

资源高效的聚类协同联邦学习客户端选择方法

李 强, 张凌羽, 孟祥宇

(吉林大学 计算机科学与技术学院, 长春 130012)

摘 要: 针对联邦学习中各客户端存在资源异构性和数据异构性的关键问题, 提出了一种资源高效的聚类协同联邦学习客户端选择方法。首先, 根据各客户端的计算能力将其分组, 在每轮训练中, 以每组客户端的平均准确率作为间接度量选择同组客户端; 其次, 在每组内根据各客户端的模型相似性对客户端进行聚类, 选择每组内不同聚类的客户端; 最后在真实数据集上评估本文方法的性能。实验结果表明: 该方法可以减少全局训练时间, 获得更快速、更平滑的收敛, 实现训练效率和全局模型准确率之间的良好平衡。

关键词: 计算机系统结构; 联邦学习; 客户端选择; 聚类; 资源异构; 数据异构

中图分类号: TP301 **文献标志码:** A **文章编号:** 1671-5497(2025)10-3337-09

DOI: 10.13229/j.cnki.jdxbgxb.20231369

Resource-efficient clustering collaborative federated learning client selection method

LI Qiang, ZHANG Ling-yu, MENG Xiang-yu

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract: To address the key issue in federated learning the resource heterogeneity and data heterogeneity of each client, this paper proposes a resource-efficient client selection method for clustered collaborative federated learning. Firstly, where each client is grouped according to its computational power, the average accuracy of each group of clients is used as an indirect metric to select clients in the same group in each round of training. Secondly, the clients are clustered according to the model similarity of each client within each group, and the clients in different clusters within each group are selected. Evaluate the performance of the method proposed in this paper on real datasets, the experimental results show that this method can reduce the global training time, obtain faster and smoother convergence, and achieve a good balance between training efficiency and global model accuracy.

Key words: computer system architecture; federated learning; client selection; clustering; resource heterogeneity; data heterogeneity

收稿日期: 2023-12-08.

基金项目: 吉林省科技厅创新平台(基地)和人才专项项目(20220508043RC).

作者简介: 李强(1975-), 男, 教授, 博士. 研究方向: 隐私保护, 安全计算, 人工智能安全, 网络攻击检测.

E-mail: li_qiang@jlu.edu.cn

通信作者: 孟祥宇(1988-), 男, 副教授, 博士. 研究方向: 网络与信息安全, 国产操作系统.

E-mail: xiangyumeng@jlu.edu.cn

0 引言

近些年来,现代移动和物联网设备每天都在产生大量数据,由于硬件计算能力的提高和大量数据的产生,人工智能技术飞速发展。在传统的集中式机器学习中,需要把分布在各个终端上的数据集中在中心节点上进行处理,若要训练性能较好的模型,则需以大量数据为基础。然而,除了少数大型组织或政府部门外,大多数个人或公司只有少量数据,并且集中收集所有的数据也会产生高昂的开销。同时,人们对数据安全和隐私保护愈发关注,如《通用数据保护条例》(General data protection regulation, GDPR)^[1]禁止将数据集中到中心节点,这使得传统的集中式机器学习难以应用于分散数据的收集和处理。此外,不同企业甚至同一企业的不同部门也可能出于数据隐私目的不愿意共享数据,从而形成“数据孤岛”。为了解决这些问题,谷歌提出了联邦学习^[2]这一概念。联邦学习要求服务器将随机初始化的全局模型下发到各个客户端设备,客户端设备不需要共享数据,直接使用本地数据更新模型后,将新的本地模型上传回服务器,服务器聚合收到的所有客户端的本地模型,形成新的全局模型,将其再次下发到全部客户端。通过联邦学习机制,可以在保障不泄露用户数据隐私和不影响数据规范的前提下,实现数据的合理利用。

然而,联邦学习模型的训练速度和准确率会受到资源异构性和数据异构性的影响,具体体现为:①参与训练的各个客户端在存储能力、计算能力、通信能力等多个方面存在差异,这些差异使得各个客户端的本地训练时间不同,甚至会出现设备掉线的情况,而联邦学习的整体训练时间会受到计算能力弱或通信速度慢的最慢客户端的限制,这一现象称为资源异构性。在分布式机器学习中,通常采用异步训练方式缓解资源异构性。但是,在联邦学习中,由于现有的隐私方法^[3-5]都建立在同步训练的假设下,因此异步训练方法在联邦学习中并不适用;②各个客户端本地训练数据的样本数量不同,导致每一训练轮次每个客户端训练时间不同,这一现象称为数据量异构性;③在分布式机器学习中,训练数据的类别和特征均匀分布在所有客户端中,即训练数据是独立同分布的。然而,在联邦学习中,由于数据仅存储于本地,数据的类别和特征取决于数据

的所有者,从而导致数据分布不一致,即数据是非独立同分布的,这一现象称为非独立同分布数据异构性。其中,数据量异构性和非独立同分布数据异构性可统称为数据异构性。实验表明^[6],这3种异构性会显著影响模型训练时间和准确率。已有研究^[2,6,7]通过客户端选择方法缓解资源异构性或数据异构性,但现有研究没有均衡考虑两种异构性的影响,部分客户端的代表性数据没有参与训练过程,使全局模型的收敛存在较大的不稳定性,从而导致模型偏置、全局模型的精度较差^[8,9]。

针对上述不足,本文提出了一种资源高效的聚类协同联邦学习客户端选择方法。首先,根据每个客户端不同的训练速度,将所有客户端分为快、中、慢3组;其次,在每组中,基于每个客户端模型的余弦相似性计算相似性矩阵,并通过相似性矩阵采用层次聚类方法将客户端聚类,使每类客户端有相近的数据分布;最后,以每组客户端的平均准确率作为间接度量推断数据的非独立同分布异构信息,动态调整每轮训练的客户端选择,可同时考虑资源异构性和数据异构性,使其对训练时间和准确率的影响降到最低。

1 相关工作

过去的研究表明,客户端选择会对模型的训练时间和准确率产生一定的影响。

最早提出客户端选择方案是在FedAvg中^[2],为了减少通信次数,服务器可以选择参与每次迭代的客户端子集,这种策略被称为客户端选择,能有效降低通信量。FedAvg提出在每次迭代中随机选择参与训练的客户端子集,其他客户端的模型更新由当前的全局模型代替。然而,当客户端数据呈非独立同分布时,这种方案会受到客户端偏置的影响^[8],最终得到的全局模型不同于每个客户端的确定性聚合。为了解决这一问题,Li等^[10]提出了无偏采样方案FedProx,FedProx提出从多项式分布中选择参与训练的客户端子集,且根据每个客户端的相对数据样本大小进行选择。虽然FedProx采用无偏采样,但仍然可能导致客户端选择结果差异较大,部分客户端的代表性数据未参与训练过程。为了增强模型聚合中每个客户端的代表性,Fraboni等^[9]提出了聚类抽样方法

对客户端进行选择,该方法能够提高客户端的代表性,减少不同客户端随机聚合时的权重方差,使聚合模型获得更平滑、更快速的收敛。Shu等^[11]提出了一种基于自适应采样的方法FLAS,通过自步学习方法自适应调整阈值,以过滤每个客户端的训练数据,并选择合适的客户端参与每轮训练。

在资源异构性方面,Nishio等^[7]提出了采用背包约束的贪婪算法FedCS,解决异构客户端选择的最大化问题,FedCS通过评估客户端的资源选择合适的客户端参与训练,在指定的截至时间内,通过最大化所选客户端的数量来聚合尽可能多的客户端更新。该方法可以过滤掉计算资源差的客户端,加快模型的训练速度,但可能会导致具有丰富资源的客户端被高频选中,而具有匮乏资源的客户端始终不被选择,没有参与模型更新的机会,这些客户端可能有代表性数据一直没有参与训练,从而影响最终全局模型的效果。Bonawitz等^[12]提出了一种处理慢速客户端的简单方法,即服务器选择目标数量130%的客户端参与初始训练,在训练过程中丢弃慢速客户端。但是这种简单删除慢速客户端的方法可能会将这些慢速客户端的数据分布排除在最终全局模型之外。

联邦学习的上述客户端选择方法虽然可以分别解决资源异构性或数据异构性问题,但是均未同时解决两种异构性问题,并且没有考虑客户端代表性数据的选择问题。

2 准备知识

2.1 联邦学习

联邦学习的系统模型如图1所示,采用典型的客户端-服务器架构,由1个中央服务器和多个分布式客户端组成。其中,中央服务器协调整个联邦学习的训练过程,并对每轮训练参与的客户端进行选择;客户端是联邦学习训练的参与者,每个客户端都拥有自己的本地数据,通过与服务器通信,采用本地数据训练模型。

如算法1所示,联邦学习以迭代方式执行,通过多轮全局训练优化模型,所有客户端共享得到的最终全局模型。联邦学习要求1个中央服务器和 m 个随机客户端($m = \lceil N \times C \rceil$,其中 N 为所有客户端的数量, C 为每轮训练中选择客户



图1 联邦学习

Fig. 1 Federated learning

端的比例, $\lceil \cdot \rceil$ 表示向上取整运算)传递自己将要训练的全局模型参数(包括参数分发、更新和上传步骤)。该算法要求被选择的客户端使用它们的本地数据计算模型更新(对应更新和上传步骤),同时要求服务器聚合多个客户端的更新,从而获得更好的全局模型(对应聚合步骤)。该算法可以确保各客户端的数据不出本地,从而保证本地数据的隐私安全,其唯一的技术要求是客户端需要拥有一定的计算资源,因为更新和上传步骤通过模型的正向传播和反向传播的多次迭代实现。

算法1 联邦学习

- N 为客户端总数, $C \in (0, 1]$ 为每轮训练中选择客户端参与训练的比例, $m = \lceil N \times C \rceil$ 为选择客户端的数量。
1. 初始化:服务器随机初始化一个全局模型,或者使用公共数据预训练一个全局模型。
 2. 客户端选择:服务器随机选择 m 个客户端。
 3. 分发:服务器将全局模型参数分发给选择的客户端。
 4. 更新和上传:每个被选择的客户端使用自己的本地数据更新全局模型,并将更新后的模型参数上传到服务器。
 5. 聚合:服务器对更新后的参数进行平均计算,并用平均后的模型替换全局模型。
 6. 重复步骤2~步骤5,直到达到预定训练轮数或全局模型收敛。

2.2 余弦相似性

相似性度量用于计算不同个体间的相似程度,相似性度量值越大说明个体间差异越小,相似性度量值越小说明个体间差异越大。余弦相似性,又称余弦距离,是通过计算向量空间中两个向量夹角的余弦值,衡量两个个体间相似性大小的度量。给定两个属性向量 A 和 B ,其余弦相似性计算公式为:

$$\text{similarity}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (1)$$

式中: A_i 、 B_i 分别为向量 A 和向量 B 的第 i 个分量。余弦相似性的取值范围为 $[-1, 1]$, 余弦值越接近 1, 夹角越趋近于 0° , 说明两个向量越相似; 余弦值越接近 0, 夹角越趋近于 90° , 说明两个向量完全不相关。

余弦相似性可以应用于聚类分析中, 用于度量样本间的相似性, 从而将相似的样本聚集在一类中。

2.3 Ward 层次聚类算法

层次聚类是聚类算法中的一种, 通过计算不同类别数据点间的相似性创建一棵有层次的嵌套聚类树。由于得到的是一棵聚类树, 层次聚类不需要指定具体的类别数目, 可以在任意层次得到指定数目的类别, 便于直观确定类的划分, 得到较理想的分类。

层次聚类方法分为自下而上的凝聚方法和自顶向下的分裂方法, 本文采用凝聚方法。凝聚方法先将所有的样本点看作单独的类别, 然后计算两个不同样本点间的相似性, 将相似性最高的两个样本点合并为同一类别, 重复执行这一过程, 直到生成一棵聚类树。

Ward 聚类方法是一种基于方差(类内距离)的层次聚类方法, 由 Ward^[13] 于 1963 年提出, 其核心思想是每次选择方差增量最小的两个类别进行合并。凝聚方法最初将所有的样本点看作单独的类别, 此时每个类的类内距离(即方差)均为 0, 当类开始合并时, 类内的方差会逐渐增大。Ward 聚类方法优先合并方差变化量最小的两个类别, 以此保证整体聚类的方差最小。

3 本文方法

本节介绍本文提出的资源高效的聚类协同联邦学习客户端选择方法, 其结构如图 2 所示。首先, 根据各客户端的计算能力将其分为 3 组; 其次, 在每组中进行聚类; 最后, 以每组客户端的平均准确率作为间接度量, 选择平均准确率最低的一组, 在该组内的每一类中分别选择一个客户端

参加训练。在第 4 节中证明了, 该方法可以有效处理联邦学习中具有资源异构性和数据异构性的客户端。

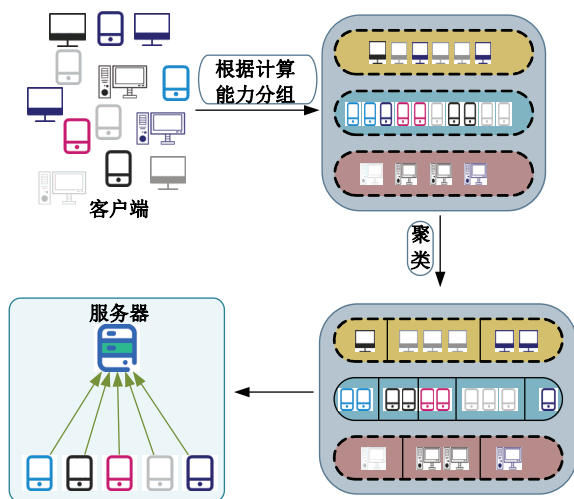


图 2 本文方法结构

Fig. 2 Structure of proposed method

3.1 客户端分组

在联邦学习中, 各客户端的计算能力、通信速度及样本数量各不相同, 存在资源异构性和数据量异构性的问题, 导致每轮训练中各客户端的本地训练时间不同, 因此每轮训练的训练时间会受到最慢速客户端的影响。为了缓解资源异构性和数据量异构性的影响, 通过算法 2 在开始联邦学习任务前将所有客户端分为快、中、慢 3 组, 每次从同一组中选择训练时间近似的客户端。

算法 2 客户端分组

输入: N 个客户端及其本地数据 D_i

输出: 客户端分组结果 $\{A\}$ 及每个客户端训练轮数后的

模型 $w_{\text{similarity}}^i$

服务器端:

1. 初始化模型 w_{resource}
2. 向全部客户端发送 w_{resource} 及训练轮数 r_{sync}
3. for 训练轮数 $r = 1, 2, \dots, r_{\text{sync}}$ do
4. for 客户端 $i \in N$ 并行 do
5. 客户端进行本地训练并返回训练时间 AT_i^r
6. if $r = r_{\text{sync}}$ then
7. 客户端返回训练得到的本地模型 $w_{\text{similarity}}^i$
8. end if
9. if $AT_i^r \leq T_{\text{max}}$ then
10. $T_i \leftarrow AT_i^r + T_i^{r-1}$
11. else

```

12.  $T_i \leftarrow T_{\max} + T_i^{-1}$ 
13. end if
14. end for
15. end for
16. for  $i = 1, 2, \dots, N$  do
17. if  $T_i = T_{\max} \times r_{\text{sync}}$  then
18. 客户端  $i$  不再参与训练
19. end if
20. end for
21. 将  $T_i$  由小到大排序,按顺序分为  $A_{\text{fast}}$ 、 $A_{\text{middle}}$ 、 $A_{\text{slow}}$  3
    组,每组  $N/3$  个客户端,并且保存各客户端的模型  $w_{\text{similarity}}^i$ 

```

首先,在客户端分组开始前,服务器将全部客户端训练时间初始化为 $T_i = 0$ 。其次,服务器向全部客户端分配客户端分组任务,根据算法 2 将初始化模型 w_{resource} 和训练轮数 r_{sync} 广播给所有客户端。所有客户端在其训练轮数内使用本地数据进行训练,在完成每轮训练后都将训练时间 AT_i^r 返回给服务器。若客户端在训练时间阈值 $T_{\max}$ 内响应,则其训练时间 T_i 随着实际训练时间 AT_i^r 累积;若客户端超时,则按照阈值 $T_{\max}$ 递增。在完成全部轮数训练后,若客户端训练时间均以阈值 $T_{\max}$ 递增,则认为该客户端处于离线状态,将其从其他计算中排除,同时各客户端将训练得到的模型 $w_{\text{similarity}}^i$ 返回给服务器,在算法 3 中将使用 $w_{\text{similarity}}^i$ 计算相似性。将剩余客户端的训练时间从小到大排序,按照各客户端的训练速度将其分为 A_{fast} 、 A_{middle} 、 A_{slow} 3 组,该分组将在接下来的算法中使用。

3.2 基于相似性聚类

在联邦学习中,训练数据的类别和特征取决于各客户端,这会导致数据分布不一致,即具有非独立同分布的数据异构性。若对客户端进行随机选择,则会导致部分具有代表性数据的客户端始终未被选择,或者每轮训练仅选择同种数据分布的客户端,这会使全局模型的收敛存在很大的不稳定性。为了解决数据异构性,增加具有代表性数据分布客户端的选择次数,减少客户端聚合权重的方差,从而获得更快速、更平滑的收敛,本文提出了基于模型相似性的聚类方法,每轮训练从每一类中分别选择客户端参与训练。

本文认为,客户端经过相应轮数训练得到的模型可在一定程度上反映其本地数据的类别和特征。模型相似性较高的客户端,其本地数据有类似分布

的概率。因此,通过模型相似性对客户端进行聚类,每一轮次从不同聚类中对客户端进行选择,以期可以包含不同种类的数据分布,减少客户端聚合权重的方差,从而获得更快速、更平滑的收敛。

算法 3 相似性聚类

N 为客户端总数, $C \in (0, 1]$ 为每轮训练中选择客户端参与训练的比例, $m = \lceil N \times C \rceil$ 为选择客户端的数量。

输入: 各客户端训练得到的模型 $\{w_{\text{similarity}}^i\}_{i=1}^N$ 、Ward 聚类算法、余弦相似性计算函数 $s(\text{式}(1))$ 、聚类数 m

输出: 客户端聚类结果 $\{B\}$

服务器端:

1. 计算每对客户端模型间的相似性 $\rho_{i,j} \leftarrow s(w_{\text{similarity}}^i, w_{\text{similarity}}^j)$, 构建 $N \times N$ 的相似性矩阵 ρ
2. 根据相似性矩阵 ρ , 使用 Ward 层次聚类算法得到嵌套聚类树 P
3. 截取聚类树 P , 将全部客户端分为 m 类, 分别为 $B_1, B_2, \dots, B_m$

研究表明,使用各客户端通过算法 2 训练得到的本地模型 $w_{\text{similarity}}^i$, 在给定的迭代过程中比较各客户端的模型相似性,是计算联邦学习参与方之间相似性的有效方法^[14]。在算法 3 中,首先,通过余弦相似性函数计算两个客户端间的模型相似性,得到一个 $N \times N$ 的相似性矩阵 ρ ;其次,根据相似性矩阵 ρ 进行 Ward 层次聚类,得到一棵聚类树 P ;最后,根据所需要的聚类数对 P 进行截取,得到客户端的分类 $\{B\}$ 。在计算相似性矩阵的过程中可能会引入大量计算开销,因此可以选择利用模型的部分参数(如卷积神经网络的第一个卷积层和最后一个全连接层的参数)进行模型相似性计算。

3.3 本文方法的提出

首先,利用算法 2 将全部客户端按照训练情况分为 3 组。其次,在每一组中利用算法 3 对客户端进行聚类,每轮训练在选择参与客户端时,需要先确定选择哪一组客户端,再对组内的每类客户端进行选择。最后,采用每层客户端的平均准确率作为间接度量,以此推测客户端非独立同分布数据的情况。为了防止对同一组客户端的过度选择,从而造成全局模型对该组客户端数据过拟合,导致引入训练偏差,本文设置阈值控制每组客户端的选择次数。同时,在各分组内每类中客户端的选择还考虑了各客户端参与训练的次数,期望

让每类中的所有客户端都可以参与训练,保证客户端选择的公平性。

在算法 4 中,首先,通过前文提到的算法 2 将全部客户端分为 A_{fast} 、 A_{middle} 、 A_{slow} 3 组,即按照客户

算法 4 资源高效的聚类协同联邦学习客户端选择方法

m 为每轮选择的客户端数量, R 为全局训练轮数, $Time$ 为被选择次数, Acc 为准确率。

输入: N 个客户端及其本地数据 D_i

输出: 客户端集合 $\{Client\}$

1. 通过算法 2 将 N 个客户端分为 A_{fast} 、 A_{middle} 、 A_{slow} 3 组
2. 通过算法 3 分别对 A_{fast} 、 A_{middle} 、 A_{slow} 3 组客户端进行层次聚类,将每组客户端分为 m 类,得到 $\{B_{fast}^i\}_{i=1}^m$ 、 $\{B_{middle}^i\}_{i=1}^m$ 、 $\{B_{slow}^i\}_{i=1}^m$
3. for $r = 1, 2, \dots, R$ do
4. if $r == 1$ then
5. $selected \leftarrow$ (从 $fast$ 、 $middle$ 、 $slow$ 中随机选择一组客户端)
6. $Time_{selected} \leftarrow Time_{selected} - 1$
7. end if
8. if $r > 1$ then
9. while True do
10. $selected \leftarrow$ (从可以选择的客户端分组中选择 Acc^{r-1} 最小的组)
11. if $Time_{selected} > 0$ then
12. $Time_{selected} \leftarrow Time_{selected} - 1$
13. break
14. else
15. 该 $selected$ 组不再参与训练
16. continue
17. end if
18. end while
19. end if
20. for $k = 1, 2, \dots, m$ do
21. 从 $B_{selected}^k$ 类中选择 $Time_{selected}^i$ 最小的客户端 i , 将 i 加入集合 $\{Client\}$
22. $Time_{selected}^i \leftarrow Time_{selected}^i + 1$
23. end for
24. for 客户端 $i \in \{Client\}$ 并行 do
25. 进行本地训练并上传本地模型
26. end for
27. 服务器聚合各客户端本地模型,将新的全局模型分发给全部客户端,客户端使用测试集计算模型准确率 Acc_r^i ,服务器计算每组客户端的平均准确率 Acc^{r-1}
28. end for

端拥有的计算资源分组,同一组客户端训练时间相近,从而缓解资源异构性问题。其次,利用算法 3 在每组中对客户端进行聚类,将数据分布近似的客户端聚为同一类,希望每轮选择的客户端都可以包含不同的数据分布,并且希望提高每个客户端的代表性,使具有独特数据分布的客户端也可以参与训练,从而获得更平滑、更快速的收敛。为了进一步保证客户端选择的公平性,为每个客户端设置变量 $Time_{selected}^i$ 时,记录该客户端被选择的次数,在从每类中选择客户端时,先按选择次数对客户端排序,把每类中被选次数最少的客户端加入训练集合中,由于是从同组但不同聚类中选择,因此也不会影响总体的训练速度,但是可以使同一个聚类中每个客户端有同样的次数加入全局训练,减少因频繁选择相同客户端造成训练偏差的可能。同时,考虑到频繁选择同一组客户端可能导致最终模型出现偏差,以及训练速度问题,希望更多选择训练速度较快的客户端组。因此,为每一组客户端设置不同的阈值 $Time_{selected}$,代表每組被选次数上限(根据 3 组客户端训练速度从快到慢,其阈值从大到小,即训练速度快的组可以适当增加选择次数,最慢的组可以适当减少选择次数),若超出阈值则该组客户端不再参与训练,在保证整体训练速度的情况下,也防止过度选择同组客户端出现训练偏差。最后,以每組客户端的平均准确率作为间接度量,决定每轮训练的客户端组选择。若某一组客户端平均准确率较低,则认为该分组在以前的轮次中训练较少,需要在后续训练中作出更大贡献,即应有更大的可能被选中参与训练,因此在每轮训练中选择平均准确率最低的组参加训练。

4 实验验证

对本文方法与 FedAvg 算法进行比较, FedAvg 算法在每轮训练中随机选择固定数量的客户端参与训练,并给出了在图像分类任务上的一系列实验结果。在两个不同的数据集 MNIST^[15] 和 CIFAR10^[16] 上,分别使用两种不同模型对上述两种方法进行测试。对于 MNIST 数据集,使用有 50 个节点隐藏层的全连接网络;对于 CIFAR10 数据集,使用 CNN 模型,该模型具有 3 层 3×3 的卷积核,其中第一层为 32 通道,第二、三层为 64 通道,前两个卷积层都使用 ReLU 函数激活,并跟随一个 2×2 的

最大池化层,在每个最大池化层后添加一个 0.2 的 Dropout,最后一个卷积层为具有 64 个单元并使用 ReLu 函数激活的全连接层。

此外,本实验不仅验证了本文方法在缩短训练时间、减少通信轮数、提升收敛速度和目标测试集准确率上的积极影响,还研究了每轮次本地训练轮数对收敛速度和模型精度的影响。详细的实验设置和实验结果如表 1 所示。

表 1 数据集统计表

Table 1 Dataset statistics table

数据集	MNIST	CIFAR10
训练集大小	60 000	50 000
测试集大小	10 000	10 000
特征数量	28×28	3×28×28
分类数量	10	10

4.1 实验设置

本实验采用两个现实中公开的分类任务数据集,如表 1 所示。一个是 MNIST 数据集,它是一个 0~9 的手写数字数据集,共由 70 000 张 28×28 像素的手写数字图片组成,其中包含 60 000 张训练样本和 10 000 张测试样本;另一个是 CIFAR10 数据集,它是一个有 10 类物体的彩色图像数据集,共有 60 000 张 32×32 像素的彩色图像,其中 50 000 张图像用于训练,10 000 张用于验证。训练集被划分给本地客户端,全局模型的测试由服务器使用全局测试集进行。本文采用两种方式将训练集划分给客户端:一种是 IID 设置,即所有客户端的数据集类别是统一的;另一种是 Non-IID 设置,即不同客户端之间的数据集存在不同的数据分布。本文使用两种方法划分每个客户端的非独立同分布数据集:第一种为与 FedAvg^[2]一致的病态非独立同分布划分方式,根据标签对数据集分类,每个客户端从不同类别获得数据,通过调整客户端获得的样本类别数量调整数据异构性;第二种为使用 Dirichlet 分布对数据集进行分割($\text{Dir}(\alpha)$),为每个客户端分配跨类别的数据分区,通过调整参数 α 调整数据异构性, α 越小数据异构性越强,当 $\alpha=0$ 时,为每个客户端分配一个类别,当 $\alpha \rightarrow \infty$ 时,为每个客户端分配统一的类别划分。通过上述设置,可以模拟联邦学习的数据异构场景,以探究客户端选择方法对收敛速度和模型性能的影响。

4.2 资源异构性

本实验在 MNIST 和 CIFAR10 数据集上,研究本文方法对于客户端资源异构性的积极影响。在训

练中,将客户端的本地训练轮数参数记为 E ,经多次实验确定 $E=5$ 为各客户端最佳的本地训练轮数,为了便于分组,将训练任务分配给 $N=90$ 个客户端,每一轮次选择 $m=9$ 个客户端进行训练。此外,设置全局训练轮数 $R=300$,学习率 $\text{lr}=0.05$,从 0~1 均匀分布中生成一组随机延时,以模拟每个客户端的资源异构性。本实验用相同的参数在 IID 和 Non-IID 两种场景下评估本文方法和 FedAvg 算法在资源异构性方面的影响,其中,Non-IID 两种场景使用病态非独立同分布划分平衡数据集,即每个客户端随机从训练数据的 10 个类别中选择 2 个,且每个客户端样本大小相同。

表 2 显示了两种方法在不同数据集和不同场景下的训练模型准确率,结合图 3 发现可知:在相同的训练轮数下,本文方法在 IID 场景可以获得与 FedAvg 相同的准确率,而在 Non-IID 场景的所有测试集上准确率均优于 FedAvg。通过观察收敛曲线可以看出,由于本文方法在同一资源组内根据各客户端数据分布进行更细粒度的选择,因此其收敛曲线的波动更加舒缓,收敛速度明显优于 FedAvg,这意味着本文方法通过使用更少全局训练轮数可以获得相同甚至更好的结果。换言之,本文方法可以在保证或者提高模型精度的前提下,加快联邦学习训练过程的收敛速度。

表 2 训练模型准确率

Table 2 Training model accuracy %

数据集	场景	FedAvg 方法	本文方法
MNIST	IID	96.95	97.12
	Non-IID	91.44	94.41
CIFAR10	IID	69.39	70.12
	Non-IID	58.9	61.41

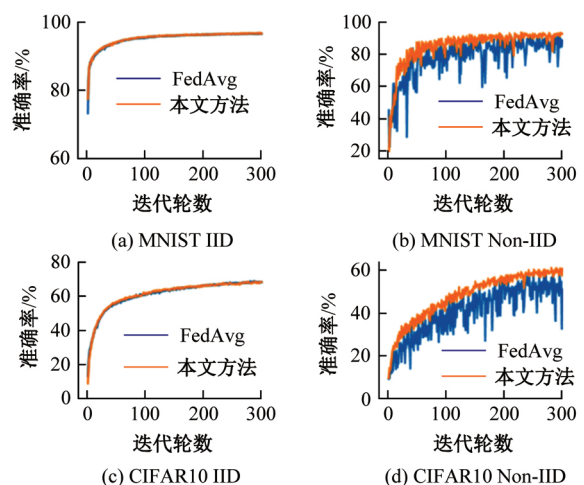


图 3 不同场景下模型准确率的效果

Fig.3 Effect of model accuracy in different scenarios

表3对比了两种方法的整体训练时间,由于本文方法根据资源异构性情况对客户端进行分组,每组设置不同的选择次数,以确保尽可能多地选择训练速度较快的分组,因此训练速度有很大提高,与FedAvg相比,在MNIST数据集的整体训练时间提高了20%以上,在CIFAR10数据集上提高了12%以上。

表4和表5显示了在达到相应固定准确率下,FedAvg和本文方法在训练轮数和训练时间上的

表3 整体训练时间

Table 3 Total training time

数据集	场景	FedAvg方法	本文方法
MNIST	IID	1670	1 302
	Non-IID	1 544	1 210
CIFAR10	IID	1 732	1 509
	Non-IID	1 894	1 631

差距,可以看出,本文方法可对客户端的资源异构性产生积极影响。

表4 在MNIST数据集上达到相应准确率所需训练时间和轮数

Table 4 Training time and number of rounds required to achieve corresponding accuracy in MNIST dataset

场景	方 法	准确率								总体/%
		85%		90%		93%		95%		
		轮数	时间/s	轮数	时间/s	轮数	时间/s	轮数	时间/s	
IID	FedAvg	3	17.70	12	68.19	31	176.90	66	376.60	96.95
	本文	3	13.92	10	43.01	30	125.73	62	270.03	97.12
Non-IID	FedAvg	83	494.94	212	1 186.58	—	—	—	—	91.44
	本文	35	222.78	79	532.86	195	921.58	—	—	94.41

表5 在CIFAR10数据集上达到相应准确率所需训练时间和轮数

Table 5 Training time and number of rounds required to achieve corresponding accuracy in CIFAR10 dataset

场景	方 法	准确率								总体/%
		50%		55%		60%		65%		
		轮数	时间/s	轮数	时间/s	轮数	时间/s	轮数	时间/s	
IID	FedAvg	27	119.76	43	186.19	79	334.64	152	735.16	69.39
	本文	26	84.24	40	129.32	73	283.69	147	615.85	70.12
Non-IID	FedAvg	129	830.14	192	1 260.76	—	—	—	—	58.9
	本文	115	604.83	164	910.11	242	1 347.81	—	—	61.41

4.3 数据异构性

本实验研究了本文方法在数据非独立同分布情况下提高全局模型准确率的能力。在CIFAR10数据集上,使用Dirichlet分布($\text{Dir}(\alpha)$)对数据集进行不平衡划分,为每个客户端分配跨类别的数据分区,每个客户端具有不同的样本数。其中,参数 α 用于控制创建的数据集的异构性, α

越小数据集的异构性越强,文献[17]提供了通过此方法获得数据集的图形说明。本实验中选择 $\alpha \in \{0.1, 0.5, 1\}$,其他参数均与4.2节保持一致。

图4展示了本文方法在不同异构性数据集上的效果。由于本文方法每轮训练均从不同聚类中选择客户端,即选择不同数据分布的客户端,尽可能确保单轮选择中包含不同种类的异构数据,因

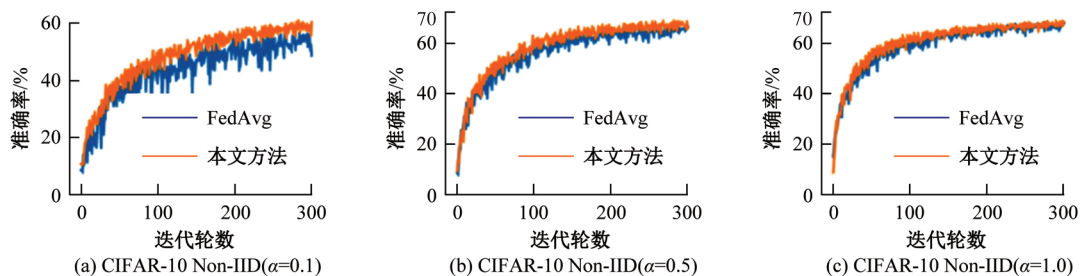


图4 不同数据异构性下模型准确率的效果

Fig. 4 Effect of model accuracy under different data heterogeneity

此,与FedAvg相比,本文方法的收敛过程更加平滑,方差更小,在非独立同分布数据集上训练模型的收敛速度和准确率均有提高,并且数据集的异构性越高,即 α 越小,本文方法的性能改善效果越显著。

5 结束语

本文提出了一种资源高效的聚类协同联邦学习客户端选择方法。为了同时解决资源异构性和数据异构性,该方法根据各客户端的计算速度将客户端分为3组,并且在每轮训练中以每组客户端的平均准确率作为间接度量,从同一组中选择客户端。每组内根据客户端的模型相似性,对客户端进行聚类,每轮训练在每组内依次选择不同聚类中的客户端。在真实数据集上评估了本文方法的性能,实验结果表明,本文方法可以有效地提高训练速度、缩短训练时间,获得更快速、更平滑的收敛,实现训练效率和全局模型准确率之间的良好平衡。

参考文献:

- [1] Tankard C. What the GDPR means for businesses[J]. Network Security, 2016, 2016(6): 5-8.
- [2] McMahan H B, Mcore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Proceeding of the 20th Tnternational Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, USA, 2017.
- [3] Abadi M, Chu A, Goodfellow L, et al. Deep learning with differential privacy[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 2016: 308-318.
- [4] Bonawitz K, Ivanow V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, USA, 2017: 1175-1191.
- [5] McMahan H B, Ramage D, Kunal T, et al. Learning differentially private recurrent language models[J/OL]. [2022-11-22]. <https://arxiv.org/abs/1710.06963>, 2017.
- [6] Chai Z, Ali A, Zawad S, et al. Tifl: A tier-based federated learning system[C]//Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing, Stockholm, Sweden, 2020: 125-136.
- [7] Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge [C]//IEEE International Conference on Communications (ICC), Piscataway, USA, 2019: 1-7.
- [8] Karimireddy S P, Kale S, Mohri M, et al. SCAFFOLD: Stochastic controlled averaging for federated learning[C] //International Conference on Machine Learning, Online, 2020: 5132-5143.
- [9] Fraboni Y, Vidal R, Kamenl L, et al. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning[C]//Preceeding of the 38th International Conference on Machine Learning, Online, 2021: 3407-3416.
- [10] Li T, Sahu A K, Zaheer M, et al. Federated optimization in heterogeneous networks[J]. Proceedings of Machine Learning and Systems, 2020, 2: 429-450.
- [11] Shu J G, Zhong W Z, Zhou Y, et al. FLAS: Computation and communication efficient federated learning via adaptive sampling[J]. IEEE Transactions on Network Science and Engineering, 2021, 9(4): 2003-2014.
- [12] Bonawitz K, Eichner H, Grieskamp W, et al. Towards federated learning at scale: System design[J]. Proceedings of Machine Learning and Systems, 2019, 1: 374-388.
- [13] Ward J H. Hierarchical grouping to optimize an objective function[J]. Journal of the American Statistical Association, 1963, 58(301): 236-244.
- [14] Sattler F, Müller K R, Samek W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(8): 3710-3722.
- [15] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [16] Krizhevsk, Hinton G. Learning multiple layers of features from tiny images[R]. Toronto: University of Toronto, 2009.
- [17] Hsu T M H, Qi H, Brown M. Measuring the effects of non-identical data distribution for federated visual classification[J/OL]. [2022-11-22]. <https://arxiv.org/abs/1909.06335>, 2019.