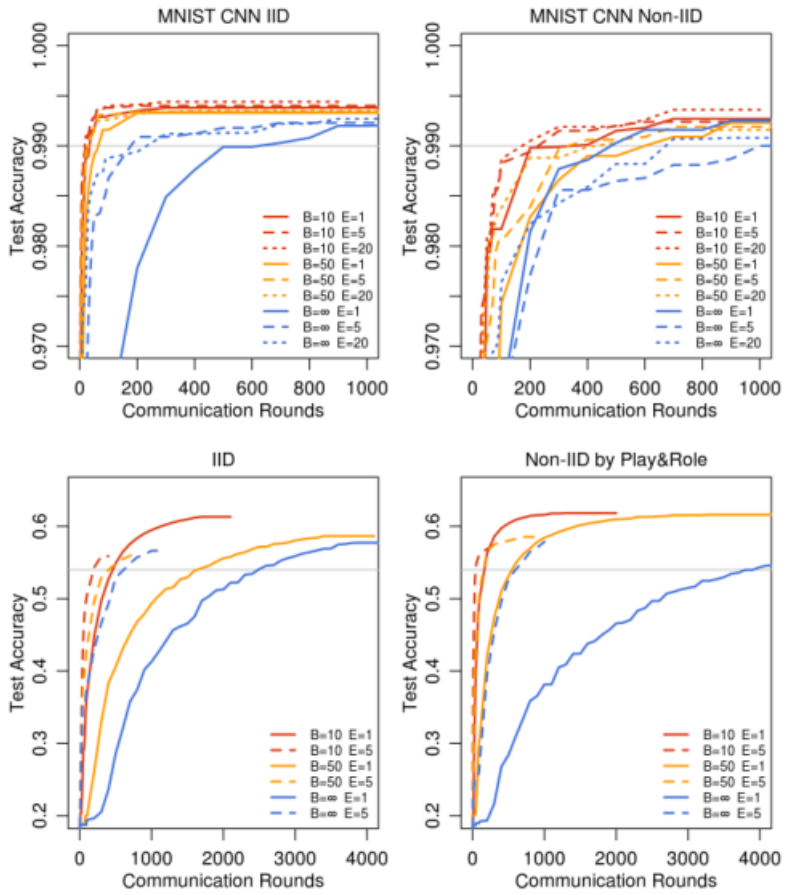
# 算法原理 实验结果

- 联邦学习的并行性-C
  - 评价标准
    - 实现测试集精度（2NN为97%，CNN为99%）所需的通信轮数，以及相对于C=0基线的加速比
  - 实验结果
    - C=0.1时训练轮次明显减少
    - 一些大批量的任务未在规定时间内达到目标精度

| **2NN** | IID | | Non-IID | |
|---|---|---|---|---|
| $C$ | $B=\infty$ | $B=10$ | $B=\infty$ | $B=10$ |
| 0.0 | 1455 | 316 | 4278 | 3275 |
| 0.1 | 1474 (1.0×) | 87 (3.6×) | 1796 (2.4×) | 664 (4.9×) |
| 0.2 | 1658 (0.9×) | 77 (4.1×) | 1528 (2.8×) | 619 (5.3×) |
| 0.5 | — (—) | 75 (4.2×) | — (—) | 443 (7.4×) |
| 1.0 | — (—) | 70 (4.5×) | — (—) | 380 (8.6×) |
| **CNN**, $E=5$ | | | | |
| 0.0 | 387 | 50 | 1181 | 956 |
| 0.1 | 339 (1.1×) | 18 (2.8×) | 1100 (1.1×) | 206 (4.6×) |
| 0.2 | 337 (1.1×) | 18 (2.8×) | 978 (1.2×) | 200 (4.8×) |
| 0.5 | 164 (2.4×) | 18 (2.8×) | 1067 (1.1×) | 261 (3.7×) |
| 1.0 | 246 (1.6×) | 16 (3.1×) | — (—) | 97 (9.9×) |

# 算法原理 实验结果

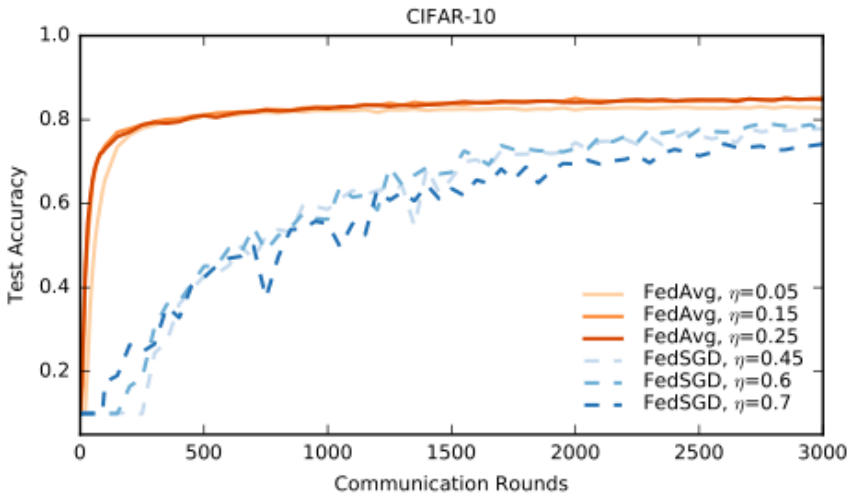
- 每个客户端计算量-B&E
  - 评价标准
    - 固定C=0.1，测试不同训练批大小和训练次数对测试准确率的影响
  - 实验结果
    - 相比于基线FedSGD，合适的B和E可以更快的达到目标准确率，甚至提升模型的最终效果（联邦平均产生了dropout效果）
    - 训练模型通信开销减少



# 算法原理 实验结果

- 鲁棒性实验
  - 实验数据：CIFAR-10
  - 实验模型：Tensorflow教程模型（2层卷积、2层全连接）
  - 实验参数：SGD使用批量为100，FedSGD和FedAvg中C=0.1，FedAvg中B=50，E=5
  - 实验结果：FedAvg以极少的训练轮次达到目标准确率

| ACC. | 80% | 82% | 85% |
|---|---|---|---|
| SGD | 18000 (—) | 31000 (—) | 99000 (—) |
| FEDSGD | 3750 (4.8×) | 6600 (4.7×) | N/A (—) |
| FEDAVG | 280 (64.3×) | 630 (49.2×) | 2000 (49.5×) |



# 算法原理 优劣分析

- 优势：
  - 在互相不接触数据的前提下，使用相对较小的通信轮数完成了模型的训练
  - 通过将计算量分配给客户端，从而减少通讯消耗，相比于FedSGD大大缩短训练的耗时，甚至可以提升模型本身的准确性。

- 局限性：
  - 异构性考虑不充分，没有考虑设备异构导致训练无法在规定时间完成的问题。
  - 未采用差分隐私或加密等技术，通信过程参数可能被窃取，一定程度上暴露各个客户端的隐私数据。

# 算法原理

# 算法原理 FedProx

| | |
|---|---|
| T | 解决联邦学习中的异构性 |
| I | 参与方各自数据 |
| P | 1、选择客户端传播模型数据<br>2、客户端添加近端项计算各自梯度<br>3、平均化聚合参数<br>4、共享模型 |
| O | 更新后的全局模型 |

| | |
|---|---|
| P | 不同参与方无法同步完成更新，还容易使本地模型偏离全局模型，影响全局收敛 |
| C | 参与方有标签和重叠的特征属性（符合横向联邦） |
| D | 设备间和数据间的异构特性 |
| L | ARXIV 2020 |

# 算法原理 FedProx

- 算法流程
  - 初始化权重$x_0$
  - 进行第t轮模型融合，从所有客户端中随机选取$S_t$个客户端
  - 基于$x_0$在每个客户端上计算$\gamma$非精确解，得到新模型参数$x_k^t$
  - 对k个模型取平均得到$x_t$
  - 依次往复，进行下一轮模型融合

**Algorithm 1:** FedProx

**Input:** $K$, $M$, $T$, $\mu$, $\gamma_k^t$, $\boldsymbol{x}^0$, $p_k$.

**Output:** $\boldsymbol{x}^T$.

**for** *each round* $t$ *from* 1 *to* $T$ **do**
  $\mathcal{S}_t \leftarrow$ random set of $M$ local models, each model $k$ chosed with probability $p_k$;
  **for** *each local model* $k \in \mathcal{S}_t$ *in parallel* **do**
    $\boldsymbol{x}_k^t \leftarrow \gamma_k^t$-inexact solution of $\min_{\boldsymbol{x}} H_k(\boldsymbol{x}; \boldsymbol{x}^{t-1})$;
  **end**
  $\boldsymbol{x}^t \leftarrow \frac{1}{M} \sum_{k \in S_t} \boldsymbol{x}_k^t$;
**end**
**return** $\boldsymbol{x}^T$;

# 算法原理 FedProx

- 与FedAvg比较：
  - 解决问题：若一些局部模型没能在规定时间内完成训练，FedAvg会丢弃这些模型，从而损失全局模型的准确率。
  - 算法改进：
    - 引入$\gamma$非精确解，若$H_k$是局部模型k的目标函数，$\gamma$越小意味着局部模型的训练完成度越高。FedProx算法允许$\gamma$随着局部模型(k)的不同和训练回合数(t)的不同产生变化，从而允许出现没有完成训练的局部模型。

$$\|\nabla H_k(\boldsymbol{x}^*;\boldsymbol{x}^t)\| \leq \gamma_k^t \|\nabla H_k(\boldsymbol{x}^t;\boldsymbol{x}^t)\|$$

    - $F_k$之后加上了近端项$\frac{\mu}{2}\|x-x^t\|^2$。由于数据异构性，局部模型进行过多的更新可能会导致全局模型不收敛，而近端项则会对偏离全局模型太多的局部模型进行惩罚。

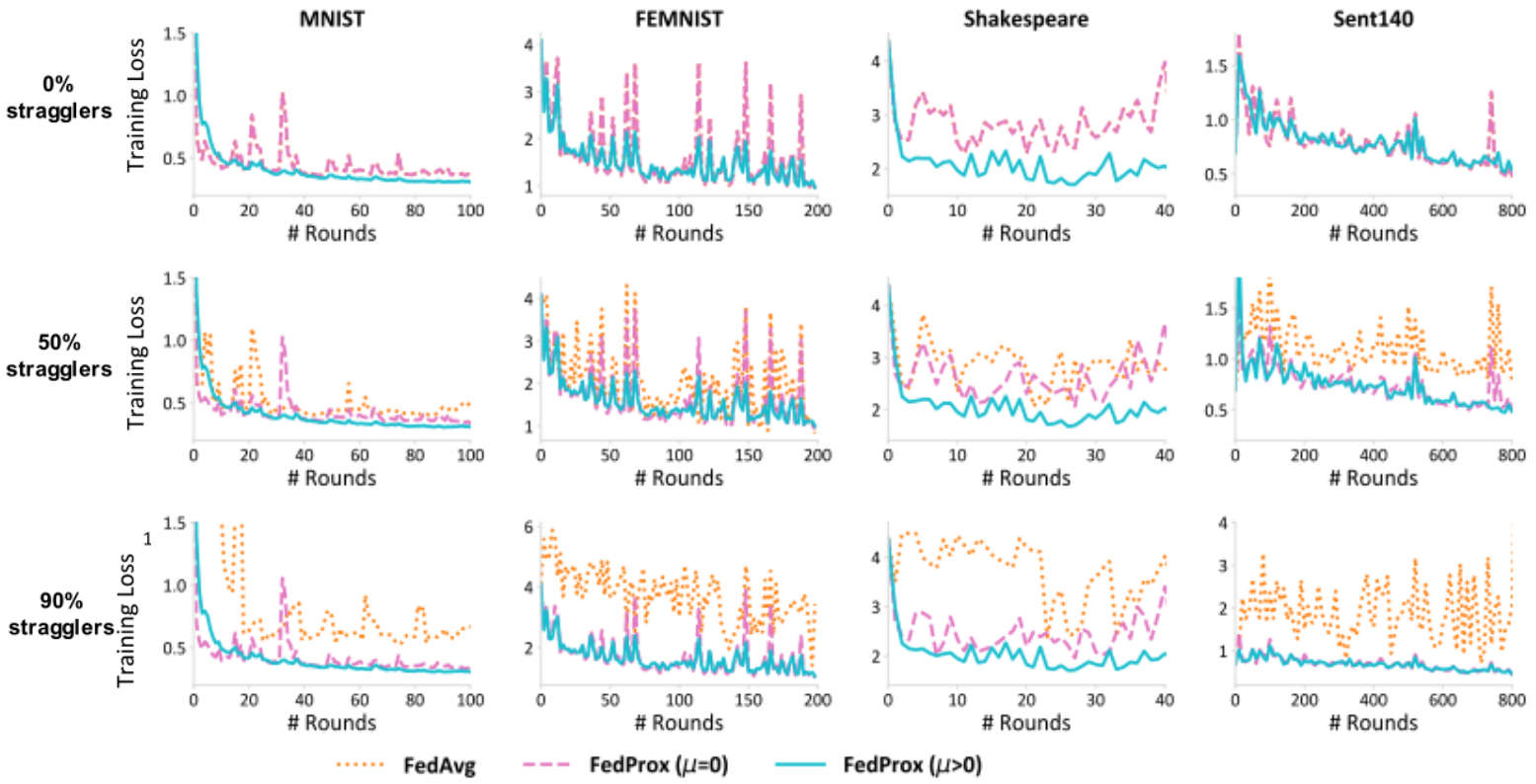
$$H_k(\boldsymbol{x};\boldsymbol{x}^t) = F_k(\boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{x}-\boldsymbol{x}^t\|^2$$

- 数据集
  - 真实数据集
    - MNIST（图像）、FEMNIST（图像）
    - Shakeespare（文本）、Sent140（文本）

| Dataset | Devices | Samples | Samples/device | |
|---|---|---|---|---|
| | | | mean | stdev |
| MNIST | 1,000 | 69,035 | 69 | 106 |
| FEMNIST | 200 | 18,345 | 92 | 159 |
| Shakespeare | 143 | 517,106 | 3,616 | 6,808 |
| Sent140 | 772 | 40,783 | 53 | 32 |

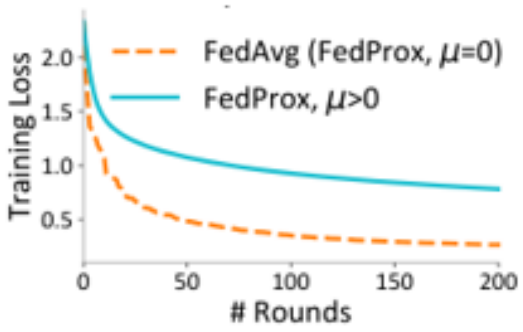
# 算法原理 实验结果

- 不同设备异构性影响
  - 方法：强制不能完成样本训练的数量，模拟设备性能差异
  - 实验结果：
    - FedProx下可变客户端更容易达到收敛
    - μ＞0时，存在近端项，收敛结果更稳定

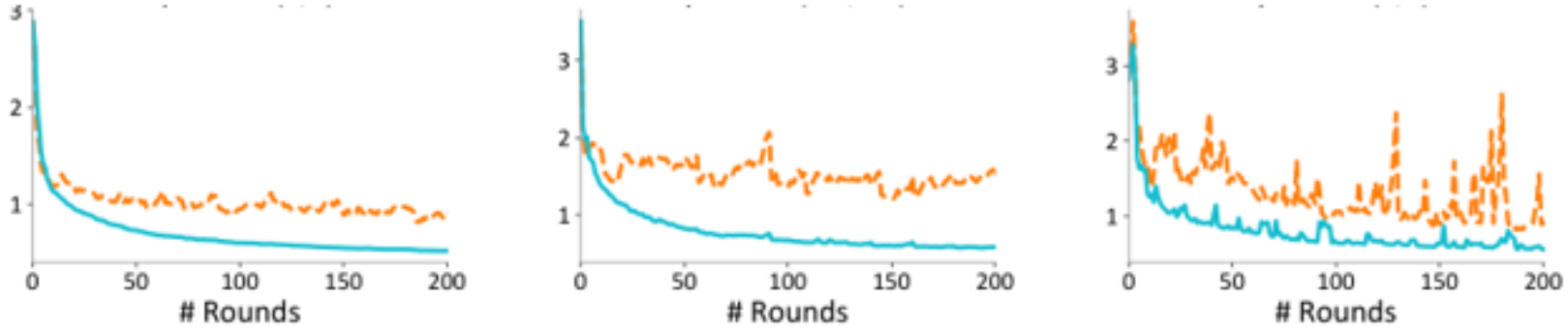


# 算法原理 实验结果

- 不同数据异构性影响
  - 方法：强制完成训练样本数量一致，改变数据分布
  - 实验结果：
    - μ＞0时，存在近端项，加快异构数据的收敛




独立同分布数据（IID）




非独立同分布数据（Non-IID）

- 优势：
  - 引入近端项，缓解了数据异构性，提高了全局模型收敛的稳定性。
  - 引入非精确解，缓解了设备异构性，提高联邦学习的泛化性。
- 局限性：
  - 未采用差分隐私或加密等技术，通信过程参数可能被窃取，一定程度上暴露各个客户端的隐私数据

# 应用总结

# 应用总结

- 应用
  - 减少分布式机器学习过程中数据隐私泄露可能
  - 为数据分散在不同区域形成的数据孤岛问题提供一种解决方案
  - 构建大数据和人工智能的跨企业，跨数据和跨域生态圈提供良好的技术支持
- 拓展方向
  - 纵向联邦学习的参数更新和模型优化
  - 联邦学习原始数据差分隐私及参数传递过程中加密
  - 高效的利用有限的通讯带宽
  - 可信追溯

# 参考文献

[1] Mothukuri V, Parizi R M, Pouriyeh S, et al. A survey on security and privacy of federated learning[J]. Future Generation Computer Systems, 2021, 115: 619-640.

[2] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. Federated optimization in heterogeneous networks. arXiv preprint arXiv:1812.06127.

[3] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.

[4] https://github.com/tao-shen/Federated-Learning-FAQ/

[5] https://blog.csdn.net/doyouseeman/article/details/108741299

大成若缺，其用不弊。
大盈若冲，其用不穷。
大直若屈。大巧若拙。
大辩若讷。静胜躁，寒
胜热。清静为天下正。

# 谢谢！