从分散数据中学习深度网络的通信效率

H.布兰登-　　麦克马汉艾德-　摩尔丹尼尔-　　拉马吉塞斯-汉普森 布莱斯-阿古时代和阿卡斯

美国华盛顿州西雅图市 N 34th St.

摘要

现代移动设备可以获取大量适合学习模型的数据，这反过来又可以极大地改善设备的用户体验。例如，语言模型可以改进语音识别和文本输入，图像模型可以自动选择好照片。然而，这些丰富的数据往往对隐私敏感，数量庞大，或两者兼而有之，因此可能无法使用传统方法登录数据中心并在那里进行训练。我们主张改变这种方法，将训练数据分布在移动设备上，通过聚合本地计算的更新来学习共享模型。我们将这种分散式方法称为 "*联邦学习*"（*Federated Learning*）。

我们提出了一种基于迭代模型平均的深度网络联合学习实用方法，并进行了广泛的经验评估，考虑了五种不同的模型架构和四个数据集。这些实验证明，该方法对非平衡和非 IID 数据分布具有很强的鲁棒性，而这正是该环境的一个显著特征。通信成本是主要的限制因素，与同步随机梯度下降法相比，我们发现所需的通信回合减少了 10-100 倍。

1　导言

手机和平板电脑越来越多地成为许多人的主要计算设备[30, 2]。这些设备上功能强大的传感器（包括摄像头、麦克风和 GPS），再加上经常随身携带，意味着它们可以访问前所未有的海量数据，其中大部分是私人数据。根据这些数据建立的模型具有

出现在 20*th* 国际人工智能与统计会议（AISTATS）2017 论文集上，美国佛罗里达州劳德代尔堡。JMLR：W&CP 第 54 卷。

但数据的敏感性意味着将数据集中存储存在风险和责任。

我们研究了一种学习技术，它能让用户集体获得从这些丰富数据中训练出来的共享模型的好处，而无需集中存储这些数据。我们将这种方法称为 "*联盟学习*"（*Federated Learning*），因为学习任务是由中央*服务器*协调的参与设备（我们称之为*客户端*）组成的松散联盟解决的。每个客户端都有一个本地训练数据集，该数据集从不上传到服务器。取而代之的是，每个客户端对服务器维护的当前全局模型进行计算更新，并且只对这一更新进行通信。这是 2012 年白宫消费者数据隐私报告[39]中提出的*集中收集*或*数据最小化*原则的直接应用。由于这些更新专门用于改进当前模型，因此一旦应用，就没有理由再存储它们。

这种方法的一个主要优点是将模型训练与直接访问原始训练数据的需要分离开来。显然，仍然需要对协调训练的服务器保持一定的信任。不过，对于可以根据每个客户端的可用数据指定训练目标的应用，联合学习可以将攻击面限制在设备上，而不是设备和云上，从而显著降低隐私和安全风险。

我们的主要贡献在于：1）将移动设备分散数据的训练问题确定为一个重要的研究方向；2）选择了一种可以应用于这种环境的简单实用的算法；3）对所提出的方法进行了广泛的实证评估。更具体地说，我们引入了联邦平均算法（`FederatedAveraging algorithm`），该算法将每个客户端上的 松散随机梯度下降算法（SGD）与执行模型平均的服务器相结合。我们对这一算法进行了大量实验，证明它对不平衡和非 IID 数据分布具有鲁棒性，并能将在分散数据上训练深度网络所需的通信轮数减少几个数量级。

联合学习 联合学习的理想问题具有以下特性：1) 与数据中心通常提供的代理数据相比，在来自移动设备的真实世界数据上进行训练具有明显优势。2) 这些数据具有隐私敏感性或数据量大（与模型的大小相比）的特点，因此最好不要纯粹为了模型训练的目的而将这些数据记录到数据中心（为*集中收集*原则服务）。3) 对于超级可见任务，数据上的标签可以从用户交互中自然推断出来。

许多支持移动设备智能行为的模型都符合上述标准。作为两个例子，我们可以*考虑图像分类，*例如预测哪些照片最有可能在未来被多次浏览或分享；以及*语言模型，*它可以通过改进解码、下一个单词预测甚至整个回复的预测来改善语音识别和触摸屏键盘上的文本输入[10]。这两项任务的潜在训练数据（用户拍摄的所有照片和在手机键盘上输入的所有内容，包括密码、URL、信息等）都可能是隐私敏感数据。这些示例的分布也可能与容易获得的代理数据集有很大不同：聊天和文本信息中的语言使用通常与标准语言语料库（如维基百科和其他网络文档）有很大不同；人们在手机上拍摄的照片可能与典型的 Flickr 照片有很大不同。最后，这些问题的标签是直接可用的：输入的文本是自标签，用于学习语言模型；照片标签可以通过用户与照片应用程序的自然交互（删除、共享或查看哪些照片）来定义。

这两项任务都非常适合学习神经网络。在图像分类方面，众所周知，前馈深度网络，尤其是卷积网络，能够提供最先进的结果 [26, 25]。对于语言建模任务，递归神经网络，尤其是 LSTM，已经取得了最先进的成果[20, 5, 22]。

隐私 联合学习与数据中心在持久化数据上进行培训相比，具有明显的隐私优势。即使持有 "匿名 "数据集，也会因为与其他数据的连接而危及用户隐私[37]。相比之下，联合学习所传输的信息是改进特定模型所需的最小更新（自然，隐私优势的强度取决于更新的内容）。[1]更新本身可以（也应该）是短暂的。它们永远不会包含更多的信息。

[1]例如，如果更新是所有本地数据损失的总梯度，而特征是一个稀疏的单词包，那么非零梯度就能准确揭示用户在设备上输入了哪些单词。相比之下，CNN 等密集模型的多个梯度之和更难成为攻击者寻求单个训练实例信息的目标（尽管攻击仍有可能发生）。

与原始训练数据相比（通过数据处理不等式），更新数据所包含的信息量通常要少得多。此外，聚合算法不需要更新的来源，因此可以通过混合网络（如 Tor [7]）或可信第三方传输更新，而无需识别元数据。我们将在本文最后简要讨论将联合学习与安全多方计算和差分隐私相结合的可能性。

联合优化 我们将联合学习中隐含的优化问题称为联合优化，并将其与分布式优化相联系（和对比）。联合优化有几个关键特性，使其有别于典型的分布式优化问题：

- **非 IID** 特定客户端的训练数据通常基于特定用户对移动设备的使用情况，因此任何特定用户的本地数据集都不能代表总体分布情况。
- **不平衡** 同样，一些用户对服务或应用程序的使用会比其他用户多得多，从而导致本地训练数据量的不同。
- **大规模分布式** 我们希望参与优化的客户端数量远远大于每个客户端的平均示例数量。
- **通讯受限** 移动设备经常处于离线状态，或者连接速度慢或连接费用高。

在这项工作中，我们的重点是优化的非 IID 和不平衡特性，以及通信约束的关键性质。已部署的联合优化系统还必须解决大量实际问题：客户端数据集会随着数据的添加和删除而发生变化；客户端可用性与本地数据分布之间存在复杂的关联（例如，讲美式英语的人的手机可能会在不同的时间插上，而讲英式英语的人的手机可能会在不同的时间插上）；以及客户端从不响应或发送损坏的更新。

这些问题超出了当前工作的范围；相反，我们使用了一种适合实验的受控环境，但仍能解决客户端可用性以及不平衡和非 IID 数据等关键问题。我们假定采用同步更新方案，通过轮次通信进行更新。有一组固定的 K 个客户端，每个客户端都有一个固定的本地数据集。在每一轮开始时，随机选择一部分客户端 C，服务器将当前的全局算法状态（如当前的模型参数）发送给每一个客户端。我们只选择一部分客户端来提高效率，因为我们的实验表明，超过一定数量后，增加更多客户端的收益会递减。然后，每个选定的客户端根据全局状态和本地数据集执行本地计算，并向服务器发送更新。然后，服务器将这些更新应用到其全局状态中，整个过程不断重复。

虽然我们关注的是非凸神经网络目标，但 所考虑的算法适用于任何形式的有限求和目标

$$\min_{w\in Rd} f(w) \quad \text{式中} \quad f(w) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(w)。 \quad (1)$$

对于机器学习问题，我们通常取 $f_i(w) = l(x_i, y_i; w)$，即使用模型参数 w 对示例 (x_i, y_i) 所做预测的损失。我们假设有 K 个客户端，数据被分割在这些客户端上，P_k 是客户端 k 上数据点的索引集，$n_k = |P_k|$。因此，我们可以将目标 (1) 改写为

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \text{ 其中 } F_k(w) = \frac{1}{n_k}\sum_{i\in P_k} f_i(w)。$$

如果分区 P_k 是通过分发训练数据形成的 P_k这就是分布式优化算法通常采用的 IID 假设；我们将这种假设不成立的情况（即 F_k 可能是f的一个任意糟糕的近似值）称为非 IID 设置。

在数据中心优化中，通信成本相对较小，计算成本占主导地位，最近的重点是使用 GPU 来降低这些成本。相比之下，在联合优化中，通信成本占主导地位--我们通常会受到 1 MB/s 或更低上传带宽的限制。此外，客户端通常只有在充电、插电和使用未计量的无线网络连接时才会自愿参与优化。此外，我们预计每个客户端每天只会参与少量的更新回合。另一方面，由于任何单个设备上的数据集与总数据集相比都很小，而且现代智能手机拥有相对较快的处理器（包括 GPU），因此与许多模型类型的通信成本相比，计算基本上是免费的。因此，我们的目标是利用额外的计算来减少训练模型所需的通信轮数。我们可以通过两种主要方式增加计算量：1）*增加并行性*，即在每轮通信之间使用更多的客户端独立工作；2）*增加每个客户端*的*计算量*，即每个客户端在每轮通信之间不执行梯度计算等简单计算，而是执行更复杂的计算。我们对这两种方法都进行了研究，但我们所实现的提速主要是由于在客户端上使用了最低水平的并行性后，在每个客户

语音识别 DNN。Zhang 等人[42] 研究了一种具有 "软 " 平均的异步方法。这些研究只考虑了集群/数据中心设置（最多 16 名工作人员，基于快速网络的壁钟时间），而没有考虑
不平衡和非 IID 的数据集，这些属性包括

对联合学习环境至关重要。我们调整
这种算法与数据中心环境中的算法所提出的问题不同，需要采用不同的方法。

Neverova 等人[29] 采用与我们类似的动机，也讨论了将敏感用户数据保存在设备上的优势。Shokri 和 Shmatikov[35]的研究在几个方面与我们的研究相关：他们专注于训练深度网络，强调在设备上保存敏感用户数据的优势。
在每一轮的测试过程中，只共享参数子集，从而降低通信成本。
端上增加了更多的计算量。

相关工作 McDon- ald 等人[28] 针对感知器和 Povey 等人[31] 针对迭代平均局部训练模型进行了分布式训练研究。

但是，它们也没有考虑不平衡和非 IID 数据，而且实证评估也很有限。

在凸设置中，分布式优化和估计问题受到了极大关注[4, 15, 33]，一些算法特别关注通信效率 [45, 34, 40, 27, 43]。除了假定凸性外，这些现有工作通常还要求客户机数量远小于每个客户机的示例数量，数据以 IID 方式分布在客户机上，并且每个节点都有相同数量的数据点--所有这些假定在联合优化设置中都被违反了。异步分布式 SGD 也被应用于神经网络的训练，如 Dean 等人 [12]，但这些方法在联合优化环境中需要的更新次数过多。分布式共识算法（如 [41]）放宽了 IID 假设，但仍不适合在非常多的客户端上进行通信受限的优化。

我们所考虑的（参数化）算法系列的一个终点是简单的单次平均，即每个客户端求解最小化其本地数据（可能是正则化的）损失的模型，然后对这些模型进行平均以产生最终的全局模型。这种方法已在 IID 数据的凸案例中得到广泛研究，众所周知，在最坏的情况下，生成的全局模型并不比在单个客户端上训练模型更好[44, 3, 46]。

2 联合平均算法

最近，深度学习的大量成功应用几乎都依赖于随机梯度下降（SGD）的变体来进行优化；事实上，许多进展都可以理解为调整模型结构（以及损失函数），使其更适于通过简单的基于梯度的方法进行优化[16]。因此，自然而然地，我们建立了用于

从 SGD 开始进行联合优化。

SGD 可以简单地应用于联合优化问题，即在每轮通信中进行一次批量梯度计算（例如在随机选择的客户端上）。这种方法计算效率高，但需要大量的训练轮次才能产生好的模型（例如，即使使用批量归一化等高级方法，Ioffe 和 Szegedy [21] 训练的 MNIST 大小为 60 的迷你批 50000 步）。我们在 CIFAR-10 实验中考虑了这一基线。

在联合环境中，让更多的客户端参与进来所需的壁钟时间成本并不高，因此我们采用了大批量同步 SGD 作为基线；Chen 等人的实验也证明了这一点。[8]表明，这种方法在数据中心环境中是最先进的，其性能优于异步方法。为了在联合环境中应用这种方法，我们在每一轮选择 C 部分的客户端，并计算这些客户端持有的所有数据的损失梯度。因此，C 控制着*全局*批量大小，$C = 1$ 相当于全批量（非随机）梯度下降。[2]我们将这种基线算法称为 `FederatedSGD`（或 `FedSGD`）。

在 $C = 1$ 和固定学习率 η 的 `FedSGD` 典型实现中，每个客户端 k *都*要计算 $g_k = aF_k(w_t)$，即其本地数据在当前模型 w_t 下的平均梯度，中央服务器汇总这些梯度并应用更新 $w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k$，因为 $\sum_{k=1}^{K} \frac{n_k}{n} g_k = af(w_t)$.等效更新如下 $\forall k,\ w_{t+1}^k \leftarrow w_t - \eta g_k$ 然后 $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$.

也就是说，每个客户端利用本地数据对当前模型进行一步梯度除痕，然后服务器对得到的模型进行加权平均。一旦算法写成这样，我们就可以在求平均值之前多次迭代本地更新 $w^k \leftarrow w^k - \eta aF_k(w^k)$，从而为每个客户端增加更多计算量。我们将这种方法称为联邦平均法（或 `FedAvg`）。计算量由三个关键参数控制：C，每轮进行计算的客户端的比例；E，每个客户端每轮对其本地数据集进行训练的次数；以及 B，用于客户端更新的本地迷你批大小。我们用 $B = \infty$ 来表示整个本地数据集被视为一个迷你批。因此，在该算法族的一个端点，我们可以取 $B = \infty$ 和 $E = 1$，这与 `FedSGD` 完全对应。对于客户端 k 个本地示例的情况下，每 $u_k = E\frac{n_k}{B}$；完整的伪代码见算法 1。

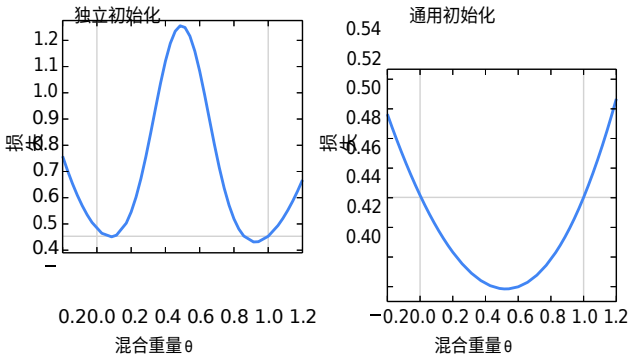


图 1：对于 50 个均匀分布的值$\theta \in$ [-0.2, 1.2]，使用$\theta w + (1 - \theta)w'$ 平均两个模型 w *和* w' 的参数所生成的模型，在完整 MNIST 训练集上的损失。模型 w 和 w' 是在不同的小型数据集上使用 SGD 训练的。左图中，w 和 w' 使用不同的随机种子初始化；右图中，使用的是共享种子。请注意不同的 y *轴*刻度。水平线给出了 w 或 w' 所达到的最佳损失（两者非常接近，分别对应于 $\theta = 0$ 和 $\theta = 1$ 时的垂直线）。在共享初始化的情况下，对模型取平均值可以显著减少总训练集的损失（远好于任一父模型的损失）。

按照古德费罗等人[17]的方法，我们可以看到当我们将两个 MNIST 数字识别模型是在不同的初始条件下训练出来的。[3]图 1 左）。在该图中，每模型 w 和 w' 分别在来自 MNIST 训练集的 600 个实例的非重叠 IID 样本上进行训练。训练是通过 SGD 进行的，固定学习率为 0.1，在 50 个样本量的迷你数据集上进行 240 次训练（或在 600 个样本量的迷你数据集上进行 $E = 20$ 次训练）。这大约是模型开始过拟合其本地数据集的训练量。

最近的研究表明，在实践中，充分过度参数化的 NN 的损失面表现出了令人惊讶的良好性能，特别是比以前认为的更不容易出现坏的局部极小值[11, 17, 9]。事实上，当我们*从相同的随机初始化*开始建立两个模型，然后在不同的数据子集上对每个模型进行独立训练时（如上所述），我们会发现天真的参数平均化效果出奇地好（图 1，右）：这两个模型的平均值$\frac{1}{2}w + \frac{1}{2}w'$，达到了显著的损失面。

对于一般的非凸目标，在参数空间平均模型可能会产生

一个任意糟糕的模型。

[2]虽然批量选择机制不同于在批次中均匀地选择单个示例。在随机情况下，FedSGD 计算出的批量梯度 g 仍然满足以下条件 $\mathbb{E}[g] = af(w)$.

在整个 MNIST 训练集上的损失，低于在任何一个小数据集上独立训练所得到的最佳模型。图 1 从随机初始化开始，但请注意，每一轮 FedAvg 都使用一个共享的起始模型 w_t，因此同样的直觉也适用。

[3]我们使用了本节所述的 "2NN "多层感知器。tion 3.

算法 1 联合平均。K *个*客户端以 k 为索引；B 是本地迷你批大小，E 是本地历时次数，η 是学习率。

服务器执行：

初始化 w_0

每轮 $t = 1$、2、.....**做**

$m \leftarrow \max(C \cdot K, 1)$

$S_t \leftarrow$（m *个*客户的随机集合）

对于并行的每个客户端 $k \in S_t$ **，做**

$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

$w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} w_{t+1}^k$ // *勘误*[4]

ClientUpdate(k, w**)：** // *在客户端* k *上运行*

$B \leftarrow$（将 P_k 分成大小为 B 的批次），**对** 1 到 E *的*每个本地历元 i **执行**

对于 $b \in B$ **做**

$w \leftarrow w - \eta \nabla l(w; b)$

将 w 返回服务器

3 实验结果

我们的灵感来自图像分类和语言建模任务，好的模型可以大大提高移动设备的可用性。对于这些任务中的每一项，我们都首先挑选了一个规模适中的代理数据集，这样我们就可以彻底研究 FedAvg 算法的超参数。虽然每个单独的训练运行相对较小，但我们为这些实验训练了超过 2000 个单独的模型。然后，我们展示了基准 CIFAR-10 图像分类任务的结果。最后，为了证明 FedAvg 在实际问题中的有效性，我们在一个大型语言建模任务中对客户端数据进行了自然划分。

我们的初步研究包括两个数据集上的三个模型系列。前两个是针对 MNIST 数字识别任务的模型[26]：1) 一个简单的多层感知器，包含 2 个隐藏层，每个隐藏层有 200 个单元，使用 ReLu 激活（总参数为 199 210），我们称之为 MNIST 2NN。2) 一个 CNN，包含两个 5x5 卷积层（第一层有 32 个通道，第二层有 64 个通道，每个通道都有 2x2 最大池化）、一个有 512 个单元和 ReLu 激活的全连接层，以及最后一个 softmax 输出层（总参数为 1,663,370 个）。要研究联合优化，我们还需要指定数据在客户端的分布方式。我们研究了将 MNIST 数据分割到客户端的两种方法：**IID**，即对数据进行洗牌，然后将其划分为 100 个客户端，每个客户端接收 600 个示例；**非 IID**，即首先按数字标签对数据进行*排序，*将其划分为 200 个大小为 300 的碎片，然后为 100 个客户端中的每个客户端分配 2 个碎片。这是一种病态的非 IID 数据分区，因为大多数客户端将只有两位数的示例，这让我们可以探索我们的算法在高度非 IID 数据上的失效程度。不过，这两个分区都是平衡的。[5]

为了进行语言建模，我们从*《威廉-莎士比亚全集》*[32] 中建立了一个数据集。我们为每部剧中至少有两句台词的每个说话角色构建了一个客户数据集。这样就产生了一个包含 1146 个客户的数据集。对于每个客户，我们将数据分成一组训练台词（该角色前 80% 的台词）和一组测试台词（该角色后 80% 的台词）。20%，四舍五入至少一行）。由此得到的数据集在训练集中有 3,564,579 个字符，在训练集中有 870,014 个字符。[6] 测试集中的字符。这些数据严重失衡

[4]本文的早期版本在此处错误地表示了对所有 K *个*客户的求和。

此外，请注意，测试集并非随机抽样，而是根据每部剧的时间顺序来区分的。此外，我们还注意到测试集不是随机的台词样本，而是根据每出戏的时间顺序进行时间上的区分。使用相同的训练/测试分割，我们还形成了一个平衡和 IID 版本的数据集，同样有 1146 个客户端。

在这些数据上，我们训练了一个堆叠字符级 LSTM 语言模型，该模型在读取一行中的每个字符后，会预测下一个字符[22]。该模型将一系列字符作为输入，并将每个字符嵌入学习到的 8 维空间。然后，嵌入的字符通过 2 个 LSTM 层进行处理，每个层有 256 个节点。最后，第二 LSTM 层的输出被发送到 softmax 输出层，每个字符有一个节点。整个模型有 866,578 个参数，我们使用 80 个字符的展开长度进行训练。

SGD 对学习率参数 η 的调整非常敏感。本文所报告的结果基于在足够宽的学习率网格上进行的训练（通常在分辨率为 10^3 或 10^6 的乘法网格上设置 11-13 个 η 值）。我们检查以确保最佳学习率处于网格的中间位置，并且最佳学习率之间没有显著差异。除非另有说明，否则我们绘制的是为每个 x 轴值单独选择的最佳学习率指标。我们发现，最佳学习率与其他参数的函数关系不大。

提高并行性 我们首先对客户机分数 C *进行了*实验，它可以控制多客户机并行性的数量。表 1 显示了改变 C 对两个 MNIST 模型的影响。我们报告了达到目标测试集准确性所需的通信轮数。为了计算出这一结果，我们为每种参数设置组合构建了一条学习曲线，如上所述优化 η，然后取测试集准确率的最佳值，使每条曲线单调递增。

[5]我们在这些数据集的非平衡版本上进行了额外的实验，发现这些数据集实际上对 `FedAvg` 来说更容易一些。

[6]我们总是用字符来指一个字节的字符串，用角色来指剧中的一个角色。

表 1：客户端分数 C 对 $E = 1$ 的 MNIST 2NN 和 $E = 5$ 的 CNN 的影响。注：$C = 0.0$ 相当于每轮一个客户端；由于我们在 MNIST 数据中使用了 100 个客户端，因此各行分别对应 1、10、20、50、和 100 个客户端。100 个客户。每个表项都给出了轮数
2NN 和 CNN 分别达到 97% 和 99% 的测试集准确率所需的通信量，以及相对于 $C = 0$ 基线的速度提升。有五次大批量运行没有在允许时间内达到目标准确率。

2NN	IID		NON-IID	
C	$B = \infty$	$B = 10$	$B = \infty$	$B = 10$
0.0	1455	316	4278	3275
0.1	1474 (1.0×)	87 (3.6×)	1796 (2.4×)	664 (4.9×)
0.2	1658 (0.9×)	77 (4.1×)	1528 (2.8×)	619 (5.3×)
0.5	- (-)	75 (4.2×)	- (-)	443 (7.4×)
1.0	- (-)	70 (4.5×)	- (-)	380 (8.6×)
CNN, $E = 5$				
0.0	387	50	1181	956
0.	1339 (1.1×)	18 (2.8×)	1100 (1.1×)	206 (4.6×)
0.	2337 (1.1×)	18 (2.8×)	978 (1.2×)	200 (4.8×)
0.	5164 (2.4×)	18 (2.8×)	1067 (1.1×)	261 (3.7×)
1.	0246 (1.6×)	16 (3.1×)	- (-)	97 (9.9×)

所有先前回合。然后，我们利用构成曲线的离散点之间的线性插值，计算出曲线与目标精度交叉的回合数。参考图 2，其中灰线表示目标，也许可以更好地理解这一点。

在 $B = \infty$ 的情况下（MNIST 将所有 600 个客户端示例作为每轮单个批次处理），增加客户端分数只有很小的优势。使用较小的批量大小 $B = 10$ 显示了使用 $C \geq 0.1$ 时的显著改进，特别是在非 IID 情况下。基于这些结果，在剩余的大部分实验中，我们将 C 固定为 0.1，这在计算效率和收敛速度之间取得了良好的平衡。比较表 1 中 $B = \infty$ 列和 $B = 10$ 列的回合数，可以发现速度有了显著提高，我们接下来将对此进行研究。

增加每个客户端的计算量 在本节中，我们固定 $C = 0.1$，并在每一轮增加每个客户端的计算量，或者减少 B，或者增加 E，或者两者兼而有之。图 2 显示，每轮增加更多本地 SGD 更新可显著降低通信成本，表 2 则量化了这些提速。每个客户端每轮的预期更新次数为 $u = (\mathsf{E}[n_k]/B)E = nE/(KB)$，其中的预期值是随机客户端 k 的抽取值。我们按照这个统计量对表 2 各部分的行进行排序。我们可以看到，通过改变 E 和 B 来增加 u 是有效的。只要 B 足够大，能充分利用客户端硬件上的并行性，那么降低 B 基本上不会带来计算时间上的损失，因此在实际操作中，这应该是第一个需要调整的参数。

表 2：`FedAvg` 与 `FedSGD` 达到目标精确度所需的通信轮数（第一行，$E = 1$，$B = \infty$）。u 列给出了 $u = En/(KB)$，即每轮的预期更新次数。

MNIST CNN，准确率 99

美国有线电视新闻网	E	B	u	IID	非 IID
FEDSGD	1	∞	1	626	483
FEDAVG	5	∞	5	179 (3.5×)	1000 (0.5×)
FEDAVG	1	50	12	65 (9.6×)	600 (0.8×)
FEDAVG	20	∞	20	234 (2.7×)	672 (0.7×)
FEDAVG	1	10	60	34 (18.4×)	350 (1.4×)
FEDAVG	5	50	60	29 (21.6×)	334 (1.4×)
FEDAVG	20	50	240	32 (19.6×)	426 (1.1×)
FEDAVG	5	10	300	20 (31.3×)	229 (2.1×)
FEDAVG	20	10	1200	18 (34.8×)	173 (2.8×)
FEDAVG	1	50	1.5	1635 (1.5×)	549 (7.1×)
FEDAVG	5	∞	5.0	613 (4.1×)	597 (6.5×)
FEDAVG	1	10	7.4	460 (5.4×)	164 (23.8×)
FEDAVG	5	50	7.4	401 (6.2×)	152 (25.7×)
FEDAVG	5	10	37.1	192 (13.0×)	41 (95.3×)

对于 MNIST 数据的 IID 分区，每个客户端使用更多计算能力可使 CNN 达到目标精度的回合数减少 35 倍，2NN 减少 46 倍（2NN 的详情请参见附录 A 中的表 4）。病理分区非 IID 数据的速度提升较小，但仍很可观（2.8 - 3.7 倍）。令人印象深刻的是，当我们对完全不同的数字对所训练的模型参数进行天真地平均时，平均结果会带来*任何*优势（与实际分歧相比）。因此，我们认为这有力地证明了这种方法的稳健性。

莎士比亚戏剧（按剧中角色）的非平衡和非 IID 分布更能代表我们所期望的现实世界应用中的数据分布。令人鼓舞的是，对于这个问题，非 IID 和不平衡数据的学习实际上要容易得多（速度提高了 95 倍，而平衡的 IID 数据提高了 13 倍）；我们推测这主要是由于某些角色拥有相对较大的本地数据集，这使得增加本地训练尤为重要。

在所有三个模型类别中，FedAvg 比基线 FedSGD 模型收敛到更高的测试集准确度水平。即使将直线延伸到图示范围之外，这一趋势仍在继续。例如，对于 CNN，$B = \infty$、$E = 1$ 的 FedSGD 模型最终在 1200 轮后达到 99.22% 的准确率（6000 轮后没有进一步提高），而 $B = 10$、$E = 20$ 的 FedAvg 模型在 300 轮后达到 99.44% 的准确率。我们推测，除了降低通信成本外，模型平均化还能产生类似于 dropout [36] 所实现的正则化优势。

我们主要关注的是泛化性能，但 FedAvg 在优化训练损失方面也很有效，甚至超过了测试集准确率的高点。我们在所有三个模型类别中都观察到了类似的行为，并在图 6 中展示了 MNIST CNN 的曲线图

莎士比亚 LSTM，54 ACCURACY

LSTM	E	B	u	IID	非 IID
FEDSGD	1	∞	1.0	2488	3906

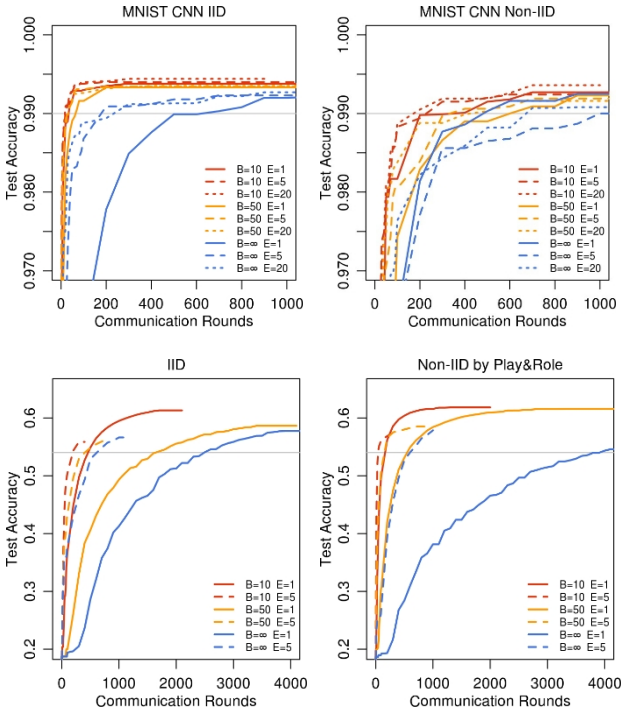


图 2：在 $C = 0.1$ 和优化的 η *条件下*，MNIST CNN（IID，然后是病理非 IID）和莎士比亚 LSTM（IID，然后是 Play&Role）的 测试集准确率与通信回合数对比。灰色线条表示表 2 中使用的目标精确度。2NN 的曲线图如下
附录 A 图 7。

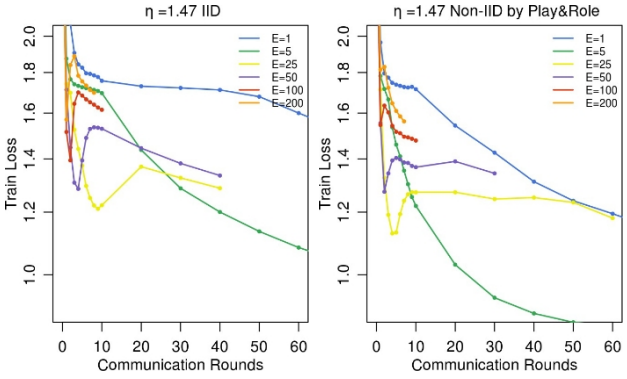
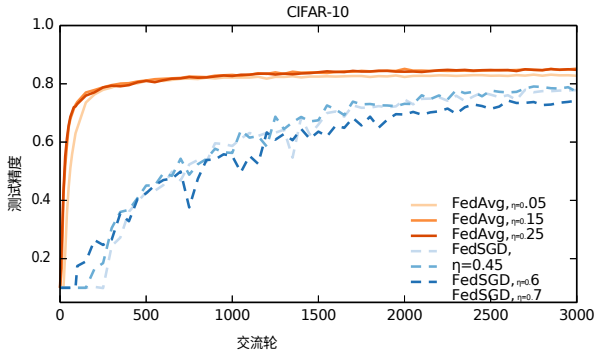


图 3：固定学习率 $\eta = 1.47$ 的莎士比亚 LSTM，在固定学习率 $B = 10$ 和 $C = 0.1$ 的情况下，在平均步骤之间训练多个局部历元（大 E）的效果。



见附录 A。

我们能否对客户数据集进行过度优化？当前的模型参数只会通过初始化影响每次客户端更新中的优化。因此，当 $E \to \infty$ 时，至少对于凸问题来说，初始条件最终应该是无关紧要的，无论初始化如何，都会达到全局最小值。即使是非凸问题，我们也可以推测，只要初始化在同一盆地，算法就会收敛到相同的局部最小值。也就是说，我们可以预期，虽然一轮平均可能会产生一个合理的模型，但更多轮的交流（和平均）不会产生进一步的改进。

图 3 显示了莎士比亚 LSTM 问题初始训练期间大 E 的影响。事实上，对于非常多的局部历时，`FedAvg` 会趋于平稳或发散。[7]这一结果表明，对于某些模型，尤其是在收敛的后期阶段，衰减

[7]请注意，由于这种行为，而且对于大 E，并非所有学习率的所有实验都运行了全部轮次，因此我们报告的是固定学习率的结果（令人惊讶的是，在 E 参数范围内，该学习率接近最优），而且没有强制要求线条具有单调性。

图 4：CI- FAR10 实验的测试精度与通信量的关系。FedSGD 采用每轮 0.9934 的学习率衰减；FedAvg 采用 $B = 50$、每轮 0.99 的学习率衰减和 $E = 5$。

每轮局部计算量（移动到更小的 E 或更大的 B）的衰减学习率同样有用。附录 A 中的图 8 给出了 MNIST CNN 的类似实验。有趣的是，对于该模型，我们没有发现 E 值越大收敛率越低的情况。不过，在下文所述的大规模语言建模任务中，我们发现 $E = 1$ 与 $E = 5$ 的性能略胜一筹（见附录 A 中的图 10）。

CIFAR 实验 我们还在 CIFAR-10 数据集 [24] 上进行了实验，以进一步验证 FedAvg。该数据集由 10 类 32x32 图像组成，包含三个 RGB 通道。其中有 50,000 个训练示例和 10,000 个测试示例，我们将这些示例划分为 100 个客户端，每个客户端包含 500 个训练示例和 100 个测试示例；由于该数据不存在自然的用户划分，我们考虑了平衡和 IID 设置。模型架构取自 TensorFlow 教程[38]，它由两个卷积层和两个全连接层组成，然后是一个线性变换层

表 3：与基准 SGD 相比，在 CIFAR10 上达到目标测试集准确率所需的回合数和速度提升。SGD 使用的迷你批大小为 100。FedSGD 和 FedAvg 使用 $C = 0.1$，FedAvg 使用 $E = 5$ 和 $B = 50$。

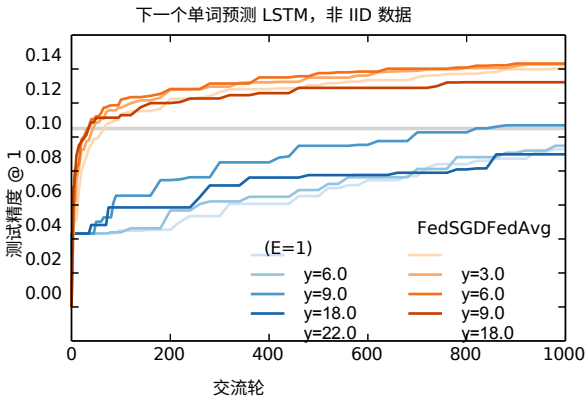
ACC.	80%		82%		85%	
新加坡元	18000	(-)	31000	(-)	99000	(-)
FEDSGD	3750 (4.8×)		6600 (4.7×)		不适用	(-)
FEDAVG	280 (64.3×)		630 (49.2×)		2000 (49.5×)	

产生对数，总共需要约 10 个[6] 参数。请注意，最先进的方法对 CIFAR 的测试准确率已达到 96.5% [19]；然而，标准的

我们所使用的模型足以满足我们的需要，因为我们的目标是评估我们的优化方法，而不是在这项任务上达到最佳可能的准确度。作为训练输入管道的一部分，我们对图像进行了预处理，包括将图像裁剪为 24x24，随机左右翻转，以及调整对比度、亮度和增白。

在这些实验中，我们考虑了一个额外的基准线，即在完整训练集（无用户分区）上使用大小为 100 的迷你批进行标准 SGD 训练。经过 197,500 次小批量更新后，我们的测试准确率达到了 86%（在联合设置中，每次小批量更新都需要一轮通信）。FedAvg 仅在 2,000 轮通信后就达到了 85% 的类似测试准确率。对于所有算法，除了初始学习率之外，我们还调整了学习率衰减参数。表 3 给出了基线 SGD、FedSGD 和 FedAvg 达到三种不同准确率目标所需的通信轮数，图 4 给出了 FedAvg 与 FedSGD 的学习率曲线。

通过对 SGD 和 FedAvg 进行大小为 $B = 50$ 的小批量实验，我们还可以观察精度与此类小批量梯度计算次数的函数关系。我们预计 SGD 在这方面会做得更好，因为每次迷你批次计算后都会采取一个连续步骤。然而，正如附录中的图 9 所示，在 C 和 E 取值适中的情况下，FedAvg 在每次迷你批次计算中取得的进展相似。此外，我们还看到，标准 SGD 和 FedAvg 在每轮只有一个客户端（$C = 0$）的情况下，准确度都会出现明显的波动，而对更多客户端进行平均后，准确度就会趋于平稳。



大规模 LSTM 实验 我们进行了大规模下一个单词预测任务的实验，以证明我们的方法在实际问题中的有效性。我们的训练数据集由来自一个大型社交网络的 1,000 万条公开帖子组成。我们按作者对帖子进行了分组，总共有超过 500,000 个客户端。该数据集真实地代表了用户移动设备上的文本输入数据类型。我们将每个客户数据集限制在最多 5000 个单词，并在 1e5 的测试集上报告准确率（在 10000 个可能性中，预测正确下一个单词的概率最高的数据部分）。

图 5：大规模语言模型单词 LSTM 的单调学习曲线。

来自不同（非训练）作者的帖子。我们的模型是一个 256 节点的 LSTM，词汇量为 10,000 个单词。每个词的输入和输出嵌入维度为 192，并与模型共同训练；总共有 4,950,544 个参数。我们使用了 10 个词的展开。

这些实验需要大量的计算资源，因此我们没有对超参数进行深入探讨：所有运行均在每轮 200 个客户端上进行训练；`FedAvg` 使用 $B = 8$ 和 $E = 1$。我们探索了 `FedAvg` 和基线 `FedSGD` 的各种学习率。图 5 显示了最佳学习率的单调学习曲线。$\eta = 18.0$ 的 `FedSGD` 需要 820 轮才能达到 10.5% 的准确率，而 $\eta = 9.0$ 的 `FedAvg` 只需 35 轮通信（比 `FedSGD` 少 23 倍）就能达到 10.5% 的准确率。我们观察到 `FedAvg` 的测试准确率差异较小，参见附录 A 中的图 10。该图还包括 $E = 5$ 的结果，其表现略逊于 $E = 1$。

4 结论和未来工作

我们的实验表明，联合学习是切实可行的，因为 `FedAvg` 只需几轮通信就能训练出高质量的模型，这一点已在多种模型架构上得到证明：多层感知器、两种不同的卷积神经网络、双层字符 LSTM 和大规模词级 LSTM。

虽然联合学习提供了许多实用的隐私优势，但通过差分权限[14, 13, 1]、安全多方计算[18]或它们的组合来提供更强的保证，是未来工作的一个有趣方向。请注意，这两类技术都非常自然地适用于 `FedAvg` 这样的同步算法。[8]

[8]在这项工作之后，Bonawitz 等人[6] 为联合学习引入了一个高效的安全聚合协议，Konec˘ny´ 等人[23] 则提出了进一步降低通信成本的算法。

参考资料

[1] Martin Abadi、 Andy Chu、 Ian Goodfellow、 Brendan McMahan、 Ilya Mironov、 Kunal Talwar 和 Li Zhang。具有差分隐私的深度学习。*第 23 届 ACM 计算机与通信安全会议（ACM CCS）*，2016 年。

[2] 莫妮卡 安德森 技术 技术 所有权： 2015. http://www.pewinternet.org/2015/10/29/technology-device-ownership-2015/, 2015.

[3] Yossi Arjevani 和 Ohad Shamir.分布式凸学习和优化的通信复杂性。*神经信息处理系统进展》第 28 期*。2015.

[4] Maria-Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour.分布式学习、通信复杂性与隐私》，*arXiv preprint arXiv:1204.3514*, 2012.

[5] Yoshua Bengio、 Re´jean Ducharme、 Pascal Vincent 和 Christian Janvin。神经概率语言模型。*J. Mach.Learn.Res.*，2003.

[6] Keith Bonawitz、Vladimir Ivanov、Ben Kreuter、Anto-nio Marcedone、 H. Brendan McMahan、Sarvar Patel、Daniel Ramage、Aaron Segal 和 Karn Seth。用户持有数据联合学习的实用安全聚合。In *NIPS Workshop on Private Multi-Party Ma- chine Learning*, 2016.

[7] David L. Chaum.无法追踪的电子邮件、回邮地址和数字假名。*Commun.ACM*，24（2），1981。

[8] Jianmin Chen, Rajat Monga, Samy Bengio, and Rafal Jozefowicz.重新审视分布式同步 SGD。在 2016 年 *ICLR 研讨会上*。

[9] 安娜-乔罗曼斯卡（Anna Choromanska）、米卡埃尔-赫纳夫（Mikael Henaff）、迈克尔-马修（Michae¨l Mathieu）、吉拉德-本-阿鲁斯（Ge´rard Ben Arous）和扬-勒存（Yann LeCun）。多层网络的损失面。*AISTATS*, 2015.

[10] 格雷格 科拉多 电脑 回复 回复 此 电子邮件。 http://googleresearch.blogspot.com/2015/11/computer-respond-to-this-email.html，2015年11月。

[11] Yann N. Dauphin, Razvan Pascanu, C¸ aglar Gu¨lc¸ehre, KyungHyun Cho, Surya Ganguli, and Yoshua Bengio.识别并解决高维非凸优化中的鞍点问题。In *NIPS*, 2014.

[12] Jeffrey Dean、Greg S. Corrado、Rajat Monga、Kai Chen、Matthieu Devin、Quoc V. Le、Mark Z. Mao、 Marc'Aurelio Ranzato、 Andrew Senior、 Paul Tucker、Ke Yang 和 Andrew Y. Ng。大规模分布式深度网络。*NIPS*, 2012.

[13] John Duchi, Michael I. Jordan, and Martin J. Wainwright.隐私意识学习。*计算机协会期刊》*，2014年。

[14] 辛西娅-德沃和亚伦-罗斯。*差分隐私的算法基础》*。理论计算机科学的基础与趋势》。Now Publishers, 2014.

[15] Olivier Fercoq, Zheng Qu, Peter Richta´rik, and Martin Taka´c.非强凸损失的快速分布式坐标下降。*信号处理机器学习（MLSP），2014 IEEE 国际研讨会*，2014 年。

[16] Ian Goodfellow、Yoshua Bengio 和 Aaron Courville.Deep learning.正在为麻省理工学院出版社准备图书，2016 年。

[17] Ian J. Goodfellow, Oriol Vinyals, and Andrew M. Saxe.定性描述神经网络优化问题。In *ICLR*, 2015.

[18] Slawomir Goryczka、Li Xiong 和 Vaidy Sunderam。具有差分优先权的安全多方聚合：比较研究。*EDBT/ICDT 2013 联合研讨会论文集*，2013 年。

[19] 本杰明-格雷厄姆分数最大池。*CoRR*，abs/1412.6071，2014.URL http://arxiv.org/ abs/1412.6071。

[20] Sepp Hochreiter 和 Ju¨rgen Schmidhuber.长短期记忆。*神经计算》*，9（8），1997年11月。

[21] Sergey Ioffe 和 Christian Szegedy.批量归一化：通过减少内部协变量偏移加速深度网络训练。在 *ICML*，2015 年。

[22] Yoon Kim、Yacine Jernite、David Sontag 和 Alexan- der M. Rush。字符感知神经语言模型。*CoRR*，abs/1508.06615，2015。

[23] Jakub Konec˘ny´、H. Brendan McMahan、Felix X. Yu、Peter Richtarik、Ananda Theertha Suresh 和 Dave Bacon。联合学习：提高通信效率的策略。2016 年 *NIPS 多方优先机器学习研讨会。*

[24] 亚历克斯-克里热夫斯基从微小图像中学习多层特征。技术报告，2009 年。

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hin- ton.利用深度卷积神经网络进行图像分类。In *NIPS*.2012.

[26] Y.LeCun, L. Bottou, Y. Bengio, and P. Haffner.基于梯度的学习应用于文档识别。*电气和电子工程师学会论文集》*，86（11），1998 年。

[27] 马晨昕、弗吉尼亚-史密斯、马丁-贾吉、迈克尔-乔丹、彼得-里希特里克、马丁-塔卡克。分布式原始二元优化中的添加与平均。*ICML*, 2015.

[28] Ryan McDonald, Keith Hall, and Gideon Mann.结构化感知器的分布式训练策略。*NAACL HLT*，2010年。

[29] Natalia Neverova、Christian Wolf、Griffin Lacey、Lex Fridman、Deepak Chandra、Brandon Barbello 和 Graham W. Taylor。从运动模式学习人类身份。*IEEE Access*, 4:1810-1820, 2016.

[30] 雅各布-普希特新兴经济体的智能手机拥有率和互联网使用率持续攀升。皮尤研究中心报告，2016年。

[31] Daniel Povey、Xiaohui Zhang 和 Sanjeev Khudanpur.利用自然梯度和参数平均法并行训练深度神经网络。2015年，*ICLR研讨会。*

[32] 威廉-莎士比亚威廉-莎士比亚全集》。可在 https：//www.gutenberg.org/ebooks/100.

[33] Ohad Shamir 和 Nathan Srebro.分布式随机优化与学习。*通信、控制和计算（Allerton）*，2014年。

[34] Ohad Shamir, Nathan Srebro, and Tong Zhang.使用近似牛顿型方法的 通信高效分布式优化。*ArXiv 预印本 arXiv:1312.7853*, 2013.

[35] Reza Shokri 和 Vitaly Shmatikov.隐私保护深度学习。*第 22Nd ACM SIGSAC 计算机与通信安全会议论文集*，CCS '15, 2015。

[36] Nitish Srivastava、Geoffrey Hinton、Alex Krizhevsky、Ilya Sutskever 和 Ruslan Salakhutdinov。辍学：防止神经网络过度拟合的简单方法。15, 2014.

[37] 拉坦娅-斯威尼简单的人口统计往往能识别出独特的人。2000.

[38] TensorFlow 团队。Tensorflow 卷积神经网络教程，2016 年。 http://www.tensorflow.org/tutorials/deep_cnn。

[39] 白宫报告。网络世界中的消费者数据隐私：全球数字经济中保护隐私和促进创新的框架》。*隐私与保密期刊》*，2013 年。

[40] 杨 天宝用计算换通信：分布式随机双坐标上升。*神经信息处理系统进展》*，2013 年。

[41] Ruiliang Zhang and James Kwok.用于共识优化的异步分布式广告。在 *ICML*.JMLR 研讨会和会议论文集，2014 年。

[42] Sixin Zhang, Anna E Choromanska, and Yann LeCun.使用弹性平均 sgd 的深度学习。In *NIPS*.2015.

[43] Yuchen Zhang and Lin Xiao.自洽经验损失的通信高效分布式优化》，*arXiv preprint arXiv:1501.00263*, 2015.

[44] Yuchen Zhang, Martin J Wainwright, and John C Duchi.统计优化的通信高效算法。In *NIPS*, 2012.

[45] Yuchen Zhang, John Duchi, Michael I Jordan, and Mar- tin J Wainwright.具有通信约束的分布式统计估计的信息论下限。*神经信息处理系统进展*》，2013 年。

[46] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J. Smola.并行化随机梯度下降。In *NIPS*.2010.

补充图表

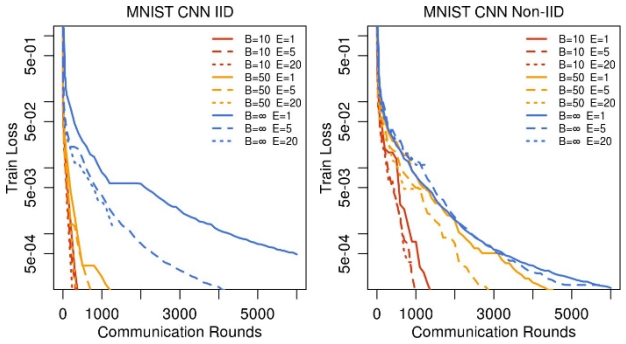


图 6：MNIST CNN 的训练集收敛。请注意，Y *轴*是对数刻度，X *轴*涵盖的训练比图 2 更多。这些图固定 $C = 0.1$。

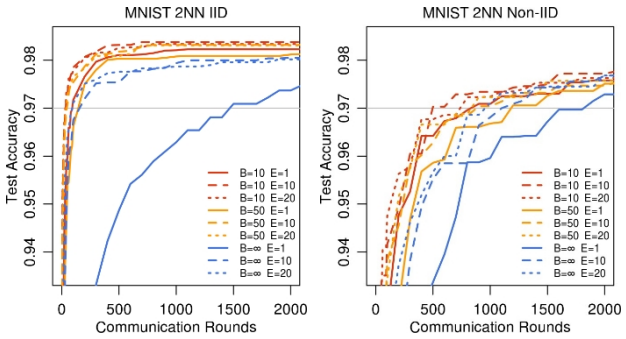


图 7：在 $C = 0.1$ 和优化的 η 条件下，MNIST 2NN 的测试集准确率与通信轮数对比。左列是 IID 数据集，右列是病态的每客户 2 位数的非 IID 数据。

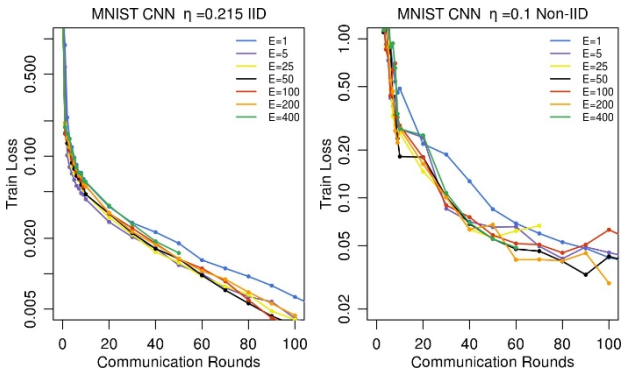


图 8：固定 $B = 10$ 和 $C = 0.1$，在平均步骤之间进行多次局部历元（大 E）训练的效果。MNIST CNN 的训练损失。请注意，由于我们的病理非 IID MNIST 数据集

表 4：在 MNIST 2NN 模型上，`FedAvg` 达到 97% 目标准确率所需的通信轮数与 `FedSGD`（第一行）相比的加速情况。

MNIST 2NN E B			u	IID	非 I I D
FE DSGD	1	∞	1	1468	1817
FE DAV G	10	∞	10	156 (9.4×)	1100 (1.7×)
FE DAV G	1	50	12	144 (10.2×)	1183 (1.5×)
FE DAV G	20	∞	20	92 (16.0×)	957 (1.9×)
FE DAV G	1	10	60	92 (16.0×)	831 (2.2×)
FE DAV G	10	50	120	45 (32.6×)	881 (2.1×)
FE DAV G	20	50	240	39 (37.6×)	835 (2.2×)
FE DAV G	10	10	600	34 (43.2×)	497 (3.7×)
FE DAV G	20	10	1200	32 (45.9×)	738 (2.5×)

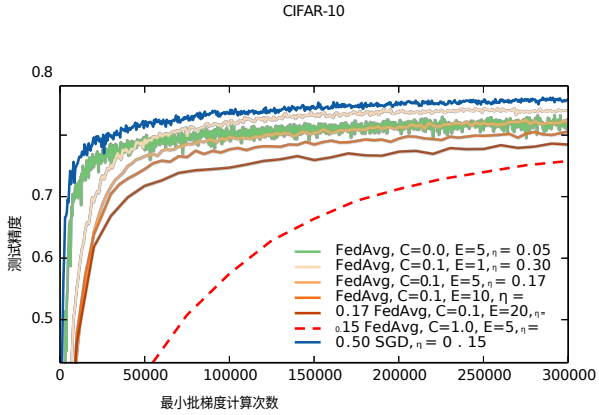
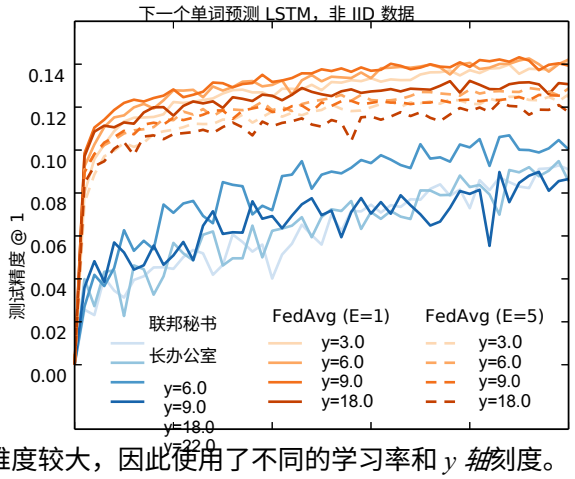


图 9：测试精度与小批量梯度计算次数的关系（$B = 50$）。基线是标准的顺序 SGD，与 `FedAvg` 相比，有不同的客户端分数 C（回顾 $C = 0$ 表示每轮一个客户端）和不同的局部历元数 E。



难度较大，因此使用了不同的学习率和 y *轴*刻度。

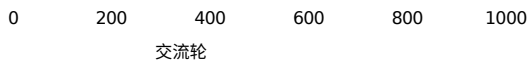


图 10：大规模语言模型单词 LSTM 的学习曲线，每 20 轮计算一次评估。与 FedSGD 相比，FedAvg 在使用较少的局部历元 E（1 对 5）时表现更好，而且各轮评估的准确率差异也更小。