Decentralized Event-Triggered Federated Learning with Heterogeneous Communication Thresholds

摘要

近年来，联邦学习是分布式学习研究的重点之一。现有的联邦学习研究主要集中在具有同步模型训练回合的星型拓扑学习体系结构上，其中设备的局部模型通过集中协调节点周期性地聚合。然而，在许多情况下，这样的协调节点可能不存在，这促使人们考虑完全去中心化的联邦学习，在这种联邦学习中，数据处理步骤以及模型聚合步骤都分布在各个设备上。因此本文作者提出了一种新的方法，通过网络图拓扑上的异步、事件触发的共识迭代来进行分布式模型聚合。作者考虑了每个设备上的异构通信事件阈值，这些阈值权衡了本地模型参数的变化和可用的本地资源，以决定每次迭代时聚合的好处。通过理论分析，证明了本文的方法在分布式学习的标准假设下实现了全局最优学习模型的渐近收敛，并且没有对底层拓扑的限制性连通性要求。通过数值结果表明，与联邦学习基线相比，本文的方法在通信要求方面取得了实质性的改进。

关键词：联邦学习；去中心化；异构通信阈值；事件触发

1 引言

随着计算机运算能力的不断提升，机器学习作为海量数据的分析处理技术，已经广泛服务于人类社会。然而，机器学习技术的发展过程中面临两大挑战：一是数据安全难以得到保障，隐私泄露问题极待解决；二是网络安全隔离和行业隐私，不同行业部门之间存在数据壁垒，导致数据形成“孤岛”无法安全共享，而仅凭各部门独立数据训练的机器学习模型性能无法达到全局最优化。为解决上述问题，谷歌提出了联邦学习技术。近年来，联邦学习也是分布式学习研究的重点之一。

现有的联邦学习工作主要关注模型同构的情况，其中同步（时间触发）是由一个中央协调节点周期性地对局部模型进行全局聚合的，即传统的联邦学习算法依赖于中央服务器，这要求所有的客户端都信任一个中央机构，然而，在许多情况下，这样的中心协调节点并不存在或者即使存在这样的节点也会存在设备间连接能耗高、传输效率低等问题。

因此，本文作者考虑了完全去中心化的联邦学习，提出了一种新的方法，具有异构通信阈值的事件触发联邦学习（EF-HC），是指通过网络拓扑上的异步、事件触发的共识迭代来进行分布式模型聚合的。在去中心化的联邦学习中，实现了设备之间的点对点通信，而不依赖于中央服务器，从而有助于减少通信成本、提高系统的鲁棒性，并提高更高的灵活性。

事件触发通信的好处：（1）根据每个设备模型的重要性定义事件触发条件，可以减少冗余的通信量；（2）消除设备在每次迭代中都进行通信的假设，可以缓解 straggler 问题 [4]；（3）通过限制仅在接收到新参数时进行聚合来提高每个设备的计算效率。

本文是通过理论分析和数值实验对比结果表明本文提出的方法（EF-HC）在通信要求方面取得了实质性的改进，不仅能够大大较少模型训练的通信时间，而且收敛速度在一致性图连通性下具有很好的扩展性。

2 相关工作

2.1 基于共识的分布式优化

在这项工作中，本文的重点是去中心化的联邦学习，它为分布式优化问题增加了两个独特的方面。（1）机器学习任务中各个设备的本地数据分布通常都是非独立同分布的（non-i.i.d.），这可能会对收敛性产生重大影响。（2）考虑了设备具有异构资源的现实情况 [1]。

2.2 资源高效的联邦学习

在这项工作中，本文关注的是去中心化联邦学习的新型学习拓扑结构，考虑的是完全去中心化的设置，即中心节点不可用。除了局部模型更新外，设备还与邻居进行共识迭代，以便以分布式的方式逐步减少全局机器学习损失。本文的方法结合了设备之间的异步事件触发通信，其中本地资源级别被考虑到事件阈值中，以考虑设备的异质性。

3 本文方法

3.1 本文方法概述

在传统的联邦学习中，同步是由一个中央协调器周期性地对局部模型进行全局聚合的。然而，在这项工作中，我们对不存在这样的中心节点的设置感兴趣，因此，除了使用优化技术来最小化局部损失函数外，还需要一种技术来对去中心化联邦学习方案中的参数达成共识。

为此，本文提出了具有异构通信阈值的事件触发联邦学习（EF-HC）。在 EF-HC 中，设备在模型训练期间进行点对点（P2P）通信，以同步其本地训练的模型，避免对本地数据集的过度拟合。EF-HC 算法的伪代码如图 1所示：

Algorithm 1 EF−HC procedure for device i.

Input: K, q

Initialize $k = 0$, $\mathbf{w}_i^{(0)} = \widehat{\mathbf{w}}_i^{(0)}$

while $k \leq K$ **do**

▷ **Event 1.** Neighbor Connection Event

  if device j is connected to device i **then**

    device i appends device j to its list of neighbors

    device i sends $\mathbf{w}_i^{(k)}$ and $d_i^{(k)}$ to device j

    device i receives $\mathbf{w}_j^{(k)}$ and $d_j^{(k)}$ from device j

  else if device j is disconnected from device i **then**

    device i removes device j from its list of neighbors

  end if

▷ **Event 2.** Broadcast Event

  if $\left(\frac{1}{n}\right)^{\frac{1}{q}} \left\| \mathbf{w}_i^{(k)} - \widehat{\mathbf{w}}_i^{(k)} \right\|_q \geq r\rho_i\gamma^{(k)}$ **then**

    device i broadcasts $\mathbf{w}_i^{(k)}$, $d_i^{(k)}$ to all neighbors $j \in \mathcal{N}_i^{(k)}$

    device i receives $\mathbf{w}_j^{(k)}$, $d_j^{(k)}$ from all neighbors $j \in \mathcal{N}_i^{(k)}$

    $\widehat{\mathbf{w}}_i^{(k+1)} = \mathbf{w}_i^{(k)}$

  end if

▷ **Event 3.** Aggregation Event

  if updated parameters $\mathbf{w}_j^{(k)}$ and $d_j^{(k)}$ received from neighbor j **then**

    $\mathbf{w}_i^{(k+1)} = \mathbf{w}_i^{(k)} + \sum_{j \in \mathcal{N}_i'^{(k)}} \beta_{ij}^{(k)} \left(\mathbf{w}_j^{(k)} - \mathbf{w}_i^{(k)} \right)$

  end if

▷ **Event 4.** Gradient Descent Event

  device i conducts SGD iteration $\mathbf{w}_i^{(k+1)} = \mathbf{w}_i^{(k)} - \alpha^{(k)} \mathbf{g}_i \left(\mathbf{w}_i^{(k)} \right)$

  $k \leftarrow k + 1$

end while

图 1. EF-HC 算法的伪代码

Event 1. 邻居连接事件：

如果新设备连接到设备 i 或现有设备由于图的时变性而断开连接，就会在设备 i 触发第一个事件。在这个事件中，模型参数 $w_i^{(k)}$ 和当时设备 i 的度 $d_i^{(k)}$ 会与新邻居交换。因此，这将导致两个设备上的聚合事件（事件 3）。

Event 2. 广播事件：

如果设备 i 上的 $w_i^{(k)}$ 和 $\hat{w}_i^{(k)}$ 之间的归一化差值大于阈值 $r\rho_i\gamma^{(k)}$，即 $\left(\frac{1}{n}\right)^{\frac{1}{q}} \left\| w_i^{(k)} - \hat{w}_i^{(k)} \right\|_q \geq r\rho_i\gamma^{(k)}$，则在该设备上触发广播事件。当此事件触发时，设备 i 会向其所有邻居广播其参数

$w_i^{(k)}$ 和瞬时度 $d_i^{(k)}$，并从他们那里接收相同的信息。开发阈值测量 $\left(\frac{1}{n}\right)^{\frac{1}{q}}\left\|w_i^{(k)}-\hat{w}_i^{(k)}\right\|_q$ 和条件 $r\rho_i\gamma^{(k)}$ 是本文相对于现有事件触发方案的贡献之一 [2]。在 EF-HC 中，我们设置 $\rho_i \propto \frac{1}{b_i}$，其中 b_i 是设备 i 输出链路的平均带宽。

Event 3. 聚合事件：

在设备 i 发生广播事件（事件 2）或邻居连接事件（事件 1）后，设备 i 及其所有邻居将触发聚合事件。该聚合通过分布式加权平均共识方法进行 [6]，为 $w_i^{(k+1)}=w_i^{(k)}+\sum_{j\in N_i'^{(k)}}\beta_{ij}^{(k)}\left(w_j^{(k)}-w_i^{(k)}\right)$，其中，$\beta_{ij}^{(k)}$ 是设备 i 在迭代 k 时分配给从设备 j 接收到的参数的聚合权重。

Event 4. 梯度下降事件：

每个设备 i 进行随机梯度下降（SGD）迭代，以进行局部模型训练。形式上，设备 i 得到 $w_i^{(k+1)}=w_i^{(k)}-\alpha^{(k)}g_i\left(w_i^{(k)}\right)$，其中 $\alpha^{(k)}$ 是学习率，$g_i\left(w_i^{(k)}\right)$ 是随机梯度近似值，定义为 $g_i\left(w_i^{(k)}\right)=\frac{1}{\left|S_i^{(k)}\right|}\sum_{\xi\in S_i^{(k)}}\nabla\ell_\xi\left(w_i^{(k)}\right)$，这里 $S_i^{(k)}$ 表示用于计算梯度的数据集点集（mini-batch），从本地数据集中均匀随机选择。

4 复现细节

4.1 与已有开源代码对比

与已有开源代码对比，我在复现的过程中增添了作图的代码，以便更加直观地分析实验结果，同时我也改动了部分代码，复现了 GT，并进行了调参训练。EF-HC 和 GT 都是同一个方法，最主要就是 rho 的设置，GT 需要把 rho 给设置成一个全局的常数，所以在做实验的时候设置为 constant，而在 EF-HC 里边 rho = 1/bandwidth，根据设置的 bandwidth 会变化，是一个变量。结合文章的意思，需要把 GT 的常数 rho 设置得小于 rho (= 1/bandwidth) 。而源代码里边设置的 rho 偏大，所以在很多实验的设定之下，每个设备上参数的变动没法达到阈值，也就不参与更新。只有当 label_per_agent 足够大的时候，也就是每个设备上都有足够的数据的时候，模型参数变化达到阈值 GT 才能够开始正常训练。

4.2 实验环境搭建

本文在 Fashion-MNIST 图像识别数据集上评估了本文提出的方法分类任务 [5]。采用支持向量机（SVM）和 LeNet5 神经网络模型作为分类器。

本文考虑了一个由 m=10 台设备组成的网络，其拓扑结构是根据具有连通性的随机几何图生成的 [3]。为了在设备间生成 non-i.i.d. 数据分布，每个设备只包含 10 个标签中一部分的 Fashion-MNIST 样本。对于 SVM 和 LeNet5，分别考虑 1 个和 2 个标签/设备。

本文将平均链路带宽设为 5000，并引入了一个资源异质性度量 H，$0 \le H < 1$，用来生成具有两种设备类型的网络（i）“弱”，其输出链路的平均带宽为 1000；（ii）“强”，其输出链路的平均带宽为 $\frac{5000-1000H}{1-H}$。本文设置 LeNet5 的 H=0.8，SVM 的 H=0.4。

学习率选择为 $\alpha^{(k)}=\frac{1}{\sqrt{1+k}}$，阈值衰减率设置为 $\gamma^{(k)}=\alpha^{(k)}$。

资源利用率分值定义为 $\frac{1}{m}\sum_{i=1}^{m}\frac{\sum_{j=1}^{m}v_{ij}^{(k)}}{d_i^{(k)}}\rho_i n$，对于本文提出 $\rho_i=\frac{1}{b_i}$ 的方法，该分支与平均传输时间相同，即 $\frac{1}{m}\sum_{i=1}^{m}\frac{\sum_{j=1}^{m}v_{ij}^{(k)}}{d_i^{(k)}}\frac{n}{b_i}$。

5 实验结果分析

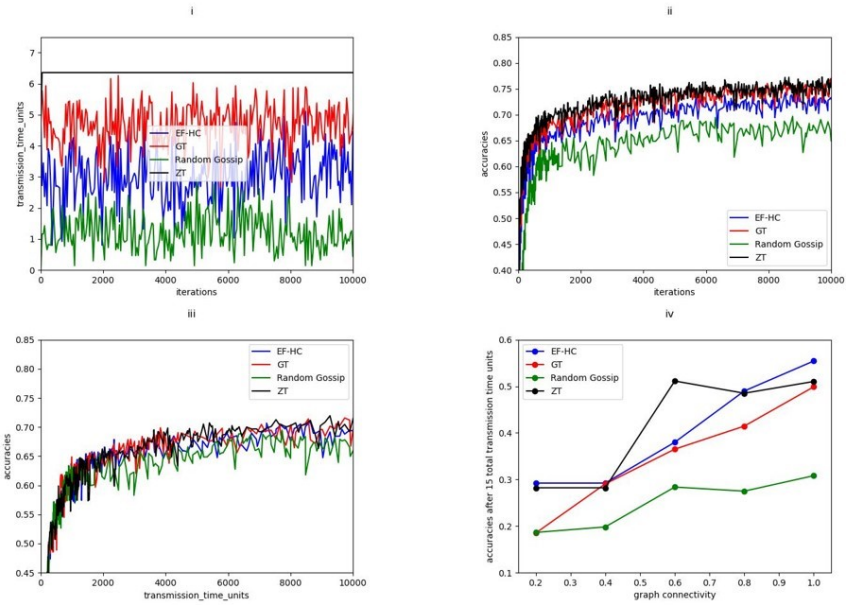
本文将 EF-HC 方法的性能与三种基线方法进行比较：

(i) 在每次迭代中都进行聚合的分布式学习，即零阈值（ZT）；

(ii) 在所有设备中都采用相同全局阈值的去中心化事件触发 FL（GT）；

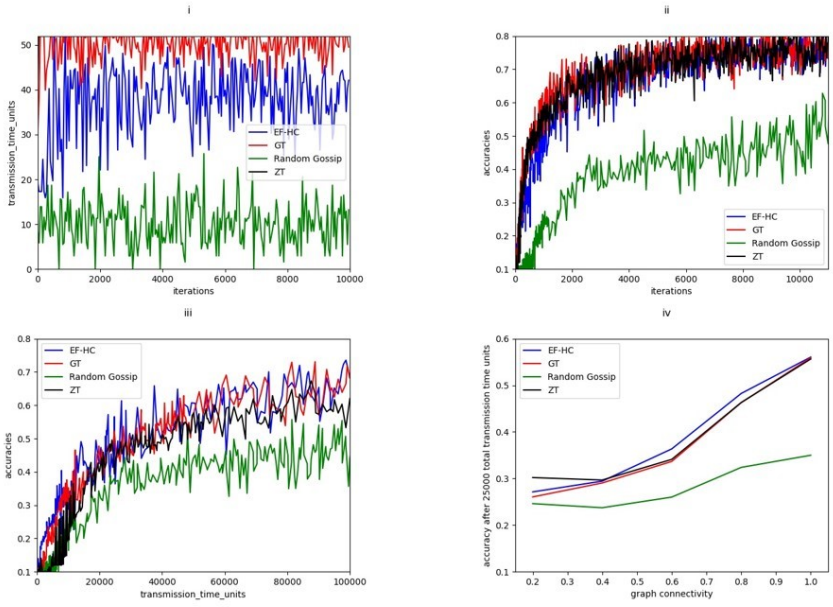
(iii) 每个设备在每次迭代中都以 $\frac{1}{m}$ 的概率进行通信的随机控制算法（RG）。

采用支持向量机（SVM）和 LeNet5 神经网络模型作为分类器的实验结果如图 2和图 3所示：



(a) SVM classifier

图 2. 采用支持向量机（SVM）作为分类器，本文的方法（EF-HC）、全局阈值（GT）、零阈值（ZT）和随机控制算法（RG）之间的性能比较



(b) LeNet5 classifier

图 3. 采用 LeNet5 神经网络模型作为分类器，本文的方法（EF-HC）、全局阈值（GT）、零阈值（ZT）和随机控制算法（RG）之间的性能比较

图 a-(i) 和图 b-(i) 显示了每种算法每次训练迭代所需的平均传输单位。可以看出，与 ZT 和 GT 相比，EF-HC 的传输延迟更少，这样就不需要从可用带宽较少的设备进行相同数量的通信和聚合，从而有助于解决 stragglers 的影响。

需要注意的是，虽然对于去中心化联邦学习算法来说，每次迭代的传输延迟更少是可取的，但当设备之间的数据分布是 non-i.i.d. 的时候，这有可能降低分类任务的准确性。因此，考虑每个传输时间单位所达到的精度，是比较多种分散算法的一个好方法。在这方面，虽然与本文提出的方法 EF-HC 相比，RG 每次迭代的传输延迟更少，但从图 a-(iii) 和图 b-(iii) 可以看出，RG 实现的模型性能明显较低，这表明本文提出的方法 EF-HC 在这些目标之间取得了有效的平衡。

图 a-(ii) 和图 b-(ii) 显示了设备每次迭代的平均精度。这些图表明了处理效率，因为它们评估了每个梯度下降计算次数的算法精度。不出所料，基线算法 ZT 每次迭代都能达到最高精度，因为它没有考虑资源效率，从而牺牲了网络资源来达到更好的精度。从图中可以看出，与 RG 不同，GT 和本文提出的 EF-HC 方法虽然使用较少的通信资源，但性能并没有显著下降。

图 a-(iii) 和图 b-(iii) 可能是最关键的结果，因为它们评估了准确性与通信时间的权衡。可以看到，无论是 SVM 分类器还是 LeNet5, 本文提出的算法 EF-HC 都可以在使用更少的传输时间的情况下达到更高的精度，即在收敛分析中有和没有模型凸性假设的情况下。这些图表明，本文的方法可以适应设备上的 non-i.i.d. 数据分布，这是联邦学习算法的一个重要特征，并且在传输时间固定的情况下，即在网络资源消耗固定的情况下，与基线方法相比，可以获得更高的精度。

图 a-(iv) 和图 b-(iv) 评估了网络连通性对本文提出的方法和基线方法的影响。由于图是在模拟中随机生成的，因此本文在蒙特卡罗实例中取了所有四种算法的平均性能，以减少随机初始化对结果的影响。可以看出，网络连通性越高，本文提出的方法和大多数基线算法的

收敛速度就越快。然而，重要的是，可以发现本文的方法在随着图连通性每次的增加，精度的提高幅度是最大的。

6 总结与展望

本文主要是针对解决传统的联邦学习存在的问题，中心协调节点不存在或者即使存在但负担较重，以及设备之间的资源异构性的情况，因此考虑了完全去中心化的联邦学习，提出了一种具有异构通信阈值的事件触发联邦学习（EF-HC）的新方法。本文对 EF-HC 进行了理论分析以及数值实验，证明了 EF-HC 下的模型训练渐近地达到了分布式学习中标准假设的全局最优模型。未来的工作重点是在不同的网络设置下，推导出设置事件触发通信条件的最优/数据驱动算法。

参考文献

[1] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecnˇ y, S. Mazzocchi, B. McMahan, and et al. Towards federated learning at scale: System design. *Proc. Machine Learn. Sys*, 1:374–388, 2019.

[2] J. George and P. Gurram. Distributed stochastic gradient descent with event-triggered communication. *Proc. AAAI Conf. Artif. Intell*, 34:7169–7178, 2020.

[3] S. Hosseinalipour, S. S. Azam, C. G. Brinton, N. Michelusi, V. Aggarwal, D. J. Love, and H. Dai. Multi-stage hybrid federated learning over large-scale d2d-enabled fog networks. *IEEE/ACM Trans. Network*, 2022.

[4] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, and H. Dai. From federated to fog learning: Distributed machine learning over heterogeneous wireless networks. *IEEE Commun. Mag*, 58(12):41–47, 2020.

[5] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[6] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Sys. Control Lett*, 53(1):65–78, 2004.