边缘计算网络中区块链赋能的异步联邦学习算法

黄晓舸* 邓雪松 陈前斌 张 杰
(重庆邮电大学通信与信息工程学院 重庆 400065)

摘 要：由于数据量激增而引起的信息爆炸使得传统集中式云计算不堪重负，边缘计算网络(ECN)被提出以减轻云服务器的负担。此外，在ECN中启用联邦学习(FL)，可以实现数据本地化处理，从而有效解决协同学习中边缘节点(ENs)的数据安全问题。然而在传统FL架构中，中央服务器容易受到单点攻击，导致系统性能下降，甚至任务失败。本文在ECN场景下，提出基于区块链技术的异步FL算法(AFLChain)，该算法基于ENs算力动态分配训练任务，以提高学习效率。此外，基于ENs算力、模型训练进度以及历史信誉值，引入熵权信誉机制评估ENs积极性并对其分级，淘汰低质EN以进一步提高AFLChain的性能。最后，提出基于次梯度的最优资源分配(SORA)算法，通过联合优化传输功率和计算资源分配以最小化整体网络延迟。仿真结果展示了AFLChain的模型训练效率以及SORA算法的收敛情况，证明了所提算法的有效性。

关键词：异步联邦学习；区块链；资源分配；边缘计算网络
中图分类号：TN92 **文献标识码**：A **文章编号**：1009-5896(2024)01-0195-09
DOI: 10.11999/JEIT221517

Asynchronous Federated Learning via Blockchain in Edge Computing Networks

HUANG Xiaoge DENG Xuesong CHEN Qianbin ZHANG Jie
(*College of Communication and Information Technology, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*)

Abstract: Because of the information explosion caused by the surge of data, traditional centralized cloud computing is overwhelmed, Edge Computing Network (ECN) is proposed to alleviate the burden on cloud servers. In contrast, by permitting Federated Learning (FL) in the ECN, data localization processing could be realized to successfully address the data security problem of Edge Nodes (ENs) in collaborative learning. However, traditional FL exposes the central server to single-point attacks, resulting in system performance degradation or even task failure. In this paper, we propose Asynchronous Federated Learning based on Blockchain technology (AFLChain) in the ECN that can dynamically assign learning tasks to ENs based on their computing capabilities to boost learning efficiency. In addition, based on the computing capability of ENs, model training progress and historical reputation, the entropy weight reputation mechanism is implemented to assess and rank the enthusiasm of ENs, eliminating low quality ENs to further improve the performance of the AFLChain. Finally, the Subgradient based Optimal Resource Allocation (SORA) algorithm is proposed to reduce network latency by optimizing transmission power and computing resource allocation simultaneously. The simulation results demonstrate the model training efficiency of the AFLChain and the convergence of the SORA algorithm and the efficacy of the proposed algorithms.

Key words: Asynchronous federated learning; Blockchain; Resource allocation; Edge Computing Network (ECN)

收稿日期：2022-12-06；改回日期：2023-05-17；网络出版：2023-05-24
*通信作者： 黄晓舸 huangxg@cqupt.edu.cn
基金项目：国家自然科学基金(61831002)，重庆市科委重庆市基础研究与前沿探索项目(cstc2018jcyjAx0383)
Foundation Items: The National Natural Science Foundation of China (61831002), Innovation Project of the Common Key Technology of Chongqing Science and Technology Industry (cstc2018jcyjAx0383)

1 引言

据GSMA预测，2025 年全球物联网设备将达250 亿台，海量数据处理请求成为亟待解决的问题[1]。机器学习(Machine Learning, ML)可从用户环境等多样特征中挖掘出复杂的“模式和见解”[2]。ML的复杂计算推动了边缘计算网络(Edge Computing Network, ECN)的发展，为减轻云端负荷并减少任务反馈延迟，将任务卸载到边缘节点 (Edge Nodes, ENs) 计算[3]。此外，用户对隐私保护的关注日益提升，使得数据本地化处理成为当前热门研究方向[4]。

联邦学习 (Federated Learning, FL)允许多个设备在保证用户隐私的前提下共享模型参数[5]，由此受到多方关注，其性能优化方案层出不穷。文献[6]引入信誉机制，根据时间序列来预测信誉值，筛选高质量节点参与FL任务，提升训练性能。文献[7]提出一种异步聚合FL方式，可在收到模型更新后可立即聚合，无需等待所有节点完成本地训练，提高训练效率。但是，设备上传模型的时间和质量不确定，异步FL性能不稳定。为克服性能回退，文献[8]提出一种半异步FL，可自适应调整本地迭代数以解决掉队者问题。考虑到异步FL的复杂性，一些学者在网络结构上寻求突破。文献[9]发现神经网络深层参数的更新频率需求低于浅层，基于此特性，提出了时间加权的异步FL，使用不同频率更新深浅层模型参数，以减少数据传输量并提高学习效率。

传统FL的服务器容易遭到单点攻击，其数据安全面临巨大挑战。区块链技术包含分布式存储、共识机制等[10]，可有效提高FL的安全性。文献[11]旨在去除传统FL的中央服务器，提出由本地链和全局链构成的双层区块链。本地链按时间顺序记录本地模型，形成本地设备的长期信誉值。全局链被划分为逻辑隔离的FL任务链，以提高效率和可信性。文献[12]使用区块链记录FL中的高质量节点，使用三重主观逻辑模型计算节点的信誉值，作为节点筛选指标。此外，设计了一种质量证明机制，提高区块链共识效率。文献[13]提出了一个基于区块链的异步FL框架，使用动态比例因子提升FL训练效率，并保证训练的有效性。同时，提出基于委员会的共识机制，以尽可能小的时间成本保证可靠性。

为提高FL的训练效率与安全性，本文提出基于区块链的异步联邦学习(Asynchronous Federated Learning based on Blockchain, AFLChain)算法。本文主要贡献如下：

首先， AFLChain将网络中的任务发布者和响应者定义为主节点 (Primary Node, PN) 和次节点 (Secondary Nodes, SNs)。为提高FL训练效率，SNs根据算力进行不同轮次的本地训练。PN持续本地训练至全局聚合，与SN构成异步FL。

其次，为解决FL中央服务器易受单点攻击的问题，使用区块链技术保证数据安全。区块链由PN和SNs领导者交替上传的两种区块组成，确保数据可追溯性。SNs领导者由熵权信誉机制选出。

此外，本文提出一种基于次梯度的最优资源分配(Subgradient based Optimal Resource Allocation, SORA)算法，通过联合优化计算和通信资源提升AFLChain的性能，最小化网络总延迟。

最后，广泛的仿真验证了所提算法的有效性。

2 系统建模及分析

2.1 系统模型

如图1，本文在ECN中实现FL架构，EN(如智能车间、智能楼宇等)具有足够的本地数据和算力，可支撑训练任务和共识算法。发布任务的EN为主节点(Primary Node, PN)，参与任务的 EN 为辅助节点(Secondary Nodes, SNs)。PN为可信节点，负责模型聚合，SN负责本地模型训练及上传。

在区块链网络中，所有节点共同维护一条联盟区块链，由PN和SN领导者交替上传的区块组成。共识机制也可用权益证明(Proof-of-Stake, PoS)等，但需要额外方案防止分叉。考虑到联盟链中节点较少，基于投票机制的实用拜占庭容错(Practical Byzantine Fault Tolerance, PBFT)机制更高效。

2.2 工作流程

设AFLChain中有1个PN和K个SNs。图2显示了一个epoch的工作流程，包含以下步骤：

步骤1 SNs从区块链网络中获取全局模型。

步骤2 在epoch h，第i轮本地迭代中，SN k使用本地数据集$(\mathcal{X}_k,\mathcal{Y}_k)$的部分样本计算交叉损失熵作为损失函数，以获取模型梯度$\mathfrak{g}(\omega_k^{h,i})$：

$$\mathfrak{g}(\omega_k^{h,i})=\frac{1}{b}\sum_{(x,y)\in(\mathcal{X}_k^b,\mathcal{Y}_k^b)}\sum_{\theta=1}^{\Theta}\mathcal{P}(y=\theta)\cdot[\nabla_\omega\log\mathcal{F}_\theta(x,\omega_k^{h,i})] \tag{1}$$

其中，b表示数据集样本大小，θ表示数据类别号，$(\mathcal{X},\mathcal{Y})$分别是数据的样本特征和标签。$\mathcal{F}_\theta$是模型权重$\omega$的函数，表示输出为类别$\theta$的概率。

本文采用随机梯度下降(Stochastic Gradient Descent, SGD)优化算法更新本地模型$\omega_k^{h,i}$：

$$\omega_k^{h,i+1}=\omega_k^{h,i}-\eta\cdot\mathfrak{g}(\omega_k^{h,i}) \tag{2}$$

其中，η表示本地学习率。

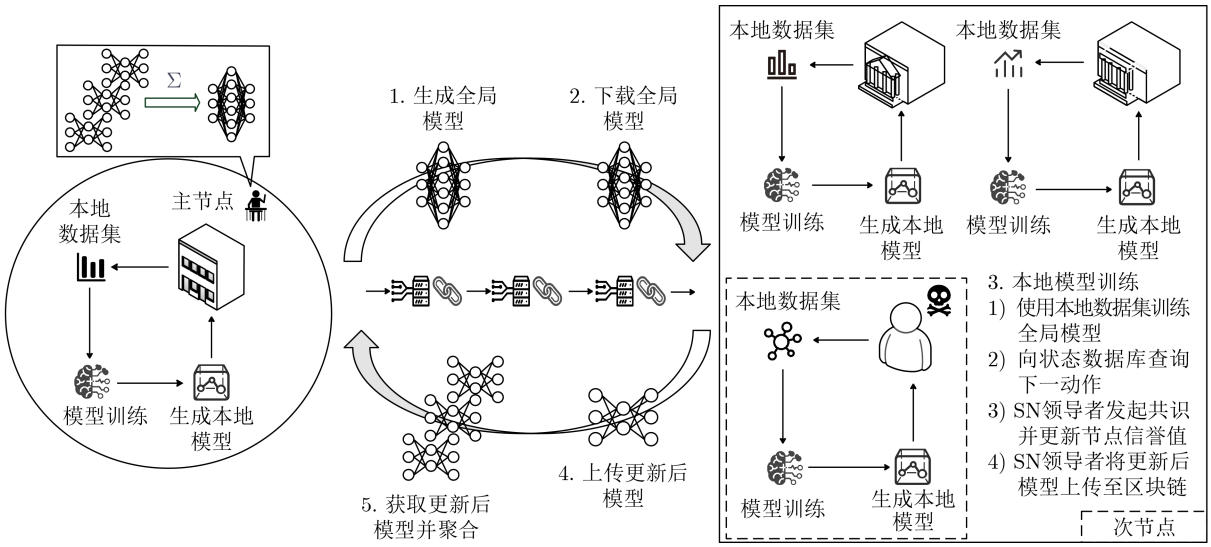


图 1 网络结构图

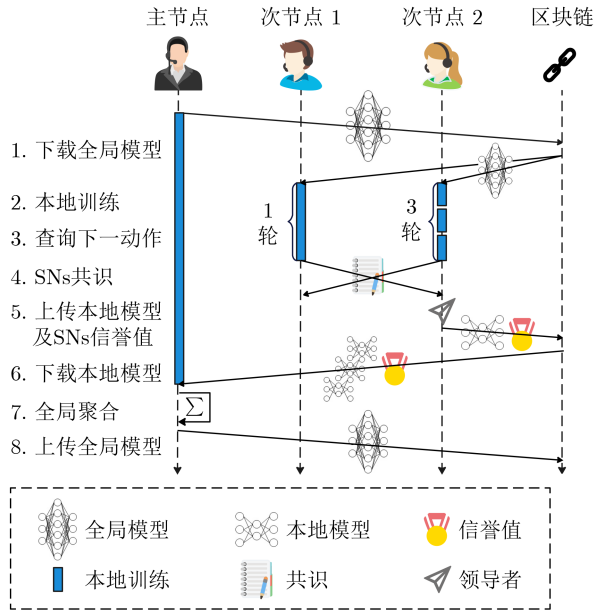


图 2 一个epoch的工作流程

步骤3　SN发送状态信息到状态数据库，查询下一个动作。如果返回操作是TRAIN，则继续进行下一轮本地训练，否则进入共识过程。

步骤4　所有SNs进入共识后，SNs领导者开启共识。共识达成后，SN k基于模型更新$\omega_k^{h,i}$，本地训练轮数以及历史信誉值更新信誉值。

步骤5　SN领导者将SNs的信誉值与模型打包成块并上传至区块链。仅SNs领导者有权上传区块，所以分叉现象将不会发生。

步骤6　智能合约(Smart Contract, SC)通知PN从区块链下载更新后的SNs模型和信誉值。

步骤7　当PN接收到聚合信号，它将基于PN和SNs模型，采用加权平均策略进行全局聚合。

$$\omega^h = \omega^{h-1} + \lambda \cdot \sum_{k=1}^{K+1} \frac{n_k}{n}(\omega_k^h - \omega^{h-1}) \tag{3}$$

其中，ω_k^h为SN k在epoch h的本地模型，λ是全局学习率，n_k表示数据集大小，且$n = \sum_{k=1}^{K+1} n_k$。

步骤8　PN将全局模型上传至区块链网络。

重复上述步骤至模型收敛或目标精度达成。

2.3　信誉机制

AFLChain依赖于多个SNs的共同维护，为保障模型质量，需对SNs进行评估。与主观加权方法相比，根据信息差异确定权重的熵权法具有更好的客观性。本文中，信誉值由模型训练进度、本地训练轮数和历史信誉值决定。信誉值大于上分位点th_{upper}的SN为候选SN，信誉值最高的候选SN为领导者。信誉值小于下分位点th_{low}的SN为潜在SN，不能参与当前FL任务，其余为普通SN。本文使用余弦相似度表示模型间差异，以保证高维向量计算的准确性[14]。epoch h中SN k模型ω_k^h与epoch $h-1$中全局模型ω^{h-1}的余弦相似度为

$$\text{sim}(\omega_k^h, \omega^{h-1}) = \frac{\omega_k^h \cdot \omega^{h-1}}{\|\omega_k^h\| \times \|\omega^{h-1}\|} \tag{4}$$

其中，$\|(\omega, \omega')\| = \sqrt{\omega^2 + \omega'^2}$表示向量的$\mathcal{L}_2$范数。

$\text{sim}(\omega_k^h, \omega^{h-1})$随着模型间的差异增大而降低。由于神经网络结构复杂，为降低复杂度，采用对输出层影响最直接的倒数第2层参数计算余弦相似度。

对于SN k, $s_k^{h,j}$表示epoch h中指标j的标准化值，均采用正标准化，其占比为

$$\phi_k^{h,j} = \frac{s_k^{h,j}}{\sum_{k=1}^{K} s_k^{h,j}},\ k = \{1, 2, \cdots, K\},\ j = \{1, 2, 3\} \tag{5}$$

其中，$j=1,2,3$分别表示SN k在epoch $h-1$的信誉值，本地训练轮数r_k和余弦相似度$\mathrm{sim}(\omega_k^h,\omega^h)$。

指标j的熵权值为

$$w^{h,j}=\frac{1-e^{h,j}}{3-\sum_{j=1}^{3}e^{h,j}},\sum_{j=1}^{3}w^{h,j}=1 \tag{6}$$

其中，$e^{h,j}=-\frac{1}{\ln(K)}\sum_{k=1}^{K}\phi_k^{h,j}\ln(\phi_k^{h,j})$。指标$j$的熵权值越大，表示其对信誉值的贡献越大。

基于此，SN k在epoch h的信誉值表示为

$$\mathcal{R}_k^h=\sum_{j=1}^{3}w^{h,j}\times s_k^{h,j}\times 100 \tag{7}$$

3 异步联邦学习架构

为缓解同步FL训练效率低下的问题，本文提出异步FL算法，如图3，通过包含SNs状态信息的状态数据库调整SNs本地训练轮数，提高训练效率，其功能可在SC中实现以克服安全问题。

SN k状态信息为$(k,h_k,r_k,T_k^{\mathrm{cmp}},T_k^{\mathrm{qry}},T_k)$，$h_k$为epoch计数，$r_k$为本地训练轮数，$T_k^{\mathrm{cmp}}$为本地训练时间，$T_k^{\mathrm{qry}}$为查询时间，$T_k$为发信时间戳。

SN k的查询与计算延迟之和$d_k=T_k^{\mathrm{qry}}+T_k^{\mathrm{com}}$，遍历状态数据库，可找出最低算力SN ξ，

$$\xi=\arg\max_k\ d_k,k\in[1,K] \tag{8}$$

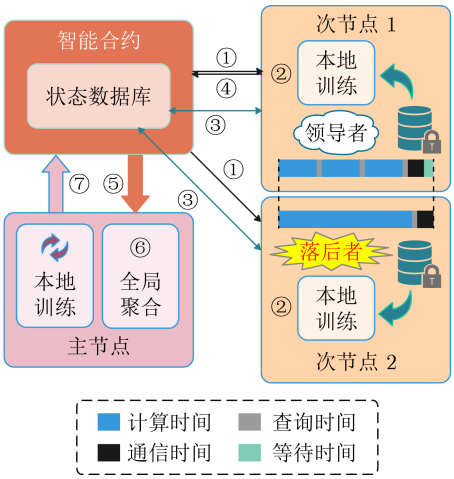
基于此,可计算最低算力SN完成训练时间$\hat{t}_\xi$，

$$\hat{t}_\xi=T_\xi+d_\xi \tag{9}$$

最后，由式(10)确认SN k是否进行继续训练，

$$t_c+d_k\le\hat{t}_\xi \tag{10}$$

其中，t_c是状态服务器的当前时间戳。



① SN下载全局模型；② SNs本地训练；③ SNs获取下一动作；④ 领导者收集SNs模型并上传至区块链；⑤⑥ PN下载SNs模型进行聚合；⑦ PN上传全局模型至区块链

图 3 异步FL概述

SNs 的时间开销如图3所示。令$\rho=T_{\max}^{\mathrm{cmp}}/T_{\min}^{\mathrm{cmp}}$表示最低和最高算力SN之间的算力比，其中，

$$T_{\max}^{\mathrm{cmp}}=\max\{T_k^{\mathrm{cmp}},1\le k\le K\} \tag{11}$$

$$T_{\min}^{\mathrm{cmp}}=\min\{T_k^{\mathrm{cmp}},1\le k\le K\} \tag{12}$$

SN k的等待时间可表示为

$$d_k^{\mathrm{wait}}=T_\xi^{\mathrm{cmp}}+T_\xi^{\mathrm{qry}}+T_\xi^{\mathrm{com}}-r_k(T_k^{\mathrm{cmp}}+T_k^{\mathrm{qry}})-T_k^{\mathrm{com}} \tag{13}$$

其中，T^{qry}为查询时间，相对于其他时延可忽略不计，则$T_\xi^{\mathrm{qry}}=T_k^{\mathrm{qry}}=0$。为优化本地训练轮数$r_k$以最小化整体等待时间，建立优化目标如下：

$$\arg\min_{r_k}\ d^{\mathrm{wait}}=\sum_{k=1}^{K}d_k^{\mathrm{wait}}(r_k),k\ne\xi \tag{14}$$

若式(10)成立，状态数据库返回$a_k=1$。否则，剩余等待时间不够完成一轮训练，SNs领导者开启共识。flag为SC发送至PN的信号，flag = 1表示聚合，否则PN持续训练。细节如算法1所述。

4 问题建模

4.1 计算模型

SN k的本地训练时间表示为

$$T_k^{\mathrm{cmp}}=\frac{c_kD}{f_k^{\mathrm{cmp}}} \tag{15}$$

其中，c_k表示SN k训练一个样本所需CPU周期，D为样本个数，f_k^{cmp}表示SN k可提供CPU频率。

算法1 状态数据库决策流程

输入：SN k状态信息，状态表 $[(k,h_\kappa,r_\kappa,T_\kappa^{\mathrm{cmp}},T_k^{\mathrm{qry}},T_\kappa,a_\kappa)]_{\forall k\in[1,K]}$

输出：下一动作a_k

更新状态表中SN k的状态信息

通过式(8)找到最低算力SN

情况1：SN k刚进入一轮本地训练，或SN ξ还未完成训练。

if $r_k=1$ or $h_k>h_\xi$

$r_k\leftarrow r_k+1$

返回$a_k=1$

end if

$t_c=\mathrm{CurrentTime}$

情况2：SN ξ已完成训练，或剩余等待时间不够完成一轮训练。

if $k=\xi$ or $a_\xi=0$ or 式(10)不成立

更新状态表动作信息

返回$a_k=0$

end if

情况3：剩余等待时间足以SN k完成一轮本地训练。

$r_k\leftarrow r_k+1$

返回$a_k=1$

SN k训练模型所需的CPU周期数表示为c_kD。因此，SN k在一次训练迭代中的能耗为

$$E_k^{\mathrm{cmp}}=\sum_{x=1}^{c_kD}\beta(f_k^{\mathrm{cmp}})^2=\beta c_kD(f_k^{\mathrm{cmp}})^2 \tag{16}$$

其中，β是SN k计算芯片组的有效电容系数。

4.2 通信模型

本场景中通信时延为共识和领导者块上传的时延。其中，共识过程分为块传播和块验证。

在块传播阶段， SNs向领导者l发送包含本地模型的块信息，传输速率(bit/s)可由式(17)计算：

$$\gamma_{k,l}=B\log_2\left(1+\frac{p_kg_{k,l}}{n_0}\right),\ l\neq k \tag{17}$$

其中，B表示带宽，p_k为SN k的传输功率，$g_{k,l}$为SN k到领导者l的信道增益，n_0为噪声功率。

因此，在共识过程中块传播时延可表示为

$$T^{\mathrm{pro}}=\frac{\delta_b}{\min\{\gamma_{k,l}\}},\ l\neq k \tag{18}$$

其中，δ_b表示块大小。

在块验证中，SN确认块信息，确认时延为

$$T^{\mathrm{ver}}=\frac{\varphi_b}{\min\{f_k^b\}} \tag{19}$$

其中，φ_b表示验证块所需的CPU周期数，f_k^b为SN k验证块的计算频率。

此外，领导者l的块上传时延可由式(20)计算：

$$T_l^{\mathrm{ul}}=\frac{\delta_b}{\gamma_l} \tag{20}$$

其中，γ_l为领导者l的传输速率，块上传能耗为

$$E_l^{\mathrm{ul}}=T_l^{\mathrm{ul}}p_l \tag{21}$$

4.3 优化问题

为使SNs的本地训练时延、共识时延和块上传时延之和最小化，将优化问题建模为

$$\begin{aligned}\min_{f_k^{\mathrm{cmp}},f_k^b,p_k,p_l}\ &\left(I\cdot r_k\frac{c_kD}{f_k^{\mathrm{cmp}}}+\frac{\delta_b}{B\log_2\left(1+\frac{p_kg_{k,l}}{n_0}\right)}\right.\\&\left.+\frac{\varphi_b}{f_k^b}+\frac{\delta_b}{B\log_2\left(1+\frac{p_lg_{l,p}}{n_0}\right)}\right)\\ \text{s.t. }\ &\mathrm{C1}:f_k^{\mathrm{cmp}}+f_k^b\le f_k^{\max},\\&\mathrm{C2}:\frac{\varphi_b}{f_k^b}\le T_{\mathrm{tol}},\\&\mathrm{C3}:\beta c_kD(f_k^{\mathrm{cmp}})^2\le\bar{E}_k^{\mathrm{cmp}},\\&\mathrm{C4}:T_l^{\mathrm{ul}}p_l\le\bar{E}_l^{\mathrm{ul}},\\&\mathrm{C5}:p_k^{\min}\le p_k\le p_k^{\max},\\&\mathrm{C6}:f_k^{\mathrm{cmp}}>0,f_k^b>0\end{aligned} \tag{22}$$

其中，C1为SN k计算资源约束；C2表示块验证时延不能超过最大可容忍延迟；C3和C4表示模型更新和块上传的能耗限制；C5为传输功率约束。

4.4 资源分配优化

本节提出SORA算法解决优化问题(22)，首先根据C4，最优块上传功率可表示为

$${p_l}^*=\frac{\bar{E}_l^{\mathrm{ul}}}{T_l^{\mathrm{ul}}} \tag{23}$$

由于$f_k^{\mathrm{cmp}},f_k^b,p_k$在目标函数和约束条件中都可以解耦，原优化问题被分解为以下两个子问题。

(1) 最优计算资源分配

将p_k视为常数并去除常量项，原问题简化为

$$\begin{aligned}&\mathrm{P1}:\arg\min_{f_k^{\mathrm{cmp}},f_k^b}\left(I\cdot r_k\frac{c_kD}{f_k^{\mathrm{cmp}}}+\frac{\varphi_b}{f_k^b}\right),\\&\text{s.t. }\ \mathrm{C1},\mathrm{C2},\mathrm{C3},\mathrm{C6}\end{aligned} \tag{24}$$

为降低复杂度，采用交替求解法获取最优解。

首先，在f_k^b固定的情况下，子问题P1是关于f_k^{cmp}的凸问题，采用拉格朗日法求得最优解$f_k^{\mathrm{cmp}*}$，

$$\begin{aligned}\mathcal{L}(f_k^{\mathrm{cmp}},\pi_1,\pi_2)=&I\cdot r_k\frac{c_kD}{f_k^{\mathrm{cmp}}}+\pi_1(f_k^{\mathrm{cmp}}+f_k^b-f_k^{\max})\\&+\pi_2(\beta c_kD(f_k^{\mathrm{cmp}})^2-\bar{E}_k^{\mathrm{cmp}})\end{aligned} \tag{25}$$

其中，π_1,π_2是C1, C3的拉格朗日乘子。由Karush Kuhn Tucker (KKT)条件，可得以下充要条件：

$$\begin{aligned}\frac{\partial\mathcal{L}(f_k^{\mathrm{cmp}},\pi_1,\pi_2)}{\partial f_k^{\mathrm{cmp}}}=&-I\cdot r_k\frac{c_kD}{(f_k^{\mathrm{cmp}})^2}+\pi_1\\&+2\pi_2\beta c_kDf_k^{\mathrm{cmp}}=0\end{aligned} \tag{26}$$

$$\pi_1(f_k^{\mathrm{cmp}}+f_k^b-f_k^{\max})=0,\pi_1\ge0 \tag{27}$$

$$\pi_2(\beta c_kD(f_k^{\mathrm{cmp}})^2-\bar{E}_k^{\mathrm{cmp}})=0,\pi_2\ge0 \tag{28}$$

$$0<f_k^{\mathrm{cmp}}<f_k^{\max} \tag{29}$$

等式(26)可转换为如下关于f_k^{cmp}的3次方程：

$$(f_k^{\mathrm{cmp}})^3+\frac{\pi_1}{2\pi_2\beta c_kD}(f_k^{\mathrm{cmp}})^2-\frac{I\cdot r_k}{2\pi_2\beta}=0 \tag{30}$$

为求解3次方程式(30)，引入盛金公式[15]。对3次方程$ax^3+cx^2+cx+d=0$，定义以下辅助变量：

$$\mathcal{A}=b^3-3ac,\mathcal{B}=bc-9ad,\mathcal{C}=c^2-3bd \tag{31}$$

结合式(30)与式(31)，可得如下判别式：

$$\Delta=\mathcal{B}^2-4\mathcal{A}\mathcal{C}=\frac{81I^2{r_k}^2}{4{\pi_2}^2\beta^2}-\frac{12I\cdot r_k{\pi_1}^4}{32{\pi_2}^5\beta^5{c_k}^4D^4} \tag{32}$$

由盛金公式可知，当$\Delta=0$时，该3次方程有3个实根，且其中两个实根相等。令$\Delta=0$，可得

$$I=\frac{{\pi_1}^4}{54{\pi_2}^3r_k\beta^3{c_k}^4D^4} \tag{33}$$

则等式(30)的3个实根表示为

$$x_1 = -\frac{b}{a} + \frac{\mathcal{B}}{\mathcal{A}} = \frac{72{\pi_2}^3 I r_k \beta^3 {c_k}^4 D^4 - {\pi_1}^4}{2{\pi_1}^3 \pi_2 \beta c_k D} \tag{34}$$

$$x_2 = x_3 = -\frac{\mathcal{B}}{2\mathcal{A}} = -\frac{18{\pi_2}^2 I r_k \beta^2 {c_k}^3 D^3}{{\pi_1}^3} \tag{35}$$

显然， $x_2 = x_3 < 0$, $f_k^{\mathrm{cmp}*}$在式(33)条件下为

$$f_k^{\mathrm{cmp}*} = x_1 = \frac{72{\pi_2}^3 I r_k \beta^3 {c_k}^4 D^4 - {\pi_1}^4}{2{\pi_1}^3 \pi_2 \beta c_k D} \tag{36}$$

π_1和π_2由拉格朗日对偶法求解，对偶函数为

$$\mathcal{F}(\pi_1, \pi_2) = \min_{\pi_1, \pi_2} \mathcal{L}(f_k^{\mathrm{cmp}}, \pi_1, \pi_2), \quad \text{s.t. C1} \tag{37}$$

因此，等式(25)的对偶问题可表示为

$$\arg\max_{\pi_1, \pi_2} \ \mathcal{F}(\pi_1, \pi_2) \quad \text{s.t. } \pi_1 > 0, \pi_2 > 0 \tag{38}$$

次梯度法可求解 (38)，π_1和π_2的更新方程为

$$\pi_1^{t+1} = \pi_1^t + \epsilon_1 (f_k^{\mathrm{cmp}} + f_k^b - f_k^{\max}) \tag{39}$$

$$\pi_2^{t+1} = \pi_2^t + \epsilon_2 (\beta c_k D (f_k^{\mathrm{cmp}})^2 - \bar{E}_k^{\mathrm{cmp}}) \tag{40}$$

其中，t是迭代因子，ϵ_1和ϵ_2分别是更新步长。

其次，基于$f_k^{\mathrm{cmp}*}$, f_k^b可由如下优化问题求解：

$$\arg\min_{f_k^b} \ \frac{\varphi_b}{f_k^b} \quad \text{s.t. C1, C2, C6} \tag{41}$$

由于式(41)是递减函数，f_k^b最大，函数值最小，

$$f_k^{b*} = f_k^{\max} - f_k^{\mathrm{cmp}*} \tag{42}$$

(2) 最优传输资源分配

基于$f_k^{\mathrm{cmp}*}$和f_k^{b*}，原优化问题(22)可以简化为

$$\mathrm{P2}: \arg\min_{p_k} \frac{\delta_b}{\min\left\{ B\log_2\left(1 + \frac{p_k g_{k,k'}}{n_0}\right)\right\}} \quad \text{s.t. C5} \tag{43}$$

P2是单调递减函数，在传输功率最大的情况下达到最小值，从而有

$$p_k{}^* = p_k^{\max} \tag{44}$$

算法2给出了SORA算法的细节，设定容忍阈值ϵ_3，问题式(38)的复杂度为$\mathcal{O}(\log_2(1/\epsilon_3))$，$K$个SN的复杂度表示为$\mathcal{O}(K\log_2(1/\epsilon_3))$，在迭代因子上限$t_{\max}$下，SORA算法的复杂度为$\mathcal{O}(Kt_{\max}\log_2(1/\epsilon_3))$。

5 仿真结果及分析

5.1 仿真场景及参数设置

本文使用Python和Pytorch评估AFLChain的性能，Matlab验证SORA的有效性，采用Mnist和FashionMnist训练卷积神经网络(Convolutional Neural Network, CNN)，Cifar10训练ResNet18。路径损耗模型采用3GPP，$140.7 + 36.7\lg(\varsigma)$，$\varsigma$为节点距离。默认算力比$\rho = 3$，其他设置见表1。

5.2 结果分析

图4展示了不同FL算法的性能。使用Mnist训练CNN，选取5个SNs参与任务。基线为单机训练CNN，比较算法为谷歌提出的Vanilla FL和文献[8]的ESync FL。由结果，非同步FL算法可以获得更高准确率和更快收敛速度。此外，与ESync相比，AFLChain可筛选高质量SN，以达到更好的性能。

如图5所示，本文比较了AFLChain分别采用异步和同步算法时的性能。在FashionMnist上，全局模型为CNN，两者准确率差异不明显，但异步算法能更快达到收敛。在更复杂的通用图片数据集

算法2 基于次梯度的最优资源分配算法(SORA)

输入：拉格朗日乘子更新步长(ϵ_1, ϵ_2)，拉格朗日乘子初始值(π_1^0, π_2^0)，最大容忍阈值ϵ_3，$t = 0$，迭代因子上限$t_{\max}$

输出：最优资源分配$(f_k^{\mathrm{cmp}*}, f_k^{b*}, {p_k}^*)$

```
while t < t_max do
    由式(36)和式(42) 分别得到 f_k^{cmp*} 和 f_k^{b*}
    由式(39)和式(40) 分别更新拉格朗日乘子 π1 和 π2
    if |π1^{t+1} − π1^t| < ε3 and |π2^{t+1} − π2^t| < ε3
        break
    else
        t = t + 1
    end if
end while
由式(44)得到 p_k*
```

表 1 仿真参数设置

参数	描述	数值
K	SN个数	30
H	epoch数量	30
B	带宽	1 MHz
δ_b	块大小	8 MB
n_0	噪声功率	–174 dBm/Hz
λ	全局学习率	1
η	本地学习率	0.001
b	数据抽样大小	32
D	训练样本大小	3 MB
c_k	SN k完成训练所需频率	20 cycles/bit
$f_k^{\max}$	SN k最大算力	0.2～1 GHz
p_k	SN k传输功率	40 W
$\mathrm{th}_{\mathrm{upper}}$	信誉值上分位点	50
$\mathrm{th}_{\mathrm{low}}$	信誉值下分位点	18

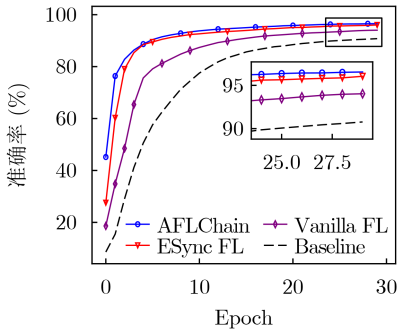


图 4 不同算法准确率与全局epoch的关系

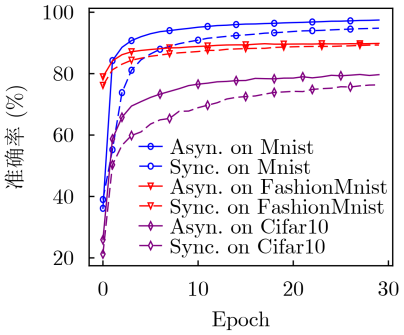


图 5 不同FL算法在不同数据集上的性能表现

Cifar10上，ResNet18被用作全局模型。相较同步算法，异步算法收敛速度更快，准确率更高。

图6展示了SN数量对算法性能的影响。设目标准确率为95%，达到该准确率的epoch数随着SN数量的增加而减少。当有6个SNs参与训练任务时，与5个SNs所需epoch数相当，因为新加入SN可能算力较低，降低平均算力水平。

图7对比了AFLChain在不同算力比ρ下的性能。随着ρ的增大，高算力SNs可进行更多轮次本地训练，对模型准确率提升的贡献更大。$\rho=1$时，即所有SNs只能完成一轮本地训练时，AFLChain性能等价于同步FL。此外，区块链技术可在不降低训练精度的同时，保证数据安全性和可追溯性。

图8描述了不同掉队者比例(低算力SN数量/参与训练SN数量)对算法性能的影响，8个SNs参与FL任务。模型准确率随低算力SN数量的增加而降低，因为低算力SN增多，本地训练轮数相应减少，导致准确率降低。但是，即使掉队者比例大于1/2时，AFLChain仍能达到95%以上的准确率。

图9对比5种场景到95%模型准确率的性能表现：(1)同步FL，$\rho=5$；(2) AFLChain，$\rho=5$；(3)同步FL，$\rho=2$；(4) AFLChain，$\rho=2$；(5)同步FL，$\rho=1$。$\rho=5$时，同步FL中高算力SN需要等待低算力SN，而AFLChain允许高算力SN在等待时间内持续训练，提高效率效率。$\rho=1$时，epoch

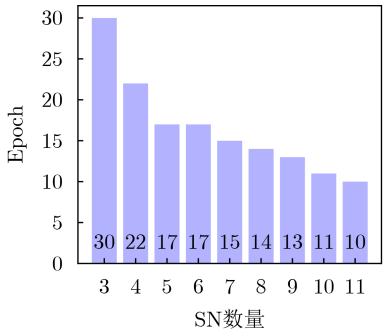


图 6 AFLChain在不同SN数量下的性能表现

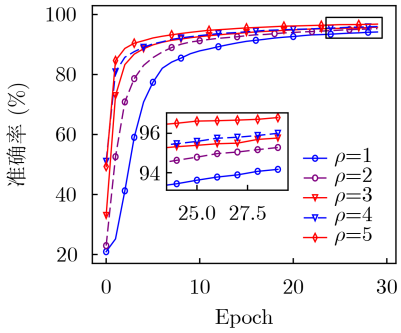


图 7 AFLChain在不同算力比时的性能表现

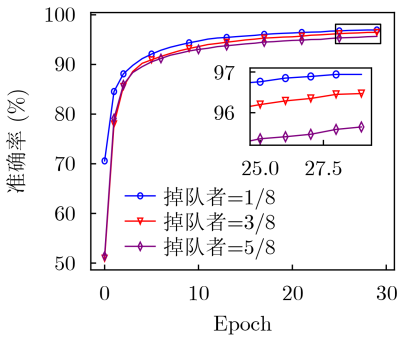


图 8 AFLChain在不同掉队者比例下的性能表现

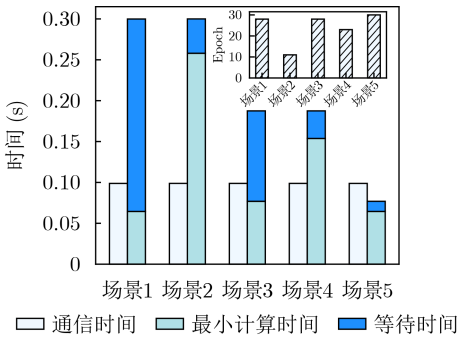


图 9 不同算法在不同场景下的时间开销

数显著增加，AFLChain等价于同步FL。综上，SN的算力比越高， AFLChain的性能增益越大。

图10展示了SNs信誉值随着epoch的变化。以SN 3为例，在epoch a，初始信誉值为$51>\mathrm{th}_{\mathrm{upper}}$，为候选SN。在epoch b，其信誉值为$16<\mathrm{th}_{\mathrm{low}}$，说

明它可能出故障，无法继续任务。在epoch c，排除故障或重启后，其信誉值重置为50。在epoch d，其信誉值根据其训练表现有相应升降。每个epoch的领导者由信誉机制选取，避免某个SN的绝对控制。

图11展示了7个SNs参与任务时，信誉策略对模型准确率的影响。理想场景中，所有SNs均持有高算力且积极训练，不会缺席训练，信誉机制影响极小；非理想场景考虑SNs算力不均且可能掉线，分别对平均权重策略与熵权法策略进行验证。平均权重策略中权重皆为1/3。结果表明，本文所提出的基于熵权法的信誉机制性能接近理想策略。

图12显示了SORA算法在不同$f^{\max}$下的网络时延。SORA算法在分别在迭代8, 10, 12次之后趋于收敛。此外，网络时延随着$f^{\max}$的增加而降低，因为$f^{\max}$越大意味着可以分配的计算资源越多， FL任务计算时延和块验证时延也相应降低。

图13比较了SORA、统一资源分配 (Uniform Resource Allocation, URA)、随机频率分配 (Random Frequency Allocation, RFA)和随机功率分配 (Random Power Allocation, RPA) 算法的网络时延。其中，SORA算法性能最好，RFA算法由于计算时延较大，性能最差。RPA算法优化了计算频率分配，训练时间大大减少，性能接近SORA算法。

6 结束语

为提高联邦学习效率，本文提出了一种基于区

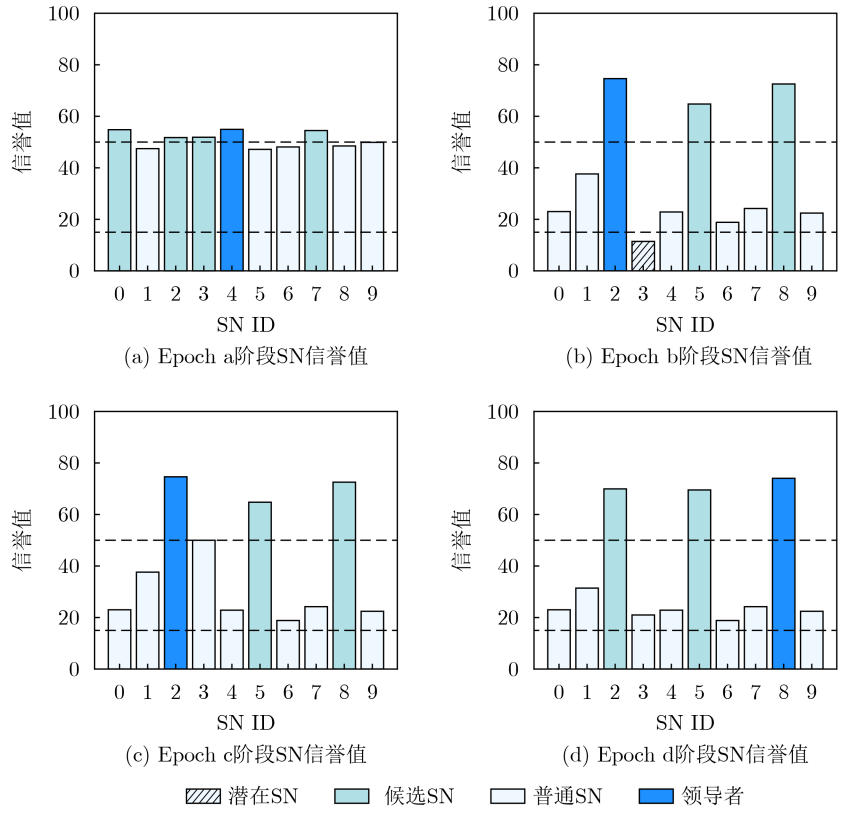


图 10 SNs信誉值的变化

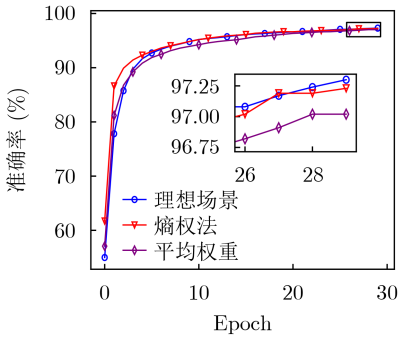


图 11 不同信誉机制对准确率的影响

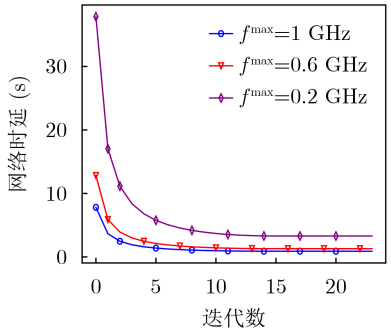


图 12 SORA算法在不同最大算力时约束的收敛情况

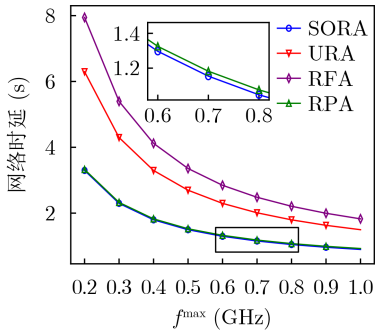


图 13 不同资源分配算法的网络时延对比

块链的异步联邦学习算法AFLChain。首先，基于ENs算力分配训练任务，提高计算资源利用率。其次，引入熵权信誉机制评估ENs并对其分级，承担更多训练任务的SN将获得更高信誉值。最后，通过联合优化传输功率和计算资源分配，最小化整体网络延迟。仿真结果验证了所提算法的有效性。

参 考 文 献

[1] ZHANG Jing and TAO Dacheng. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things[J]. *IEEE Internet of Things Journal*, 2021, 8(10): 7789–7817. doi: 10.1109/JIOT.2020.3039359.

[2] JIANG Chunxiao, ZHANG Haijun, REN Yong, *et al*. Machine learning paradigms for next-generation wireless networks[J]. *IEEE Wireless Communications*, 2017, 24(2): 98–105. doi: 10.1109/MWC.2016.1500356WC.

[3] LIM W Y B, LUONG N C, HOANG D T, *et al*. Federated learning in mobile edge networks: A comprehensive survey[J]. *IEEE Communications Surveys & Tutorials*, 2020, 22(3): 2031–2063. doi: 10.1109/COMST.2020.2986024.

[4] LI Tian, SAHU A K, TALWALKAR A, *et al*. Federated learning: Challenges, methods, and future directions[J]. *IEEE Signal Processing Magazine*, 2020, 37(3): 50–60. doi: 10.1109/MSP.2020.2975749.

[5] IEEE Std 3652.1-2020 IEEE guide for architectural framework and application of federated machine learning[S]. IEEE, 2021. doi: 10.1109/IEEESTD.2021.9382202..

[6] SHEN Xin, LI Zhuo, and CHEN Xin. Node selection strategy design based on reputation mechanism for hierarchical federated learning[C]. 2022 18th International Conference on Mobility, Sensing and Networking (MSN), Guangzhou, China, 2022: 718–722. doi: 10.1109/MSN57253.2022.00117.

[7] LIU Jianchun, XU Hongli, WANG Lun, *et al*. Adaptive asynchronous federated learning in resource-constrained edge computing[J]. *IEEE Transactions on Mobile Computing*, 2023, 22(2): 674–690. doi: 10.1109/TMC.2021.3096846.

[8] LI Zonghang, ZHOU Huaman, ZHOU Tianyao, *et al*. ESync: Accelerating intra-domain federated learning in heterogeneous data centers[J]. *IEEE Transactions on Services Computing*, 2022, 15(4): 2261–2274. doi: 10.1109/TSC.2020.3044043.

[9] CHEN Yang, SUN Xiaoyan, and JIN Yaochu. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, 31(10): 4229–4238. doi: 10.1109/TNNLS.2019.2953131.

[10] CAO Mingrui, ZHANG Long, and CAO Bin. Toward on-device federated learning: A direct acyclic graph-based blockchain approach[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2023, 34(4): 2028–2042. doi: 10.1109/TNNLS.2021.3105810.

[11] FENG Lei, YANG Zhixiang, GUO Shaoyong, *et al*. Two-layered blockchain architecture for federated learning over the mobile edge network[J]. *IEEE Network*, 2022, 36(1): 45–51. doi: 10.1109/MNET.011.2000339.

[12] QIN Zhenquan, YE Jin, MENG Jie, *et al*. Privacy-preserving blockchain-based federated learning for marine internet of things[J]. *IEEE Transactions on Computational Social Systems*, 2022, 9(1): 159–173. doi: 10.1109/TCSS.2021.3100258.

[13] XU Chenhao, QU Youyang, LUAN T H, *et al*. An efficient and reliable asynchronous federated learning scheme for smart public transportation[J]. *IEEE Transactions on Vehicular Technology*, 2023, 72(5): 6584–6598. doi: 10.1109/TVT.2022.3232603.

[14] LI Qinbin, HE Bingsheng, and SONG D. Model-contrastive federated learning[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, USA, 2021: 10708–10717. doi: 10.1109/CVPR46437.2021.01057.

[15] 范盛金. 一元三次方程的新求根公式与新判别法[J]. 海南师范学院学报(自然科学版), 1989, 2(2): 91–98.
FAN Shengjin. A new extracting formula and a new distinguishing means on the one variable cubic equation[J]. *Natural Science Journal of Hainan Normal College*, 1989, 2(2): 91–98.

黄晓舸：女，教授，博士，研究方向为移动通信技术、区块链、联邦学习、资源分配相关技术.

邓雪松：男，硕士生，研究方向为移动通信技术、联邦学习相关技术.

陈前斌：男，教授，博士生导师，研究方向为新一代移动通信网络、未来网络、LTE-Advanced异构小蜂窝网络.

张　杰：男，教授，博士，研究方向为移动通信、IoT等.

责任编辑：于　青