异构设备间的联邦原型学习

Yue Tan , Guodong Long , Lu Liu , Tianyi Zhou ,Qinghua Lu , Jing Jiang and Chengqi Zhang

摘要

摘要：设备间的异构性往往会阻碍联邦学习的优化收敛和泛化性能，例如，设备可能在数据分布、网络延迟、输入/输出空间或模型架构方面存在差异，这很容易导致其局部梯度聚合的结果存在偏差，为了适应客户间的异构性，提出了一种新的联邦原型学习 (FedProto) 框架，在该框架中，设备和服务器通信的是类原型而不是梯度。FedProto 聚合从不同设备收集的局部原型，然后将全局原型发送回所有设备，对局部模型的训练进行正则化。在每个设备上的训练旨在最小化本地数据的分类错误，同时保持生成的本地原型与相应的全局原型足够接近。通过实验，提出了一种针对异构联邦学习的基准设置，其中 FedProto 在多个数据集上的性能优于最近的几种联邦学习方法。

关键词：关键词 1；异构联邦学习 关键词 2：原型学习

1 引言

联邦学习 (FL) 旨在在集中式服务器上训练一个全局模型，而所有数据都分布在许多本地设备上，出于隐私或通信考虑，不能自由传输。FL 的迭代过程分为两个步骤:(1) 每个局部设备由全局模型同步，然后使用其局部数据进行训练;(2) 服务器通过聚合所有的局部模型来更新全局模型。考虑到模型聚合发生在梯度空间，传统的 FL 由于数据和模型的异质性仍然存在一定的现实挑战。能够同时克服这两个挑战的有效算法还没有完全开发出来或被系统地研究过。

为了开发一个具有鲁棒性的 FL 框架，结合前面提到的在通信效率和隐私保护方面的异构 FL 挑战，提出了一个新的基于聚合的原型 FL 框架，它可以在服务器和设备之间传输原型。提出的解决方案不需要聚合模型参数或梯度，因此它有可能成为各种异构 FL 场景的鲁棒性框架。因此，每个设备可以有不同的模型架构和输入/输出空间，但它们仍然可以通过共享原型来交换信息。在一个抽象的隐藏空间中，每个原型通过同一个类中训练样本的平均表示来表示一个类。在不公开模型参数的情况下，嵌入结果无法使用上传的原型重构。传输原型的通信开销通常比发送模型参数的通信开销小得多。

2 相关工作

2.1 异构联邦学习

跨设备的数据异构性 (也称为非 iid 问题) 是 FL 最重要的挑战。FedProx[1]提出了一个本地优化项以优化每个设备的本地模型。最近的一些研究[2][3]使用全局共享信息和每个设备的个性化部分训练个性化模型。另一种方法是通过聚类局部模型来提供多个全局模型[4][5]。模型结构的异构性是 FL 的另一个主要挑战，最近提出的基于蒸馏的联邦学习[6][7]可以解决这一挑战。特别是假设在联邦设置中添加一个共享数据集，这些基于知识蒸馏的 FL 方法可以将知识从教师模型提取到具有不同模型架构的学生模型。最近的一些研究也试图将神经结构搜索与联邦设置结合起来, 可以为每组具有不同计算能力和配置的设备组配置合适的模型。总之，上述提到的大部分 FL 方法都只关注一种具有挑战性的异构场景。它们都使用了基于模型参数的聚合方法，但是会引起较大通信开销和梯度攻击。

2.2 原型学习

原型的概念 (多个特征的平均值) 已经在各种任务中进行了探索。在图像分类中，原型可以作为一个类的代理，并计算为每个类内特征向量的平均值[8]。在动作识别中，可以将不同时间戳的视频特征取平均作为视频的表示[9]。聚合的局部特征可以作为图像检索的描述符[10]。使用原型来表示分布式机器学习中的任务无关信息，并提出一种新的融合范式来集成这些原型，从而为新任务生成新模型。在本文中，我们借用原型的概念来表示一个类别，并将原型聚合应用于异构 FL 的设置中[11]。

一般来说，原型学习广泛应用于数据有限的训练场景，这种学习场景与跨设备 FL 的假设是一致的: 每个设备都有有限数量的实例来独立训练一个具有理想性能的模型。

3 本文方法

3.1 本文方法概述

在 FL 中，每个设备使用自己的私有数据集 D_i 来训练模型 $F_i(x;W_i)$ 由可学习权重 W_i 和输入特征 x 参数化。在 FedAvg 中，不同模型之间的 F_i 共享相同的参数和模型架构。然而，本方法所提出的假定是不同设备间的 F_i 是不相同的，对于第 i 个客户，在训练过程中需要最小化的损失是

$$\arg\min_{W_i} L_s(F_i(x;W_i),y)$$

在 FedAvg 中，FL 的目标函数可以写为

$$\arg\min_{W} \sum_{i}^{m} \frac{|D_i|}{\sum_k^K |D_k|} L_s(F_i(x;W_i),y)$$

F_i 具有不同的模型架构这一事实将导致 W_i 具有不同的格式和大小。因此，全局模型的参数 W 不能通过平均 W_i 来优化。所以提出基于原型聚合的联邦学习方法。

3.2 基于原型的聚合

一般来说，基于深度学习的模型由两部分组成:(1) 表示层或嵌入函数或编码器，用于对原始特征空间的输入进行编码或转换到新的特征空间;(2) 决策层，用于对给定的学习任务进行分类决策。定义原型 C_j 表示 C 中的第 j 个类，对于第 i 个设备，原型是第 j 类实例的嵌入向量的平均值。

$$C_j^{(i)} = \frac{1}{|D_{i,j}|} \sum_{(x,y)\in D_{i,j}} f_w(x)$$

其中，$D_{i,j}$ 是本地数据集 D_i 的子集，由本地数据中第 j 类训练数据组成。在学习任务的推理阶段，我们可以通过测量实例的特征向量与原型之间的 L2 距离，简单地预测实例 x 的标签。

3.3 FedProto 框架

在 FL 中使用原型，我们不需要交换梯度或模型参数，这意味着提出的解决方案可以处理异构的模型架构。服务器接收来自 m 个本地设备的原型集合 $C^{(1)},C^{(2)},\ldots,C^{(m)}$ 并将它们进行聚合。聚合后得到的模型再发送给每一个客户端。FedProto 算法概述如图 1 所示：

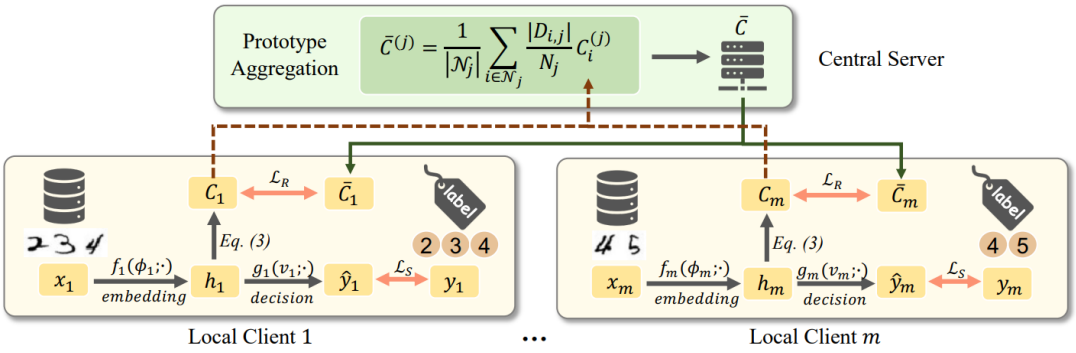

图 1: 方法概述图

异构设备上联邦原型学习的目标可以表述为

$$\arg\min_{\{\overline{C}_j\}_{j=1}^{|C|}} \sum_{i=1}^{m} \frac{n_i}{N} L_s(F_i(x;W_i),y) + \lambda \cdot \sum_{j=1}^{|C|} \sum_{i=1}^{m} \frac{n_{i,j}}{N_j} R(\overline{C}_j^{(i)}, C_j^{(i)})$$

其中 $L_s(\cdot)$ 是本地学习任务的损失。$R(\cdot,\cdot)$ 测量本地原型 $C^{(i)}$ 和对应的全局模型 $\overline{C}_i$ 之间的 $L2$ 距离。n_i 是设备 i 上的实例数量。$n_{(i,j)}$ 是设备 i 上第 j 类实例的数量。N 是所有设备数据的数量。N_j 是所有设备第 j 类的数量。

设备需要通过考虑监督学习的损失和正则化项来更新本地模型，从而在设备和服务器之间生成一致的原型。在局部损失的基础上增加正则化项，使局部原型 $C_j^{(j)}$ 趋近全局原型 $\overline{C}_j^{(i)}$，同时使分类误差损失最小化。损失函数定义如下:

$$L(D_i,W) = L_s(F_i(x;W_i),y) + \lambda \cdot R(\overline{C}_j^{(i)}, C_j^{(i)})$$

其中 $L_s(\cdot)$ 是监督学习的损失，λ 是原型距离的权重，$R(\cdot,\cdot)$ 是正则化项，可以定义为：

$$R = \sum_j d(\overline{C}_j^{(i)}, C_j^{(i)})$$

其中 d 是本地产生的原型和全局原型之间的距离。

4 复现细节

4.1 与已有开源代码对比

复现过程实现本文所提出算法 FedProto，同时使用先前的 FL 算法 FedAvg 进行实验数据对比，每个算法都使用 MNIST，FEMNIST，CIFAR10 这三个数据集。每个实验使用 20 个客户端，每个客户端从指定的数据集中抽取平均数为 n，标准差为 stdev 的种类数，每个类所对应的数据平均数为 100，使用这种抽样方法，确保了每个客户端所拥有的数据与其他客户端相比是完全不一样的，模拟了数据的异构性。对所有客户端的训练轮数为 rounds，n 取不同值，进行 rounds 轮训练之后，所得到的精度存在差异。Fedproto 具有聚合不同结构的模型的能力，所以设计实验 FedProto_mh，在该实验中，将客户端分为两部分，每个部分的客户端模型的架构是不同的，通过聚合原型，实现了不同结构模型之间的聚合。

文中只考虑到了客户端对模型进行一轮训练的情况，但是在实际的应用中，客户端的训练轮数是不止一轮的，适当地增加客户端的训练轮数，可以更好地利用本地的计算资源，同时也能减少一定的全局训练轮数，通信量也就随之下降，并且还能达到与原本的方法相似的精度。具体的做法是，在每

一轮全局训练结束之后，对所有的客户端损失进行排序，取损失最小的前十五个客户端。对这十五个客户端，在下一轮的全局训练中，本地的训练轮数上调至 2，其余的客户端本地训练轮数保持不变，以此类推，直到训练结束。

4.2 创新点

在实际应用当中，应该考虑适度增加训练轮数，参与训练的客户端存在着系统异构性的问题，不同的客户端在计算能力，通信能力因客户端的不同以及时间的变化而不同。对于本地的数据，客户端仅进行一轮训练是不够的，考虑到系统异构性问题，在每一轮全局训练中选择训练效果较好的客户端，让其在下一轮训练中进行多轮训练，充分利用了客户端的计算和通信能力。

5 实验结果分析

Dataset	Method	Stdev	n=3	n=4	n=5	rounds
MNIST	FedAvg	2	95.1±6.32	94.3±4.58	93.2±4.25	120
	FedProto	2	96.9±0.72	97.0±0.22	96.5±0.16	100
	FedProto_mh	2	97.1±0.42	96.6±0.25	96.1±0.40	100

表 1: MNIST 实验结果

Dataset	Method	Stdev	n=3	n=4	n=5	rounds
FEMNIST	FedAvg	1	94.4±5.24	91.3±5.16	90.9±7.16	120
	FedProto	1	96.7±1.83	94.9±1.47	93.5±2.12	110
	FedProto_mh	1	97.2±1.49	94.7±1.50	97.8±2.39	110

表 2: FEMNIST 实验结果

Dataset	Method	Stdev	n=3	n=4	n=5	rounds
CIFAR10	FedAvg	1	80.5±3.26	77.8±2.53	74.9±2.94	110
	FedProto	1	84.3±2.11	79.1±2.25	77.2±2.12	100
	FedProto_mh	1	83.2±1.56	79.6±1.64	76.8±1.29	100

表 3: CIFAR10 实验结果

表 1，表 2，表 3 分别是不同的算法在 MNIST,FEMNIST,CIFAR10 三个数据集上运行之后的结果。FedProto_mh 表示算法在客户端拥有不同模型的情况下运行的模式，在该模式中，10 个用户使用 2 个卷积层和 2 个全连接层的模型，另外 10 个模型使用 LeNet 模型。可以看出，在客户端平均种类数量 n 为 3，4，5 时，FedProto 和 FedProto_mh 相比先前的 FedAvg 算法能够在更少的训练轮数 rounds 中实现更高的精度。

Method	Stdev	n=3	n=4	n=5	rounds
FedProto	2	96.9±0.72	97.0±0.22	96.5±0.16	100
FedProto_mr	2	97.3±0.62	97.0±0.34	95.9±0.27	70

表 4: 改进结果对比

在原算法的基础上，对于每一轮训练，取训练误差最小的 15 个客户端，说明这 15 个客户端在训练中有着较好的表现，所以将它们在下一轮的训练中本地训练轮数设为 2，直到训练结束，将该方法命名为 FedProto_mr，可以看到相比 FedProto，该方法使用更少的全局训练轮数就达到了比原算法略高或相近的精度。

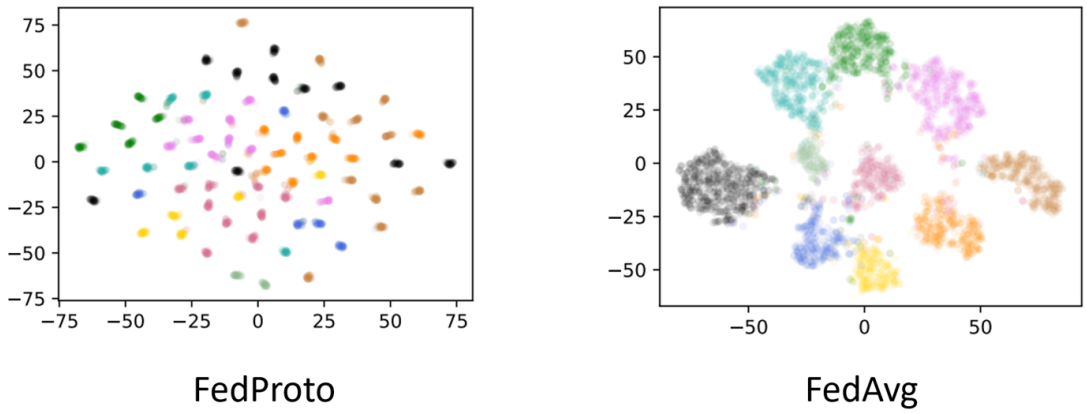

图 2: t-SNE 图

图 2 展示了利用 t-SNE 对 MNIST 测试集中的样本和全局原型进行可视化。不同颜色的点代表不同类别的数据，从 FedProto 和 FedAvg 的结果图中可以看出，FedAvg 相较 FedProto 有着更强的泛化性，但是 FedProto 更加个性化。由此可见，FedProto 使得不同客户端的模型更加个性化。

6 总结与展望

本文档讲述了 FedProto 算法的提出背景，相比以往方法所取得的改进以及该算法的具体架构。接着讲述对该算法的复现过程，以及做出的一些改进并展示了各个实验的结果。未来进一步的研究可以设置评估标准，在服务器端将原型相似度高的客户端原型进行聚合，进一步实现模型的个性化。

参考文献

[1] LI T. Federated optimization in heterogeneous networks[J]. MLSys, 2020.

[2] ARIVAZHAGAN M G. Federated Learning with Personalization Layers[J]. arXiv, 2019.

[3] LIANG P P. Think locally, act globally: Federated learning with local and global representations[J]. NeurIPS, 2020.

[4] MANSOUR Y. Three approaches for personalization with applications to federated learning[J]. arXiv:2002.10619, 2020.

[5] GHOSH A. An efficient framework for clustered federated learning[J]. NeurIPS, 2020.

[6] JEONG E. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data[J]. NeurIPS, 2018.

[7] LI D. Fedmd: Heterogenous federated learning via model distillation[J]. NeurIPS, 2020.

[8] SATTLER F. Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints[J]. IEEE Transactions on Neural Networks, 2019.

[9] Simonyan. Two-stream convolutional networks for action recognition in videos[J]. NeurIPS, 2014.

[10] Babenko. Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. [J]. ICCV, 2015.

[11] HOANG N. Learning Task-Agnostic Embedding of Multiple Black-Box Experts for Multi-Task Model Fusion[J]. ICML, 2020.