arXiv:2303.06189v1 [cs.LG] 10 Mar 2023

CS294-196 去中心化系统项目里程碑：联邦学习，但实现完全去中心化

Ram Kripa Andy Zou RyanJia KennyHuang加州大学伯克利分校 加州大学伯克利分校 加州大学伯克利分校 加州大学伯克利分校

1 摘要

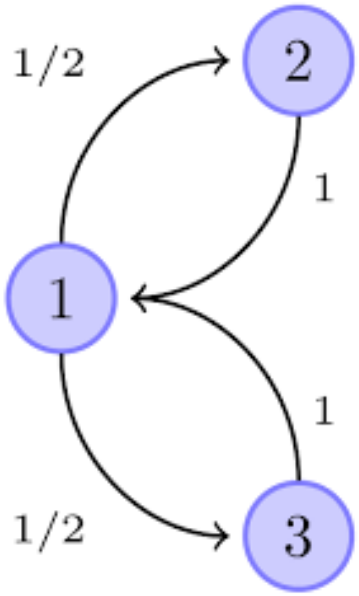
联邦学习系统使用中央服务器聚合模型更新。这种做法对带宽和资源的要求很高，还会带来隐私问题。为此，我们实现了一种点对点学习系统，其中节点使用各自的数据进行训练，并根据学习得到的信任矩阵，定期将自身参数与对等节点的参数进行加权平均。目前，我们已创建一个模型客户端框架，并利用它在所提出的系统上运行实验；这些实验使用多个虚拟节点，而这些节点实际上都运行在同一台计算机上。我们按照提案中迭代1所述的策略，验证了通过共享参数实现点对点学习的可行性。接下来，我们希望开展更多实验，并为此构建一个更易部署、适用于现实场景的系统。

2 问题陈述

传统的分布式机器学习系统通常存在隐私问题，因为在模型更新期间，各个节点与中央服务器之间需要进行通信。在中央服务器上一次性聚合参数更新的做法会损害安全性和隐私。点对点联邦学习系统则有所改进：它们按照一些简单的更新规则，在某个随机群组内更新节点，从而彻底消除对中央聚合器的需求。然而，先前提出的系统大多停留在理论层面，且设定较为简化。它们没有充分利用本地更新的结构，也没有采用经过学习的点对点更新规则。我们希望构建一个稳健的点对点联邦学习框架，以满足这一需求，并提供良好的训练时间和准确率。

3 项目描述：点对点学习

为了解决经典联邦学习系统中隐私受限且需要中央服务器的问题，我们提出了一种点对点联邦学习系统。我们旨在开发这样一种系统：网络中的各个节点存储部分数据，并各自训练一个机器学习模型实例（特别是深度神经网络），同时根据学习得到的信任矩阵，在随机选取的节点子集之间定期共享参数。这类似于使用 DeGroot模型进行社会学习的经典情形。

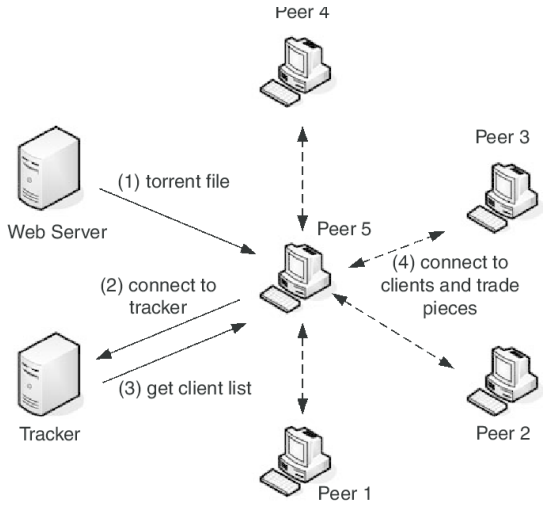


4 项目描述：系统架构

在构建点对点学习系统时，我们设想采用一种类似 BitTorrent 的架构。任何实体都可以通过初始化一台“跟踪器”服务器来启动点对点学习会话，并将其地址存储在种子文件中，再通过网络服务器或其他方式将该文件分发给许多客户端。参与点对点学习系统的客户端将下载种子文件并更新跟踪器

其元数据。

因此，跟踪器会获得参与点对点学习过程的所有客户端列表。客户端独立训练几个训练轮次，然后在与伙伴进行加权平均时，必须向跟踪器发送 HTTP 请求；跟踪器的响应中必须包含一些对等节点的地址，以便与这些节点交换模型更新。



对等节点将模型更新发送给随机选定的合作节点，不受跟踪器影响，然后继续训练过程。所有训练轮次完成后，每个节点都会针对数据拟合出一个模型。

通过这种方式，我们预计集中式跟踪器所需的带宽将很小，而且恶意实体无法直接从跟踪器获取模型更新。我们还可以在通信模型更新时实施差分隐私机制，以降低信息泄露的可能性。

为构建这一系统，我们需要创建一个框架，用于客户端（点对点学习过程中的节点）、跟踪器（我们预计采用简单的 HTTP 服务器），以及一种让客户端彼此通信而无需跟踪器介入的机制。

客户端之间可以采用 ZeroMQ、RabbitMQ 等方式进行通信。我们需要实施差分隐私机制，并监测模型更新通信所需的带宽。

5 项目成果

点对点学习系统的 Python 库；用于验证该系统运行情况的配套笔记本/脚本

介绍其开发过程及用途；说明性论文。

6 进展与初步结果

	平均准确率	std
独立同分布且均衡	92.0	0.03
非独立同分布且均衡	91.1	0.08
独立同分布 非平衡	92.1	0.04

表1：比较点对点分布式学习中不同数据设置的 MNIST 实验。非独立同分布是指不同节点的训练标签分布各不相同。非平衡是指不同节点的训练数据比例各不相同。

	平均准确率	std
独立同分布且均衡	84.2	0.10
非独立同分布且均衡	82.7	0.07
独立同分布 非平衡	83.4	0.12

表2：采用相同设置的 Fashion-MNIST 实验。

	平均准确率	std
不共享	83.2	0.20
固定共享	84.1	0.10
学习式共享	84.2	0.10

表3：Fashion-MNIST 实验比较了点对点分布式学习中不同参数共享方案的设置。不共享是指各节点分别进行训练。固定共享是指共享权重预先设定且相等。学习式共享是指共享权重从固定共享初始化，并在训练过程中学习。

我们将采用相同的整体流程，针对多个模型以及相应数据集的不同划分来评估结果。初步结果侧重于在 MNIST 和 Fashion MNIST 等数据集上训练小型逻辑回归模型，将数据划分给 6 个虚拟训练节点，并使用留出的测试集进行评估。为分析实验结果，我们考察了准确率及其标准差。

关于数据集的分布，我们将数据划分为两个互不重叠的集合，一个用于训练，另一个用于测试。在训练数据集中，我们会进一步将数据划分为训练数据集和验证数据集两个子集，比例约为 80%:20%。除了节点之间的 IID平衡划分外，我们还将考虑节点之间的非平衡 IID数据和非平衡非独立同分布数据划分，并评估

这些划分下的性能。非平衡划分是指每个节点各自拥有有意设置为不均衡的标签分布。非独立同分布划分是指每个节点拥有不同数量的训练数据点。在表1和表2中，对于更困难且更贴近实际的设置（非平衡和非独立同分布），我们观察到性能下降极小，这表明我们采用的去中心化参数共享方法具有可行性。

为了实现学习得到的参数共享，我们允许梯度通过权重传播，并在训练过程中一路反向传播以更新权重。表3显示，节点之间进行参数共享时，性能会有所提升；这使得整个网络可以共享更多数据，这些数据编码在模型参数中。结果还表明，与所有节点均等共享相比，采用学习得到的信任矩阵可能更理想。

7 未来挑战与讨论

对于点对点学习系统，我们预计需要测试其隐私、可扩展性和带宽要求。目前，这种点对点学习系统还只是纯理论构想。将该框架转化为可实际部署的系统时，我们必须确保它能够像联邦学习模型一样提供隐私保障和可扩展性。我们预计，由于跟踪器不会聚合模型更新，因此在联邦学习框架中，我们的解决方案对跟踪器的带宽要求将低于对中央服务器的带宽要求。跟踪器只是一个目录，用于访问训练过程中其他对等节点。每个对等节点所需的带宽可能会成为一个难题；在这种情况下，可以采用压缩以及限制交换参数数量的协议来缓解这一问题。

关于我们计划在下一阶段开展的实验，我们面临一些挑战。

首先，我们计划从更多维度扩展已开展的实验。正如下一节所述，我们纳入了一些迄今为止已开展实验的初步结果。我们重点研究在 3 种设置下，使用 6 个训练节点训练 MNIST数据集和 Fasion MNIST 数据集：(1) 独立同分布且平衡；(2) 非独立同分布且平衡；(3) 独立同分布且非平衡。

我们计划在不同数量的训练节点上开展实验，以观察训练节点数量与准确率之间的相关性。我们会密切监控过程中每个节点的性能。我们也可能考虑使用 MNIST 以外的数据集。最后，我们可能会考虑使用不同的数据分布策略。

随着节点数量增加、设置日益多样化，我们预计会面临计算资源短缺的挑战。我们可能还需要重新定义并扩展神经网络，以适应大量训练节点，因此需要更多训练时间。接下来，我们需要权衡取舍。在未来 2 至 3 周内，我们需要完成足够数量的实验，同时避免花太多时间等待任务结束。

第二个挑战在于准确率之外的其他成功指标。如上一节所述，我们计划实现一个类似于 BitTorrent 的系统架构。除了评估模型的准确率外，我们还需要其他指标来确定系统的效率。我们计划记录并评估不同对等节点向跟踪器发送数据以及从跟踪器获取数据所需的时间。我们还计划确定一个能够评估系统安全性的指标。

8 项目迭代与里程碑

8.1 迭代1：已完成

训练将在单台计算机上进行，节点也将在这台计算机上模拟。本次迭代旨在验证我们新颖的点对点学习方法是否可行，并确保其准确率与集中式训练相当，且高于各节点独立训练、没有参数共享时的模型准确率。本次迭代将使用多个模型及其对应的数据集，并将结果与点对点学习领域近期的领先研究进行比较。

8.2 迭代 2：预计在未来 12 天内完成

模型训练所需的数据将分布在多个物理计算节点上，并将使用消息传递协议在节点之间通信，以实现参数共享。目前建议采用的协议包括 ZeroMQ、Gossip 和 FTP。我们将实现一个虚拟“跟踪器”，但在这个阶段，跟踪器本身将启动训练过程。此外，我们还将在更大的模型上为点对点学习系统运行更多测试和实验。

8.3 迭代 3：预计下个月完成

在迭代 2 的基础上，数据将被分发，模型将在各个节点上单独训练，并定期

通过上述通信协议之一进行参数共享。此外，我们将把跟踪器实现为 HTTP 服务器，并使用 torrent 文件测试系统，模拟其在真实世界中的部署情况。

9 分工

9.1 迭代 2

Ryan 和 Ram：使用客户端和追踪器类构建真实世界系统

Andy 和 Kenny：在更大的模型和数据集上运行更多实验。

9.2 迭代 3

我们将根据可用情况和知识水平轮换角色，以便每位成员都能全面了解系统，同时也能考虑到我们期末考试和项目安排的不确定性。

参考文献

[1] Jakub Koneˇcný、H. Brendan McMahan、Felix X.Yu、Peter Richtárik、Ananda Theertha Suresh 和 Dave Bacon。《联邦学习：提升通信效率的策略》。arXiv，2017。[2] Anusha Lalitha、Osman Cihan Kilinc、Tara Javidi 和 Farinaz Koushanfar。《图上的点对点联邦学习》。arXiv，2019。https://arxiv.org/pdf/1901.11173.pdf

arXiv:2303.06189v1 [cs.LG] 10 Mar 2023

CS294-196 去中心化系统项目提案：联邦学习，但完全去中心化

Ram Kripa Andy Zou RyanJia KennyHuang加州大学伯克利分校 加州大学伯克利分校 加州大学伯克利分校 加州大学伯克利分校

1 问题陈述

传统的分布式机器学习系统往往缺乏隐私保护，因为在模型更新期间，单个节点与中央服务器之间需要进行通信。在中央服务器上一次性聚合参数更新的做法会损害安全性和隐私。点对点联邦学习系统通过按照简单的更新规则更新某个随机群组中的节点，改进了这一点，从而彻底消除了对中央聚合器的需求。然而，以往提出的系统大多停留在理论层面，且设定较为简化。它们未能充分利用本地更新的结构，也没有采用经过学习的点对点更新规则。我们希望满足这一需求，构建一个稳健的点对点联邦学习框架，并实现良好的训练时间和准确率。

2 设计

为了解决经典联邦学习系统隐私保护有限、且需要中央服务器的问题，我们提出一种点对点联邦学习系统。我们旨在开发这样一个系统：网络中的各个节点存储部分数据，并且每个节点都训练一个机器学习模型实例（尤其关注深度神经网络），同时按照学习得到的信任矩阵，定期在一个随机子集的节点之间共享参数。这与使用DeGroot模型进行社会学习的经典设定类似。

3 实现

3.1 迭代1

训练将在一台计算机上进行，节点也将在这台计算机上模拟。本次迭代的目标是验证我们提出的点对点学习新方法能否正常运行，并确保其准确率达到与集中式训练相当的水平，且高于各节点独立训练且不进行参数共享的模型。本次迭代将使用多个模型及其对应的数据集，并将结果与点对点学习领域近期的领先研究进行比较。

3.2 迭代 2

用于模型训练的数据将分配到多个物理计算节点，并通过消息传递协议在节点之间进行通信，从而实现参数共享。目前建议采用的协议包括 ZeroMQ、Gossip 和 FTP。我们将考虑并测试多个选项，并在最终迭代中采用最佳方案。

3.3 迭代 3

在迭代 2 的基础上，数据将被分配到各个节点，模型将在每个节点上单独训练，并通过上述通信协议之一定期共享参数。此外，还将使用区块链分布式账本来存储模型更新日志。

4 评估

在实现的 3 次迭代中，我们将遵循相同的大致流程来评估结果。我们会选择一些知名的机器学习数据集，例如 MNIST 、 Fashion MNIST 和 CIFAR-10。这些数据集将作为基线，用于展示我们实现的可复现性和良好准确率；希望参数共享的效果优于独立训练。在项目的后期阶段，我们可能会考虑使用一些更贴近现实场景的数据集，例如来自 Google BigQuery 的公开 Reddit 帖子数据 [1]，以模拟生产环境。为了分析实验结果，我们可能会考虑一些经典的数据分析指标，例如准确率、召回率、精确率和 F1 分数。我们还将使用混淆矩阵

以及上述数据分析工具来了解模型性能。

关于数据集的分布，我们计划将数据划分为两个互不重叠的集合，分别用于训练和测试。我们还会将训练数据集进一步划分为训练数据集和验证数据集两个子集，比例约为 80%:20%。此外，我们将进行各种交叉验证，以降低项目设计中的方差，并跟踪超参数调优结果。除了在节点之间进行 IID 平衡划分外，我们还将考虑节点之间的非平衡 IID 数据划分和均衡非 IID 数据划分，并评估这些划分下的性能。

最后，虽然上述指标更多用于理论机器学习项目，我们也会纳入适用于本项目的其他评估指标。例如，我们将评估不同通信轮次下的准确率，并横向比较上述提到的三次迭代。此外，对于迭代 2 和迭代 3，我们将评估不同通信轮次之间共享的数据量对准确率的影响。我们还希望测试客户端数量对联邦学习设计最终准确率的影响。所有这些基于准确率的分析很可能都会以折线图呈现，便于直观比较。

5 讨论

截至我们完成项目提案撰写时，大多数关于联邦学习系统的论文都以理论研究为主。本项目旨在复现各类联邦学习论文中的理论结果，并对用于参数共享的信任矩阵的学习方式进行小幅调整。具体而言，在迭代1中，我们计划通过在单台设备上模拟多个节点，构建一个基线理论联邦学习系统。在迭代 2 中，我们计划使用通信协议，构建一个由多个节点组成的可运行联邦学习系统。我们预计能够取得基于论文结果的合理成果 [1,2]。最后，在迭代 3 中，我们希望挑战自我，将区块链技术与联邦学习相结合，构建一个隐私保护的完全去中心化联邦学习系统。我们将通过充分的数据分析来展示这一方法的可行性。

6 项目里程碑与计划时间表

项目将从对我们方案有效性的初步实证验证，推进到系统构建阶段，即创建一个真正的去中心化系统，并纳入我们从迭代1中获得的发现。我们计划将时间大致平均分配给三个迭代（里程碑），每个迭代都有各自的一系列挑战。迭代1在概念层面最具挑战性，因为我们需要明确学习问题，并为信任矩阵制定合理的学习规则，然后在模拟环境中进行实证评估。迭代 2 将扩展到多台独立计算机上的多个节点，并设计一种协议，以便节点之间通信和共享。接着，迭代 3 会将我们的框架部署到区块链上。与迭代 1 相比，后续迭代需要投入更多系统构建和测试工作。总体而言，我们计划每个迭代用一到两周完成，并保留一定的灵活性。在接下来的一个月左右，我们会定期碰面，共同讨论进展和后续步骤。由于每个迭代中的工作不太容易高度并行化，我们会共享一个 Google Colab，或使用 GitHub 进行版本控制与协作。由于所有里程碑都由我们共同完成，我们会根据不同函数或模块的实现来分配工作。

参考文献

[1] Jakub Koneˇcný、H. Brendan McMahan、Felix X. Yu、Peter Richtárik、Ananda Theertha Suresh 和 Dave Bacon。《联邦学习：提升通信效率的策略》。arXiv，2017。[2] Anusha Lalitha、Osman Cihan Kilinc、Tara Javidi 和 Farinaz Koushanfar。《图上的点对点联邦学习》。arXiv，2019。https://arxiv.org/pdf/1901.11173.pdf