arXiv:2303.12296v1 [cs.LG] 22 Mar 2023

原型助力联邦学习：迈向更快收敛

Yu Qiao[1], Seong-Bae Park[2],Sun Moo Kang[2],和 Choong Seon Hong[2*1] 大韩民国龙仁市 *17104*，庆熙大学人工智能系[2] 大韩民国龙仁市 *17104*，庆熙大学计算机科学与工程系电子邮件： {qiaoyu, sbpark71, etxkang, cshong}@khu.ac.kr

摘要

联邦学习（FL）是一种分布式机器学习技术，多个客户端通过协作训练共享模型，而无需交换其原始数据。然而，客户端之间数据分布的异质性通常会导致模型推理效果不佳。本文提出了一种基于原型的联邦学习框架，只需对典型联邦学习流程的最后一次全局迭代进行少量改动，即可实现更好的推理性能。在最后一次迭代中，服务器聚合来自分布式客户端的原型，然后将其发送回本地客户端，用于各自的模型推理。在两个基准数据集上的实验表明，在不同的异质性设置下，与两种流行的基线方法相比，我们提出的方法能够取得更高的准确率（至少提高 1%），同时通信效率相对较高。

I. 引言

联邦学习（FL）最初由谷歌于 2016 年提出，最初用于解决安卓终端的更新问题。其初衷是基于分布在多个计算节点上的数据集开展隐私保护机器学习（ML） [1]。本质上，联邦学习是一种分布式机器学习范式。随着人工智能（AI）的进一步发展，联邦学习的概念也得到进一步完善和发展。联邦训练的基本特征是，各方数据均保留在本地，不会泄露隐私信息。此外，联邦场景中客户端的数据分布通常具有异质性。因此，面临的关键挑战是：客户端之间的数据分布通常并非独立同分布（非独立同分布），这可能导致模型推理性能不佳 [2], [11]。

目前已有多项研究尝试解决非独立同分布问题。FedAvg [1] 是首个面向联邦场景的优化算法。MOON [3] 采用对比损失，以缩小全局模型和本地模型所学习的表征之间的差异。FedNova [4] 则侧重于聚合阶段。它允许不同客户端在每次全局迭代中执行不同数量的本地训练轮数，然后在聚合所有本地更新之前，对其进行归一化和缩放，以确保聚合后的全局更新偏差较小。

受 [5], 启发，我们发现，在联邦设置中，模型的分类器比其他层具有更大的偏差。换言之，通过分类器进行的预测往往存在较大的偏差。因此，本文创新性地提出通过分类器的前一层进行预测。分类器前一层的输出是类别的特征空间，同一类别空间中特征空间的平均值可以定义为该类别的“原型”。他们的工作 [6] 表明，可以有效利用原型信息来解决联邦学习中的异质性。

主要贡献总结如下：

- 我们设计了一种新颖的基于原型的联邦学习框架，客户端像典型的联邦训练那样与服务器通信，但模型推理是基于聚合后的原型进行的，而非基于分类器。
- 我们提出了一种原型聚合方法，客户端可以在完成最后一轮联邦迭代之前，将其原型上传至服务器进行聚合。
- 在两个常用基准数据集 MNIST [7] 和 Fashion-MNIST [8],上的实验表明，我们提出的方法具有更高的测试准确率（至少高出 1%），且与两种常用基线方法相比，通信效率相对较高。

II. 所提出的框架

A. 问题陈述

考虑分布式边缘网络中的一个分布式客户端集合 ϕ，其拥有私有敏感数据集 $\mathcal{D}_i = \{(\boldsymbol{x}_i,\ y_i)$，数据集大小为 D_i。按照联邦学习的典型训练流程，客户端与边缘服务器协作训练共享模型$\mathcal{F}(\omega;\ \boldsymbol{x}_i)$，其中 ω 是全局模型的模型参数， $\boldsymbol{x}_i$ 表示一个客户端i的特征向量。我们的目标是最小化异构客户端上的损失函数，如下所示：

$$\arg\min_{\omega} \mathcal{L}(\omega) = \sum_{i\in\phi} \frac{D_i}{\sum_{i\in\phi} D_i} \mathcal{L}_i(\mathcal{F}(\omega;\boldsymbol{x}_i), y_i), \quad (1)$$

其中， y_i 表示样本的标签， $\mathcal{L}_i$ 表示客户端 i的经验风险（例如交叉熵损失）。

B. 所提出的联邦学习框架

典型的联邦训练流程可概括如下：(1) 服务器将模型分发给各客户端进行本地训练；(2) 客户端分别更新模型参数，并将其发送回服务器进行聚合；(3) 服务器聚合上传的最新模型参数，然后再次将其发送给本地客户端，以进行下一轮全局迭代。重复上述过程，直至收敛。

我们的方案与这一典型流程相似，但在最后一个全局轮次中，客户端不仅需要像往常一样更新自己的

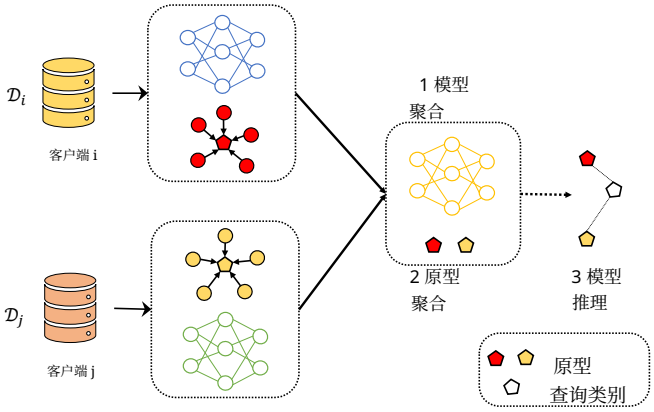


图1：所提出的基于原型的联邦学习框架概览。请注意，在此前的 T-1 个全局轮次中，客户端仅向服务器传输模型参数，以训练共享模型；这一过程未在图中展示。我们仅展示 T 次全局迭代（即最后一次全局迭代）中的训练过程。在第 T 轮中，客户端不仅向服务器传输模型参数，还传输其原型，以进行模型聚合（图中步骤 1）和原型聚合（步骤 2）。最后，可根据每个查询样本与聚合原型之间的距离进行预测（步骤 3）。

本地模型参数，还需要为每个类别计算自己的原型。计算得到的原型与模型参数一同发送到服务器进行聚合。最后，将聚合后的原型和最新模型参数发送回本地客户端，用于模型推理。图 1 展示了我们提出框架的总体概览。

C. 基于原型的模型推理策略

通常，深度学习模型由两部分组成：特征提取层和决策层。前者表示为 $f_e(\omega_e; \boldsymbol{x})$，后者表示为 $f_d(\omega_d; y)$。因此，共享模型可写为 $\mathcal{F}(\omega; \boldsymbol{x}) = f_d(f_e(\omega_e; \boldsymbol{x}) y)$，这意味着特征提取层的输出作为决策层的输入。

1) 原型计算：根据 [6], 原型的定义，表示客户端 i 的第 j类的原型可表示为:

$$\widetilde{y}_{i,j} = \frac{1}{D_{i,j}} \sum_{(x,y)\in\mathcal{D}_{i,j}} f_e(\omega_e; \boldsymbol{x}), \tag{2}$$

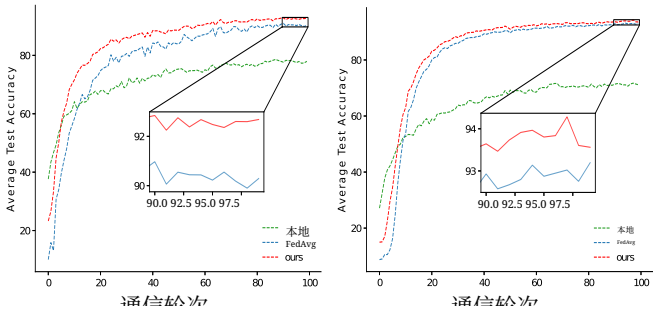
其中，$\mathcal{D}_{i,\ j}$ 是客户端 i属于第 j类的分布，而$D_{i,\ j}$ 是 $\mathcal{D}_{i,\ j}$的大小。

2）全局原型聚合：根据公式 2 的计算结果，这些计算得到的原型可以发送到服务器进行聚合，定义如下:

$$\overline{y}_j = \frac{1}{|\phi|} \sum_{i\in\phi} \widetilde{y}_{i,j}, \tag{3}$$

算法 1： 基于原型的联邦学习（ProtoFed）

输入：
数据集 $\mathcal{D}_i$, ω_i
模型训练：
初始化 ω^0, $\{y\}$。
对于 $t = 1, 2, ..., T$ **do**
  对于 $i = 0, 1, ..., N$ **并行执行**
    如果 $t = T$ **则**
      本地模型更新。
      根据公式（3）进行原型聚合。
    否则如果 $t < T$ **则**
      本地模型更新。
    结束如果
  结束循环
结束循环
模型测试：
对于测试数据集中的每个样本 i **执行**
  对于每个类别 j 在 $\{y\}$ **执行**
    测量 $f_e(\omega_e; \boldsymbol{x}_i)$ 与 y_j 之间的 L2距离
  结束循环
  根据公式（4）生成最终预测
结束循环



(a) $\alpha = 0.05$　　(b) $\alpha = 0.1$

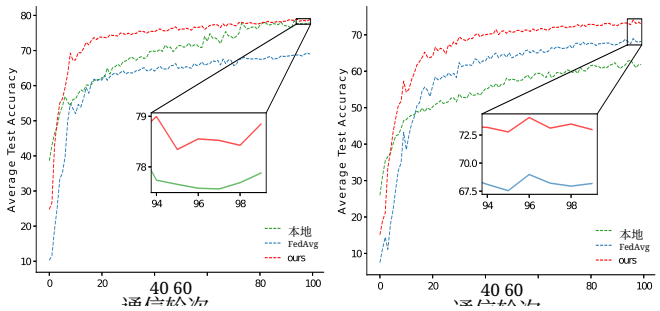
图2：在偏斜度为 $\alpha = 0.05$和 $\alpha = 0.1$的情况下，不同通信轮次中MNIST上所有方法的top-1平均测试准确率。

其中，$|\phi|$ 表示网络中的客户端数量。该公式表示对来自类别为 j 的客户端的原型求平均，得到全局原型。

3）基于原型的模型推理：公式3的结果可视为用于模型推理的每个类别的全局表示。因此，可以通过测量本地原型$f_e(\omega_e; \boldsymbol{x})$与全局表示原型$y_j$之间的L2距离来进行预测，表示如下： [10]

$$\hat{y} = \arg\min_j \|f_e(\omega_e; \boldsymbol{x}) - \overline{y}_j\|_2, \tag{4}$$

其中最终的预测标签记为 $\hat{y}$。基于原型的联邦学习的更多细节见算法1。



(a) $\alpha = 0.05$ (b) $\alpha = 0.1$

图 3：在偏斜度为 $\alpha = 0.05$和 $\alpha = 0.1$的情况下，不同通信轮次中各方法在 Fashion-MNIST 上的 Top-1平均测试准确率。

III. 实验

A. 数据集与本地模型

我们使用 MNIST [7] 和 Fashion-MNIST [8] 作为对比基准。MNIST 是一个包含 10 个数字、图像尺寸为 28x28x1 的手写数字识别数据集。Fashion-MNIST 比 MNIST 更复杂，涵盖 10 个类别的不同物品，共有 70,000 张图像，图像尺寸为 28x28x1。我们为 MNIST 和 Fashion-MNIST 使用一个包含 2 个卷积层和 2 个全连接层的 4 层卷积神经网络，与其工作 [5], [6]中采用的网络类似。

B. 实现细节

所有实验均使用 20 个客户端。MNIST 和 Fashion-MNIST 数据集的默认训练参数设为 $B = 8$、$E = 1$、$\eta = 0.01$，分别表示本地批量大小、本地训练轮数和学习率。我们从训练数据集中抽取 5000 个样本，并将其分配给所有客户端，以模拟现实世界中训练样本有限的情况 [6]。此外，我们采用狄利克雷分布 [9]来模拟客户端之间的异质性设置，可表示为 Dir(α)；其中α越小，客户端之间的数据分布越不均衡。

C. 准确率与通信效率比较

我们在不同异质性条件下，将所提出的方法与两种流行的基线方法 Local 和 FedAvg 进行比较。其中，Local 表示客户端各自独立训练模型，不与服务器或其他客户端进行任何通信；而 FedAvg 作为首个联邦算法，是最流行的基线方法。需要注意的是，为了比较各方法在每次全局迭代后的测试准确率，我们在训练期间的每一轮都进行基于原型的模型推理。

图 2 和图 3 分别展示了所有方法在 MNIST 和 Fashion-MNIST 上的准确率和通信效率对比。结果表明，在这两个数据集上，我们的基于原型的模型推理策略在不同异质性设置下，能够获得更高的测试准确率，并且收敛速度相对更快，且每个全局通信轮次的表现均优于其他方法。

具体而言，在 $\alpha = 0.05$ 时，我们的方法在 MNIST 上的准确率分别比 Local 和 FedAvg 高 18.9% 和 2.4%；在 $\alpha = 0.05$ 时，准确率也分别高出 31.5% 和 1.0%。此外，在 $\alpha = 0.05$ 时，我们的方法在 Fashion-MNIST 上的准确率分别比 Local 和 FedAvg 高 1.4% 和 14.3%；在 $\alpha = 0.1$ 时，准确率也分别高出 18.4% 和 7.4%。

IV. 结论

本文提出了一种基于原型的联邦学习框架，可提升模型推理性能。我们首先介绍原型计算方法，然后介绍原型聚合方法，最后提出基于原型的模型推理策略。实验表明，与 Local 和 FedAvg 这两种广泛使用的基线方法相比，我们的策略在 MNIST 和 Fashion-MNIST 上均能将准确率至少提高 1%，同时实现较高的通信效率。未来，我们的方法有望与其他最先进的方法相结合，并在更多数据集上进行测试。

参考文献

[1] McMahan B, Moore E, Ramage D 等。“基于去中心化数据的深度网络通信高效学习”，人工智能与统计学，第 1273–1282 页，2017 年。[2] Kevin Hsieh、Amar Phanishayee、Onur Mutlu 和 Phillip Gibbons。“去中心化机器学习中的非独立同分布数据困境”，国际机器学习会议，第 4387–4398 页，2020 年。[3] Li, Q., B. He, D. Song。“模型对比联邦学习”，*IEEE/CVF*计算机视觉与模式识别会议，2021 年。[4] J. Wang, Q. Liu, H. Liang, G. Joshi 和 H. V. Poor。“解决异构联邦优化中的目标不一致问题”，神经信息处理系统进展，第 33 卷，2020 年。[5] Luo M, Chen F, Hu D 等。“无需担忧异质性：面向非独立同分布数据联邦学习的分类器校准”，神经信息处理系统进展，第 34 卷，第 5972–5984 页，2021 年。[6] Tan Y, Long G, Liu L 等。“Fedproto：异构客户端间的联邦原型学习”，人工智能会议，第 1 卷，第 3 期，2022 年。[7] LeCun, Yann 等。“应用于文档识别的基于梯度的学习”，IEEE会议录，第 86 卷，2324，1998 年，第 11 期，2278 页。[8] Xiao, Han, Kashif Rasul 和 Roland Vollgraf。“Fashion-MNIST：用于机器学习算法基准测试的新型图像数据集”，arXiv 预印本 arXiv:1708.07747，2017 年。[9] Yurochkin M, Agarwal M, Ghosh S 等。“神经网络的贝叶斯非参数联邦学习”，国际机器学习会议，第 7252–7261 页，2019 年。[10] Qiao, Yu 等。“基于多原型的联邦学习框架：迈向边缘智能。”2023 年国际信息网络会议。IEEE，2023 年。[11] Qiao, Yu、Munir, Md Shirajum 等。“CDFed：用于管理系统异质性与统计异质性的基于贡献的动态联邦学习。”IEEE/IFIP网络运营与管理研讨会，2023 年。