联邦学习中基于时分多址接入的用户调度策略

陶梅霞，王栋，孙瑞，张乃夫
（上海交通大学电子信息与电气工程学院，上海 200240）

摘 要：为了提高联邦学习的通信效率，针对用户计算能力和信道状态异构的场景，提出了一类基于时分多址接入的用户调度策略，在满足给定单轮模型训练所需计算的样本数量约束下，最小化单轮模型更新的系统时延。理论分析了该调度策略的预期收敛速度，探究收敛性能与系统总时延的均衡关系，并进一步分析最优批大小的选择问题。仿真结果显示，所提算法与基准算法相比，模型收敛速率提升 30%以上。

关键词：联邦学习；背包问题；用户调度；收敛性分析；边缘智能
中图分类号：TN929.5
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021056

TDMA-based user scheduling policies for federated learning

TAO Meixia, WANG Dong, SUN Rui, ZHANG Naifu
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Abstract: To improve the communication efficiency in FL (federated learning), for the scenario with heterogeneous edge user's computing capacity and channel state, a class of time division multiple access (TDMA) based user scheduling policies were proposed for FL. The proposed policies aim to minimize the system delay in each round of model training subject to a given sample size constraint required for computing in each round. In addition, the convergence rate of the proposed scheduling algorithms was analyzed from a theoretical perspective to study the tradeoff between the convergence performance and the total system delay. The selection of the optimal batch size was further analyzed. Simulation results show that the convergence rate of the proposed algorithm is at least 30% higher than all the considered benchmarks.

Keywords: federated learning, knapsack problem, user scheduling, convergence analysis, edge intelligence

1 引言

随着 5G 移动网络在全球范围内逐渐普及，万物互联时代已到来，人工智能技术的应用也正从云端向网络边缘延伸。基于智能手机、可穿戴设备、无人机等边缘设备的智能应用需求，如人脸识别、智能监控、智能驾驶、路径规划等不断涌现。传统的机器学习算法（包括训练和推理）通常部署在云数据中心。为了训练更准确的人工智能模型，边缘设备需要将所采集的海量原始数据通过移动网络发送至云端，这会给网络带来巨大的带宽压力，并面临用户隐私泄露的风险。得益于移动边缘计算架构的发展[1]，将机器学习部署在网络边缘成为可能。边缘学习[2-4]允许终端设备将原始数据保留在本地，在边缘服务器的协调下参与模型训练和推理，从而有效缓解网络带宽压力，降低响应时延，并提升数据隐私性。边缘学习是通信与计算学科融合的前沿方向，被认为是“人工智能的最后一千米”[3]。

联邦学习是一种非常有潜力的边缘学习框架，由 Google 研究人员于 2016 年提出[5]，受到了学术界和工业界的广泛关注。联邦学习允许多个分布式边缘设备在边缘服务器的协调下，共同训练一个模

收稿日期：2020−08−03；**修回日期**：2020−11−13
基金项目：国家重点研发计划基金资助项目（No.2019YFB1802702）；国家自然科学基金资助项目（No.61941106）
Foundation Items: The National Key Research and Development Program of China (No.2019YFB1802702), The National Natural Science Foundation of China (No.61941106)

型，而不需要上报各自的原始数据样本。在典型的联邦学习过程中，每个参与用户会先从服务器中下载当前最新的全局模型，然后利用本地数据样本进行局部训练，并将梯度信息上传给服务器，服务器聚合各个用户的梯度信息后更新模型参数，再将更新后的模型返回给参与用户。

联邦学习虽是一种特殊的分布式学习，但基于无线网络边缘，其性能受限于可用的无线网络通信资源及边缘设备自身的计算能力。由于联邦学习是一个多轮次的迭代更新过程，如果每轮都有大量的用户将自己的梯度信息上传给服务器，那么整个训练过程会消耗大量的通信资源。目前，提高无线网络中联邦学习的通信效率已有不少研究。一类方法是降低单次模型聚合中的通信消耗，如模型量化[6]、稀疏化[7]等。另一类方法是用户调度，即在每一轮模型更新中选择部分用户参与训练。调度的用户数越少，通信资源消耗越小，但是模型收敛越慢。因此，用户调度需要在资源消耗与模型收敛性能之间找到最佳的平衡。此外，边缘设备的算力和数据分布的异构性也为用户调度增加了挑战性。因此，用户调度策略是联邦学习的主要研究内容。有学者提出采用空中计算来提升联邦学习中模型聚合的通信效率，即利用无线信道天然的叠加特性，让所有参与学习的用户同时传输模拟信号，在空中完成模型聚合运算[8-11]，但是这种方法在实际过程中需要非常严格的同步。

本文旨在提出一种新的用户调度策略，并对基于该策略的学习模型收敛性进行分析。该策略采用时分多址接入（TDMA, time division multiple access）方式，允许每个尚未被调度的用户在其他用户上传梯度信息的时间段内继续训练本地样本数据，直到被调度，实现系统整体“边算边传”的计算通信高效融合，从而提升无线网络中联邦学习的性能。

本文的主要贡献如下。

1) 提出了一种充分挖掘 TDMA 系统边算边传特性的用户调度策略。该策略考虑用户信道增益与计算能力的异构性，在每一轮模型更新时，以满足所有参与用户的训练样本总量不少于给定门限为约束，通过优化参与用户集、用户上传顺序及各自训练所用的样本数量，最小化模型更新时延。本文证明最优的用户调度顺序应满足计算能力和信道状态相对较强的用户较后上传。通过将原问题退化为一维背包问题，以当前用户的已计算样本量作为背包价值，运用动态规划算法获得最优的调度用户集。

2) 由于用户间数据的异构性，单纯满足计算样本量无法保证非独立同分布（Non-IID, non independent and identically distributed）场景下模型收敛。本文考虑用户数据的类别分布差异，在独立同分布（IID, independent and identically distributed）场景用户调度策略的基础上，引入约束样本均衡度的数学模型，让边缘服务器自动决策用户的计算样本类别和各类数量，提高调度用户训练样本的均衡性，在降低系统时延的基础上提高模型训练的准确率。

3) 分析了模型的收敛性能，并基于所提出的用户调度策略，分析收敛性能与系统总时延之间的均衡关系，探讨了在给定收敛性能目标下能够最小化系统总时延的批大小选择。

4) 仿真结果表明，本文算法具有良好的降低通信系统时延的性能。与基准调度策略的比较，验证了本文算法的有效性。

2　现有相关工作

有很多工作对联邦学习的用户调度策略进行了研究。其中，文献[12-15]主要关注在给定通信资源与计算资源的情况下用户的调度策略。具体来说，文献[12]以更新样本的年龄（AoU, age of update）作为性能指标，提出了一种用户调度策略，通过联合考虑用户的 AoU 与信道状态，提升联邦学习的运行效率。文献[13]通过贪心算法选择时延最小的用户，最小化单轮传输的总时延。文献[14]分析了用户调度策略与小区间干扰对模型收敛率的影响，并比较了随机调度、轮询和比例公平调度策略的收敛性。文献[15]提出了一种用户调度策略，联合考虑用户信道特性与用户本地模型的“重要性”，与只考虑单一用户特性的调度策略相比，提升了模型的收敛率与准确度。文献[16-20]研究了用户调度策略与无线资源分配的联合优化。具体地，文献[16]提出了一种基于非正交多址接入的用户调度策略，通过联合优化用户调度和功率分配最大化加权数据速率和，以提高学习效率。文献[17]提出了一种启发式的调度和资源分配策略，联合考虑信道状况和本地更新模型的重要性，并分析了该策略的模型收敛性。文献[18]联合优化了用户调度策略

与带宽分配，最小化系统能量消耗。文献[19]联合优化了上行链路资源分配和调度用户数目，最大化联邦学习的渐近收敛性能。以上研究均优化单轮的用户调度策略，没有给出收敛率与总资源消耗（如时间）的关系，且无法从理论上确定全局最优用户调度策略。文献[20]提出了一种联合用户调度和资源分配策略，通过分析模型收敛率与调度用户数量和迭代次数的关系，找到最优的用户调度数量，最大化模型收敛速度。

一般来说，联邦学习传输过程通常采用正交多址接入的方式，如 OFDMA（orthogonal frequency division multiple access）和 TDMA 与边缘服务器通信，文献[18-20]均采用 OFDMA 的传输模型。OFDMA 被长期演进（LTE, long term evolution）系统采用，允许所有参与调度的用户在完成计算任务的前提下，在不同的频谱资源块上同时传输更新的模型。而 TDMA 传输被 Wi-Fi 系统所采用，在同一时间只允许单个用户上传任务，但可以占用整个带宽资源。与基于 OFDMA 的联邦学习相比，基于 TDMA 的联邦学习在一个用户上行传输时允许其他用户利用此时间继续任务计算。因此，基于 TDMA 的学习在消耗同等传输带宽与传输时间的情况下，不需要额外消耗计算时间，可以降低系统的总时延，并且模型更新的计算量越大，时延降低越显著。

然而，现有工作都没有充分挖掘 TDMA 的这一优势。文献[13]虽提出了一种基于 TDMA 方式的用户调度策略，但是没有考虑所选用户的计算任务对模型的贡献度。另一方面，文献[13]假设用户用于模型更新的样本量固定，本地更新模型所需计算资源固定。然而，在异构网络中，计算能力强的设备，在同等时间内可以计算更多的样本，从而加快模型的收敛。因此，上述工作的假设无法充分利用设备计算资源。

与现有工作对比，本文提出的基于 TDMA 的用户调度策略不仅充分利用了系统边算边传的优势，还考虑了设备的计算能力和信道增益的异构性，分别在 IID 数据集与 Non-IID 数据集下，优化了用户调度集合和用户训练的数据量，最小化单轮模型更新时延。本文还进一步基于模型收敛率分析，探讨了批大小与训练总时延的关系。

3 系统模型

3.1 联邦学习模型

考虑如图 1 所示的联邦学习系统，M 个边缘设备（用户）在一个边缘服务器（无线接入点）的协调下共同训练一个学习模型。边缘设备与边缘服务器之间通过无线信道进行上下行通信。主要系统参数如表 1 所示。

定义集合 $\mathcal{M}=\{1,2,\cdots,M\}$ 为全部用户集。用户 m 的本地数据集定义为 $\mathcal{D}_m$，记 $D_m=|\mathcal{D}_m|$ 为用户 m 所拥有的数据样本数量，则全部用户所拥有的总的本地数据样本数量为

$$D=\sum_{m=1}^{M}D_m \tag{1}$$

由于通信和计算资源受限，在每一轮的模型更新过程中，边缘服务器只选择部分边缘设备参与梯度聚合。第 $t\in\{1,2,\cdots\}$ 轮的学习过程包括以下步骤。

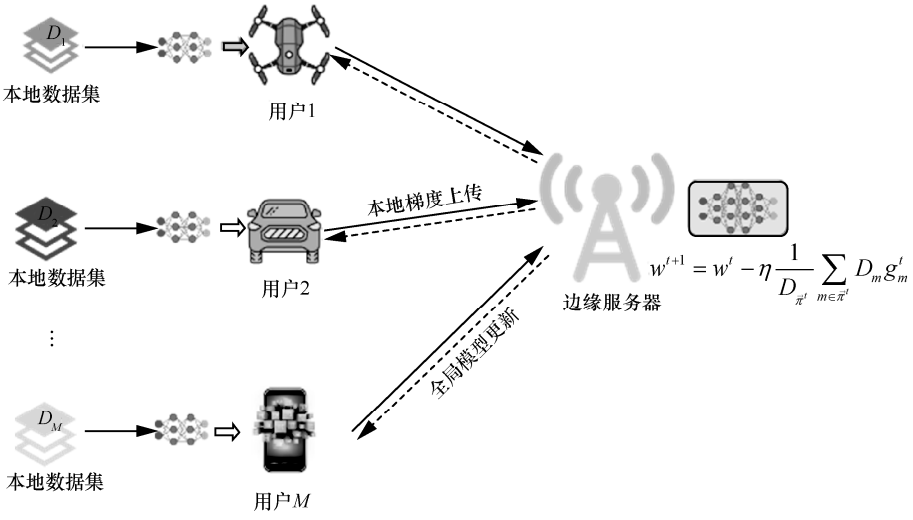


图 1 联邦学习系统

表 1　　主要系统参数

参数	含义
$\mathcal{D}_m$	用户 m 的数据集
$\mathcal{D}$	全体样本空间
D_m /个	用户 m 的数据样本量
D_m^j /个	用户 m 的第 j 类数据样本量
D /个	全体用户的数据样本量
d_m /个	服务器让用户 m 计算的样本量
d_m^j /个	服务器让用户 m 计算的第 j 类样本量
$\boldsymbol{w}^t$	第 t 轮更新的全局模型
$\boldsymbol{w}_m^t$	用户 m 第 t 轮更新的本地模型
$\boldsymbol{g}_m^t$	用户 m 第 t 轮更新的模型梯度
η	学习率
Q /bit	模型量化的比特大小
W /Hz	传输带宽
γ_m^t /dB	第 t 轮用户 m 的上行链路信噪比
τ_m^t /s	第 t 轮用户 m 的上行传输时间
c_m^t /s	第 t 轮用户 m 的计算时间
p_m /(样本·秒$^{-1}$)	用户 m 的计算能力
B/个	所有调度用户计算的总样本量大小
$\mathcal{M}$	全体用户集
$\vec{\pi}^t$	第 t 轮调度用户集
N	全局聚合次数
$f(\cdot)$	损失函数
$R(\cdot)$	收敛边界函数
$G_{\max}$	目标收敛性能对应的收敛边界值
T /s	单论训练和调度时延
T_{all} /s	系统总时延

1) 用户选择。边缘服务器根据一定的策略进行用户选择，记 $\mathcal{M}^t \subseteq \mathcal{M}$ 为第 t 轮被调度的用户集合。本文提出的用户调度策略将在第 4 节详细介绍。

2) 全局模型广播。边缘服务器将最新的全局模型 $\boldsymbol{w}^t$ 通过无线信道广播给全体用户。

3) 本地训练。被调度的用户 $m \in \mathcal{M}^t$ 接收到全局模型 $\boldsymbol{w}^t$ 后，基于本地数据集，采用随机梯度下降（SGD, stochastic gradient descent）法执行本地模型训练。本文参考有监督的机器学习任务，定义损失函数 $f(\boldsymbol{w}^t, x)$ 表示在模型 $\boldsymbol{w}^t$ 下的对训练样本 x 的预测误差。如果值较大，则预测误差较大；反之则较小。定义 $\mathcal{D}' \subseteq \mathcal{D}_m$ 为第 t 轮用户 m 本地训练用到的数据集，记该数据集大小为 $d_m = \left|\mathcal{D}_m^t\right|$，则损失函数定义为

$$F_m(\boldsymbol{w}^t) = \frac{1}{d_m} \sum_{x \in \mathcal{D}_m^t} f(\boldsymbol{w}^t, x) \tag{2}$$

用户 m 由此可获得本地梯度值为

$$\boldsymbol{g}_m^t = \nabla F_m(\boldsymbol{w}^t) \tag{3}$$

其中，∇ 是梯度操作。

4) 梯度聚合。所有被调度的用户将本地梯度 $\boldsymbol{g}_m^t$ 通过无线信道上传至边缘服务器，边缘服务器收到后，执行聚合操作，获得全局梯度为

$$\boldsymbol{g}^t = \sum_{m \in \mathcal{M}^t} d_m \boldsymbol{g}_m^t \tag{4}$$

在这一过程中，用户可以对梯度信息进行量化和压缩，来减少通信资源的消耗和系统的时延。

5) 全局模型更新。边缘服务器根据全局梯度信息更新全局模型，更新过程为

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \frac{1}{\sum_{m \in \mathcal{M}^t} d_m} \boldsymbol{g}^t \tag{5}$$

其中，$\eta \geqslant 0$ 表示学习率。

3.2　通信与计算模型

联邦学习过程所消耗的时延包括通信时延和计算时延两部分。通信时延通常与用户的信道状态、通信带宽、传输功率以及梯度量化比特有关。计算时延的主要影响因素通常有设备的计算能力、数据集的大小、单个数据样本的特性和训练算法的选择。下面分别介绍 2 种时延的建模方法。

由于下行链路的通信速率通常远大于上行链路的通信速率，并且边缘服务器可以采用广播的方式与用户进行下行通信，因此在实际系统中，联邦学习的通信时延由上行链路的通信时延主导。因此，本文不考虑下行链路通信引起的时延。本文采用基于 TDMA 的上行传输，同一时间段只有一个用户在通信，并且可以占用整个系统带宽。基于“边算边传”的特点，被调度用户的梯度上传顺序至关重要。在每一轮模型更新中，将被调度的用户集 $\mathcal{M}^t$ 按照用户上传顺序排列，重新定义为有序集 $\vec{\pi}^t = \{\pi^t(1), \pi^t(2), \cdots, \pi^t(k)\}$，其中，$\pi^t(m)$ 表示在第 t 轮里被选择第 m 个上传梯度的用户索引，$k \triangleq \left|\mathcal{M}^t\right|$ 是在该轮被调度的用户总数（一般来说，每轮调度用户的总数可能不同，此处为简化表述，省去了 t）。用户本地计算和梯度传输的时隙关系如图 2 所示，其中，用户 $\pi(m)$ 一直处于本地计算状

态，直至前一用户 $\pi(m-1)$ 上行传输结束才停止本地计算，并开始上传计算所得的梯度。在图 2 中，$c_{\pi(m)}$、$T_{\pi(m)}$ 和 $\tau_{\pi(m)}$ 分别表示用户 $\pi(m)$ 的本地训练时间、梯度上传时间点和梯度上传所需通信时间。

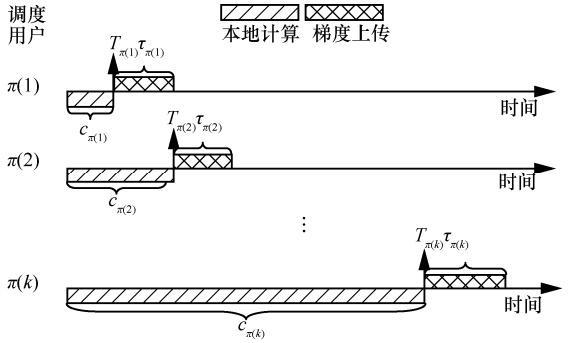


图 2 用户本地计算和梯度传输的时隙关系

由于每个本地梯度包含的元素个数相同，为方便起见，采用 Q 个比特来量化每个本地梯度。考虑用户 m 被调度，则第 t 轮用户 m 梯度上传时延（单位为 s）可以表示为

$$\tau_m^t = \frac{Q}{W \operatorname{lb}(1+\gamma_m^t)} \tag{6}$$

其中，W 为传输带宽，γ_m^t 为用户 m 在第 t 轮的信噪比。

定义 p_m 为用户 m 的计算能力，若考虑用户 m 的计算样本数量为 d_m，则用户 m 计算用时（单位为 s）可以表示为

$$c_m^t = \frac{d_m}{p_m} \tag{7}$$

如果有计算时间 c_m^t，则用户 m 可计算的样本数量为

$$d_m = p_m c_m^t \tag{8}$$

从图 2 可以看到，在每一轮全局模型更新里，第一个被调度的用户 $\pi(1)$ 拥有最短的本地计算时延预算，而最后被调度的用户 $\pi(k)$ 的本地计算时延预算最长。基于上述通信与计算模型，第 t 轮全局模型更新所需总时延可以用最后一个被调度的用户的梯度上传时间点和通信时间来表示，记为 $T_{\pi(k)}^t + \tau_{\pi(k)}^t$。

4 用户调度策略

根据第 3 节介绍的通信与计算模型，用户调度有序集 $\vec{\pi}^t$ 是影响联邦学习的模型更新时延和模型精度的关键。由现有联邦平均算法的收敛性分析可知[21]，模型精度取决于全局模型更新总的次数以及每轮更新所用的训练样本量。本节提出一种新的用户调度策略，以期在每一轮全局模型更新里满足给定所需全局样本数量的情况下，最小化所需系统时延。本节只讨论单轮传输下的最优用户调度策略，因此省略迭代次数索引 t。下面，先介绍 IID 数据分布场景下的最优用户调度策略，然后将其扩展至 Non-IID 数据分布场景。

4.1 IID 数据分布用户调度

定义 B 为联邦学习单轮参与全局梯度更新的所有用户训练的数据样本量。在已知 B 的情况下，只需要满足每轮调度用户的数据样本总量大于或等于 B 即进入下一轮迭代训练过程，那么必然存在一个基于 TDMA 通信方式的用户调度策略来最小化总的模型更新时延。

根据 3.2 节的时延模型，问题模型可以描述为

$$\min_{\forall \vec{\pi}} \ T_{\pi(k)} + \tau_{\pi(k)} \tag{9}$$

$$\text{s.t. } T_{\pi(m)} - T_{\pi(m-1)} \geqslant \tau_{\pi(m-1)}, \ \forall m \in \{2,\cdots,k\} \tag{9a}$$

$$c_{\pi(m)} \leqslant T_{\pi(m)}, \ \forall m \in \{1,\cdots,k\} \tag{9b}$$

$$\sum_{m\in\{1,\cdots,k\}} p_{\pi(m)} c_{\pi(m)} \geqslant B \tag{9c}$$

其中，$T_{\pi(m)}$ 表示用户 $\pi(m)$ 的上传时间点。结合图 2 所示的时隙图，上述约束条件式 (9a) 表示用户 $\pi(m)$ 的上传时间点至少要等到它的前一个上传用户 $\pi(m-1)$ 上传完毕，式 (9b) 表示用户 $\pi(m)$ 的计算时间不能超过其上传时间点，式 (9c) 表示本轮调度用户集的全部计算样本量不能少于门限值 B。

显然，上述优化问题属于 NP 完全问题，难以获得最优解的闭式表达式。通过引入以下 2 个引理，可以将原问题的求解退化为一维背包问题，进而获得该问题的最优数值解。定义 $\lambda_m = p_m / \tau_m$ 为用户 m 的调度重要度。

引理 1 任意 2 个相邻的调度用户满足上行传输时间点无间隙，即满足 $c_{\pi(m)} - c_{\pi(m-1)} = \tau_{\pi(m-1)}$。

引理 2 调度用户有序集 $\vec{\pi}$ 中的最优的用户调度顺序必须满足

$$\lambda_{\pi(k)} \geqslant \lambda_{\pi(k-1)} \geqslant \cdots \geqslant \lambda_{\pi(1)} \tag{10}$$

引理 1 和引理 2 的证明过程分别如附录 1 和附录 2 所示。

基于引理 1 和引理 2，原问题的求解可以退化

为一维背包问题的求解。区别于传统的背包问题，该问题下物品的价值（即每个用户能计算的样本）不是常量。该问题的求解需要上述引理与动态规划算法的结合，通过对背包空间的搜索和物品价值的度量，寻找在给定的背包空间（单轮迭代总时延）下所能容纳的最大物品价值（单轮迭代所能计算的样本总量），再通过二分法搜集解空间，寻求问题的最优解。

将单轮迭代总时延 S_{total} 定义为总的背包空间大小；$U[i][j]$ 定义为背包空间大小为 j 时，放入物品 i 后的价值大小。在本问题中，物品 i 的价值 $V[i]$ 定义为用户 i 的计算样本量，即

$$V[i]=p_i c_i \tag{11}$$

算法 1　基于背包问题的用户调度策略

定义 s 为背包空间搜索的步长。初始化 $\vec{\pi}$，$\mathcal{K}\leftarrow\{\}$，$\lambda\leftarrow\{\}$，$S_{\text{total}}$，当 $i\leqslant 0$ 时对全体 j 有 $U[i][j]=0$，对全体 i 有 $U[i][0]=0$。

1) 循环

2) for $i=1:1:M$，计算 λ_i，并按照从大到小的顺序依次放入 $\mathcal{K}$ 中

3) end for

4) for $k=1:1:\text{length}(\mathcal{K})$

$i=\mathcal{K}[k]$

5) for $j=0:s:S_{\text{total}}$

6) $U[i][j]=\max(U[i-1][j],U[i-1][j-\tau_i]+V[i])$

其中，$V[i]=p_i(j-\text{前}(i-1)\text{个用户占背包空间大小})$

7) end for

8) end for

9) for $i=M:1:1$

10) 递归查找 $U[i][j]$

11) 如果 i 放入背包，将 i 放入 $\vec{\pi}$

12) end for

13) 返回 $U[M][S_{\text{total}}]$ 和 $\vec{\pi}$

算法 2　基于二分法寻找最优时间

定义 $S_{\text{total}}^{\text{up}}$ 和 $S_{\text{total}}^{\text{down}}$ 为搜索最优时延的上界和下界，定义 $V[i][S]$ 为执行算法 1 后的 $U[M][S_{\text{total}}]$，即 $V[i][S]$=$U[M][S_{\text{total}}]$，其中 S 表示背包的总空间大小，并且 $V[i][S_{\text{total}}^{\text{up}}]>B$ 和 $V[i][S_{\text{total}}^{\text{down}}]<B$，初始化 ε，ε 定义为二分法求解时延的精度要求。

1) 循环

2) $S_{\text{total}}=(S_{\text{total}}^{\text{up}}+S_{\text{total}}^{\text{down}})/2$

3) 调用算法 1，获得 $V[i][S_{\text{total}}]$

4) 如果 $V[i][S_{\text{total}}]>B$ 并且 $S_{\text{total}}^{\text{up}}-S_{\text{total}}^{\text{down}}>\varepsilon$

5) $S_{\text{total}}^{\text{up}}=S_{\text{total}}$

6) 如果 $V[i][S_{\text{total}}]<B$ 并且 $S_{\text{total}}^{\text{up}}-S_{\text{total}}^{\text{down}}>\varepsilon$

7) $S_{\text{total}}^{\text{down}}=S_{\text{total}}$

8) 否则输出 $\vec{\pi}$ 和 S_{total}

通过算法 1 可以解出在给定单轮通信系统时延为 S_{total} 时的最优用户调度顺序集和所能计算的本地数据样本量。通过算法 2 对 S_{total} 的搜索可以找到满足单轮本地数据样本量大于或等于 B 所需的精度在任意 ε 以内的最小单轮通信系统时延。

计算复杂度分析。在该算法中计算和排序的复杂度均为 $\mathcal{O}(M\log(M))$，由一维背包问题的复杂度可知其复杂度为 $\mathcal{O}(MS_{\text{total}}/s)$，二分法寻找最优解的复杂度为 $\mathcal{O}(\log(S_{\text{total}}^{\text{up}}-S_{\text{total}}^{\text{down}})/\varepsilon)$，则总的计算复杂度为 $\mathcal{O}(MN(2\log(M)+S_{\text{total}}/s)\log(S_{\text{total}}^{\text{up}}-S_{\text{total}}^{\text{down}})/\varepsilon)$，其中 N 为总的通信轮次。

4.2　Non-IID 数据分布用户调度

4.1 节讨论了在 IID 场景下最优的用户调度策略。联邦学习实际场景中，由于不同设备所处的环境不同，用户行为习惯也不同，因此用户本地的数据样本会呈现 Non-IID 特性。如果仍然以最快达到边缘服务器单轮训练所需的数据样本数量为目标，那么全局模型更新方向必然偏向重要度较大的用户，导致部分用户在整个训练过程中被调度的概率非常小。以分类问题为例，要保证模型收敛，必须使全部调度用户集已训练的样本呈现均衡的特性。因此在 Non-IID 场景下，在最小化单轮系统时延的同时，需要考虑用户之间数据样本不平衡的问题。为了解决该问题，本文引入数据分布的特性，令边缘服务器知晓每个参与联邦学习的用户的每一类样本数量。假设全体数据集共有 Y 类样本，分别定义 D_m^j、d_m^j 和 B^j 为用户 m 所拥有的第 j 类样本的数量、边缘服务器需要用户 m 所计算的第 j 类样本的数量以及边缘服务器在每一轮对第 j 类样本的需求量。

基于上述关于数据平衡问题的讨论，Non-IID 场景下的用户调度策略不仅要考虑用户的重要度 λ_m，也要考虑本轮参与调度用户的总的训练样本中每个类别的样本是否满足给定的数量要求。因此，问题模型可以描述为

$$\min_{\vec{\pi}}\ T_{\pi(k)}+\tau_{\pi(k)} \tag{12}$$

$$\text{s.t. } T_{\pi(m)}-T_{\pi(m-1)}\geqslant\tau_{\pi(m-1)},\ \forall m \tag{12a}$$

$$c_{\pi(m)}\leqslant T_{\pi(m)},\ \forall m \tag{12b}$$

$$0 \leqslant d_{\pi(m)}^{j} \leqslant D_{m}^{j} \quad \forall m, j = 1,2,\cdots,Y \tag{12c}$$

$$\sum_{j=1}^{Y} d_{\pi(m)}^{j} \leqslant p_{\pi(m)} c_{\pi(m)}, \forall m \tag{12d}$$

$$0 \leqslant \sum_{m=1}^{k} d_{\pi(m)}^{j} - B^{j} \leqslant \mu, \ j = 1,2,\cdots,Y \tag{12e}$$

其中，$\pi(k)$ 表示优化调度集 $\vec{\pi}$ 中的最后一个上传用户；式 (12c) 表示边缘服务器要求用户 $\pi(m)$ 所计算的第 j 类样本数量不能超过它自身所拥有的第 j 类样本数量；式 (12d) 表示边缘服务器要求用户 $\pi(m)$ 计算的所有类别的样本数量不能大于其单轮最大的样本计算数量；式 (12e) 表示边缘服务器要求本轮所有参与调度的用户计算的第 j 类样本数量总和需满足预先设定最低要求 B^j，同时不能超过 $B^j+\mu$，否则不同类别的不均衡度会增加。

由于在建立系统模型时引入了用户的样本类别分布 D_m^j 参量，此问题若继续使用背包问题求解则需要建立背包空间与全局计算样本量以及每类样本的数据分布的关系，这是十分困难的。根据 Weierstrass 定理，存在一个标量 $\overline{\gamma}$ 使 $\{\pi(k) | c_{\pi(k)} + \tau_{\pi(k)} \leqslant \overline{\gamma}\}$ 非空和有界，因此系统模型一定存在最优解。本文提出 2 种启发式的调度方案，在联邦学习 Non-IID 场景下兼顾数据均衡性和系统时延。

4.2.1 数据平衡优先的用户调度

由于原问题不能直接求解，一个启发式的方案为首先按照引理 2 定义的用户重要度规则确定调度用户有序集 $\vec{\pi}$；然后，边缘服务器根据 $\vec{\pi}$ 计算出 d_m^j 以使更新时延最小化。

由于边缘服务器需要每个本地用户计算的第 j 类样本数量大于或等于 B^j，因此在调度用户时必须满足调度用户集中单个类别的样本总量大于或等于 B^j，算法 3 需要根据这一规则选择合理的调度用户集 $\vec{\pi}$，基于此提出基于数据平衡优先的调度策略。该策略的思想是，首先按照重要度从大到小对全体用户进行传输排序，然后边缘服务器按照此顺序遍历每个用户的数据样本使其满足式 (12) 的约束条件，在每个可能的用户组合下，通过简化问题式(12)为约束条件式(12c)～式(12e)。显然，上述 3 个约束条件都为凸函数，因此原问题转换为凸优化问题，通过常用的求解凸优化的工具包（如 CVX）可解出最优的调度集合 $\vec{\pi}$ 和 $d_{\pi(m)}^j$，其中 $\pi(m) \in \vec{\pi}$。

算法 3 基于数据平衡优先的用户调度策略

定义向量 $\boldsymbol{B}^{1\times Y}$ 为每个类别需要的样本数量。输入全部用户每个类别的样本数量 $D_{m,j}^{M\times Y}$，初始化 $\vec{\pi}$，$\mathcal{K} \leftarrow \{\}$，$\xi \leftarrow \{\}$，$S^{1\times Y} \leftarrow \{\}$。

1) 循环
2) for $i = 1:1:M$
3) 计算 λ_i，并按照从大到小的顺序放入 λ
4) end for
5) for $i = 1:1:M$
6) $a = \lambda[i]$，$S^{1\times Y} = \boldsymbol{B}^{1\times Y} - D_{m,j}^{M\times Y}[a]$，将 $S^{1\times Y}$ 中的小于零的元素设为 0；令 b 等于 $S^{1\times Y}$ 中所有元素的和
7) 如果 $b > 0$，将 a 放入 $\vec{\pi}$ 中
8) end for
9) 计算 $d_{\pi(m)}^j$，$\pi(m) \in \vec{\pi}$ 和最优 $\Gamma = c_{\pi(k)} + \tau_{\pi(k)}$
10) 将 λ 中删除 $\vec{\pi}$ 的用户顺序放入 $\mathcal{K}$ 中
11) for i in $\mathcal{K}$
12) 将 i 放入 $\vec{\pi}$ 中的末位
13) for j in $\vec{\pi}$
14) 从 $\vec{\pi}$ 中删除 $\vec{\pi}[j]$
15) 计算 $d_{\pi(m)}^j$，$\pi(m) \in \vec{\pi}$ 和最优 $\Gamma_1 = c_{\pi(k)} + \tau_{\pi(k)}$
16) 如果 $\Gamma_1 > \Gamma$，放回 $\vec{\pi}[j]$；否则不放回
17) end for
18) end for
19) 返回最小 Γ_1 以及满足它的 $d_{\pi(m)}^j$，$\pi(m) \in \vec{\pi}$

算法 3 对处理用户之间数据不平衡程度较高的场景有很好的效果。因为在做用户选择时不仅考虑了用户的计算能力和通信状态，同时由于边缘服务器根据不同类别的样本数量参与用户的选择，使单轮每一种类别的样本数量都能满足要求。但是为了平衡某些稀有类样本到计算能力或者信道状态不好的用户设备上，系统会等待直至它完成计算该类别所需的样本数量，由此会造成单轮系统时延的增加。

4.2.2 更新时延优先的用户调度

为了解决算法 3 遗留的某些稀有类样本恰好处于计算能力和信道状态都不好的用户上所带来的系统时延较高的问题，本文提出了一种不改变单轮通信系统最小时延的情况下提升数据平衡度的用户调度策略。问题描述如下。

$$\min_{\vec{\pi}} \ \sum_{j=1}^{Y} \left(\sum_{m=1}^{k} d_{\pi(m)}^{j} - B^{j} \right)^2$$

$$\text{s.t. } 0 \leqslant d_{\pi(m)}^{j} \leqslant D_m^j \ ,\forall m, j = 1,2,\cdots,Y$$
$$\sum_{j=1}^{Y} d_{\pi(m)}^{j} \leqslant p_{\pi(m)} c_{\pi(m)}, \forall m \tag{13}$$

其中，优化目标式(13)是由当前所有参与调度用户所能贡献的每一类样本实际训练数与所期望的每一类样本训练数之间的均方差。均方差目标函数由于其可导性更易于目标函数的求解是一种常用的逼近目标值的数学建模方法，通过最小化该均方差，可以有效保证数据的平衡性。当全体用户的数据样本不均衡度较低，每轮调度需要的调度用户数目满足一定数量时，选择的调度用户便有很大概率包含全体全部类别样本，因此在这一场景下本文更关心系统时延，只要使参与调度的用户计算样本均衡，就能获得较好的收敛特性。由于问题式(13)的求解需要已知当前参与的调度用户以及调度顺序，通过将式(9)的结果用于该问题的求解，可以在最小化单轮时延的基础上，同时做到联邦学习下的用户类别均衡，加快 Non-IID 场景下的模型收敛速率。

同样地，问题式(13)满足 Weierstrass 定理，在算法 1 和算法 2 后通过简单的解优化工具便能找到满足约束条件下问题式(13)的最优解。

通过合理采用数据平衡优先的用户调度策略和更新时延优先的用户调度策略，可以分别在数据不平衡度高和不平衡度低时，兼顾系统时延和模型收敛问题。

4.2.3 计算复杂度分析

在该算法中排序的复杂度为 $\mathcal{O}(M)$，通过资料查阅可知，本文问题解优化工具的计算复杂度为 $\mathcal{O}(\sqrt{2M(Y+1)}[M^3D^3+2M^4D^3+2M^3D^2+2M^3D+\ 2M^2D])$，由于需要遍历所有用户，因此总的复杂度为 $\mathcal{O}(N\sqrt{2M(Y+1)}[M^4D^3+2M^5D^3+2M^4D^2+2M^4D+\ 2M^3D])$，其中 N 为总的通信轮次。

5　收敛性分析与优化

第 4 节讨论了单轮聚合时满足给定计算样本数量（即批大小）B 的前提下，使单次聚合的系统时延最小化的用户调度问题。本节首先分析单轮总批大小对模型收敛性能的影响；然后，分析所提出的用户调度策略下单轮系统时延与批大小的关系，从而进一步探究系统总时延与收敛性能之间的均衡关系；最后，探讨在给定收敛性能目标下能够最小化系统总时延的最优批大小的选择。

5.1　学习算法收敛性

由于每轮全局模型聚合时用户的本地迭代次数为 1，收敛性分析可以等价为分布式小批量随机梯度下降算法的收敛性分析。令全体样本空间为 $\mathcal{D}$，那么训练对于该空间中样本 x 的损失函数为 $f(\boldsymbol{w},x)$，其中 $\boldsymbol{w}$ 是模型参数向量，使用大小为 B 的数据进行一轮更新的全局损失函数为

$$F(\boldsymbol{w}) = \frac{1}{B}\sum_{x\in\mathcal{B}} f(\boldsymbol{w},x) \tag{14}$$

其中，$\mathcal{B}$ 为批大小为 B 的数据组成的集合。

本文采用后悔评估函数来刻画学习算法的收敛性能。后悔值表示损失函数与最优损失的累计误差，那么 N 轮聚合结果为

$$\text{Regret} = \sum_{t=1}^{N}\left(F(\boldsymbol{w}^t) - F(\boldsymbol{w}^*)\right) \tag{15}$$

其中，$\boldsymbol{w}^*$ 是最优解，即模型的最终可达目标，$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathbb{E}_{x\sim\mathcal{D}}[f(\boldsymbol{w},x)]$。

假设损失函数满足如下条件。

1) 凸性。$f(\cdot)$ 是一个凸函数。

2) 平滑性。对于任意样本 $x\sim\mathcal{D}$，$f(\cdot,x)$ 满足 $L-$平滑条件，即对任意的模型参数向量 $\boldsymbol{w}$ 和 $\boldsymbol{w}'$，都存在 $\|\nabla f(\boldsymbol{w},x) - \nabla f(\boldsymbol{w}',x)\| \leqslant \|\boldsymbol{w}-\boldsymbol{w}'\|$。

3) 梯度分散边界。对于任意的模型参数向量 $\boldsymbol{w}$，其梯度分散情况存在一个边界值 σ，即 $\mathbb{E}_{x\sim\mathcal{D}}[\|\nabla f(\boldsymbol{w},x) - \nabla F(\boldsymbol{w})\|^2] \leqslant \sigma^2$。

4) 模型参数边界。对于任意轮次 t 的全局模型参数向量 $\boldsymbol{w}^t$，存在边界值使 $\|\boldsymbol{w}^t - \boldsymbol{w}^*\|^2 \leqslant \Delta^2$。

基于以上假设，根据文献[21]的分析结果，可得到后悔值上界，记为收敛边界函数 $R(N,B)$。

$$R(N,B) = \frac{L\Delta^2}{N} + \frac{\sigma\Delta}{\sqrt{NB}} \tag{16}$$

从式(16)可以看出，算法以 $\mathcal{O}(1/N + 1/\sqrt{NB})$ 的速率收敛。当给定批大小 B 时，收敛边界随着全局聚合次数 N 的增大以 $\mathcal{O}(1/N)$ 速率减小，并且无论 B 取值如何（$B=1$ 时等价为随机梯度下降算法），算法都会随着全局聚合次数 N 的增大最终收敛。当给定全局聚合次数 N 时，收敛边界随着 B 的增大以 $\mathcal{O}(1/\sqrt{B})$ 速率减小；当 B 趋近于无穷时，算法可等价为梯度下降算法，模型仍需要一定次数的全局聚合次数才能最终收敛。

5.2 模型收敛性与系统总时延的关系

本文模型训练的最终目标是在保证学习的收敛性能的情况下，最小化全局系统时延。下面，首先根据前文中 IID 数据分布下的单轮次调度策略推导单轮次时延，即一轮本地训练和聚合所需的时间。

记 T 为一轮总的系统时延。根据前文提出的调度策略有

$$\sum_{m=1}^{k} p_{\pi(m)}(T-\tau_{\pi(k)}-\tau_{\pi(k-1)}-\cdots-\tau_{\pi(m)})=B \quad (17)$$

定义 $p_{\min}$ 为参与调度用户中计算性能最差用户的计算能力，则

$$\sum_{m=1}^{k} p_{\min}(T-\tau_{\pi(k)}-\tau_{\pi(k-1)}-\cdots-\tau_{\pi(m)})\leqslant B \quad (18)$$

$$\sum_{m=1}^{k} (T-\tau_{\pi(k)}-\tau_{\pi(k-1)}-\cdots-\tau_{\pi(m)})\leqslant \frac{B}{p_{\min}} \quad (19)$$

定义 $\tau_{\max}$ 为上传用户集中信道状态最差的用户的上传时间，则

$$\tau_{\pi(m)}\leqslant \tau_{\max},\ \ \forall m\in\{1,\cdots,k\} \quad (20)$$

令 $k=\left|\vec{\pi}^{t}\right|$，则

$$\sum_{m=1}^{k}(T-\tau_{\pi(k)}-\tau_{\pi(k-1)}-\cdots-\tau_{\pi(m)})\geqslant\sum_{m=1}^{k}(T-m\tau_{\max}) \quad (21)$$

或

$$kT-\frac{k(k-1)}{2}\tau_{\min}\leqslant\frac{B}{p_{\min}} \quad (22)$$

进一步地，因为 $\tau_{\max}$ 为信道状态最差的用户的上传时间，按照本文的调度规则，即

$$k\tau_{\max}\geqslant T \quad (23)$$

代入式(19)，则

$$\frac{T^2}{\tau_{\max}}-\frac{\frac{T^2}{\tau_{\max}}-T}{2}=\frac{T^2}{2\tau_{\max}}+\frac{T}{2}\leqslant\frac{B}{p_{\min}} \quad (24)$$

则 T 与 B 的关系为

$$\frac{p_{\min}T^2+p_{\min}\tau_{\max}T}{2\tau_{\max}}\leqslant B \quad (25)$$

对上述不等式取等得到最终的 $T(B)$表达式为

$$T=\frac{-p_{\min}\tau_{\max}+\sqrt{p_{\min}^2\tau_{\max}^2+8p_{\min}\tau_{\max}B}}{2p_{\min}}<\sqrt{\frac{2\tau_{\max}B}{p_{\min}}} \quad (26)$$

由式(26)可以看出，单轮调度时间与总批大小成 $\mathcal{O}(\sqrt{B})$ 比例的关系，随着总批大小的增加，单论调度时间的上界呈1/2 指数次幂增长，虽然对于时间的推导进行了很大尺度的缩放，假定所有用户的计算能力与信道状态都与最差的用户一致，但是最终单轮调度时间与总批大小总是成 $\mathcal{O}(\sqrt{B})$ 比例的关系。

最终系统的总时延为

$$T_{\text{all}}=N\sqrt{\frac{2\tau_{\max}B}{p_{\min}}} \quad (27)$$

从式(27)可以看出，系统总时延随着总批大小 B 和全局聚合次数 N 的增大而增大。结合前文所得收敛边界结果，即式(16)，可以看出增加聚合次数和总批大小会对收敛带来增益，但是同样增加了系统总时延，并且性能增益 $\mathcal{O}(1/N+1/\sqrt{NB})$ 与系统时延 $\mathcal{O}(N\sqrt{B})$ 处于同一量级。因此，最终肯定存在最优的调度方针使满足性能需求的前提下最小化总时延。

接下来，进一步分析收敛性能与系统总时延之间的关系，设 $G_{\max}$ 为所需要达到的收敛边界，那么必然要求收敛边界满足

$$\frac{L\Delta^2}{N}+\frac{\sigma\Delta}{\sqrt{NB}}\leqslant G_{\max} \quad (28)$$

首先，探讨聚合次数为定值的情况下，收敛性能与系统总时延的均衡关系。将总批大小写成关于聚合次数的表达式，即

$$\sqrt{B}\geqslant\frac{\sigma\Delta}{\left(G_{\max}-\frac{L\Delta^2}{N}\right)\sqrt{N}} \quad (29)$$

在给定聚合轮次下，总系统时延随着总批大小的增加而增加，因此 B 取下界，此时总批大小选择是一个关于系统可达性能的表达式，将结果代入总时延表达式可得

$$T_{\text{all}}=\sqrt{\frac{2\tau_{\max}}{p_{\min}}}\frac{\sigma\Delta}{\frac{G_{\max}}{N^{1/2}}-\frac{L\Delta^2}{N^{3/2}}} \quad (30)$$

归一化系统总时延与归一化收敛性能二者之间的关系如图 3 所示。从式(30)中可以看出，当给定聚合次数 N 时，系统总时延与收敛性能近似成反比关系，符合图 3 中显示的变化趋势。结合式(29)绘制图 3 中选取点的批大小 B 的取值，可以看出，随着批大小的增大，收敛边界值与总批大小的1/2 次幂以一定比例值缩小，系统总时延以该比例值近似放大。此外，由于影响因子 $L\Delta^2/N^{3/2}$ 的存在，收敛边

界存在最小值，即无论如何训练，收敛总是不能变为零，而只能趋近于一个值。

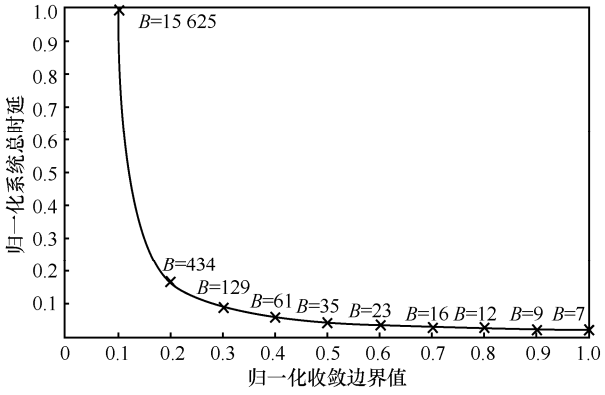


图 3　收敛边界与系统总时延的均衡关系（固定聚合次数）

接着，进一步探讨最优总批大小选择的问题。将聚合次数写成关于总批大小的表达式，则

$$N \geqslant \frac{\dfrac{\sigma^2\Delta^2}{B}+2G_{\max}L\Delta^2+\sigma\Delta\sqrt{\dfrac{\sigma^2\Delta^2}{B^2}+\dfrac{4G_{\max}L\Delta^2}{B}}}{2G_{\max}^2} \tag{31}$$

在给定总批大小下，总系统时延随着聚合轮次的增加而增加，因此 N 取下界，将结果代入总时延表达式(30)可得

$$T_{\text{all}}=\Delta^2\sqrt{\frac{2\tau_{\min}}{p_{\min}}}\frac{\dfrac{\sigma^2}{\sqrt{B}}+2G_{\max}L\sqrt{B}+\sigma\sqrt{\dfrac{\sigma^2}{B}+4G_{\max}L}}{2G_{\max}^2} \tag{32}$$

在给定可达目标的情况下，式(32)中存在最小值，即批大小存在最优选择。由于式(32)存在很多的假设和缩放，因此无法得出实际数值解，只能得到变化趋势，最优批大小的选择会通过仿真得到。

6　实验结果

本文基于开源框架 PyTorch，利用真实数据集来验证所提用户调度策略相较于其他算法的有效性，并检验收敛性分析的准确性。本文实验使用了以下 2 个常用的数据集。

1) MNIST。MNIST 手写数字数据集是机器学习领域非常经典的数据集，包含手写数字 0~9 共 10 种类型的图片。该数据集包含 60 000 个用于训练的训练样本和 10 000 个用于测试的测试样本，每个图片都是 28 像素×28 像素、值为 0～1 的固定大小。

2) CIFAR-10。CIFAR-10 是一个用于识别通用对象的小型数据集，由 10 类共 60 000 个 32 像素×32 像素的彩色图像组成。每类有 6 000 个图像。这些图像分为 50 000 个训练图像和 10 000 个测试图像，包含飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船、卡车。

6.1　IID 场景

本节选用 3 种常用的用户调度策略，即随机调度、轮询调度和比例公平调度，以文献[13,18]提出的 2 种不同通信接入方式下的用户调度策略作为基准，与本文所提用户调度策略进行性能对比。

1) 随机调度。在每一轮迭代中，边缘服务器随机选择部分用户作为当前轮次调度用户集，当满足全局聚合所需的批大小时，用户停止上传，边缘服务器通过聚合将更新的模型广播给所有用户，并继续按照此规则进行下一轮迭代。

2) 轮询调度。在每轮迭代中，边缘服务器轮流选择用户集中的用户上传，当满足全局聚合所需的批大小时，用户停止上传，边缘服务器通过聚合将更新的模型广播给所有用户，并继续按照此规则进行下一轮迭代。该算法的优点是简单，不需要记录当前的连接状态，是一种无状态调度。

3) 比例公平调度。该算法保证用户间的长期公平性，能兼顾信道状态较好的用户与信道状态较差的用户，在选择用户时考虑瞬时速率和长期平均速率的比值，按照比值从大到小的顺序调度用户。注意到，该算法仅考虑用户在传输速率上的公平性，并不考虑用户在计算能力上的公平性。

4) 文献[13]策略。该文献采用 TDMA 的接入方式，通过每轮选择总时延 τ_m+T_m 较小的若干用户进行上传，在仿真中本文选择总时延小的用户逐一上传，直到满足所需计算的样本量 B 为止。

5) 文献[18]策略。该文献采用 OFDMA 的接入方式，通过约束用户 m 的上传时间小于给定的阈值 $\bar{\tau}_m$，设计用户选择方案与带宽分配方案以最小化单轮的全局能量消耗。用户选择方案为

$$\beta_m=\min\left\{\max\left\{\frac{\mu_m W\bar{\tau}_m}{Q}\log\left(\frac{\gamma_m^t}{\ln 2\,p_m Q}\right),0\right\},1\right\} \tag{33}$$

其中，β_m 表示用户 m 是否参与当前轮调度，β_m=0 表示不参与，β_m=1 表示参与；μ_m 表示带宽分配比率；p_m 表示用户 m 单位带宽的传输功率，单位为 watt/Hz。

6.1.1　仿真设置

本仿真实验使用 64 bit Intel(R) Core(TM) i5-4460@3.20 GHz 处理器，运行内存大小为 12 GHz。用户数目 $M=100$，传输带宽为 500 kHz，发送信噪比 γ_m^t =5 dB，$p_m=2\times10^{-6}$ watt/Hz，不考虑大尺度衰落对用户的上行传输带来的影响，每个用户的上行信道随时间变化服从均值为 1 的瑞利分布。默认情况下，用户的计算能力服从 (100,900) 的均匀分布，且不随时间变化。

首先，以 IID 的方式随机将 MNIST 和 CIFAR-10 数据集分配给所有用户，其中，MNIST 下每个用户 600 个训练样本，CIFAR-10 下每个用户 500 个训练样本。

6.1.2　仿真结果

基于 5.2 节关于最优批大小的探讨，本节通过将理论分析与实验仿真相结合，验证本文基于通信与计算融合的用户调度策略下，能达到目标性能所需的总批大小与系统总时延的关系。以 MNIST 数据集为例，主要测试了 2 个性能目标，即测试精度为 80%和 90%，以及理论趋势的结果。

图 4 显示了理论趋势与不同测试精度的批大小与总时延的关系。随着精度的增加，系统总时延不断增加。对于同一精度目标，虽然总时延随着批大小的增加存在波动性，但是总体变化符合理论的结果，呈现先减小后增加的趋势。结合理论分析的结果，初始的总时延急速减小是由于批过小，从而导致模型梯度分散值很大，收敛速度很慢甚至对于过高的精度而言难以进行收敛。由于批大小的增加，需求的用户调度数目增加，而这种增加带来的通信时延的增加逐渐超过系统性能的提升，致使系统总时延随着批大小先减后增，从而存在最优的批大小选择。

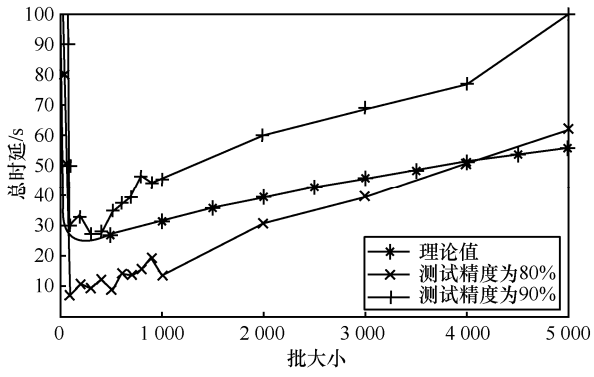


图 4　批大小与总时延的关系

使用卷积神经网络模型，学习率 $\eta=0.01$。在 MNIST 数据集下，模型组成为 2 个卷积层和 2 个全连接层。在 CIFAR-10 数据集下，模型包括 2 个卷积层、一个池化层和 3 个全连接层。不同于 MNIST 数据集，在该试验下，用户的计算能力服从 (50,500) 的均匀分布。

图 5 给出了 MNIST 数据集下 $B=200$ 时，不同调度策略的性能对比。需要注意的是，文献[13,18]策略不考虑用户的上传顺序对全局计算量的影响，比例公平调度不考虑用户的计算能力，随机调度和轮询调度既不考虑信道状态也不考虑计算能力。由仿真结果可知，在测试精度为 80%时，本文所提算法在收敛速度方面，较比例公平调度与文献[13,18]提出的算法性能提升超过 30%，较随机调度和轮询调度性能提升近 50%。

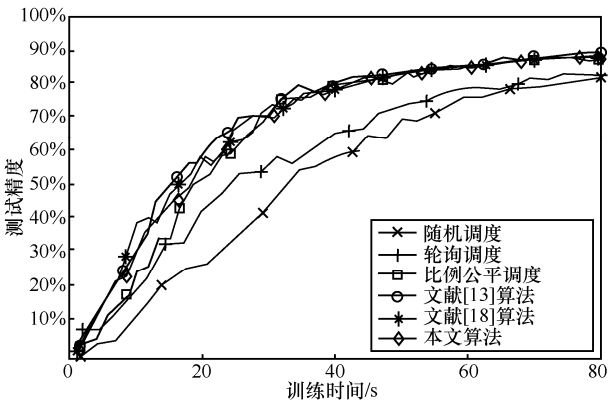


图 5　MNIST 数据集下的性能对比

图 6 给出了 CIFAR-10 数据集下不同算法的收敛性能。不同于 MNIST 数据集，CIFAR-10 数据集下的收敛相对较慢，在该场景下，设置 $B=20\,000$。由仿真结果可知，以测试精度为 70%为参考，本文算法在收敛速度方面，较比例公平调度与文献[13,18]算法性能提高近 25%，较随机调度和轮询调度性能提升近 50%。

6.2　Non-IID 场景仿真设置

在 Non-IID 场景中，使用 MNIST 数据集作为本文算法的试验验证数据集，与 IID 下 MNIST 数据集试验场景设置相同。研究用户数据不平衡度较低（100 个用户，每个用户拥有 2 类数据样本）和较高（100 个用户，每个用户拥有一类数据样本）2 种 Non-IID 场景下提出算法的性能。

6.2.1　不平衡度较高场景仿真结果

为了满足用户之间的不平衡度较高，设置用户 0～

用户 9 拥有类别为 0 的样本数据，用户 10～用户 19 拥有类别为 1 的样本数据，数量为 600，依次类推。同时在该实验下，设置 $B=5\,000$，$B^j=500$，$j=0,1,\cdots,9$，μ=100。仿真结果如图 7 所示。由于用户的不平衡度较高，虽然更新时延优先的调度策略一定程度上提高了当前调度用户的数据均衡度，但是由于用户之间不平衡度较高，调度的用户有很大概率不能包含所有类别的样本，导致仍然存在数据不均衡情况。同时，数据平衡优先的调度策略仍然可以满足当前轮调度的总数据样本每种类别满足大于或等于 500。因此，在用户数据不平衡度较高场景下，数据平衡优先调度性能优于更新时延优先调度。虽然随机调度和比例公平调度分别引入了随机性和公平性，由于无法保证每轮调度的数据样本保持均衡，因此模型始终处于发散状态。同样的，文献[13,18]算法也无法保证全局样本类别的均衡性，导致模型收敛性能较差。轮询调度由于在实验中，每轮只有一种样本被训练上传，因此，在该场景下模型不收敛。

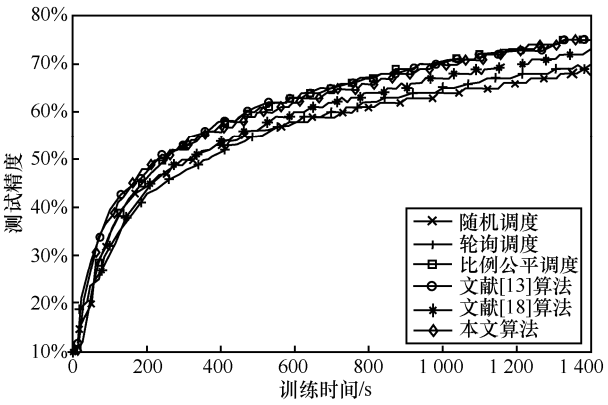


图 6　CIFAR-10 数据集下的性能对比

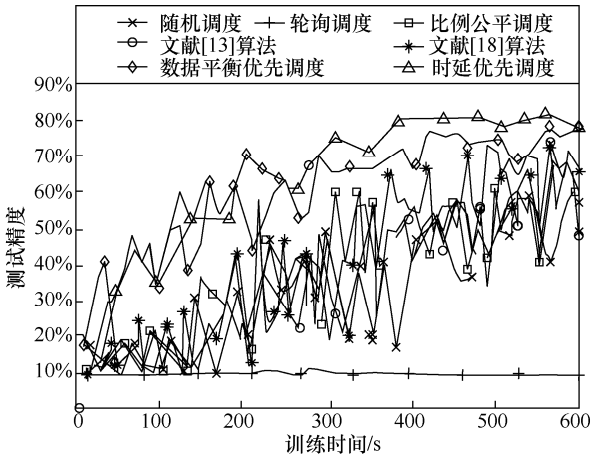


图 7　用户含有一种类别数据性能对比

6.2.2　不平衡度较低场景仿真结果

不平衡度较低场景仿真与不平衡度较高场景仿真的设置相同，不同的是在该场景下，设置用户 0～用户 9 和用户 50～用户 59 拥有 0 和 1 这 2 种样本，用户 10～用户 19 和用户 60～用户 69 拥有 2 和 3 这 2 种样本，依次类推。

如图 8 所示，由于用户之间数据的不平衡度降低，每轮调度的用户包含所有类别样本的概率大大增加，因此更新时延优先调度策略不仅能在时间上保证最优。同时，由于包含所有类别样本的概率增加，使当前轮调度用户的数据均衡度提升，因此在收敛性能上有了明显的提升。数据平衡调度策略虽然可以保证调度用户的调度数据集均衡，但是由 4.2 节的分析可知，其在时延上的性能下降导致其模型收敛速率小于更新时延优先调度策略。由于用户数据集的不均衡度降低，随机调度和比例公平调度以及文献[13,18]算法的性能也有所提升，但是依然存在模型发散和收敛速度慢的缺点。

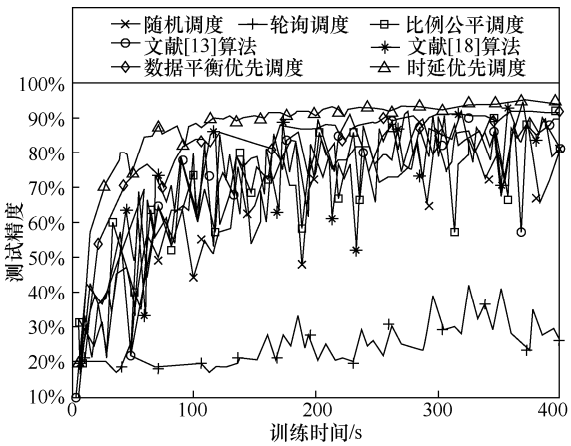


图 8　用户含有 2 种数据性能对比

7　结束语

本文首先在联邦学习用户 IID 数据分布情况下，提出了基于 TDMA 的用户调度策略，以降低系统时延。考虑到 Non-IID 场景下用户的样本类别不均衡问题，进一步提出了 2 种启发式用户调度策略，分别应对不平衡度高和不平衡度低的场景。此外，还进行了理论上的收敛分析，并基于此收敛边界和调度策略提出了全局最优策略。原模型的求解配合数值结果，保证了本文策略具有良好的收敛性能。

附录 1 引理 1 的证明

利用反证法证明。假设最小的系统时延满足 2 个相邻的调度用户上行传输时间点无间隙，则最优系统时延上原系统模型必然满足用户计算样本量等于 B，假设当前轮有 k 个用户参与调度，那么

$$A_1=\sum_{m=1}^{k}p_{\pi(m)}\left(c_{\pi(1)}+\sum_{i=1}^{m}\tau_{\pi(i)}\right)=B \tag{34}$$

假设在用户 $\pi(i-1)$ 与 $\pi(i)$（$\pi(i)\neq\pi(1)$）之间存在一个长度为 $\tau_{\text{gap}}>0$ 的时间间隙，使该情况下全局计算样本量为

$$\begin{aligned}A_2=&\sum_{m=1}^{i-1}p_{\pi(m)}(c_{\pi(1)}^{'}+\tau_{\pi(1)}+\tau_{\pi(2)}+\ldots+\tau_{\pi(m-1)})+\\&\sum_{m=i}^{k}p_{\pi(m)}(c_{\pi(1)}^{'}+\tau_{\pi(1)}+\tau_{\pi(2)}+\ldots+\tau_{\pi(m-1)}+\tau_{\text{gap}})\end{aligned} \tag{35}$$

其中，$c_{\pi(1)}^{'}$ 为引入 τ_{gap} 后用户 $\pi(1)$ 新的计算时间，则

$$A_2<\sum_{m=1}^{k}p_{\pi(m)}(c_{\pi(1)}^{'}+\tau_{\pi(1)}+\tau_{\pi(2)}+\cdots+\tau_{\pi(m-1)}+\tau_{\text{gap}})=A_3 \tag{36}$$

令 $A_3=B$，因为要满足约束条件式(9a)，则必然有

$$\begin{aligned}&c_{\pi(1)}^{'}+\tau_{\text{gap}}=c_{\pi(1)}\\&c_{\pi(1)}^{'}+\tau_{\pi(1)}+\tau_{\text{gap}}=c_{\pi(2)}\\&\quad\vdots\\&c_{\pi(1)}^{'}+\tau_{\pi(1)}+\tau_{\pi(2)}+\ldots+\tau_{\pi(k-1)}+\tau_{\text{gap}}=c_{\pi(k)}\end{aligned} \tag{37}$$

则 $A_3=A_1$，因此当 $A_1=B$ 时必然有 $A_2<B$。同理，如果 $\pi(i)=\pi(1)$，必然有 $\tau_{\text{gap}}=0$。

证毕。

附录 2 引理 2 的证明

假设当前轮有 k 个用户参与调度，最优的有序集为 $\vec{\pi}^{t}$，且满足条件式(10)。如果随机交换 $\vec{\pi}^{t}$ 中用户 $\pi(a)$ 和 $\pi(b)$ 的上传顺序，假设原始上传顺序用户 $\pi(a)$ 先于 $\pi(b)$，即 $a<b$。基于引理 1，交换上传顺序后不影响用户 $\pi(a)$ 之前和用户 $\pi(b)$ 之后的用户上传顺序以及它们各自训练的样本数量，系统总的更新时延也不受影响。因此，只需证明交换上传顺序后，对 $\pi(a)$、$\pi(b)$ 以及它们之间的用户所训练的全部样本数量的影响。分别定义 B_{before} 和 B_{after} 为交换上传顺序前和交换上传顺序后这些用户所训练的全部样本数量。注意到，交换顺序后，用户 $\pi(b)$ 的实际计算时间等于交换前用户 $\pi(a)$ 的计算时间，即 $c_{\pi(a)}$，则

$$B_{\text{before}}=p_{\pi(a)}c_{\pi(a)}+\sum_{m=a+1}^{b}p_{\pi(m)}\left(c_{\pi(a)}+\sum_{i=a}^{m-1}\tau_{\pi(i)}\right) \tag{38}$$

$$\begin{aligned}B_{\text{after}}=&p_{\pi(b)}c_{\pi(a)}+\sum_{m=a+1}^{b-1}p_{\pi(m)}\Big(c_{\pi(a)}+\tau_{\pi(b)}+\\&\sum_{i=a+1}^{m-1}\tau_{\pi(i)}\Big)+p_{\pi(a)}\left(c_{\pi(a)}+\tau_{\pi(b)}+\sum_{i=a+1}^{b-1}\tau_{\pi(i)}\right)\end{aligned} \tag{39}$$

$$\begin{aligned}B_{\text{before}}-B_{\text{after}}=&p_{\pi(b)}\left(\sum_{m=a}^{b-1}\tau_{\pi(m)}\right)+\\&\sum_{m=a+1}^{b-1}p_{\pi(m)}(\tau_{\pi(a)}-\tau_{\pi(b)})-p_{\pi(a)}\left(\sum_{m=a+1}^{b}\tau_{\pi(m)}\right)\end{aligned} \tag{40}$$

因为 $\lambda_{\pi(a)}\leqslant\lambda_{\pi(b)}$，假设其他用户仍按重要度排序，则有

$$\frac{p_{\pi(b)}}{\tau_{\pi(b)}}\geqslant\ldots\geqslant\frac{p_{\pi(b-j)}}{\tau_{\pi(b-j)}}\geqslant\ldots\geqslant\frac{p_{\pi(a)}}{\tau_{\pi(a)}} \tag{41}$$

其中，$a<b-j<b$，则化简式(40)为

$$p_{\pi(b-j)}(\tau_{\pi(a)}-\tau_{\pi(b)})+p_{\pi(b)}\tau_{\pi(b-j)}-p_{\pi(a)}\tau_{\pi(b-j)}\geqslant 0 \tag{42}$$

则

$$B_{\text{before}}-B_{\text{after}}\geqslant p_{\pi(b)}\tau_{\pi(a)}-p_{\pi(a)}\tau_{\pi(b)}\geqslant 0 \tag{43}$$

这意味着当顺序交换后，重要度较大的用户先于重要度较小的用户上传梯度，导致系统在相同时延下所能计算的样本数量变小。

证毕。

参考文献：

[1] HU Y C, PATEL M, SABELLA D, et al. Mobile edge computing—a key technology towards 5G[J]. ETSI White Paper, 2015, 11(11): 1-16.

[2] PARK J, SAMARAKOON S, BENNIS M, et al. Wireless network intelligence at the edge[J]. Proceedings of the IEEE, 2019, 107(11): 2204-2239.

[3] ZHOU Z, CHEN X, LI E, et al. Edge intelligence: paving the last mile of artificial intelligence with edge computing[J]. Proceedings of the IEEE, 2019, 107(8): 1738-1762.

[4] TAO M X, HUANG K B. Editorial: special topic on machine learning at network edges[J]. ZTE Communications, 2020, 18(2): 1, 30.

[5] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[C]//2017 Artificial Intelligence and Statistics. Saarland: DBLP, 2017: 1273-1282.

[6] ALISTARH D, GRUBIC D, LI J, et al. QSGD: communication-efficient SGD via randomized quantization and encoding[J]. Advances in Neural Information Processing Systems, 2018, 3: 1710-1721.

[7] AJI A F, HEAFIELD K. Sparse communication for distributed gradient descent[J]. arXiv Preprint, arXiv: 1704.05021, 2017.

[8] ZHU G X, WANG Y, HUANG K B. Broadband analog aggregation for low-latency federated edge learning[J]. IEEE Transactions on Wireless Communications, 2020, 19(1): 491-506.

[9] MOHAMMADI A M, GÜNDÜZ D. Machine learning at the wireless

edge: distributed stochastic gradient descent over-the-air[J]. IEEE Transactions on Signal Processing, 2020, 68: 2155-2169.

[10] YANG K, JIANG T, SHI Y M, et al. Federated learning via over-the-air computation[J]. IEEE Transactions on Wireless Communications, 2020, 19(3): 2022-2035.

[11] ZHANG N F, TAO M X. Gradient statistics aware power control for over-the-air federated learning[J]. arXiv Preprint, arXiv: 2003.02089, 2020.

[12] YANG H H, ARAFA A, QUEK T Q S, et al. Age-based scheduling policy for federated learning in mobile edge networks[C]//2020 IEEE International Conference on Acoustics. Piscataway: IEEE Press, 2020: 8743-8747.

[13] NISHIO T, YONETANI R. Client selection for federated learning with heterogeneous resources in mobile edge[C]//2019 IEEE International Conference on Communications. Piscataway: IEEE Press, 2019: 1-7.

[14] YANG H H, LIU Z Z, QUEK T Q S, et al. Scheduling policies for federated learning in wireless networks[J]. IEEE Transactions on Communications, 2020, 68(1): 317-333.

[15] REN J K, HE Y H, WEN D Z, et al. Scheduling for cellular federated edge learning with importance and channel awareness[J]. IEEE Transactions on Wireless Communications, 2020, 19(11): 7690-7703.

[16] MA X, SUN H, HU R Q. Scheduling policy and power allocation for federated learning in NOMA based MEC[J]. arXiv Preprint, arXiv: 2006.13044, 2020.

[17] AMIRIA M M, GÜNDÜZB D, KULKARNI S R, et al. Convergence of update aware device scheduling for federated learning at the wireless edge[J]. arXiv Preprint, arXiv: 2001.10402, 2020.

[18] ZENG Q S, DU Y Q, HUANG K B, et al. Energy-efficient radio resource allocation for federated edge learning[C]//2020 IEEE International Conference on Communications Workshops. Piscataway: IEEE Press, 2020: 1-6.

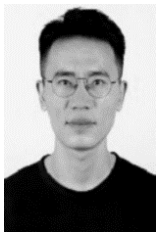
[19] CHEN M Z, YANG Z H, SAAD W, et al. A joint learning and communications framework for federated learning over wireless networks[J]. IEEE Transactions on Wireless Communications, 2021, 20(1): 269-283.

[20] SHI W Q, ZHOU S, NIU Z S, et al. Joint device scheduling and resource allocation for latency constrained wireless federated learning[J]. IEEE Transactions on Wireless Communications, 2021, 20(1): 453-467.

[21] DEKEL O, GILAD-BACHRACH R, SHAMIR O, et al. Optimal distributed online prediction using mini-batches[J]. The Journal of Machine Learning Research, 2012, 13: 165-202.

[作者简介]

陶梅霞（1978− ），女，江西九江人，博士，上海交通大学教授，主要研究方向为无线缓存、边缘计算、资源分配等。



王栋（1992− ），男，陕西渭南人，上海交通大学博士生，主要研究方向为联邦学习、边缘计算和无线通信网络等。

孙瑞（1996− ），女，江苏盐城人，上海交通大学硕士生，主要研究方向为联邦学习、移动边缘计算、分布式计算等。

张乃夫（1993− ），男，黑龙江哈尔滨人，上海交通大学博士生，主要研究方向为边缘学习、联邦学习等。