中国科学: 信息科学 2023 年 第 53 卷 第 3 期: 437–453
SCIENTIA SINICA Informationis
论文

基于进化策略的自适应联邦学习算法

公茂果[1*], 高原[1], 王炯乾[1], 张元侨[1], 王善峰[2], 谢飞[3]

1. 西安电子科技大学智能感知与图像理解教育部重点实验室, 西安 710071
2. 西安电子科技大学网络与信息安全学院, 西安 710071
3. 西安电子科技大学前沿交叉研究院, 西安 710068
* 通信作者. E-mail: mggong@mail.xidian.edu.cn

收稿日期: 2021–05–31; 修回日期: 2021–11–01; 接受日期: 2022–04–02; 网络出版日期: 2023–03–13

国家自然科学基金 (批准号: 62036006, 61973249) 和陕西省重点研发计划 (批准号: 2021ZDLGY02-06) 资助项目

摘要 联邦学习是一种多设备参与的, 保护数据隐私的深度学习技术. 它能够在私有数据不出本地的同时训练全局共享模型. 然而, 在复杂的物联网环境中, 联邦学习面临着统计异构性和系统异构性的挑战. 不同的本地数据分布和高额的通信计算成本, 使得过参数化的模型不适合在物联网应用中直接部署. 同时, 非独立同分布的数据也使采用参数平均聚合的联邦学习更加难以收敛. 联邦学习场景下的研究难点在于, 如何根据私有数据为每个客户端建立个性化的轻量级模型的同时, 把这些模型汇总成为联合模型. 为了解决这一问题, 本文提出了一种基于进化策略的自适应联邦学习算法. 该方法将模型结构进行编码, 把每个参与者视作进化策略中的个体, 通过全局优化来为每个客户端自适应地生成不同的个性化子模型. 客户端根据网络单元重要性和编码在服务器端超网中抽取相应的子网来进行本地更新, 而这种网络局部更新的方法天然契合 dropout 的思想. 在真实数据集上进行的大量实验证明, 本文提出的框架相比于经典的联邦学习方法, 模型性能得到了显著改善. 在客户端数据非独立同分布的情况下, 该算法在有效降低了客户端在通信带宽和计算力受限条件下参与联邦学习门槛的同时, 提高了全局模型的泛化能力.

关键词 联邦学习, 进化策略, 模型编码, 网络剪枝, 本地个性化

1 引言

面对着越来越多的智能设备产生的海量数据, 传统的机器学习方法将所有客户端产生的数据集中在一个计算中心进行模型训练, 而这个过程可能导致用户隐私数据的泄露 [1]. 随着用户隐私意识的提高和相关法律法规的限制, 如欧盟在 2018 年开始执行的通用数据保护条例 [2], 使得智能设备的本地数据与其他设备产生隔离. 数据孤岛现象普遍存在. 为了解决这一问题, 许多研究着眼于设计利用大

引用格式: 公茂果, 高原, 王炯乾, 等. 基于进化策略的自适应联邦学习算法. 中国科学: 信息科学, 2023, 53: 437–453, doi: 10.1360/SSI-2021-0190
Gong M G, Gao Y, Wang J Q, et al. Adaptive federated learning algorithm based on evolution strategies (in Chinese). Sci Sin Inform, 2023, 53: 437–453, doi: 10.1360/SSI-2021-0190

规模分散数据的新方案. 在保护数据隐私的同时, 安全高效地提高联合建模性能. 其中, 多方学习得到了广泛关注 [3]. 该方案在用户数据不出本地的同时联合训练一个深度学习模型, 其主要包括两个步骤: 客户端利用私有数据训练本地模型后, 将本地模型上传给服务器; 服务器将所有客户端上传的本地模型进行聚合, 并将产生的全局模型下发给客户端用于下一阶段的本地更新.

联邦学习可以看作是分布式机器学习的一种特殊形式, 也是未来的发展方向 [4], 二者都使用了分散的数据集和分布式的模型训练. 相比分布式机器学习系统 [5], 联邦学习旨在保护参与者的数据隐私, 并放宽了对私有数据的类型和分布的约束. 联邦学习的研究始于 Federated SGD (FedSGD) [6], 该方法将随机梯度下降 (stochastic gradient descent, SGD) 直接应用于联邦学习中. 客户端在每轮进行一次梯度计算后, 将得到的梯度用于更新服务器上的全局模型. 受 FedSGD 思想的启发, Federated Averaging (FedAvg) [7] 使得用户在上传更新之前可以进行多次梯度下降, 并将训练好的模型参数而非梯度上传到服务器. 该方法显著提高了通信效率, 被认为是联邦学习算法中的典型代表.

然而, 联邦学习面临着用户私有数据异质性与全局模型泛化性之间的矛盾. 具体而言, 由于用户的环境和工作模式的不同, 不同设备产生的数据通常是非独立同分布的 (non-IID), 因此需要构建一个过参数化的模型来学习这些不同的分布 [8]. 而过于庞大和复杂的全局模型会带来一个困境, 即尽管这类联邦学习方案对于通信带宽和设备计算能力有着较高的要求, 它在特定用户私有数据上的拟合能力甚至不如未采用联合训练的本地模型. 相比传统的集中式训练, 联邦学习很难学习到所有数据的特性, 从而导致聚合后的全局模型性能出现偏差. FedAvg 被证明能够处理部分 non-IID 数据. 但是, 面对高度偏斜的数据分布时, 每个客户端的本地数据只能更新到网络的少部分参数, 而其他未被更新的参数在上传到服务器后同样参与模型聚合, 会对其他客户端上传的更新了这部分参数的模型造成影响, 因此 FedAvg 的性能会显著下降 [9]. 避免这些未更新的参数在聚合中干扰已更新的参数能够有效提高联邦模型性能和收敛速度. 此外, 不同设备受计算力、存储和通信等方面的资源限制, 过参数化模型的推理和应用变得十分困难. 这个问题在服务器和用户部署同样架构模型的场景中普遍存在. 一种有效的解决方法是为每个客户端提供不同架构的模型 [10, 11] 来适配私有数据, 从而能够有效处理 non-IID 数据并在本地测试中提高精度. 然而, 由于联邦学习需要建立一个鲁棒的全局模型, 如何将这些不同结构的个性化模型有效聚合的问题仍未得到解决.

为了解决上述挑战, 本文提出了一种基于进化策略的自适应联邦学习算法, 通过自适应剪枝为拥有 non-IID 数据的客户端生成个性化模型, 避免了未训练参数干扰服务器端模型聚合, 从而大幅提高全局模型与本地模型性能, 同时有效降低通信和计算成本. 具体地说, 与传统的联邦学习框架相比, 本文引入了超网 – 子网的架构来进行模型的本地个性化适配. 该框架将过参数化的超网部署在服务器中, 每个客户端根据私有数据分布在超网中提取子网来进行本地训练, 并将更新后的子网知识汇集到超网上. 针对每个客户端子网的结构, 本算法采用进化策略对其进行编码及优化. 具体而言, 子网由超网经过剪枝得到, 通过衡量每个网络单元的重要性并结合网络编码的剪枝率, 选择对私有数据最有价值的网络单元进行保存, 形成适配私有数据分布的最优模型结构. 该方法能够为每一个客户端训练不同的个性化子网, 有效地解决了不同客户端统计异质性和设备异构型的问题, 并大幅节省了设备的计算和通信成本. 同时, 子网通过 dropout 的方式为超网提供高质量的更新, 避免了冗余参数的干扰, 提高了全局模型的泛化能力.

本文的主要贡献是首次提出一种基于网络剪枝的个性化联邦学习框架, 引入超网 – 子网的设计, 为应对联邦学习遇到的挑战提供了新方案; 由于联邦学习天然契合进化计算中的并行扩展性, 本文将联邦学习任务中的每个参与者视作种群中的个体, 利用进化策略来驱动每个客户端优化本地模型架构. 在不显著提升本地计算量的同时, 对子网的设计进行高效的无梯度随机优化; 借鉴网络训练中 dropout

的思想, 在超网中提取不同的子网并分发给对应的客户端进行训练, 避免了全局模型在 non-IID 私有数据的情况下受到未训练冗余参数的干扰, 提高了本地模型的拟合效果和全局模型的泛化能力, 同时显著加快了模型的收敛速度和推理速度.

2 相关工作

2.1 联邦学习

在联邦学习中, 为实现同一优化目标, 多个参与者共同训练一个深度学习模型, 而无需共享自己的私有数据. 联邦学习通常可以分为跨设备的场景和跨机构的场景, 跨机构联邦学习又可以根据样本 ID 空间和特征空间的重合程度进一步分为横向、纵向和迁移式的学习方法. 本文主要针对跨设备的联邦学习和横向跨机构的联邦学习, 即基于服务器上模型参数平均聚合的联邦学习方法, 而 FedAvg 算法 [7] 是其中的典型代表. 服务器随机选取 K 个客户端在本地多次执行 SGD 算法, 第 k 个客户端计算当前模型 w_t 的梯度 $g_k = \nabla F_K(w_t)$ 并进行梯度下降来优化模型参数. 本地客户端的模型权重更新表示为

$$w_{t+1}^k \leftarrow w_t - \eta g_k, \tag{1}$$

其中, η 代表预设的学习率. 多次更新后, 客户端将本地模型参数上传服务器进行平均聚合, 以得到全局模型. 第 t 轮通信服务器的模型聚合更新为

$$\overline{w}_{t+1} \leftarrow \sum_{k=1}^{K} \frac{N_k}{N} w_{t+1}^k, \tag{2}$$

其中, N 和 N_k 分别代表全部和第 k 个客户端上的样本数. 相比较 FedSGD 算法, FedAvg 算法能够在更少的通信轮数内训练出更高质量的模型.

然而, FedAvg 算法在 non-IID 数据上容易发生权重偏差, 导致全局模型性能显著下降, 收敛速度大幅降低. 这不仅给理论分析带来难度, 而且对联邦学习的算法设计提出了挑战. 为了应对客户端数据统计异质性问题, Li 等 [12] 提出了 FedProx 算法, 该算法对 FedAvg 算法的参数做了修改, 允许每个参与设备执行可变次数的 SGD 算法. 同时, 它允许出现未完成训练的局部模型. 不论一个局部模型是否完成了训练, FedProx 会整合所有参与训练的局部模型. 局部模型的目标函数采用了损失函数加近端项的方法, 对偏离全局模型太多的局部模型进行惩罚. 在高度异构的环境中, FedProx 展现出比 FedAvg 更加稳定的收敛性. Zhao 等 [8] 发现, 框架精度的降低与客户端模型之间的权重偏差有直接关系, 权重偏差可以通过每个客户端的本地数据分布与整体数据分布之间的 Wasserstein 距离来量化, 因此他们提出通过将一部分共享数据分发给所有客户端来减少 non-IID 环境下客户端数据的分布差异. 针对统计异质性问题, 现有的联邦学习方法通常弱化客户端模型之间的差异性来提高聚合得到联邦模型时的稳定性. 但是, 这些约束往往会导致联邦模型更慢收敛, 且不同客户端差异性的削弱不利于全局模型在客户端上的个性化适配. 因此, 我们不对客户端模型施加约束, 而是考虑保留每个本地模型训练过程贡献较大的神经元, 将几乎不被更新的神经元删除, 避免其对其他客户端更新过的参数造成影响, 从而在保证客户端拥有个性化模型的同时, 提高联邦模型单元对特征的提取效用.

2.2 网络剪枝

近年来, 深度神经网络受到了越来越多的关注, 其被应用到不同的领域并在许多任务中取得了显著的精度提高. 这部分性能提升通常依赖于具有数百万甚至数十亿个参数的过参数化深层网络. 其需

要巨大的存储和计算开销的特点严重阻碍了资源受限环境下的模型推理、部署和应用 [13]. 因此, 大量工作开始研究将模型部署到设备之前对其进行压缩. 此方向的探索引出了量化 [14]、剪枝 [15]、蒸馏 [16] 等一系列模型压缩技术. 网络剪枝是一种通过减小神经网络规模提高运算效率的方法. 剪枝技术能够删除模型中不重要的网络单元, 从而降低网络训练过程中的计算成本, 提高压缩后的神经网络运行速度. 在将过参数化的模型部署到联邦学习中的边缘设备上时, 网络剪枝的作用更加明显 [17].

剪枝可以分为结构化剪枝 [18] 和非结构化剪枝 [19]. 结构化剪枝主要包括通道剪枝、层剪枝等; 非结构化剪枝一般指权重剪枝. 在结构化剪枝中, 通过某种策略衡量通道或神经元对输出的重要性, 并把重要性小的通道和神经元删除. 该方法剪枝粒度较大, 可以得到一个参数少、内存小、计算量低的稠密网络. 在非结构化剪枝中, 通常对神经网络中权重的大小进行排序, 并去掉低于预设阈值的连接, 从而得到剪枝后的网络. 最终网络的权重大部分为零, 该方法对压缩模型体积有明显的作用. 在联邦学习中, 通过对不同客户端的本地模型进行剪枝, 不仅能够有效解决本地模型过参数化导致的训练和推理速度慢、内存占用量大等问题, 并且缓解了服务器与客户端之间的通信压力. Zhu 等 [20, 21] 提出多目标进化联邦学习算法, 为种群中的每一个个体进行模型结构的染色体编码, 并利用进化算法来优化得到适合该联邦学习场景的最佳网络结构. 然而, 由于每个个体在种群的一次迭代中都要完整地执行一次联邦学习过程来进行适应度的评估, 受限于传统进化算法在计算量上的约束, 该方法在实际场景中的应用受到一定限制, 无法带来推理速度的提高和内存、通信成本的减少. 与该方法侧重于为联邦学习寻找最优的模型结构不同, 本文着眼于在现有模型结构下利用剪枝策略使得全局模型和本地模型同时达到最佳的性能. 因此, 我们探索一种将联邦视作种群, 将客户端视作个体的方案, 在遵循传统联邦学习流程的同时, 将进化优化引入其中, 同时保持本地模型的个性化和全局模型的鲁棒性. 进化策略为该方法提供了可行性.

2.3 进化策略

进化策略 (evolutionary strategy, ES) [22] 是模仿生物进化的一种求解参数优化问题的方法, 属于进化计算 [23] 的范畴. 它将变异、重组和选择应用于包含候选解的个体群体中, 以便迭代地进化出越来越好的解. 进化策略可以应用于所有优化领域, 包括连续、离散、无约束和有约束的组合搜索空间以及混合搜索空间 [24]. 进化策略使用实数值对优化问题进行编码, 并主要采用变异和选择作为搜索运算符. 对于实值搜索空间, 通过在每个基因上进行均值为零的正态分布采样来产生新的个体. 一个新解可以记作

$$x = \mu + \sigma y,\ y \sim N(0, C), \tag{3}$$

其中, μ 为决定搜索区域的均值, σ 为决定搜索范围大小和强度的步长参数, C 为决定搜索方向相对尺度的协方差矩阵. 进化策略在搜索中通常反复迭代以下步骤: 采样产生一个或一组候选解; 对新产生的解计算对应的目标函数值; 根据目标函数值选择部分解; 使用选择的解更新分布参数. 对于最常见的一种形式, (1+1)-ES 在每次迭代中产生一个新解, 通过和父代进行比较, 保留较好的一个解成为下一次迭代的父代, 将另一个淘汰, 并相应地调节分布参数.

(1+1)-ES 形式简单, 便于分析, 通常集中在局部进行搜索, 并能得到良好的性能. 对于所求解的联邦学习问题, 引入进化策略的目的是求得适配客户端私有数据分布的本地模型. 为了在不增加客户端计算压力的同时逐渐收敛到更优解, 本文将使用 (1+1)-ES 方案. 在不引入大量额外计算的前提下, 每个客户端根据本地私有数据来优化模型架构, 从而迭代生成体积小精度高的子模型.

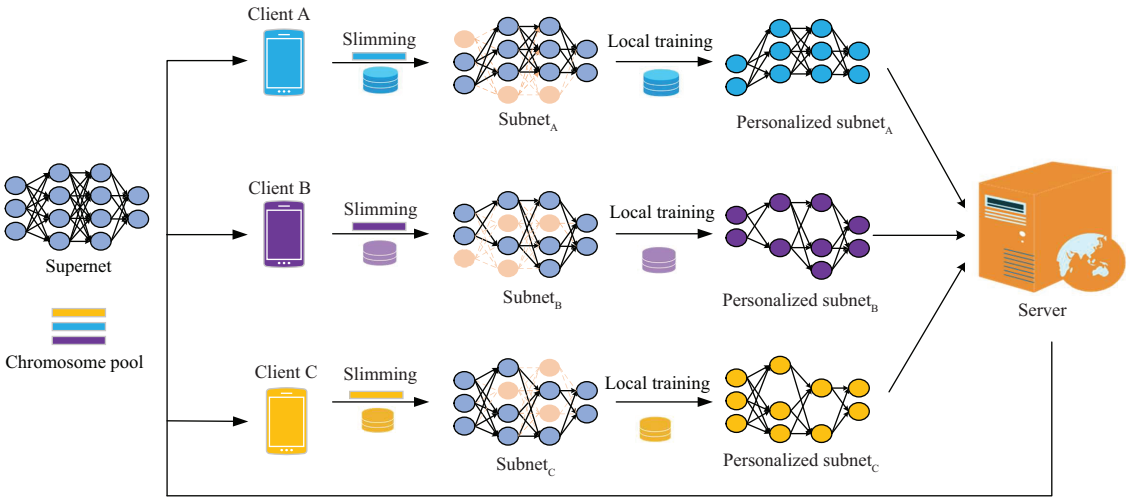

图 1 (网络版彩图) 基于进化策略的自适应联邦学习框架. (1) 服务器将超网和个体的染色体编码发送给每个客户端; (2) 客户端执行染色体引导的本地网络剪枝与子网生成; (3) 客户端选择子代染色体和对应子网发送给服务器进行迭代更新

Figure 1 (Color online) Adaptive federated learning framework based on evolutionary strategies. (1) The server distributes the supernet and individual chromosome to each client; (2) clients conduct local network slimming and subnet derivation associated with chromosome; (3) clients select offspring chromosome and the subnet to the server for iteratively updating

3 基于进化策略的自适应联邦学习

3.1 问题定义

定义 K 个客户端, 每个客户端都有一个只能在本地存储和处理的私有数据集. 联邦学习的目标是从多客户端间分散的数据集中学习一个全局模型. 以非凸神经网络优化为例, 本地模型的目标函数 $f(w)$ 可以定义为

$$\min_{w\in\mathbb{R}^d} f(w), \quad \text{where} \quad f(w)\overset{\text{def}}{=}\frac{1}{N}\sum_{i=1}^{N} f_i(w), \tag{4}$$

其中, 模型参数 w 对样本 (x_i, y_i) 进行预测的损失表示为 $f_i(w)=\ell(x_i, y_i; w)$. 因此, K 个客户端进行联邦学习联合建模的目标函数为

$$f(w)=\sum_{k=1}^{K}\frac{N_k}{N}F_k(w), \quad \text{where} \quad F_k(w)=\frac{1}{N_k}\sum_{i=1}^{N_k} f_i(w). \tag{5}$$

本地模型个性化能够使模型性能显著改善 [25] 的同时, 有效解决联邦学习中的统计异质性和模型异构性问题. 因此, 本文将设计模型自适应的联邦学习算法, 为每个客户端分配个性化模型, 同时为服务器上的全局模型提供可用的高质量更新. 在不降低全局模型性能的前提下, 为每个客户端动态地产生适配其资源和应用需求的本地模型, 并降低客户端的通信成本和计算成本, 使得模型能够被部署在计算资源受限和需要快速响应时间的设备上.

3.2 模型自适应的联邦学习

为了解决上述问题, 本文提出一种自适应的联邦学习框架, 如图 1 所示. 具体地说, 在服务器端部署一个过参数化的超网, 以便学习到所有客户端多样的私有数据分布. 为了提高模型的本地个性化程

度, 每个客户端都可以建立与其私有数据分布适配的子网. 为了缓解通信和计算压力, 通过剪枝的压缩方法减小子网体积. 子网由超网通过结构化剪枝和非结构化剪枝得到, 其参数继承自超网中对应的网络单元. 剪枝操作被编码进子网对应的染色体中, 并使用 (1+1)-ES 对其进行优化.

每一轮通信中, 服务器将超网和个体的染色体分发给对应的客户端, 客户端根据本地私有数据分布计算超网中各个网络单元的重要性, 并结合染色体中编码的剪枝率, 对超网进行结构化剪枝得到子网. 由于子网只保留对本地私有数据最有价值的部分网络单元, 对应的模型参数对网络的预测影响较大. 而网络更新和反向传播也只在这部分最重要的网络单元上进行, 从而增强了网络对数据的拟合能力. 在父代和子代染色体上进行相同操作, 得到两个子网. 通过训练这两个子网并进行评估, 将适应度更大的子网和对应染色体上传到服务器. 服务器接受所有的染色体后, 更新进化策略参数并得到下一代染色体; 同时, 将子网按原位置聚合到超网上, 完成对超网的更新. 不同客户端分别优化超网的不同区域, 减弱了超网上网络单元之间形成复杂的内在互相关性, 同时避免了未经训练的冗余参数在聚合过程中对超网的干扰, 从而增强了超网的泛化能力和鲁棒性. 服务器端和客户端的更新流程分别如算法 1 和 2 所示.

算法 1 Server update procedure

Input: Communication rounds T, K clients indexed by k;
Output: Supernet $\mathcal{M}$;
Initialize supernet $\mathcal{M}$ and chromosome pool $C^1 = [c_1, c_2, \ldots, c_K]$;
for each communication round $t = 1, 2, \ldots, T$ **do**
  Send supernet $\mathcal{M}$ and chromosome c_k^t to each client k;
  for each client k **in parallel do**
    Client k conducts local updates;
    Receive subnet $\mathcal{S}_k^t$ and updated chromosome c_k^t from client k;
  end for
  Generate offspring chromosome pool C^{t+1} from C^t;
  Update $\mathcal{M}$ with all subnets $\mathcal{S}^t$;
end for
return supernet $\mathcal{M}$.

算法 2 Client update procedure

Input: Chromosome c^{t-1} and c^t, supernet $\mathcal{M}$, private training dataset $\mathbb{D}_T$, and validation dataset $\mathbb{D}_V$;
Output: Subnet $\mathcal{S}$;
Receive supernet $\mathcal{M}$ and chromosome c^t from the server;
Derive subnet $\mathcal{S}^{t-1}$ using parental chromosome c^{t-1} and $\mathcal{M}$ based on $\mathbb{D}_V$;
Derive subnet $\mathcal{S}^t$ using offspring chromosome c^t and $\mathcal{M}$ based on $\mathbb{D}_V$;
Train subnets $\mathcal{S}^{t-1}$ and $\mathcal{S}^t$ on $\mathbb{D}_T$;
Subnet $\mathcal{S} = \max_{\mathcal{S}}\ (\text{fitness}(\mathcal{S}^{t-1}, \mathbb{D}_V),\ \text{fitness}(\mathcal{S}^t, \mathbb{D}_V))$;
Update chromosome with higher fitness and keep it locally as c^t;
Send $\mathcal{S}^t$ and c^t to the server;
return subnet $\mathcal{S}$.

3.3 联邦进化策略

进化策略作为经典高效的随机优化方法, 其并行扩展性天然契合联邦学习的组织模式. 对于每一个客户端, 将网络的逐层剪枝率进行实数编码来形成该个体的染色体, 以将其作为该客户端的剪枝引

导. 以卷积神经网络为例, 每一个卷积层的结构化剪枝率对应染色体中的一个基因, 每一个全连接层的结构化剪枝率和非结构化剪枝率对应染色体上的两个基因. 因此染色体长度 $L = N_c + N_f$, 其中 N_c 为网络中卷积层的数目, N_f 为网络中全连接层的数目. 由于客户端本地数据是固定的, 因此超网中单元重要性也得以确定, 则染色体编码与剪枝得到的子网结构是一一对应的. 通过优化染色体中的结构化剪枝率, 构建适配本地数据的网络结构并提高模型的运算速度; 通过优化非结构化剪枝率, 在不影响模型精度的前提下减少上游通信成本. 进化策略作为客户端剪枝率的优化算法, 作为 plus-and-play 的独立模块插入联邦学习框架中, 保持了原有的学习流程, 而 FedAvg 算法可以看作本文方法在剪枝率恒为 1 时的特殊情况.

在应用联邦进化策略之前, 对染色体的更新方法进行分析. 由于每个客户端的私有数据分布不同, 其形成的最优子网络有所差异, 即每个个体的优化目标不同, 给基于全局优化目标的群优化算法的应用带来困难. 然而, 虽然总的剪枝率与私有数据分布复杂程度对应, 网络各层的剪枝率之间有着内在的联系. 对于一种复杂的数据分布, 各层可能都需要较小的剪枝率来保证网络的学习性能; 而对于简单的数据分布, 可以对每一层采用更大的剪枝率来删除冗余参数. 另外, 神经网络的每一层将执行不同的功能, 如底层网络通常负责提取通用的底层特征, 不能删掉过多单元; 而上层网络负责融合与任务相关的模式, 可以根据输出类别进行结构调整. 这种剪枝率之间固有的内在联系可以被归纳并用于指引优化过程. 因此, 可以引入协方差矩阵 C 来表征染色体上不同基因之间的相关性 [26], 并通过反复迭代调整一个正态分布来进行搜索. 进化策略的核心是对正态分布的均值、步长和协方差矩阵进行调整, 来达到较好的搜索效果, 从而算法能够使得产生更好解的概率逐渐增大, 即沿好的搜索方向进行搜索的概率增大. 所有客户端共享同一个协方差矩阵而拥有自适应的步长, 在宏观上保持神经网络各层剪枝率内在联系的同时, 为不同客户端提供了个性化的模型搜索. 由于每个个体的子目标不同, 将每一代个体搜索过程的起点, 即分布均值 μ, 修正为父代染色体. 在第 t 轮种群生成过程中, 第 k 个子代染色体可以表示为

$$x_k^{t+1} = x_k^t + \sigma_k^t y^{t+1},\ y^{t+1} \sim \mathcal{N}(0, C^t), \tag{6}$$

其中, 协方差矩阵 C 决定着分布的形状和变量之间的依赖关系, 其更新公式为

$$C^t = \frac{1}{K}\sum_{k=1}^{K} y_k^t\ {y_k^t}^{\mathrm{T}}, \tag{7}$$

其中, y_k^t 为竞争胜出保留下来的的搜索方向. 参数 σ_k^t 控制着分布范围的搜索步长, 而不同的子目标使得所有个体相同步长的设置不再适用. 因此, 为每个客户端自适应地调节 σ_k^t 可以加快到达最优值的收敛速度. 我们沿用了协方差自适应调整的进化策略 CMA-ES [27], 在生成子代时引入了历史搜索信息来调整步长, 即评估路径 p_σ, 其描述了分布均值的移动, 并且将个体在前几轮中移动的搜索方向进行加和来构建评估路径, 使得这些方向中相反的方向分量相互抵消, 相同的分量则进行叠加, 起到了神经网络优化中 Momentum 的作用, 因此评估路径代表了最好的搜索方向之一, 为染色体生成子代提供了有效引导. 在这个过程中, 超网和子网的权重参数也在不断更新, 因此需要给最近的子代搜索方向赋予更高的权重. 设置学习率 α_σ, 使用指数滑动平均来更新评估路径, 并平衡权重以使 p_σ 在更新前后保持共轭性并服从 $\mathcal{N}(0, I)$. 由式 (6) 可得

$$y^{t+1} = \frac{x_k^{t+1} - x_k^t}{\sigma_k^t} \sim C^{t(\frac{1}{2})}\mathcal{N}(0, I), \tag{8}$$

$$C^{t(-\frac{1}{2})} \cdot \frac{x_k^{t+1} - x_k^t}{\sigma_k^t} \sim \mathcal{N}(0, I). \tag{9}$$

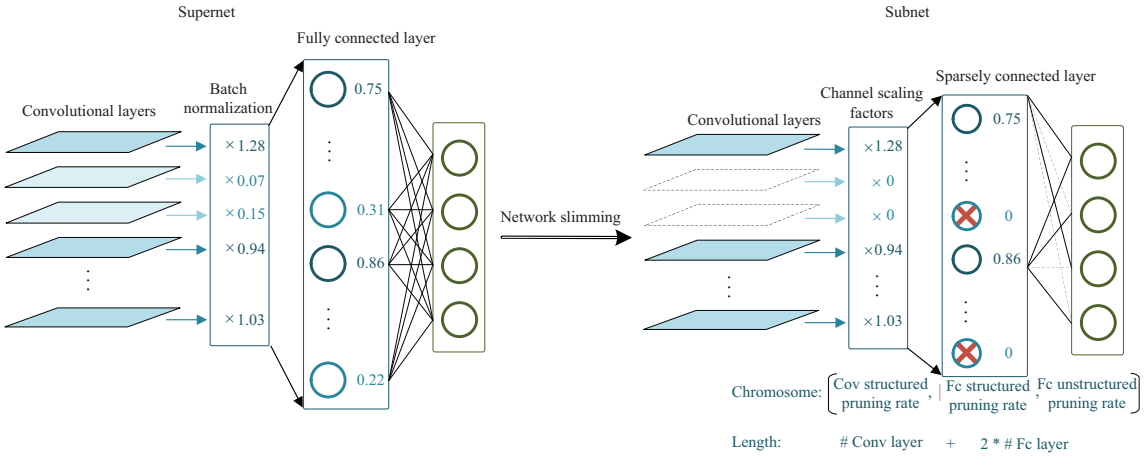

图 2 (网络版彩图) 染色体引导的本地网络剪枝与子网生成, 其中网络每一层的剪枝率被编码进染色体中

Figure 2 (Color online) An illustration of local network slimming and subnet derivation associated with the chromosome, where slimming rate of each layer in the network is encoded into the chromosome

评估路径 p_σ 的更新公式记作

$$p_\sigma^{t+1} = (1-\alpha_\sigma)p_\sigma^t + \sqrt{1-(1-\alpha_\sigma)^2}C^{t(-\frac{1}{2})}\ \frac{x_k^{t+1}-x_k^t}{\sigma_k^t}. \tag{10}$$

通过将连续多步搜索保留下来的方向和与随机搜索的方向和进行比较, 可以对应地调节 σ_k^t 的值, 即

$$\sigma_k^{t+1} = \sigma_k^t \cdot \exp\left(\alpha_\sigma\left(\frac{||p_\sigma^{t+1}||}{\mathbb{E}||\mathcal{N}(0,I)||}-1\right)\right), \tag{11}$$

从而完成子代的生成, 并将其与超网共同分发给客户端.

子代的选择过程在客户端进行. 客户端根据子代和父代的染色体生成两个不同结构的子网, 分别记作 $\mathcal{S}^t$ 和 $\mathcal{S}^{t+1}$. 客户端训练两个子网, 并选择性能较好的一个保留. 在每一轮选择中采用贪婪的策略, 以子网在本地验证集上的准确率作为其适应度, 即

$$\text{fitness}(\mathcal{S}) = \text{Acc}(\mathcal{S}, \mathbb{D}_V), \tag{12}$$

其中, $\mathbb{D}_V$ 代表本地验证集. 若子网 $\mathcal{S}^t$ 与 $\mathcal{S}^{t+1}$ 的准确率相等, 则根据模型结构和剪枝率计算模型体积, 并保留体积较小的子网, 将对应的染色体作为子代上传到服务器, 从而完成联邦进化策略的迭代.

3.4 染色体引导的子网生成

为了与本地私有数据相适配, 客户端根据染色体编码的逐层剪枝率, 结合超网单元对本地数据的重要性, 对超网进行逐层剪枝来得到个性化的子网, 这一染色体引导的子网生成过程如图 2 所示. 在该过程中, 由于每个客户端的本地数据假设为固定不变, 每一轮中超网单元的重要性也是确定的, 因此其根据一个染色体剪枝得到的子网是唯一的, 即每个染色体与子网一一对应. 由于染色体的更新和遗传都是在服务器端进行的 (见 3.3 小节), 客户端只需在接收到染色体后更新子网和计算适应度, 而不参与进化策略的参数更新过程. 首先, 在本地预留的验证数据集上对超网中的网络单元进行重要性评估. 具体地说, 对于卷积层中的每个通道, 将批归一化 (batch normalization, BN) 层中与通道对应的缩放因子 γ 作为这一通道的重要性判别标准; 对于全连接层中的每个神经元, 将其激活函数的输出作

表 1 卷积神经网络的模型架构

Table 1 Model architectures of CNNs

Architecture	MNIST	CIFAR-10	FEMNIST
Convolutional	64, pool, 128, pool, 256, pool	64, pool, 192, pool, 384, 256, 256, pool	64, pool, 128, pool, 256, pool, 512, 512
Fully-connected	1024, 1024, 10 (input size: 4096)	4096, 1024, 10 (input size: 4096)	1024, 1024, 62 (input size: 8192)
Conv/FC/all params	369.2 K/5.2 M/5.6 M	2.3 M/20.9 M/23.2 M	443.5 K/9.5 M/9.9 M
Conv/FC/all FLOPs	29.4 M/5.2 M/34.6 M	166.9 M/20.9 M/187.9 M	10.2 M/9.5 M/19.7 M

为神经元的重要性判别标准. 将验证数据集输入超网, 并记录每一层中可训练网络单元对于模型输出的重要程度. 由于将权重清零不会节省任何空间, 因此将删除固定百分比的网络单元并建立新的网络. 根据染色体中对应于每一层的剪枝率, 剪掉重要程度最低的部分网络单元, 并根据剪枝比例记录保留的网络单元索引, 从而构建相应体积的子网并将超网中对应位置的网络参数迁移到子网, 以便进行后续的训练和上传. 非结构化剪枝在子网训练结束后进行, 从而进一步压缩网络体积.

4 实验结果与分析

4.1 实验设置

4.1.1 数据集

我们在 3 个图像数据集上进行实验, MNIST [28] 数据集由 0∼9 个手写数字组成, 训练集中有 60000 个手写数字, 测试集中有 10000 个手写数字. 每张图片都是 28×28 的单通道灰度图像. FEMNIST [29] 数据集有 62 类图像, 包括 10 种数字和 52 种大小写英文字母, 它是流行的 MNIST 数据集的更复杂的扩展版本. 在 non-IID 的设置中, 同一作者的手写体数据被分到一个客户端上. CIFAR-10 [30] 数据集是一个更接近真实对象的小型彩色图像数据集, 有 50000 个训练图像和 10000 个测试图像. 它包含 10 类 RGB 三通道彩色图像, 每个图像大小为 32×32, 每个类别有 6000 幅图像.

4.1.2 模型

表 1 展示了不同数据集对应的卷积神经网络 (CNN) 结构、参数量和计算量, 其中 Conv 为卷积层, 全称为 convolutional layer; FC 为全连接层, 全称为 fully-connected layer. 对于 MNIST 和 CIFAR-10 数据集, 实验沿用了类似 VGG 结构的卷积网络, 并使用 3×3 的卷积层, 2×2 的池化层和全连接层来构建超网. 每个卷积操作采用了 Conv-ReLU-BN 的顺序. 对于 FEMNIST 数据集, 本文使用了一个高效的 MobileNet [31]. 相比传统 CNN 中的标准卷积, 该网络采用深度可分离卷积来实现模型轻量化. 它将一个标准卷积分解为深度卷积和 1×1 的逐点卷积, 对每个输入通道使用单一的卷积核, 利用两步处理来大幅减少模型计算量. 在本文设置中, MobileNet 中卷积层的计算量与全连接层的计算量处于同一量级, 有利于其在资源受限的边缘设备上的部署和应用, 实验将测试本框架在该网络上的有效性.

4.1.3 实验细节

对于所有实验, 将 80% 的客户端用于训练, 20% 的客户端用于测试. 而对于训练客户端上的样本, 同样将 80% 的样本用于训练, 其余样本用于测试本地模型性能. 对于本文提出的框架, 客户端上 10%

的训练样本被划分为验证集, 以作为进化策略的评价标准. 这部分验证集不是固定的, 而是在每一轮训练中随机产生. 实验构建了一个包含 20 个客户端的联邦学习场景, 将 MNIST 数据集对应的客户端样本数设为 800, CIFAR-10 数据集对应的客户端样本数设为 1600, 且每个客户端上随机分配两类数据来模拟 non-IID 场景. 对于 FEMNIST 数据集, 随机选取 100 个不包含所有样本类别标签的客户端. 实验将传统的联邦学习算法 FedAvg 作为基线算法, 且本地模型在上传到服务器之前, 将在测试数据上进行评估. FedAvg 可以看作本文算法在剪枝率固定为 1 时的特殊情况. 同样选取了针对个性化联邦学习的 Per-FedAvg [32] 算法和针对联邦学习中模型体积压缩的 T-FedAvg [33] 进行对比. 前者在模型不可知元学习的基础上, 为所有客户端找到可以快速适应用户本地数据的初始化模型, 并允许客户端针对其私有数据执行一个或多个梯度下降步骤来适配本地数据集. 根据文献 [33] 中的最佳实验结果, 最大本地更新次数 τ 被设为 10, 元学习的本地更新步长为 0.004. 后者提出一种联合训练的三元量化算法, 将浮点权重量化到三值权重, 通过自学习量化因子来优化客户端上的量化网络以减少推理阶段的计算量, 并提出一种三元联邦平均协议来减少联邦学习的上下游通信. 本文在 3 个数据集上对于不同的网络结构进行了广泛的实验, 其中网络结构、客户端数量、学习率等超参数均与本文保持一致. 通过与 FedAvg 算法在精确度、收敛速度、参数量和计算量等方面进行对比, 验证所提出的自适应联邦学习框架相比于 FedAvg 算法的优越性, 以及在应对统计异构性和系统异构性的挑战时的有效性. 实验被重复 5 次并报告平均值以确保其准确性.

4.2 收敛速度对比

图 3 展示了本文提出的算法与 FedAvg 在不同数据集上的收敛速度对比, 以探究本地剪枝对模型收敛的影响. 其中, FedAvg_G (虚线) 为全局模型在测试客户端上的准确率, FedAvg_L (实线) 为本地模型在训练客户端的测试样本上的准确率. 由于在 MNIST 和 CIFAR-10 数据集的 non-IID 设置中, 只给每个客户端分配了两类数据, 因此本地模型的测试准确率要显著优于全局客户端. 在 MNIST 数据集上, 两种算法的本地模型都达到了 98% 以上的准确率, 本文算法略优于 FedAvg. 由于数据分布和模型结构较为简单, 算法对于 MNIST 的准确率和收敛速度影响较小. 在 CIAFR-10 数据集上, 本文算法相对 FedAvg 取得了明显的性能提升. 由于在本地模型训练前进行了结构化剪枝, 可训练参数减少, 模型结构更加紧凑, 对私有数据特征的提取能力更强. FedAvg 算法中模型可训练参数更多, 所有客户端采用同一个全局模型来拟合各自不同的数据分布, 模型参数之间形成了内在的约束关系, 使得本地模型在面对私有数据时产生了部分性能下降. 在 FEMNIST 数据集上, MobileNet 的使用使得全连接层占据了大部分参数量和接近一半的计算量. 另外, 由于数据类别较多而本地私有数据类别和样本量较少, 网络过参数化现象更为严重. 因此, 本算法的本地模型在训练中期相比 FedAvg 取得了较大的性能提升.

在 CIFAR-10 和 FEMNIST 数据集上, 本算法全局模型的收敛速度显著优于 FedAvg, 对于后者第 50 轮的测试精度已经达到 FedAvg 算法在第 150 轮的准确度, 甚至取得了可与 FedAvg 局部模型比拟的性能. 由于 CNN 中不同通道负责提取不同的特征, 其与各类样本的关联性也不同. 对于本地的 non-IID 数据, FedAvg 算法中过参数化的模型可能只有部分参数得到有效更新. 然而, 所有参数都将参加服务器的聚合过程, 这部分未经训练的参数会对其他客户端上传的已被更新的参数产生负面影响. 本文提出的算法只保留全局网络中与私有数据关联性大的最有价值的网络单元进行更新, 被删除的参数不参与服务器的聚合. 这一操作避免了未经训练的冗余参数对全局模型的干扰, 并显著加快了全局模型的收敛速度. 同时, 由于每个客户端私有数据分布不同, 所保留的网络单元也不同. 客户端对超网中的不同子网进行更新, 减缓了超网中通道间复杂的内在约束和联系, 有效提高了全局模型的泛

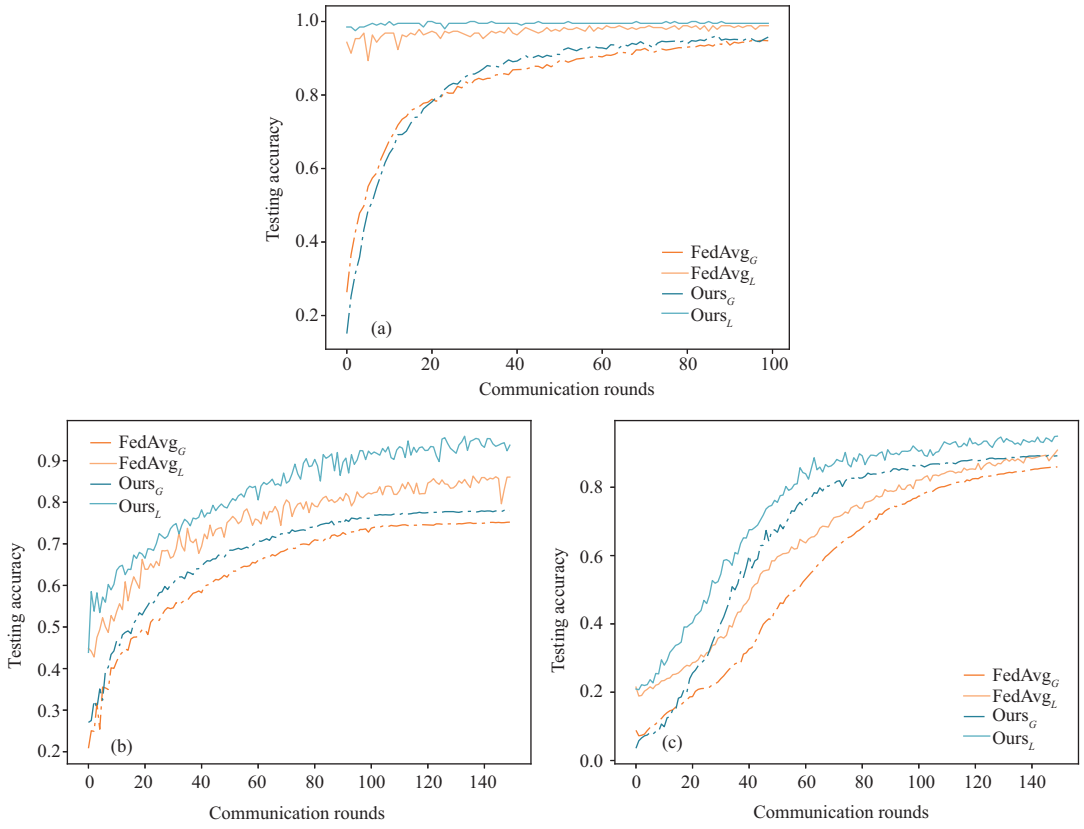

图 3　(网络版彩图) 3 个数据集上的测试精度

Figure 3　(Color online) Testing accuracy on three datasets. (a) MNIST; (b) CIFAR-10; (c) FEMNIST

化性能.

4.3　层级剪枝率分析

为了探究进化策略对卷积层和全连接层不同优化结果的影响, 实验对 MNIST 数据集上 CNN 的逐层剪枝率进行了分析. 图 4 展示了层级剪枝率对参数量和计算量的影响. 可观察到, 该网络的计算量主要集中在第二和第三卷积层, 而参数主要集中在第一全连接层. 在该网络中, 第一卷积层, 即输入层, 是样本与后续网络的通道. 该层卷积核较少, 且通常负责提取对各类数据通用的底层特征, 破坏该层结构将导致模型性能下降, 因此该层的剪枝比例最小. 第二和第三个卷积层对模型计算量影响最大, 分别有 128 和 256 个通道. 由于进化策略的适应度函数向着参数量和计算量更小的方向优化剪枝率, 因此这两个卷积层中的大部分通道被裁剪.

全连接层同时将结构化剪枝率和非结构化剪枝率编码进染色体, 进一步减少参数量并压缩模型体积. 在第一全连接层中, 参数量被压缩为原来的 1%, 因此在客户端与服务器的模型传输过程中, 稀疏化压缩等多种高效的压缩方法可以被利用, 从而降低通信成本. 由于引入了非结构化剪枝, 全连接层的参数量相比计算量在剪枝后减少了更多. 类似于输入层, 第三全连接层作为输出层, 神经元和权重的重要性更大, 且对模型体积压缩几乎没有影响, 因此该层在剪枝前后变动较小.

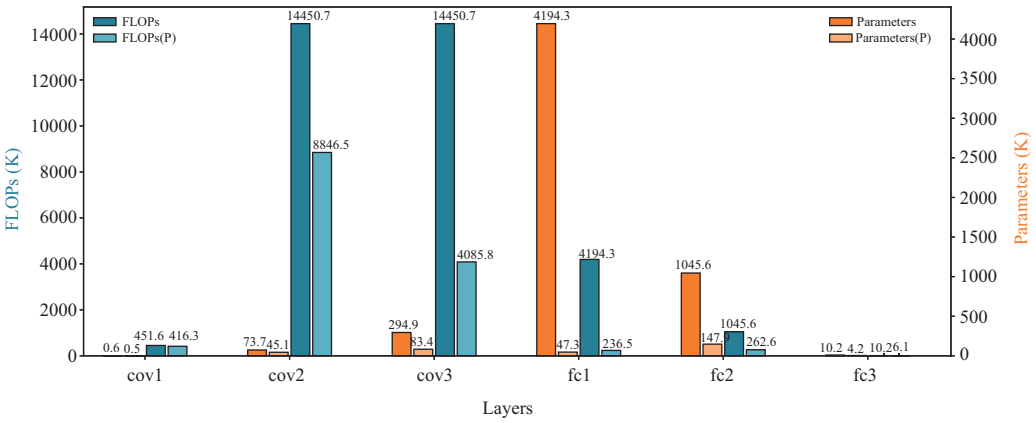

图 4　(网络版彩图) 层级剪枝率对参数量和计算量的影响

Figure 4　(Color online) Effect of layer-wise pruning rate on the quantity of parameters and FLOPs

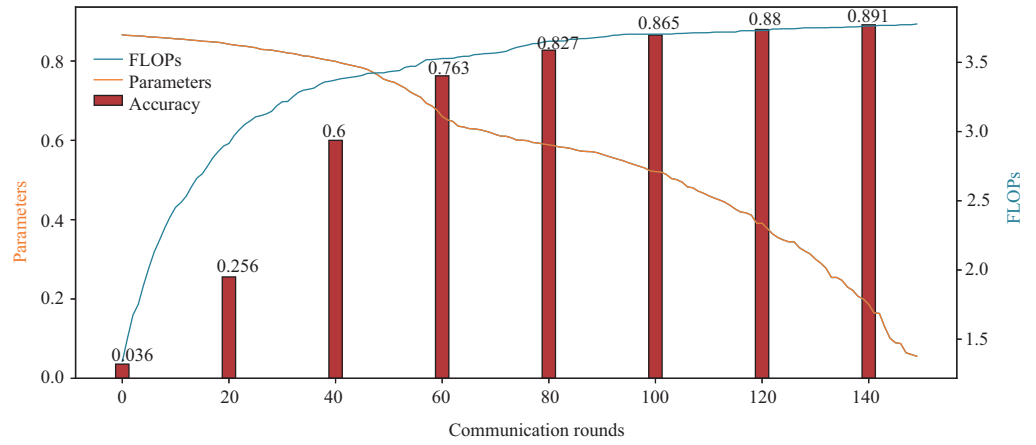

图 5　(网络版彩图) 迭代过程中模型准确性、压缩率和加速比的变化

Figure 5　(Color online) Model accuracy, compression ratio and speedup rate during iterations

4.4　压缩率和加速比研究

图 5 展示了随着迭代的进行, MobileNet 在 FEMNIST 数据集上的模型准确性、压缩率和加速比的变化. 可以观察到, 全局模型的精确度随着通信轮数的增加逐渐上升并最终趋向于收敛, 而计算量的变化也呈现出这种趋势. 然而, 模型的压缩率始终保持下降, 这意味着剪枝逐渐由卷积层向全连接层偏移. 即, 模型趋于收敛后, 受到所设计适应度函数和选择机制的驱动, 进化策略会在不影响性能的前提下进一步对全连接层进行剪枝并减少参数量, 从而提升压缩比, 降低训练过程中客户端和服务器间的通信成本.

为了全面比较自适应的联邦学习框架和传统的 FedAvg 算法, 实验在 3 个数据集上进行了准确度、压缩率和加速比的对比研究, 如表 2 所示. 本文提出的框架相比 FedAvg 在全局模型精度和本地模型精度都有着不同程度的提升, 且显著减少了模型的参数量和计算量. 该框架在所有数据集上均取得了 5% 左右的模型压缩率, 该数值与数据集和模型复杂程度呈正相关. 考虑到该框架对模型收敛速度的加快, 模型达到收敛所需的上游通信成本将进一步减小. 另外, 将 3 个模型的运算速度均提升两

表 2 联邦学习算法的性能比较

Table 2 Performance comparison of federated learning algorithms

		Acc_G (%)	Acc_L (%)	Parameter	FLOPs
MNIST	FedAvg	94.83	98.85	5.62 M	34.61 M
	Per-FedAvg	95.21	99.44	5.62 M (100%)	34.61 M (1×)
	T-FedAvg	95.04	98.39	0.36 M (6.36%)	34.61 M (1×)
	Ours	96.23	99.50	0.39 M (7.11%)	13.85 M (2.49×)
		Acc_G (%)	Acc_L (%)	Parameter	FLOPs
CIFAR-10	FedAvg	74.80	86.31	23.23 M	187.9 M
	Per-FedAvg	76.35	95.76	23.23 M (100%)	187.9 M (1×)
	T-FedAvg	75.12	86.88	1.44 M (6.27%)	187.9 M (1×)
	Ours	77.61	95.89	1.59 M (6.86%)	70.92 M (2.65×)
		Acc_G (%)	Acc_L (%)	Parameter	FLOPs
FEMNIST	FedAvg	85.98	90.86	9.94 M	19.73 M
	Per-FedAvg	87.36	95.23	9.94 M (100%)	19.73 M (1×)
	T-FedAvg	86.49	91.15	0.53 M (5.32%)	19.73 M (1×)
	Ours	89.41	94.95	0.42 M (4.18%)	5.14 M (3.83×)

倍以上, 在 FEMNIST 数据集所使用的轻量化网络 MobileNet 上甚至取得了近 4 倍的提升. 这一实验结果验证了本文算法在相比非剪枝策略模型各方面性能上的优越性.

对于两个先进的对比算法, Per-FedAvg 主要解决模型个性化问题. 由于本地模型可能经过多次训练, 得到了更为高质量的更新, 从而在全局模型上取得了优于 FedAvg 的效果. 而本文算法通过着重更新重要性高的网络单元, 在本地模型上与 Per-FedAvg 性能接近. T-FedAvg 主要通过三元量化将更新后的模型进行压缩, 以此减少上游和下游的通信成本. 它对于本地模型和全局模型性能提高不明显, 且对模型的训练和推理没有加速作用. 由于本文算法中客户端需要在每一轮通信中评估网络全部单元的重要性, 因此无法在下游通信中压缩模型体积. 而在上游通信, 即客户端向服务器上传过程中, 与 T-FedAvg 的通信成本处于同一量级. 综上, 本文提出的算法可以同时达到与先进的个性化算法接近的准确率和与压缩算法接近的压缩率, 另外, 在全局模型性能和模型加速比两个衡量指标上都有着显著提升.

4.5 消融实验

在进化策略的更新中, 协方差矩阵的调整能够让算法沿更好的搜索方向进行搜索的概率增大, 使得算法产生更好解的概率增大. 为了验证协方差矩阵对加速搜索进程的有效性, 我们将协方差矩阵 (记作 Cov) 替换为高斯噪声矩阵 (记作 Gaus) 进行实验, 即将每一次搜索方向随机化, 来观察模型在协方差矩阵与高斯噪声矩阵下收敛速度、压缩率与准确率的对比. 实验在 CIFAR-10 数据集上进行, 实验结果如图 6 所示.

如图 6 所示, 对于本地个性化模型, 协方差矩阵会带来部分性能提升, 这表现在本地模型的测试准确率、参数量和计算量上. 其中, 参数量和计算量的提升较为显著, 即协方差矩阵有助于在不损失精度的情况下取得更好的压缩效果, 并更快地搜索到个性化子网络. 在联邦模型上, 协方差矩阵同样在训练阶段有着明显的加速效果, 训练结束后, 两种对比方法得到的联邦模型性能接近, 而此时本地模型在参数量和计算量上仍然有差距. 这说明联邦模型的性能提升主要得益于不同子网的类 dropout

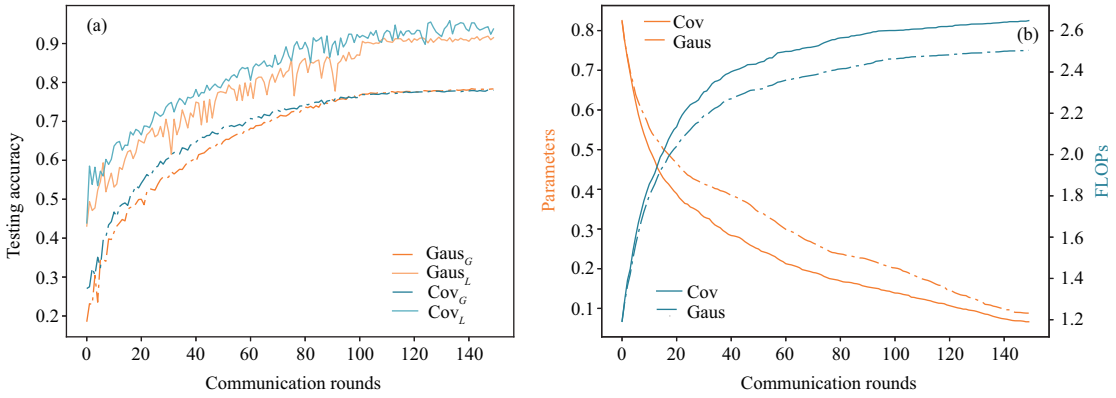

图 6 (网络版彩图) 协方差矩阵对搜索进程影响的消融实验

Figure 6 (Color online) Ablation study on the effect of covariance matrix on searching process. (a) Testing accuracy; (b) FLOPs and parameters

聚合方式, 而受优化方法和子网结构影响较小. Dropout 式的分发训练和模型聚合保证了子网只更新与数据关联性强的参数, 有效提高了联邦模型的鲁棒性.

4.6 剪枝与结构搜索

在上面的实验中, 本文提出的基于进化策略的自适应联邦学习框架相比经典的联邦平均 FedAvg 算法取得了显著的性能提升. 并且, 自适应联邦学习框架通过模型剪枝在本地模型测试准确率上达到了与个性化联邦学习 Per-FedAvg 接近的性能, 且取得了与专注于压缩模型体积的 T-FedAvg 媲美的通信效率. 作为同样改变模型结构的方法, 网络结构搜索 (NAS) 同样能够在联邦学习场景提高模型性能和通信效率. 因此, 在本小节中, 与两个经典的联邦学习网络结构优化方法: FedEA [20] 和 FedNAS [34] 进行对比, 来探究联邦学习场景下剪枝与网络结构搜索在性能和效率方面的区别.

FedEA 在联邦学习中使用多目标进化算法来优化神经网络模型的结构, 以同时最小化通信开销和全局模型测试误差. 为了提高深层神经网络进化的效率, 在联邦学习中采用了一种可扩展的网络连通性编码方法. 其中, 进化算法中使用到的交叉、变异、稀疏度等参数均遵循原文中的设置, 种群大小和演化次数被设为 20, 客户端数量、通信轮数等参数与本文一致. FedNAS 提出了基于神经网络架构搜索的自动化联邦学习方法, 来自动设计模型架构以提高模型精度并减少手动设计工作, 并帮助分散的客户端合作搜索具有更高精度的体系结构. FedNAS 采用文献 [20] 中的搜索单元结构, 客户端数量、通信轮数、本地更新次数与本文一致.

由于 FedEA 算法会进行多次联邦学习流程来优化模型结构, 我们取最后一次演化中性能最佳的个体作为 FedEA 算法的最终结果. 本文自适应剪枝方法不对联邦模型结构做出改变, 因此在 FedEA 算法得到优化结果后, 我们沿用搜索出的模型架构和学习率等超参数, 使用本文自适应剪枝方法在其基础上再次运行联邦学习, 记作 Ours(EA). 由于 FedNAS 采用了可微分架构搜索, 最终得到的结构无法进行常规剪枝操作, 因此我们仅与该算法进行对比, 并未在得到最终结构后将我们的算法应用于其上. 即, FedAvg 与 Ours 采用同样的架构和参数, FedNAS 与之参数相同, 架构不同. Ours(EA) 沿用了 FedEA 的最后一次演化的搜索结果中的架构和参数. 实验结果如表 3 所示.

在性能方面, 与所有对比算法相比, FedAvg 都有不同程度的提高. FedNAS 在模型聚合的过程中将网络单元的参数和选择概率同时集中在了服务器上进行平均, 这种单元概率平均的方式也使得其准确率得到提升. 经过多次演化, FedEA 仅靠搜索到的模型架构和超参数便将性能提高了 2%, 证明了自

表 3 剪枝和结构搜索性能对比

Table 3 Performance comparison of pruning and NAS algorithms

	MNIST		CIFAR-10		FEMNIST	
	Accuracy (%)	Efficiency (h)	Accuracy (%)	Efficiency (h)	Accuracy (%)	Efficiency (h)
FedAvg	94.56	2	75.08	11	86.12	6
FedNAS	96.11	1	77.32	6	88.82	4
FedEA	97.03	79	78.13	268	89.53	114
Ours	96.25	3	77.95	18	89.47	10
Ours(EA)	97.74	1 (+79)	79.46	9 (+268)	89.77	4 (+114)

动架构搜索相比手动预设的优越性. 在此基础上, 本文方法利用 FedEA 搜索到的架构参数, 进一步取得了超越所有对比算法的性能提升. 由于 FedEA 已对网络结构进行了优化, Ours(EA) 对 FedEA 的性能提升相比 Ours 对 FedAvg 的性能提升较小.

在效率方面, 我们在 NVIDIA Tesla V100 GPU 集群上进行实验, 并记录算法的总 GPU 时间. 由于 FedNAS 在训练过程中同时优化网络结构, 因此训练和通信用时最短. 由于 FedEA 每次搜索都要进行一次联邦学习来评估适应度, 即使有着早停策略, 其耗时依然最长. 本文方法与 FedAvg 算法相比效率仅达到一半, 因为子代在客户端上的评估同样需要先经过模型训练. FedEA 的多目标优化提高了模型性能和通信效率, 压缩了模型体积, 因此在其搜索结果上重新训练的 Ours(EA) 相比原始的 FedAvg 算法效率得到提升, 达到与 FedNAS 相同的量级, 且与 FedNAS 相比取得了显著的性能提升. 需要注意的是, Ours(EA) 达到该性能的总运行时间需将 FedEA 的运算考虑在内, 即表中括号内容. 由此可得, 本文方法遵循传统联邦聚合学习进程, 会受到超网结构的影响, 可作为联邦学习结构搜索算法的后续优化方法.

5 结论

本文提出了基于进化策略的自适应联邦学习框架, 引入了超网 – 子网的架构来进行模型的本地个性化适配. 通过在服务器端部署过参数化的超网, 每个客户端根据私有数据分布在超网中提取子网来进行本地训练. 该算法对每个客户端子网的结构进行编码并采用进化策略进行优化. 子网由超网经过剪枝得到, 通过衡量每个网络单元的重要性并结合网络编码的剪枝率, 使得对私有数据最有价值的网络单元被保存下来, 从而形成适配私有数据分布的最优模型结构. 子网的更新以 dropout 的模式汇集到超网上, 不同子网的参数聚合提升了超网的泛化性和鲁棒性. 本文在多个数据集上进行了广泛实验, 验证了所提出的算法在全局和本地模型精度、收敛速度、计算量和参数量等方面都有着显著提升. 在未来的工作中, 将探究其他的模型架构优化方法, 从而进一步提高该框架的可解释性. 同时, 将研究该框架在循环神经网络上的性能和表现, 使其更具有通用性.

参考文献

1 Chen X Y, Gao Y Z, Tang H L, et al. Research progress on big data security technology. Sci Sin Inform, 2020, 50: 25–66 [陈性元, 高元照, 唐慧林, 等. 大数据安全技术研究进展. 中国科学: 信息科学, 2020, 50: 25–66]

2 Yang Q, Liu Y, Chen T J, et al. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol, 2019, 10: 1–19

3 Gao Y, Gong M G, Xie Y, et al. Multiparty dual learning. IEEE Trans Cybern, 2022. doi: 10.1109/TCYB.2021.3139076

4 Phong L T, Phuong T T. Privacy-preserving deep learning via weight transmission. IEEE Trans Inform Forensic Secur, 2019, 14: 3003–3015

5 Verbraeken J, Wolting M, Katzy J, et al. A survey on distributed machine learning. ACM Comput Surv, 2021, 53: 1–33

6 Chen J M, Monga R, Bengio S, et al. Revisiting distributed synchronous SGD. 2016. ArXiv:1604.00981

7 McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the Machine Learning Research, 2017. 1273–1282

8 Zhao Y, Li M, Lai L Z, et al. Federated learning with non-IID data. 2018. ArXiv:1806.00582

9 Sattler F, Wiedemann S, Müller K R, et al. Robust and communication-efficient federated learning from non-i.i.d. data. IEEE Trans Neural Netw Learning Syst, 2020, 31: 3400–3413

10 Wang L, Yoon K J. Knowledge distillation and student-teacher learning for visual intelligence: a review and new outlooks. IEEE Trans Pattern Anal Mach Intell, 2022, 44: 3048–3068

11 Lin T, Kong L J, Stich S, et al. Ensemble distillation for robust model in federated learning. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 2351–2363

12 Wang L P, Wang W, Li B. CMFL: mitigating communication overhead for federated learning. In: Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems, 2019. 954–964

13 Deng L, Li G Q, Han S, et al. Model compression and hardware acceleration for neural networks: a comprehensive survey. Proc IEEE, 2020, 108: 485–532

14 Gong R H, Liu X L, Jiang S H, et al. Differentiable soft quantization: bridging full-precision and low-bit neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition, 2019. 4852–4861

15 Liu Z, Sun M J, Zhou T H, et al. Rethinking the value of network pruning. 2018. ArXiv:1810.05270

16 Li D L, Wang J P. FedMD: heterogenous federated learning via model distillation. 2019. ArXiv:1910.03581

17 Molchanov P, Mallya A, Tyree S, et al. Importance estimation for neural network pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 11264–11272

18 Jiang C H, Li G Y, Qian C, et al, Efficient DNN neuron pruning by minimizing layer-wise nonlinear reconstruction error. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018. 2298–2304

19 Li G Y, Jiang C H, Qian C, et al, Optimization based layer-wise magnitude-based pruning for DNN compression. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018. 2383–2389

20 Zhu H Y, Jin Y C. Multi-objective evolutionary federated learning. IEEE Trans Neural Netw Learning Syst, 2020, 31: 1310–1322

21 Zhu H Y, Zhang H Y, Jin Y C. From federated learning to federated neural architecture search: a survey. Complex Intell Syst, 2021, 7: 639–657

22 Gong M G, Li H, Meng D Y, et al. Decomposition-based evolutionary multiobjective optimization to self-paced learning. IEEE Trans Evol Computat, 2019, 23: 288–302

23 Gong M G, Jiao L C, Yang D D, et al. Research on evolutionary multi-objective optimization algorithms. J Softw, 2009, 20: 271–289 [公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究. 软件学报, 2009, 20: 271–289]

24 Salimans T, Ho J, Chen X, et al. Evolution strategies as a scalable alternative to reinforcement learning. 2017. ArXiv:1703.03864

25 Hu R, Guo Y X, Li H N, et al. Personalized federated learning with differential privacy. IEEE Internet Things J, 2020, 7: 9530–9539

26 Hansen N, Ostermeier A. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: Proceedings of IEEE International Conference on Evolutionary Computation, 1996. 312–317

27 Hansen N, Müller S D, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolary Comput, 2003, 11: 1–18
28 Deng L. The MNIST database of handwritten digit images for machine learning research [Best of the Web]. IEEE Signal Process Mag, 2012, 29: 141–142
29 Caldas S, Wu P, Li T, et al. Leaf: a benchmark for federated settings. 2018. ArXiv:1812.01097
30 Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. 2009. https://www.cs.toronto.edu/~kriz/cifar.html
31 Howard A G, Zhu M L, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. 2017. ArXiv:1704.04861
32 Fallah A, Mokhtari A, Ozdaglar A. Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 3557–3568
33 Xu J J, Du W L, Jin Y C, et al. Ternary compression for communication-efficient federated learning. IEEE Trans Neural Netw Learning Syst, 2022, 33: 1162–1176
34 He C Y, Annavaram M, Avestimehr S. Towards non-IID and invisible data with FedNAS: federated deep learning via neural architecture search. 2020. ArXiv:2004.08546

Adaptive federated learning algorithm based on evolution strategies

Maoguo GONG[1*], Yuan GAO[1], Jiongqian WANG[1], Yuanqiao ZHANG[1], Shanfeng WANG[2] & Fei XIE[3]

1. *Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an* 710071*, China*;
2. *School of Cyber Engineering, Xidian University, Xi'an* 710071*, China*;
3. *Academy of Advanced Interdisciplinary Research, Xidian University, Xi'an* 710068*, China*
* Corresponding author. E-mail: mggong@mail.xidian.edu.cn

Abstract Federated learning is a deep learning technique that ensures data privacy with multiple device participation by training a globally shared model while storing private data locally. However, in a complex Internet-of-Things (IoT) environment, federated learning faces challenges of statistical heterogeneity and systematic heterogeneity. Because of different local data distributions and high communication costs, over-parameterized models are unsuited for direct deployment in IoT applications. Moreover, nonindependent, identically distributed data make federated learning with parameter-averaging aggregation more difficult to converge. Determining how to build personalized lightweight models for each client based on individual data and then aggregate these models has become a research problem with regard to federated learning. To solve this problem, we propose an adaptive federated learning algorithm based on an evolution strategy. The method regards each participant as an individual by encoding the model architecture through an evolution strategy, and it can adaptively generate a different customized subnet for each client through global optimization. According to the importance of the network unit and genotype, clients extract the corresponding subnets from the server-side supernets and perform local updates, which naturally fits the idea of the dropout. Extensive experiments on real-world datasets demonstrate that the proposed framework considerably improves the model performance compared with conventional federated learning. In particular, when the local data is not independent and uniformly distributed, the framework facilitates clients with limited communication bandwidths and computing power to participate in federated learning; the generalization ability of the global model is improved.

Keywords federated learning, evolution strategy, model encoding, network pruning, local customization