文章编号：2095-2163(2022)05-0061-06 中图分类号：TP399 文献标志码：A

面向联邦学习的本地差分隐私设计

张 昊

(东华大学 计算机科学与技术学院，上海 201620)

摘 要：为探究联邦学习的权重聚合框架下本地差分隐私对模型精度影响并提升服务端聚合速度以及本地模型的收敛速度。本文面向联邦学习的本地差分隐私机制的实现探讨 *cells* 的个数、ε、客户端数量、扰动机制和数据分配方式对模型精度的影响。本文还设计自适应隐私预算策略，提升模型的收敛速度，方法是使用相邻轮模型的相似性与初始隐私预算建立反比关系从而自适应调整隐私预算。实验表明，从精度损失来看，随 *cells* 个数、ε、客户端数量增加而变小；在同等 ε 下，*RR*，*OUE* 的结果近似一致，*NonIID* 比 *IID* 精度损失率高；自适应隐私预算策略能够根据相邻轮模型的相似度提升模型收敛速度。

关键词：联邦学习；本地差分隐私；隐私保护；自适应隐私预算

Towards-federated-learning local differential privacy design

ZHANG Hao

(College of Computer Science and Technology, Donghua University, Shanghai 201620, China)

【Abstract】To investigate the impact of local differential privacy on model accuracy and improve the server-side aggregation speed as well as the convergence speed of the local model under the weight aggregation framework of federated learning, this paper explores the impact of the number of *cells*, ε, the number of clients, the perturbation mechanism and the data distribution method on the model accuracy for the implementation of the local differential privacy mechanism for federated learning. This paper also designs an adaptive privacy budget strategy to improve the convergence speed of the model by using the similarity of neighboring round models to establish an inverse relationship with the initial privacy budget and thus adaptively adjust the privacy budget. Experiments show that in terms of accuracy loss, it becomes smaller as the number of *cells*, ε, and the number of clients increase; the results of *RR*, *OUE* are approximately the same for equal ε, and *NonIID* has a higher accuracy loss rate than *IID*; the adaptive privacy budget strategy can improve the model convergence speed according to the similarity of the adjacent round models.

【Key words】federated learning; local differential privacy; privacy preservation; adaptive privacy budget

0 引 言

联邦学习(F-L)作为一种分布式部署的深度学习的框架，不仅让多方共同完成一个训练目标模型，而且摒弃传统的深度学习中数据集都需要统一部署在服务端的特性。每个分布式的客户端(分布式的节点)仅需各自在本地储存自己私有的数据集，并且无需上传私有数据集给服务器。因此理论上联邦学习相比传统的集中式训练的方法更有效地保证客户端的隐私。但是现在仍有可能泄露客户端私有隐私。攻击者可以通过观察客户端的上传的信息，并且通过某些手段，例如成员推理攻击[1]推断出客户端的隐私信息，从而泄露了客户端的隐私。因此违背了联邦学习满足隐私保护的特性。

在联邦学习的环境下，常用的隐私保护手段有同态加密[2-3]和本地差分隐私[4-7]。其中，同态加密由于是密码学的方法，优点是不降低数据的准确性，可以有效地还原隐私数据，缺点是同态加密的加解密和通信代价很高。本地差分隐私本质是扰动数据，因此优点是适合计算性能较差的设备，例如移动设备，缺点是会牺牲一定的效用性，因为添加的噪声是随机的，而且估计值会有一定的方差。

基于此研究时立足于联邦学习情况下，本地差分隐私更加适用于保护客户端的隐私。但是少有人研究本地差分隐私与模型精度之间的影响。因此本文考虑随机响应[8]以及优化一元编码[9]机制下，探讨 *cells* 的个数、ε、客户端数量、扰动机制和数据分配方式对模型精度的影响。本文还设计了自适应隐私预算策略，可以根据相邻轮模型相似度来提升模型的收敛速度。

1 隐私保护联邦学习实现

在权重聚合的框架中，客户端在每轮训练结束后上传的是模型权重，并且聚合的过程是 Federated Averaging[10]算法。在此基础上，本文设计权重聚合的隐私保护系统实现，具体如算法 1 所示。

作者简介：张 昊(1997-)，男，硕士研究生，主要研究方向：本地差分隐私、联邦学习。

收稿日期：2021-12-06

算法中,总共有 N 个客户端一起参与联邦学习的训练,服务器随机抽取 S_t 个客户端参与 t 轮训练。每个客户端按照是否为 IID[11] 或者 $NonIID$[11] 分配到数据集是DB_i, 训练迭代总轮数 E 和本地最小训练迭代轮数 E_L 都相同。客户端 p_i 根据私有数据集以及预先设定的本地训练次数 E_L 训练本地模型,并更新本地模型 M_i^t。 扰动模块中,客户端在本地上传扰动模型权重 PM_i^t 给服务器端。聚合模块中,服务端收集所有当前轮客户端的模型更新后,在 Federated Averaging[10] 算法基础上进行估计,同时完成模型权重的聚合平均,并作为下一轮 $t+1$ 的全局模型。重复过程 4~11,经过 E 轮迭代训练后,服务端的全局模型会最终收敛于理想状态。

算法 1 权重聚合隐私保护系统

N :联邦学习中参与客户端总数,每个客户端记作 $p_i(1 \leqslant i \leqslant N)$

$cfraction$: 每轮参与训练客户端数的比例

M_i^t : 第 t 轮 p_i 的模型更新

PM_i^t :第 t 轮扰动 p_i 的模型

DB_i: p_i 的训练数据集

B: 本地训练最小批量尺寸

E: 训练迭代总轮数

E_L: 本地总迭代轮数

η: 学习率

1: Parameter Server Executes:

2: $M_G^0 \leftarrow$ 服务器初始化全局模型

3: //客户端分配数据集

4: $DataSetBalanceAllocation(IsIID)$

5: for epoch t in $range(E)$ do

6: for client $i \in S_t$ do

7: //更新模型

8: $M_i^t = ClientUpdateModel(i, M_G^t)$

9: //上传前扰动模型权重

10: $PM_i^t = Perturb(i, M_i^t, \varepsilon^t)$

11: end for

12: //聚合估计扰动模型

13: $M_G^{t+1} = Aggeration(PM_1^t, \cdots, PM_n^t)$

14: end for

15: return M_G^{t+1}

16: //权重聚合下训练模型

17: $ClientUpdateModel(i, M_G^t)$:

18: $M_i^t \leftarrow M_G^t$

19: $B \leftarrow DB_i$ 随机分成大小为 B 的批量

20: for local epoch e in $range(E_L)$ do

21: for batch $b \in B$ do

22: //小批量梯度下降

23: $M_i^t \leftarrow M_i^t - \eta \tilde{\nabla} L(M_i^t; b)$

24: end for

25: end for

26: return M_i^t

1.1 随机响应机制实现与分析

此章节主要描述常用 2 种本地差分隐私机制随机响应[8],优化的一元编码[9] 在联邦学习的裁剪、编码、估计的具体实现,并且优化本地差分隐私在服务器端聚合速度。

1.2 扰动机制实现步骤

使用随机响应[8] (Random Response) 和优化一元编码[9] (Optimized Unary Encoding) 的基本思路是客户端 i 将第 t 轮的上传数据 D_i^t 的范围直接裁剪到 $[-1,1]$ 范围之间并且对客户端上传的数据放大 $cells = \lfloor \frac{n}{2} \rfloor$ 倍进行 one-hot 编码,再上传扰动数据 PD_i^t。 中心服务器收到第 t 轮的所有数据并聚合平均所有状态数扰动的频率 $\widetilde{Feq_p}$。 为获得聚合估计的平均模型参数,服务器还需要转化扰动状态数的频率。具体操作是估计状态数扰动的频率 $\widetilde{Feq_p}$ 获得 $\mathsf{E}[Feq_p]$, 再将估计的过状态数的频率反向映射为聚合估计的平均模型参数 M_G^{t+1}。 理论上, $cells$ 的值越大,则扰动模型的精度越接近未加噪的模型精度,模型精度损失率也越低。同时由于 one-hot 编码的关系,编码后的每个状态都是独立的。这里拟展开研究分述如下。

(1)裁剪:客户端上传的数据直接裁剪至 $[-1, 1]$ 并以 $cells = \lfloor \frac{n}{2} \rfloor$ 倍放大并映射,其中 n 理解为映射后所有状态数,例如 n 为 101,则 $cells$ 为 50,数据的范围映射范围为 $[-50,50]$, 获得离散化且放大的数据。

(2)编码:客户端 i 中第 j 个数据记作 $p_{i,j}$, 映射后所有状态数为 n,这里的数学公式可写为:

$$\boldsymbol{Encode}_{\boldsymbol{RR}}(p_{i,j}) = [0, \cdots, 1, \cdots, 0] \tag{1}$$

其中, $\boldsymbol{Encode}_{\boldsymbol{RR}}(p_{i,j})$ 是一个长度为 n 的向量,并且第 $round(\lfloor (\frac{n}{2}) \rfloor \times p_{i,j})$ 的状态为 1。

(3)扰动:对于随机响应机制,经过 RR 编码后的数据需要扰动后才能上传给中心服务器:

$$Pr[\widetilde{Encode_{RR}}(p_i)[k]=1]=\begin{cases}p=\dfrac{e^{\varepsilon}}{e^{\varepsilon}+1}\\ q=\dfrac{1}{e^{\varepsilon}+1}\end{cases} \tag{2}$$

where $\begin{cases}p=\dfrac{e^{\varepsilon}}{e^{\varepsilon}+1} & \text{if } \boldsymbol{Encode_{RR}}(p_{i,j})[k]=1\\ q=\dfrac{1}{e^{\varepsilon}+1} & \text{if } \boldsymbol{Encode_{RR}}(p_{i,j})[k]=0\end{cases}$

对于优化一元编码机制，扰动 $p_{i,j}$ 得到 $Perturb(Encode_{OUE}(p_{i,j}))$ 来保证隐私，输出结果为 $\widetilde{Encode_{OUE}}(p_{i,j})$：

$$Pr[\widetilde{Encode_{PM}}(p_{i,j})[i]=1]=\begin{cases}p=\dfrac{1}{2}\\ q=\dfrac{1}{e^{\varepsilon}+1}\end{cases} \tag{3}$$

where $\begin{cases}p=\dfrac{1}{2} & \text{if } Encode_{OUE}(p_{i,j})[i]=1\\ q=\dfrac{1}{e^{\varepsilon}+1} & \text{if } Encode_{OUE}(p_{i,j})[i]\neq 1\end{cases}$

针对参与当前轮训练客户端们对应的第 j 个模型参数，聚合平均所有状态数的扰动频率得到 $\widetilde{\boldsymbol{Feq}_{p_j}}$ 如下：

$$\widetilde{\boldsymbol{Feq}_{p_j}}=\frac{\sum_{i=1}^{N}\widetilde{Encode_{*}}(p_{i,j})}{N} \tag{4}$$

其中 $\widetilde{\boldsymbol{Feq}_{p_j}}$ 是一个 n 维的向量（$*$ 为 RR 或者 OUE），因为总共有 n 个不同频率的状态数。

(4) 聚合：随机响应算法下的 $Aggeration()$ 估计。为获得估计后所有模型参数，需要估计所有模型参数的状态数扰动频率均值 $\mathsf{E}[\boldsymbol{Feq}_p]$，再将估计后的状态数频率转化为模型参数。令训练模型所有参数扰动状态数频率记作 $\widetilde{\boldsymbol{Feq}_p}=[\widetilde{\boldsymbol{Feq}_{p_1}},\widetilde{\boldsymbol{Feq}_{p_2}},\cdots,\widetilde{\boldsymbol{Feq}_{p_N}}]$（$N$ 为训练模型中所有模型参数数量），则估计值为：

$$\mathsf{E}[\boldsymbol{Feq}_p]=\frac{[\widetilde{\boldsymbol{Feq}_{p_1}},\cdots,\widetilde{\boldsymbol{Feq}_{p_N}}]+p-1}{p-q} \tag{5}$$

其中，$\mathsf{E}[\boldsymbol{Feq}_p]$ 是一个 $N\times n$ 维的变量矩阵。

然后模型参数的估计状态数频率的矩阵需要转化为真实估计值 $\mathsf{E}[\boldsymbol{p}]$，具体如下：

$$\mathsf{E}[\boldsymbol{p}]=\mathsf{E}[\boldsymbol{Feq}_p]\times\begin{bmatrix}-\lfloor\frac{n}{2}\rfloor\\ -\lfloor\frac{n}{2}\rfloor+1\\ \vdots\\ \lfloor\frac{n}{2}\rfloor-1\\ \lfloor\frac{n}{2}\rfloor\end{bmatrix}\times\frac{1}{\lfloor\frac{n}{2}\rfloor} \tag{6}$$

其中，$\mathsf{E}[\boldsymbol{p}]$ 是一个 N 维向量，是聚合后的全局模型参数。

聚合的过程利用到矩阵的乘法，因此时间复杂度从 $O(N\times n)$ 减少至 $O(1)$，大大提升服务器端聚合时间的速度，减轻中心服务器的运算压力。而且当模型规模越大，节省的时间越多。

(5) 估计方差：为简化讨论，这里只考虑聚合时全局模型的一个模型参数的方差变化，由于每个模型参数是不相关的。所以在 Wang 等人[9]基础上计算方差 Var，对应的公式可推得为：

$$Var[\widetilde{C_{RR}(p_j)}]=N\frac{4e^{\varepsilon}}{(e^{\varepsilon}-1)^2} \tag{7}$$

显然，估计模型参数的方差是与 N 正相关，即，更多的参与者会带来更多的方差。

2　自适应隐私预算分配

本文在实验基础上观察同精度不同隐私预算下的收敛速度，提出自适应隐私预算分配的保护策略。该策略能够让联邦学习在训练过程中，保证总隐私预算不变的情况下，动态分配客户端的隐私预算。好处是能够提升模型的收敛速度。

2.1　同精度不同隐私预算的收敛速度

不同扰动机制的收敛速度比较如图 1 所示。观察图 1 就会发现，随着隐私预算增加，模型的收敛速度整体是上升的。

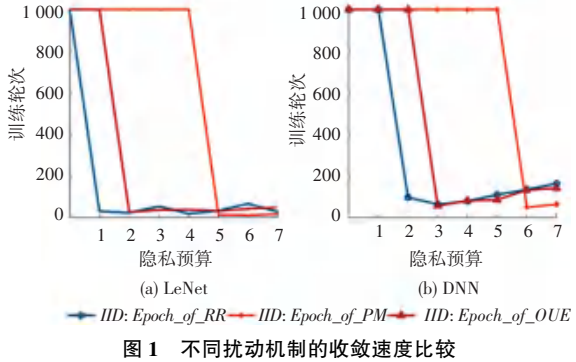

图 1　不同扰动机制的收敛速度比较

Fig. 1　Comparison of convergence rate with different perturbation mechanism

2.2 自适应隐私预算策略实现

因此本文考虑到模型最初开始训练时,由于模型没能有效地记住训练集的数据,所以模型能够泄露的训练集数据较少,导致隐私泄露率很低,因此可以用较大的隐私预算来提升模型的收敛速度。当模型逐渐接近于收敛时,模型本身记住有关训练集的信息会逐渐变多,因此需要逐渐减小隐私预算来保护模型,详见算法 2。

算法 2 自适应隐私预算策略

1: $M_G^t \leftarrow$ 第 t 轮的全局模型
2: $M_G^{t+1} \leftarrow$ 第 $t+1$ 轮的全局模型
3: M_G^t of vector $\boldsymbol{V}_G^t \leftarrow [\,]$
4: M_G^{t+1} of vector $\boldsymbol{V}_G^{t+1} \leftarrow [\,]$
5: for *layer* in M_G^t do
6: $\quad layer = flatten(layer)$
7: $\quad \boldsymbol{V}_G^t = Append(\boldsymbol{V}_G^t, layer)$
8: end for
9: for *layer* in M_G^{t+1} do
10: $\quad layer = flatten(layer)$
11: $\quad \boldsymbol{V}_G^{t+1} = Append(\boldsymbol{V}_G^{t+1}, layer)$
12: end for
13: $\varepsilon^{t+1} = similarity(\boldsymbol{V}_G^t, \boldsymbol{V}_G^{t+1})$
14: return ε^{t+1}
15:
16: $similarity(\boldsymbol{V}_G^t, \boldsymbol{V}_G^{t+1})$:
17: // cosine similarity
18: $S_C \leftarrow \dfrac{\boldsymbol{V}_G^t \cdot \boldsymbol{V}_G^{t+1}}{\|\boldsymbol{V}_G^t\| \|\boldsymbol{V}_G^{t+1}\|}$
19: //Angular distance and similarity
20: $A_s = 1 - \dfrac{\cos^{-1} S_C}{\pi}$
21: $\varepsilon = \dfrac{\varepsilon}{A_S}$
22: return ε

综合前述可知,对于相关的步骤可给出阐释论述如下。

步骤 1 当服务器接收到所有 t 轮上传的扰动数据并完成估计获得 $t+1$ 轮的全局模型 M_G^{t+1},首先逐层展平全局模型 M_G^t 和 M_G^{t+1} 并逐层添加至向量 $\boldsymbol{V}_G^t$ 和 $\boldsymbol{V}_G^{t+1}$。

步骤 2 服务器为了获得相邻 2 轮模型之间的相似度,来计算 $\boldsymbol{V}_G^t$ 和 $\boldsymbol{V}_G^{t+1}$ 之间的 cosine 相似度获得 S_C。但是 S_C 的值域处于 $[-1,1]$,无法将隐私预算与相邻模型相似度连接起来。

步骤 3 服务器将 S_C 转换成 A_S,A_S 的值域为 $[0,1]$。随后再用 $\dfrac{\varepsilon}{A_S}$ 即可获得 ε^{t+1},其中 ε^{t+1} 与模型相似度成反比关系。相邻模型相似度越小,隐私预算越大;相邻模型相似度越大,隐私预算越接近于初始隐私预算。

3 实验

3.1 实验设置

本文在 pytorch1.8 进行实验,实验硬件设置见表 1。

表 1 隐私保护联邦学习实验硬件配置

Tab. 1 Privacy-preserving F-L hardware configuration

名称	型号	备注
CPU	Intel i9-10980XE	主频 3.0 GHz×18
GPU	RTX3090×4	显存 24 G×4
内存	SK Hynix	128 GB
操作系统	Ubuntu20.04	64 位

本文实验中联邦学习的客户端数目为 1 000 个,默认参与训练比例为 0.5。*IID* 和 *NonIID* 分别训练 500,1 000 轮。实验的模型-数据集为 LeNet-MNIST 和 DNN-Purchase-100。MNIST 和 Purchase-100 分别作为常用的数字识别数据集与成员推理攻击的数据集。LeNet 和 DNN 模型的学习率在非扰动情况下,分别为 0.1 和 0.001;扰动情况下,分别为 0.1 和 0.01。LeNet 使用 SGD 优化器,DNN 使用 Adam 优化器。*IID* 下 LeNet 和 DNN 的基准模型精度分别为 0.988 5、0.909 8;非 *IID* 下 LeNet 基准模型精度为0.986 6。

隐私预算的值设为 0.075 * 2 的幂指数为单位,取值范围为 0~7(2 代表隐私预算值为 0.075 * 2^2),精度损失率的范围设为 0~1,以 0.1 间隔递增,隐私泄露率为 0.45~0.75,以 0.1 间隔递增。

定义 1 精度损失率 经过本地差分隐私保护的模型,扰动后模型收敛的精度相较于扰动前模型收敛的精度损失的百分比(%),其值可由如下数学公式计算得出:

$$\text{精度损失率} = \frac{\text{非扰动的精度} - \text{扰动后精度}}{\text{非扰动的精度}} \quad (8)$$

当精度损失率接近 0 时,代表扰动后的模型精度相较于原模型精度几乎无损失。当精度损失率接近于 1 时,代表扰动后的模型精度损失很高,即扰动后模型几乎不可用。

3.2　隐私保护联邦学习分析

3.2.1　*cells* 个数对精度损失率影响

在扰动模型权重的框架下，无论数据集是 *IID* 还是 *NonIID*，精度损失率都会随着 *cells* 数量的增加而减小。*cells* 对精度损失率影响如图 2 所示。因此为了此后的研究考虑，在模型尽可能满足收敛的情况下，这里选择最小的 *cells*（在图 2 中以红色星号表示）并继续后文的实验。

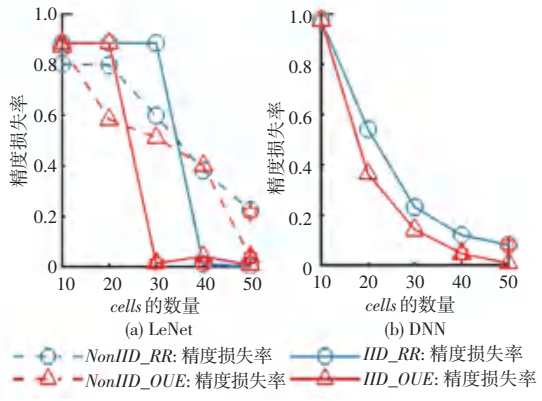

图 2　*cells* 对精度损失率影响

Fig. 2　Impact of *cells* on accuracy loss

3.2.2　隐私预算对精度损失率影响

隐私预算对精度损失率影响如图 3 所示。实验结果表明，对于模型的精度损失率来说，精度损失率会随着隐私预算的增加而减小。原因是当隐私预算变大，扰动机制所添加的噪声也会变少，扰动模型权重的变化较小，从而提升模型的精确度。当隐私预算足够大的时候，扰动模型权重的变化几乎可以忽略不计，那么模型收敛后的精度几乎等于不扰动模型收敛后的精度，即精度损失率接近于 0。当隐私预算足够小的时候，由于扰动模型权重的变化太大，会导致模型无法收敛，因此模型的精度损失率很高。

3.2.3　客户端数量对精度损失率影响

客户端数量对精度损失率影响如图 4 所示。由图 4 可知，观察到当客户端数量增加，精度损失率会逐渐下降。本文认为原因是当客户端数量比较少的时候，本地差分隐私添加噪声后，虽然估计方差小了，但是估计均值到真实均值附近的概率较小，因此获得的估计值与原先的均值的方差区别较大，导致模型不容易收敛。当客户端数量比较多的时候，纵使 *RR* 算法和 *OUE* 算法的方差变大，但是估计值到真实均值附近的概率较大，因此训练扰动模型中间值和正常训练模型中间值近似，所以模型的收敛后精度损失率较小。

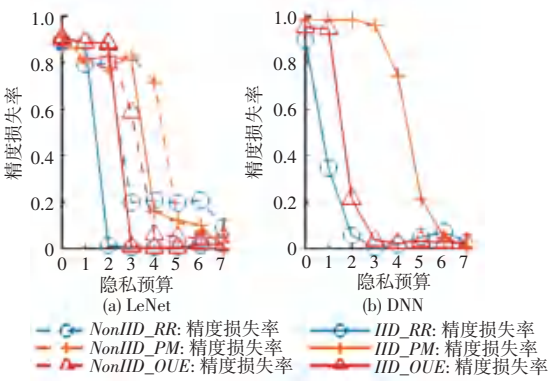

图 3　隐私预算对精度损失率影响

Fig. 3　Impact of privacy budget on accuracy loss

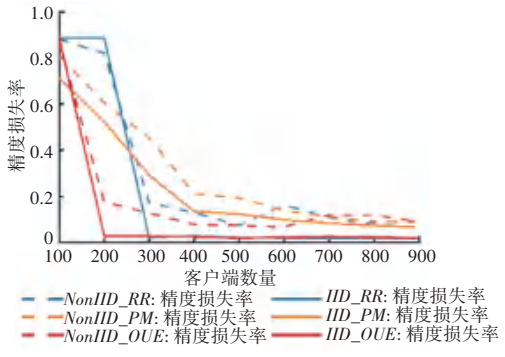

图 4　客户端数量对精度损失率影响

Fig. 4　Impact of number of clients on accuracy loss

3.2.4　自适应隐私预算策略结果

自适应隐私预算策略实现对比如图 5 所示。在图 5 中，自适应隐私预算可以提升模型的收敛速度。实验发现 LeNet 相较于 DNN 模型能够显著地提升模型的收敛速度，原因是 LeNet 模型能在初始 10 个轮次提升精度至 0.6～0.8，但是 DNN 模型却需要 100～300 轮才大致提升精度至 0.5～0.7。因此如果只看 2 个模型的相邻 2 轮的近似度，LeNet 的相邻轮变化明显，所以提升收敛速度明显。

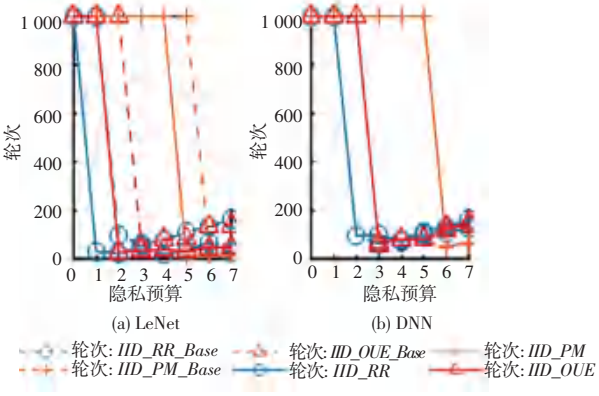

图 5　自适应隐私预算策略实现对比

Fig. 5　Comparison of adaptive privacy budget strategy implementation

（下转第 69 页）