基于边缘的联邦学习模型清洗和设备聚类方法

刘 艳[1),2),3)] 王 田[1),2)] 彭绍亮[4),5)] 王国军[6)] 贾维嘉[1),2)]

[1)](北京师范大学人工智能与未来网络研究院 广东 珠海 519000)

[2)](北京师范大学-香港浸会大学联合国际学院广东省人工智能与多模态数据处理重点实验室 广东 珠海 519000)

[3)](华侨大学计算机科学与技术学院 福建 厦门 361021)

[4)](湖南大学信息科学与工程学院 长沙 410000)

[5)](国家超级计算长沙中心 长沙 410000)

[6)](广州大学计算机科学与网络工程学院 广州 510000)

摘 要 参与联邦学习的终端设备只需在各自的本地数据集上训练本地模型,并在服务器的协同下共同训练一个全局预测模型.因此,联邦学习可以在不共享终端设备的隐私和敏感数据的情况下实现机器学习的目的.然而,大量终端设备对服务器的高并发访问会增加模型更新的传输延迟,并且本地模型可能是与全局模型收敛方向相反的恶意模型,因此联邦学习过程中会产生大量额外的通信成本.现有工作主要集中在减少通信轮数或清除本地脏数据,本文研究了一种基于边缘的模型清洗和设备聚类方法,以减少本地更新总数.具体来说,通过计算本地更新参数和全局模型参数在多维上的余弦相似度来判断本地更新是否是必要的,从而避免不必要的通信.同时,终端设备根据其所在的网络位置聚类,并通过移动边缘节点以簇的形式与云端通信,从而避免与服务器高并发访问相关的延迟.本文以 Softmax 回归和卷积神经网络实现 MNIST 手写数字识别为例验证了所提方法在提高通信效率上的有效性.实验结果表明,相比传统的联邦学习,本文提出的基于边缘的模型清洗和设备聚类方法减少了 60%的本地更新数,模型的收敛速度提高了 10.3%.

关键词 联邦学习;移动边缘计算;模型清洗;聚类;余弦相似度

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2021.02515

Edge-Based Model Cleaning and Device Clustering in Federated Learning

LIU Yan[1),2),3)] WANG Tian[1),2)] PENG Shao-Liang[4),5)] WANG Guo-Jun[6)] JIA Wei-Jia[1),2)]

[1)](*Institute of Artificial Intelligence and Future Networks, Beijing Normal University, Zhuhai, Guangdong* 519000)

[2)](*Guangdong Key Lab of AI and Multi-Modal Data Processing, Beijing Normal University-Hong Kong Baptist University United International College, Zhuhai, Guangdong* 519000)

[3)](*College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian* 361021)

[4)](*College of Computer Science and Electronic Engineering, Hunan University, Changsha* 410000)

[5)](*National Supercomputing Center in Changsha, Changsha* 410000)

[6)](*School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou* 510000)

Abstract The end devices participating in federated learning train the local model on their local datasets and collaboratively learn a global prediction model with the server, so federated learning can achieve the purpose of machine learning without sharing private and sensitive data. In fact, federated learning typically takes several iterations between the terminal device and the cloud server to reach the target accuracy. Therefore, when a large number of end devices communicate with servers, the limited network bandwidth between the server and terminal devices will inevitably

收稿日期:2020-09-15;在线发布日期:2021-06-22.本课题得到国家自然科学基金(62172046)、福建省自然科学基金(2020J06023)、UIC 科研启动项目(R72021202)、广东省教育厅普通高校重点领域专项项目(2021ZDZX1063)和珠海市产学院合作项目(ZH22017001210133PWC)资助.**刘 艳**,硕士研究生,主要研究方向为边缘计算. E-mail: lyliuxixi@163.com. **王 田**(通信作者),博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为物联网、边缘计算, E-mail: cs_tianwang@163.com. **彭绍亮**,博士,教授,中国计算机学会(CCF)杰出会员,主要研究领域为高性能计算、大数据和人工智能等. **王国军**,博士,教授,中国计算机学会(CCF)杰出会员,主要研究领域为网络信息安全、物联网和云计算等. **贾维嘉**,博士,教授,IEEE Fellow,主要研究领域为智慧城市和分布式系统等.

lead to a large model transmission delay. In addition, due to the heterogeneity of end devices and the non-independent identically distributed characteristics of local data, local models may be malicious models that converge in the opposite direction to the global model. These models not only poison the accuracy of the global model but increase additional communication costs. Therefore, reducing network occupancy and improving the communication efficiency of federated learning becomes crucial. The existing research mainly focuses on reducing communication rounds or cleaning dirty data from local. One of the studies is to calculate the number of identical symbolic parameters between the global model and the local update to determine the importance of the local update, ultimately reducing the communication rounds. It only considers the difference in the direction of model parameters and does not consider the parameter deviation between the global model and the local model. Different from the existing work, this paper proposes an edge-based model cleaning and device clustering method to reduce the number of local updates. Specifically, we calculate the cosine similarity between the local update parameters and the global model parameters to determine whether the local update is necessary to be uploaded. If the cosine similarity between the two is less than the set threshold, the update will not be uploaded to the server for global aggregation, thereby avoiding unnecessary communication. Meanwhile, end devices clustered according to their network locations and communicate with the cloud in the form of clusters through mobile edge nodes, thereby avoiding the delay associated with high concurrent access to the server. Considering that model updates usually contain large gradient vectors, a large amount of data needs to be transferred by mobile edge nodes, which may increase model transmission time. Therefore, before the model is transferred to the cloud, the mobile edge node uses its computing resources to first aggregate the local updates in the cluster and then transmits the aggregated cluster model to the cloud server for global aggregation. Each edge aggregation consumes computing resources from mobile edge nodes, and each global aggregation consumes network communication resources. This paper takes Softmax regression and convolutional neural networks to realize MNIST handwritten digit recognition as an example to verify the effectiveness of the proposed method in improving communication efficiency. The experimental results show that compared with the traditional federated learning, the edge-based model cleaning and device clustering method proposed in this paper reduces the local update of the Softmax regression model by 60%, and the convergence speed of the model increases by 10.3%.

Keywords federated learning; mobile edge computing; model cleaning; clustering; cosine similarity

1 引 言

由于行业竞争和保护数据隐私的结果，在大多数行业中，数据往往以孤岛的形式存在. 即使在同一家公司，不同部门之间的数据整合也面临着巨大的阻力，更不用说整合来自各个机构的数据，这在现实中几乎是不可能的. 此外，随着大数据的进一步发展，对数据隐私和安全的重视已成为全球趋势[1]. 因此，通过把终端数据发送到云端进行深度学习的传统机器学习方式面临着极大的挑战. 作为人工智能(Artificial Intelligence，AI)[2]的核心技术，联邦学习(Federated Learning，FL)[3]是解决这一挑战的一种很有前途的方法. 在 FL 的学习过程中，由云服务器维护的全局模型为所有终端设备共享，终端设备仅需利用其本地数据集训练全局模型，并将训练好的本地更新上传到云服务器参与全局聚合，然后不断迭代这一过程[4]. 联邦学习的整个过程都没有涉及到数据的传输，因而它保护了数据的隐私和安全，并且在保护数据隐私的情况下实现了机器学习(Machine Learning，ML)的目的[5].

然而，FL 的通信效率仍然面临许多挑战[6]. 一

方面，部署在终端设备上的高级 ML 应用程序越来越多地使用复杂的神经网络，因此本地更新通常包含较大的梯度向量. 相比之下，终端设备与云服务器之间的网络通常存在两个问题：(1) 网络的带宽有限，且高带宽服务的服务器成本昂贵；(2) 本地和云端之间的网络连接具有不对称特性：网络的上行速度通常比下行速度慢很多. 因此，当大量的终端设备参与联邦学习时，对服务器的高并发访问势必会增加模型传输的通信延迟，网络的不稳定也会导致训练瓶颈[7]. 另一方面，参与 FL 的设备存在异构性，参与训练的本地数据往往是服从非独立同分布(Non-Independently Identically Distribution，Non-IID)的[8]，因此这些设备和数据训练的本地模型往往是差强人意的，这些本地模型可以称之为脏模型. 如果来自脏模型的本地更新被发送到云端参与聚合，这不仅严重影响全局模型的精度，还增加了额外的通信成本. 因此，减少 FL 的网络占用变得至关重要.

针对上述问题，本文讨论了如何有效地利用边缘的计算和通信资源来获得最佳的 FL 性能. 本文考虑了一个典型的移动边缘计算架构，在不同的网络位置部署移动边缘节点[9]，使其成为远程云和终端设备通信的枢纽，如图 1 所示. 终端设备根据其局域网(Local Area Network，LAN)地址被划分为多个簇. 在本地更新阶段，每个终端设备计算本地更新参数和全局模型参数之间的余弦相似度，如果两者之间的相似性较低，则认为该本地模型是恶意或不必要的更新而不上传到服务器参与聚合，从而避免了额外的通信成本. 由于每次全局聚合不仅消耗网络通信资源，还会消耗云计算资源. 因此，在边缘聚合阶段，由部署在每个局域网中的移动边缘节点收集和聚合必要或非恶意的本地更新，然后将边缘聚合后的模型发送到云服务器进行全局聚合，从而避免与服务器高并发访问相关的延迟，并合理的利用移动边缘节点的计算资源，缓解云计算的压力.

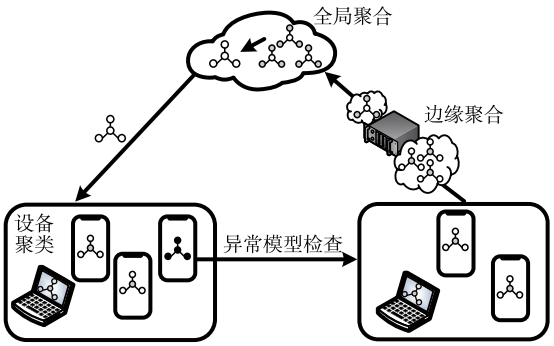


图 1　系统框架

本文的主要贡献如下：

(1) 本文提出了模型清洗的方法，通过计算全局模型参数和本地更新参数的余弦相似度来比较参数值和模型收敛方向上的差异，并以此差异作为度量来清除不必要的更新，减少本地更新总数.

(2) 本文介绍了一种基于终端设备网络位置的聚类方法. 移动边缘节点部署在每个簇中作为云服务器和终端设备的通信枢纽，收集和聚合本地更新，以解决与服务器高并发访问相关的延迟.

(3) 本文在两个 FL 模型中验证了所提方法在优化通信上的有效性. 在 MNIST 数据集上的广泛实验表明，与传统 FL 相比，所提方法减少了网络中 60% 的更新数，模型的收敛速度加快了 10.3%.

本文第 2 节对优化 FL 通信效率领域相关工作进行概述；第 3 节介绍相关定义并描述问题；第 4 节介绍基于边缘的联邦学习模型清洗和设备聚类方法；第 5 节对所提方法进行实验分析；最后，总结全文.

2　相关工作

近年来，FL 的通信优化工作得到了研究者的持续关注，提出了一系列的优化方法，主要包括减少通信轮数和终端设备清洗两个方面.

2.1　减少通信轮数

现有的减少通信的工作主要是通过增加边缘的计算和优化算法来实现的. 增加边缘的计算量是指利用边缘的资源计算训练出具有更高收敛性和模型精度的局部模型，从而减少云服务器上的计算量，加快全局模型的收敛速度. 例如，McMahan 等人[10]提出的优化方案把重点放在增加终端设备的数量以及增加终端设备的计算量，以加速全局模型的收敛速度. 此外，Liu 等人[11]提出了一个三层 FL 系统，该系统在边缘服务器和云服务器两层上实现模型聚合，在通信计算中实现了很好的折衷. 在算法优化方面，考虑到 FL 涉及的数据分布在多个终端设备上，Wang 等人[12]提出了一种控制算法，在给定资源预算下，确定局部更新和全局参数聚合之间的最佳权衡，最终使损失函数最小. Wang 在文献[13]中提出了一种名为通信缓和的联邦学习(Communication-Mitigated Federated Learning，CMFL)方法，通过检查本地更新是否符合全局趋势来避免无关更新，最终减少了 FL 的累计通信轮数.

2.2 脏设备清洗

在 FL 的训练过程中，终端设备可能会上传恶意的模型，导致 FL 的学习计划被破坏. 脏设备可能会故意进行不稳定的更新，比如数据中毒攻击，或者无意中产生低质量的数据. 如果这些设备训练的本地模型直接更新到云端参与聚合，势必会影响全局模型的收敛速度，甚至造成大量的不必要通信. 针对此问题，一些工作[14-15]引入了信誉的概念作为度量指标，提出了一种可靠终端设备的选择方案. 此外，Li 等人[16]提出了一种方法，利用预先训练好的异常检测模型来检测终端设备的异常行为，消除其对全局模型产生的不利影响. 具体来说，该方法生成模型权向量的低维代理，并利用它们进行异常检测. Sattler 等人[17]提出了一种基于参数更新之间两两余弦相似度聚类的聚类联邦学习框架，通过将终端设备分成不同的簇来排除异常参与者. 另外，Han 等人[8]专注于本地异常数据清理，以减少异常数据训练出的本地模型对全局模型的毒害.

与上述研究相比，本文的工作更关注服务器高并发性带来的通信延迟和无关本地更新上传到云端时带来的额外通信问题. 本文的解决方案是比较本地更新参数和全局模型参数之间的差异，排除那些差异大于所设阈值的无关更新，并将网络位置相同的设备聚类，通过移动边缘节点以簇的形式与服务器通信，从而提高学习的通信效率.

3 相关概念及问题描述

3.1 联邦学习

本文考虑一个由一个服务器和 N 个终端设备组成的横向 FL 系统[18]，并使用 FedAvg 算法[19]协同训练一个全局模型. 考虑到现有的联邦学习因其参与者众多且分布甚广，因此本文将探讨的联邦学习方案限定在移动通信网络场景. 假设每个终端设备 $i\in N$ 拥有一个独立的本地数据集 D_i，对于任意一个数据样本$\{x_j,y_j\}$，其中 x_j 表示模型的输入，设备的学习任务是找到一个模型参数 w 来描述 y_j，并使得模型的损失函数 $f_j(w)$ 最小. 损失函数是用来评估模型预测结果和真实情况差距的，差距越小，说明模型越好. 例如，对于线性回归模型[20]，损失函数可以表示为 $f_j(w)=\frac{1}{2}(x_j^Tw-y_j)$，$y_j\in R$. 为了方便，本文使用 $|D_i|$ 来表示数据集的大小，并通过 $D=\sum_{i=1}^{N}D_i$ 来定义参与学习的数据总大小. 因此，在联邦学习中，终端设备 i 在其数据集上的损失函数定义为

$$F_i(w)=\frac{1}{|D_i|}\sum_{j\in D_i}f_j(w) \tag{1}$$

根据 FedAvg 算法，服务器上的全局损失函数可以定义为

$$F(w)=\sum_{i=1}^{N}\frac{D_i}{D}F_i(w) \tag{2}$$

FL 的学习目标是将式(2)表示的全局损失函数最小化.

3.2 模型参数

FL 训练开始时，服务器初始化一个全局模型参数，并由终端设备去优化这一参数. 联邦学习算法通常需要 T 次全局迭代来实现损失函数的收敛. 同理，每次全局迭代过程中，终端设备 i 需要在其本地数据集 D_i 上经过多轮本地训练来找到最佳的模型参数

$$w_i^{(t)}=\arg\min F(w) \tag{3}$$

由于大多数机器学习模型固有的复杂性，式(3)通常使用随机梯度下降法(Stochastic Gradient Descent，SGD)[21]来解决. 这些终端设备在其本地数据集上训练的最佳本地更新参数 $w_i^{(t)}$ 需要上传到服务器参与全局聚合. 根据 FedAvg 算法，全局聚合过程可以表示为

$$w^{(t+1)}=\sum_{i=1}^{N}\frac{D_i}{D}w_i^{(t)} \tag{4}$$

全局聚合的目标是使式(2)中的全局损失函数 $F(w)$ 最小化，然后服务器将 $w^{(t+1)}$ 广播到所有终端设备作为下一次迭代的全局模型参数. 经过多次全局迭代后，全局模型收敛，最终获得稳定的全局模型精度 $\varepsilon(\|\nabla F(w^{(t)})\|\leq\varepsilon\leq\|\nabla F(w^{(t-1)})\|$[22].

3.3 问题描述

在每次全局迭代中，每个终端设备在其本地数据集上最小化其损失函数. 由于终端设备的异构性和本地数据的非独立同分布特性，本地模型和全局模型之间的差异很大[13]. 一些本地模型的梯度可能与全球模型的梯度方向相反. 换句话说，在模型聚合阶段，一些训练效果不理想的本地模型可能会毒化全局模型 $w^{(t+1)}$. 如果这些本地模型上传到云端聚合，不仅影响全局模型的精度，还占用了大量网络带宽，造成模型传输的通信延迟. 因此，迫切需要从终端设备中清除异常的本地模型，以减少额外

的通信成本.

本文定义 δ_i 表示在第 t 次全局迭代中从终端设备 i 上传的更新大小，并且由于模型参数的维度一致性，本文假设 δ_i 是恒定的. 对于给定数量的终端设备 N，那么在第 t 次全局迭代中从终端设备上传的数据总数为 $\Upsilon_t=\sum_{i=1}^{N}\delta_i$. 假设 w^* 是最终的目标全局模型参数，本文寻求解决以下问题的方法：

$$\min \sum_{t=1}^{T}\Upsilon_t$$
$$\text{s.t. } w^*=\arg\min F(w) \tag{5}$$

4　基于边缘的模型清洗和聚类方法

在本节中，本文提出了一种提高 FL 通信效率的解决方案，即基于边缘的联邦学习模型清理和设备聚类方法（edge-based model cleaning and device clustering in Federated Learning，简称 eFL）. 对于上一节提出的问题，本文给出了相应的解决方案.

4.1　异常模型检测

与全局模型收敛无关的本地模型通常被称为异常本地模型或脏模型. 本文的目的是检测脏模型，避免上传脏模型，从而减少不必要的通信成本. 在相关的工作中，前文提到 CMFL 通过计算全局更新和本地更新之间符号相同的参数的数量来确定本地更新的重要性，比如，那些满足 $\frac{1}{N}\sum_{j=1}^{N}I(sgn(u_j)=sgn(\bar{u}_j))<Threshold$ 的本地更新被认为不重要，不会被上传，其中 u_j 表示上一轮全局迭代的全局模型参数，$\bar{u}_j$ 表示当前全局迭代中的本地模型参数，N 表示参数总数.

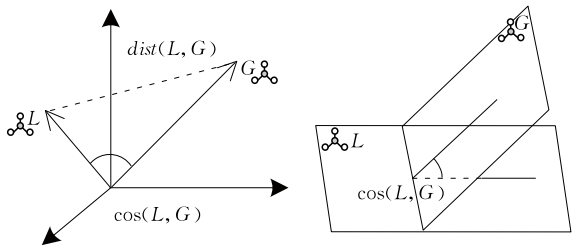
直观地看，虽然模型更新的参数符号决定了模型参数在各维数的改进方向（增加或减少），但参数的值也反映了模型参数在各个方向上的变化程度. 例如，在典型的 Softmax 回归模型中，模型参数的值可以理解为各个类别的 Softmax 概率值，因此本地更新与全局模型对应的参数值应该是相似的. 如果全局模型和本地模型对应参数的符号相同，但参数值相差很大，直观上我们认为这两个模型参数没有关系. 那么一个自然的问题是：是否有其他方法来确定全局更新和本地更新之间的相关性？

本文把注意力转向了被视为联邦学习基础的机器学习和边缘计算. 在机器学习中，判断相似度的方法通常有欧氏距离、余弦相似度算法和 Jaccard 相似度系数等，其中应用最广泛的是欧氏距离和余弦相似度算法. 欧氏距离度量空间中每个点的绝对距离，它与每个点的坐标直接相关；而余弦距离度量空间向量的夹角，它更多的体现在方向上的差异. 考虑到模型参数隐含着模型的收敛方向，本文猜想余弦相似度比欧氏距离更适合用于异常模型检测.

我们在基于边缘计算的数据清洗算法中也发现了一种基于角度的异常检测方法[23]（Angle-Based Anomaly Detection method，ABOD）. ABOD 包含一组数据点集合，对于任意点 o，ABOD 计算满足 $x\neq o$ 且 $y\neq o$ 的一对点 (x,y) 与点 o 的角度 $\angle xoy$，并根据各点的角度方差来确定数据是否异常. 在边缘计算中的发现证实了本文的猜想，即侧重于角度变化的余弦相似度算法更适合作为异常模型检测算法.

为了消除与全局收敛无关的本地更新，本文的第一个任务是确定全局模型的收敛趋势. 由于设备的本地训练是基于前一次迭代中的全局模型参数，因此在进行全局聚合之前，终端设备无法知道当前迭代的全局收敛趋势. 然而，文献[13]验证了连续两次全局更新的差值约为 0.05，最大不超过 0.21，所以前一次迭代的全局更新趋势可以作为判断本地更新重要性的评估标准. 因此，到目前为止，本文已经将异常模型检测问题转化为计算当前迭代中的本地更新参数与前一次迭代的全局模型参数之间的余弦相似度问题.

为了更好地理解的模型参数，本文在几何空间中将模型参数抽象为二维向量和高维空间平面，如图 2 所示，其中 G 表示全局模型，L 表示本地模型. 从图中可以看出，余弦距离度量的是空间矢量的角度，更多的是体现在方向上的差异，而不是位置上. 对于图 2(a)中给定的模型 L 和 G，它们之间的距离 $dist(L,G)$ 是绝对的. 如果 L 的位置保持不变，而 G 的参数在任意维度上发生变化，那么此时余弦距离是变化的（因为角度发生了变化）. 对于图 2(b)中给



(a) 低维空间的模型参数　　(b) 高维空间的模型参数

图 2　模型参数在不同维度空间的抽象表示

定的模型 L 和 G，两平面的角度表示它们的余弦距离，角度越小，余弦距离越小，表示两模型越相似.

对于任意参与训练的终端设备，为方便计算模型参数间的相似度，本文借鉴数字图像处理方法，首先对本地更新和全局模型进行向量化. 假设设备当前迭代中的本地更新 $L_t=[l_1,l_2,\cdots,l_m]$，前一次迭代中的全局更新 $G_{t-1}=[g_1,g_2,\cdots,g_m]$，依据余弦相似度算法，可以计算出全局模型与本地模型之间的相似度，即：

$$similarity_{(L_t,G_{t-1})}=\cos(L_t,G_{t-1})=\frac{\langle L_t,G_{t-1}\rangle}{|L_t||G_{t-1}|}=\frac{\sum_{j=1}^{m}(l_j\times g_j)}{\sqrt{\sum_{j=1}^{m}(l_j)^2}\times\sqrt{\sum_{j=1}^{m}(g_j)^2}} \tag{6}$$

根据三角函数定理，余弦值的取值范围为 $[-1,1]$. 余弦值越接近 1，表示两个向量越接近；余弦值越趋于 -1，表示两者在方向上的差异越大；当余弦值接近 0 时，意味着这两个模型几乎是正交的. 一般来说，相似度值小于 0 往往不符合人们的阅读习惯，因此本文将模型的相似度归一化在 $[0,1]$ 的范围内，即：

$$Similarity_{(L_t,G_{t-1})}=0.5\times\cos(L_t,G_{t-1})+0.5 \tag{7}$$

同理，当 $Similarity_{(L_t,G_{t-1})}$ 趋近于 0 时，说明局部模型参数的有效性更低；反之，有效性较高.

本文设置了阈值 $Threshold$，当本地模型与全局模型的相关性 $Similarity_{(L_t,G_{t-1})}$ 小于所设阈值 $Threshold$ 时，模型参数设置为 NULL，表示模型更新无效，不参与全局聚合. 算法 1 详细说明了终端设备进行本地训练和实施异常模型检测的过程. 阈值的设置将在实验部分进行详细描述.

算法 1. 异常模型检测算法.

输入：终端设备 i 的数据集 D，上一次迭代的全局模型 G 和阈值 f，学习率 λ

输出：本地更新 L

1. 将数据集 D 分割成多个 minibatch，得到 minibatch 集 Bt；
2. FOR b IN Bt DO
3. $\quad L\leftarrow L-\lambda\nabla F(L)$
4. END FOR
5. $Similarity_{(L,G)}=0.5\times\cos(L,G)+0.5$；//计算本地更新参数和全局更新参数之间的余弦相似度
6. IF $Similarity_{(L,G)}<f$ THEN
7. $\quad L=$NULL；
8. END IF

4.2 边缘聚类

FL 的通信时间主要来自于本地更新的上传时间和全局模型的下载时间. 由于网络的下行带宽远大于上行带宽，所以与模型上传时间相比，模型的下载时间可以忽略不计，因此在本工作中不考虑. 前文提到，服务器的网络带宽是有限的，因此到服务器的并发连接数量也是有限的. 如果所有终端设备都直接与服务器通信，这可能会增加能源消耗[24]. 假设数据传输速率与网络中的设备数量成反比. 受 Shannon 定理约束，模型在网络中的传输速率可以假设为

$$v=reverse(\mu)\times B\ln\left(1+\frac{S}{N_p}\right) \tag{8}$$

其中 B 和 $\frac{S}{N_p}$ 分别表示上行链路的带宽和信噪比，μ 表示和服务器连接的设备数量，$reverse(\mu)$. 表示与 μ. 成反比关系的函数. 假设信道的信噪比 $\frac{S}{N_p}$ 在 FL 的学习过程中保持不变，本文把模型更新 w_i 的大小定义为 M，并且假设这一值是恒定的，那么终端设备参与一次全局训练的通信时间可以表示为

$$T_{\text{com}}=\frac{M}{v}=\frac{M}{reverse(\mu)\times B\ln\left(1+\frac{S}{N_p}\right)} \tag{9}$$

由此可发现通信时间 T_{com} 随设备连接数 μ 单调递增，即与服务器连接的设备数越多，联邦学习的通信时间越长. 因此，一个自然的问题是如何有效地利用给定数量的网络资源来最大化模型训练的通信效率？根据上述分析，可以将问题归纳为减少与服务器直连的设备数量.

现有研究指出，与局域网带宽相比，广域网带宽是一种非常稀缺的资源. Hsieh 等人[25]测量了 11 个不同区域（弗吉尼亚、加利福尼亚、俄勒冈等）网络站点之间的平均网络带宽. 结果表明，网络站点之间的广域网（Wide Area Network，WAN）带宽平均比站点内的 LAN 带宽小 15 倍，最坏的情况下高达 60 倍. 它证实了不同区域间广域网带宽的稀缺，并提供了另一条信息：局域网带宽比广域网带宽大得多.

利用局域网带宽比广域网带宽大得多的特点，本文考虑让终端设备尽可能多地在局域网内通信. 但由于局域网中的计算资源是有限的，大规模的全局聚合任务仍然必须在远程云中完成. 因此，本文计划在局域网和广域网中实现 FL 的联合通信. 简而

言之，就是聚类的思想，让终端设备以某种方式形成簇，然后以簇的形式与云服务器通信，既减小模型传输延迟，又达到联邦学习目的.

假设 A_i 为终端设备 i 的 LAN 地址，$cluster_m$ 表示地址为 A_m 的设备聚成的簇，那么设备聚类过程可表示为

$$cluster_m=\{i\,|\,A_i=A_m\} \tag{10}$$

也就是说，根据设备所在的局域网不同，可以将其划分为多个簇，每个簇都是相互独立的. 聚类完成后，由谁完成簇和云端之间的通信成为了进一步要考虑的问题.

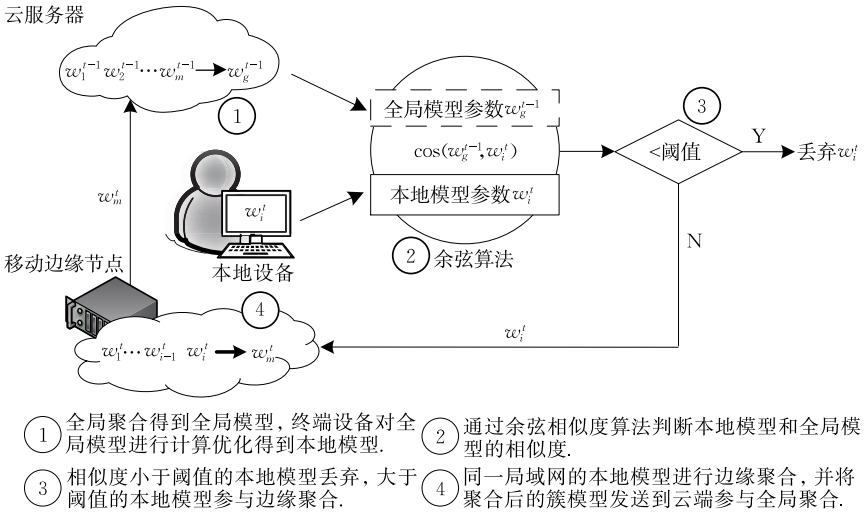
本文的第一个想法是在局域网中选择一个簇头设备，这一灵感来源于传统的无线传感器网络(Wireless Sensor Network，WSN)[26-27]. 在传感器网络中，簇头负责汇合传感器收集到的信息并将这些信息发送给汇聚节点. 类似地，本文考虑使用簇头设备收集 LAN 中的所有本地更新，并将它们发送到云服务器进行全局聚合. 为了验证本文的想法在实践中是否成立，本文做了一个理论分析：假设在 LAN 中有 p 个终端设备，并且每个设备的本地更新大小为 k，那么簇头设备需要传输的总数据大小为 $p\times k$. 换句话说，单个设备和服务器之间传输的数据包的大小已经从 k 变成了 $p\times k$. 簇头的设置减少了与服务器直连的设备数量 μ，提高了数据的传输速率 v，但同时也增大了需要传输的数据量 M. 根据前文对通信时间的定义 $T_{\text{com}}=M/v$ 可以发现，在 M 和 v 同时变化的情况下，很难判断通信时间 T_{com} 的变化趋势. 因此，簇头的思想不但不能减少训练通信时间，还可能增加模型的传输时延.

本文的第二个想法是部署移动边缘节点来收集局域网中的本地更新. 移动边缘计算(Mobile Edge Computing，MEC)是一种提供边缘用户所需的服务和云计算资源的网络架构，可以加速网络中各种应用程序的快速下载. 在之前的研究中[28-29]，作者提出了移动边缘节点，并为边缘节点设计了最大吞吐量和最小传输延迟的移动路由. 因此，本文可以利用移动边缘节点来解决上述问题.

考虑到复杂的 DL 模型训练每次更新可能包含数百万个参数，因此本地模型包含一个较大的梯度向量[30]；且由于聚类机制，边缘和云之间传输的数据大小从 k 变成了 $p\times k$，本地更新的大小更不可忽略不计. 因此，为了保证通信过程中模型变量大小的一致性，本文考虑利用移动边缘节点的计算资源对簇中的本地模型计算加权平均，即本地模型中满足 $Similarity>Threshold$ 的本地更新将被传输到移动边缘节点参与边缘聚合，得到簇模型. 本文将边缘聚合的过程表示为

$$F_m(w)=\sum_{i\in cluster_m}\frac{D_i}{D_m}f_i(w) \tag{11}$$

其中 $D_m=\sum_{i\in cluster_m}D_i$，所有的簇模型都要发送到云端参与全局聚合，以最小化全局损失函数，然后进行下一次的训练迭代. 算法 2 详细说明了边缘聚合的过程. 图 3 描述了 eFL 中模型清理的工作流程.



① 全局聚合得到全局模型，终端设备对全局模型进行计算优化得到本地模型.
② 通过余弦相似度算法判断本地模型和全局模型的相似度.
③ 相似度小于阈值的本地模型丢弃，大于阈值的本地模型参与边缘聚合.
④ 同一局域网的本地模型进行边缘聚合，并将聚合后的簇模型发送到云端参与全局聚合.

图 3　模型清洗的工作流程

算法 2. 模型清洗和设备聚类算法.

输入:终端设备集合 N,设备 i 所在的局域网位置 $Address(i)$

输出:全局模型 G

1. FOR $iteration=1,2,\cdots$ DO
2. FOR 设备 i IN N DO
3. //根据设备局域网地址聚类
4. $Cluster_m=\{i|i \text{ in } Address(m)\}$;
5. END FOR
6. FOR $m=1,2,\cdots$ DO
7. FOR 设备 i IN $Cluster_m$ DO
8. 执行异常模型检测算法得到本地更新 L_i;
9. //收集簇内的本地更新
10. $Clustermodel_m=\{L_i|L_i \text{ is NOT NULL}\}$;
11. END FOR
12. $Edgemodel_m=\sum_{i\in cluster_m}\frac{D_i}{D_m}L_i$; //得到边缘模型
13. END FOR
14. $G=\sum\frac{D_m}{D}Edgemodel_m$; //把边缘模型传送到云服务器参与全局聚合,得到下一次迭代的全局模型 G
15. END FOR

4.3 算法分析

本文的目标是在保证学习收敛的同时尽量减少联邦学习的通信时间.然而,考虑到终端设备的数据集服从非独立同分布,最小化通信时间仍然是一个开放的问题[25].因此,本文只在理论上证明所提算法是保证收敛的,而依靠仿真实验来说明所提方案在通信上的提高.从定性上分析,在本地模型的上传过程中,所提方法首先将本地模型传送到距离较近的移动边缘节点,然后由移动边缘节点将局部聚合后的模型传送到云端,这一过程不但没有增加额外通信开销,反而减小了通信开销.从定量上分析,假设共有 p 个终端设备参与学习,本地更新大小固定且从本地上传到云服务器一次的通信代价为 x,那么传统联邦学习模式下一次全局更新的总通信代价为 $p\times x$;相反,假设本地更新上传到移动边缘节点的通信代价为 y,共设置 $z(z<p)$ 个移动边缘节点,由于移动边缘节点更靠近终端设备,因此 $y\ll x$,那么所提 eFL 一次全局更新的总通信代价为 $z\times x+p\times y$,其远小于传统模式下的通信开销.

此外,假设联邦学习训练达到目标精度时的损失值为 $L[w^*]$,即 $L[w^*]=|F(w)-F(w^*)|$.那么本文可以将联邦学习模型训练的时间复杂度限定为 $\lim_{T\to\infty}\frac{1}{T}L[w^*]=\frac{1}{T}\left[O\left(\frac{1}{\lambda T}\right)+O\left(\sum_{t=1}^{T}f_t\right)\right]$,由于 f 恒定,且经过模型清洗后可以减小训练的的迭代数 T,则本文基本保持了与传统联邦学习在通信上相似的时间复杂度.

5 实验分析

5.1 数据集描述和实验准备

本文评估了两种不同模型在 MNIST 数据集上联邦学习的训练结果.模型包括 Softmax 回归和卷积神经网络(Convolutional Neural Network,CNN).

MNIST. MNIST 是一个手写数字数据集,其中训练集包含 60000 个样本,测试集包含 10000 个样本.MNIST 数据集中的每一幅图像由 28×28 个像素组成,每个像素用一个灰度值表示.

Softmax 回归. Softmax 回归[31]是一种多分类算法.它的本质是将任何一个 k 维实向量映射到另一个 k 维实向量.建模过程首先通过 Softmax( )函数将 MNIST 测试图像转换为概率分布,然后计算真实分布和预测分布之间的交叉熵,最后使用梯度下降不断更新参数,实现损失函数收敛.

MLP. MLP 全称 Multi-Layer Perceptron neural network,即多层感知器神经网络.它由一个输入层、一个输出层和多个隐藏层组成,本文实验中设置了两个隐藏层.

CNN. CNN[32]模型包含共 8 层的网络,即输入层(INPUT)、卷积层(Convolutions,C1)、池化层(Subsampling,S2)、卷积层(C3)、池化层(Subsampling,S4)、卷积层(C5)、全连接层(F6)和输出层(OUTPUT),卷积核的大小是 5×5,激活函数为 ReLU.

本文的实验均基于 TensorFlow[33]框架实现.本文训练了两个模型:MNIST CNN 和 MNIST Softmax 回归.对于 MNIST Softmax 回归模型,本文将 MNIST 训练样本平均分配到 20 个终端设备中,每个终端设备得到 3000 个样本.此外,每次全局迭代每个终端设备在其本地数据集上进行的本地训练轮数 epoch 设置为 93,而用于本地更新的 mini-batch 大小设置为 32.同样,MNIST CNN 模型的终端设备数和本地训练轮数 epoch 分别设置为 10 和 15.

本文试图通过修改数据集中的标签来模拟真实学习场景中的脏数据,从而模拟使用脏数据训练出的与全局收敛方向无关额的不必要模型.现有研究表明,只有当脏标签的比例大于 0.7 时,模型的准确性才会受到影响.为了验证这一观点的正确性,本文随机修改了一定比例的数据集标签,并将这些噪声比不同的数据集用于训练 CNN 模型.图 4 是噪声

标签比例在 0～100%范围内变化时，模型精度的变化结果. 由图可知，当噪声标签比例低于 70%时，模型的精度依然可以达到 0.9，而当噪声标签比例大于 70%时，模型精度急速骤降. 因此，为了反映 eFL 算法在清洗模型方面的有效性，本文随机修改了 MNIST 数据集中超过 70%的样本标签来模拟 FL 环境中的脏数据.

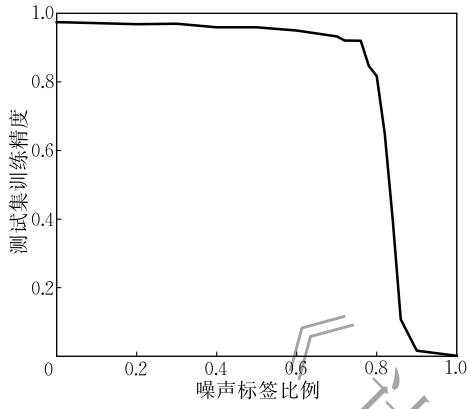


图 4　不同噪声标签比例的数据集训练的模型精度

5.2　阈值设置

为了在不受阈值影响的情况下全面分析 eFL 在模型清洗和提高联邦学习通信效率上的表现，本文测量了在 Softmax 回归模型、CNN 模型以及 MLP 模型上每次迭代过程中全局参数和本地参数之间的相似性，并在图 4 中描述了它们的累计分布(CDF)情况. 如图 5(a)、图 5(b)和图 5(c)所示，和使用未添加噪声标签的数据集进行训练的传统联邦学习方式相比，加入噪声标签后，本地模型和全局模型之间的余弦相似度发生了显著减小的变化，在 Softmax 回归模型中，当使用无脏标签的数据集训练时，全局模型和本地模型之间的相似度接近 1.0，而使用脏标签后二者之间相似度分布在 0.7～1.0 之间，显著低于使用无脏标签数据集训练的结果；在 CNN 模型中，当使用无脏标签的数据集训练时，全局模型和本地模型之间的相似度分布在 0.9998～1.0 之间，而使用脏标签后二者之间相似度分布在 0.9996～1.0 之间，显著低于使用无脏标签数据集训练的结果；在 MLP 模型中，使用无脏标签数据集训练本地模型时，存在 20%的本地模型与全局模型相似度低于 0.9987，而未使用脏标签数据集时，相似度低于 0.9987 的本地模型几乎为 0. 此外，在 Softmax 回归模型中，70%的本地模型与上一次迭代中的全局模型之间的参数相似度小于 0.97；在 CNN 模型中，70%的本地模型与上一次迭代的全局模型之间的参数相似度小于 0.9998. 为了验证 eFL 算法能在脏模型存在的条件下，精准清除脏模型，从而达到无脏模型条件下的同等训练精度这一论点，本文在 Softmax 回归模型中围绕 0.97 测试了一组阈值：{0.95，0.955，0.96，0.965，0.97，0.975，0.98，0.985}，在 CNN 模型中围绕 0.9998 测试了一组阈值：{0.9995，0.9996，0.9997，0.9998，0.9999}，在 MLP 模型中围绕 0.9995 测试了另一组阈值：{0.9990，0.9991，0.9992，0.9993，0.9994，0.9995}. 实验结果表明，当 MNIST CNN 模型、MNISTMLP 模型和 MNIST Softmax 回归模型的阈值分别设为 0.9999、0.9995 和 0.98 时，eFL 能获得最佳性能.

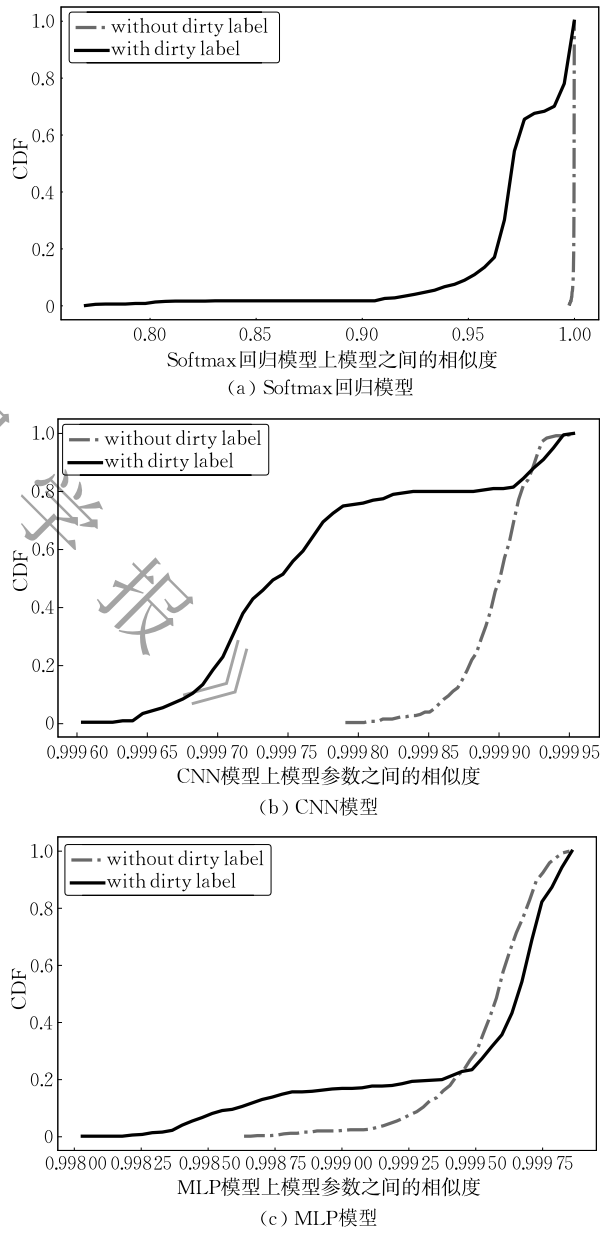


图 5　全局和本地模型参数之间余弦相似度的累积分布图

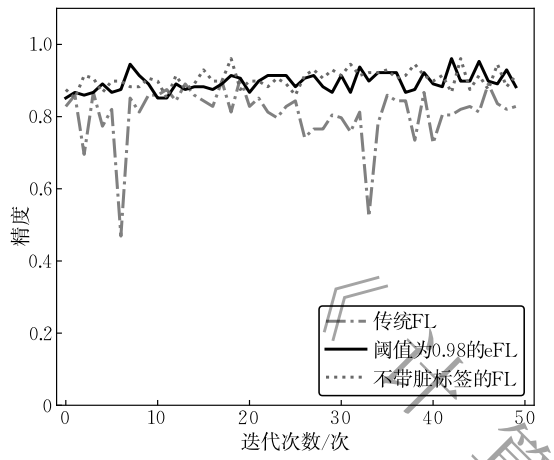
5.3　结果和分析

本小节评估 eFL 在模型清洗和减少网络占用方

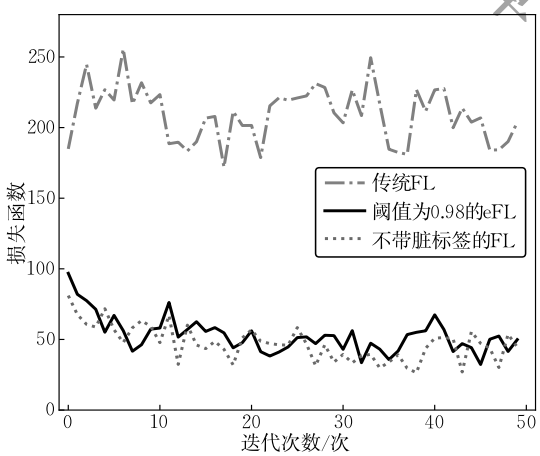
面的性能.在所有实验之前,本文在没有脏标签的数据集上训练了一个完整的模型作为对照.

5.3.1 在 Softmax 回归模型上的实验结果分析

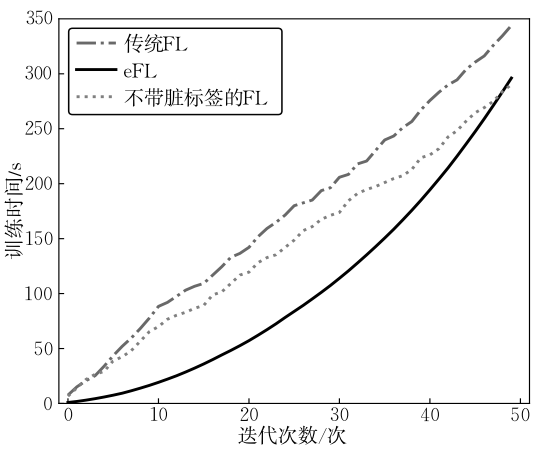
本文将脏标签均匀地分布在 16 个终端设备中,并随机地将一组 LAN 地址匹配到所有设备.模型的学习率设为 $\lambda=0.001$.本文测量了全局迭代过程中模型的准确率和训练损失,并在图 6(a)和图 6(b)中描述了它们的关系.



(a) Softmax回归模型通过不同方式学习的精度



(b) Softmax回归模型通过不同方式学习的损失值



(c) Softmax回归模型通过不同方式学习的训练时间

图 6 Softmax 回归模型通过不同方式学习的表现

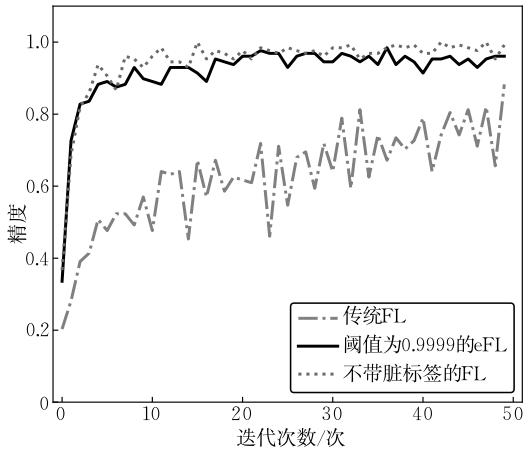
由于终端设备的本地训练轮数 epoch 高达 93,因此在服务器的第一次全局聚合时,全局模型的精度就达到 0.8 以上.总体而言,eFL 在模型清洗方面的效果较显著,模型准确率平稳提高,训练损失显著降低;而未进行模型清洗的传统 FL 的模型精度极不稳定,训练损失无明显下降趋势.在 eFL 方式下训练的模型损失值和在不带脏标签的 FL 方式下训练的模型损失值均显著低于传统 FL 方式下训练的模型.具体来说,在 eFL 方式下训练的模型的精度相比在不带脏标签的 FL 方式下训练的模型仅下降 1%,且二者均显著高于在未进行模型清洗的传统 FL 方式下训练的模型.

本文根据不同设备和服务器之间的距离以及服务器设备的状态,对网络中模型的传输过程随机添加延迟,从而模拟 FL 的训练时间并描绘在图 6(c)中.实验结果表明,在相同的迭代次数下,经过设备聚类和边缘聚合后的 eFL 训练时间最优,传统 FL 和不带脏标签的理想 FL 训练时间相近;与传统 FL 相比,所提 eFL 方法的训练时间减少了 10.3%.

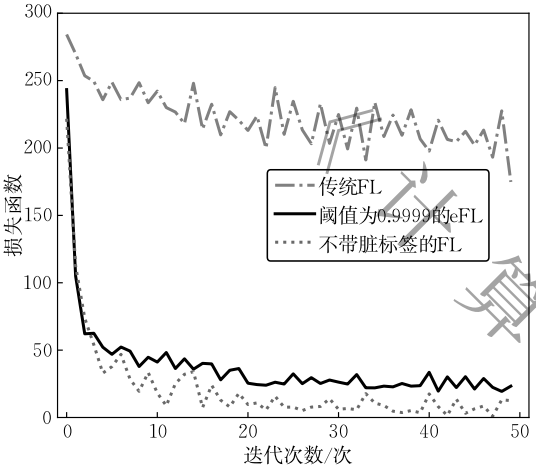
5.3.2 在 CNN 模型上的实验分析

本文在 CNN 模型上进行了类似的实验,其中恶意终端设备的数量为 8,学习率设为 $\lambda=0.001$,$dropout=0.5$.本文同样测量了传统 FL、eFL 和不带脏标签的 FL 三种学习方式在训练 Softmax 回归模型和 CNN 模型的表现,如图 7(a)和图 7(b)所示.在数据噪声比为 0.7 的条件下,eFL 的模型精度远高于未进行模型清洗的传统的 FL.eFL 加速了模型的收敛速度,仅经过 5 次全局迭代,模型的训练损失就减少了 80%.从图的纵向来看,由于脏模型的影响,在经过 50 次全局迭代的同等条件下,未进行模型清洗的传统 FL 精度仅为 0.7,损失值仅降低 33%.从图的水平方向看,当全局模型精度达到 0.6 时,未进行模型清洗的传统 FL 需要经过 13 次迭代,而 eFL 只需 2 次迭代就可以达到相同的精度.

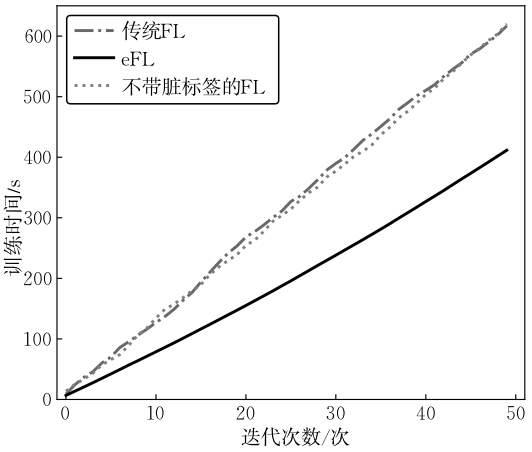
与不带脏标签的 FL 训练方式相比,eFL 在准确性上与之相近,仅下降 2.1%.此外,未经过模型清洗(即保留学习过程中“不符合全局趋势”的部分)的传统 FL 的训练结果远不如模型清洗(即去掉了学习过程中“不符合全局趋势”的部分)后的联邦学习(eFL)的训练结果,由此也可以侧面反映出去掉部分与全局趋势不同的本地模型不会影响整体模型结果的正确性.同样,本文在传统的 FL 模型更新过程中加入延迟来模拟真实场景中的网络状况.如图 7(c)的实验结果显示,在相同的迭代次数下,通



(a) CNN模型通过不同方式学习的精度



(b) CNN模型通过不同方式学习的损失值



(c) CNN模型通过不同方式学习的训练时间

图 7　CNN 模型通过不同方式学习的表现

过设备聚类和边缘聚合的 eFL 学习的模型的训练时间比传统的 FL 缩短了 30.8%.

5.3.3　通信效率评估

本文在表 1 中总结了传统 FL 对比 eFL 和不带脏标签的 FL 在减少通信延迟上的表现，其中三者的通信迭代数均为 50 次. 从整体上看，传统 FL 和不带脏标签的 FL 迭代相同轮数时所需要的通信时间较接近，即脏模型的存在对模型的训练时间影响较大，对传输过程中的通信时间的影响较小；而所提 eFL 方法在迭代相同轮数时所需的通信时间远少于传统 FL，即设备聚类和边缘聚合后，模型上传和下载所需要的时间明显减少，侧面反映出 eFL 可以有效减少联邦学习过程中的网络延迟.

表 1　不同学习方式耗费的通信时间

	传统 FL	eFL	不带脏标签的 FL
Softmax 回归模型	344.0	296.1	290.2
CNN 模型	616.9	411.6	620.0

为了更直观地观察本文提出的 eFL 在优化 FL 通信效率的表现，本文设计了一个称为 *effect* 的度量标准. 对于一个给定的学习精度，*effect* 被定义为在传统 FL 下需要上传的本地更新的总数相比 eFL 模式下需要上传的本地更新总数的归一化值：

$$effect = \frac{\sigma_t - \sigma_e}{\sigma_t} \tag{12}$$

其中，σ_t 表示在不清除模型的情况下需要上传的本地更新的总数，而 σ_e 表示 eFL 模式下需要上传的本地更新的总数. 直观地说，*effect* 越大，eFL 在提高通信效率方面的表现就越好. 本文测量了在不同精度下 eFL 在 MNIST Softmax 回归模型和 MNIST CNN 模型中的 *effect*. 表 2 给出了 *effect* 在不同模型精度上的结果.

表 2　*effect* 在不同模型精度上的结果

	σ_t	σ_e	*effect*
精度 0.6 的 MNIST CNN 模型	110	20	0.73
精度 0.8 的 MNISTCNN 模型	340	30	0.91
精度 0.85 的 Softmax 回归模型	20	8	0.6
精度 0.87 的 Softmax 回归模型	240	10	0.95

对于 Softmax 回归模型，由于模型的收敛速度较快，第一轮全局训练的模型精度就达到 0.8 以上，因此本文在 0.85 和 0.87 两个模型精度上考察 eFL 学习方式对 Softmax 回归模型的影响. 当模型的精度达到 0.85 时，传统 FL 需要从本地上传的模型更新总数为 20，而在 eFL 学习方式下，本地更新只要上传的更新数为 8，由此计算出 *effect* 为 0.6，即相比传统学习方式而言，eFL 减少了 60% 的网络占用. 当精度提高到 0.87 时，未进行模型清理而需要上传的模型数量高达 240（其中包括一些与全局收敛无关的本地模型），而 eFL 将其减少到 10，即经过模型清洗，95% 的本地更新被淘汰，网络的通信成本明显下降.

此外，对于 CNN 模型，由于模型的收敛速度较

平缓,因此本文在 0.6 和 0.8 两个模型精度上衡量 eFL 在提高联邦学习通信效率上的表现. 当模型精度达到 0.6 时,相比传统 FL 学习方式,eFL 学习方式少上传的本地更新数为 90,减少了 73%. 当精度提高到 0.8 时,传统 FL 学习方式下需要上传的更新数为 340,而 eFL 方式下本地仅需上传的更新数为 30. 相比传统 FL,eFL 减少了 91%的模型更新数. 因此,eFL 在提高联邦学习通信效率上有明显成效.

6 结 论

本文提出了一种基于边缘的联邦学习框架 eFL,旨在提高联邦学习的通信效率和增强联邦学习模型的鲁棒性. eFL 引入了基于设备网络位置的聚类思想,并部署移动边缘节点作为云边通信的枢纽,以减少服务器高并发性带来的延迟. eFL 的关键思想是通过计算全局模型参数与局部更新参数之间的余弦相似度来判断局部更新是否符合全局模型的收敛趋势,只有相似度大于所设阈值的局部更新将被移动边缘节点收集和聚合,从而达到模型清洗的目的. 在 Softmax 回归和 CNN 两个学习模型上的实验验证了 eFL 在清理模型和提高通信效率方面的有效性. 与传统的联邦学习相比,eFL 减少了 95%的网络占用,且在 CNN 模型上的收敛速度提高了 30.8%.

参 考 文 献

[1] Zhang Peng-Cheng, Jin Hui-Ying. Privacy protection QoS forecasting in mobile edge environment. Chinese Journal of Computers, 2020, 43(8): 1555-1571(in Chinese)
(张鹏程, 金惠颖. 一种移动边缘环境下面向隐私保护 QoS 预测方法. 计算机学报, 2020, 43(8): 1555-1571)

[2] Liu Y, Huang A, Luo Y, et al. FedVision: An online visual object detection platform powered by federated learning//Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA, 2020: 13172-13179

[3] Yang Z, Chen M, Saad W, et al. Energy efficient federated learning over wireless communication networks. IEEE Transactions on Wireless Communications, 2020, 20(3): 1935-1949

[4] Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, 2019, 10(2): 1-19

[5] Hu H, Wang D, Wu C. Distributed machine learning through heterogeneous edge systems//Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA, 2020: 7179-7186

[6] Li T, Sahu A K, Talwalkar A, et al. Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 2020, 37(3): 50-60

[7] Ren Jie, Gao Ling, Yu Jia-Long, et al. Energy-efficient deep learning task scheduling strategy for edge device. Chinese Journal of Computers, 2020, 43(3): 440-452(in Chinese)
(任杰, 高岭, 于佳龙等. 面向边缘设备的高能效深度学习任务调度策略. 计算机学报, 2020, 43(3): 440-452)

[8] Han Y, Zhang X. Robust federated learning via collaborative machine teaching//Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA, 2020: 4075-4082

[9] Wang T, Cao Z, Wang S, et al. Privacy-enhanced data collection based on deep learning for internet of vehicles. IEEE Transactions on Industrial Informatics, 2019, 16(10): 6663-6672

[10] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data//Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (PMLR). Florida, USA, 2017: 1273-1282

[11] Liu L, Zhang J, Song S H, et al. Edge-assisted hierarchical federated learning with non-IID data. arXiv preprint arXiv: 1905.06641, 2019

[12] Wang S, Tuor T, Salonidis T, et al. Adaptive federated learning in resource constrained edge computing systems. IEEE Journal on Selected Areas in Communications, 2019, 37(6): 1205-1221

[13] Wang L, Wang W, Li B. CMFL: Mitigating communication overhead for federated learning//Proceedings of the IEEE 39th International Conference on Distributed Computing Systems. Texas, USA, 2019: 954-964

[14] Kang J, Xiong Z, Niyato D, et al. Reliable federated learning for mobile networks. IEEE Wireless Communications, 2020, 27(2): 72-80

[15] Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge//Proceedings of the IEEE International Conference on Communications. Shanghai, China, 2019: 1-7

[16] Li S, Cheng Y, Liu Y, et al. Abnormal client behavior detection in federated learning. arXiv preprint arXiv: 1910.09933, 2019

[17] Sattler F, Muller K R, Wiegand T, et al. On the Byzantine robustness of clustered federated learning//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Barcelona, Spain, 2020: 8861-8865

[18] Liu F, Wu X, Ge S, et al. Federated learning for vision-and-language grounding problems//Proceedings of the AAAI

Conference on Artificial Intelligence. New York, USA, 2020: 11572-11579

[19] Lim W Y B, Luong N C, Hoang D T, et al. Federated learning in mobile edge networks: A comprehensive survey. IEEE Communications Surveys Tutorials, 2020, 22(3): 2031-2063

[20] Dong Q, Zhu X, Gong S. Single-label multi-class image classification by deep logistic regression//Proceedings of the AAAI Conference on Artificial Intelligence. Hawaii, USA, 2019: 3486-3493

[21] Wang S, Tuor T, Salonidis T, et al. When edge meets learning: Adaptive control for resource-constrained distributed machine learning//Proceedings of the IEEE Conference on Computer Communications. Hawaii, USA, 2018: 63-71

[22] Tran N H, Bao W, Zomaya A, et al. Federated learning over wireless networks: Optimization model design and analysis//Proceedings of the IEEE Conference on Computer Communications. Paris, France, 2019: 1387-1395

[23] Wang T, Ke H, Zheng X, et al. Big data cleaning based on mobile edge computing in industrial sensor-cloud. IEEE Transactions on Industrial Informatics, 2019, 16(2): 1321-1329

[24] Zhou Q, Guo S, Lu H, et al. Falcon: Addressing stragglers in heterogeneous parameter server via multiple parallelism. IEEE Transactions on Computers, 2020, 70(1): 139-155

[25] Hsieh K, Harlap A, Vijaykumar N, et al. Gaia: Geo-distributed machine learning approaching LAN speeds//Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation. Boston, USA, 2017: 629-647

[26] Wang J, Gao Y, Zhou C, et al. Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs. Computers, Materials & Continua, 2020, 62(2): 695-711

[27] Wang J, Gao Y, Wang K, et al. An affinity propagation-based self-adaptive clustering method for wireless sensor networks. Sensors, 2019, 19(11): 2579

[28] Wang T, Luo H, Jia W, et al. MTES: An intelligent trust evaluation scheme in sensor-cloud-enabled industrial Internet of Things. IEEE Transactions on Industrial Informatics, 2019, 16(3): 2054-2062

[29] Liang Yu-Zhu, Mei Ya-Xin, Yang Yi, et al. A low-coupling method in sensor-cloud systems based on edge computing. Journal of Computer Research and Development, 2020, 57(3): 639-648(in Chinese)
(梁玉珠，梅雅欣，杨毅等. 一种基于边缘计算的传感云低耦合方法. 计算机研究与发展，2020，57(3)：639-648)

[30] He S, Li Z, Tang Y, et al. Parameters compressing in deep learning. Computers, Materials & Continua, 2020, 62(1): 321-336

[31] Shankar V, Singh K. An intelligent scheme for continuous authentication of smartphone using deep auto encoder and Softmax regression model easy for user brain. IEEE Access, 2019, 7: 48645-48654

[32] Zhou S, Tan B. Electrocardiogram soft computing using hybrid deep learning CNN-ELM. Applied Soft Computing, 2020, 86: 105778

[33] Abadi M, Barham P, Chen J, et al. TensorFlow: A system for large-scale machine learning//Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. Savannah, USA, 2016: 265-283

LIU Yan, M. S. candidate. Her main research interest is edge computing.

WANG Tian, Ph. D., professor. His research interests include Internet of Things and edge computing.

PENG Shao-Liang, Ph. D., professor. His research interests include high performance computing, big data, and artificial intelligence.

WANG Guo-Jun, Ph. D., professor. His research interests include network and information security, the Internet of Things, and cloud computing.

JIA Wei-Jia, Ph. D., professor. His research interests include smart city and distributed systems.

Background

As a result of industry competition and data privacy, data is often in the form of an island in most industries. Even in the same company, data integration between different departments is faced with massive resistance, not to mention integrating data from various agencies, which is almost impossible in reality. Besides, with the further development of big data, the emphasis on data privacy and security has become a worldwide trend. As the essential technology of artificial intelligence (AI), federated learning (FL) is a promising approach to resolve this challenge. End devices

participating in federated learning only need to train the model locally and send the trained model to the cloud server for aggregation, thus achieving the purpose of machine learning (ML) under the condition of protecting data privacy. However, the communication efficiency of FL still faces many challenges. On the one hand, the bandwidth of the network is limited, and the uplink is typically much slower than the downlink, so an unstable network may lead to training bottlenecks. On the other hand, the devices and data participating in FL have the problems of heterogeneity and Non-IID, respectively, so the local models trained by these devices and data might not be up to expectations. There may be updated parameters from malicious models. If the malicious models are sent to the cloud for aggregation, it will not only seriously affect the accuracy of model training but also increase the additional communication cost.

We discussed how to effectively leverage the computation and communication resources at the edge to obtain the best FL performance. We consider a typical mobile edge computing architecture in which mobile edge nodes are interconnected with the remote cloud and end devices. End devices are divided into clusters based on their local area network (LAN) address. In the local update phase, each end device calculates the cosine similarity between the local model parameters and the global model parameters. If the similarity between the two is low, the local model is considered an unnecessary update, thereby avoid additional communication costs. The mobile edge nodes deployed in each LAN collect and aggregate non-malicious updates and then sends the aggregated model to the cloud server for global aggregation, thereby avoiding the latency associated with high server concurrency.

This work was supported in part by grants from the National Natural Science Foundation of China under Grant No. 62172046, the Natural Science Foundation of Fujian Province of China (No. 2020J06023), the UIC Start-up Research Fund under Grant No. R72021202, Provincial Department of Education in Key Fields of Colleges and Universities (2021ZDZX1063), and the Jiont Project of Production, Teaching and Research of Zhuhai (ZH22017001210133PWC).