计算和通信高效轻量级垂直联邦学习在智能建筑物联网中的应用

Heqiang Wang, Xiang Liu, Yucheng Liu, Jia Zhou, Weihong Yang, Xiaoxiong Zhong

Department of New Network,
Peng Cheng Laboratory, Shenzhen, 518066, China

摘要—随着智能建筑中物联网设备数量的增加和能力的提升，这些设备正在从基础的数据收集与控制发展到积极参与深度学习任务。联邦学习（FL）作为一种去中心化的学习范式，非常适合此类场景。然而，物联网设备有限的计算和通信资源带来了显著的挑战。尽管现有研究已经广泛探索了在水平联邦学习中的效率改进，但由于数据分割和模型结构的基本差异，这些技术不能直接应用于垂直联邦学习。为了解决这一差距，我们提出了一种轻量级垂直联邦学习（LVFL）框架，该框架共同优化计算和通信效率。我们的方法引入了两种不同的轻量化策略：一种用于降低特征模型的复杂性以改进本地计算，另一种用于压缩特征嵌入以减少通信开销。此外，我们为所提出的 LVFL 算法推导了一个收敛界，明确地结合了计算和通信轻量化的比率。在图像分类任务上的实验结果表明，LVFL 有效地减轻了资源需求，同时保持了竞争性的学习性能。

***Index terms*— 联邦学习，模型剪枝，智能建筑，物联网部署。**

I. 介绍

随着智能建筑中物联网设备的持续普及和功能提升，它们的角色已经从基础感知和控制任务扩展到深度学习过程中的主动参与。联邦学习（FL），一种去中心化的机器学习范式，使多个客户端能够在中央服务器的协调下共同训练一个全局模型，每个客户端都存储了本地数据。通过将数据保留在边缘设备上，联邦学习减少了对数据传输的需求，并缓解了隐私风险。图 1 展示了智能建筑环境中联邦学习的一个典型应用，其中从不同楼层收集的传感器数据代表了不同的特征空间，自然地与垂直联邦学习（VFL）的原则相吻合。在 VFL 中，每个客户端都持有同一组样本的一套独特特征。在 VFL 训练过程中，每个客户端开发一个特征模型，将原始数据特征转换为称为“特征嵌入”的向量表示形式。在 VFL 中，客户端发送这些特征嵌入到服务器，而不是传输它们的特征模型。然后，服务器将这些嵌入整合进头部模型以确定最终损失。这一工作流程突显了 VFL 与传统 FL 之间的根本差异，并带来了许多无法通过直接应用为传统 FL 设计的技术来有效解决的独特挑战。因此，满足 VFL 独特需求需要专门为其结构定制的解决方案。

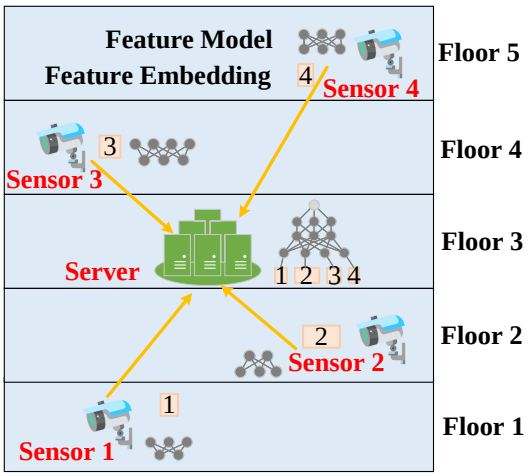


图 1: 基于物联网的智能建筑垂直联邦学习

在 VFL 训练过程中，传感器之间的多样性导致了计算和通信能力的变化，引发了对同步解决方案的需求。虽然以前在传统 FL 中的研究已经解决了这个问题，在传统 FL 中，由于客户端的特征空间相同，因此局部模型和全局模型是一致的，只需要进行局部模型剪枝来满足特定客户端的要求。相比之下，在 VFL 中，各客户端的特征空间不同，导致需要在本地特征模型上分别训练，并随后将训练好的特征嵌入上传到服务器。这种方法对每个传感器提出了不同的计算和通信需求。尽管之前的研究已经探讨了 VFL 中的特征嵌入压缩以减少通信负载 [1], [2]，但关键的计算负担问题却很大程度上被忽略了。解决这个计算负荷问题是 VFL 部署中一个更为紧迫的挑战。

为了应对智能建筑物联网场景中部署 VFL 时遇到的各种计算负担，本文介绍了轻量级垂直联邦学习（LVFL）

的概念，旨在减轻计算和通信效率方面的挑战。本研究的主要贡献总结如下：(1) 本研究引入了 LVFL 框架，专门设计用于适应异构工人不同的通信和计算能力。LVFL 动态调整训练特征模型的计算成本以及更新特征嵌入的通信成本，确保在不同系统架构中保持高效。(2) 我们建立了 LVFL 算法的全面收敛性分析，并提出了描述累积特征模型与特征嵌入轻量化误差之间关系以及通信和计算轻量化比率的收敛界限。(3) 我们的实验是在 CIFAR-10 数据集上进行的，验证了 LVFL 算法调整计算和通信轻量化比率的能力，无论是恒定还是动态调整。提出的 LVFL 算法在显著降低通信和计算负担的同时，能够达到可比的测试准确性水平。理论证明的详细内容可以在在线补充材料中找到，链接：https://github.com/ystex/LVFL/tree/main。

II. 相关工作

近年来，VFL 受到了广泛关注。具有垂直分割数据的联邦学习概念在 [3] 中被提出。关于 VFL 的综合调查已在 [4]–[6] 中呈现。然而，不同于传统的联邦学习，VFL 带来了它自己的挑战。一些研究，如 [7]，旨在优化数据利用以增强 VFL 中的联合模型效果。相比之下，像 [8] 这样的研究则专注于实施隐私保护协议来应对潜在的数据泄露威胁。除此之外，在提高训练效率方面也进行了探索。这些努力主要针对减少通信开销，要么允许参与者在每次迭代中进行多次本地更新 [9]，要么压缩各参与方之间交换的数据 [10]。尽管这些研究主要强调了减少通信开销的重要性，但它们的一个缺点是忽视了通过提高计算效率来增强训练效率的方法。

现代的 DNN 通常包含数千万个参数 [11]。在资源受限设备上的联邦学习中，存储和训练这些高度参数化的模型可能具有挑战性。因此，在训练过程中对模型进行轻量化是一个不可或缺的话题 [12]。考虑到训练设备的计算能力各异，一种可行策略是动态调整本地模型的大小并减少其参数，主要通过模型剪枝来实现。一般来说，模型剪枝技术可以分为两类：非结构化剪枝 [13], [14] 和结构化剪枝 [15]–[17]。非结构化剪枝消除神经网络中非必要的权重，特别是神经元之间的连接，从而导致参数的稀疏性显著增加 [18]。然而，随之而来的稀疏矩阵不规则性在内存中的参数压缩带来了挑战，需要专门的硬件或软件库来实现高效的训练 [19]。相比之下，结构化剪枝旨在丢弃冗余模型结构，如卷积滤波器 [16]，而不引入稀疏性。因此，所得模型可以被视为初始神经网络的一个子集或子配置，包含更少的参数，从而有助于减少计算开销。

虽然模型剪枝研究的很大一部分集中在集中式学习环境中，近期的研究已将这一技术的应用扩展到联邦学习设置中 [20], [21]。[20] 中的工作提出了 PruneFL，这是一种适应性的联邦学习策略，它从选定客户端的初始剪枝开始，并在整个联邦学习过程中继续进行后续剪枝，旨在减少通信和计算开销。[21] 中概述的研究介绍了 FedMP 框架，该框架采用自适应剪枝和恢复技术来提高通信和计算效率，利用多臂赌博机算法选择剪枝比例。值得注意的是，上述模型剪枝技术主要源自传统的联邦学习场景。然而，由于传统联邦学习与垂直联邦学习之间存在固有的差异，这些技术无法无缝应用于垂直联邦学习设置中。因此推动了我们对垂直联邦学习场景中的模型和特征嵌入剪枝研究的动力。

III. 系统模型

首先，我们定义智能建筑物联网中 VFL 的几个基础概念。该 VFL 系统由 K 个客户端和一个中央服务器组成。数据集记为 $\mathbf{x} \in \mathbb{R}^{N \times D}$，其中 N 表示总的数据样本数，D 表示特征的数量。在 $\mathbf{x}$ 中，由 i 索引的行对应于数据样本 x^i。每个样本 x^i 具有一组唯一的特征 x_k^i，这些特征由客户 k 保留，并被识别为不同的子集。每个实例 x^i 都配有一个标签 y^i。向量 $\mathbf{y} \in \mathbb{R}^{N \times 1}$ 表示所有样本标签。客户端 k 维护一个局部特征数据集 $\mathbf{x}_k \in \mathbb{R}^{N \times D_k}$，其中 D_k 表示客户端 k 的特征数量，第 i 行表示相应的特征 x_k^i。在此背景下，我们假设服务器和所有客户端都保留标签 $\mathbf{y}$ 的副本，这与之前的研究一致。

每个客户端，用 $k \in K$ 表示，都有一个独特的特征模型参数集，θ_k。服务器维护头部模型，θ_0。函数 $h_k(\theta_k, x_k^i)$ 表示由客户端 k 从样本 x_k^i 中提取的特征嵌入。该特征嵌入操作通过多层 DNN 将高维原始数据转换为低维表示，有效捕获输入数据中的关键信息同时大幅降低其维度。整个模型的参数统称为 $\Theta = [\theta_0^\top, \theta_1^\top, ..., \theta_K^\top]^\top$。然后，我们可以将学习目标表示为最小化以下方程：

$$F(\Theta; \mathbf{x}; \mathbf{y}) := \frac{1}{N} \sum_{i=1}^{N} l(\theta_0, \{h_k(\theta_k; x_k^i)\}_{k=1}^K; y^i) \quad (1)$$

其中 $l(\cdot)$ 表示单个数据样本的损失函数。在服务器模型框架内，方程 $h_0(\theta_0, x^i) = \theta_0$ 始终适用。为了在整个论文中提高清晰度并确保符号的一致性，我们在后续讨论中实施了几项简化措施。首先，数据集 $\mathbf{x}_k$ 的特征嵌入由 $h_k(\theta; \mathbf{x}_k)$ 表示，我们通常将其简写为 $h_k(\theta_k; \mathbf{x}_k) = h_k(\mathbf{x}_k)$。其次，我们将 $k = 0$ 分配给 FC 中的头模型表示，其中 $h_0(\theta_0) = \theta_0$ 用于方便起见。最后，我们经常表示 $F(\Theta) = F(h_0(\theta_0), h_1(\theta_1), ..., h_K(\theta_K))$，特别是在后续的

算法细节和收敛性分析中。在以下部分，我们将详细阐述 LVFL 的具体算法工作流程。

IV. 轻量级垂直联邦学习

在本节中，我们介绍了智能建筑物联网场景下 LVFL 算法的细节。考虑到 VFL 的独特结构属性，其中计算需求由模型大小决定，通信要求与特征嵌入大小相关，从传统联邦学习派生出的轻量化方法并不直接适用于 VFL。为解决这一局限性，我们采用了一种针对每个客户端特征模型的渐进式结构剪枝策略和一种针对每个客户端特征嵌入的非结构化剪枝策略。这些方法由每个客户端不同的计算和通信能力指导。通过这些剪枝技术，我们在智能建筑物联网场景中确保所有客户端的一致同步，并促进服务器的有效更新。

在此场景中，我们引入了一种双重轻量化比率机制以明显优化计算和通信。我们将 α_k^t 表示为计算轻量化比率并将 β_k^t 表示为通信轻量化比率。更高的 α_k^t 意味着处理数据样本时对 CPU 周期的需求减少。相应地，更大的 β_k^t 标志着上传过程中嵌入大小需求的减少。假设每个客户端在将模型传输到服务器之前执行 E 个本地迭代，并且训练过程跨越总共 R 轮，导致累计时间为 $T = RE$ 个本地迭代。算法 1 展示了 LVFL 的工作流程，集成了双重轻量化比率机制。

在每个全局轮次中，当 $t \mid E = 0$ 时，每个客户端将基于通信轻量化比率 β_k^t（第 5 行）决定非结构化剪枝特征嵌入 $\hat{h}_k^t(\hat{\theta}_k^t; \mathbf{x}_k)$，并将其剪枝后的特征嵌入传输到服务器进行全局更新（第 6 行）。一旦服务器收集了所有嵌入，它将继续获取模型表示 $\hat{\Phi}^{t_0}$。随后，它将模型表示分发给所有客户端（第 8-9 行），而 t_0 是最近一次共享嵌入的全局轮次开始时的时间点。在接收到模型表示后，每个客户端需要根据其计算轻量化比率 α_k^t（第 11 行）剪枝特征模型 $\hat{\theta}_k^t$。随后，每个客户端将使用在第 t_0 次迭代中收到的稀疏嵌入及其自身的未剪枝嵌入 $h_k(\hat{\theta}_k^t; \mathbf{x}_k)$ 进行 E 轮本地迭代。来自其他客户端（不包括客户端 k）的嵌入集合表示为 $\hat{\Phi}_{-k}^{t_0}$，所有嵌入集体表示为 $\hat{\Phi}_k^t$。在每次本地迭代中，客户端 k 将使用步长为 η^{t_0} 的小批量 SGD 更新特征模型 $\hat{\theta}_k$（第 15-16 行）。

我们利用逐元素乘积来表示剪枝后的特征模型和特征嵌入。具体来说，剪枝后的特征模型可以表示为 $\hat{\theta}_k^t = \theta_k^t \odot \mathbf{m}_k^t$，其中 θ_k^t 表示客户端 k 的本地模型的原始结构（未进行剪枝），而 $\mathbf{m}_k^t$ 则表示掩码向量，包含在 θ_k^t 中被剪枝参数位置上的零。类似地，对于修剪后的特征嵌入，我们将其表示为 $\hat{h}_k^t(\hat{\theta}_k^t) = h_k^t(\hat{\theta}_k^t) \odot \mathbf{l}_k^t$，其中 $h_k^t(\hat{\theta}_k^t)$ 代表未进行修剪的客户端 k 的初始嵌入，而 $\mathbf{l}_k^t$ 表示对已经修剪掉的参数

Algorithm 1 LVFL

1: **初始化**初始局部模型 θ_k^0 对于所有客户端 k 和服务器模型 θ_0^0。
2: **for** $t = 1, 2, ..., T-1$ **do**
3: **if** $t \mid E = 0$ **then**
4: **for** $k = 1, 2, ..., K$ in parallel **do**
5: 确定 $\hat{h}_k^t(\hat{\theta}_k^t; \mathbf{x}_k)$ 根据 β_k^t
6: 发送 $\hat{h}_k^t(\hat{\theta}_k^t; \mathbf{x}_k)$ 至服务器
7: **end for**
8: 服务器收集模型表示 $\hat{\Phi}^{t_0}$
9: 服务器将 $\hat{\Phi}^{t_0}$ 发送到所有客户端
10: **for** $k = 1, 2, ..., K$ in parallel **do**
11: 根据 α_k^t 剪枝特征模型 $\hat{\theta}_k^t$
12: **end for**
13: **end if**
14: **for** $k = 0, 1, 2, ..., K$ in parallel **do**
15: 获得 $\hat{\Phi}_k^t \leftarrow \left\{\hat{\Phi}_{-k}^{t_0}, h_k(\hat{\theta}_k^t; \mathbf{x}_k)\right\}$
16: 更新特征或头部模型 $\hat{\theta}_k^{t+1}$
17: **end for**
18: **end for**

在 $h_k^t(\hat{\theta}_k^t)$ 中的零掩码向量。在每个全局轮次中，客户端的特征模型和特征嵌入根据计算轻量化比率 α_k^t 和通信轻量化比率 β_k^t 进行调整。接下来我们将详细解释如何分别对特征模型和特征嵌入进行修剪以实现轻量化。

特征模型轻量化在特征模型中，我们采用结构化模型剪枝方法将特征模型从 θ_k^t 调整到 $\hat{\theta}_k^t$，指导原则是计算轻量化比率 α_k^t。这种方法与现有研究一致。为了简化模型并避免引入特定层的超参数，我们在所有层中应用统一的剪枝比率，这符合先前的研究 [16] 的建议。在每一层内，根据重要性分数对滤波器或神经元进行排序，并基于预设的轻量化比率修剪那些得分较低的部分。我们建议使用 l_1 范数来计算这些重要性分数。

特征嵌入轻量化对于特征嵌入，我们采用无结构模型剪枝方法将特征嵌入从 $h_k^t(\hat{\theta}_k^t)$ 精炼到 $\hat{h}_k^t(\hat{\theta}_k^t)$，这一过程由通信轻量化比率 β_k^t 引导。该方法涉及将绝对值最低的权重置零以满足剪枝标准。虽然这种方法保留了嵌入的结构，但通过仅传输非零值显著降低了通信成本。

利用上述机制，客户端可以简化特征模型和特征嵌入，由参数 α_k^t 和 β_k^t 指导。我们的方法旨在根据需求最优地减少模型的复杂性。然而，鉴于结构化剪枝应用于特征模型，必须在每个全局轮次训练开始前重新评估进一步剪枝特征模型的必要性。相反，对于使用非结构化剪枝的特征嵌入，

则无需额外验证步骤。随后，我们详细阐明确定特征模型计算轻量化比率的机制。

1) **如果** $\alpha_k^t > \alpha_k^{t-}$ 其中 $\alpha_k^{t-} = \max\left\{\alpha_k^1, \cdots, \alpha_k^{t-1}\right\}$ 表示在全局轮次 t 之前，客户端 k 的特征模型所达到的最大计算轻量化比率。在这种情况下，进一步执行计算轻量化以满足当前轮次所需的轻量化比率。

2) **如果** $\alpha_k^t \leq \alpha_k^{t-}$ 由于计算轻量化比率已经达成，现有的特征模型已经被充分简化，当前轮次无需进一步进行轻量化。

此迭代过程会重复进行直到训练收敛或满足预设的终止条件。下一节将深入分析 LVFL 算法的收敛界。

V. 收敛性分析

在本节中，我们将深入分析我们的 LVFL 算法的收敛性。首先，我们需要建立一些将在后续讨论中使用的符号和定义。具体来说，我们将定义两种由轻量化机制引起的误差：通信轻量化误差和计算轻量化误差。

通信轻量级错误此误差量化了轻量级特征嵌入在多大程度上准确地逼近原始特征嵌入。它被数学表达为 $\epsilon_k^t := h_k^t(\theta_k^t) - \hat{h}_k^t(\theta_k^t)$。第 t 轮来自客户端 k 的通信轻量级误差平方表示为 $\Omega_k^t \triangleq \mathbb{E}\|\epsilon_k^t\|^2$。

计算轻量级错误此误差评估轻量化特征模型与原始特征模型的相似程度。它定义为 $\varphi_k^t := h_k^t(\theta_k^t) - h_k^t(\hat{\theta}_k^t)$。来自客户端 k 的第 t 轮的平方计算轻量化误差为 $\Psi_k^t \triangleq \mathbb{E}\|\varphi_k^t\|^2$。

令 $\hat{\mathbf{G}}^t$ 为第 t 次迭代的堆叠偏导数：

$$\hat{\mathbf{G}}^t := \left[(\nabla_0 F(\hat{\Phi}_0^t; \mathbf{y}))^T, \ldots, (\nabla_K F(\hat{\Phi}_K^t; \mathbf{y}))^T\right]^T \quad (2)$$

然后全局模型更新为：$\Theta^{t+1} = \Theta^t - \eta^{t_0}\hat{\mathbf{G}}^t$。

我们将 Φ^{t_0} 定义为在未应用计算和通信轻量级错误的情况下，每个客户端在第 t_0 次迭代中接收到的嵌入集合：$\Phi^{t_0} \leftarrow \{\theta_0^t, h_1^t(\theta_1^t), \cdots h_k^t(\theta_k^t)\}$。在这里我们定义 $\Phi_{-k}^{t_0}$ 为来自其他方 $j \neq k$ 的嵌入集合，且我们有 $\Phi_k^{t_0} := \left\{\Phi_{-k}^{t_0}, h_k(\theta_k^t; \mathbf{x}_k)\right\}$。我们的收敛性分析将利用以下关于 VFL 的标准假设。

假设 1 (平滑度)**.** *存在正常数 $L < \infty$ 和 $L_k < \infty$，使得对于所有 Θ_1, Θ_2，目标函数满足：$\|\nabla F(\Theta_1) - \nabla F(\Theta_2)\| \leq L\|\Theta_1 - \Theta_2\|$ 和 $\|\nabla_k F(\Theta_1) - \nabla_k F(\Theta_2)\| \leq L_k\|\Theta_1 - \Theta_2\|$。*

假设 2 (有界黑塞矩阵)**.** *存在正常数 H_k 对于 $k = 0, \ldots, K$，使得对于所有 Θ，F 的二阶偏导数满足：$\left\|\nabla_{h_k}^2 F(\Theta)\right\|_{\mathcal{F}} \leq H_k$。其中 $\|X\|_{\mathcal{F}}$ 是矩阵 X 的弗罗贝尼乌斯范数。*

假设 3 (有界嵌入梯度)**.** *存在正的常数 G_k，对于 $k = 0, \ldots, K$，使得对于所有的 θ_k，嵌入梯度有界：$\|\nabla_{\theta_k} h_k(\theta_k; x_k)\|_{\mathcal{F}} \leq G_k$。*

假设 1 保证了函数的斜率变化平滑，没有任何突然或剧烈的变化。假设 2 控制了函数的曲率，防止其表现出极端或异常的行为。假设 3 管理嵌入的变化，确保它们不会导致严重的波动，从而有助于稳定学习过程。在这些假设下，我们可以得到以下引理。

引理 1. *计算和通信轻量化的对象函数值与不进行计算和通信轻量化时的对象函数值之差的范数有界，如下所示：*

$$\begin{aligned}&\mathbb{E}\|\nabla_k F(\hat{\Phi}_k^t) - \nabla_k F(\Phi_k^t)\|^2 \\ \leq& 2G_k^2 H_k^2 \sum_{j\neq 0}^{K} \Psi_k^{t_0} + 2G_k^2 H_k^2 \sum_{j\neq 0, j\neq k}^{K} \Omega_k^{t_0}\end{aligned} \quad (3)$$

使用引理 1，我们可以详细地限制轻量化错误的影响。随后我们得出定理 1：

定理 1. *在假设 1-3 下，LVFL 的 R 全局轮次上的平均平方梯度有界为：*

$$\begin{aligned}\frac{1}{R}\sum_{t_0=0}^{R-1}\mathbb{E}\|\nabla F(\Theta^{t_0})\|^2 \leq& \frac{3\left[F(\Theta^0) - \mathbb{E}[F(\Theta^T)]\right]}{\eta T} \\ &+ \frac{108E^2}{R}\sum_{t_0=0}^{R-1}\sum_{k=0}^{K} H_k^2 G_k^2 \sum_{j\neq 0}^{K}\Psi_j^{t_0} \\ &+ \frac{108E^2}{R}\sum_{t_0=0}^{R-1}\sum_{k=0}^{K} G_k^2 H_k^2 \sum_{j\neq 0, j\neq k}^{K}\Omega_j^{t_0}\end{aligned} \quad (4)$$

证明可以在在线补充材料的附录中找到。上述收敛边界揭示了几点见解：第一项说明了初始模型与最终模型之间的差异。第二项强调了使用轻量级特征模型所带来的误差，而第三项则代表由特征嵌入轻量化引起的错误。这一分析表明，影响该边界的主要是与通信轻量化和计算轻量化相关的误差。换句话说，更大的误差将导致最终边界更大。为了深入探讨误差与轻量化比率之间的联系，我们必须转向随后的附加假设。

假设 4 (有界嵌入和模型)**.** *存在正常数 $\delta > 0$ 和 $\mu > 0$，使得对于 $k = 0, \ldots, K$，以下条件成立：$\mathbb{E}\left\|h_k(\theta_k; x_k)\right\|^2 \leq \delta^2$，$\mathbb{E}\left\|\theta_k\right\|^2 \leq \mu^2$。*

假设 5 (利普希茨连续)**.** *也存在正常数和 $M_k < \infty$，使得对于所有的 θ_1 和 θ_2 满足：$\|h_k(\theta_1) - h_k(\theta_2)\| \leq M_k\|\theta_1 - \theta_2\|$。*

假设 4 和假设 5 是建立轻量化比率与轻量化误差之间联系所需要的。基于这两个假设，我们能够推导出后续的结果，即推论 1。

推论 1. *在假设 1-5 下，通过计算轻量化比率 α_k^t 和通信轻量化比率 β_k^t，我们可以进一步获得以下界限：*

$$\frac{1}{R}\sum_{t_0=0}^{R-1}\mathbb{E}\|\nabla F(\Theta^{t_0})\|^2 \leq \frac{3\left[F(\Theta^0)-\mathbb{E}[F(\Theta^T)]\right]}{\eta T} + \frac{108E^2\mu^2}{R}\sum_{t_0=0}^{R-1}\sum_{k=0}^{K}H_k^2G_k^2M_k^2\sum_{j\neq 0}^{K}\alpha_j^{t_0} + \frac{108E^2\delta^2}{R}\sum_{t_0=0}^{R-1}\sum_{k=0}^{K}G_k^2H_k^2\sum_{j\neq 0,j\neq k}^{K}\beta_j^{t_0} \tag{5}$$

该结果的详细证明位于在线补充材料中的附录中。根据推论 1 可知，一个需要强调的关键观察是，通信轻量化误差由通信轻量化比率 β_k^t 控制，而计算轻量化误差则受计算轻量化比率 α_k^t 影响。因此，较高的轻量化比率倾向于导致较为宽松的收敛界限，而较低的比率则导致更为严格的收敛界限。

VI. 实验

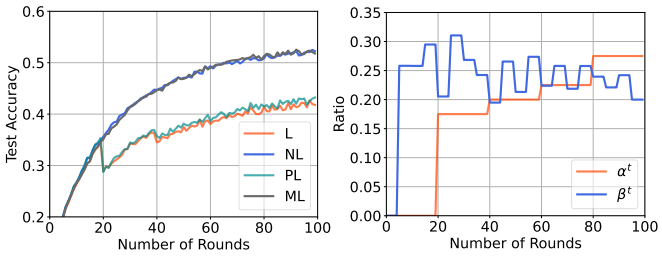
在本节中，我们进行实验以评估 LVFL 的性能。具体而言，我们通过利用基于 VFL 的数据集：CIFAR-10 来探索所提出的算法的有效性。随后，我们将提供有关实验中使用数据集和相应模型的详细信息。

CIFAR-10CIFAR-10 是一个用于图像中对象分类的数据集。在一个特定的训练设置中，有 4 个参与者参与其中，每个参与者负责每张图像的不同象限。训练数据样本的数量 $N = 10000$，批量大小 $b_s = 256$。每个参与者（客户端）使用 VGG16 作为特征模型进行训练，而服务器则专注于训练一个 3 层的全连接网络（FCN）作为头部模型。

在实验中，采用以下方法进行性能比较。**无轻量化 (NL)**，其中不实现任何计算或通信轻量化机制。**计算轻量化仅此而已 (PL)**，仅应用计算轻量化。**通信轻量化仅 (ML)**，仅利用通信轻量化。**轻量化 (L)**，结合计算和通信轻量化的策略以提高效率。接下来我们首先展示上述方法在动态计算和通信轻量化比率下的性能比较。

A. 性能比较

本节展示了优化计算和通信效率的学习性能。在每一轮中，每个客户端将被分配计算轻量化比率 α_k^t 和通信轻量化比率 β_k^t。然后将这些比率应用于个别定制的轻量化过程以适应每个客户端的独特需求。为了说明目的，我们假设所有客户端具有一组统一的比例，具体数值如图 2b 所示。与我们的方法相关的学习性能在图 2a 中进行了说明，从中可以得出几个关键观察结果。主要的是，计算轻量化对学习性能的影响比通信轻量化更为显著。值得注意的是，在实施点，计算轻量化导致测试准确率明显下降，随后逐渐恢复。此外，计算轻量化的频率和强度与其对测试准确性的影响成正比。

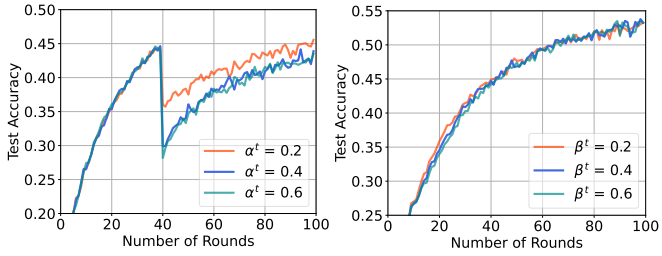


(a) 测试准确性　　(b) 每轮比率

图 2: 每轮次使用 CIFAR-10 的性能比较

B. 比率对学习性能的影响

在本节中，我们进一步研究了选择 α_k^t 和 β_k^t 对学习性能的影响。在这里，我们分别探讨计算和通信的影响。在计算轻量化的情况下，如图 3a 所示，在 $t = 40$ 处对特征模型进行了调整，使用了 $\alpha^t \in [0.2, 0.4, 0.6]$。结果表明，较高的 α^t 值导致测试准确率大幅下降，从而需要更长的恢复期。在通信轻量化的情况下，如图 3b 所示，应用了不同比例的通信轻量化 $\beta^t \in [0.2, 0.4, 0.6]$ 到特征嵌入中。研究结果表明，较低的 β^t 值与更快的收敛速率相关，特别是在范围 $t \in [10, 40]$ 内尤为明显。然而，通信轻量化的影响力不如计算轻量化所观察到的效果显著。



(a) 计算轻量化　　(b) 通信轻量化

图 3: 比率对学习性能的影响

VII. 结论

我们的论文介绍了在智能建筑物联网场景下的轻量级垂直联邦学习（LVFL）概念。由于 VFL 与传统 FL 在结构上的区别，基于 VFL 的轻量化算法设计和收敛性分析显著不同。我们的收敛性证明阐明了收敛界限与通信及计算轻量化的比率之间的关联。此外，实验结果部分强调了采用轻量化机制的好处。在未来的研究中，我们打算将 LVFL

扩展为一个更实用的问题。我们计划制定一个长期优化问题，准确反映客户在整个训练阶段面临的计算和通信困难。这将有助于确定具体的最优轻量化比率。

参考文献

[1] T. J. Castiglia, A. Das, S. Wang, and S. Patterson, "Compressed-vfl: Communication-efficient learning with vertically partitioned data," in *International Conference on Machine Learning*. PMLR, 2022, pp. 2738–2766.

[2] H. Wang and J. Xu, "Online vertical federated learning for cooperative spectrum sensing," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[3] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *arXiv preprint arXiv:1711.10677*, 2017.

[4] L. Yang, D. Chai, J. Zhang, Y. Jin, L. Wang, H. Liu, H. Tian, Q. Xu, and K. Chen, "A survey on vertical federated learning: From a layered perspective," *arXiv preprint arXiv:2304.01829*, 2023.

[5] K. Wei, J. Li, C. Ma, M. Ding, S. Wei, F. Wu, G. Chen, and T. Ranbaduge, "Vertical federated learning: Challenges, methodologies and experiments," *arXiv preprint arXiv:2202.04309*, 2022.

[6] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, "Vertical federated learning," *arXiv preprint arXiv:2211.12814*, 2022.

[7] S. Feng, "Vertical federated learning-based feature selection with non-overlapping sample utilization," *Expert Systems with Applications*, vol. 208, p. 118097, 2022.

[8] J. Sun, Y. Yao, W. Gao, J. Xie, and C. Wang, "Defending against reconstruction attack in vertical federated learning," *arXiv preprint arXiv:2107.09898*, 2021.

[9] Y. Liu, X. Zhang, Y. Kang, L. Li, T. Chen, M. Hong, and Q. Yang, "Fedbcd: A communication-efficient collaborative learning framework for distributed features," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4277–4290, 2022.

[10] M. Li, Y. Chen, Y. Wang, and Y. Pan, "Efficient asynchronous vertical federated learning via gradient prediction and double-end sparse compression," in *2020 16th international conference on control, automation, robotics and vision (ICARCV)*. IEEE, 2020, pp. 291–296.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] Z. Wei, Q. Pei, N. Zhang, X. Liu, C. Wu, and A. Taherkordi, "Lightweight federated learning for large-scale iot devices with privacy guarantee," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3179–3191, 2021.

[13] X. Dong, S. Chen, and S. Pan, "Learning to prune deep neural networks via layer-wise optimal brain surgeon," *Advances in neural information processing systems*, vol. 30, 2017.

[14] V. Sanh, T. Wolf, and A. Rush, "Movement pruning: Adaptive sparsity by fine-tuning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 378–20 389, 2020.

[15] X. Ding, G. Ding, Y. Guo, and J. Han, "Centripetal sgd for pruning very deep convolutional networks with complicated structure," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4943–4953.

[16] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[17] Z. You, K. Yan, J. Ye, M. Ma, and P. Wang, "Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[18] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[19] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: Efficient inference engine on compressed deep neural network," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 243–254, 2016.

[20] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[21] Z. Jiang, Y. Xu, H. Xu, Z. Wang, J. Liu, Q. Chen, and C. Qiao, "Computation and communication efficient federated learning with adaptive model pruning," *IEEE Transactions on Mobile Computing*, 2023.