Exploiting Shared Representations for Personalized Federated Learning

摘要

深度神经网络已显示出从图像和文本等数据中提取通用特征表征的能力，这对各种学习任务非常有用。然而，表征学习的成果尚未在联邦环境中得到充分实现。虽然联邦环境中的数据在客户端之间往往是非同源的，但集中式深度学习的成功表明，数据往往共享一个全局特征表征，而客户端或任务之间的统计异质性则集中在标签上。基于这种直觉，我们提出了一种新颖的联邦学习框架和算法，用于学习跨客户端的共享数据表示和每个客户端的唯一本地头。我们的算法利用各客户端的分布式计算能力，在每次更新表示时，针对低维局部参数执行多次局部更新。我们证明，这种方法在线性环境中以接近最优的样本复杂度获得了对真实信息表示的线性收敛，证明它可以有效地降低每个客户端的问题维度。除了联邦学习之外，这一结果还适用于我们旨在学习数据分布间共享低维表示的各类问题，例如元学习和多任务学习。此外，大量实验结果表明，在具有异构数据的联邦环境中，我们的方法比其他个性化联邦学习方法有经验上的改进。

关键词：联邦学习；个性化学习；数据异构

1 引言

现代机器学习的许多成功案例都是在集中式环境下取得的，即在大量集中存储的数据上训练一个模型。然而，随着数据收集设备的日益增多，需要采用分布式架构来训练模型。联邦学习旨在通过提供一个平台来解决这个问题，在这个平台上，一组客户端通过利用所有客户端的本地计算能力、内存和数据，协作为每个客户端学习有效的模型。客户端之间的协调任务由一个中央服务器完成，该服务器将每一轮从客户端收到的模型进行合并，并向客户端广播更新的信息。重要的是，服务器和客户端只能使用满足通信和隐私约束的方法，无法直接应用集中式技术。

然而，联邦学习中最重要的挑战之一是数据异质性问题，即客户端任务的基础数据分布可能彼此大相径庭。在这种情况下，如果服务器和客户端学习一个共享模型（例如，通过最小化平均损失），那么得到的模型对于网络中的许多客户端来说可能表现不佳（而且也不能在不同数据之间很好地泛化 [10]）。事实上，对于某些客户来说，使用自己的本地数据（即使数据量很小）来训练本地模型可能会更好；见图 1。

最后，联邦训练好的模型可能无法很好地泛化到未参与训练过程的未见客户身上。这些情况就导致了一个问题：” 我们如何利用数据异构环境中所有客户的数据和计算能力，为每个客户学习个性化模型？”

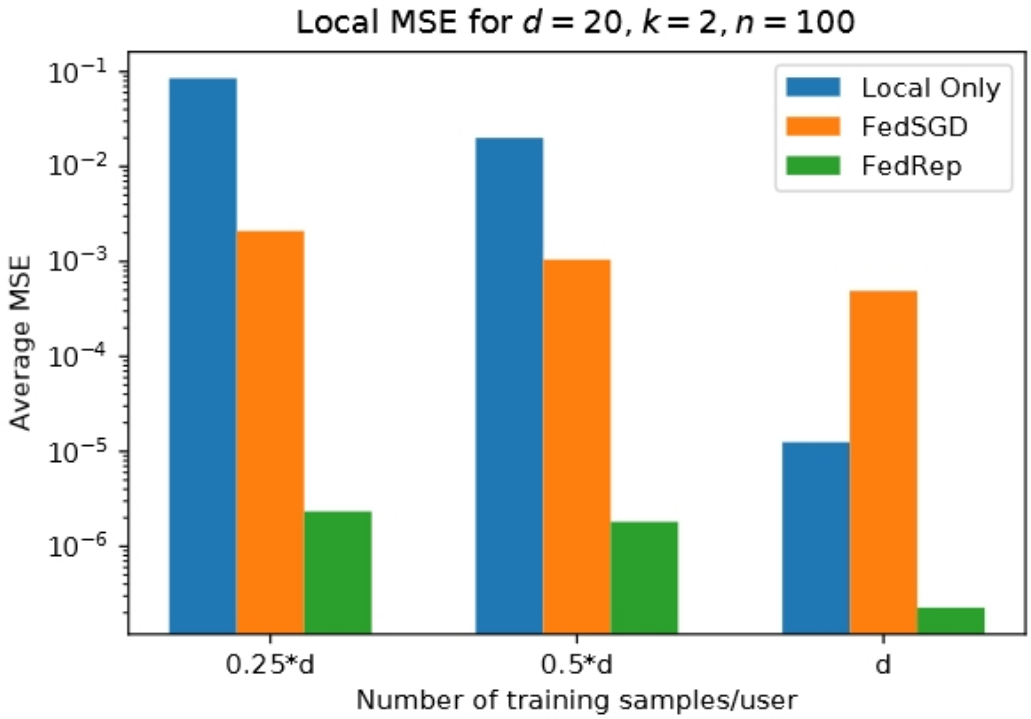

图 1. 在训练数据较少的情况下，仅进行局部训练会受到影响，而使用 FedSGD 训练单一全局模型，即使训练样本数量较多，也无法克服客户异质性。FedRep 利用客户机的共同表征，在所有情况下都能实现较小的误差。

2 相关工作

2.1 问题的提出

有 n 个客户端的联邦学习的一般形式是

$$\min_{(q_1,\ldots,q_n)\in\mathbb{Q}_n} \frac{1}{n}\sum_{i=1}^{n} f_i(q_i) \tag{1}$$

其中，f_i 和 q_i 分别是第 i 个客户的误差函数和学习模型，$\mathbb{Q}_n$ 是 n 个模型的可行集空间。我们考虑在有监督的情况下，第 i 位客户的数据由分布 (x_i, y_i) D_i 生成。学习模型 q_i ： $\mathbb{R}^d \to \mathcal{Y}$ 将输入 $x_i \in Rd$ 映射到预测标签 $q_i(xi) \in \mathcal{Y}$ 我们希望这些标签与真实标签 y_i 相似。误差 f_i 的形式是 $\mathcal{D}_i$ 上的预期风险，即 $f_i(q_i) := \mathbb{E}_{(xi,yi)\,\mathbb{D}_i}[(qi(xi), yi)]$，其中 $\mathcal{Y}\times\mathcal{Y} \mapsto \mathbb{R}$ 是一个损失函数，用于惩罚 $q_i(x_i)$ 与 y_i 的距离。

2.2 个性化联邦学习

最近有多种团队研究了关解决个性化联邦学习中的问题，例如使用局部微调 [10]、mete-Learning [6]、局部和全局模型的加法混合物 [3] 以及多任务学习 [8]。在所有这些方法中，每个客户的子问题仍然是全维的–没有学习降维局部参数集的概念。最近, [7] 也提出了一种联邦学习的表征学习方法，但他们的方法试图学习许多局部表征和一个全局头，而不是一个全局表征和许多局部头。早些时候, [2] 提出了一种学习局部头部和全局网络主体的算法，但他们的局部程序联合更新头部和主体（使用相同的更新次数），而且他们没有为自己提出的方法提供任何理论依据。与此同时，另一个工作方向研究了异构环境中的联邦学习 [5]，这些工作中基于优化的见解可用于补充我们的表述和算法。

2.3 线性表示学习

学习任务的共享表征是多任务学习的一种经典方法 [1]。具体来说，我们的目标是学习一个低维子空间，线性回归任务集合的真实回归因子就在这个子空间中。这个问题与 [4] 所考虑的线性表示学习问题最为相似。这三篇论文都显示了 ERM 目标的解向真实信息表示的统计收敛率，其中 [9]（在可实现情况下）的 $O(\frac{d}{n})$ 收敛率提高到了 $O(\frac{d}{mn})$。 [4] 还提供了类似的基于复杂度的结果，用于学习访问 ERM 甲骨文的非线性表征，但他们在线性情况下的结果要求每个任务需要 $m = \Omega(d)$ 个样本，从而减少了合作带来的好处。 [9] 进一步提出并分析了一种基于矩量法的算法来解决 ERM 问题，该算法以高效的维度依赖性 $\theta(\frac{d}{n})$ 实现了每个任务的样本复杂度，但每个任务需要 $m = \Omega(\frac{1}{n\epsilon^2})$ 个样本才能找到准确的表示。与此相反，我们证明交替最小化-后裔法只需要每个客户 $m = \Omega((d/n + log(n))log(1/\epsilon))$ 个样本就能获得精度的表示。

3 本文方法

3.1 学习共同表征

集中式机器学习的启示表明，分布在不同任务中的异构数据尽管具有不同的标签，但可以共享一种共同的表示方法；例如，在多种类型的图像或单词预测任务中共享特征。利用这

种共同的（低维）表示法，可以使用线性分类器或浅层神经网络简单地学习每个客户端的标签。

从形式上看，我们考虑的环境包括一个全局表示 $\phi : \mathbb{R}^d \to \mathbb{R}^k$, 它将数据点映射到大小为 k 的下层空间，以及客户特定的头 $h_i : \mathbb{R}^k \to \mathcal{Y}$. 第 i 个客户端的模型是客户端本地参数和表示法的组合：$q_i(x) = (h_i \circ \phi)(x)$。关键是，$k \ll d$ 意味着每个客户端必须在本地学习的参数数量很少。因此，我们可以假定，对于任何固定的表示，任何客户端的最优分类器都很容易计算，这也是下面重新编写的全局目标的动机：

$$\min_{\phi \in \Phi} \frac{1}{n} \sum_{i=1}^{n} \min_{h_i \in \mathcal{H}} f_i(h_i \circ \phi) \tag{2}$$

其中，Φ 是可行表示的类别，$\mathcal{H}$ 是可行头部的类别。在我们提出的方案中，客户通过合作使用所有客户的数据来学习全局模型，而他们则使用自己的本地信息来学习自己的个性化头部。

3.2 本文方法概述

传统的联邦学习是通过服务器将模型参数发放到每一个参与训练的客户端，客户端根据自身的数据集对模型进行训练，经过若干个 epoch 后将训练好的模型参数上传至服务器，服务器将收集到的模型参数求和取平均，得到新的模型。经过多次迭代直到服务器模型达到所需精度。本文与传统的联邦学习框架不同的是：客户端并不将所有训练好的模型参数上传到服务器，而是将一部分保留到本地作为自身独特“头”。如图 2所示：对于每一个客户端需要上传至服务器的部分是模型中的 ，即全局表征部分。通过这样的方法来解决因为不同客户端的数据不同从而导致模型的精度差的问题。

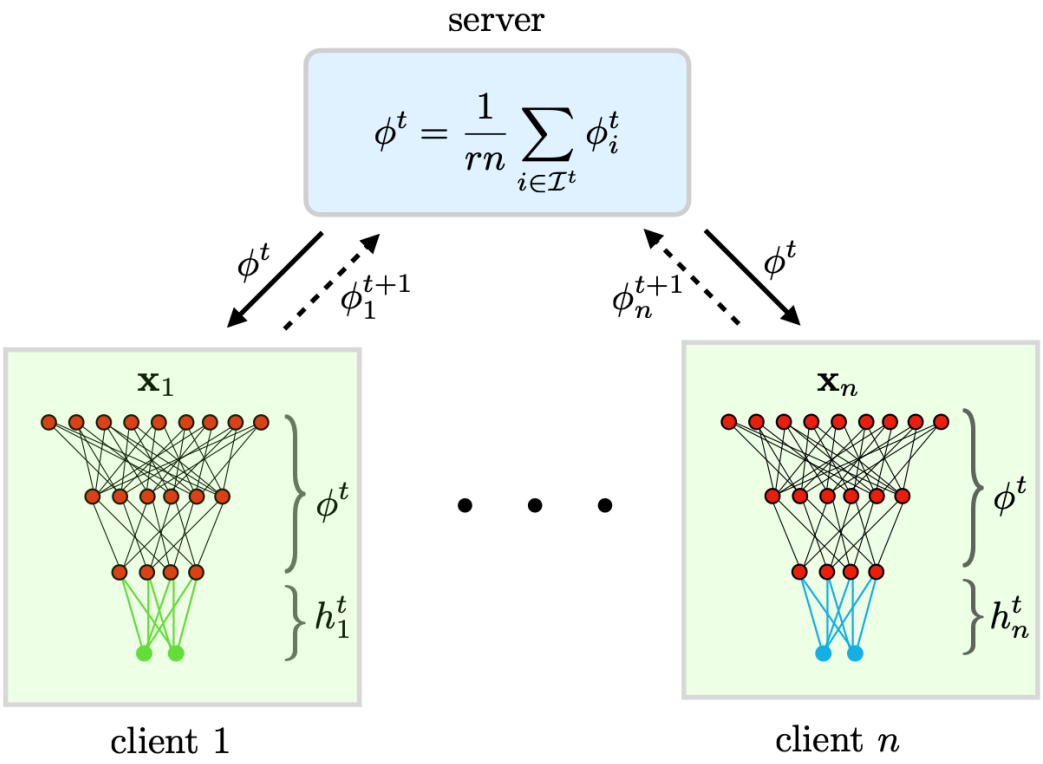

图 2. 联邦表征学习结构，其中客户端和服务器旨在共同学习一个全局表征 ϕ，而每个客户端 i 在本地学习其唯一的头部 h_i

3.3 客户端本地更新

在每一轮中，都会选择一部分恒定的 $r \in (0,1]$ 客户端执行客户端更新。在客户端更新中，客户 i 会进行 τ^g 次基于梯度的局部更新，以根据服务器发送的当前全局表示 ϕ^t 求解其最优头部。也就是说，对于 s = 1, . , τ^h，客户机 i 按以下方式更新其头部：

$$h_i^{t,s} = GRD(f_i(h_i^{t,s-1}, \phi^t), h_i^{t,s-1}, \alpha) \tag{3}$$

其中，$GRD(f, h, \alpha)$ 是使用函数 f 相对于 h 的梯度和步长 α 对变量 h 进行更新的通用符号。例如，$GRD(f_i(h_i^{t,s-1}, \phi^t), h_i^{t,s-1}, \alpha)$ 可以是梯度下降法、随机梯度下降法（SGD）、带动量的 SGD 等。通常情况下，我们会选择较大的 τ^h，因为头部的局部历时越多，就意味着我们越接近于求解内部最小化，也就意味着表示的更新越精确。接下来，客户端从全局表示 ϕ^{t-1} 开始，对其表示执行 ϕ^{t-1} 局部更新：

$$\phi_i^{t,s} = GRD(f_i(h_i^{t,\tau_h}, \phi_i^{t,s-1}), \phi_i^{t,s-1}, \alpha) \tag{4}$$

当 s=1，.....τ_ϕ

3.4 服务端模型聚合

一旦关于头部和表示的本地更新结束，客户端就会参与服务器更新，向服务器发送其本地更新的表示 ϕ^t, τ_i^ϕ。然后，服务器对本地更新进行平均，计算出下一个表示 ϕ^t。

4 复现细节

4.1 实验环境搭建

pytorch 版本为 2.1.1，torchvision 版本为 0.16.1，cuda 版本为 11.8，cudnn 版本为 8005。显卡型号为 RTX-3090.

4.2 与已有开源代码对比

这篇文章通过在客户端训练本地特有的“头”，即个性化学习来解决联邦学习中不同客户端之间的数据异构。文章实验部分只对 cifar100 中 share_class 作了两种不同的设置，通过对文章实验的扩展，发现该模型在数据越接近同分布的情况下越不如基准算法如 FedAvg。因此通过对开源代码的学习，本次复现的工作如下：

1）通过对文章中未提及的参数进行实验发现，当数据异构型越强时，模型的精度越接近100，在数据接近同分布的情况下，精度越低，分析可能是模型中的“头”记住了数据集中的样本，因此尝试改变文章中所设计的“头”对应的模型的层，来尝试提高模型的精度。文章所使用的 CNN 卷积神经网络由两个卷积层、三个全连接层所组成，文章中的“head”对应的是卷积神经网络中最后一层全连接层，通过在其他条件相同的情况下，仅修改文章中的“head”所对应的神经网络结构，例如将“head”设置为最后两个全连接层，来观察模型是否有精度上的提高。

可以得出结论：想要通过改变文章中设置的“head”对应的卷积神经网络中的结构来进一步提高模型精度这种方法是行不通的。

2）经过在 cifaro100 上与 FedAvg 对比实验，得到实验结果如图 3，分析实验中的 loss 发现在数据同分布的情况下，模型收敛性很差，当对代码进行逐条阅读，发现文章的采样方式是将 cifar100 中的每一个类数据根据（share_class* 客户数量）/100 来划分等大小的子数组，因此在数据越接近同分布式，每一个客户分到类数据样本越少，通过重写文章中的采样方式，当数据接近同分布式，一样有足够的样本进行模型的训练。修改完采样方式后，在其他条件相同的情况下，进行实验，实验结果如图 4，模型在 share_class=40、60、100 的情况均比原模型提高，尤其是在 share_class=100 即数据独立同分布的情况下，提高了接近百分之 10。在提高模型精度的同时，也通过实验与基准算法 FedAvg 作对比，在不同的参数下均比基准算法要优秀，实验结果如图 5所示。

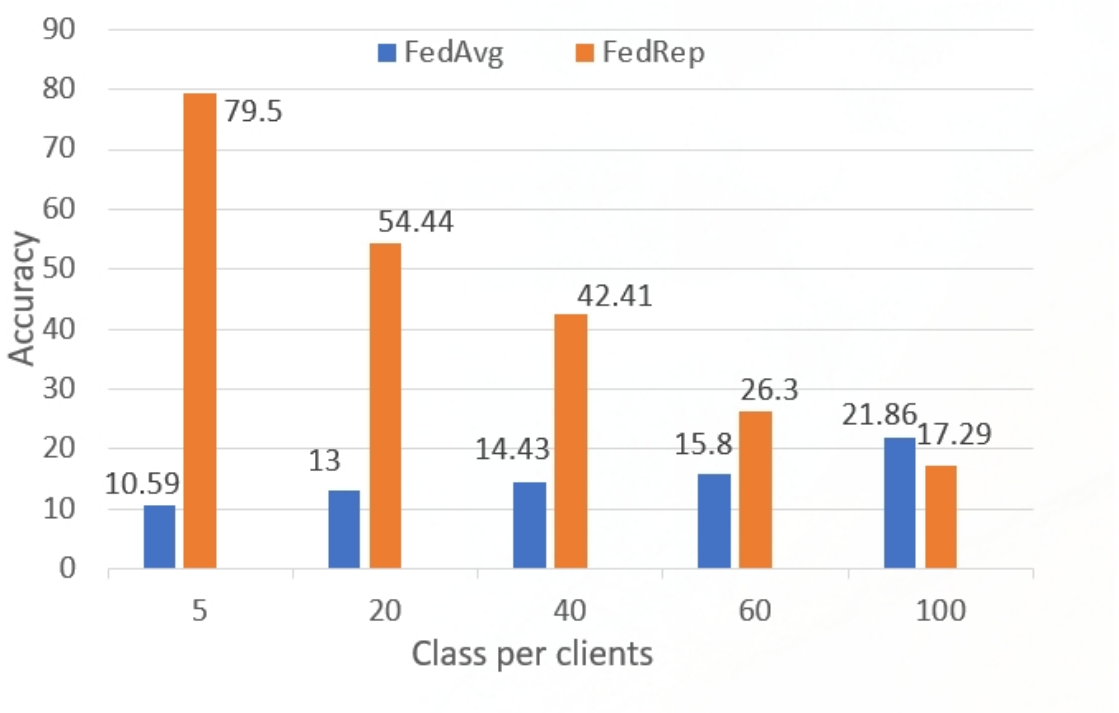

图 3. 原有的模型实验数据

图 4. 更改后的模型与原模型的实验数据

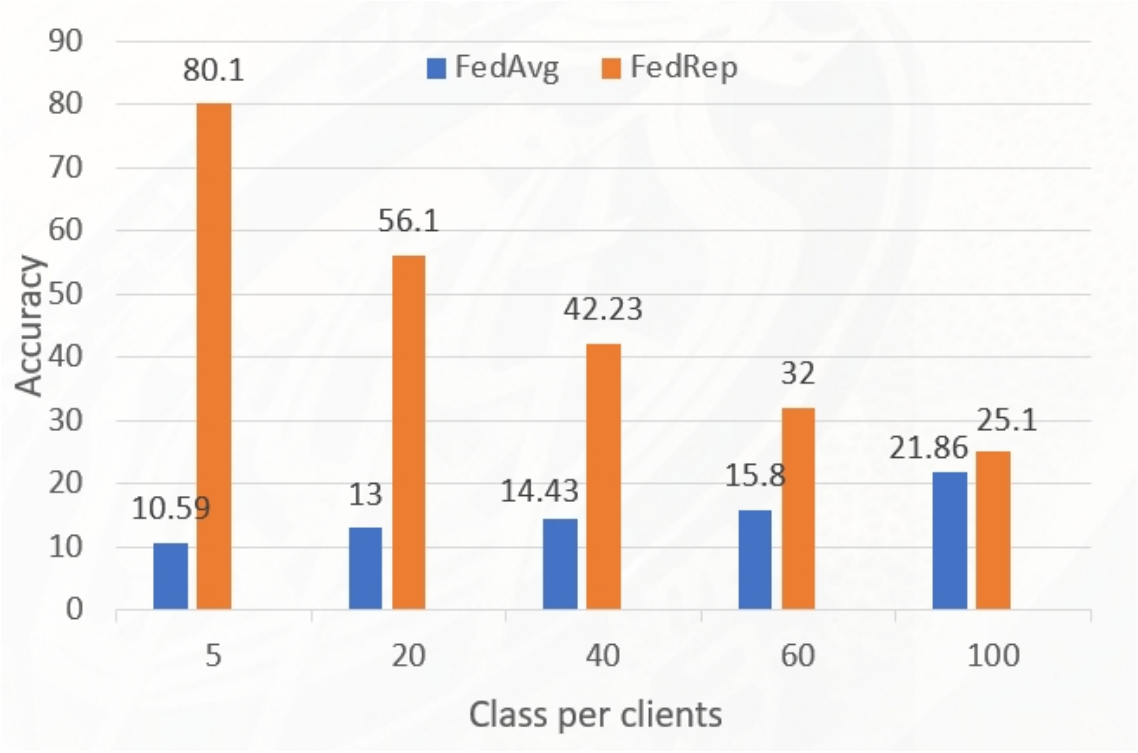

图 5. 更改后的模型与基准算法实验数据

3) 本文中通过全局共享学习 ϕ，再根据本地数据训练自身的“head”，因此对于每一个客户端模型来说，当测试数据与本地训练数据很接近时，模型输出结果的置信度就较大。当测试数据与本地数据相差较大时，模型输出的结果的置信度就很小，受集成学习的启发，尝试在服务端 Server 上构建出一个准确率较高的全局模型，因此在本文的基础上运用集成学习中

弱分类器输出结果加权求和的方法，来实现这个功能。将本文中每一个客户独立训练好的神经网络单位作为集成学习中的一个弱分类器，根据下列公式得到输出结果

$$logs = \sum_{i=1}^{n} \sum_{n=1}^{users} D_n(x_i) \tag{5}$$

$$pred = \max(logs)[1] \tag{6}$$

其中 D_i 表示不同用户的神经网络模型，$x_i \subset \mathbb{D}$ 表示测试集合的不同样本。当每一个测试数据发放到参与训练的全部客户端后，客户端 i 会根据自己的模型得到一个输出结果 $logs_i$，将所有客户端的输出结果相加，得到一个最终输出 pred，取 pred 中置信度最大的值即为本次测试样本的输出结果。

理论分析可知，模型的输出结果是由弱分类器中对该测试样本的输出结果求和取最大值，那么最终的输出结果会取决于置信度较大的分类器，因此在服务器进行数据测试时，只要所参与训练的客户端中存在对测试数据有较大把握的神经网络，就在服务器上就能输出一个较为不错的预测结果。在数据集图像数据集 Cifar10 上，保持其他参数不变，分别设置 Classes per client 的值为 2、4、6、8、10 与基准算法 FedAvg 作比较。实验结果如图 5所示。由实验结果可以看到，在数据分布较为极端的情况下，改进后的算法精度稍微落后于基准算法 FedAvg，当数据异构型逐渐减弱时，改进后的算法要比基准算法有提高。未来的工作尝试进一步改进算法，能够较大程度的提高算法的性能。

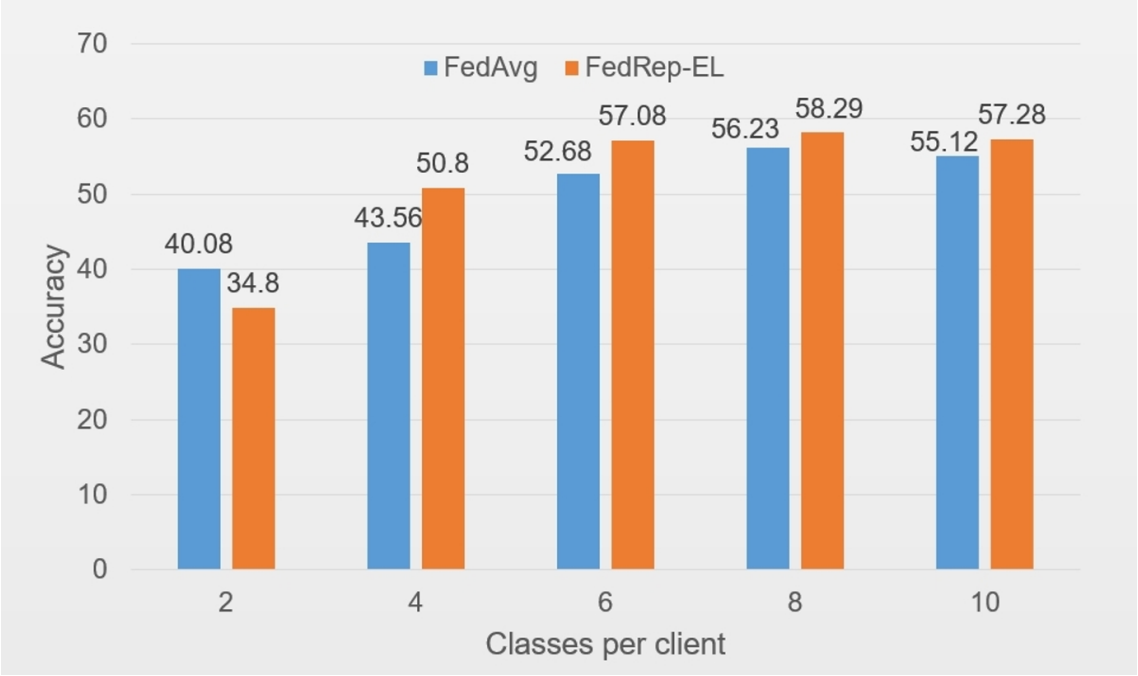

图 6. FedRep 集成算法与 FedAvg 联邦平均算法在 Cifar10 上的实验结果

5 未来工作

在本次作业的创新基础上进行完善和改进，借鉴目前集成学习中较优秀的算法，在本文的基础上进行一个创新，来解决联邦学习中异构的问题，在未来能够发表相关的论文。

6 总结与展望

本文为联邦学习引入了一种新颖的表征学习框架和算法，并为其在联邦环境中的实用性提供了理论和实证依据。特别是，我们提出的框架通过以下方式利用了联邦学习的结构：(i) 利用所有客户端的数据来学习全局表示，从而增强每个客户端的模型，并能泛化到新用户；(ii) 利用客户端的计算能力来运行多个局部更新，以学习其局部头部。我们的分析进一步表明，交替最小化-后裔可以高效地学习线性表示，因此其相关性超出了联邦学习的范围。未来的工作还包括分析 FedRep 在非线性环境中的表征学习能力。

7 收获与感悟

1. 通过本次论文复现，对联邦学习的基础相关代码编写有了较深刻的理解，从神经网络的设计，用户样本的采样，到客户端的选取，再到模型的训练与服务器对所收集到的参数的操作。通过代码的学习是了解一个领域如何具体实现的最好方式，在阅读代码的过程中确实是收获良多！

2. 本身的研究方向就是如何处理联邦学习中异构数据，提高模型的精度，通过对本篇论文以及其相关参考文献的学习，为我提供了一种解决数据异构的思路：通过表征学习来解决客户端数据分布不同的方法。

参考文献

[1] Rie Kubota Ando and Tong Zhang. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data.

[2] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated Learning with Personalization Layers, December 2019. arXiv:1912.00818 [cs, stat].

[3] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive Personalized Federated Learning, November 2020. arXiv:2003.13461 [cs, stat].

[4] Simon S. Du, Wei Hu, Sham M. Kakade, Jason D. Lee, and Qi Lei. Few-Shot Learning via Learning the Representation, Provably, March 2021. arXiv:2002.09434 [cs, math, stat].

[5] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated Learning with Compression: Unified Analysis and Sharp Guarantees.

[6] Yihan Jiang, Jakub Konecˇny, Keith Rush, and Sreeram Kannan. IMPROVING FEDERATED LEARNING PERSONALIZA- TION VIA MODEL AGNOSTIC META LEARNING.

[7] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think Locally, Act Globally: Federated Learning with Local and Global Representations.

[8] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated Multi-Task Learning.

[9] Nilesh Tripuraneni, Chi Jin, and Michael I Jordan. Provable Meta-Learning of Linear Representations.

[10] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging Federated Learning by Local Adaptation.