

## 工业物联网中基于信息熵的联邦增量学习算法与优化

杨睿哲<sup>\*①②</sup> 谢欣儒<sup>①②</sup> 滕颖蕾<sup>③</sup> 李萌<sup>①②</sup> 孙艳华<sup>①②</sup> 张大君<sup>④</sup>

<sup>①</sup>(北京工业大学信息学部 北京 100124)

<sup>②</sup>(北京工业大学先进信息网络北京实验室 北京 100124)

<sup>③</sup>(北京邮电大学电子工程学院 北京 100083)

<sup>④</sup>(卡尔顿大学 渥太华 K1S 5B6)

**摘要:** 面对工业生产过程中大规模、多样且随时间增长的数据和机器学习任务, 该文提出一种基于信息熵的联邦增量学习(FIL)与优化方法。基于联邦框架, 各本地计算节点可利用本地数据进行模型训练, 并计算信息平均熵上传至服务器, 以此辅助识别类增任务; 全局服务器则根据本地反馈的平均熵选择参与当前轮次训练的本地节点, 并判决任务是否产生增量后, 进行全局模型下发与聚合更新。所提方法结合平均熵和阈值进行不同情况下的节点选择, 实现低平均熵下的模型稳定学习和高平均熵下的模型增量式扩展。在此基础上, 采用凸优化, 在资源有限的情况下自适应地调整聚合频率和资源分配, 最终实现模型的有效收敛。仿真结果表明, 在不同的情景下, 该文所提方法都可以加速模型收敛并提升训练精度。

**关键词:** 工业物联网; 联邦增量学习; 信息平均熵

中图分类号: TN919.8

文献标识码: A

文章编号: 1009-5896(2024)08-3146-09

DOI: 10.11999/JEIT231240

## Entropy-based Federated Incremental Learning and Optimization in Industrial Internet of Things

YANG Ruizhe<sup>①②</sup> XIE Xinru<sup>①②</sup> TENG Yinglei<sup>③</sup> LI Meng<sup>①②</sup>

SUN Yanhua<sup>①②</sup> ZHANG Dajun<sup>④</sup>

<sup>①</sup>(Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China)

<sup>②</sup>(Beijing Laboratory of Advanced Information Networks, Beijing University of Technology,

Beijing 100124, China)

<sup>③</sup>(Electronic Engineering School, Beijing University of Posts and Telecommunications, Beijing 100083, China)

<sup>④</sup>(School of Information Technology, Carleton University, Ottawa K1S 5B6, Canada)

**Abstract:** In the face of large-scale, diverse, and time-evolving data, as well as machine learning tasks in industrial production processes, a Federated Incremental Learning(FIL) and optimization method based on information entropy is proposed in this paper. Within the federated framework, local computing nodes utilize local data for model training, and compute the average entropy to be transmitted to the server to assist in identifying class-incremental tasks. The global server then selects local nodes for current round training based on the locally provided average entropy and makes decisions on task incrementality, followed by global model deployment and aggregation updates. The proposed method combines average entropy and thresholds for nodes selection in various situations, achieving stable model learning under low average entropy and incremental model expansion under high average entropy. Additionally, convex optimization is employed to adaptively adjust aggregation frequency and resource allocation in resource-constrained scenarios, ultimately achieving effective model convergence. Simulation results demonstrate that the proposed method accelerates model convergence and enhances training accuracy in different scenarios.

**Key words:** Industrial Internet of Things (IIoT); Federated Incremental Learning (FIL); Information entropy

收稿日期: 2023-11-08; 改回日期: 2024-04-21; 网络出版: 2024-05-13

\*通信作者: 杨睿哲 yangruizhe@bjut.edu.cn

基金项目: 国家自然科学基金(62171062, 62371012)

Foundation Items: The National Natural Science Foundation of China (62171062, 62371012)

## 1 前言

工业物联网(Industrial Internet of Things, IIoT)基于物联网技术,融合互联网服务,为工业化与信息化深度融合带来了革命性的推进,在工业制造中实现数据驱动的自动化与智能控制<sup>[1,2]</sup>。目前,IIoT已进入与传统产业深度融合发展的崭新阶段。然而工业环境设备节点具有连接密度高、数量大、种类多、特征维度高等特点,为场景中的应用带来了如大量设备连接困难、大规模数据计算处理复杂、传输隐私安全威胁、智能高效操作系统缺乏等挑战<sup>[3]</sup>。

另一方面,在工业场景下,数据规模随时间变化的特性导致传统学习无法在持续长时间的训练问题保持新旧数据类的性能平衡,即模型在动态场景中的表现不佳,特别是当面向生产需求,训练的类别增加、本地节点接收到新类并依此进行新的训练后,巨大的数据集规模与差异造成了标签训练的负担<sup>[4]</sup>,模型对于旧类的性能会显著下降,产生灾难性遗忘(catastrophic forgetting)<sup>[5]</sup>。

鉴于此,将联邦学习(Federated Learning, FL)<sup>[6]</sup>与增量学习(Incremental Learning, IL)<sup>[7]</sup>模型相结合,将不同类集间形成增量的分布式计算与聚合完成全局全类的模型学习结合,以形成一种有效的方案<sup>[8]</sup>。那么在有限的传输、计算资源条件下,如何高效地完成收集信息并开展判断增量任务的产生,以有效地开展模型训练,是亟待解决的问题。

针对这一问题,本文面向工业生产,提出联邦增量学习模型(Federated Incremental Learning, FIL),以解决大规模数据学习的问题和增量数据带来的灾难性遗忘问题;根据信息平均熵的高低衡量信息量的稳定性和增长性,借鉴强化学习思想,以阈值法平衡控制模型优化给出节点选择策略及资源分配自适应策略,据此实现数据模型的稳定学习和增量式的模型扩展;在资源受限的场景中,联合优化计算资源优化与聚合频率,实现与模型更新动态同步的自适应训练方法。

## 2 相关工作

IL作为一种旨在解决灾难性遗忘问题的机器学习技术,在从新数据中实时学习信息并学习新任务的同时,仍能高效应对已学习过的任务,确保旧知识不会被新知识所替代<sup>[7,9-12]</sup>。其中,正则化技术通过正则化参数选择性地保留旧知识来减轻灾难性遗忘,如无遗忘学习(Learning without Forgetting, LwF)、基于Cramer-Wold距离的目标层自适应正则化(Cramer-Wold distances-based Target

Layer adaptive Regularization, CW-TaLaR)等<sup>[7,9]</sup>。而回放技术则将旧知识存储在生成模型中来减轻遗忘,如增量分类器和表征学习(incremental Classifier and Representation Learning, iCaRL)及分层相关性重放(Hierarchical Correlations Replay, HCR)等方法<sup>[10,11]</sup>。另一方面,上述两种方法都运用到了一种名为知识蒸馏(Knowledge Distillation, KD)的技术, KD是在教师模型的监督下,利用软目标支持学生模型的训练过程,被广泛结合其他IL方法应用在各种场景<sup>[12]</sup>。

FL作为一种有效保护数据隐私并解决数据孤岛问题的机器学习方法,目前在工业化场景中备受研究关注。文献<sup>[13]</sup>中提出一种用于工业场景中结合FL实现人工智能的安全隐私保护方案,进一步提升IIoT中FL系统的安全性。工业场景存在计算能力有限、资源有限且设备异构的特点,因此针对节点选择及资源分配的研究不断涌现。目前已有许多研究通过优化调度策略中的变量或是改变调度策略的机制来提升FL模型性能<sup>[14]</sup>,除此之外,相当一部分学者考虑到有限的资源,通过优化FL的各部分资源分配策略来提升效率<sup>[15-17]</sup>。

## 3 系统模型

如图1所示,在工业场景下,大量IIoT传感采集设备作为数据源,为具有计算能力的本地节点 $m$ ,  $m \in M = \{1, 2, \dots, M\}$ 提供异构本地数据 $D_m$ 并储存。本地节点 $m$ 根据节点选择策略利用不断收集更新的数据 $D_m$ 完成局部学习获得局部模型,定期与全局进行交互以得到全局模型,并在过程中根据全局模型 $\theta^{r,t}$ 与本地数据对任务增量进行判断,迭代上述过程,完成模型训练,如图2所示。FIL区别于一般机器学习,其数据具有多样性(类别不同)、可扩展、时间变化性(类别随机增加)等特点,在此分为模型的训练和模型的更新机制两个部分进行

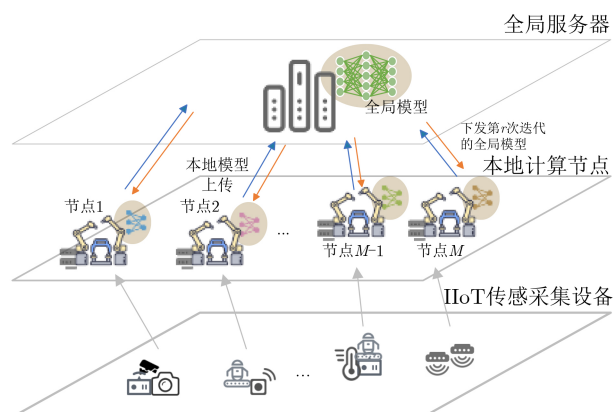


图1 工业场景中的系统模型图

构建。涉及到的重要符号表示及其对应描述如表1所示。

### 3.1 模型训练

基于FL, 节点 $m$ 将从数据采集设备处收集数据后整合为其本地数据 $\mathcal{D}_m = \{(\mathbf{x}_{m,i}, y_{m,i})\}_{i=1}^{N_m}$ ,  $N_m$ 为本地数据集的大小, 设每个节点 $m$ 所拥有的数据 $\mathcal{D}_m$ 与其余节点的数据集 $\mathcal{D}_{m' \neq m}$ 具有差异性。节点 $m$ 采用梯度下降方法训练局部模型, 根据训练进度, 阶段性地将更新的模型传递到中心CPU处进行聚合, 实现本地与全局的交互迭代。

基于IL, 采用 $\mathcal{T} = \{\mathcal{T}^t\}, t = 1, 2, \dots, |\mathcal{T}|$ 来表示一个序列的流式任务, 其中 $|\mathcal{T}|$ 表示任务数量, 如图2所示。标签空间 $\mathcal{Y}^t = \{y_{m,i} | s_m^t = 1\}$ 表示当前第 $t$ 个任务中所选取节点的标签 $y_{m,i}$ 构成的集合, 其中 $s_m^t = 1$ 表示节点 $m$ 参与任务 $t$ 的训练, 否则 $s_m^t = 0$ 。此时, 根据本轮节点选择的不同, 将产生不同于以往的标签 $\cup_{k=1}^{t-1} \mathcal{Y}^k$ 的新标签空间 $\mathcal{Y}^t$ 。

在一个典型的FL问题中, 节点 $m$ 基于本地数据和全局模型的交互, 通过本地迭代来获得模型参数 $\theta_m$ 以使得经验损失函数 $F_c(\theta_m)$ 最小化<sup>[14]</sup>

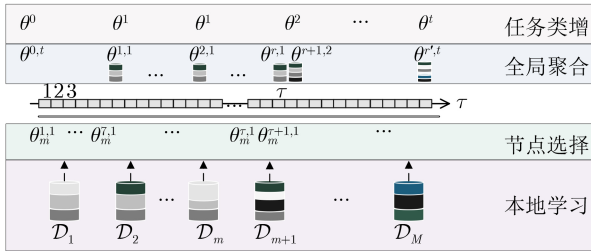


图2 联邦式增量学习系统流程

表1 符号描述

符号	描述
$\theta^{r,t}, \theta_m^{r,t}$	全局模型及本地节点 $m$ 处的局部模型
$f_c(\cdot), f_d(\cdot)$	模型训练损失与蒸馏损失
$\mathcal{S}^r$	第 $r$ 次迭代选择的节点集合
$M, N$	节点总数与每轮选中节点数
$\mathcal{H}_m^{r,t}$	本地数据与模型间的信息熵
$\mathcal{I}(\cdot)$	熵函数
$A_{low}, A_{high}$	贪婪算法下节点选择的阈值
$\eta$	学习率
$c^{loc}, c^{ent}$	本地节点训练任务负载
$c^{glo}$	全局聚合复杂度
$t^*, E^*$	模型迭代中的时延及能耗
$R^{glo}, R^{loc}$	全局及本地训练迭代次数
$f_m^{cmp}$	计算能力
$\varphi, \beta, \delta$	凸优化过程中理论存在的中间值
$\hat{\varphi}^{r,t}, \hat{\beta}^{r,t}, \hat{\delta}^{r,t}$	凸优化过程中实测的中间值

$$F_c(\theta_m) = N_m^{-1} \sum_{i=1}^{N_m} f_c(\mathbf{x}_{m,i}, y_{m,i}, \theta_m) \quad (1)$$

其中,  $\theta_m$ 为节点 $m$ 以数据 $\{(\mathbf{x}_{m,i}, y_{m,i})\}$ 训练的局部模型参数, 以损失函数 $f_c(\mathbf{x}_{m,i}, y_{m,i}, \theta_m)$ 来表征输入 $\mathbf{x}_{m,i}$ 和参数 $\theta_m$ 得到的预测值与实际值 $y_{m,i}$ 的差异。

针对FIL类增过程中数据和模型间的新旧作用, 根据KD建立蒸馏损失, 基于已有模型 $\theta^{old}$ , 建立模型参数 $\theta_m$ 的蒸馏损失 $F_d(\theta_m, \theta^{old})$ 为

$$F_d(\theta_m, \theta^{old}) = N_m^{-1} \sum_{i=1}^{N_m} f_d(\mathbf{x}_{m,i}, y_{m,i}, \theta_m, \theta^{old}) \quad (2)$$

基于式(1)、式(2), 设定 $\theta_m^{\tau,t}$ 为节点 $m$ 在第 $\tau$ 次迭代且处于任务 $t$ 上的模型参数,  $\alpha_1, \alpha_2$ 为不同损失部分的权重值, 损失函数进一步明确为<sup>[10]</sup>

$$\begin{aligned} F(\theta_m^{\tau,t}, \theta^{old}) &= \alpha_1 F_c(\theta_m^{\tau,t}) + \alpha_2 F_d(\theta_m^{\tau,t}, \theta^{old}) \\ &= N_m^{-1} \sum_{i=1}^{N_m} f(\mathbf{x}_{m,i}, y_{m,i}, \theta_m^{\tau,t}, \theta^{old}), \\ f(\mathbf{x}_{m,i}, y_{m,i}, \theta_m^{\tau,t}, \theta^{old}) &= \alpha_1 f_c(\mathbf{x}_{m,i}, y_{m,i}, \theta_m^{\tau,t}) \\ &\quad + \alpha_2 f_d(\mathbf{x}_{m,i}, y_{m,i}, \theta_m^{\tau,t}, \theta^{old}) \end{aligned} \quad (3)$$

其中, 设定全局 $F(\theta^{r,t}) = \sum_M^{m=1} \omega_m F(\theta_m^{r,t}, \theta^{old})$ ,  $\omega_m$ 为权重值。旧模型 $\theta^{old}$ 通过与全局交互获得, 即取决于第 $r$ 次全局聚合后下发的处于任务 $t$ 上的全局更新模型 $\theta^{r,t}$ 和旧模型 $\theta^{old}$ 。当本地第 $\tau-1$ 次迭代和第 $\tau$ 次迭代间具有全局交互时, 令 $\theta_m^{\tau-1,t} = \theta_m^{\tau,t}$ , 否则节点 $m$ 保持本地训练, 有 $\theta_m^{\tau-1,t} = \theta_m^{\tau-1,t}$ 。

基于梯度下降法, 参数 $\theta_m^{\tau,t}$ 进行更新为

$$\theta_m^{\tau,t} = \theta_m^{\tau-1,t} - \eta \cdot \nabla F(\theta_m^{\tau-1,t}, \theta^{old}) \quad (4)$$

其中, 梯度 $\nabla F(\theta_m^{\tau,t}, \theta^{old})$ 是损失 $F(\theta_m^{\tau,t}, \theta^{old})$ 对模型 $\theta_m^{\tau,t}$ 的导数函数,  $\eta > 0$ 为迭代过程中的学习率。

在全局聚合迭代更新的阶段, CPU根据 $\theta_m^{\tau,t}$ 计算获得全局参数 $\theta^{r,t}$

$$\theta^{r,t} = \sum_{m \in \mathcal{S}^r} \omega_m \theta_m^{\tau,t} \quad (5)$$

其中,  $\mathcal{S}^r = \{m | q_m^r = 1, \forall m \in \mathcal{M}\}$ 表示第 $r$ 次迭代选择的节点集合且 $|\mathcal{S}^r| = N$ , CPU广播全局模型 $\theta^{r,t}$ 与当前存储的上一个任务中的最优模型 $\theta^{t-1}$ 。

### 3.2 联邦增量学习的更新迭代机制

已有的FL研究表明, 当计算资源受限, 数据的本地与全局交互存在动态性<sup>[18]</sup>。由此, 认为模型的有效性依赖于从数据本身获得信息的紧密性和稳定性。对应于数据异构的IL, 在资源有限且数据逐渐增加的学习中, 控制信息紧密性和稳定性成为提升模型有效性的关键。

### 3.2.1 信息增益和任务流

由3.1节，模型训练过程中，判断新类的行为，即从 $t-1$ 到 $t$ 的 $\theta^{\text{old}}$ 、 $\theta^{r,t}$ 更新，对损失 $F(\cdot)$ 尤为重要。类增的本质是数据信息的增加，平均熵可以量化本地数据与模型间的信息增益<sup>[19]</sup>，以此判断节点 $m$ 的数据与当前全局模型的相关性<sup>[8]</sup>。具体地，在第 $r$ 个全局聚合中，每个节点接收到全局更新模型 $\theta^{r,t}$ ，并在其当前训练数据 $\mathcal{D}_m$ 上计算平均熵 $\mathcal{H}_m^{r,t}$ ，并反馈给CPU

$$\mathcal{H}_m^{r,t} = N_m^{-1} \mathcal{I}(\{P(\mathbf{x}_{m,i}, \theta^{r,t})\}) \quad (6)$$

其中， $P(\cdot)$ 为归一化指数(Softmax)预测概率， $\mathcal{I}(\cdot)$ 为熵函数，用以描述蒸馏损失，表征两个网络的输出近似程度

$$\mathcal{I}(P((\mathbf{x}_{m,i}, y_{m,i}), \theta^{r,t})) = - \sum_{\mathbf{x}_{m,i}} P((\mathbf{x}_{m,i}, y_{m,i}), \theta^{r,t}) \cdot \log_2 P((\mathbf{x}_{m,i}, y_{m,i}), \theta^{r,t}) \quad (7)$$

基于平均熵原理，当 $\mathcal{H}_m^{r,t}$ 较大时，节点 $m$ 所拥有的数据与当前全局模型的不确定性较大，即节点 $m$ 所拥有的数据对于目前全局模型来说是新类的概率较大。因此，各节点在本轮迭代训练开始前，根据CPU下发的模型和本地数据计算此时的平均熵 $\mathcal{H}_m^{r,t}$ 与熵增 $\Delta \mathcal{H}_m^{r,t}$ 并上报提交给CPU。当熵增超过阈值 $r_h$ 时，即 $\Delta \mathcal{H}_m^{r,t} \geq r_h$ （依据经验，将阈值定为 $r_h = 1.2$ ），CPU认为发现了新的类，执行全局类增，更新 $\theta^{\text{old}} = \theta^{r,t}$ ，并广播 $t \leftarrow t + 1$ ，则各节点根据此通知更新 $\theta^{\text{old}}$ 和 $t \leftarrow t + 1$ 。

### 3.2.2 根据信息增量的节点选择

由于节点持有不同数据类，在大规模IL中，随机选择节点参与训练会导致进度缓慢，效果具有较高的不确定性<sup>[8]</sup>。因此，基于KD对旧模型的迁移，提出基于类增过程中所计算的信息熵进行节点选择。

借鉴强化学习中经验积累的思想，选择较低平均熵的节点集合参与训练( $\Lambda_{\text{low}} \leq r_{\text{and}} < \Lambda_{\text{high}}$ 的概率选择低 $\mathcal{H}_m^{r,t}$ )，确保在一段时间内有稳定的模型经验积累与提升。此外，以一定概率的贪婪学习对数据和类别进行扩展(以 $\Lambda_{\text{high}} \leq r_{\text{and}}$ 的概率选择高 $\mathcal{H}_m^{r,t}$ 的节点)，实现增量过程<sup>[20]</sup>。具体见**算法1**，一般， $\mathcal{E}$ ( $\mathcal{E}$ 次聚合前，以 $r_{\text{and}} < \Lambda_{\text{low}}$ 概率随机选择节点)取较小的正整数。

## 4 优化问题构建与求解

本节针对该模型训练过程进行进一步的分析，在此基础上，通过动态调整聚合周期来与节点选择部分相协调，以提高模型的全局收敛性能。

## 4.1 资源分析

下面将从模型的本地训练和全局聚合两个部分，分别讨论计算资源和时间成本。

### 4.1.1 训练任务中的计算资源与时间分析

对于设备节点 $m$ 有如下假设：式(3)中损失函数 $f(\cdot)$ 的每个样本计算负载为 $c^{\text{loc}}$ ，认为 $c^{\text{loc}}$ 与节点 $m$ 处模型大小相关，式(6)中对于每个样本平均熵的计算负载表示为 $c^{\text{ent}}$ ；节点 $m$ 的计算能力表示为 $f_m^{\text{cmp}}$ ，则 $\mathcal{F} = \{f_m^{\text{cmp}} \mid \forall m \in \mathcal{M}\}$ 表示所有设备节点的计算能力的集合。则节点 $m$ 在 $R^{\text{loc}}$ 次本地迭代之后进行一次聚合提交的时延 $t_m^{\text{cmp}}$ 以及能耗 $e_m^{\text{cmp}}$ 可以由式(8)表示

$$\left. \begin{aligned} t_m^{\text{cmp}} &= (R^{\text{loc}} c^{\text{loc}} N_m + c^{\text{ent}} N_m) (f_m^{\text{cmp}})^{-1} \\ e_m^{\text{cmp}} &= 2^{-1} \psi (R^{\text{loc}} c^{\text{loc}} N_m + c^{\text{ent}} N_m) (f_m^{\text{cmp}})^2 \end{aligned} \right\} \quad (8)$$

其中 $\psi/2$ 是在设备计算芯片组的有效电容系数。

### 4.1.2 全局聚合的计算资源与时间分析

CPU根据所有节点提交的信息熵选择节点，见**算法1**，并据式(5)进行聚合，这两部分的计算复杂度合计为 $c^{\text{glo}}$

$$c^{\text{glo}} = O(M^2 + N|\theta^{r,t}|) \quad (9)$$

其中， $|\theta^{r,t}|$ 表示模型 $\theta^{r,t}$ 的大小。

则 $R^{\text{glo}}$ 次聚合的学习过程中，在CPU中产生的时延以及能量消耗可以表示为

$$\left. \begin{aligned} t^{\text{glo}} &= R^{\text{glo}} c^{\text{glo}} f_0^{-1} \\ e^{\text{glo}} &= 2^{-1} \psi R^{\text{glo}} c^{\text{glo}} f_0^2 \end{aligned} \right\} \quad (10)$$

其中 $f_0$ 为CPU处的计算能力。

由此，系统整体的时延和能量消耗表示为

$$\left. \begin{aligned} e^{\text{sys}} &= e^{\text{glo}} + \sum_{r=1}^{R^{\text{glo}}} \sum_{m \in \mathcal{S}^r} e_m^{\text{cmp}} \\ t^{\text{sys}} &= t^{\text{glo}} + \sum_{r=1}^{R^{\text{glo}}} \max_{m \in \mathcal{S}^r} \{t_m^{\text{cmp}}\} \end{aligned} \right\} \quad (11)$$

**算法1** 基于信息熵的节点选择

- 
- (1) 输入：待选节点数 $M$ ；输出：选中的节点集合 $\mathcal{S}^r$
  - (2) 初始化选择系数 $\Lambda_{\text{low}}$ 和 $\Lambda_{\text{high}}$ ，开始轮次 $\mathcal{E}$
  - (3) **for**  $r = 1, 2, \dots$ , epoch
  - (4) **for**  $\forall m \in M$  **then** 利用式(6)计算 $\mathcal{H}_m^{r,t}$ 上传给CPU
  - (5)  $\mathcal{H}^{r,t} = \{\mathcal{H}_m^{r,t}\}$ ，随机获取 $r_{\text{and}} \in (0, 1)$
  - (6) **if**  $r < \mathcal{E}$  **and**  $r_{\text{and}} < \Lambda_{\text{low}}$
  - (7) **then** 随机选择 $N$ 个节点为 $\mathcal{S}$ , **go to** (10)
  - (8) **if**  $\Lambda_{\text{low}} \leq r_{\text{and}} < \Lambda_{\text{high}}$  **then** 选 $\mathcal{H}_m^{r,t}$ 最小的 $N$ 个节点为 $\mathcal{S}^r$
  - (9) **else**: 排序 $\mathcal{H}^{r,t}$ ，选 $\mathcal{H}_m^{r,t}$ 最大的 $N$ 个节点为 $\mathcal{S}^r$
  - (10) 被选节点 $s_{m^*}^t = 1$ ,  $q_{m^*}^r = 1$
-

## 4.2 优化模型

基于4.1的分析, 构建最小化损失函数的目标

$$\arg \min_{R^{\text{loc}}, r} F(\boldsymbol{\theta}^{r,t}) = \arg \min(F(\boldsymbol{\theta}^{r,t}) - F(\boldsymbol{\theta}^*)) \quad (12)$$

其中,  $\boldsymbol{\theta}^*$ 代表训练收敛时的理想全局模型。

为进一步分析式(12), 令损失函数 $F(\boldsymbol{\theta}^{r,t})$ 满足以下假设<sup>[21]</sup>:  $F(\cdot)$ 是凸的;  $F(\cdot)$ 是 $\varphi$ -Lipschitz的; 且 $F(\cdot)$ 是 $\beta$ -Smooth的。此时, 局部损失函数的梯度和全局损失函数的梯度间的散度满足

$$\|\nabla F(\boldsymbol{\theta}_m^{\tau,t}) - \nabla F(\boldsymbol{\theta}^{r,t})\| \leq \delta \quad (13)$$

其中, 设定 $\nabla F(\boldsymbol{\theta}^{r,t}) = \sum_m \omega_m \nabla F(\boldsymbol{\theta}_m^{\tau,t})$ 。

对于 $R^{\text{loc}}$ , 可给出式(14)定义

$$h(R^{\text{loc}}, \beta, \delta) = \delta \beta^{-1} ((\eta \beta + 1)^{R^{\text{loc}}} - 1) - \eta \delta R^{\text{loc}} \quad (14)$$

当 $\eta(1 - \beta\eta/2) - B_0 h(R^{\text{loc}}, \beta, \delta)/R^{\text{loc}} > 0$  (其中 $B_0$ 为控制常数)且满足 $\eta\beta \leq 1$ 时, 可获得收敛上界可以表示为

$$\begin{aligned} & F(\boldsymbol{\theta}^{r,t}) - F(\boldsymbol{\theta}^*) \\ & \leq \left( R^{\text{glo}} \left( \eta \left( 1 - \frac{\beta\eta}{2} \right) - \frac{h(R^{\text{loc}}, \beta, \delta)}{R^{\text{loc}} B_0^{-1}} \right) \right)^{-1} \end{aligned} \quad (15)$$

因此, 优化模型可定义为

$$\left. \begin{aligned} & \min_{R^{\text{loc}}, R^{\text{glo}}, f_m^{\text{cmp}}} \left( R^{\text{glo}} \left( \eta \left( 1 - \frac{\beta\eta}{2} \right) - \frac{B_0 h(R^{\text{loc}}, \beta, \delta)}{R^{\text{loc}}} \right) \right)^{-1} \\ & \text{s.t. C1: } f_m^{\text{cmp}} \leq f^{\text{max}}, f_0 \leq f^{\text{max}} \\ & \text{C2: } t^{\text{sys}} \leq t^* \\ & \text{C3: } e^{\text{sys}} \leq E^* \end{aligned} \right\} \quad (16)$$

以上目标函数是使得损失函数达到最小。C1限制每个节点的计算资源。C2-C3分别从系统总时延、系统总能耗限制整个系统的训练过程, 其中认为总能量 $E^*$ 和总时间 $t^*$ 都会随迭代次数增加而减少。

## 4.3 自适应方案

基于 $\varphi$ ,  $\beta$ 和 $\delta$ 的理论存在性, 在实际训练中, 节点 $m$ 计算获得对应的估计值 $\hat{\varphi}_m^{r,t}$ ,  $\hat{\beta}_m^{r,t}$ ,  $\hat{\delta}_m^{r,t}$ , 进而由CPU根据提交的 $\{\hat{\varphi}_m^{r,t}\}$ ,  $\{\hat{\beta}_m^{r,t}\}$ 和 $\{\hat{\delta}_m^{r,t}\}$ 集合, 获得其全局均值 $\hat{\varphi}^{r,t}$ ,  $\hat{\beta}^{r,t}$ 和 $\hat{\delta}^{r,t}$

$$\left. \begin{aligned} \hat{\varphi}_m^{r,t} &= \|F(\boldsymbol{\theta}_m^{\tau,t}) - F(\boldsymbol{\theta}^{r,t})\| (\|\boldsymbol{\theta}_m^{\tau,t} - \boldsymbol{\theta}^{r,t}\|)^{-1} \\ \hat{\beta}_m^{r,t} &= \|\nabla F(\boldsymbol{\theta}_m^{\tau,t}) - \nabla F(\boldsymbol{\theta}^{r,t})\| (\|\boldsymbol{\theta}_m^{\tau,t} - \boldsymbol{\theta}^{r,t}\|)^{-1} \\ \hat{\delta}_m^{r,t} &= \|\nabla F(\boldsymbol{\theta}_m^{\tau,t}) - \nabla F(\boldsymbol{\theta}^{r,t})\| \end{aligned} \right\} \quad (17)$$

此时, 重写式(16)如式(18)

$$\left. \begin{aligned} & \min U(R^{\text{loc}}, R^{\text{glo}}, f_m^{\text{cmp}}) \\ &= \min_{R^{\text{loc}}, R^{\text{glo}}, f_m^{\text{cmp}}} \left( R^{\text{glo}} \left( \eta \left( 1 - \frac{\hat{\beta}^{r,t} \eta}{2} \right) - \frac{B_0 h(R^{\text{loc}}, \hat{\beta}^{r,t}, \hat{\delta}^{r,t})}{R^{\text{loc}}} \right) \right)^{-1} \\ & \text{s.t. C1: } f_m^{\text{cmp}} \leq f^{\text{max}}, f_0 \leq f^{\text{max}} \\ & \text{C2: } t^{\text{sys},r} \leq t^* \\ & \text{C3: } e^{\text{sys},r} \leq E^* \end{aligned} \right\} \quad (18)$$

其中, 总能量 $E^*$ 和总时间 $t^*$ 随迭代次数更新, 即 $t^* = t^* - t^{\text{sys},r}|_{R^{\text{glo}}=1}$ ,  $E^* = E^* - e^{\text{sys},r}|_{R^{\text{glo}}=1}$ 。参数 $\hat{\varphi}^{r,t}$ ,  $\hat{\beta}^{r,t}$ 和 $\hat{\delta}^{r,t}$ 均基于模型训练过程中的实际测量结果获得, 即参数的实际获取也是自适应算法的一部分。

使用凸优化理论对上节优化模型进行求解。设 $\zeta = \{R^{\text{loc}}, R^{\text{glo}}, \{f_m^{\text{cmp}}\}\}$ , 则式(18)可表示为

$$\left. \begin{aligned} & (\zeta) : \min U(\zeta) \\ & \text{s.t. C1; C2; C3} \end{aligned} \right\} \quad (19)$$

采用连续凸逼近方法, 设第 $v$ 次迭代变量有 $\zeta^{(v)} = \{R^{\text{loc},(v)}, R^{\text{glo},(v)}, \{f_m^{\text{cmp},(v)}\}\}$ , 将上述优化目标进行泰勒展开, 有 $\tilde{P}(\zeta, \zeta^{(v)})$ 为

$$\begin{aligned} \tilde{P}(\zeta, \zeta^{(v)}) &: \min \tilde{U}(\zeta, \zeta^{(v)}) \\ &= U(\zeta^{(v)}) + U'(\zeta^{(v)})(\zeta - \zeta^{(v)}) \\ &\quad + 2^{-1} \gamma_o \|\zeta - \zeta^{(v)}\|^2 \end{aligned} \quad (20)$$

其中, 为确保问题是仍是凸且可解的, 为平方项添加系数 $\gamma_o$ , 一般 $\gamma_o$ 设置为极小值。

结合式(8)、式(10)和式(11), 针对约束中变量相乘项 $R^{\text{glo}} f_0^2, R^{\text{glo}} (f_m^{\text{cmp}})^2$ , 进行如式(21)的变形

$$\left. \begin{aligned} & R^{\text{glo}} f_0^2 = \tilde{g}_1(R^{\text{glo}}, f_0^2, \varpi_{1,0}), R^{\text{glo}} + f_0^2 \leq \varpi_{1,0} \\ & R^{\text{glo}} (f_m^{\text{cmp}})^2 = \tilde{g}_1(R^{\text{glo}}, f_m^{\text{cmp}}, \varpi_{1,m}) \\ & R^{\text{glo}} + (f_m^{\text{cmp}})^2 \leq \varpi_{1,m} \\ & R^{\text{glo}} R^{\text{loc}} (f_m^{\text{cmp}})^2 \\ &= 2^{-1} ((\varpi_{1,m})^2 - (R^{\text{glo}})^2 - (f_m^{\text{cmp}})^4) R^{\text{loc}} \\ &\leq \frac{1}{A, B} \frac{1}{2} \left[ \begin{aligned} & \tilde{g}_1(R^{\text{loc}}, (\varpi_{1,m})^2, \varpi_{2,m}) \\ & + \tilde{g}_2((R^{\text{glo}})^2, R^{\text{loc}}, R^{\text{glo},(v)}, R^{\text{loc},(v)}) \\ & + \tilde{g}_2((f_m^{\text{cmp}})^4, R^{\text{loc}}, f_m^{\text{cmp},(v)}, R^{\text{loc},(v)}) \end{aligned} \right] \end{aligned} \right\} \quad (21)$$

其中, A为使用平方和公式和引入新变量的方法进行放缩, B使用1阶泰勒展开, 具体推导为

$$\left. \begin{aligned}
ab \leq \tilde{g}_1(a, b, \omega) \leq c \\
\Rightarrow \left\{ \begin{aligned}
&\tilde{g}_1(a, b, \omega) = (\omega^2 - a^2 - b^2)/2 \leq c \\
&a + b \leq \omega \\
&-ab \leq \tilde{g}_2(a, b, a^{(v)}, b^{(v)}) \leq c
\end{aligned} \right. \\
\Rightarrow \tilde{g}_2(a, b, a^{(v)}, b^{(v)}) \\
= (a^2 + b^2)/2 - (a^{(v)} + b^{(v)})^2/2 \\
- (a^{(v)} + b^{(v)})(a + b)/2 \leq c
\end{aligned} \right\} \quad (22)$$

将目标与约束进行近似与缩放，优化模型可以表述为如式(23)线性求解模型。其中，C4'引入新变量 $\hat{f}_m^{\text{cmp}}$ ，满足 $(f_m^{\text{cmp}})^{-1} \leq \hat{f}_m^{\text{cmp}}$ ，将C3改写为C3'，进而与C2'具有相似结构，可采用与式(21)相似的缩放方法；C5'和C6'是逼近过程中的引入变量，见式(21)。由此，相应的，可获得 $\zeta$ 的线性函数 $f_{1,L}(\zeta)$ 和 $f_{2,L}(\zeta)$ 。即 $\tilde{\mathcal{P}}(\zeta, \zeta^{(v)})$ 转化为 $\zeta$ 的线性凸优化问题，结合拉格朗日梯度下降方法可进行求解，如算法2所示。

其中， $k^{(v)}$ 表示变量 $\zeta^{(v)}$ 的迭代步长。令 $k^{(0)} = 1$ ，有 $k^{(v)} = k^{(v-1)}(1 - c_k k^{(v-1)})$ ， $c_k$ 为 $k^{(v)}$ 的更新系数。

综合前述内容，给出基于凸优化与信息增益的FIL的完整流程：

(1) 每一轮迭代，CPU下发全局模型 $\theta^{r,t}$ ， $\theta^{\text{old}}$  (当 $r = 0$ 时， $\theta^{r,t} = \theta^{\text{old}} = \theta^{\text{init}}$ )。

(2) 每个设备节点 $m$ 利用式(6)计算本地熵值和熵增，并将结果提交给CPU。

(3) CPU根据提交的熵增进行任务判断，记录任务数 $t$ ，同时决定是否对旧模型 $\theta^{\text{old}}$ 进行更新。

(4) CPU根据算法1进行节点选择，并通知此时被选中的设备节点 $m$ ，( $\forall m \in \mathcal{S}^r$ )。

(5) 被选节点 $m$ 根据式(17)计算估计得到 $\hat{\varphi}_m^{r,t}$ ， $\hat{\beta}_m^{r,t}$ 和 $\hat{\delta}_m^{r,t}$ 并上传提交至CPU处。

(6) CPU依照4.2节和4.3节的凸优化方法求解，返回当前最优的本地迭代次数 $R^{\text{loc}}$ 与资源 $\mathcal{F}$ 。

(7) 本地节点 $m$ 根据CPU的指示进行本地训练，提交局部模型 $\theta_m^{r,t}$ ，CPU进行全局迭代获得全局模型 $\theta^{r,t}$ 。重复上述操作直至资源耗尽。

## 5 实验

### 5.1 实验设计

采用CIFAR-100数据集作为数据源<sup>[22]</sup>，并进行

算法2 优化模型求解

(1)初始化: $\zeta^{(0)}$ , $v = 0$
(2)if $\zeta^{(v)}$ 为稳态解, then返回 $\zeta^{(v)}$
(3) $\hat{\zeta}(\zeta^{(v)}) \triangleq \text{argmin} \tilde{U}(\zeta, \zeta^{(v)})$
(4) $\zeta^{(v+1)} = \zeta^{(v)} + k^{(v)}(\hat{\zeta}(\zeta^{(v)}) - \zeta^{(v)})$
(5) $v \leftarrow v + 1$ , 转到(2)

了分块操作，使得本地数据存在不一致性。仿真配置：采用增量学习模型<sup>[10]</sup>及分类模型ResNet-18<sup>[23]</sup> (学习率1.0的SGD优化器，经验池 $\mathcal{P}$ 内存为200)。数据集 $\mathcal{D}_m$ 拥有随机分配的本地数据集，初始该标签数据20%，并随训练推进实现增长。每轮迭代选中10个节点进行本地训练。其余仿真参数如表2所示。仿真实验中给出以下3种利用不同策略进行节点选择的方法作为对比方案：(1)模型训练采用完全随机方式进行节点选择<sup>[8]</sup>；(2)模型训练采用利用数据标签模拟数据稳定性的节点选择<sup>[24]</sup>；(3)模型训练采用利用平均熵选择节点(该文提出的节点选择方法)。

### 5.2 实验结果

图3显示在不同资源条件下，不同节点分布式训练模型，使全局模型收敛的过程及结果。可见，在各种资源条件下，完全随机选择节点进行训练的模型的准确度最低，而平均熵进行节点选择后，性能显著上升。此外，提出方案与方案(3)相比，除节点选择不同，模型的聚合频率也有差异。提出方案通过自适应优化实现基于资源约束的最优训练策略，而方案(3)则根据资源的初始值规定了固定的聚合频率。比较图3(a)和图3(b)，在资源有限时，完全随机的训练将消耗更多的成本，资源有限将无法支持随机模型进行足够多的迭代训练后达到收敛。如图3(b)所示，方案(3)可以提升模型训练效果，但与提出方案相比仍有差距，即基于平均熵的节点选择能够较好地指导模型进行训练，在此基础上进行资源的分配与自适应优化能提升模型训练的效率，以优化资源并有针对性地训练模型直至收敛。

图4显示了在能量充足，模型训练收敛时，不同的阈值( $A_{\text{low}}$ ,  $A_{\text{high}}$ )对模型的影响。 $A_{\text{low}}$ 控制模型根据熵选择节点参与训练的起始，较低的 $A_{\text{low}}$ 可以使包含更多类信息的节点参与训练，影响起始段的信息稳定性，反之则会导致选择的节点数量较少，

表2 参数说明

参数	定义	值
$M$	节点数量	[10, 200]
$E$	总能量	[500, 1 000] J
$t$	总时间	[2 000, 20 000] s
$ \mathcal{D}_m $	节点 $m$ 数据集大小	[120, 200]
$c^{\text{loc}}$	本地计算负荷	15 cycles
$c^{\text{gl}}$	全局计算负荷	25 cycles
$f_m^{\text{cmp}}$	节点 $m$ 计算能力	[1, 10] GHz
$\psi/2$	节点计算芯片组的有效电容系数	$10^{-28}$
$p_m^{\text{min}}, p_m^{\text{max}}$	设备节点功率约束	[0.2, 1] W

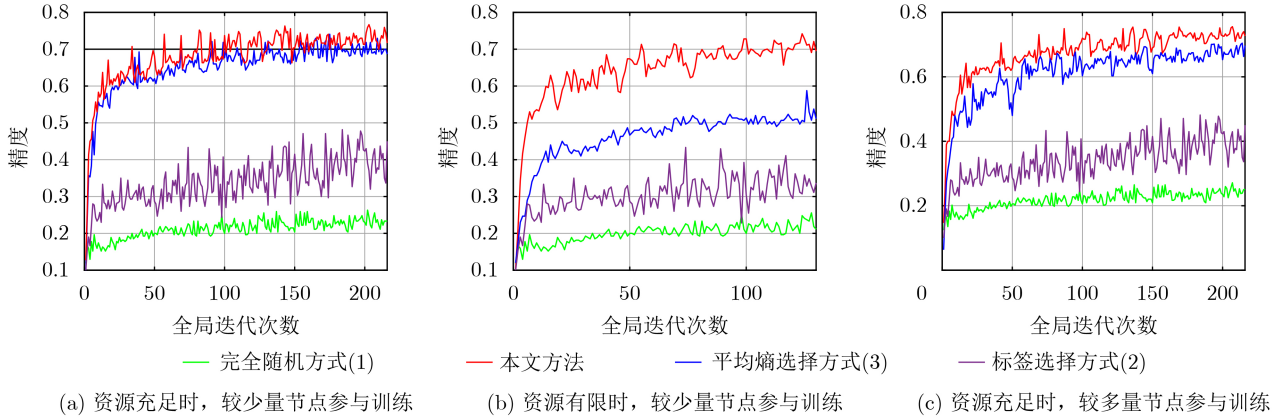


图3 不同情况下4种算法的平均精度

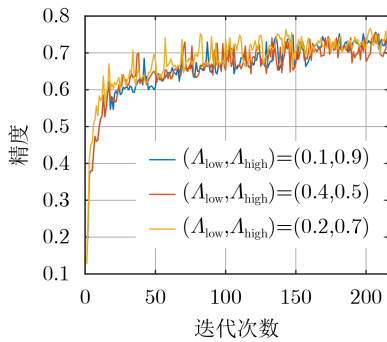


图4 资源充足时, 阈值对模型训练的影响

包含类信息过低, 影响后续的扩展性。较低的  $A_{high}$  会鼓励模型更多地探索新的信息和样本, 反之则导致模型更多地利用已知信息, 但缺乏探索新知的能力, 和最终全类模型的精度。

图5给出在能量充足时, 规定全局迭代轮数一致以讨论参与节点数量对模型准确性的影响。当各节点持有3类数据时(数据量相同), 10节点全部参与聚合获得的模型精度远低于50节点中每轮选择10节点聚合的情况。这说明当节点数增加时, 数据总量的增加能够获得更为精确的模型。

$$\left. \begin{aligned}
 & \tilde{\mathcal{P}}(\zeta, \zeta^{(v)}) : \min \tilde{U}(\zeta, \zeta^{(v)}) = U(\zeta^{(v)}) + U'(\zeta^{(v)}) (\zeta - \zeta^{(v)}) + \frac{1}{2} \gamma_o \|\zeta - \zeta^{(v)}\|^2 \\
 & \text{s.t. C1} : f_m^{\text{cmp}} \leq f_m^{\text{max}}, f_0 \leq f_m^{\text{max}} \\
 & \text{C2}' : 2^{-1} \psi c^{\text{glo}} R^{\text{glo}} f_0^2 + \sum_{m \in S^r} 2^{-1} \psi (c^{\text{loc}} N_m R^{\text{glo}} R^{\text{loc}} (f_m^{\text{cmp}})^2 + c^{\text{ent}} N_m R^{\text{glo}} (f_m^{\text{cmp}})^2) \leq f_{1,L}(\zeta) \leq t^* \\
 & \text{C3}' : c^{\text{glo}} R^{\text{glo}} \hat{f}_0 + \left( c^{\text{loc}} N_m R^{\text{loc}} R^{\text{glo}} \hat{f}_m^{\text{cmp}} + c^{\text{ent}} N_m R^{\text{glo}} \hat{f}_m^{\text{cmp}} \right) \leq f_{2,L}(\zeta) \leq E^*, m \in S^r \\
 & \text{C4}' : -\hat{f}_m^{\text{cmp}} f_m^{\text{cmp}} \leq -1 \Rightarrow \tilde{g}_2 \left( \hat{f}_m^{\text{cmp}}, f_m^{\text{cmp}}, \hat{f}_m^{\text{cmp},(v)}, f_m^{\text{cmp},(v)} \right) \leq -1 \\
 & \text{C5}' : R^{\text{glo}} + f_0 \leq \varpi_{1,0}; R^{\text{glo}} + (f_m^{\text{cmp}})^2 \leq \varpi_{1,m}; R^{\text{loc}} + (\varpi_{1,m})^2 \leq \varpi_{2,m} \\
 & \text{C6}' : R^{\text{glo}} + \hat{f}_0 \leq \hat{\varpi}_{1,0}; R^{\text{glo}} + \hat{f}_m^{\text{cmp}2} \leq \hat{\varpi}_{1,m}; R^{\text{loc}} + (\hat{\varpi}_{1,m})^2 \leq \hat{\varpi}_{2,m}
 \end{aligned} \right\} \quad (23)$$

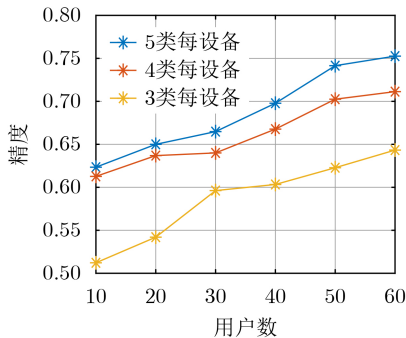


图5 资源充足时, 数据量与节点类对模型训练的影响

另一方面, 增加单节具有的类获得更高的模型精度, 如20节点每节点4类的性能远高于30节点每节点3类的情况, 虽然总数据量后者更高, 但由于前者节点间交叉重叠的信息量更有利于分布式的增量学习, 同时说明了提出算法中以平均熵来控制学习过程中的信息量和增量的依据性。

图6给出不同系统能量条件下, 不同方案的模型准确度与资源优化后训练的迭代次数。如图所示, 在有限资源的情况下, 提出方法利用节点选择和自适应优化获得了最高的模型精度。其中, 提出算法与方案(1)对比表明节点选择, 即模型信息的

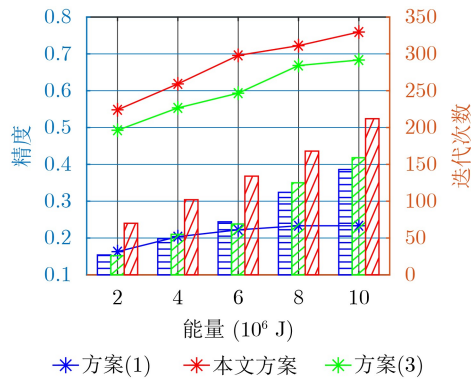


图6 不同资源下, 不同模型的训练情况

获取控制获得性能提升较高; 在此基础上, 与基于平均熵选择的方案(3)对比表明动态调整本地训练次数和资源优化分配可进一步提升收敛性能。同时根据迭代次数可知, 对比算法中的基于完全随机选择的方案(1)由于不涉及资源分配, 在限制资源的情况下, 极少次迭代后便会停止, 导致模型过早停止训练无法收敛。基于平均熵选择的方案(3)与方案(2)的迭代次数相近, 其算法性能的改善得益于节点选择和任务增量的控制, 然而其未进行资源优化使得训练迭代不足, 与提出算法相比仍具有较为明显的性能差距。

## 6 结束语

本文基于工业场景提出了综合考虑信息和资源优化的联邦式增量学习模型。在模型训练阶段, 设计了基于平均熵衡量模型相关性的节点选择方法与通过凸优化实现的自适应训练方案。其次, 将联合优化资源与聚合频率的目标建模为一个凸问题, 利用凸优化求解, 以实现模型自适应地在资源受限的情况下高效完成训练任务。最后, 实验证明了上述模型的有效性。未来的工作将会结合网络传输条件进行联合优化, 并从可持续学习的角度, 拉长增量的时间尺度。

## 参考文献

- [1] BJORNSON E and SANGUINETTI L. Scalable cell-free massive MIMO systems[J]. *IEEE Transactions on Communications*, 2020, 68(7): 4247–4261. doi: [10.1109/tcomm.2020.2987311](https://doi.org/10.1109/tcomm.2020.2987311).
- [2] YU Wanke and ZHAO Chunhui. Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability[J]. *IEEE Transactions on Industrial Electronics*, 2020, 67(6): 5081–5091. doi: [10.1109/tie.2019.2931255](https://doi.org/10.1109/tie.2019.2931255).
- [3] HUO Ru, ZENG Shiqin, WANG Zhihao, et al. A comprehensive survey on Blockchain in industrial internet of things: Motivations, research progresses, and future challenges[J]. *IEEE Communications Surveys & Tutorials*, 2022, 24(1): 88–122. doi: [10.1109/comst.2022.3141490](https://doi.org/10.1109/comst.2022.3141490).
- [4] SU Hang, QI Wen, HU Yingbai, et al. An incremental learning framework for human-like redundancy optimization of anthropomorphic manipulators[J]. *IEEE Transactions on Industrial Informatics*, 2022, 18(3): 1864–1872. doi: [10.1109/tii.2020.3036693](https://doi.org/10.1109/tii.2020.3036693).
- [5] KIRKPATRICK J, PASCANU R, RABINOWITZ N, et al. Overcoming catastrophic forgetting in neural networks[J]. *Proceedings of the National Academy of Sciences of the United States of America*, 2017, 114(13): 3521–3526. doi: [10.1073/pnas.1611835114](https://doi.org/10.1073/pnas.1611835114).
- [6] SHOHAM N, AVIDOR T, KEREN A, et al. Overcoming forgetting in federated learning on non-IID data[J]. arXiv: 1910.07796, 2019. doi: [10.48550/arXiv.1910.07796](https://doi.org/10.48550/arXiv.1910.07796).
- [7] LI Zhizhong and HOIEM D. Learning without forgetting[C]. The 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 2016: 614–629. doi: [10.1007/978-3-319-46493-0\\_37](https://doi.org/10.1007/978-3-319-46493-0_37).
- [8] DONG Jiahua, WANG Lixu, FANG Zhen, et al. Federated class-incremental learning[C]. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, USA, 2022: 10154–10163. doi: [10.1109/cvpr52688.2022.00992](https://doi.org/10.1109/cvpr52688.2022.00992).
- [9] MAZUR M, PUSTELNIK Ł, KNOP S, et al. Target layer regularization for continual learning using Cramer-Wold distance[J]. *Information Sciences*, 2022, 609: 1369–1380. doi: [10.1016/j.ins.2022.07.085](https://doi.org/10.1016/j.ins.2022.07.085).
- [10] REBUFFI S-A, KOLESNIKOV A, SPERL G, et al. iCaRL: Incremental classifier and representation learning[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, USA, 2017: 5533–5542. doi: [10.1109/cvpr.2017.587](https://doi.org/10.1109/cvpr.2017.587).
- [11] WANG Qiang, LIU Jiayi, JI Zhong, et al. Hierarchical correlations replay for continual learning[J]. *Knowledge-Based Systems*, 2022, 250: 109052. doi: [10.1016/j.knosys.2022.109052](https://doi.org/10.1016/j.knosys.2022.109052).
- [12] JI Zhong, LI Jin, WANG Qiang, et al. Complementary calibration: Boosting general continual learning with collaborative distillation and self-supervision[J]. *IEEE Transactions on Image Processing*, 2023, 32: 657–667. doi: [10.1109/tip.2022.3230457](https://doi.org/10.1109/tip.2022.3230457).
- [13] HAO Meng, LI Hongwei, LUO Xizhao, et al. Efficient and privacy-enhanced federated learning for industrial artificial intelligence[J]. *IEEE Transactions on Industrial Informatics*, 2020, 16(10): 6532–6542. doi: [10.1109/tii.2019.2945367](https://doi.org/10.1109/tii.2019.2945367).
- [14] CHEN Zhixiong, YI Wenqiang, DENG Yansha, et al. Device scheduling for wireless federated learning with

- latency and representativity[C]. 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Maldives, Maldives, 2022: 1–6. doi: [10.1109/iceccme55909.2022.9988590](https://doi.org/10.1109/iceccme55909.2022.9988590).
- [15] YANG Zhaohui, CHEN Mingzhe, SAAD W, *et al.* Energy efficient federated learning over wireless communication networks[J]. *IEEE Transactions on Wireless Communications*, 2021, 20(3): 1935–1949. doi: [10.1109/twc.2020.3037554](https://doi.org/10.1109/twc.2020.3037554).
- [16] DINH C T, TRAN N H, NGUYEN M N H, *et al.* Federated learning over wireless networks: Convergence analysis and resource allocation[J]. *IEEE/ACM Transactions on Networking*, 2021, 29(1): 398–409. doi: [10.1109/tnet.2020.3035770](https://doi.org/10.1109/tnet.2020.3035770).
- [17] JING Shusen and XIAO Chengshan. Federated learning via over-the-air computation with statistical channel state information[J]. *IEEE Transactions on Wireless Communications*, 2022, 21(11): 9351–9365. doi: [10.1109/twc.2022.3175887](https://doi.org/10.1109/twc.2022.3175887).
- [18] YANG Ruizhe. The adaptive distributed learning based on homomorphic encryption and blockchain[J]. *High Technology Letters*, 2022, 28(4): 337–344. doi: [10.3772/j.issn.1006-6748.2022.04.001](https://doi.org/10.3772/j.issn.1006-6748.2022.04.001).
- [19] ZHAO Rui, SONG Jinming, YUAN Yufeng, *et al.* Maximum entropy population-based training for zero-shot human-AI coordination[C]. The AAAI Conference on Artificial Intelligence, Washington, USA, 2023: 6145–6153. doi: [10.1609/aaai.v37i5.25758](https://doi.org/10.1609/aaai.v37i5.25758).
- [20] ZHANG Peiying, WANG Chao, JIANG Chunxiao, *et al.* Deep reinforcement learning assisted federated learning algorithm for data management of IIoT[J]. *IEEE Transactions on Industrial Informatics*, 2021, 17(12): 8475–8484. doi: [10.1109/tii.2021.3064351](https://doi.org/10.1109/tii.2021.3064351).
- [21] WANG Shiqiang, TUOR T, SALONIDIS T, *et al.* Adaptive federated learning in resource constrained edge computing systems[J]. *IEEE Journal on Selected Areas in Communications*, 2019, 37(6): 1205–1221. doi: [10.1109/jsac.2019.2904348](https://doi.org/10.1109/jsac.2019.2904348).
- [22] KRIZHEVSKY A and HINTON G. Learning multiple layers of features from tiny images[J]. *Handbook of Systemic Autoimmune Diseases*, 2009, 1(4).
- [23] SHAFIQ M and GU Zhaoquan. Deep residual learning for image recognition: A survey[J]. *Applied Sciences*, 2022, 12(18): 8972. doi: [10.3390/app12188972](https://doi.org/10.3390/app12188972).
- [24] MINAEE S, KALCHBRENNER N, CAMBRIA E, *et al.* Deep learning-based text classification: A comprehensive review[J]. *ACM Computing Surveys*, 2022, 54(3): 62. doi: [10.1145/3439726](https://doi.org/10.1145/3439726).
- 杨睿哲: 女, 副教授, 研究方向为区块链技术、无线通信网络等。  
 谢欣儒: 女, 硕士生, 研究方向为联邦学习、机器学习。  
 滕颖蕾: 女, 教授, 研究方向为网络组网理论、随机优化等。  
 李 萌: 男, 副教授, 研究方向为边缘计算、无线通信网络等。  
 孙艳华: 女, 副教授, 研究方向为机器学习、边缘计算等。  
 张大君: 男, 助理研究员, 研究方向为人工智能, 计算机、通信的智能网络等。

责任编辑: 余 蓉