计算机工程
Computer Engineering
ISSN 1000-3428,CN 31-1289/TP

《计算机工程》网络首发论文

题目：一种基于本地模型质量的客户端选择方法
作者：温依霖，赵乃良，曾艳，韩猛，岳鲁鹏，张纪林
DOI：10.19678/j.issn.1000-3428.0065658
网络首发日期：2022-12-09
引用格式：温依霖，赵乃良，曾艳，韩猛，岳鲁鹏，张纪林．一种基于本地模型质量的客户端选择方法[J/OL]．计算机工程.
https://doi.org/10.19678/j.issn.1000-3428.0065658

网络首发时间：2022-12-09 10:23:14
网络首发地址：https://kns.cnki.net/kcms/detail//31.1289.TP.20221208.1139.002.html

计 算 机 工 程
Computer Engineering

一种基于本地模型质量的客户端选择方法

温依霖[1]，赵乃良[1]，曾艳[1,2,3]，韩猛[1]，岳鲁鹏[1]，张纪林（通信）[1,2,3]

(1. 杭州电子科技大学，计算机学院，杭州 310018；2.复杂系统建模与仿真教育部重点实验室，杭州 310018；
3.数据安全治理浙江省工程研究中心，杭州 310018)

摘　要：联邦学习是一种针对数据分布于多个客户端的环境下，客户端共同协作训练模型的分布式机器学习方法。理想情况下全部客户端均参与每轮训练，但实际应用中只随机选择了一部分客户端参与。随机选择的客户端通常不能全面的反映全局数据分布特征，会导致全局模型训练效率低、模型精度低等问题。针对上述问题，提出一种基于本地模型质量的客户端选择方法能提高模型训练的性能。首先分析影响模型精度和收敛速度的重要因素，提取可反映客户端模型质量的损失值和训练时间两个重要指标；其次，将本地损失值和训练时间融合建模，用于评估客户端模型质量；最后，基于客户端质量指导客户端选择，同时与随机选择策略进行一定比例的结合，以提高全局模型精度。通过选择拥有高质量的数据且计算性能较好的客户端参与训练，能够提升模型精度和收敛速度。针对FEMNIST、CIFAR-10、MNIST、CINIC-10 和EMNIST五种数据集，对比了三种基线算法FedAvg、FedProx、FedNova，实验结果表明，该方法可以将基线算法的收敛速度提高 10%左右，模型精度提高4%左右。

关键词：联邦学习；数据异构；损失值；训练时间；客户端选择

开放科学(资源服务)标志码(OSID)：

A Client Selection Strategy Based on Local Model Quality

WEN Yilin[1], ZHAO Nailiang[1], ZENG Yan[1,2,3], HAN Meng[1], YUE Lupeng[1], ZHANG Jilin[1,2,3]

(1.School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China;
2.Key Laboratory of Complex System Modeling and Simulation, Ministry of Education, Hangzhou 310018, China;
3.Zhejiang Engineering Research Center of Data Security Governance, Hangzhou 310018, China)

【Abstract】Federated Learning is a kind of distributed machine learning method which aims at training model by multiple clients in the environment where data is distributed among multiple clients. In the process of Federated learning and training, ideally, all clients need to participate in each round of training. However, due to the client state and network conditions, the existing methods only randomly select a part of clients to participate in each round of training in practical application, which makes the global model quality affected by the depth of the clients participating in the training. There are usually differences in data distribution among different clients. The randomly selected clients can not fully reflect the global data distribution characteristics, which will lead to low efficiency of global model training and low model accuracy. In view of the above problems, a client selection method based on data heterogeneity is proposed to improve the performance of model training. Firstly, the important factors affecting the accuracy and convergence speed of the model are analyzed, and two important indexes, the loss value and the training time, which can reflect the quality of the client model, are extracted; Secondly, the local loss value and the training time are fused to evaluate the quality of the client model; Finally, the client selection is guided based on the client quality. By selecting the client with high-quality data and good computing performance to participate in the training, the model accuracy and convergence speed are improved. Three baseline algorithms FedAvg,

基金项目：国家重点研发计划“面向城市公共服务的高效融合与动态认知技术和平台项目”(2019YFB2102101)；国家自然科学基金“‘学习驱动型’边缘智能系统性能优化关键技术研究”(62072146)；国家自然科学基金“移动边缘云中高效资源组合及存储分配的关键技术研究”(61972358)；浙江省重点研发计划项目“云-边协同的海洋渔业及港航物流智能服务关键技术、平台及应用示范”(2021C03187)

作者简介：温依霖、1998、女、硕士研究生、联邦学习；赵乃良、张纪林，教授；曾艳，副教授；韩猛、岳鲁鹏，博士研究生。**E-mail：**yz@hdu.edu.cn；jilin.zhang@hdu.edu.cn

FedProx and FedNova are compared for the five data sets of FEMNIST, CIFAR-10, MNIST, CINIC-10 and EMNIST. The experimental results show that this method can improve the convergence speed of the baseline algorithm by about 10% and the model accuracy by about 4%.

【**Key words**】federated learning；heterogeneous data；loss；training time；client selection

DOI:10.19678/j.issn.1000-3428.0065658

0 概述

目前的机器学习正在经历将数据收集和模型训练从中心云到边缘云的转变。智能手机、穿戴式智能设备和智能家电等现代移动设备和物联网设备每天产生大量的数据，如何有效地从这些数据中挖掘数据价值，同时保证隐私和安全性是一项挑战。数据从设备端传输到云端，一方面，传输过程中增加了安全风险，另一方面，设备端数据汇聚到云端共享处理，导致用户数据隐私泄露。为了降低数据安全风险和保证用户数据隐私，谷歌和苹果等大公司提出联邦学习[1-3]，实现了用户数据本地化的情况下，模型可以高效训练并挖掘数据价值。

联邦学习的多个客户端在服务器的协调下协同训练一个模型，同时保持训练数据的分散性。联邦学习体现了集中数据收集和最小化的原则，可以减轻传统的集中式机器学习带来的许多隐私问题、安全性风险和成本。目前，联邦学习已经在一些应用场景中成功使用，包括一些消费端设备以及制药、医学研究、金融、制造业等等。此外，也出现了大量联邦学习开发工具和平台，例如Tensorflow Federated[4], LEAF[5], PaddleFL[6]和PySyft[7]，进一步推动联邦学习的发展。

不同于传统机器学习的独立同分布（Independent and Identically Distribution，IID）数据环境，联邦学习面向的是一个复杂且异构的环境，不同客户端数据样本量以及数据的分布差异极大。这表明联邦学习系统中客户端本地数据之间存在Non-IID（Non-Independent and Identically Distribution，Non-IID）特性[8-10]。Non-IID数据造成参与方本地模型之间存在较大差异，阻碍全局模型的收敛。

近几年，许多学者提出了针对数据Non-IID的联邦学习优化方法，例如FedProx[11]算法，主要思想是对客户端本地损失函数添加正则项来限制模型参数差异，从而缓解Non-IID数据的影响。文献[12]指出数据Non-IID造成了基于各个客户端本地数据训练得到的模型聚合性能较差。因此联邦学习在Non-IID数据下的训练效果存在优化空间，合理的优化可以有效提高联邦学习模型的收敛速度以及模型精度。针对该问题，首先对Non-IID问题进行分析，然后在考虑计算和通信开销[13]的情况下，提出联邦优化方法。

一方面，分析客户端选择是联邦学习中重要的环节，客户端的选择意味着训练样本的选择，优秀的训练样本对全局模型具有较高的价值，能加快模型的收敛。然而随机选择的客户端通常不能较全面的反映全局数据分布特征，这正是导致全局模型训练效率低、模型精度低的原因。另一方面，有研究人员分析了在数据异构的前提下无差别的随机选择客户端会影响联邦学习的性能[14]。

针对联邦学习中Non-IID数据导致的模型精度低、收敛效率低等问题，本文提出一种基于本地模型质量的客户端选择方法ChFL（The method of choosing clients of Federated Learning），具体工作如下：

（1）分析Non-IID环境下影响模型精度和收敛速度的重要因素，提取出可以反映客户端模型质量的关键指标，包括客户端本地损失值和训练时间。

（2）根据指标构建模型质量评分模型，用于评估客户端本地模型质量。

（3）以客户端本地模型质量指导每轮同步训练时客户端的选择，为模型质量高的客户端赋予更高的选择概率，使得迭代训练时，优先选择该客户端，以保证模型准确度的前提下，提升收敛速度。

（4）为了验证本文方法的有效性，本文在FEMNIST、CIFAR-10、MNIST、CINIC-10、EMNIST数据集下，针对FedAvg、FedProx和FedNova算法进行了对比实验。实验结果表明在模型精度、收敛时间和通信开销上，本文方法均优于随机选择方法。

1 相关工作

联邦学习基于存储在数百万个远程客户端设备中的数据来训练全局模型。在训练期间，客户端必须定期与中央服务器通信。目前，联邦学习存在通信开销大、系统异构、数据异构、数据隐私等问题。现有的联邦学习方法很少考虑到数据异构的问题，大部分方法是为了缩小权重和压缩模型，以降低通信成本。

文献[15]提出的FedAvg算法是联邦学习的经典算法。FedAvg算法是一种基于平均模型更新的随机梯度下降方法。在每一轮次的训练过程中将计算开

销分担给多个本地客户端，利用随机梯度下降法更新全局模型，并通过实验验证了算法的收敛性。但是FedAvg算法并没有针对Non-IID问题进行优化，从而导致模型精度较低。

针对联邦学习中的数据异构问题，文献[16]提出Scaffold算法，分析出Non-IID数据使本地模型经过训练后进一步偏离全局模型。Scaffold算法通过使用一个控制变量来纠正客户端局部更新产生的偏离，从而提高模型收敛精度。但是传输控制变量也增加了大量的通信开销。文献[17]提出了一种知识蒸馏法来处理数据Non-IID问题。其中服务器学习一个轻量级生成器，以data-free的方式集成用户信息，然后广播给用户，使用学习到的知识作为"归纳偏置"来调节局部训练。文献[18]针对Non-IID数据导致的局部最优，提出了一种新的联邦学习个性化方法：聚类(多任务)联邦学习。在训练的过程中将参数相似的节点划分为同一个节点簇，同一个节点簇共享参数的变化量。文献[19]提出一种面向非独立同分布数据的联邦学习架构，将Non-IID的训练数据转换成多个独立同分布的数据子集，计算出从各类别数据中提取的特征在全局模型更新时的权重，从而缓解训练数据不均衡的负面影响。文献[20]提出基于本地数据特征的公平性联邦学习模型，以解决训练数据分布不均衡情况下产生的聚合模型对各个客户端模型不公平的问题。

部分研究者对联邦学习的参与者选择策略进行优化。文献[21]认为在每轮训练中只识别和传输信息量较大的客户端更新，能够减轻联邦学习的传输压力。通过选择每轮参与更新的客户端来减小训练过程中的通信开销，并且同时保证模型的准确度。文献[22]提出了一个客户端选择框架Oort，可以识别和挑选有价值的客户端进行训练和测试。文献[23]提出了一个混合式联邦学习框架HFL，该框架包括一个同步内核和一个异步更新器。既考虑正常设备的同步学习，也考虑落后设备的异步学习。文献[24]提出了一种队列系统中常用的Power-of-Choice策略，分析出偏向选择本地损失值较高的客户端会提高整个模型的收敛速度。

上述研究分析和验证了随机选择客户端会对联邦学习带来通信时间增长、模型准确度降低等问题，一般针对某一方面进行研究，忽略了一些有价值的指标。因此本文主要研究在不过多增加计算开销和通信开销的同时，能够提高Non-IID场景下的联邦学习收敛效率的优化方法。

2 问题描述

Non-IID数据会导致联邦学习模型精度降低，收敛效率低等问题。在本节通过对比联邦学习设置和集中式学习设置下学习任务的梯度更新过程来详细分析Non-IID问题对于联邦学习模型聚合过程的影响。以在样本空间 X 和标签空间 $Y=\{1,\dots,C\}$ 上的一个分类任务为例。数据点 $\{x,y\}$ 服从 $X\times Y$ 上的分布 h。函数 $f:X\to S$ 是一个 X 到对应概率 S 的映射函数，其中 $S=\left\{z\,|\,\sum_{i=1}^{C}z_i=1,z_i\geq 0,\forall i\in[C]\right\}$。将损失函数 $\ell(w)$ 定义为广泛应用的交叉熵损失函数，如式(1)所示：

$$\ell(w)=\sum_{i=1}^{C}h(y=i)\boldsymbol{E}_{x|y=i}\left[\log_e f_i(x,w)\right] \tag{1}$$

其中，参数 w 是神经网络的参数，f_i 表示神经网络预测数据样本为标签 i 的可能性。为了简单分析，忽略了泛化误差。因此集中式学习的学习任务可以通过式(2)所示：

$$\min_w \ell(w)=\min_w\left\{\sum_{i=1}^{C}h(y=i)\boldsymbol{E}_{x|y=i}\left[\log_e f_i(x,w)\right]\right\} \tag{2}$$

通常使用SGD来迭代求解机器学习任务中的模型权重 w。假设 $w_t^{(c)}$ 表示集中式学习设置下服务器第 t 轮更新的神经网络参数，η 表示学习率，∇_w 表示在第 t 轮更新的模型参数。因此，集中式学习设置下的SGD梯度更新公式如式(3)所示：

$$\begin{aligned}&w_t^{(c)}=w_{t-1}^{(c)}-\eta\nabla_w\ell(w_{t-1}^{(c)})=\\&w_{t-1}^{(c)}-\eta\sum_{i=1}^{C}h(y=i)\nabla_w\boldsymbol{E}_{x|y=i}\left[\log_e f_i(x,w_{t-1}^{(c)})\right]\end{aligned} \tag{3}$$

在联邦学习系统中假设总共有 K 个客户端，每一个客户端都在本地执行单独的SGD优化。n_k 表示客户端 k 的本地训练数据集，w_t^k 表示客户端 k 在第 t 轮进行SGD更新后的权重，如式(4)所示：

$$w_t^{(k)}=w_{t-1}^{(k)}-\eta\nabla_w\ell(w_{t-1}^{(k)}) \tag{4}$$

假设每执行 T 次SGD迭代后进行一次中央服务器的同步处理，$w_{mT}^{(f)}$ 表示第 m 次聚合后的联邦学习全局模型，则全局模型 $w_{mT}^{(f)}$ 的聚合式子如式(5)所示：

$$w_{mT}^{(f)}=\sum_{k=1}^{K}p_k w_{mT}^{(k)}$$

$$\sum_{k=1}^{K}p_k=1(p_k\geq 0) \tag{5}$$

其中，$w_{mT}^{(k)}$ 表示客户端 k 经过 mT 次迭代后的本地模型的参数，$p_k=\frac{n_k}{\sum_{i}^{N}n_i}$ 表示客户端 k 的模型聚合权重。下一次模型聚合时，服务器将向所有参与训练的客户端设备发送 $w_{mT}^{(f)}$。客户端将使用 $w_{mT}^{(f)}$ 初始化本地模型并在本地执行SGD迭代。

联邦学习中比较重要的一个步骤是模型聚合，每个客户端在本轮更新完局部模型后上传至服务器，通过加权平均的方式聚合生成一个全局的更新，形成新的全局模型。在FedAvg算法中作者提出使用客户端本地数据作为聚合的依据，如式(6)所示：

$$w_{mT}^{(f)}=\sum_{k=1}^{K}\frac{n_k}{\sum_{i=1}^{K}n_i}w_{mT}^{(k)} \tag{6}$$

其中，n_k 表示客户端 k 的本地数据量。

在上述分析的基础上，可以看出：1）对于集中式学习，通过对所有客户端的本地数据进行收集，建立了全局训练模型；2）为了实现联邦学习训练，在客户端本地训练不同的局部模型之后聚合为全局模型。因为不同的客户端本地数据通常是非独立同分布的，因此不同客户端的本地模型参数之间有显著差异。这种差异会降低加权平均聚类模型的总体收敛精度和收敛性能。因此对于联邦学习来说，获得与集中式学习相同的模型精度和收敛性是一个重要的问题。

由于通信方面的限制，参加联邦学习训练的设备无法确保所有客户端都参加每一轮训练过程，通常联邦学习在每一轮训练开始前先随机选择一个客户端子集作为参与者。因此选择客户端策略非常重要，因为它直接影响训练数据的质量。

联邦学习最常见的客户端选择策略是随机选择，所有客户端有同样的选中概率 $q_i=1/n$。最经典的联邦学习算法FedAvg就是采用随机选择策略，FedAvg算法在每一轮训练开始前，先以相同概率随机选择一定数量的客户端集，该客户端子集组织进行本地训练和更新模型等工作。随机选择策略使这些低质量客户端的选择概率与高质量客户端的选择概率相同，这可能会减慢整体训练进度，影响收敛效果。

3 基于本地模型质量的客户端选择方法

ChFL客户端选择策略基于本地模型质量对随机选择客户端方法进行了优化，使得高损失值和计算较快的客户端拥有更高的被选择概率。ChFL的流程图如图 1 所示，大体流程如下：

（1）将客户端损失值和训练时间融合建模，用于评估客户端本地模型质量。

（2）基于客户端本地模型质量指导客户端选择，质量较高的客户端赋予较高的选择概率。

（3）服务器使用ChFL客户端选择策略选择下一轮参与训练的客户端集合，在客户端训练完毕后将本地模型上传至服务器以聚合得到全局模型。

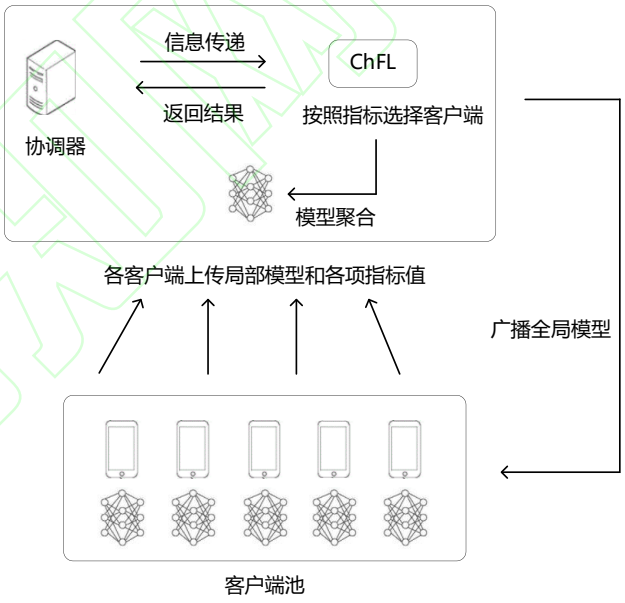


图 1　ChFL架构图

Fig.1　Architecture diagram of ChFL

3.1 关键指标

大多数联邦学习算法采用随机客户端选择策略，能确保所有客户端上的数据可以无偏的参与到训练中来，使得联邦学习模型拥有和集中式模型相似的收敛特性。但当训练数据高度Non-IID时，偏斜的客户端数据会使得样本的抽取变得不稳定，全局模型无法快速从局部训练中得到未知的知识，从而导致收敛速度变慢。为了加速全局模型的收敛速度，我们使用基于本地模型质量的客户端选择策略，该策略倾向于选择具有较高损失值和较快训练速度的

客户端参与训练。

深度学习的基本要义是要去挖掘数据中的信息，所以数据质量对于模型的训练至关重要。而且，训练数据的质量对于模型的精度是有着直接影响的。从样本的选择角度而言，客户端训练中产生的损失值可以反映全局模型对于该客户端上数据的预测能力，损失值越高说明预测能力越差。尤其在Non-IID数据下，高损失值的出现可能意味着该客户端的训练数据为少数类，在总体数据中占比不多，需要加强学习。为了加快收敛，服务器应该更多的选择这些客户端以得到缺乏的知识。

从收敛性能方面考虑，关于偏向高损失值能加快收敛的理论分析，可参见文献[24]的工作，更多的选择高损失值客户端可以使模型更快逼近收敛界。从收敛时间方面考虑，具有较快训练速度的客户端会减少整个模型的收敛时间。并且若将损失值和训练时间作为客户端选择依据，则不需要增加额外计算开销和通信开销。因此本文分析了这两个指标，对客户端的整体质量进行评估。

本文首先给出了几种训练指标的定义，为了更好的量化这些指标，本文针对每种指标都选取了合适的度量方式，将度量值正则化以使得多种度量值在同一值域下，正则化公式如式(7)所示：

$$Normalize(x)=\frac{x-\min_{val}}{\max_{val}-\min_{val}} \tag{7}$$

其中，$\max_{val}$表示该度量值的最大值，$\min_{val}$表示该度量值的最小值，x表示当前回合中该度量值。

首先分析损失值重要性并定义损失值重要性的公式。文献[25]证明了与样本的梯度范数成正比的分布抽样在最小化梯度方差方面是最优的，会导致随机梯度下降（SGD）的收敛速度加快。为此，本文采用重要性抽样方法[26]，旨在通过减少梯度估计的方差来提高SGD的收敛速度。

假设x_i，y_i，是训练集中的第i个输入输出对。$\varphi(\cdot;\theta)$是由向量θ参数化的机器学习模型，$L(\cdot;\cdot)$是训练过程中需要最小化的损失函数。模型训练的最终目标是为了找到能使损失函数最小的参数θ，如式(8)所示：

$$\theta^*=\arg\min_{\theta}\frac{1}{N}\sum_{i=1}^{N}L(\varphi(x_i;\theta),y_i) \tag{8}$$

其中N为训练集中的样本数量。使用学习率为η的随机梯度下降法，在两个连续的迭代t和$t+1$之间更新模型的参数，如式(9)所示：

$$\theta_{t+1}=\theta_t-\eta\alpha_i\nabla_{\theta_t}L(\varphi(x_i;\theta_t),y_i) \tag{9}$$

其中i是根据概率分布采样P的离散随机变量，概率为p_i并且α_i为样本权重。

在两个连续的迭代t和$t+1$之间，将SGD的收敛速度定义为参数向量θ和最佳参数向量θ^*之间的最小距离，如式(10)和(11)所示：

$$S=-\boldsymbol{E}_P\left[\left\|\theta_{t+1}-\theta^*\right\|_2^2-\left\|\theta_t-\theta^*\right\|_2^2\right] \tag{10}$$

$$\begin{aligned}&\boldsymbol{E}_P\left[\alpha_i\nabla_{\theta_t}L(\varphi(x_i;\theta_t),y_i)\right]\\&=\nabla_{\theta_t}\frac{1}{N}\sum_{i=1}^{N}L(\varphi(x_i;\theta_t),y_i\end{aligned} \tag{11}$$

设置$G_i=\alpha_i\nabla_{\theta_t}L(\varphi(x_i;\theta_t),y_i)$，并推导公式如式(12)所示：

$$\begin{aligned}S&=-\boldsymbol{E}_P\left[\left(\theta_{t+1}-\theta^*\right)^T\left(\theta_{t+1}-\theta^*\right)-\left(\theta_t-\theta^*\right)^T\left(\theta_t-\theta^*\right)\right]\\&=-\boldsymbol{E}_P\left[\theta_{t+1}^T\theta_{t+1}-2\theta_{t+1}\theta^*-\theta_t^T\theta_t+2\theta_t\theta^*\right]\\&=-\boldsymbol{E}_P\left[\left(\theta_t-\eta G_i\right)^T\left(\theta_t-\eta G_i\right)+2\eta G_i^T\theta^*-\theta_t^T\theta_t\right]\\&=-\boldsymbol{E}_P\left[-2\eta\left(\theta_t-\theta^*\right)G_i+\eta^2G_i^TG_i\right]\\&=2\eta\left(\theta_t-\theta^*\right)\boldsymbol{E}_P\left[G_i\right]-\eta^2\boldsymbol{E}_P\left[G_i\right]^T\boldsymbol{E}_P\left[G_i\right]-\eta^2Tr\left(\boldsymbol{V}_P\left[G_i\right]\right)\end{aligned} \tag{12}$$

从公式中可以观察到，通过从分布中采样的方法，可以使$Tr\left(\boldsymbol{V}_P\left[G_i\right]\right)$最小化，从而使收敛速度加快。重要性抽样的目标是使式(13)最小化：

$$Tr\left(\boldsymbol{V}\left[\nabla_\theta L(\varphi(x_i;\theta),y_i)\right]\right)=\boldsymbol{E}_P\left[\left\|\nabla_\theta L(\varphi(x_i;\theta),y_i)\right\|_2^2\right] \tag{13}$$

为了进行重要性抽样，根据概率p_i的分布P进行抽样，并使用每个样本的权重α_i，以获得梯度的无偏估计。因此，梯度的方差如式(14)所示：

$$\begin{aligned}&\boldsymbol{E}_P\left[\left\|\alpha_i\nabla_\theta L(\varphi(x_i;\theta),y_i)\right\|_2^2\right]\\&=\sum_{i=1}^{N}p_i\alpha_i^2\left\|\nabla_\theta L(\varphi(x_i;\theta),y_i)\right\|_2^2\\&=\sum_{i=1}^{N}\alpha_i\frac{1}{N}\left\|\nabla_\theta L(\varphi(x_i;\theta),y_i)\right\|_2^2\end{aligned} \tag{14}$$

假设神经网络是具有常数K的Lipschitz连续（当权重不是无穷大时成立的假设），推导出方差的以下上界：

$$\begin{aligned}&\boldsymbol{E}_P\left[\left\|\alpha_i\nabla_\theta L(\varphi(x_i;\theta),y_i)\right\|_2^2\right]\\&\le\sum_{i=1}^N\alpha_i\frac{1}{N}\left\|\nabla_\theta\varphi(x_i;\theta)\right\|_2^2\left\|\nabla_{\varphi(x_i;\theta)}L(\varphi(x_i;\theta),y_i)\right\|_2^2\\&\le K^2\sum_{i=1}^N\alpha_i\frac{1}{N}\left\|\nabla_{\varphi(x_i;\theta)}L(\varphi(x_i;\theta),y_i)\right\|_2^2\end{aligned}\tag{15}$$

因为样本是有限的，所以存在一个常数C，使得有如下式子：

$$\begin{aligned}&L(\varphi(x_i;\theta),y_i)+C\ge\left\|\nabla_{\varphi(x_i;\theta)}L(\varphi(x_i;\theta),y_i)\right\|\\&\forall i\in\{1,2,\ldots,N\}.\end{aligned}\tag{16}$$

可以看出以上上界要比均匀采样的上界更好一些，因为$L(\varphi(x_i;\theta),y_i)$和$\left\|\nabla_{\varphi(x_i;\theta)}L(\varphi(x_i;\theta),y_i)\right\|$是同时增长和收缩的，特别是当以下式子成立时：

$$\begin{aligned}&M=\max\left\|\nabla_{\varphi(x_i;\theta)}L(\varphi(x_i;\theta),y_i)\right\|,C<M\\&L(\varphi(x_i;\theta),y_i)+C-\left\|\nabla_{\varphi(x_i;\theta)}L(\varphi(x_i;\theta),y_i)\right\|\\&<M-\left\|\nabla_{\varphi(x_i;\theta)}L(\varphi(x_i;\theta),y_i)\right\|,\forall i.\end{aligned}\tag{17}$$

能够得出损失值可以用来创建比普通均匀采样更严格的梯度范数上限。与均匀采样相比，利用损失值采样是一种改进，能够提高SGD的收敛速度。

假设每个客户端k都有一个本地存储的训练样本集合B_k。样本重要性公式为$\left|B_k\right|\sqrt{\frac{1}{\left|B_k\right|}\sum_{x\in B_k}\left\|\nabla f(x)\right\|^2}$，其中$\left\|\nabla f(x)\right\|^2$是$B_k$中样本的梯度$\nabla f(x)$的$L2$梯度范数。这意味着在所有的样本中选择具有较大聚合梯度范数的样本。

使用梯度范数需要在后向传递期间计算二阶量，并且样本的梯度范数会随着模型的不断更新而产生变化。通过上述的理论分析也能得知利用损失值可以构造一个更好的梯度范数的上界。因此本文将客户端本地模型训练产生的损失值作为度量指标，从而代替样本的梯度范数。可以避免的是后向传递期间二阶量的计算成本和通信成本。

梯度是通过取损失值相对于当前模型权重的导数得出的，其中损失值衡量的是模型预测值与真实值之间的估计误差。通常来说，较大的梯度范数一般会导致较大的损失值。

关于使用损失值而不是梯度范数作为重要性度量的实验依据，可参考文献[25]。文献在训练期间对训练集前 20000 个样本进行了计算精确梯度范数和损失值的实验。梯度范数在每个小批次中进行标准化，以考虑权重范数的变化。并且绘制了按梯度范数排序的损失值图像，是一个单调递增的函数。能够得知梯度范数和损失值之间的相关性。在所有情况下，具有高梯度范数的样本也具有高损失。

因此定义客户端k的重要性公式如式(18)所示：

$$U(k)=\left|B_k\right|\sqrt{\frac{1}{\left|B_k\right|}\sum_{x\in B_k}Loss(x)^2}\tag{18}$$

其中$Loss(x)$为训练过程中样本x自动生成的训练损失值，收集开销可以忽略不计。因此把重点放在训练损失值较大的客户端对未来模型训练更为重要。

定义 1 客户端损失值：通过客户端的本地训练获取损失值，客户端k的本地损失值φ_L^k被定义为$U(x)$的归一化值，如式(19)所示：

$$\varphi_L^k=Normalize(U(k))\tag{19}$$

客户端在每轮训练中可能对不同数量的样本进行培训且数据分布不均匀，并导致不同的轮次时间，这会影响训练时间，也可能影响准确性。

参与训练的客户端之间的数据异构性可能会导致客户端不同的响应延迟（即客户端接收训练任务到返回结果的时间），这通常被称为掉队者问题（Straggler problem）。令客户端c_i的响应延迟为L_i，则全局训练模型的延迟为：

$$L_r=Max\left(L_1,L_2,L_3,L_4,\ldots L_{|C|}\right)\tag{20}$$

其中，L_r表示第r轮的延迟。可以看出，一轮全局训练的延迟是由客户端中的最大训练响应延迟，即最慢的客户端决定的。

如果将客户端划为t个层级，相同层级的客户端响应延迟相差不大。随机选择客户端从而形成一个由多个客户端层级组成的客户端群。假设总层数为m，t_m是最慢的层级。除最慢层级的客户端t_m之外的客户端层级中选择$|C|$个客户端的概率为：

$$P_r=\frac{\binom{|K|-|t_m|}{|C|}}{\binom{|K|}{|C|}}\tag{21}$$

据此，C中至少有一个客户端来自t_m的概率可以表述为：

$$P_{r_s}=1-P_r\tag{22}$$

对P_{r_s}进行拆分计算如式(23)所示：

$$P_{r_s}=1-\frac{\binom{|K|-|t_m|}{|C|}}{\binom{|K|}{|C|}}$$
$$=1-\frac{(|K|-|t_m|)\ldots(|K|-|t_m|-|C|+1)}{|K|\ldots(|K|-|C|+1)}$$
$$=1-\frac{|K|-|t_m|}{|K|}\cdots\frac{|K|-|t_m|-|C|+1}{|K|-|C|+1} \tag{23}$$

由于具有如下约束：

$$\frac{a-1}{b-1}<\frac{a}{b},while1<a<b \tag{24}$$

所以 P_{r_s} 满足：

$$P_{r_s}>1-\frac{|K|-|t_m|}{|K|}\cdots\frac{|K|-|t_m|}{|K|}$$
$$=1-\left(\frac{|K|-|t_m|}{|K|}\right) \tag{25}$$

在实际场景中，每一轮都会有大量的客户端被选中，这使得$|K|$非常大，作为 K 的子集，C 的大小也可以足够大。因为 $\frac{|K|-|t_m|}{|K|}<1$，可以得到 $\left(\frac{|K|-|t_m|}{|K|}\right)^{|C|}\approx 0$，这使得 $P_{r_s}\approx 1$。因此，每轮从最慢层级中至少选择一个客户端的概率是相当高的。因此，随机选择客户端的策略可能会导致全局模型训练性能缓慢。所以考虑将客户端的训练时间作为反映客户端质量的指标。对客户端训练时间的定义如下：

定义 2 客户端训练时间：客户端 k 接收到训练任务的时间为 T_{start}，客户端返回结果的时间为 T_{end}。客户端 k 的训练时间 φ_T^k 定义为 T_k 的归一化值，如式(27)所示：

$$T_k=T_{end}-T_{start} \tag{26}$$
$$\varphi_T^k=Normalize(T_k) \tag{27}$$

3.2 客户端模型质量评分建模

ChFL希望依据损失值和训练时间反映客户端模型质量，通过上一小节提出的两个指标对客户端局部模型进行评估，并将两种指标融合建模。

两种指标分别从两个维度对客户端的质量进行了评估，它们的评估重要性各有不同。由于高损失值的样本数据既能加快收敛速度又能反映全局模型对客户端数据的预测能力，所以二者重要性关系为损失值>训练时间。如何将两种指标合理的结合成一个数值以作为客户端模型质量评分的设置依据是接下来要解决的问题[27]。为此，本文结合两种指标以及它们的重要性定义客户端本地模型质量评分，具体的定义如下：

定义 3 客户端本地模型质量评分：本文考虑到指标之间的重要性，关联性以及变化性，为客户端 k 定义的模型质量评分 F_k 如式(28)所示：

$$F_k(\varphi_L^k,\varphi_T^k)=\lambda_1\cdot\varphi_L^k+\lambda_2\cdot\varphi_T^k \tag{28}$$
$$s.t.\lambda_1=1$$
$$\lambda_2=\lambda_1\cdot\varphi_L^k$$

其中，φ_L^k 表示客户端 k 本地损失值的归一化值，φ_T^k 表示客户端 k 训练时间的归一化值，λ_1,λ_2 是两个指标的权重。

客户端本地模型质量评分也可以扩展成有 N 个指标的情况，详细信息如定义 4 所示。

定义 4 客户端本地模型质量评分（N 个指标）：假设目前存在 N 个与客户端质量相关的指标，那么为客户端 k 定义的客户端质量评分 F_k 如式(29)所示：

$$F_k(\varphi_1^k,\varphi_2^k,\ldots,\varphi_N^k)=\sum_{i=1}^{N}\lambda_i\cdot\varphi_i^k,\varphi_i^k\in[0,1]$$
$$\lambda_i=\lambda_{i-1}\cdot\varphi_{i-1}^k,(\lambda_1=1) \tag{29}$$

其中 $\varphi_i^k(i\in[1,N])$ 表示与客户端模型质量相关的第 i 个指标，λ_i 表示第 i 个指标的权重，指标之间的重要性排序顺序为 $\varphi_1^k>\varphi_2^k>\ldots>\varphi_N^k$。

服务器无法计算在每个训练周期中所有客户端的损失值，因为它需要所有客户端都参与训练过程，这对于缺乏良好通信条件和硬件设施的移动设备来说无疑是困难的。因此，当服务器通过损失值和训练时间进行选择时，不能将当前全局模型中所有客户端设备的损失值和训练时间作为选择的基础，而是使用上一轮训练后的每个客户端设备的损失值和训练时间。虽然这样的选择策略有一定的陈旧性，但并不会影响选择策略的表现，因为客户端的损失

值会随着不断的训练而减少。一个客户端在完成本次训练后更新了一个较小的损失值，从而减小了以后选择的概率，也为其他没有被选择的客户端提供了更多被选中并参与训练的机会，防止训练过程陷入局部最优的状态，从而提高全局模型的收敛性能。

我们对本文提出的基于客户端本地模型质量的选择策略进行理论分析，具体分析选择策略对联邦学习全局模型收敛能力以及对收敛上限的影响。首先，我们对用于收敛性分析的假设和定义进行说明。

假设 1 局部目标函数 $F_1,\dots,F_k$ 都是 $L-$平滑，即 $F_k(v)\le F_k(w)+(v-w)^T\nabla F_k(w)+\frac{L}{2}\|v-w\|_2^2$。

假设 2 $F_1,\dots,F_k$ 都是 $\mu-$强凸，即 $F_k(v)\ge F_k(w)+(v-w)^T\nabla F_k(w)+\frac{\mu}{2}\|v-w\|_2^2$。

假设 3 对于从用户 k 的数据 B_k 均匀采样的 mini-batch ξ_k，所得到的随机梯度是无偏的，即 $\boldsymbol{E}[g_k(w_k,\xi_k)]=\nabla F_k(w_k)$。对于所有的 $k=1,\dots,K$，随机梯度的偏差是有界的：

$$\boldsymbol{E}\|g_k(w_k,\xi_k)-\nabla F_k(w_k)\|^2\le\sigma^2$$

假设 4 随机梯度的期望平方范数是一致有界的，即对于所有的 $k=1,\dots,K$，

$$\boldsymbol{E}\|g_k(w_k,\xi_k)\|^2\le G^2$$

为了体现局部与全局的数据分布差异，接下来我们定义局部-全局目标差这一概念。

定义 5 局部-全局目标差： 对于全局最优 $w^*=\arg\min_w F(w)$ 和局部最优 $w_k^*=\arg\min_w F_k(w)$，我们将局部-全局目标差定义为：

$$\Gamma\cong F^*-\sum_{k=1}^{K}p_kF_k^*=\sum_{k=1}^{K}p_k\left(F_k(w^*)-F_k(w_k^*)\right)\ge 0 \tag{30}$$

其中，p_k 指的是数据百分比。

Γ 是局部和全局目标函数的固有属性，它独立于客户端选择策略。Γ 越大，就意味着越高的统计异构性。如果 $\Gamma=0$，则意味着局部和全局最优值是一致的。接下来，我们定义名为选择偏斜的度量，它能够感知客户端选择策略对局部-全局目标差距的影响。

定义 6 选择偏斜： 对于任意 $k\in S(\pi,w)$，本文定义用以反映客户端选择策略 π 的偏斜程度的变量 ρ。

$$\rho(S(\pi,w),w')=\frac{\boldsymbol{E}_{S(\pi,w)}\left[\frac{1}{m}\sum_{k\in S(\pi,w)}\left(F_k(w')-F_k^*\right)\right]}{F(w')-\sum_{k=1}^{K}p_kF_k^*}\ge 0 \tag{31}$$

$\rho(S(\pi,w),w')$ 中的第一个 w 表示客户端集合 S 产生的全局模型参数向量，w' 是客户端训练过程中新产生的模型参数。

由于在选择策略 π 下，可能产生多种不同的客户端选择集合 S，故计算局部当前损失值与局部最优损失值的差值 $F_k(w')-F_k^*$ 时，需要计算该差值的期望 $E_{S(\pi,w)}[\cdot]$ 以考虑随机性的影响。$\rho(S(\pi,w),w')$ 是一个与模型参数 w 以及 w' 相关的变量，随模型参数变化而变化，在此定义两个独立于 w 以及 w' 的指标 $\underline{\rho}$ 和 $\overline{\rho}$ 如下：

$$\underline{\rho}=\min_{w,w'}\rho(S(\pi,w),w') \tag{32}$$

$$\overline{\rho}=\max_{w}\rho(S(\pi,w),w') \tag{33}$$

通过定义这两个度量以确定选择策略偏斜程度的上下界。其中，$w^*=\arg\min_w F(w)$。

对于无偏的选择策略来说，$\rho(S(\pi,w),w')$ 结果总是为 1，所有客户端具有相同的被选择几率使得该式子的分子分母相同。对于其他有偏的选择策略来说，$\rho(S(\pi,w),w')$ 的结果会有所不同。例如，给予具有更高损失值 $F_k(w)$ 的客户端 k 以更高的选中概率，$\rho(S(\pi,w),w')$ 中分子上期望的计算会因为客户端选择概率的变化而变化。在此情况下，由于高 $F_k(w)$ 值的客户端的选中概率被提高，$\rho(S(\pi,w),w')$ 结果将大于 1，上下界 $\underline{\rho}$ 和 $\overline{\rho}$ 也会因此提高。

假设学习率存在衰减的学习率 $\eta_t=\frac{1}{\mu(t+\gamma)},\gamma=\frac{4L}{\mu}$，结合假设 1-4 以及 Cauchy–Schwarz不等式，客户端选择策略下的损失误差满足下式：

$$\boldsymbol{E}\left[F\left(\overline{w}^{(T)}\right)\right]-F^*\le \frac{1}{(T+\gamma)}\left[\frac{4L\left(32\tau^2G^2+\sigma^2/m\right)}{3\mu^2\underline{\rho}}+\frac{8L^2\Gamma}{\mu^2}+\frac{L\gamma\left\|\overline{w}^{(0)}-w^*\right\|^2}{2}\right]+\frac{8L\Gamma}{3\mu}\left(\frac{\overline{\rho}}{\underline{\rho}}-1\right) \tag{34}$$

其中，γ 表示衰减系数，τ 表示迭代轮次，μ 是

凸系数，L 为光滑系数。

更大的 $\bar{\rho}$ 带来更快的收敛。一个收敛序列向其极限逼近的速度称为收敛速度，由第一项可得，更大的选择偏差 $\bar{\rho}$ 带来更快的收敛速度，这个收敛速度加成达到 $O\left(\frac{1}{(T+\gamma)\bar{\bar{\rho}}}\right)$。注意，由于我们通过取选择倾斜 $\rho(S(\pi,w),w')$ 的最小值来得到 $\bar{\rho}$，这是真实收敛速率的一个保守界。在实践中，由于选择倾斜 $\rho(S(\pi,w),w')$ 在训练过程中随当前的全局模型 w 和局部模型 w' 而变化，真实收敛率可以通过一个大于等于的 $\bar{\rho}$ 来提高。

由有偏选择策略引起的解偏差。第二项 $Q(\bar{\rho},\bar{\bar{\rho}})=\frac{8L\Gamma}{3\mu}\left(\frac{\bar{\bar{\rho}}}{\bar{\rho}}-1\right)$ 表示解偏差，这取决于选择策略。根据 $\bar{\rho}$ 和 $\bar{\bar{\rho}}$ 的定义可以得出，$\bar{\rho}$ 和 $\bar{\bar{\rho}}$ 的关系总是满足 $\bar{\rho}\geq\bar{\bar{\rho}}$，这意味着 $Q(\bar{\rho},\bar{\bar{\rho}})\geq 0$。对于无偏选择策略，满足 $\bar{\rho}=\bar{\bar{\rho}}=1$，$Q(\bar{\rho},\bar{\bar{\rho}})=0$，恢复了无偏选择策略的先前边界为 μ-强凸边界。对于 $\bar{\rho}>1$，虽然我们获得了 $O\left(\frac{1}{(T+\gamma)\bar{\bar{\rho}}}\right)$ 的收敛速度加成，但我们不能保证 $Q(\bar{\rho},\bar{\bar{\rho}})=0$。因此本文提出的有偏的客户端选择方法能提高全局模型的收敛效率。

在数据异构的情况下，偏斜的客户端数据会使样本的抽取变得不稳定，全局模型无法快速从局部训练中得到未知的知识，从而导致收敛速度变慢。典型的FedAvg算法采用均匀选择的客户端选择方式，低质量客户端有更大概率存在计算力低、网络延迟大的问题。而本文提出的利用客户端本地模型质量评分优化了客户端选择方法，在选择客户端时舍弃常规的无偏随机选择策略。该方法可以使模型更快逼近收敛界，使得收敛速度加快。并且客户端训练中产生的损失值可以反映全局模型对于该客户端上数据的预测能力，高损失值的出现可能意味着该客户端的训练数据为少数类，选择这些客户端能得到缺乏的知识，能够更好地加强学习。

3.3 基于客户端质量选择客户端

联邦学习的随机选择策略中所有客户端有同样的选中概率 $q_i=1/n$。随机选择策略使低质量客户端的选择概率与高质量客户端的选择概率相同，可能会影响全局模型收敛效果。因此针对这一问题本文作出了改进。

在计算了基于客户端损失值和训练时间的本地模型质量评分 F_k 后，服务器在客户端选择时利用本地模型质量评分为不同客户端赋予不同的选择概率，具体的概率计算公式如式(35)所示：

$$p_k=e^{\eta F_k} \tag{35}$$

对于高损失值客户端的选择倾向尽管可以给客户端带来收敛速度的提升，但也使得全局损失函数的最优值与理想最优值之间多出一个偏差项[28]。并且会对客户端有效信息的完整性造成一定程度的影响，过度的追求高损失值客户端带来的收敛速度加成可能会导致模型的精度损失。因此，本文不会根据客户端的损失值和计算能力来抽取所有的客户端。为了保证客户端有效信息的完整性，将利用客户端本地模型质量策略和随机选择策略进行一定比例的结合。

当服务器从所有 N 台客户端中选择 k 台客户端进行当前训练周期时，ChFL选择策略会通过超参数 α 来控制利用两个指标值进行选择的策略 p_k 和随机选择策略 p_{rand} 的比例。首先，通过 p_k 策略选择 αk 台客户端，加入集合 s_1；再通过 p_{rand} 策略抽取剩余的 $(1-\alpha)k$ 台客户端，加入集合 s_2。当 $\alpha=0$ 时，算法整体是随机选择策略，与FedAvg算法相同。客户端集合 $s=s_1+s_2$ 就是下一轮要参与训练的设备。

3.4 算法实现

在上述定义和分析的基础上，本文提出了一种改进的客户端选择策略ChFL，以提高数据Non-IID情况下联邦学习全局模型的收敛效率。结合算法 1，ChFL的具体流程描述如下：

（1）客户端接收到服务器广播的全局模型后进行本地训练，并在每次同步中计算出客户端损失值和训练时间的归一化值。

（2）客户端将局部模型参数、本地损失值和训练时间上传到服务器。服务器将这两个指标值作为选择客户端的依据，计算出客户端模型质量评分。

（3）全局模型聚合。将参与训练的客户端模型更新聚合为全局模型。

（4）服务器选择下一轮参与训练的客户端并下发全局模型。利用ChFL客户端选择策略，结合 p_k 和 p_{rand} 进行一定比例的策略分配。

以此往复，直至全局模型精度达到预设要求。

上述已经说明策略规则以及具体实现方式，接下来以参与方 k 为例，描述基于本地模型质量的客户端选择方法，具体如算法 1 所示。

算法 1 基于本地模型质量的客户端选择方法

输入 训练数据集 D，全局迭代次数 T，指标更新阈值 α，参与训练客户端集合 K_t，本地 $batch\ size\ B$，本地迭代次数 E，学习率 η

输出 全局模型

1.初始化全局模型 w_0。

2.在训练的每一轮次 $t=1,2,\ldots,T$，对每个客户端 $k \in K_t$，获取 $w_t^k, U(k), T_{end}, T_{start}$。

3.计算 $T_k, \varphi_L^k, \varphi_T^k$。

4.计算客户端本地模型质量评分 $F_k^{(t+1)}$。

5.上传 w_t^k， $U(k), T_{end}, T_{start}$ 到服务器。

6.全局模型聚合，更新本地模型 w_{t+1}^g。

7.根据 $F_k^{(t+1)}$ 选择下一轮参与训练的客户端。

4 实验设计与分析

在本节中设置大量对比实验去验证ChFL客户端选择策略的优越性。本文将三种常用的联邦学习算法FedAvg、FedProx和FedNova作为基线算法，将随机选择客户端策略和ChFL客户端选择策略进行对比。在FEMNIST、CIFAR-10、MNIST、CINIC-10、EMNIST这五个不同的图像数据集上进行了充分的实验。并且从模型精度、收敛时间和通信开销角度对实验结果进行对比分析，从而多维度验证了ChFL客户端选择策略优越性。

4.1 实验设置

4.1.1 实验环境

在本文中基于FedML[29]框架进行联邦学习算法的实验。该框架的目的是为研究者提供一个扩展性好、通用性强的代码框架，并实现了许多著名的联邦学习算法，以帮助研究人员进行对比实验。本文实验采用Python编程语言实现，Python版本为 3.7.2，采用Pytorch框架来构建深度学习模型，Pytorch版本为 1.7.0。本文实验使用的系统版本为Ubuntu 18.04.4 LTS，操作系统内核为 4.15.0-123-generic，其他系统的详细配置如表 1 所示。

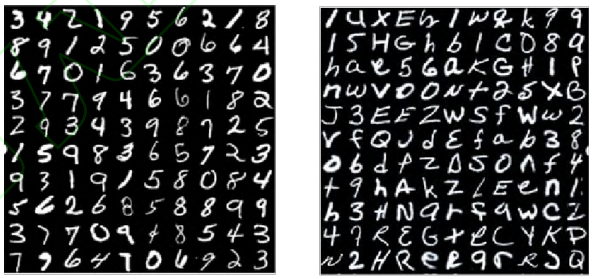
表 1 实验环境的软件及硬件配置

Table 1 Software and hardware configuration of the experimental environment

环境名称	配置信息
系统版本	Ubuntu 18.04.4 LTS
GPU	NV Tesla P100
Linux 内核版本	4.15.0-123-generic
显卡驱动版本	418.87.00
Cuda	10.1.243

4.1.2 数据集

本文采用五种常见的图像数据集来验证所提出方法的优越性，分别是 FEMNIST、CIFAR-10、MNIST、CINIC-10、EMNIST 数据集。MNIST 数据集是手写体识别数据集，其中包含了大量人类书写的数字图片。包含 60000 张手写体样本图片，其中 50000 张被划分为训练集，另外 10000 张则被划分为测试集。其数据集示例如图 2(a)所示。EMNIST 数据集是 MNIST 数据集的拓展，在同样的图片格式前提下，增多了图片的类别和种类。其数据集示例如图 2(b)所示。CIFAR-10 数据集是包含 60000 张 RGB 彩色图像的图像数据集，包含 10 个不同的类别，其中 50000 张图片数据为训练集，而另外 10000 张为测试集。



（a）MNIST数据集　　（b）EMNIST数据集

图 2 实验数据集样本示例

Fig.2 Sample of experimental data set

4.1.3 评价指标

本文算法实验是图像识别任务，都是多分类任务，对于分类任务一般可以通过计算准确率来对模型最终的结果进行度量。准确率是衡量分类模型的常用指标，其定义如式(36)所示：

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{36}$$

其中，TP 即True Positive，指被模型预测为正类的正样本；TN 即True Negative，指被模型预测为负类的负样本；FP 即False Positive，指被模型预测为正类的负样本；FN 即False Negative，指被模型预测为负类的正样本。该评估指标考虑到了所有的情

况，所以能够全面地反映出模型的训练情况。

在注重高准确率的同时，本算法同样注重模型的收敛性能，会根据不同的数据集设置收敛精度，并记录各算法收敛所需的轮数。

4.2 实验结果分析

在面对数据 Non-IID 情况下，ChFL 客户端选择策略通过对数据和模型的良好感知能为联邦学习训练全局模型带来一定的优化。在本节中共设置了三部分实验，第一部分是将决定 ChFL 客户端选择策略的两个指标分开进行实验，验证单指标单独工作的性能。第二部分是在三种联邦学习基线算法 FedAvg、FedProx、FedNova 的基础上将 ChFL 客户端选择策略和随机选择策略进行对比，验证采用 ChFL 客户端选择策略时全局模型的收敛速度和模型精度有所提升。第三部分是对 ChFL 客户端选择策略中超参数阈值的设置展开实验，以寻找最优的阈值设置。

4.2.1 关键指标损失值和训练时间分析

在五种数据集下对三种基线算法 FedAvg、FedProx、FedNova 进行单指标实验。单指标重要性实验是为了验证两种指标在单独工作时对客户端选择策略优化的效果。

ChFL-Time 表示根据客户端训练时间来改变选择概率，优先选择具有较好计算性能的客户端参与训练，从而加快模型收敛速度。ChFL-Loss 表示根据客户端本地损失值来改变选择概率，优先选择具有高质量数据的客户端，能够提高模型精度。ChFL 表示将客户端本地损失值和训练时间结合，共同作用改变客户端选择概率。

如图 3 所示，对 FedAvg 算法进行单指标实验。从实验结果能够分析：1）ChFL-Time 和 ChFL-Loss 刚开始都能快速提高模型的准确性，ChFL-Time 一味地选择训练能力优秀的客户端，模型收敛速度有所提高，但是这些客户端数据质量可能不高，最终对于模型准确率提高作用不大；2）ChFL-Loss 选择具有高质量数据的客户端，虽然收敛轮数比 ChFL-Time 更多，但是能较大的提高模型的准确率。

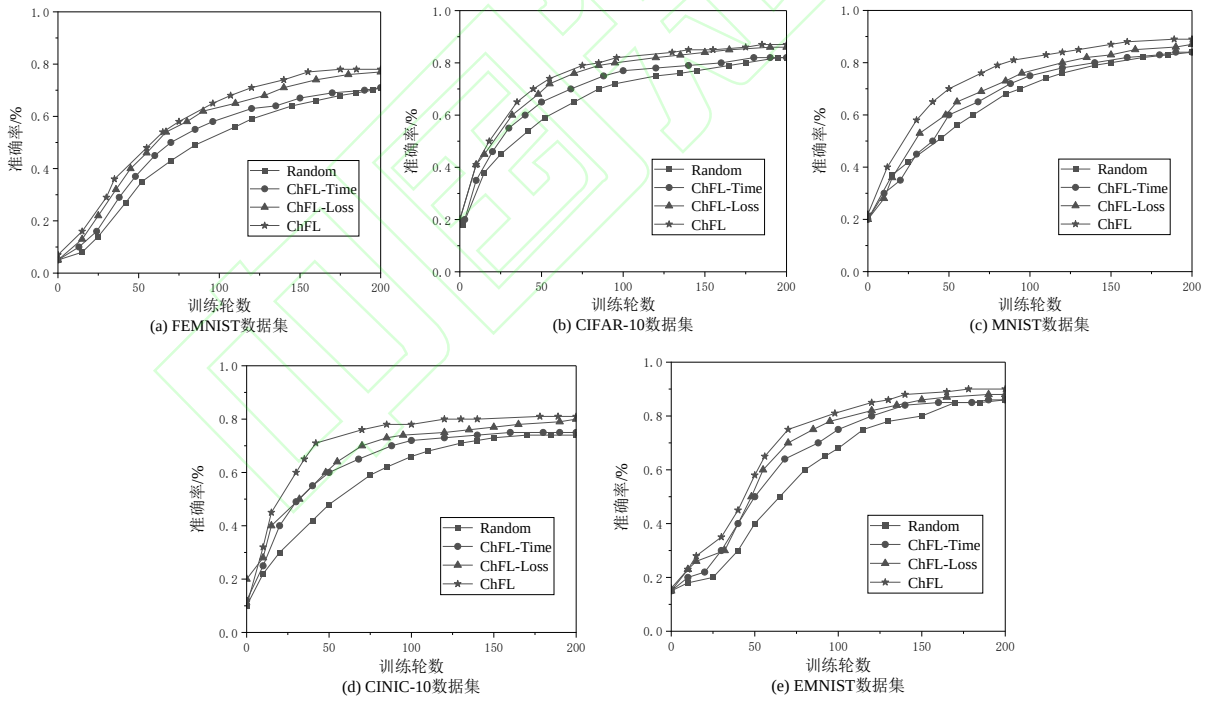


图 3 单指标的FedAvg算法精度对比实验图

Fig.3 Experimental Chart of Single Index FedAvg Algorithm Accuracy Comparison

如表 2 所示，ChFL-Time、ChFL-Loss 和 ChFL 策略在模型精度和收敛时间指标上都优于 Random 策略。其中 1) ChFL-Time 收敛速度能提高 9%左右，尤其在 CINIC-10 数据集上能提高 13.3%；2）ChFL-Loss 在五种数据集下收敛速度能提高 3%左右，模型精度能提高 4%左右，在 FEMNIST 和

CINIC-10 数据集下精度能提高 6.2%和 5.6%；3) ChFL 策略在五种数据集下收敛速度能提高 12%左右，在 CINIC-10 数据集下收敛速度提高 14.8%，模型精度能提高 5%左右，在 FEMNIST 数据集下模型精度能提高 7.1%。

表 2 单指标的FedAvg算法精度对比实验表

Table 2 Single index FedAvg algorithm precision comparison experiment table

指标	FEMNIST		CIFAR-10		MNIST		CINIC-10		EMNIST	
	精度/%	时间/s	精度/%	时间/s	精度/%	时间/s	精度/%	时间/s	精度/%	时间/s
Random	71.2	2876	82.2	1386	84.2	4057	74.5	4092	86.1	2245
ChFL-Time	71.8	2632	82.7	1136	84.6	3746	75.3	3546	86.5	2152
ChFL-Loss	77.4	2843	86.3	1285	87.1	3921	80.1	3652	88.4	2140
ChFL	78.3	2810	87.2	1107	89.2	3802	81.3	3485	89.5	2073

上述实验结果分析说明：两个指标对比来看，本地损失值的实验表现效果较好。在ChFL策略中，将本地损失值和训练时间聚合时，本地损失值的权重要大于训练时间的权重，最终结合实验效果最佳。

4.2.2 ChFL 客户端选择策略实验

本节中在基线算法 FedAvg、FedProx 和 FedNova 的基础上将 ChFL 客户端选择策略和随机选择策略进行对比实验。ChFL 客户端选择策略不仅能在收敛轮数方面优于随机选择策略，在模型精度方法比随机选择策略也有较大的提升。在实验中设定 80%为各数据集的收敛精度。

从图 4 能够分析出，在五种数据集下 ChFL 客户端选择策略能率先达到收敛精度。由于 ChFL 几乎没有为客户端和服务器端增加额外的计算量，这种轮数上的优势也体现到了时间方面，能以较快的时间完成收敛。对于三种基线算法，FedAvg 算法最终提升的精度最多，在五种数据集下精度均能提升 4.5%左右，其中在 MNIST 数据集中精度提高 4.8%，在 CINIC-10 数据集中精度提高 4.9%。ChFL 策略优化 FedProx 算法最终模型精度能提高 4.2%左右，ChFL 策略优化 FedNova 算法最终模型精度能提高 4%左右。

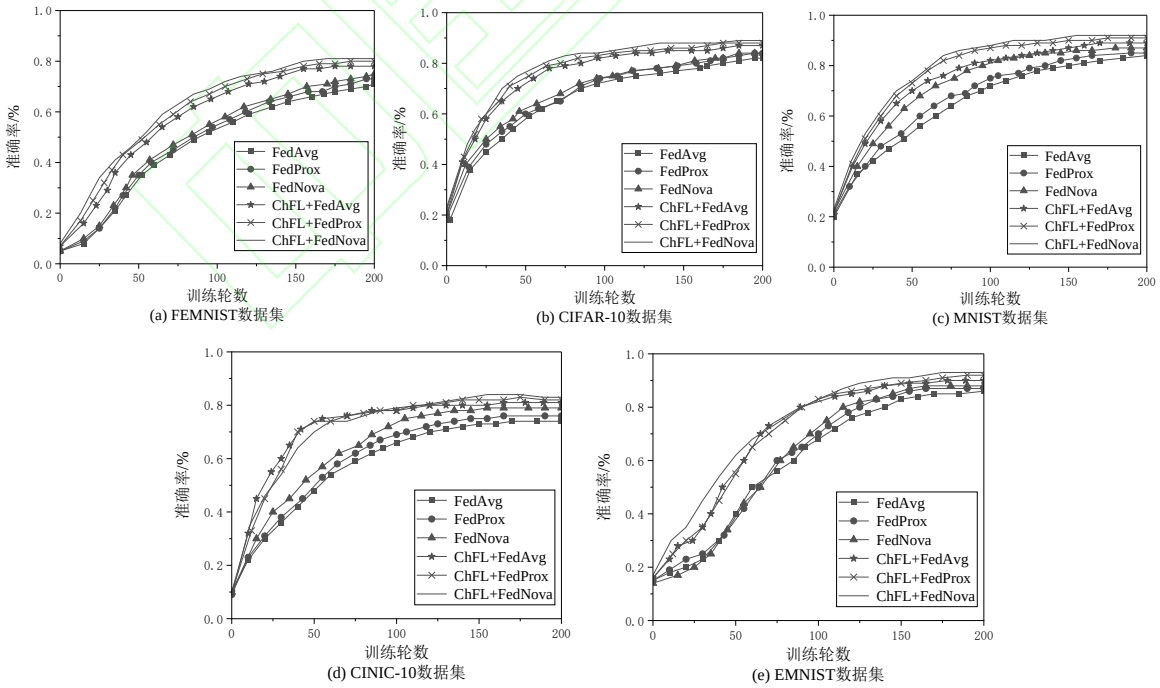


图 4 Non-IID设置下基线精度对比实验图

Fig.4 Experimental Chart of Baseline Accuracy Comparison under Non IID Setting

如表 3 所示，在五种数据集中 MNIST 数据集和 CINIC-10 数据集中优化基线算法的实验效果最好。在 MNIST 数据集中三种基线算法整体精度分别提高 4.8%、4.5%、4.2%，在 CINIC-10 数据集中三种基线算法整体精度分别提高 4.9%、4.4%、4%。在其余三种数据集中三种基线算法整体精度均能提高 4%左右。

表 3　Non-IID设置下基线精度对比实验表

Table 3　Baseline Accuracy Comparison Experiment Table under Non IID Setting　%

算法	FEMNIST	CIFAR-10	MNIST	CINIC-10	EMNIST
FedAvg	72.2	82.6	84.2	75.3	86.0
FedProx	75.9	84.2	86.7	76.8	87.2
FedNova	75.8	84.7	87.8	79.1	88.1
ChFL+FedAvg	76.3	87.4	89.0	80.2	90.4
ChFL+FedProx	80.0	88.6	91.3	81.2	92.3
ChFL+FedNova	80.5	89.1	92.0	83.1	93.1

表4-5的实验结果表示ChFL客户端选择策略的收敛轮数和时间都优于随机选择策略。其中 *Accuracy@acc* 表示模型首次达到精度 *acc* 时所需的轮数和时间。从表中结果能够得出：在 FEMNIST 数据集下算法优化的效果最好。FedAvg 算法收敛轮数提升 15.2%，收敛时间快 10.6%；FedProx 算法收敛轮数提升 13.6%，收敛时间快 7.1%；FedNova 算法收敛轮数提升 13.7%，收敛时间快 6.2%。在其他数据集中三种基线算法收敛轮数均能提升 10%左右，收敛时间加快 7%左右。

表 4　FEMNIST数据集的收敛轮数和时间对比实验表

Table 4　Comparison Experiment Table of Convergence Rounds and Time of FEMNIST Dataset

算法	Accuracy@50		Accuracy@60		Accuracy@70		Accuracy@80	
	轮数	时间/s	轮数	时间/s	轮数	时间/s	轮数	时间/s
FedAvg	85	1088	120	1146	175	1543	250	2857
FedProx	85	1056	113	1155	164	1493	242	2846
FedNova	84	1032	110	1159	159	1474	225	2873
ChFL+FedAvg	72	972	105	1120	155	1520	215	2832
ChFL+FedProx	70	962	93	1128	135	1470	209	2756
ChFL+FedNova	71	928	85	1072	122	1440	194	2754

表 5　CIFAR-10 数据集的收敛轮数和时间对比实验表

Table 5　Comparison Experiment Table of Convergence Rounds and Time of CIFAR-10 Dataset

算法	Accuracy@60		Accuracy@70		Accuracy@80		Accuracy@90	
	轮数	时间/s	轮数	时间/s	轮数	时间/s	轮数	时间/s
FedAvg	53	526	85	597	175	968	265	1386
FedProx	50	502	84	583	158	828	240	1293
FedNova	49	523	74	674	146	795	238	1127
ChFL+FedAvg	48	472	69	563	132	762	214	1107
ChFL+FedProx	45	453	65	562	134	744	209	953
ChFL+FedNova	40	446	63	526	128	739	203	922

由表 6 可以看出，在三种基线算法上 ChFL 客户端选择策略相比于随机选择策略没有传输额外的信息量，达到收敛时需要的总通信量在大多数数据集中优于随机选择策略。

表 6　Non-IID设置下算法通信量对比实验表

Table 6　Comparison Experiment Table of Algorithm Traffic under Non IID Setting　　s

算法	FEMNIST		CIFAR-10		MNIST		CINIC-10		EMNIST	
	每轮	收敛	每轮	收敛	每轮	收敛	每轮	收敛	每轮	收敛
FedAvg	22	1341	25	1225	24	1306	25	1375	23	1315
FedProx	22	1306	25	1124	24	1285	25	1207	23	1274
FedNova	23	1453	31	1573	25	1340	31	1875	25	1385
ChFL+FedAvg	22	1258	25	1176	24	1237	25	1249	23	1141
ChFL+FedProx	22	1226	25	1087	24	1129	25	1108	23	1075
ChFL+FedNova	22	1217	25	1196	24	1146	25	1386	23	1106

4.2.3 超参数α设置实验

在五种数据集下对超参数α的设置进行了对比实验。超参数α决定了使用指标值作为选择依据和随机选择策略的比例，α的值越小表示更多的客户端将会由指标值选出，剩下的客户端将会由随机选择策略选出。虽然使用指标值选择客户端可以更快的帮助全局模型达到收敛，但是过多的采用这种策略会影响最终模型的精度上限。因此在本节中选择了三个α的值，分别是 0.2,0.4 和 0.8，通过实验验证最合适的α取值。

从图 5 和表 7 可以分析出三种α值下的 ChFL 都比随机选择策略的算法精度高，其中$\alpha=0.4$时算法表现最好，在 CINIC-10 数据集中算法精度比随机选择策略的精度要高 5.3%，在其余数据集中精度都要高 4%左右。当$\alpha=0.2$时表示利用指标值选出大部分的客户端，小部分利用随机选择策略选出，FedAvg 算法也能提高 3%左右的精度。尽管$\alpha=0.2$时算法精度表现效果不错，但是精度低于$\alpha=0.4$时的算法精度，这表明利用指标值选择客户端也不能一味的使用，需要适度的搭配随机选择策略效果最佳。

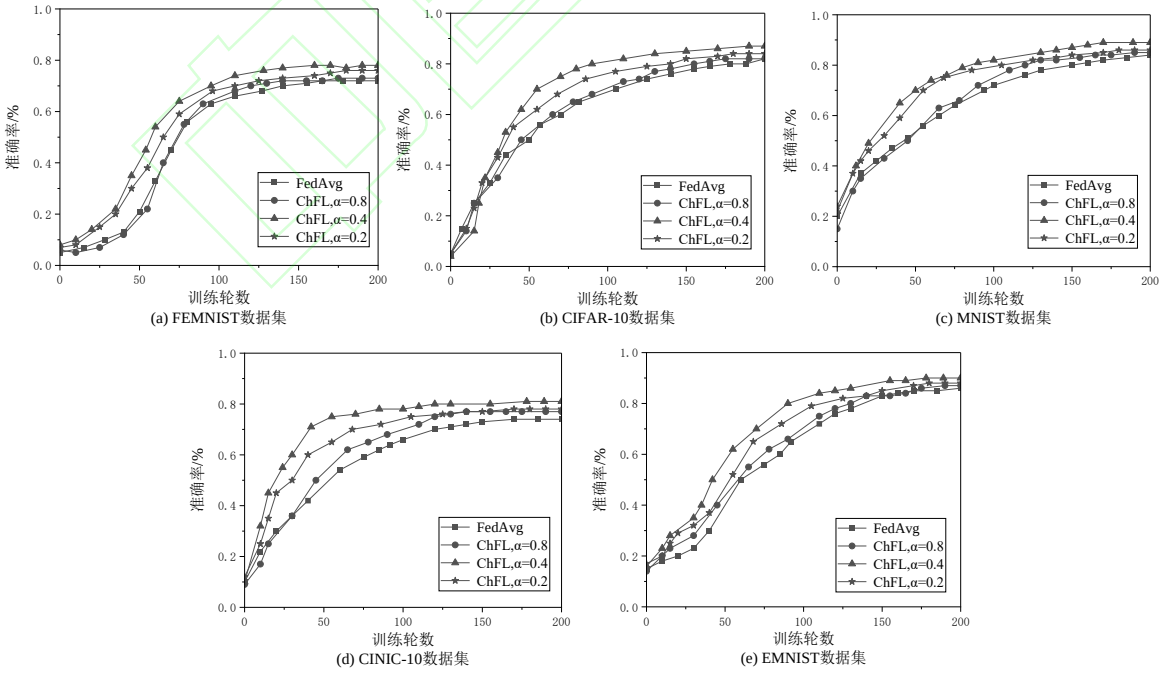


图 5　Non-IID设置下FedAvg算法精度实验图

Fig.5　Experiment Chart of FedAvg Algorithm Accuracy under Non IID Setting

表 7　Non-IID设置下FedAvg算法精度实验表

Table 7　Experiment Table of FedAvg Algorithm Accuracy under Non IID Setting　%

α值	FEMNIST	CIFAR-10	MNIST	CINIC-10	EMNIST
FedAvg	72.8	82.2	84.3	75.9	86.3
ChFL,α=0.2	76.8	84.5	86.2	78.6	88.4
ChFL,α=0.4	77.2	87.1	89.1	81.2	90.1
ChFL,α=0.8	73.6	82.7	85.7	77.9	87.6

5 结束语

在联邦学习系统中，传统的随机选择客户端的策略使得全局模型质量受到参与训练的客户端的深度影响，本文提出了一种客户端选择策略 ChFL 以提高模型训练的性能。该方法利用损失值和训练时间对模型收敛的影响来改变客户端的选择概率,达到提升模型精度和收敛速度的效果。ChFL 提取可反映客户端模型质量的损失值和训练时间两个重要指标，将指标融合建模，用于评估客户端模型质量。根据客户端质量指导客户端选择，选择拥有高质量的数据且计算性能较好的客户端参与训练。实验结果表明与随机选择客户端策略相比，该方案能够最大限度地降低通信和计算成本，提高资源利用效率。下一步将扩展本文可反映客户端本地模型质量的指标，优化指标值融合建模的方法，研究更好的基于本地模型质量指导客户端选择的方法,进一步提升训练效果。

参考文献

[1] McMahan H B, Ramage D, Talwar K, et al. Learning differentially private recurrent language models[J]. arXiv preprint arXiv:1710.06963, 2017.

[2] Apple Differential Privacy Team. Learning with privacy at scale. InApple Machine Learning Journal, 2017.

[3] Hartmann F, Suh S, Komarzewski A, et al. Federated learning for ranking browser history suggestions[J]. arXiv preprint arXiv:1911.11807, 2019.

[4] https://www.tensorfow.org/federated 2020. Tensorflow Federated. [Online; accessed 21-January-2020].

[5] Caldas S, Duddu S M K, Wu P, et al. Leaf: A benchmark for federated settings[J]. arXiv preprint arXiv:1812.01097, 2018.

[6] https://github.com/PaddlePaddle/PaddleFL 2020. PaddleFL. [Online; accessed 21-January-2020].

[7] Ryffel T, Trask A, Dahl M, et al. A generic framework for privacy preserving deep learning[J]. arXiv preprint arXiv:1811.04017, 2018.

[8] Li Q, Diao Y, Chen Q, et al. Federated learning on non-iid data silos: An experimental study[C]//2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE, 2022: 965-978.

[9] Zhu H, Xu J, Liu S, et al. Federated learning on non-IID data: A survey[J]. Neurocomputing, 2021, 465: 371-390.

[10] Zhao Y, Li M, Lai L, et al. Federated learning with non-iid data[J]. arXiv preprint arXiv:1806.00582, 2018.

[11] 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.

[12] Wang H, Kaplan Z, Niu D, et al. Optimizing federated learning on non-iid data with reinforcement learning[C]//IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020: 1698-1707.

[13] Konečný J, McMahan H B, Yu F X, et al. Federated learning: Strategies for improving communication efficiency[J]. arXiv preprint arXiv:1610.05492, 2016.

[14] 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. 2020: 125-136.

[15] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR, 2017: 1273-1282.

[16] Karimireddy S P, Kale S, Mohri M, et al. SCAFFOLD: Stochastic Controlled Averaging for On-Device Federated Learning[J]. 2019.

[17] Zhu Z, Hong J, Zhou J. Data-free knowledge distillation for heterogeneous federated learning[C]//International Conference on Machine Learning. PMLR, 2021: 12878-12889.

[18] 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.

[19] 邱天晨, 郑小盈, 祝永新, 等. FedFog：面向非独立同分布数据的联邦学习架构[J]. 计算机工程, doi: 10.19678/j.issn.1000-3428.0064016.
QIU Tianchen, ZHENG Xiaoying, ZHU Yongxin, et al. FedFog: Federated Learning Architecture for Non-IID Data[J]. Computer Engineering, doi: 10.19678/j.issn.1000-3428.0064016.

[20] 陈乃月, 金一, 李浥东, 等. 基于区块链的公平性联邦学习模型[J]. 计算机工程, 2022, 48(6): 33-41.
CHEN Naiyue, JIN Yi, LI Yidong, et al. Federated Learning Model with Fairness Based on Blockchain[J]. Computer Engineering, 2022, 48(6): 33-41.

[21] Ribero M, Vikalo H. Communication-efficient federated learning via optimal client sampling[J]. arXiv preprint arXiv:2007.15197, 2020.

[22] Lai F, Zhu X, Madhyastha H V, et al. Oort: Informed participant selection for scalable federated learning[J]. arXiv preprint arXiv:2010.06081, 2020.

[23] Li X, Qu Z, Tang B, et al. Stragglers are not disaster: A hybrid federated learning algorithm with delayed gradients[J]. arXiv preprint arXiv:2102.06329, 2021.

[24] Cho Y J, Wang J, Joshi G. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies[J]. arXiv preprint arXiv:2010.01243, 2020.

[25] Katharopoulos A, Fleuret F. Biased importance sampling for deep neural network training[J]. arXiv preprint arXiv:1706.00043, 2017.

[26] Katharopoulos A, Fleuret F. Not all samples are created equal: Deep learning with importance sampling[C]//International conference on machine learning. PMLR, 2018: 2525-2534.

[27] da Costa Pereira C, Dragoni M, Pasi G. Multidimensional relevance: Prioritized aggregation in a personalized Information Retrieval setting[J]. Information processing & management, 2012, 48(2): 340-357.

[28] Huang T, Lin W, Wu W, et al. An efficiency-boosting client selection scheme for federated learning with fairness guarantee[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(7): 1552-1564.

[29] He C, Li S, So J, et al. Fedml: A research library and benchmark for federated machine learning[J]. arXiv preprint arXiv:2007.13518, 2020.