Main idea

ABSTRACT

移动设备和物联网设备占据了世界网络流量的一半以上，是各类机器学习尤其是联邦学习应用程序的重要数据源。联邦学习基于用户的隐私保护，允许边缘的移动设备和物联网设备在中央参数服务器的协调下训练出全局可共享的机器学习模型。

实际上，因为数据源的不同，边缘设备经常会训练不同版本的模型，或者针对相同的个任务训练不同的模型。而现有的联邦学习方案都假设参与的边缘设备共享一个公共的模型架构，并不容易实现具有异构模型之间的联邦学习。而本篇文章就针对该架构异构性挑战，寻找一种可以适应边缘设备、提高模型准确率和训练速度的联邦学习方案。

本篇文章介绍了一种针对异构模型架构的跨边缘设备联邦学习方案“FlexiFed”，并且提出了在"FlexiFed"下的适应模型异构性的三种模型聚合策略。通过在四个公共数据集上训练四种常见机器学习模型上进行实验证明了：

> ● FlexiFed 的有效性
> ● 与现有的联邦学习方案相比，FlexiFed将模型精度提高了2.6%-9.7%，并使模型收敛速度提高了1.24-4.04倍

INTRODUCTION

受限于针对私人数据的保护政策，数据从网络边缘传输到云服务器进而训练机器学习模型的策略无法很好的实现，一种直接的解决方式是直接在移动和物联网设备上去完成训练，但是由于训练数据和资源的有限，模型准确率往往不高，即存在两个主要的挑战：privacy issue、performance issue

而联邦学习（Federated Learning）则针对这两个挑战提出了一个具有前瞻性的策略：允许一组客户端在本地进行模型的训练，位于云端的中央参数服务器通过聚合本地的模型进而得到全局的模型。但是由于客户端数据的异构性，全局模型在数据分布不满足iid的客户端上表现依旧不佳，于是产生了一系列个性化的联邦学习策略：

> ● Cluster-based FL：基于聚类的联邦学习，通过对客户端的模型参数进行聚类，分别对于每个类设计个性化参数
> ● Common and personal parm：将模型参数划分为公共参数和私人参数，云端训练公共参数，本地训练私人参数

除了数据的异构性，个性化联邦学习还需要考虑模型架构的异构性（这是现有的 FL 和 personalized FL 没有考虑到的）：
① 由于客户端运算能力和精度要求不同，导致模型架构不同：VGG-16/VGG-19 or VGG-11/VGG-13
② 由于客户端的OS版本不同，导致模型架构不同：早期版本的OS仅能支持旧版本的 APP 和 ML Model

显然由于模型架构异构性（需要注意的是如果两个模型没有任何共同点，则它们间不存在异构性）的存在，数据多样性的不足会导致模型精度低，参与的客户端的不足会导致模型收敛慢，而本文提出的FlexiFed就是针对该问题而提出的一种具有创新性的联邦学习方式，核心思想是促进用户公共基本层的协同训练。客户端在本地的ML模型中识别出通用的公共基础层（Common base layer）用于协同训练，以融合这些层的参数。在实验部分，本篇文章对比了FlexiFed和baseline FL scheme、state-of-the-art FL scheme在不同数据集和模型上的效果。

BACKGROUND AND MOTIVATION

随着对高精度DNN的追求，移动设备制造商不断更新迭代硬件，进而带来软件的更新迭代，有时是为了充分利用新的OS所提供的功能，有时只是单纯得想要保证软件的正常运行。而那些无法更新的移动设备则不得不使用旧版的APP和规模较小的ML模型，由此产生了模型架构的异质性。

以一个比较典型的情况为例：存在一组用于进行人脸识别的客户端使用的模型包括 VGG-11、VGG-13、VGG-16和VGG-19，现有的联邦学习方案给出的解决方式是将客户端按照模型划分为4组并构建4个 FL 实例，针对每组再单独得进行联邦学习，这显然是与联邦学习的设计目的相违背的，既不能便利大部分的用户，也不能充分利用大量且多样的数据以获得模型的高准确率和高收敛速度。而FlexiFed则利用类似于迁移学习的思想，协同训练不同客户端的不同模型间的 common base layers。

实验过程如下：
① 设置两个客户端，它们运行不同架构的ML Model
② 两个客户端在各自的数据集上训练自己的本地模型，并将本地模型更新量上传到参数服务器进行聚合
③ 参数服务器识别到两个客户端共享的所有模型层，并采用FedAvg的方法聚合这些模型层
④ 为了进行对比，两个客户端也在无协同训练的前提下训练本地模型
⑤ 在四个数据集上进行训练以印证FlexiFed的普适性

PRELIMINARIES

- Federated Learning：

  FL的目的在于解决数据的异构性，一个FL System最小化损失函数的过程可以定义为：（加权平均的思想）

$$\min_{\omega} f(w) \triangleq \sum_{k=1}^{K} \frac{|D_k|}{|D|} f_k(w)$$
$$f_k(w) \triangleq \frac{1}{|D_k|} \sum_{i=1}^{|D_k|} \mathcal{L}(w; x_i, y_i)$$

  其中 K 为 FL System中客户端的数量，D_k 为第 k 个客户端的数据集，$f_k(w)$ 为第 k 个客户端的局部损失函数，$\mathcal{L}(w; x_i, y_i)$ 为交叉熵损失函数

  则聚合所得的一个梯度为：

$$\frac{\partial f(w)}{\partial w} = \sum_{k=1}^{K} \frac{|D_k|}{|D|} \frac{\partial f_k(w)}{\partial w} = \frac{1}{|D|} \sum_{k=1}^{K} \sum_{i=1}^{|D_k|} \frac{\partial \mathcal{L}(w; x_i, y_i)}{\partial w}$$

  FedAvg（Federated Averaging）算法思想：由于网络通信的成本较高，一般客户端是将多轮梯度下降的梯度更新量上传至参数服务器，再进行一次聚合

- Architecture Heterogeneity：

  架构异构性指的是：第 u 个客户端在 FL System中的模型 w_u 由和其他客户端模型共享的 common base layers 和与其他模型不同的 personal layers所组成，我们记为：

$$w_{u,t} = [w_{u,t}^{comm}, w_{u,t}^{pers}]$$

  同时我们将参数服务器协同训练出的全局common base layer记为：

$$w_g^{comm}$$

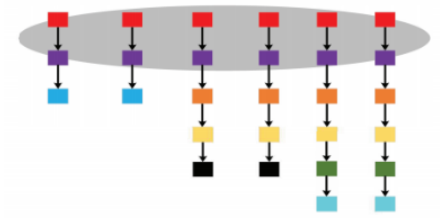
  在本次研究中，我们仅考虑的FL System涉及到应用于相同任务的具有模型架构异构性的客户端。

FLEXIFED DESIGN

本节主要探讨模型的聚合策略，FlexiFed的策略启发于迁移学习的原理：在深层神经网络中，因为较低层主要学习共同和一般的特征，所以具有高可迁移性；而较高层则主要负责用于学习个性化的特定特征。FlexiFed的关键在于参数服务器的模型聚合策略，即识别客户端的公共层并进行聚合。

- Basic-Common Strategy：

这是一个较为直观的模型聚合策略，它识别并具有所有客户端模型的公共基础层，如下图所示：



(b) Basic-Common

Input: Client set $\mathbb{U}$, learning rate lr, local epoch E, current round t
Output: Global common model $\omega_{g,t+1}^{comm}$

/* Initialize global variables. */
$\omega_{u,0} = \theta_u$ ▷ Each client initializes its local model
$t = 0$ ▷ Round counter
$\mathbb{W}_t = \emptyset$ ▷ Stores received local models from clients
/* Clients train local models on private data */
Function ClientLocalTraining($\omega_{u,t-1}, lr, E$)
  for *local epoch e from 1 to E* **do**
    updates model parameters on private data ▷ via Eq. (2)
  return $\omega_{u,t}$
/* Parameter server aggregates local models with `Basic-Common` */
Function Basic-Common($\mathbb{U}$)
  /* Receives and stores local models from clients */
  for *client* $u \in \mathbb{U}$ **do**
    $\omega_{u,t}$ = ClientLocalTraining($\omega_{u,t-1}, lr, E$)
    Add $\omega_{u,t}$ to $\mathbb{W}_t$
  Receives local model set $\mathbb{W}_t = [\omega_{1,t}, ..., \omega_{|\mathbb{U}|,t}]$
  /* Aggregates local models */
  Identifies common base layers
  $\omega_{g,t+1}^{comm} = \frac{1}{|\mathbb{U}|} \sum_u^{\mathbb{U}} \omega_{u,t}^{comm}$ ▷ via Eq. (4)
  return $\omega_{g,t+1}^{comm}$
/* Clients receive common base layers from parameter server */
Function ClientLocalUpdating($\omega_{g,t+1}^{comm}$)
  $\omega_{u,t+1} = [\omega_{g,t+1}^{comm}, \omega_{u,t}^{pers}]$ ▷ via Eq. (5)

Alg. 2: FL with `Basic-Common` under FlexiFed

训练过程可以概括为3步（FL System中共有 K 个客户端）：
Step1：所有的客户端训练自己本地模型 $w_{u,t}$ 并将 $w_{u,t}$ 发送给参数服务器（伪码中的 `ClientLocalTraining`）
Step 2：参数服务器识别它们的 common base layer并进行聚合，得到全局的common base layer参数（伪码中的 `Basic-Common`）

$$w_{g,t+1}^{comm} = \frac{1}{K} \sum_{u=1}^{K} w_{u,t}^{comm}$$

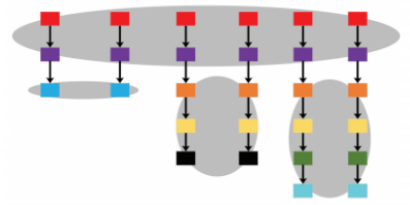
Step 3：参数服务器将计算结果返回给各个客户端，客户端进行更新：

$$w_{u,t+1} = [w_{g,t+1}^{comm}, w_{u,t}^{pers}]$$

Basic-Common Strategy具有较大的局限性，因为只有当客户端间的模型架构异构性较小时才能充分发挥该聚合策略的优势。

- Clustered-Common Strategy：

  在Basic-Common Strategy的基础上，按照 w_u^{pres} 的架构对模型进行分组，并在每组的personal layers上使用FedAvg策略进行聚合



(c) Clustered-Common

Input: Client set $\mathbb{U}$, learning rate lr, local epoch E, current round t
Output: Aggregated global model set $\mathbb{W}_{t+1}$

/* Initialize global variables. */
$\omega_{u,0} = \theta$; ▸ Each client initializes its local model
$t = 0$; ▸ Round counter
$\mathbb{W}_{t+1} = \emptyset$ ▸ Stores aggregated global models
/* Parameter server aggregates local models with `Clustered-Common` */
Function Clustered-Common($\mathbb{U}$)
  /* Receives and stores local models from clients */
  $\omega^{comm}_{g,t+1}$ = Basic-Common($\mathbb{U}$) ▸ via Alg. 2
  Clusters local models into C groups
  for $\mathbb{C}_i \in \mathbb{C}$ **do**
    $\omega^{pers}_{\mathbb{C}_i,t+1} = \frac{1}{|\mathbb{C}_i|}\sum_{u}^{\mathbb{C}_i} \omega^{pers}_{u,t}$ ▸ via Eq. (6), i=1, 2,..., C
    for $u \in \mathbb{C}_i$ **do**
      $\omega_{u,t+1} = [\omega^{comm}_{g,t+1}, \omega^{pers}_{\mathbb{C}_i,t+1}]$
      Add $\omega_{u,t+1}$ to $\mathbb{W}_{t+1}$
  return $\mathbb{W}_{t+1}$

Alg. 3: FL with `Clustered-Common` under FlexiFed

训练过程可以概括为4步：
Step1-Step2：与Basic-Common Strategy相同
Step3：按照 personal layers 将客户端分为 C 组，按照相同的取平均的方式对模型进行聚合：

$$C = \{C_i : i \in [1, C]\}$$

$$w^{pers}_{C_i,t+1} = \frac{1}{|C_i|}\sum_{c=1}^{|C_i|} w^{pers}_{u,t} \quad (u \in C_i)$$

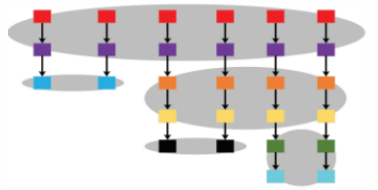
Step 4：分别组合common base layer 和 personal layer的聚合结果

$$u \in C_i \Rightarrow w_{u,t+1} = [w^{comm}_{g,t+1}, w^{pers}_{C_i,t+1}]$$

相比于Basic-Common，Clustered-Common释放了FlexiFed的更多潜力，并且可以得到更快的收敛速度和更高的准确率。

- Max-Common Strategy：

  我们设想这样一个情况：一组客户端运行VGG-11、VGG-16、VGG-19，因为VGG-11和VGG-19仅共享1层卷积层，因此Basic-Common仅能聚合全部客户端的一层，而进一步地采用Clustered-Common，则会将VGG-11、VGG-16、VGG-19分为三组，VGG-16和VGG-19间还有6层可共享的卷积层则无法实现聚合，Max-Common Strategy便是在找到全部客户端的common base layer的基础上寻找部分客户端间的common personal layer，其思想如下图所示：



(d) Max-Common

Input: Client set $\mathbb{U}$, learning rate lr, local epoch E, current round t
Output: Global model set $\mathbb{W}_{t+1}$

/* Initialize global variables. */
$\omega_{u,0} = \theta$ ▷ Each device initializes local model.
$t = 0$ ▷ Round counter.
$\mathbb{W}_{t+1} = \emptyset$ ▷ Stores aggregated common layers
/* Parameter server aggregates local models with Max-Common */
Function Max-Common($\mathbb{U}$)
  $\omega_{g,t+1}^{comm}$ = Basic-Common($\mathbb{U}$) ▷ via Alg. 2
  for $u \in \mathbb{U}$ **do**
    Add $\omega_{u,t+1}^{comm}$ to $\mathbb{W}_{t+1}$
  if $|remaining\ personal\ layers| == 0$ **then**
    Break
  else
    Clusters models into C groups based on remaining personal layers
    for $\mathbb{C}_i \in \mathbb{C}$ **do**
      Max-Common($\mathbb{C}_i$) ▷ i=1, 2,..., C
  return $\mathbb{W}_{t+1}$

Alg. 1: FL with Max-Common under FlexiFed

训练过程可以概括为4步：
Step1-Step2：与Basic-Common Strategy相同
Step3：参数服务器将客户端划分为多组，保证每一组中的所有客户端的personal layer间至少共享1层common base layer，而不同组别间的客户端的personal layer不存在任何common base layer，将Basic-Common应用于每个组别
Step4：参数服务器在每组中递归地重复Step3，直到客户端的personal layer被全部处理完为止

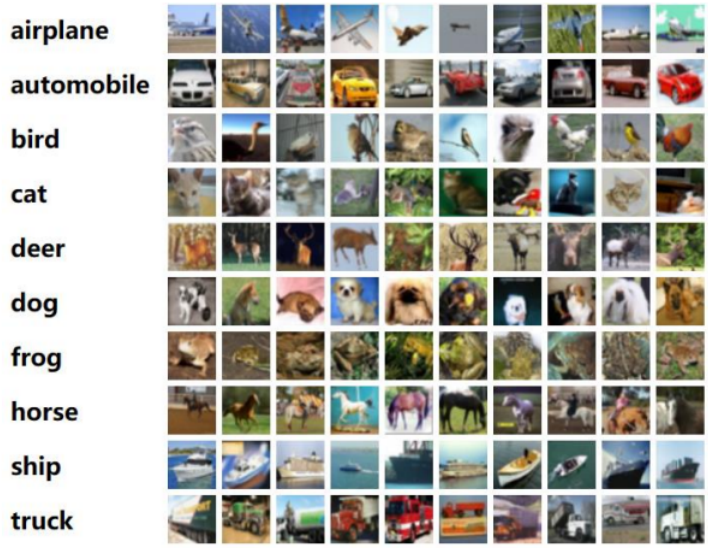
相比于Clustered-Common，Max-Common利用递归的思想充分利用了personal layer间的数据共享，得到了更好的表现

EXPERIMENTS

Experiment Setup

(1) Datasets and Implementations：

- Image Classification：VGG、ResNet利用相关的数据集进行训练
  - CIFAR-10：在本数据集中总共含有 60000 张 32×32 大小的 RGB 图像， 每个种类有 6000 张图像。总共有 50000 张用作训练集，10000 张用作测试集，具体类别如下：



  - CINIC-10：该数据集共含有 27w 张 32×32 大小的 RGB 图像，每个种类有 2.7w 张图像，等分为训练集、验证集、测试集，所涉及的类别与CIFAR-10相同

- Text Classification：CharCNN、VDCNN利用相关的数据集进行训练
  - AG News：在本数据集中总共含有4类文本：World、Sports、Business、Sci/Tec；每类文本包含 30000 个训练样本和 1900 个测试样本
- Speech Recognition：VGG、ResNet利用相关的数据集进行训练
  - Speech Commands：在本数据集中共包含65000条 1s 长的短语，共涉及30个短关键词

(2) Baseline：

用两种最具代表性的FL方案与实施Basic-Common、Clustered-Common和Max-Common的FlexiFed进行对比：

- Standalone：所有客户端在本地训练各自的模型，不进行协同训练与数据共享
- Clustered-FL：所有客户端在本地训练各自的模型，将模型交付给参数服务器，参数服务器按照模型架构进行分组（架构相同的模型分为一组），对于每个分组采用FedAvg进行模型的聚合

Overall Evaluation

本节对比了Standalone、Clustered-FL、Basic-Common、Clustered-Common、Max-Common这5种策略在处理模型架构异构性上的表现，主要从准确率和收敛速度上进行比较。

- Model Accuracy：

  ① Clustered-FL、Basic-Common、Clustered-Common、Max-Common在准确率的表现上都优于Standalone
  ② Clustered-Common、Max-Common在所有情况下表现都优于Clustered-FL，说明处理模型架构异构性需要有针对性的适当方式
  ③ Max-Common无疑是最佳的聚合策略

- Model Convergence：

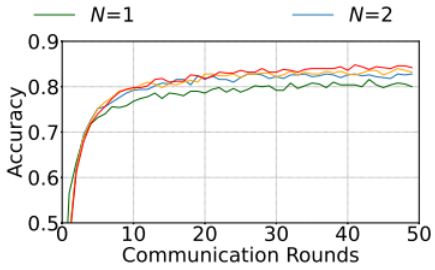
  Max-Common相较于Clustered-FL，在多种情况下收敛速度快了1.24-4.04倍

In-depth Evaluation

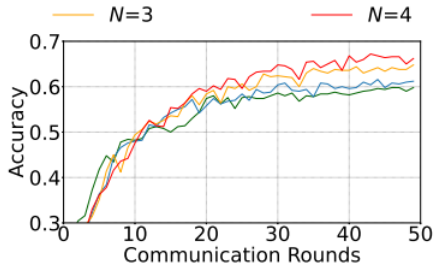
本节主要从模型版本数量、客户端数量、common layer数量等角度分析影响Max-Common性能的相关因素。

- Impact of the number of models (N)：

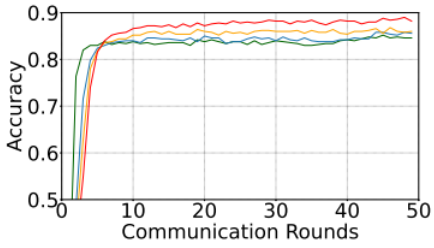
  ① 客户端包含的模型版本数量 N ($N \in \{1, 2, 3, 4\}$) 越多，模型最终收敛到的精度越高
  ② $N = 1$ 时，所有客户端运行相同的模型，共享全部layer的知识，因此收敛速度会更快一些



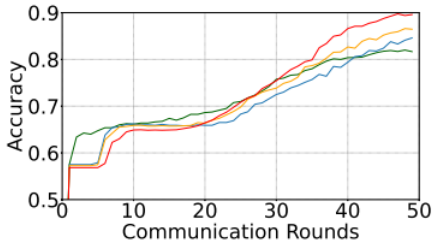
(a) VGG on CIFAR-10



(b) VGG on CINIC-10



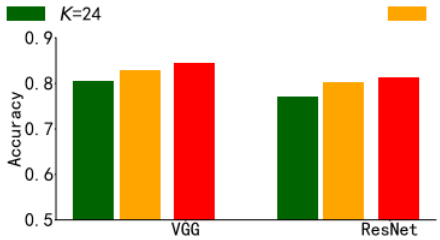
(c) VDCNN on AG News



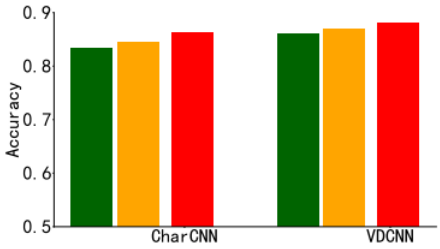
(d) ResNet on Speech Commands

- Impact of the number of clients (K):

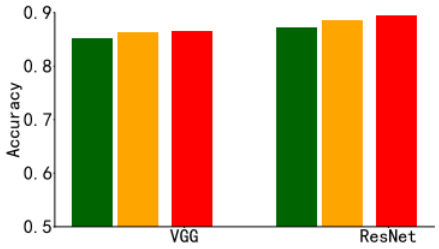
  FL System所包含的客户端数量越多，模型最终收敛的的精度越高



(a) VGG and ResNet on CIFAR-10 (b) VGG and ResNet on CINIC-10



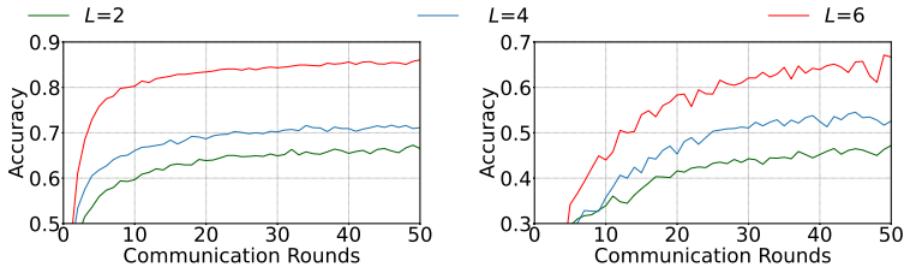
(c) CharCNN and VDCNN on AG News



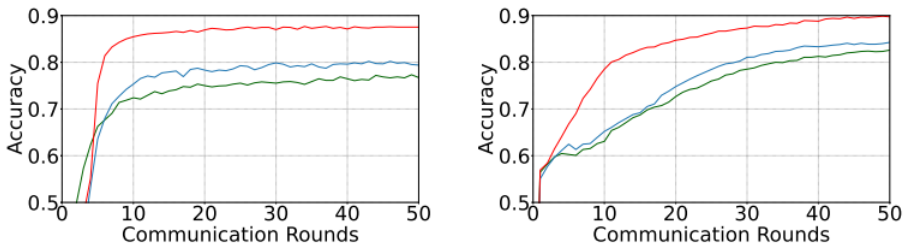
(d) VGG and ResNet on Speech Commands

- Impact of the number of common layers (L):

  显然可共享的common base layer越多，模型最终收敛到的精度越高，这与FlexiFed的思想是相互印证的，即最大化common base layer以最大限度地实现客户端间的知识共享



(a) VGG-{13, 16, 19} on CIFAR-10 (b) VGG-{13, 16, 19} on CINIC-10



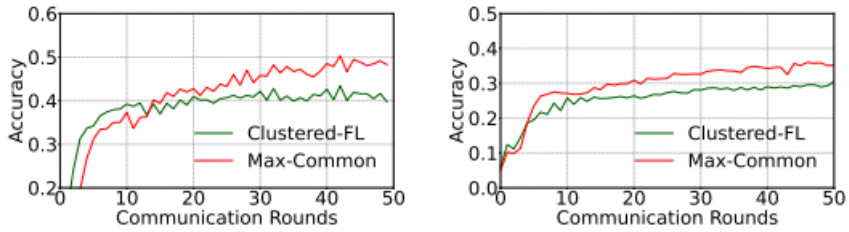
(c) VDCNN-{17, 29, 49} on AG News (d) ResNet-{32, 44, 56} on Speech Commands

- Impact of the shared classifier layers：

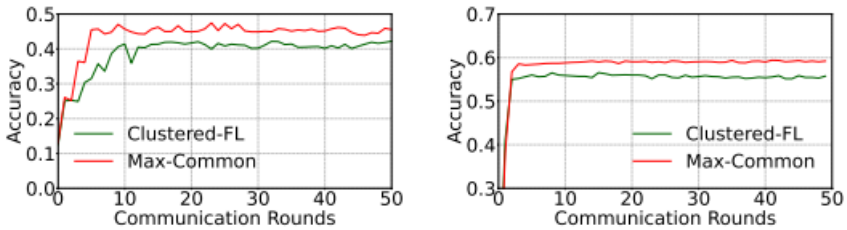
  整篇文章关注点聚焦在了最大化common base layer上，但是即便两个模型的personal base layer不存在common base layer，但是它们的全连接层仍然可能相同，那么是否要共享它们的数据呢？实验结果说明如果对全连接层进行共享，反而会降低模型的准确率，这是因为客户端的个性化比较依赖于全连接层。

- Non-IID：

  在训练数据不满足独立同分布时，Max-Common的表现依旧优于Clustered-FL，这也说明了FlexiFed的广泛适用性



(a) VGG on CIFAR-10 (b) VGG on CINIC-10



(c) VDCNN on AG News (d) ResNet on Speech Commands