计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(10):199−205 [doi: 10.15888/j.cnki.csa.008748]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

提升联邦学习通信效率的梯度压缩算法①

田金箫

(西南交通大学 计算机与人工智能学院, 成都 611756)
通信作者: 田金箫, E-mail: tianjinx@foxmail.com

摘　要: 联邦学习通过聚合客户端训练的模型, 保证数据留在客户端本地, 从而保护用户隐私. 由于参与训练的设备数目庞大, 存在数据非独立同分布和通信带宽受限的情况. 因此, 降低通信成本是联邦学习的重要研究方向. 梯度压缩是提升联邦学习通信效率的有效方法, 然而目前常用的梯度压缩方法大多针对独立同分布的数据, 未考虑联邦学习的特性. 针对数据非独立同分布的联邦场景, 本文提出了基于投影的稀疏三元压缩算法, 通过在客户端和服务端进行梯度压缩, 降低通信成本, 并在服务端采用梯度投影的聚合策略以缓解客户端数据非独立同分布导致的不利影响. 实验结果表明, 本文提出的算法不仅提升了通信效率, 而且在收敛速度和准确率上均优于现有的梯度压缩算法.

关键词: 联邦学习; 通信效率; 非独立同分布数据; 梯度压缩

引用格式: 田金箫.提升联邦学习通信效率的梯度压缩算法.计算机系统应用,2022,31(10):199–205. http://www.c-s-a.org.cn/1003-3254/8748.html

Gradient Compression Algorithm for Improving Communication Efficiency of Federated Learning

TIAN Jin-Xiao

(School of Computer and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China)

Abstract: Federated learning protects user privacy by aggregating trained models of the client and thereby keeping the data local on the client. Due to the large numbers of devices participating in training, the data is non-independent and identically distributed (non-IID), and the communication bandwidth is limited. Therefore, reducing communication costs is an important research direction for federated learning. Gradient compression is an effective method of improving the communication efficiency of federated learning. However, most of the commonly used gradient compression methods are for independent and identically distributed data without considering the characteristics of federated learning. For the scene of non-IID data in federated learning, this study proposes a sparse ternary compression algorithm based on projection. The communication cost is reduced by gradient compression on the client and server, and the negative impact of non-IID client data is mitigated by gradient projection aggregation on the server. The experimental results show that the proposed algorithm not only improves communication efficiency but also outperforms the existing gradient compression algorithms in convergence speed and accuracy.

Key words: federated learning; communication efficiency; non-independent and identically distributed (non-IID) data; gradient compression

1　引言

近年来, 随着人工智能技术的快速发展和广泛应用, 数据隐私保护也得到了密切关注. 欧盟出台了首个关于数据隐私保护的法案《通用数据保护条例》(General Data Protection Regulation, GDPR)[1], 明确了对数据隐私保护的若干规定. 中国自 2017 年起实施的

① 收稿时间: 2021-12-29; 修改时间: 2022-01-29, 2022-02-22; 采用时间: 2022-03-03; csa 在线出版时间: 2022-06-28

《中华人民共和国网络安全法》和《中华人民共和国民法总则》中也对用户隐私数据的使用做出了明确的规定. 在机器学习中, 模型的好坏很大程度上依托于建模的数据. 但由于相关法律法规的限制, 数据孤岛问题变得十分普遍, 导致企业很难获取训练数据. 为此, 谷歌在 2016 年提出了联邦学习的概念. 联邦学习是一种基于分布式机器学习的框架, 在这种框架中, 多个客户端在中央服务器的协调下共同训练模型, 并保证训练数据可以保留在本地, 不需要像传统的机器学习方法一样将数据上传至中央服务器[2], 从而保护了用户隐私.

构建一个高性能的联邦模型通常需要多轮通信, 同时规模庞大的神经网络模型, 往往包含数百万个参数[3], 这导致了巨大的通信开销. 此外, 相较于传统的分布式机器学习, 联邦学习还面临如下问题:

1) 客户端数据非独立同分布: 在传统分布式机器学习中的训练数据随机均匀地分布在客户端上[4], 即遵循独立同分布 (independent and identically distributed, IID). 这在联邦学习中通常是不成立的, 由于用户的喜好不同, 客户端的数据通常是非独立同分布 (non-IID) 的. 即客户端拥有的局部数据集不能代表整体数据的分布, 不同客户端之间的数据分布也不同.

2) 数据不平衡: 不同的客户端可能拥有不同的数据量.

3) 客户端数量庞大且不可靠: 参与训练的客户端为大量的移动设备, 通常大部分客户端经常离线或者处于不可靠的连接上, 因此无法确保客户端参与每一轮的训练.

本文主要研究联邦学习中的通信效率问题, 利用梯度稀疏化的思想减少客户端与服务器之间通信的参数量, 并在服务器聚合时使用投影的方式缓解非独立同分布数据带来的影响. 经过在 MNIST 和 CIFAR10 数据集上的实验证明, 本文提出的算法能够在联邦学习的约束条件下高效训练模型.

2 相关工作

一般来说, 减少联邦学习中的通信开销有两种策略, 一种是减少训练过程中的通信轮次, 另一种是减少每轮传递的通信量. 减少通信轮次的经典方案是联邦学习中最常用的 FedAvg 算法[2], 即令客户端在本地执行多轮本地更新, 服务器再进行全局聚合, 来减少通信轮数. FedAvg 在每次通信中, 客户端需要上传或下载整个模型, 由于联邦客户端通常运行在缓慢且不可靠的网络连接上, 这一要求使得使用 FedAvg 训练大型模型变得困难. 在实际应用中, FedAvg 算法可以较好地处理非凸问题, 但该算法不能很好处理联邦学习中数据 non-IID 的情况, 在此应用场景很可能导致模型不收敛[5]. 因此针对 non-IID 场景, Briggs 等[6] 在 FedAvg 的基础上引入层次聚类技术, 根据局部更新与全局模型的相似度对客户端进行聚类和分离, 以减少总通信轮数. 此外 Karimireddy 等[7] 通过估计服务器与客户端更新方向的差异来修正客户端本地更新的方向, 有效地克服了 non-IID 问题, 能在较少的通信轮次达到收敛.

另一类方法的核心思想在于减少传输的数据量, 主要通过量化、稀疏化等一系列方法对模型参数或者梯度进行压缩. 量化通过将元素低精度表示或者映射到预定义的一组码字来减少梯度张量中每个元素的位数, 例如 Dettmers[8] 将梯度的 32 位浮点数量化至 8 位, SignSGD[9–11] 则只保留梯度的符号来更新模型, 将负梯度量化为–1, 其余量化为 1, 实现了 32 倍的压缩. 稀疏化方法通过只上传部分重要的梯度来进行全局模型的更新, 如何选择这些梯度成为该方法的关键. Strom[12] 提出使用梯度的大小来衡量其重要性, 通过预先设立阈值, 当梯度大于该阈值时对其进行上传. 然而在实际情况中, 由于不同的网络结构参数分布差异较大, 导致我们无法选择合适的阈值. 因此目前稀疏化方法通常使用 Aji 等[13] 提出的固定稀疏率, 每次传递一定比例的最大梯度或每次传递前 k 个最大梯度的 Topk 方法[14]. 上述工作有效地解决了分布式机器学习中的通信开销问题, 针对联邦学习的训练环境, Rothchild 等[15] 使用了一种特殊的数据结构计数草图 (count sketch) 对客户端梯度进行压缩. Chen 等[16] 将神经网络的不同层分为浅层和深层, 并认为深层参数更新频率低于浅层参数, 因此提出了异步更新策略, 有效减少了每轮传递的参数量. Haddadpour 等[17] 在 FedAvg 的基础上对每轮传递的参数进行压缩, 并针对 non-IID 场景采用梯度跟踪技术对客户端梯度方向进行修正, 在收敛速度和准确率上都取得了较好的效果.

Sattler 等[18] 也针对联邦学习的训练环境提出了稀疏三元压缩 (sparse ternary compression, STC), 该方法在 Topk 梯度稀疏化的基础上进行了量化进一步减少了通信量, 并利用错误反馈机制实现了客户端与服务器之间的双向压缩, 在联邦学习场景中表现出了良好

的效果. 该方法考虑了联邦学习中客户端 non-IID 数据的场景, 通过利用稀疏的特性以及减少本地训练次数与服务器端频繁通信去减轻 non-IID 数据带来的问题, 但该方法对 non-IID 数据的优化能力有限. 因此本文将在稀疏三元压缩算法的基础上, 关注 non-IID 下的联邦场景, 提升联邦学习的通信效率.

3 算法设计

3.1 稀疏三元压缩

常规的 Topk 稀疏方法以全精度传递稀疏元素, Sattler 等[19] 证明了当稀疏化与非零元素的量化相结合时, 可以获得更高的压缩增益. 如算法 1 所示, 当获得 Topk 稀疏元素T^{masked}后, 会将其量化为稀疏元素的平均值, 因此最后只需要传递一个包含值$\{-\mu,0,\mu\}$的三元张量. 如果将每一层的梯度看做一个矩阵, 那么使用 Topk 和稀疏三元压缩后得到的结果如图 1 所示, 原始梯度是一个稠密矩阵, 颜色深浅代表值的大小, 通过 Topk 方法会得到一个保留较大值的稀疏矩阵, 值较小的则置为 0, 而稀疏三元压缩则在 Topk 的基础上做了量化, 进一步提升了压缩率.

算法 1. STC[18]: 稀疏三元压缩算法

输入: 张量$T\in\mathbb{R}^n$, 稀疏率 p

1. $k\leftarrow\max(np,1)$
2. $v\leftarrow\text{top}_k(|T|)$
3. $mask\leftarrow(|T|\geqslant v)\in\{0,1\}^n$
4. $T^{\text{masked}}\leftarrow mask\odot T$
5. $\mu\leftarrow\frac{1}{k}\sum_{i=1}^{n}|T_i^{\text{masked}}|$
6. 输出$T^*\leftarrow\mu\times\text{sign}\left(T^{\text{masked}}\right)$

Sattler 等[18] 在联邦学习中使用稀疏三元压缩对客户端和服务器之间通信的梯度进行双向压缩, 并结合错误反馈机制[20] 在客户端和服务器保留压缩前后的误差累加至下一轮训练过程.

$$\hat{g}_i^t = STC\left\{g_i^t + error^{t-1}\right\} \tag{1}$$

$$error^t = g_i^t - \hat{g}_i^t \tag{2}$$

其中, g_i^t为第i个客户端第t轮训练得到的原始梯度, $\hat{g}_i^t$为压缩后的梯度, $error^t$为压缩前后的误差. 该方法取得了与非压缩算法相似的收敛速度并大大减少了每一轮的通信量, 因此本文也将使用稀疏三元压缩方法进行梯度压缩.

3.2 Non-IID 数据的处理

目前在联邦学习中, 我们通常采用平均各个客户端梯度的方法计算全局模型. 当不同客户端数据满足 IID 条件时, 各客户端梯度更新方向相近, 且聚合后梯度与基于传统的集中式学习获得的梯度相似性较高. 故此方法能获得全局目标函数的最优解. 若客户端数据 non-IID 且数据量差异较大, 各客户端梯度差异性较大, 存在相互干扰的情况, 导致全局模型收敛速率降低. 同时, 简单平均各方梯度易使数据量多的客户端占主导作用, 使得全局模型无法较好地处理数据量较少的客户端, 最终导致全局模型整体性能低下.

(a) 原始梯度 (b) Topk (c) STC

图 1 梯度压缩效果

Wang 等[21] 提出使用梯度投影处理 non-IID 数据的问题, 服务器端在进行梯度平均之前, 通过修改梯度方向减轻 non-IID 数据带来的影响. 该方法首先对客户端之间的梯度冲突做出定义, 当客户端i的梯度g_i和客户端j的梯度g_j满足$g_i\cdot g_j<0$时, 则称为客户端i和客户端j之间存在梯度冲突. 当客户端之间存在梯度冲突时, 梯度方向差异性较大, 这时可以通过将一个客户端的梯度投影到另一个有冲突的客户端梯度平面上, 使用原梯度减去投影来缩小客户端之间的梯度差异, 如式 (3) 所示:

$$g_i = g_i - \frac{\left(g_i\cdot g_j\right)g_j}{\|g_j\|^2} \tag{3}$$

此外, 该方法定义了内部冲突和外部冲突, 分别对其进行投影处理. 将参与训练的客户端之间的梯度冲突定义为内部冲突, 将客户端梯度按照训练损失从小到大排序得到$PO_t=\left\{g_1^t,g_2^t,\cdots,g_m^t\right\}$, 并引入参数$\alpha$来控制每轮参与投影的客户端数目. 从$PO_t$中选择损失较小的客户端$S_t^{\alpha}$迭代的判断与其他客户端之间的梯度冲突, 并进行投影修改梯度方向以缓解内部冲突. 对于未选择的损失较大的客户端则保持原有的梯度, 此后进行梯度平均得到聚合后的梯度g^t, 如算法 2 所示.

在实际联邦场景中, 客户端 non-IID 程度较大, 在每轮聚合中, 若对所有客户端统一采用投影方案, 则导致训练损失大的客户端的梯度方向不断靠近损失小的

客户端. 这将导致聚合模型无法学习到所有客户端的信息. 但通过调整参数α, 自适应地让部分训练损失较大的客户端直接参与最终的聚合阶段, 有效地缓解了上述问题.

算法 2. MitigateInternalConflict[21]: 缓解内部冲突算法

输入: 客户端梯度投影顺序PO_t, 参数α

服务器从PO_t选择损失较小的客户集合$S_t^{1-\alpha}$参与投影, 保留α比例损失较大的客户端梯度
for each client $k\in S_t^{1-\alpha}$ in parallel **do**
  $g_k^{PC}\leftarrow g_k^t$
  for each $g_i^t\in PO_t$, $i=1,\cdots,m$ **do**
    if $g_k^{PC}\cdot g_i^t<0$ and $k\neq i$ **then**
      投影修正客户端梯度: $g_k^{PC}\leftarrow g_k^{PC}-\frac{(g_i^t)\cdot g_k^{PC}}{\|g_i^t\|^2}g_i^t$
    end if
  end for
end for
计算聚合梯度: $g^t\leftarrow\frac{1}{m}\sum_{k=1}^{m}g_k^{PC}$
返回聚合梯度g^t

由于联邦学习中客户端的部分参与和不可靠连接, 在第t轮未被选中参与训练的客户端可能会遭受被全局模型遗忘的风险, 因此可以在服务端保留其最近一次参与训练的梯度$GH=\left\{g_1^{t_1},g_2^{t_2},\cdots,g_K^{t_K}\right\}$, 根据它们的近邻历史梯度来估计真实梯度以避免客户端被遗忘, 如算法 3 第 6 步所示. 第t轮未被选中客户端的估计梯度g_{con}与参与更新的客户端平均后的梯度g^t之间的冲突称为外部冲突, 通过将g^t迭代的投影到不同轮次的估计梯度g_{con}的法平面以缓解外部冲突, 通过参数τ控制投影的轮次. 具体步骤如算法 3 所示.

算法 3. MitigateExternalConflict[21]: 缓解外部冲突算法

输入:聚合梯度g^t, 所有客户端近邻历史梯度GH, 参数τ

for round $t-i$, $i=\tau,\tau-1,\cdots,1$ **do**
  初始化估计梯度:$g_{\text{con}}\leftarrow 0$
  for each client $k=1,2,\cdots,K$ **do**
    if $t_k=t-i$ **then**
      if $g^t\cdot g_k^{t_k}<0$ **then**
        计算未被选中客户端的估计梯度: $g_{\text{con}}\leftarrow g_{\text{con}}+g_k^{t_k}$
      end if
    end if
  end for
  if $g^t\cdot g_{\text{con}}<0$ **then**
    对聚合梯度投影修正: $g^t\leftarrow g^t-\frac{g^t\cdot g_{\text{con}}}{\|g_{\text{con}}\|^2}g_{\text{con}}$
  end if
end for
返回聚合梯度g^t

3.3 基于投影聚合的稀疏三元压缩算法

鉴于投影能够有效地处理联邦学习中的 non-IID 数据问题, 因此本文将在稀疏三元压缩的基础上, 在服务器端使用投影聚合的方式, 进一步提高模型的正确率与收敛速度, 具体步骤如算法 4 所示.

在算法 4 中使用网络模型更新量表示客户端梯度$g_i^t\leftarrow w_i^{t-1}-w_i^t$, 因此在算法的第 5–6 行客户端接收到聚合梯度$\bar{g}$后首先计算本轮的初始网络训练模型w_i^t. 在第 7 行根据随机梯度下降方法训练新的模型, 并计算本轮的客户端梯度g_i^t. 此后在第 8–9 行使用结合错误反馈机制的 STC 算法压缩梯度, 在客户端本地保留压缩前后的梯度差异在下一轮压缩时重新引入, 减小由于压缩导致的梯度信息丢失. 之后客户端将压缩后的梯度与训练损失发送给服务器.

服务器端接收到客户端梯度与训练损失后, 首先在算法第 14 行更新每个客户端最近一次参与训练的梯度$GH=\left\{\hat{g}_1^{t_1},\hat{g}_2^{t_2},\cdots,\hat{g}_K^{t_K}\right\}$以便在缓解外部冲突时使用, 其中$K$是所有客户端个数, t_K是客户端最近一次参与训练的轮次. 之后在第 15 行根据训练损失的大小对本轮参与训练的客户端梯度进行排序得到$PO_t=\left\{\hat{g}_1^t,\hat{g}_2^t,\cdots,\hat{g}_m^t\right\}$, 其中$m$是本轮参与训练的客户端个数. 然后依次根据算法 2 中的缓解内部冲突算法和算法 3 中的缓解外部冲突算法得到聚合梯度g^t. 算法 2 和算法 3 的主要作用是对聚合梯度g^t的方向进行修正以缓解 non-IID 问题, 因此在第 20 行中, 保留修正后的聚合梯度g^t的方向与原始聚合梯度的大小得到最终的聚合梯度. 最后使用与客户端相同的 STC 压缩算法压缩聚合梯度并发送至客户端.

算法 4. 基于投影聚合的稀疏三元压缩算法

输入: 初始化模型w

1. **for** $t=1,\cdots,T$ **do**
2. 服务器从 K 个客户端随机选取 m 个客户端参与训练
3. **for** $i=1,\cdots,m$ in parallel **do**
4. 客户端C_i:
5. 从服务器端下载聚合梯度$\bar{g}$
6. $w_i^t\leftarrow w_i^{t-1}-\bar{g}$
7. $g_i^t\leftarrow SGD\left(w_i^t,Data_i\right)-w_i^t$
8. $\hat{g}_i^t\leftarrow STC\left(g_i^t+error^{t-1},p\right)$
9. $error^t=g_i^t-\hat{g}_i^t$
10. 上传客户端梯度$\hat{g}_i^t$和训练损失l_i^t至服务器
11. **end for**
12. 服务器器端:
13. 接收参与训练的客户端梯度$\hat{g}_i^t$和训练损失l_i^t

14. 更新所有客户端近邻历史梯度信息: $GH=\{\hat{g}_1^{t1},\hat{g}_2^{t2},\cdots,\hat{g}_K^{tK}\}$
15. 根据客户端训练损失对梯度排序: $PO_t=\{\hat{g}_1^t,\hat{g}_2^t,\cdots,\hat{g}_m^t\}$
16. 缓解内部冲突: $g^t\leftarrow MitigateInternalConflict(PO_t,\alpha)$
17. **if** $t\geqslant\tau$ **then**
18. 缓解外部冲突: $g^t\leftarrow MitigateExternalConflict(g^t,GH,\tau)$
19. **end if**
20. $g^t=g^t/\|g^t\|*\|\frac{1}{m}\sum_i^m\hat{g}_i^t\|$
21. $\bar{g}=STC(g^t+error,p)$
22. $error=g^t-\bar{g}$
23. 发送聚合梯度$\bar{g}$至客户端
24. **end for**

算法 4 中的步骤可简化为图 2, 在客户端, 首先接收聚合梯度$\bar{g}$, 然后根据模型和客户端数据进行本地训练得到客户端梯度g_i^t, 本地训练完成后使用 STC 算法压缩梯度上传至服务器, 并计算压缩误差存储在本地, 在下一轮被选中训练时进行梯度修正.

服务端接收到所有参与训练的客户端发送的梯度后判断客户端梯度之间是否存在梯度冲突, 并依次通过缓解内部冲突和外部冲突的算法对梯度方向进行修正. 最终聚合投影后的梯度生成全局梯度g^t, 采用 STC 算法压缩全局梯度g^t得到$\bar{g}$发送至客户端. 该算法实现了客户端与服务器之间的双向压缩, 并且在服务器端进行投影缓解数据异构的问题.

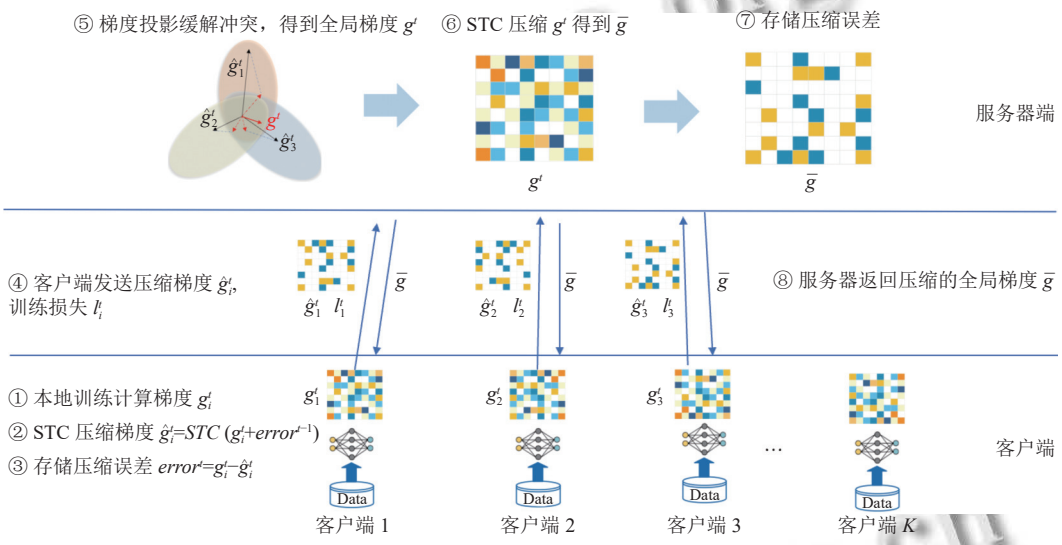

图 2 基于投影聚合的稀疏三元压缩算法流程

4 实验分析

4.1 实验设置

本文的实验使用了 MNIST 和 CIFAR10 数据集. MNIST 数据集包含 60 000 张训练图片, 10 000 张测试图片, 每张图片是 2 828 的灰度手写数字图像, 实验使用带有 3 个卷积层的 CNN 模型对 MNIST 进行训练. CIFAR10 数据集包含 50 000 张训练图片, 10 000 张测试图片, 每张图片是 3 232 的 RGB 图像, 使用文献 [18] 中简化的 VGG11 网络进行训练. 客户端数据集划分参照文献 [2], 首先按照数据集的类别进行排序, 然后将数据集划分为 200 个分片, 每个客户端随机选择两个不会替换的分片来模拟客户端数据非独立同分布的场景. 实验中部分参数设置如表 1 所示.

4.2 实验结果

我们将本文提出的算法与 FedAvg 以及稀疏三元压缩算法进行了对比, 图 3 和图 4 是在 MNIST 数据集上的结果, 图 3 是全局模型在所有客户端上的平均测试准确率, 图 4 为测试准确率的方差, 其中稀疏三元压缩以及本文提出的算法在实验中设置了 0.1 的稀疏率, 也就是每轮传递 10% 的参数进行训练, 根据图 1 的实验结果可以看到本文提出的算法相较于其他算法收敛速度和收敛精度都略有提升, 特别是相较于 STC 算法, 在相同压缩率的条件下本文提出的算法大约在第 75 轮收敛, 而 STC 算法在训练过程非常震荡, 并且在大约 100 轮才收敛.

表 1 参数设置

数据集	网络	客户端数量	客户端参与率	本地训练轮数	训练批次大小
MNIST	CNN	200	0.1	1	64
CIFAR10	VGG11	100	0.2	1	64

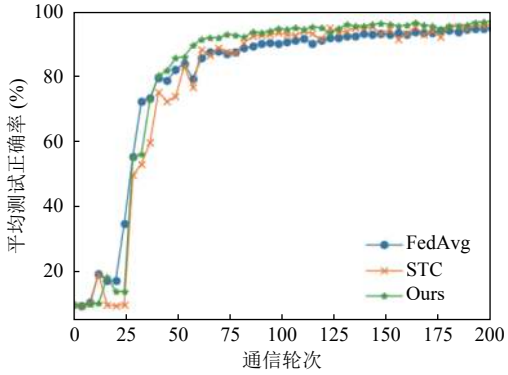

图 3 MNISTS 数据集测试正确率

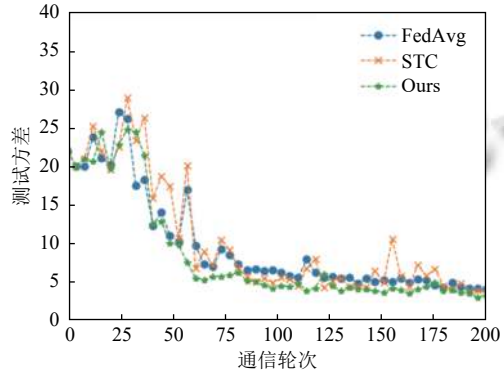

图 4 MNISTS 数据集测试方差

图 5 和图 6 是在 CIFAR10 数据集上的测试准确率和测试方差, 稀疏率同样为 0.1, 与 MNIST 数据集相比, 在 CIFAR10 数据集上的训练过程更加震荡, 但是本文提出的算法相较其他算法收敛速度和收敛精度都有大幅度提升, 并且训练过程中的震荡幅度远远小于 FedAvg 和 STC 算法, 这说明本文的算法是非常有效的.

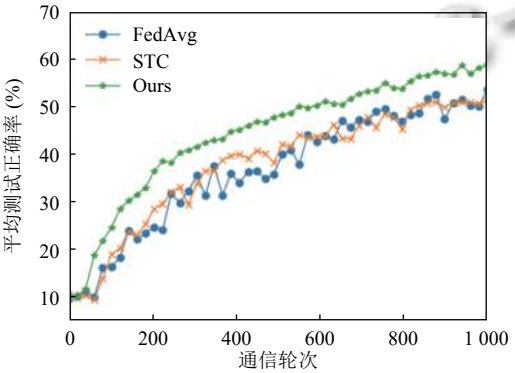

图 5 CIFAR10 数据集平均测试正确率

表 2 中记录了客户端与服务器之间每轮通信的参数大小, 通信轮次是达到固定正确率 (MNIST 95% CIFAR10 50%) 大约所用的通信轮数, 以 FedAvg 作为基线算法, 本文提出的算法在上传和下载时都进行了压缩, 在 MNIST 数据集上相较于 FedAvg 每轮的通信量减少了 45 倍, 并且本文的算法在第 100 轮时就达到了指定的正确率, 相较于 FedAvg 和 STC 分别减少了 97 和 57 个通信轮次, 在 CIFAR10 数据集上每轮的通信量更是减少了 47 倍, 通信轮次相较于 FedAvg 和 STC 减少了 295 轮和 300 轮.

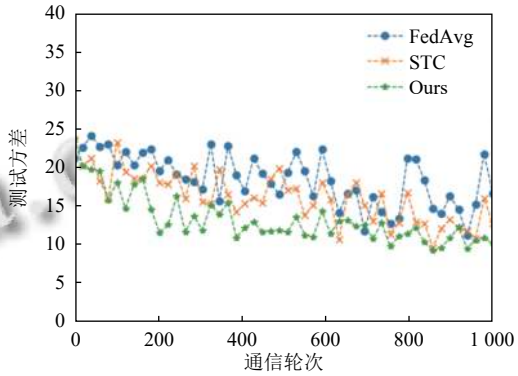

图 6 CIFAR10 数据集测试方差

表 2 通信开销计算

模型	算法	上传 (MB)	下载 (MB)	压缩倍数	通信轮次
CNN @MINST	FedAvg	1.36	1.36	—	197
	STC	0.03	0.03	45	157
	本文	0.03	0.03	45	100
VGG11 @CIFAR10	FedAvg	3.3	3.3	—	785
	STC	0.07	0.07	47	790
	本文	0.07	0.07	47	490

5 结论

本文提出了基于投影聚合的稀疏三元压缩算法, 提升联邦学习的通信效率. 该算法在客户端和服务端采用稀疏三元压缩减少客户端在每一轮训练过程中上传和下载的通信量, 同时在服务器端利用梯度投影的方式缓解了由于客户端数据异构以及部分参与导致的梯度冲突问题. 通过在 MNIST 和 CIFAR10 数据集上的实验验证, 本文提出的算法在通信量、收敛速度和正确率 3 个方面都要由于传统的 FedAvg 算法和稀疏三元压缩算法. 由于梯度压缩会略微改变原始梯度的方向, 在未来我们将针对不同的压缩方法对投影聚合的方式做进一步的研究, 进一步提高算法的有效性.

参考文献

1 General Data Protection Regulation. Complete guide to GDPR compliance. https://gdpr.eu/. [2021-12-26].

2 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. Fort Lauderdale: PMLR, 2017. 1273–1282.
3 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778.
4 Boyd S, Parikh N, Chu E. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning, 2011, 3(1): 1–122.
5 Li T, Sahu AK, Zaheer M, *et al*. Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems 2020. Austin: MLSys, 2020. 429–450.
6 Briggs C, Fan Z, Andras P. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. 2020 International Joint Conference on Neural Networks (IJCNN). Glasgow: IEEE, 2020. 1–9. [doi: 10.1109/IJCNN48605.2020.9207469]
7 Karimireddy SP, Kale S, Mohri M, *et al*. Scaffold: Stochastic controlled averaging for federated learning. Proceedings of the 37th International Conference on Machine Learning. Online: PMLR, 2020. 5132–5143.
8 Dettmers T. 8-bit approximations for parallelism in deep learning. Proceedings of the 4th International Conference on Learning Representation. San Juan: ICLR, 2016. 1–14.
9 Bernstein J, Wang YX, Azizzadenesheli K, *et al*. signSGD: Compressed optimisation for non-convex problems. Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018. 559–568.
10 Karimireddy SP, Rebjock Q, Stich SU, *et al*. Error feedback fixes signsgd and other gradient compression schemes. Proceedings of the 36th International Conference on Machine Learning. Long Beach: PMLR, 2019. 3252–3261.
11 Zheng S, Huang ZY, Kwok JT. Communication-efficient distributed blockwise momentum SGD with error-feedback. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2019. 1027.
12 Strom N. Scalable distributed DNN training using commodity GPU cloud computing. 16th Annual Conference of the International Speech Communication Association. Dresden: ISCA, 2015. 1488–1492.
13 Aji AF, Heafield K. Sparse communication for distributed gradient descent. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017. 440–445.
14 Stich SU, Cordonnier JB, Jaggi M. Sparsified SGD with memory. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montreal: NeurIPS, 2018. 4452–4463.
15 Rothchild D, Panda A, Ullah E, *et al*. Fetchsgd: Communication-efficient federated learning with sketching. Proceedings of the 37th International Conference on Machine Learning. Online: PMLR, 2020. 8253–8265.
16 Chen Y, Sun XY, Jin YC. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(10): 4229–4238. [doi: 10.1109/TNNLS.2019.2953131]
17 Haddadpour F, Kamani MM, Mokhtari A, *et al*. Federated learning with compression: Unified analysis and sharp guarantees. Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. San Diego: PMLR, 2021. 2350–2358.
18 Sattler F, Wiedemann S, Müller KR, *et al*. Robust and communication-efficient federated learning from non-iid data. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(9): 3400–3413. [doi: 10.1109/TNNLS.2019.2944481]
19 Sattler F, Wiedemann S, Müller KR, *et al*. Sparse binary compression: Towards distributed deep learning with minimal communication. 2019 International Joint Conference on Neural Networks (IJCNN). Budapest: IEEE, 2019. 1–8. [doi: 10.1109/IJCNN.2019.8852172]
20 Stich SU. Local SGD converges fast and communicates little. 7th International Conference on Learning Representations. New Orleans: ICLR, 2019. 1–19.
21 Wang Z, Fan XL, Qi JZ, *et al*. Federated learning with fair averaging. Proceedings of the 13th International Joint Conference on Artificial Intelligence. Montreal: IJCAI, 2021. 1615–1623.

(校对责编: 孙君艳)