题目：Federated Learning With Differential Privacy: Algorithms and Performance Analysis

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor

摘要

联邦学习允许多个在不同地理位置的客户端在中央服务器的协同下通过互联网交互模型参数来共同训练一个机器学习模型，从而保证了客户端的隐私数据始终不会暴露给攻击者。然而，攻击者依然可以通过分析这些暴露的模型参数来获取客户端的隐私信息，例如样本重构攻击和成员推断攻击。为此，本文提出了一个基于差分隐私的联邦学习框架 NbAFL，该框架在客户端和服务器传输模型参数之前生成满足特定高斯分布的零均值随机噪声来扰乱模型参数以同时实现 (ϵ, δ)-本地差分隐私和 (ϵ, δ)-全局差分隐私。本文进一步分析了 NbAFL 的理论收敛上界，该收敛上界揭示了隐私保护和收敛性能的权衡，并表明了给定隐私水平下存在最优的总迭代轮数 T^* 实现最优的性能。此外，本文将 NbAFL 框架从所有 N 个客户端参与拓展为更通用的随机采样 K 个客户端参与，并分析了部分客户端参与 NbAFL 的理论收敛上界。该上界不仅得到了和全客户端参与一样的结论，还表明了给定隐私水平下存在最优的客户端采样数量 K^* 实现最优的性能。最后，在 MNIST 数据集上的实验结果和理论分析结果保持一致，证明了理论分析结果的正确性。

关键词：联邦学习；差分隐私；收敛分析；高斯机制

1 引言

近几年来，随着人工智能的发展，机器学习已经被广泛应用于许多领域，比如自动驾驶、医疗诊断和推荐系统等。在传统的集中式机器学习中，客户端负责收集数据并将其上传到服务器，服务器负责存储数据并训练机器学习模型。然而，机器学习需要大量的训练数据以实现令人满意的模型性能，大量的数据传输不仅带来了巨大的通讯压力和昂贵的通信成本，还引发了对数据隐私泄露的担忧。为此，联邦学习 (Federated Learning, FL) [13] 作为分布式机器学习新范式被提出。在联邦学习中，服务器负责聚合和分发全局模型，客户端负责收集数据并在本地利用本地数据集训练本地模型。具体地，如图 1 所示，在每轮全局训练中，服务器将最新的全局模型参数分发给客户端，客户端在本地利用其本地数据集执行本地训练得到本地模型参数后将其传输到服务器，服务器聚合客户端上传的本地模型参数得到全局模型参数。由于客户端和服务器通过互联网进行通信，因此攻击者很容易通过监听通信信道来获取客户端和服务器交互的模型参数。

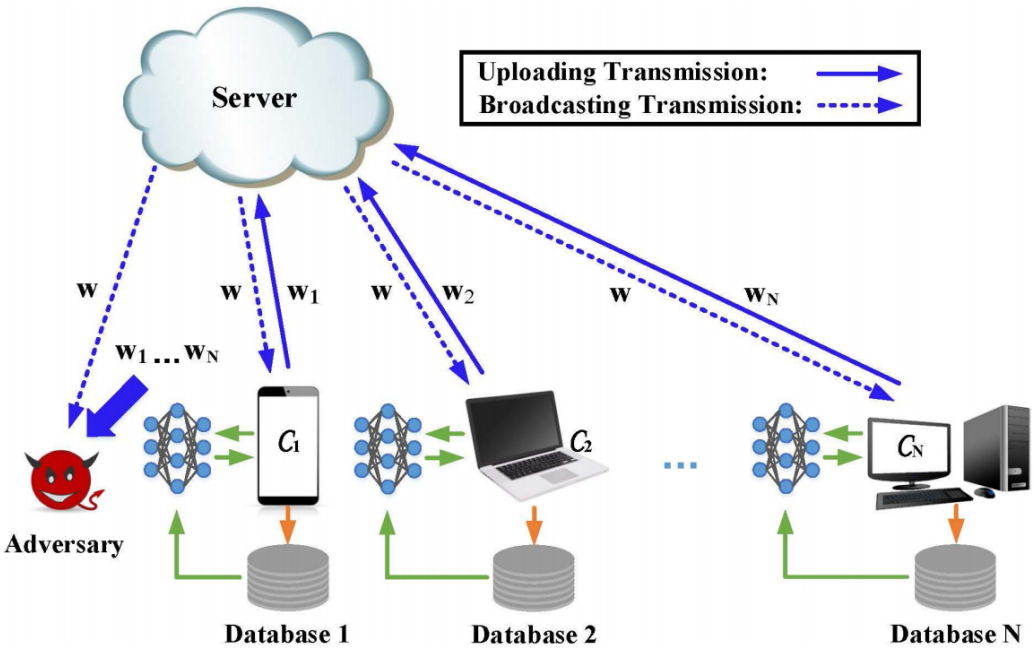

图 1. 联邦学习的训练过程以及威胁模型

虽然联邦学习保证了客户端的本地数据集只保留在本地，客户端只需要和服务器交互模型参数即可完成全局模型的训练，从而避免了客户端本地数据集的直接泄露，但是其暴露的本地模型参数可能造成客户端数据隐私的间接泄露。因为本地模型参数是本地数据集在全局模型上的训练结果，也就是说这些参数隐式地包含了本地数据集的信息。Zhu 等人 [16] 提出了样本重构攻击，攻击者可以利用本地模型参数和全局模型参数来计算本地模型参数的真实梯度。同时，攻击者随机初始化一个攻击样本并计算其在全局模型上的梯度，通过不断优化攻击样本梯度和真实梯度的距离来重构真实的训练样本。Shokri 等人 [8] 使用对抗学习和全局模型参数来训练攻击模型以推断给定样本的成员归属，即判断给定样本来自哪个客户端。

为了进一步防止模型参数泄露隐私，差分隐私 (Differentially Private, DP) 被引入到联邦学习中 [1, 11]。差分隐私通过添加服从指定分布的随机噪声使输出变得不可区分，从而达到保护数据隐私的作用。现有的基于差分隐私的研究工作主要分为两类，一类是本地差分隐私，另一类是全局差分隐私。本地差分隐私旨在保护本地数据集中每个样本的隐私，确保了攻击者无法从输出中推断出任一样本是否参与了输出的计算，因此也被称为样本级别的差分隐私。Abadi 等人 [1] 在客户端本地生成高斯噪声来扰乱训练后本地模型参数的梯度，并实时追踪每轮迭代的隐私消耗。由于攻击者无法从加噪梯度中提取原始梯度，因此样本重构攻击的成功率会大大降低。全局差分隐私旨在保护每个参与客户端的隐私，确保了攻击者无法从多个客户端的聚合输出中推断出任一客户端是否参与了聚合，这从整体上保护了每个客户端的整个本地数据集，因此也被称为用户级别的差分隐私。Geyer 等人 [3] 提出在服务器上对聚合后的梯度添加高斯随机噪声后再更新全局模型，同时根据客户端的实际梯度范数来自适应地更新梯度裁剪参数。由于随机噪声的影响，攻击者很难从加噪全局模型中推断出给定样本的成员归属，因此提高了对客户端的隐私保护。在实际的联邦学习环境中，攻击者可以轻易地获取客户端和服务器交互的本地模型参数和全局模型参数，因此，同时实现本地差分隐私和全局差分隐私来全面保护客户端的隐私是必要的。然而，上述相关工作均仅只实现了单一的差分隐私，即本地差分隐私或全局差分隐私。同时，这些研究工作并没有理论分析差分隐私噪声对联邦学习算法性能的影响，因此隐私保护和收敛性能之间的权衡尚不明确。

本文提出的 NbAFL 框架是第一个使用高斯机制同时实现本地差分隐私和全局差分隐私的差分隐私联邦学习框架。给定隐私水平 (ϵ, δ) 下，本文分别分析了本地模型参数和全局模型参数理论上所需添加的高斯随机噪声的标准差，并基于加噪本地模型参数聚合后全局模型参数中噪声的标准差和理论上全局模型参数所需添加的噪声的标准差来决定实际上全局模型所需添加的高斯随机噪声的标准差。此外，本文还分析了 NbAFL 算法的收敛上界，这是第一个分析差分隐私噪声对联邦学习收敛上界影响的理论工作。该收敛上界不仅揭示了差分隐私随机噪声对联邦学习收敛性能的影响，而且表明了给定隐私水平下存在最优的全局迭代总轮数 T^* 使 NbAFL 实现最小的收敛损失。进一步，本文将客户端采样策略引入到 NbAFL，即每轮迭代服务器随机采样 K 个客户端参与全局训练。基于随机客户端采样策略，本文分析了 K 个加噪本地模型参数聚合后全局模型参数中噪声的标准差，并推导了 K 个客户端参与下全局模型实现 (ϵ, δ)-差分隐私实际所需添加的高斯随机噪声的标准差。同时，本文推导了部分客户端参与 NbAFL 的收敛上界，该上界不仅得到了和全客户端参与 NbAFL 相同的结论，还表明了给定隐私水平和全局总迭代轮数下存在最优的客户端参与数量 K^* 实现最优的模型性能。最后，本文在流行数据集 MNIST 上进行了大量实验，实验结果进一步验证了收敛上界分析结果的正确性。因此，本文为后续的差分隐私联邦学习工作提供了算法基础和理论分析基础。

2 相关工作

2.1 联邦学习及其威胁模型

联邦学习的本质是分布式机器学习 [13]，其允许多个客户端在不传输本地隐私数据集的前提下在中央服务器的协同下共同训练一个全局模型。2017 年, McMahan 等人提出了联邦学习常用的训练算法 FedAvg 和 FedSGD [6]。FedSGD 是 FedAvg 的特例，本地迭代轮数为 1 并且训练批量为整个数据集的 FedAvg 就是 FedSGD。在 FedAvg 算法中，客户端使用随机梯度下降算法来训练本地模型，并将训练后的本地模型参数发送到服务器，服务器按比例聚合多个客户端的本地模型参数得到全局模型参数。随后，Li 等人 [5] 分析了 FedAvg 的收敛速率。为了进一步提高联邦学习的性能，Li 等人 [14] 设计了一种新的本地模型参数更新方式 FEDAC 来加速 FedAvg 的收敛速度。此外，FedProx 算法 [4] 通过在损失函数中增加一个正则项来约束本地模型和全局模型的距离从而提高了 FedAvg 在异构网络中的鲁棒性。

虽然联邦学习确保了客户端的本地数据集无需传输到服务器进行模型训练，但是攻击者很容易获取客户端和服务器交互的模型参数，并试图从中提取客户端的隐私信息。LLG [10] 和 DLG [16] 利用全局模型和本地模型计算本地模型的真实梯度来分别重构训练标签和对应的训练样本。Yang 等人进一步提出了 HCGLA 算法 [12]，该算法设计了一个可伸缩的目标函数和一种新的随机样本初始化方法来提高在高度压缩的梯度上的重构性能。Shokri 等人 [8] 将给定样本及其标签以及全局模型参数作为攻击模型的输入，使用对抗学习来训练一个二分类器以推断给定样本来自是否来自给定的客户端，从而推断出给定样本的客户端归属。

2.2 联邦学习中的隐私保护

由于联邦学习交互的模型参数依然很容易受到攻击，为了进一步保护模型参数的隐私，许多安全机制被引入到联邦学习中，比如差分隐私 [1, 11]、多方安全计算 [7] 以及同态加密 [15]

等。然而，由于多方安全计算和同态加密需要大量的计算，并且联邦学习中的客户端可能不具备强大的计算能力，因此这两类算法并不适用于联邦学习。相比之下，差分隐私只需要根据隐私保护水平来确定随机噪声的分布，并从分布中采样随机噪声添加到模型参数上即可实现隐私保护，这几乎不消耗额外的计算资源和存储资源。此外，差分隐私可以从概率论的角度提供理论的隐私保障，因此差分隐私被广泛于联邦学习中。

差分隐私机制通过生成零均值的随机噪声来扰乱输出，从而使得在相邻集合上的输出变得不可区分以保护集合中每个个体的隐私。Wu 等人 [11] 使用拉普拉斯分布来生成随机噪声以实现差分隐私，即拉普拉斯机制。DPSGD 算法 [1] 和 CDPFO 算法 [3] 使用高斯噪声来保护输出的隐私，即高斯机制。相比于拉普拉斯机制，高斯机制实现了更好的模型性能，但是其隐私保护效果更弱。Triastcyn 等人 [9] 使用贝叶斯差分隐私来追踪隐私的消耗，由于贝叶斯差分隐私是差分隐私的放松版本，因此实现了更高的模型准确率。根据差分隐私的保护对象可以将其分为两类，分别是本地差分隐私和全局差分隐私。本地差分隐私 [1, 9, 11] 由客户端本地实现，旨在扰乱本地模型参数以保护客户端每个样本的隐私，因此可以用于抵御样本重构攻击。全局差分隐私 [3] 由服务器实现，旨在扰乱全局模型参数以保护每个客户端的隐私，即整个数据集的隐私，因此可以用于抵御成员推断攻击。然而，现有的工作都只是单独实现本地差分隐私或者全局差分隐私，这并不能全面地保护交互的模型参数的隐私安全。本文提出的 NbAFL 可以同时实现本地差分隐私和全局差分隐私，因此提供了更好的隐私保护。

3 本文方法

3.1 本文方法概述

本文使用高斯机制来实现差分隐私，并使用 FedAvg 的变体 FedProx 来进行本地训练以提高联邦学习的性能。NbAFL 在模型参数传输之前生成特定高斯随机噪声以保护模型参数。具体地，如算法 1 所示，$\mathbf{w}^{(0)}$ 为初始化的全局模型，在每轮全局迭代 t 中，

1. 参与训练的客户端 i 使用 FedProx 和本地数据集 $\mathcal{D}_i$ 进行本地训练并更新本地参数 $\mathbf{w}_i^{(t)}$;
2. 参与训练的客户端 i 利用裁剪参数 C 对本地模型参数进行裁剪以确定其 L2 敏感度;
3. 参与训练的客户端 i 根据隐私预算和本地模型参数的 L2 敏感度来生成高斯随机噪声 $\mathbf{n}_i^{(t)}$ 以扰乱 $\mathbf{w}_i^{(t)}$，并将加噪后的本地模型参数 $\tilde{\mathbf{w}}_i^{(t)}$ 上传到服务器;
4. 服务器将客户端的样本数量和总样本数量之比作为聚合比例得到聚合后的全局模型 $\mathbf{w}^{(t)}$;
5. 服务器根据隐私预算和全局模型的 L2 敏感度来生成高斯随机噪声 $\mathbf{n}_D^{(t)}$ 以扰乱 $\mathbf{w}^{(t)}$，并将加噪后的全局模型参数分发给所有客户端。
6. 所有客户端在本地利用本地数据集测试最新全局模型 $\tilde{\mathbf{w}}^{(t)}$ 的损失函数值。

3.2 差分隐私和高斯机制

由于 NbAFL 框架使用高斯机制来实现差分隐私以保护客户端的数据隐私，因此，这里首先介绍差分隐私和高斯机制的定义。

Algorithm 1 Noising Before Aggregation FL

Data: T, $\mathbf{w}^{(0)}$, μ, ϵ and δ

Initialization: $t = 1$ and $\mathbf{w}_i^{(0)} = \mathbf{w}^{(0)}, \forall i$
while $t \leq T$ **do**
  Local training process:
  while $\mathcal{C}_i \in \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N\}$ **do**
    Update the local parameters $\mathbf{w}_i^{(t)}$ as $\mathbf{w}_i^{(t)} = \arg\min_{\mathbf{w}_i}(F_i(\mathbf{w}_i) + \frac{\mu}{2}\|\mathbf{w}_i - \mathbf{w}^{(t-1)}\|^2)$
    Clip the local parameters as $\mathbf{w}_i^{(t)} = \mathbf{w}_i^{(t)} / \max(1, \frac{\|\mathbf{w}_i^{(t)}\|}{C})$
    Add noise and upload parameters $\tilde{\mathbf{w}}_i^{(t)} = \mathbf{w}_i^{(t)} + \mathbf{n}_i^{(t)}$
  end while
  Model aggregating process:
  Update the global parameters $\mathbf{w}^{(t)}$ as $\mathbf{w}^{(t)} = \sum_{i=1}^{N} p_i \tilde{\mathbf{w}}_i^{(t)}$
  The server broadcasts global noised parameters $\tilde{\mathbf{w}}^{(t)} = \mathbf{w}^{(t)} + \mathbf{n}_D^{(t)}$
  Local testing process:
  while $\mathcal{C}_i \in \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N\}$ **do**
    Test the aggregating parameters $\tilde{\mathbf{w}}^{(t)}$ using local dataset
  end while
  $t \leftarrow t + 1$
end while
Result: $\tilde{\mathbf{w}}^{(T)}$

定义 1. (ϵ, δ)-差分隐私 *[2]*。输入域为 $\mathcal{X}$，输出域为 $\mathcal{R}$ 的随机算法 $\mathcal{M}: \mathcal{X} \to \mathcal{R}$ 满足 (ϵ, δ)-差分隐私当且仅当对于任意的输出 $\mathcal{S} \subseteq \mathcal{R}$ 有：

$$Pr[\mathcal{M}(\mathcal{D}_i) \in \mathcal{S}] \leq e^{\epsilon} Pr[\mathcal{M}(\mathcal{D}_i') \in \mathcal{S}] + \delta, \tag{1}$$

其中 $\mathcal{D}_i, \mathcal{D}_i' \in \mathcal{X}$ 是一对相邻数据集，即最多只相差一个样本的两个数据集。

在该定义中，ϵ 表示隐私的泄露程度，称为隐私预算，δ 表示相邻数据集上得到相同输出的概率之比大于 e^{ϵ} 的概率。当 $\delta = 0$ 时则表示为 ϵ-纯差分隐私，其隐私保护水平高于 (ϵ, δ)-差分隐私。ϵ 越小，δ 越小，则隐私保护水平越高，即泄露的隐私越少。

高斯机制是最常用的差分隐私机制之一，其定义如下：

定义 2. 高斯机制 *[2]*。假设函数 $s(\mathcal{D})$ 是以数据集 $\mathcal{D}$ 为输入的实值输出函数，则原始输出 $s(\mathcal{D})$ 的 *L2* 敏感度定义为 $\Delta s = \max_{\mathcal{D}_i, \mathcal{D}_i'} \| s(\mathcal{D}_i) - s(\mathcal{D}_i') \|$，其中 $\mathcal{D}_i, \mathcal{D}_i' \in \mathcal{X}$ 是一对相邻数据集。当扰乱 $s(\mathcal{D})$ 的高斯噪声 $n \sim \mathcal{N}(0, \sigma^2)$ 满足 $\sigma \geq c\Delta s/\epsilon$，其中 c 是满足 $c \geq \sqrt{2\ln(1.25/\delta)}$ 的常数并且 $\epsilon \in (0, 1)$，则随机算法 $\mathcal{M}(s(\mathcal{D})) = s(\mathcal{D}) + n$ 满足 (ϵ, δ)-差分隐私。

定义 2 描述了暴露一个输出 $s(\mathcal{D})$ 并实现 (ϵ, δ)-差分隐私所需添加的高斯随机噪声的分布，如果要暴露 L 个输出并实现 (ϵ, δ)-差分隐私，则每个输出所需添加的高斯随机噪声的分布应满足以下组合定理。

定理 1. 高斯机制的组合定理 *[2]*。给定总隐私水平 (ϵ, δ) 以及 L 个原始输出 $s_1(\mathcal{D}), \cdots, s_L(\mathcal{D})$，则每个输出所需添加的高斯噪声的标准差 σ 应该满足 $\sigma \geq cL\Delta s/\epsilon$。

3.3 全客户端参与 NbAFL

3.3.1 客户端的本地差分隐私

根据算法 1 可得客户端 i 在第 t 轮全局迭代的原始本地模型参数为

$$s(\mathcal{D}_i) = \mathbf{w}_i^{(t)} = \frac{1}{|\mathcal{D}_i|}\sum_{j=1}^{|\mathcal{D}_i|}\arg\min_{\mathbf{w}_{i,j}}(F_i(\mathbf{w}_{i,j}\ \mathcal{D}_{i,j}) + \frac{\mu}{2}\|\mathbf{w}_{i,j} - \mathbf{w}^{(t-1)}\|^2). \tag{2}$$

其中，$|\mathcal{D}_i|$ 是数据集 $\mathcal{D}_i$ 的样本总数，$\mathcal{D}_{i,j}$ 表示数据集 $\mathcal{D}_i$ 中的第 j 个样本。由于客户端 i 对更新后的本地模型参数 $\mathbf{w}_i^{(t)}$ 进行裁剪以确保其参数的 L2 范数 $\| \mathbf{w}_i^{(t)} \| \leq C$，其中 C 是裁剪参数。因此客户端 i 裁剪后的本地模型参数的 L2 敏感度 $s_U^{\mathcal{D}_i}$ 如下：

$$\Delta s_U^{\mathcal{D}_i} = \max_{\mathcal{D}_i, \mathcal{D}_i'} \| s(\mathcal{D}_i) - s(\mathcal{D}_i') \| = \frac{2C}{|\mathcal{D}_i|}, \tag{3}$$

则所有本地模型参数的 L2 敏感度定义为 $\Delta s_U \triangleq \max\{\Delta s_U^{\mathcal{D}_i}\}, \forall i$，假设最小样本数量为 m，则 $\Delta s_U = \frac{2C}{m}$。考虑到每个客户端只发送自身的本地模型参数，并且实际传输过程中有丢包和延时等问题，因此本文假设在 T 轮传输本地模型参数过程中，攻击者最多可以获取到 $L(L \leq T)$ 次本地模型参数，也就是说客户端最多只会暴露 L 次本地模型。根据定理 1 ，每轮迭代本地模型参数所需添加的高斯噪声的标准差为 $\sigma_U = cL\Delta s_U/\epsilon$。

3.3.2 服务器的全局差分隐私

假设数据集集合 $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_i, \cdots, \mathcal{D}_N\}$ 和 $\mathcal{D}' = \{\mathcal{D}_1, \cdots, \mathcal{D}_i', \cdots, \mathcal{D}_N\}$ 是一对相邻数据集集合，由于服务器按 $s_D^{\mathcal{D}} = \mathbf{w}^{(t)} = p_1 s(\mathcal{D}_1) + \cdots + p_i s(\mathcal{D}_i) + \cdots + p_N s(\mathcal{D}_N)$ 来聚合多个客户端的本地模型参数得到全局模型参数，因此聚合后全局模型参数的 L2 敏感度为：

$$\Delta s_D = \max_i \frac{2Cp_i}{m}. \tag{4}$$

给定样本总数，当客户端的样本数量相等，即 $p_i = \frac{1}{N}, \forall i$ 时，全局模型参数的 L2 敏感度最小，此时高斯噪声方差最小，模型性能最好。因此，本文令 $p_i = \frac{1}{N}, \forall i$ 以及 $\Delta s_D = \frac{2C}{mN}$。

考虑到每轮迭代中服务器会将全局模型参数发送给所有 N 个客户端，尽管同样存在丢包和延时等问题，但是总共传输 N 个相同的全局模型参数基本可以保证攻击者可以稳定获取每轮迭代的全局模型参数，因此 T 轮全局迭代中全局模型最多会暴露 T 次。根据定理 1 ，每轮迭代全局模型参数理论上需添加的高斯噪声的标准差为 $\sigma_A = cT\Delta s_D/\epsilon$。然而，这里的理论结果忽略了全局模型参数本身就是由加噪的本地模型参数聚合得到的，因此其聚合后的全局模型本身就有高斯噪声的影响，根据聚合规则可以得到全局模型中噪声的影响为 $\sum_{i=1}^N p_i \mathbf{n}_i = \frac{1}{N}\sum_{i=1}^N \mathbf{n}_i$。由于客户端本地添加的高斯噪声 $\mathbf{n}_i \sim \mathcal{N}(0, \sigma_U^2)$，其中 $\sigma_U = cL\Delta s_U/\epsilon$，因此全局模型本身存在的高斯噪声的标准为 $\frac{\sigma_U}{\sqrt{N}}$。如果全局模型本身存在的高斯噪声的标准差小于其理论所需添加的高斯噪声的标准差，则应该添加额外的高斯噪声来满足理论上的标准差差以实现 (ϵ, δ)-全局差分隐私，否则全局模型参数无需额外添加高斯噪声，即全局模型参数实际添加的高斯噪声的标准差为

$$\sigma_D = \sqrt{\sigma_A^2 - \frac{\sigma_U^2}{N}} = \begin{cases} \frac{2cC\sqrt{T^2 - L^2 N}}{mN\epsilon}, & T > L\sqrt{N}, \\ 0, & T \leq L\sqrt{N}. \end{cases} \tag{5}$$

3.3.3 收敛上界分析

为了分析 NbAFL 的收敛上界，根据 [5] ，本文对损失函数的性质做了以下假设：

假设 1. *对于任一客户端 i 的本地损失函数 $F_i(\cdot)$ 和全局损失函数 $F(\cdot) \triangleq \sum_{i=1}^{N} p_i F_i(\cdot)$ 有：*

1) $F_i(\mathbf{w})$ *是凸函数；*

2) $F(\mathbf{w}) - F(\mathbf{w}^*) \leq \frac{1}{2l} \parallel \nabla F(\mathbf{w}) \parallel^2$，*其中* $\mathbf{w}^*$ *是全局最优参数，并且* l *是正实数；*

3) $F(\mathbf{w}^{(0)}) - F(\mathbf{w}^*) = \Theta$*;*

4) 对于任意的 $\mathbf{w}$ *和* $\mathbf{w}'$ *有* $\parallel F_i(\mathbf{w}) - F_i(\mathbf{w}') \parallel \leq \beta \parallel \mathbf{w} - \mathbf{w}' \parallel$；

5) 对于任意的 $\mathbf{w}$ *和* $\mathbf{w}'$ *有* $\parallel \nabla F_i(\mathbf{w}) - \nabla F_i(\mathbf{w}') \parallel \leq \rho \parallel \mathbf{w} - \mathbf{w}' \parallel$。

基于上述对损失函数的假设，全客户端参与 NbAFL 的收敛结果如下：

定理 2. *(全客户端参与 NbAFL 的收敛上界)。在给定隐私预算* ϵ *下，* T *轮全局迭代后全客户端参与 NbAFL 的收敛上界为*

$$\mathbb{E}\{F(\tilde{\mathbf{w}}^{(T)}) - F(\mathbf{w}^*)\} \leq P^T \Theta + \left(\frac{\kappa_1 T}{\epsilon} + \frac{\kappa_0 T^2}{\epsilon^2}\right)(1 - P^T), \tag{6}$$

其中 $P = 1 + 2l\left(-\frac{1}{\mu} + \frac{\rho B}{\mu^2} + \frac{\rho B^2}{2\mu^2}\right)$，$\kappa_1 = \frac{\left(\frac{1}{\mu} + \frac{\rho B}{\mu}\right)\beta c C}{m(1-P)} \sqrt{\frac{2}{N\pi}}$ *以及* $\kappa_0 = \frac{\rho c^2 C^2}{2m^2(1-P)N}$。

基于上述收敛上界可以得出以下结论：

1) 隐私预算 ϵ 越大，即隐私保护水平越低，则 NbAFL 的收敛上界越小，即性能越好，这是因为差分隐私随机噪声的影响变小了；

2) 客户端总数 N 越大，收敛上界越小，这是因为客户端总数变大不仅增加了训练样本的数量，还减少了聚合后全局模型中噪声的影响；

3) 很容易证明上述收敛上界是一个关于全局迭代总轮数 T 的凸函数，因此存在最优全局迭代总轮数 T^* 使 NbAFL 性能最优。

3.4 部分客户端参与 NbAFL

3.4.1 客户端的本地差分隐私

尽管在部分客户端参与场景下每轮全局迭代只随机采样 K 个客户端参与训练，但是考虑到最坏情况，也就是某个客户端在每一轮全局迭代中都被选中，则该客户端在 T 轮全局迭代中的参与次数为 T 次。也就是说，任一客户端在 T 轮全局迭代中最多被选中 T 次，这意味着其最多只会暴露 L 次本地模型给攻击者。因此，每轮全局迭代中，被选中的客户端所需添加的高斯随机噪声的标准差为 $\sigma_U = cL\Delta s_U/\epsilon$，该结果和全客户端参与的结果是一致的。

3.4.2 服务器的全局差分隐私

由于每轮全局迭代只有 K 个客户端参与训练并上传本地模型参数到服务器参与聚合，因此 K 个客户端聚合后全局模型中由加噪本地模型参数带来的高斯随机噪声标准差为 $\frac{\sigma_U}{\sqrt{K}}$。此外，理论上实现 (ϵ,δ)-全局差分隐私所需的高斯随机噪声标准差应该调整为 $\sigma_A=\frac{c\Delta s_D T}{b\epsilon}$，其中 $b=-\frac{T}{\epsilon}\ln(1-\frac{N}{K}+\frac{N}{K}e^{\frac{-\epsilon}{T}})$。因此，根据 $\frac{\sigma_U}{\sqrt{K}}$ 和 σ_A 的大小关系来决定全局模型参数额外添加的高斯噪声标准差 σ_D 为：

$$\sigma_D=\sqrt{\sigma_A^2-\frac{\sigma_U^2}{K}}=\begin{cases}\frac{2cC\sqrt{\frac{T^2}{b^2}-L^2K}}{mK\epsilon}, & T>\frac{\epsilon}{\gamma},\\ 0, & T\le\frac{\epsilon}{\gamma}.\end{cases}\tag{7}$$

其中，$\gamma=-\ln(1-\frac{K}{N}+\frac{K}{N}e^{\frac{-\epsilon}{L\sqrt{K}}})$。

3.4.3 收敛上界分析

类似于全客户端参与，部分客户端参与 NbAFL 的收敛结果如下:

定理 3. *(部分客户端参与 NbAFL 的收敛上界)。在给定隐私预算 ϵ 下，T 轮全局迭代后部分客户端参与的 NbAFL 的收敛上界为*

$$\begin{aligned}&\mathbb{E}\{F(\tilde{\mathbf{v}}^{(T)})-F(\mathbf{w}^*)\}\\ \le&Q^T\Theta+\frac{1-Q^T}{1-Q}\Big(\frac{cC\alpha_1\beta}{-mK\ln(1-\frac{N}{K}+\frac{N}{K}e^{-\frac{\epsilon}{T}})}\sqrt{\frac{2}{\pi}}+\frac{c^2C^2\alpha_0}{m^2K^2\ln^2(1-\frac{N}{K}+\frac{N}{K}e^{-\frac{\epsilon}{T}})}\Big),\end{aligned}\tag{8}$$

其中 $Q=1+\frac{2l}{\mu^2}(\frac{\rho B^2}{2}+\rho B+\frac{\rho B^2}{K}+\frac{2\rho B^2}{\sqrt{K}}+\frac{\mu B}{\sqrt{K}}-\mu)$，$\alpha_0=\frac{2\rho K}{N}+\rho$，$\alpha_1=1+\frac{2\rho B}{\mu}+\frac{2\rho B\sqrt{K}}{\mu N}$ 并且 $\tilde{\mathbf{v}}^{(T)}=\sum_{i=1}^K p_i(\mathbf{w}_i^{(T)}+\mathbf{n}_i^{(T)})+\mathbf{n}_D^{(T)}$。

定理 3 不仅和定理 2 有相同的结论，还揭示了存在最优的客户端参与数量 K^* 使 NbAFL 算法性能最优（包括全客户端参与）。

4 复现细节

4.1 代码来源

本实验代码没有参考任何相关源代码，是本人独立复现的。本实验代码复现了全客户端和部分客户端参与 NbAFL 框架，并且实现其无噪声版本 Non-private($\epsilon=\infty$) 作为性能上界。

4.2 实验环境搭建

本实验代码使用 python 语言编写，python 版本为 3.8.16，并基于 pytorch 进行模型训练，pytorch 版本为 1.13.1。

本实验代码使用 MNIST 手写数据集进行分类任务训练，并使用 MLP 网络来训练 MNIST 数据集，该网络包含一个隐藏层，节点数量为 256。实验设置一个中央服务器和 N 个客户端来模拟联邦学习的训练过程，每个客户端随机分配相同数量的训练样本作为本地数据集，默认情况下每个客户端分配 100 个样本，并使用 FedProx 进行本地训练，学习率为 0.002。此

外，客户端利用所有未裁剪本地模型参数 L2 范数的中位数作为裁剪参数 C。其余实验默认设置如表 1 所示：

客户端总数 N	全局迭代总轮数 T	暴露次数 L	正则化参数 μ	δ	高斯参数 c
50	25	1	1	0.01	$1.25\sqrt{2\ln(1.25/\delta)}$

表 1. 实验默认参数

4.3 界面分析与使用说明

在代码文件夹中，各文件的功能如下：

- “model.py” 文件存储 MNIST 数据集的训练模型 MLP；
- “Client.py” 文件存储客户端的模型训练过程；
- “Picture_1_diff_epsilon_all_client.py”,“Picture_2_diff_epsilon_all_client_small.py”, “Picture_3_diff_epsilon_partial_client.py”, “Picture_4_diff_C_all_client.py”, “Picture_5_diff_N_all_client.py”,“Picture_6_diff_T_all_client.py” 和 “Picture_7_diff_K_partial_client.py” 文件分别用于运行图 2-8 的实验设置。
- “Plot_Picture_1_diff_epsilon_all_client.py”, “Plot_Picture_2_diff_epsilon_all_client_small.py”,“Plot_Picture_3_diff_epsilon_partial_client.py”,“Plot_Picture_4_diff_C_all_client.py”,“Plot_Picture_5_diff_N_all_client.py”,“Plot_Picture_6_diff_T_all_client.py” 和 “Plot_Picture_7_diff_K_partial_client.py” 文件分别用于生成图 2-8。

5 实验结果分析

5.1 实验指标

本实验的性能指标是全局模型在所有 N 个客户端上的本地损失函数的平均值，后续简称为全局模型损失函数值。全局模型损失函数值越小则说明全局模型的性能越好，反之则说明全局模型性能越差。

5.2 不同隐私水平 ϵ 对比

图 2-3 展示了不同隐私预算下全客户端参与 NbAFL 的性能比较，其中图 2 展示了隐私预算较大的结果，即 $\epsilon = 50, 60, 100$，而图 3 展示了隐私预算较小的结果，即 $\epsilon = 6, 8, 10$。注意，图 3 中每个客户端的样本数量为 512，并且高斯参数 $c = \sqrt{2\ln(1.25/\delta)}$。实验结果表明，随着隐私保证的放松，即隐私预算 ϵ 的增加，全局模型的损失函数值逐渐减小，因为根据定义 2 随机噪声的方差减小了，这与定理 2 中的结论 1) 相吻合。

图 4 展示了部分客户端参与 NbAFL 在隐私水平 $\epsilon = 50, 60, 100$ 下每轮迭代全局模型损失函数值，其中每轮迭代参与的客户端数量 $K = 20$，暴露次数 $L = 2$。实验结果同样表明，随着隐私预算 ϵ 的增加，全局模型损失函数值逐渐减小。

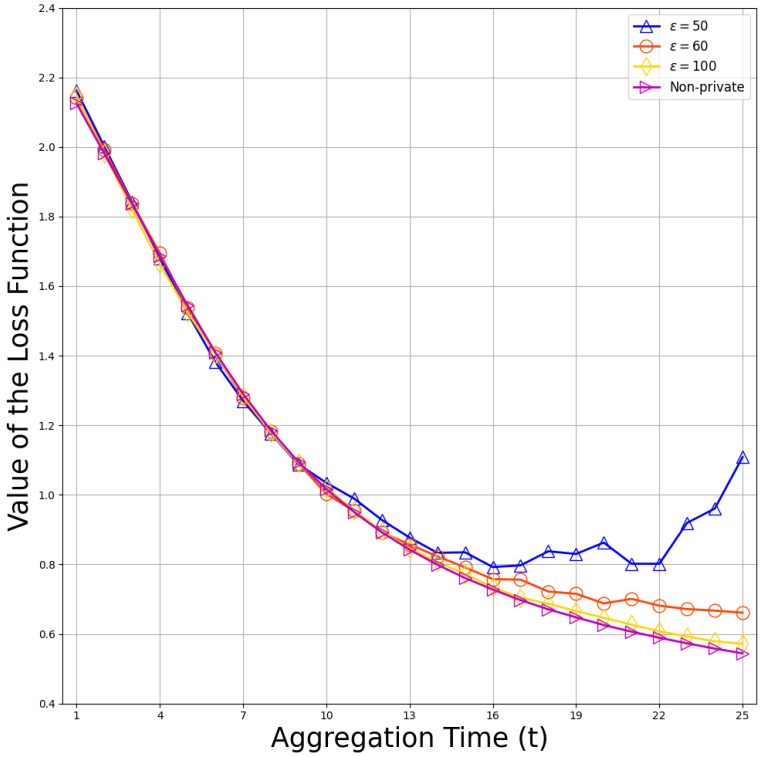

图 2. 不同隐私水平 $\epsilon = 50, 60, 100$ 下，全客户端参与 NbAFL 的全局模型损失函数值对比

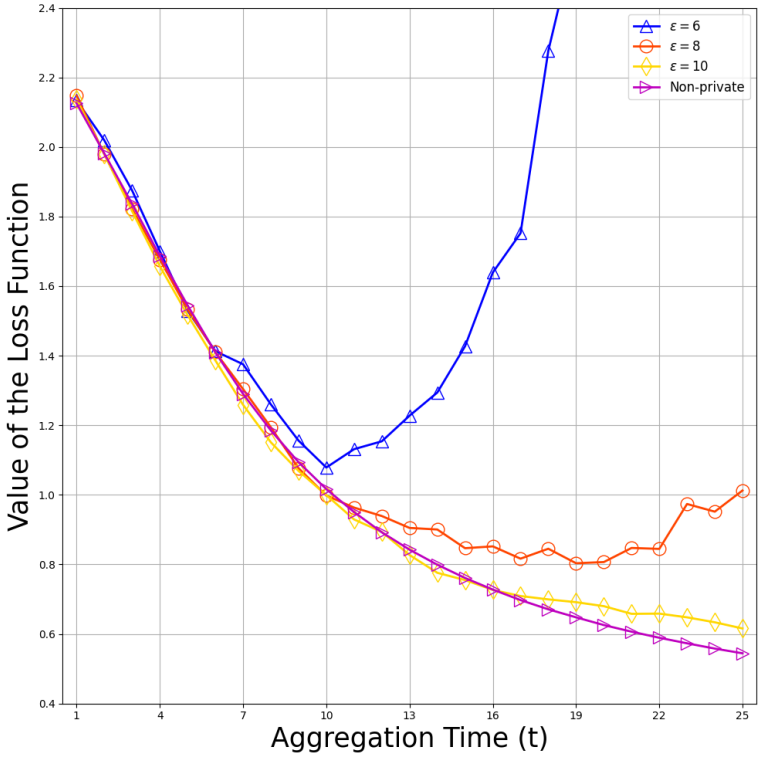

图 3. 不同隐私水平 $\epsilon = 6, 8, 10$ 下，全客户端参与 NbAFL 的全局模型损失函数值对比

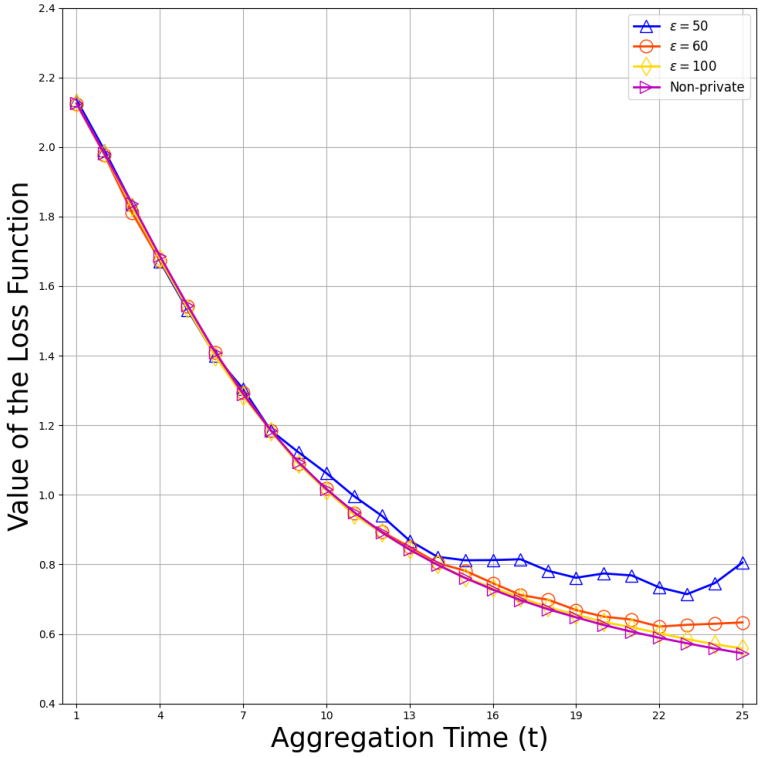

图 4. 不同隐私水平 $\epsilon = 50, 60, 100$ 下，部分客户端参与 $(N = 50, K = 20)$ NbAFL 的全局模型损失函数值对比

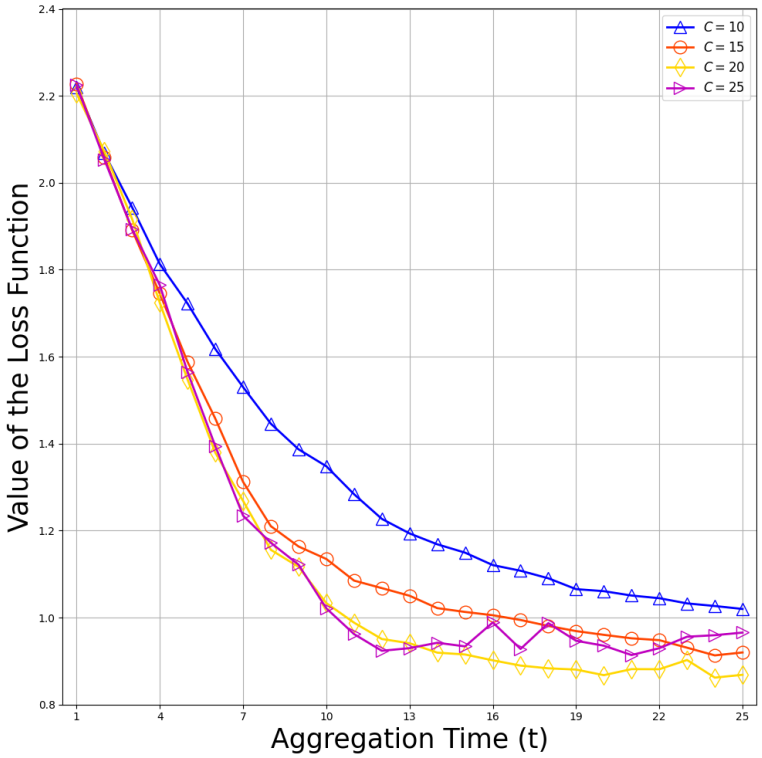

图 5. 不同裁剪参数 $C = 10, 15, 20, 25$ 下，全客户端参与 NbAFL 的全局模型损失函数值对比

5.3 不同裁剪参数 C 对比

图 5 选择了不同的裁剪参数 $C = 10, 15, 20, 25$ 来展示其对 NbAFL 性能的影响，其中客户端的隐私预算为 $\epsilon = 60$。实验结果表明，当裁剪参数 $C = 20$ 时，$T = 25$ 轮全局迭代后的全局模型损失值最小。裁剪参数对 NbAFL 性能的影响应该从两方面来分析：一方面，裁剪参数太小会导致裁剪后的本地模型参数偏离裁剪前的本地模型参数，因此降低了模型可用性，这导致 $C = 10, 15$ 的性能低于 $C = 20$ 的性能；另一方面，虽然相比 $C = 20$，$C = 25$ 裁剪后的本地模型更接近裁剪前的本地模型，然而，根据高斯机制的定义可知，裁剪参数和高斯噪声的标准差成正比，因此裁剪参数太大会导致过大的高斯噪声，从而严重扰乱本地模型参数而降低了模型性能，这就是 $C = 25$ 在训练前期 $(1 \leq T \leq 13)$ 有更小的全局模型损失函数值，而在训练后期 $(14 \leq T \leq 25)$ 反而不如 $C = 20$ 的原因。

5.4 不同客户端总数 N 对比

图 6 比较了不同客户端总数 $N = 50, 60, 80, 100$ 对全客户端参与 NbAFL 性能的影响，其中客户端的隐私预算为 $\epsilon = 60$，高斯参数 $c = 1.5\sqrt{2\ln(1.25/\delta)}$。实验结果表明，越多的客户端可以达到越小的全局模型损失函数值，这与定理 2 中的结论 2) 相吻合，有两方面的原因：一方面，由于实验设置每个客户端都有 100 个样本作为训练数据集，因此更多的客户端会提供更大的训练数据集；另一方面，客户端本地模型参数聚合得到的原始全局模型参数中高斯噪声的标准差为 $\frac{\sigma_U}{\sqrt{N}}$，因此更大的 N 会带来更小的高斯噪声。

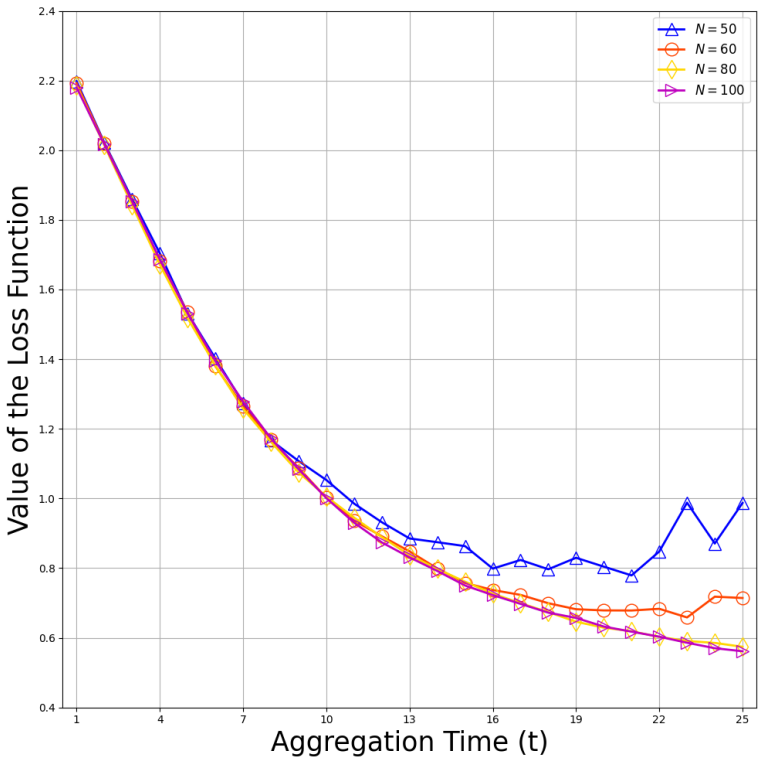

图 6. 不同客户端总数 $N = 50, 60, 80, 100$ 下，全客户端参与 NbAFL 的全局模型损失函数值对比

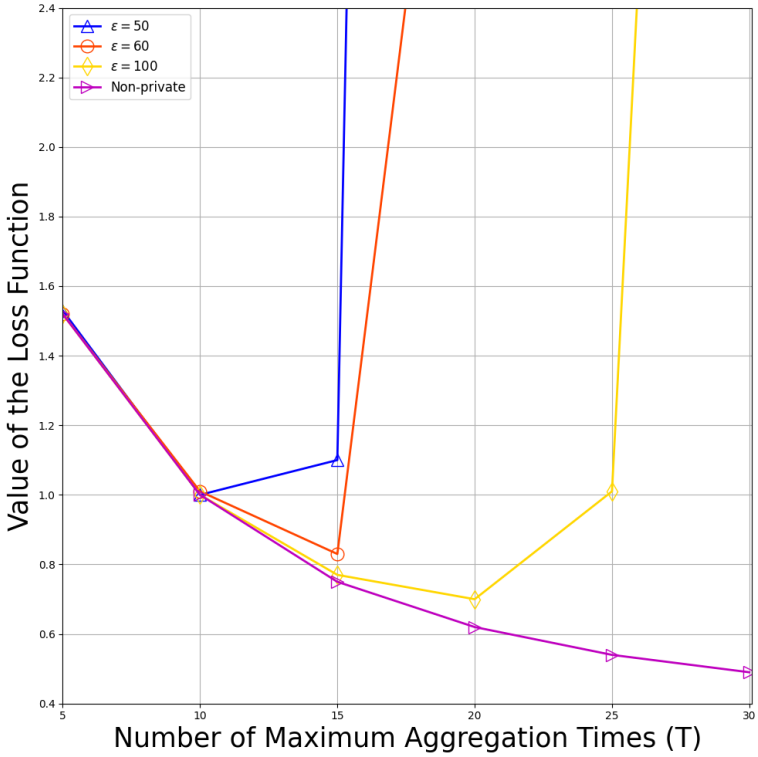

图 7. 不同全局迭代总轮数 $T = 5, 10, 15, 20, 25, 30$ 下，全客户端参与 NbAFL 的全局模型损失函数值对比

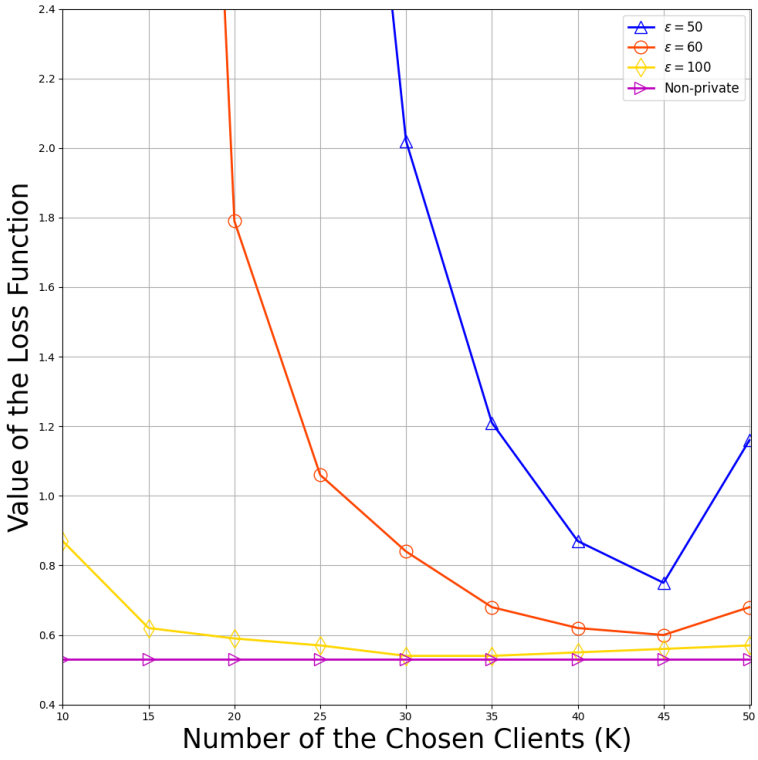

图 8. 不同客户端参与数量 $K = 10, 15, 20, 25, 30, 35, 40, 45, 50$ 下，部分客户端参与 NbAFL $(N = 50)$ 的全局模型损失函数值对比

5.5 不同全局总迭代轮数 T 对比

图 7 比较了给定隐私预算 $\epsilon = 50, 60, 100$ 下，不同全局迭代总轮数 $T = 5, 10, 15, 20, 25, 30$ 对全客户端参与 NbAFL 性能的影响。实验结果表明，不同的隐私水平下均存在最优的全局总迭代轮数 T^* 实现最小的全局模型损失函数值，这与定理 2 中的结论 3) 相吻合，并且随着 ϵ 的增加，T^* 也在增加。最优全局总迭代轮数 T^* 的影响因素有两个，一是随着 T 的增加，客户端暴露本地模型参数的次数 L 在增加，在给定总隐私预算的前提下，根据定理 1, 随着 L 的增加，每一轮所需添加的高斯噪声的标准差 $\sigma_U = \frac{cL\Delta s_U}{\epsilon}$ 就会线性增加，从而导致更大的噪声；二是根据 Non-private 的结果可知，在无噪声影响的情况下，全局总迭代轮数越多，模型的性能也越好。随着 ϵ 的增加，T^* 增加的原因是：可以认为模型参数可以容忍的最大噪声标准差相同 $\sigma_U^{max} = \frac{cL^{max}\Delta s_U}{\epsilon}$，随着 ϵ 的增加，可以容忍的最大暴露次数 L^{max} 也会增加，这就意味着可以容忍的最大全局迭代总轮数 T^* 会增加。

5.6 不同客户端参与数量 K 对比

图 8 比较了给定隐私预算 $\epsilon = 50, 60, 100$ 下，不同客户端参与数量 $K = 10, 15, 20, 25, 30, 35, 40, 45, 50$ 对部分客户端参与 NbAFL 性能的影响。实验结果表明，不同的隐私水平下均存在最优客户端参与数量 K^* 实现最小的全局模型损失函数值，这与定理 3 中的结论相匹配。存在以下两方面的权衡来决定最优的客户端参与数量 K^*：第一，随着客户端参与数量 K 的增加，每轮全局迭代的训练样本数量增加，并且聚合后原始全局模型中噪声的标准差 $\frac{\sigma_U}{\sqrt{K}}$ 减小，这有利于提高模型性能；第二，K 的增加也就意味着每轮迭代中每个客户端暴露本地模型的概率 $\frac{K}{N}$ 会增加，即客户端暴露本地模型参数的次数 L 会增加，这导致了更大的本地高斯噪声标准差 $\sigma_U = \frac{cL\Delta s_U}{\epsilon}$，即对本地模型参数带来更大的扰乱从而降低模型性能。

6 总结与展望

本文档从研究背景、研究目的、算法设计、收敛分析和实验复现五个方面对论文 “Federated Learning With Differential Privacy: Algorithms and Performance Analysis” 进行了学习和总结。原论文设计了第一个同时实现本地差分隐私和全局差分隐私的联邦学习框架 NbAFL，并使用高斯机制来确定每轮迭代本地模型参数和全局模型参数所需添加的高斯噪声分布。此外，原论文还分析了 NbAFL 的收敛上界，该收敛上界揭示了隐私保护和模型效用之间的权衡。进一步，原论文将 NbAFL 拓展到更通用的随机采样 K 个客户端参与的训练场景中。原论文提出的 NbAFL 框架及其收敛上界的理论推导是差分隐私联邦学习的初步尝试，为后续差分隐私联邦学习的发展奠定了算法基础和理论基础。

然而，NbAFL 使用最简单的组合定理 (定理 1) 来追踪隐私的消耗，这种方式导致了每轮迭代的高斯随机噪声标准差随模型总暴露次数的增加而线性增加。不幸的是，在更复杂的数据集 (比如 CIFAR10) 和模型 (比如 ResNet) 上往往需要更多的全局迭代总轮数以实现令人满意的性能，因此在这种情况下 NbAFL 会引入较大的高斯噪声影响，这导致了较差的模型性能。未来的研究方向可以使用更严格的隐私追踪技术，例如 [1] 中提出的时刻审计 (Moments Accountant)，来追踪 NbAFL 中的隐私消耗以减少高斯噪声对模型性能的影响，进一步提高 NbAFL 的实用性。

参考文献

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 308–318, New York, NY, USA, 2016. Association for Computing Machinery.

[2] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[3] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[4] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020.

[5] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[6] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[7] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 19–38. IEEE Computer Society, 2017.

[8] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[9] Aleksei Triastcyn and Boi Faltings. Federated learning with bayesian differential privacy. In Chaitanya K. Baru, Jun Huan, Latifur Khan, Xiaohua Hu, Ronay Ak, Yuanyuan Tian, Roger S. Barga, Carlo Zaniolo, Kisung Lee, and Yanfang (Fanny) Ye, editors, *2019 IEEE International Conference on Big Data (IEEE BigData), Los Angeles, CA, USA, December 9-12, 2019*, pages 2587–2596. IEEE, 2019.

[10] Aidmar Wainakh, Fabrizio Ventola, Till Müßig, Jens Keim, Carlos Garcia Cordero, Ephraim Zimmer, Tim Grube, Kristian Kersting, and Max Mühlhäuser. User-level label leakage from gradients in federated learning. *Proc. Priv. Enhancing Technol.*, 2022(2):227–244, 2022.

[11] Nan Wu, Farhad Farokhi, David Smith, and Mohamed Ali Kaafar. The value of collaboration in convex machine learning with differential privacy. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 304–317, 2020.

[12] Haomiao Yang, Mengyu Ge, Kunlan Xiang, and Jingwei Li. Using highly compressed gradients in federated learning for data reconstruction attacks. *IEEE Trans. Inf. Forensics Secur.*, 18:818–830, 2023.

[13] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[14] Honglin Yuan and Tengyu Ma. Federated accelerated stochastic gradient descent. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5332–5344. Curran Associates, Inc., 2020.

[15] Qingchen Zhang, Laurence T. Yang, and Zhikui Chen. Privacy preserving deep computation model on cloud for big data feature learning. *IEEE Trans. Computers*, 65(5):1351–1362, 2016.

[16] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.