一种面向深度神经网络的差分隐私保护算法

周治平*①② 钱新宇①

①(江南大学物联网工程学院 无锡 214122)

②(江南大学物联网技术应用教育部工程研究中心 无锡 214122)

摘 要：深度神经网络梯度下降过程中存在较大的梯度冗余，应用差分隐私机制抵御成员推理攻击时，会引入过量噪声。针对上述问题，该文利用Funk-SVD矩阵分解算法将梯度矩阵分解，分别在低维特征子空间矩阵和残差矩阵中添加噪声，利用梯度重构过程消除冗余梯度噪声。重新计算分解矩阵范数并结合平滑敏感度降低噪声规模。同时根据输入特征与输出相关性，将更多隐私预算分配给相关系数大的特征以提高训练精度。最后，根据分解矩阵范数均值提出一种自适应梯度剪裁算法以解决收敛缓慢的问题。算法利用时刻统计计算了在多种优化策略下的累计隐私损失。在标准数据集MNIST和CIFAR-10上验证了该文算法更有效地弥补了与非隐私模型之间的差距。

关键词：差分隐私；Funk-SVD；平滑敏感度；相关性；梯度剪裁

中图分类号：TN918; TP309 **文献标识码**：A **文章编号**：1009-5896(2022)05-1773-09

DOI: 10.11999/JEIT210276

Differential Privacy Algorithm under Deep Neural Networks

ZHOU Zhiping①② QIAN Xinyu①

①(*School of Internet of Things Engineering, Jiangnan University, Wuxi* 214122, *China*)

②(*Engineering Research Center of Internet of Things Technology Applications of Ministry of Education, Jiangnan University, Wuxi* 214122, *China*)

Abstract: Gradient redundancy exists in the process of deep neural network gradient descent. When differential privacy mechanism is applied to resist member inference attack, excessive noise will be introduced. So, the gradient matrix is decomposed by Funk-SVD algorithm and noise is added to the low-dimensional eigen subspace matrix and residual matrix respectively. The redundant gradient noise is eliminated in the gradient reconstruction process. The decomposition matrix norm is recalculated and the smoothing sensitivity is combined to reduce the noise scale. At the same time, according to the correlation between input features and output features, more privacy budget is allocated to features with large correlation coefficients to improve the training accuracy. The noise scale is reduced by recalculating the decomposition matrix norm and the smoothing sensitivity. Moment accountant is used to calculate the cumulative privacy loss under multiple optimization strategies. The results show that Deep neural networks under differential privacy based on Funk-SVD (FSDP) can bridge the gap with the non-privacy model more effectively on MNIST and CIFAR-10.

Key words: Differential privacy; Funk Singular Value Decomposition (Funk-SVD); Smooth sensitivity; Correlation; Gradient clipping

1 引言

随着深度神经网络在现实生活中的广泛应用，为了提供更优质的服务，越来越多的真实数据集被用来训练。目前广泛应用的指纹、人脸识别系统的原始数据集包含大量敏感信息，通过某一攻击手段能对其还原。主流攻击方式有3类[1]。(1)成员推理攻击：Nasr等人[2]提出对神经网络输出和梯度攻击，实现CIFAR数据集75.1%的攻击精度。(2)模型倒推攻击：Hitaj等人[3]利用生成对抗网络(Generative Adversarial Networks, GANs)强大的数据生成能力，在生成的数据中复现用户人脸数据。(3)模型提取攻击：Juuti等人[4]提出一种PRADA的迭代模型提取策略，通过查询模型结构来训练替代模型，将更新后的数据模型通过雅可比矩阵的数据增强合成新样本，从而造成隐私泄露。

传统匿名化方法包括K-匿名、L-多样性、T-近邻，其主要问题有：(1)无法抵御背景攻击，例

收稿日期：2021-04-06；改回日期：2021-08-16；网络出版：2021-09-24

*通信作者： 周治平 zzp@jiangnan.edu.cn

如Netflix隐私泄露事件；(2)与深度学习融合存在较大难度。该类方法修改原始数据集中的记录实现对用户敏感属性的泛化或抑制，而深度学习需要直接对数据特征进行提取继而训练，这使得二者在理论上难以结合。差分隐私[5]不惧背景攻击并具有严格理论基础，为在深度学习中实现隐私保护提供了可能性。在现有研究中，差分隐私技术应用于深度神经网络时，噪声添加位置包括输入、成员参数，如梯度和权重参数、目标函数和输出。差分隐私技术提供了可靠的隐私保护，确保对手即使拥有除目标对象外的所有信息，也无法推断出数据库中是否包含某条特定记录。在深度神经网络中应用差分隐私已经有了一些研究成果。Abadi等人[6]提出一种DP-SGD算法计算训练样本随机子集的梯度，按L2范数裁剪每个梯度，并为每个批次的累计梯度添加噪声。Xu等人[7]在生成对抗网络梯度添加全局梯度噪声，利用一部分公开数据集计算梯度均值作为最优初始梯度剪裁阈值参与深度差分隐私训练。Phan等人[8]利用非均匀高斯机制计算噪声，将异质噪声加入批次累计梯度，有效提高算法鲁棒性。这些研究可以很好抵御成员推理攻击，但是其代价就是降低数据可用性，如何在相同隐私预算下，提高训练精度是本类方法一大难点。为此，本文提出一种新的噪声添加方式以实现差分隐私，并提出多种优化方法提高模型收敛速度和分类精度。

本文的主要贡献有以下几点：

(1)提出一种基于Funk奇异值分解(Funk Singular Value Decomposition, Funk-SVD)的差分隐私深度学习方法，将梯度矩阵分解，对特征子空间矩阵和残差矩阵添加定量噪声，重建梯度矩阵继续训练，算法训练过程中利用时刻统计实时跟踪隐私损失。

(2)提出多种优化策略提高训练精度，分别对特征矩阵和残差矩阵引入平滑敏感度以降低噪声规模；利用输入特征与输出相关性重新分配隐私预算，提高训练精度；提出分别设置特征矩阵，残差矩阵和梯度矩阵剪裁阈值的自适应梯度剪裁方法，有效加快算法收敛速度，最后在真实数据集上进行全面评估，证明本方法能够在相同隐私预算下获得更好的训练精度。

2 前期准备及相关工作

差分隐私是一种已知的隐私模型，通过最小化个人记录识别的机会来呈现最大化的隐私。文献[6]将高斯机制应用于深度神经网络梯度，并将松弛差分隐私定义引入以提高训练精度，该定义如下所示：

定义1 ((ε,δ)-差分隐私) 假设随机扰动算法为$M:D^n\to R$，其所有可能的输出构成的集合为R。那么对于任意两个给定的邻近数据集D, $D'\in D^n$，以及R的任何子集S_M满足

$$\Pr[M(D)\in S_M]\le \exp(\varepsilon)\Pr[M(D')\in S_M]+\delta \quad (1)$$

则算法M满足(ε,δ)-差分隐私，其中ε为隐私预算，影响隐私保护程度的重要因素，δ为松弛因子，表示在差分隐私条件下隐私被披露的概率。

在随后的研究中，多种自适应添加噪声方式被提出，有学者对添加噪声位置进行探索，希望在相同隐私预算下提供更强隐私保障或更高数据可用性。Phan等人[9]率先提出将隐私预算分配的概念引入深度学习，根据隐私信息在训练过程出现的位置将隐私预算拆分，分别对输入特征、隐藏层和目标函数注入分配后隐私预算下的噪声以实现差分隐私机制。随后，基于隐私预算分配的思想，Gong等人[10]和Adesuyi等人[11]提出计算神经元与输出相关性，根据相关系数调整梯度神经元隐私预算以提高训练精度。Gong等人提出根据相关性大小，将特征分为强弱相关特征并分配大小不同的隐私预算，根据不同特征所得隐私预算对目标函数展开式系数加噪。Adesuyi等人选择直接计算神经元相关系数来确定不同梯度神经元所得隐私预算。在此基础上，Wu等人[12]利用指数规则提出一种自适应噪声添加规则，将预定义噪声集合注入权重参数，对其实现更细微控制以提高隐私性。Zhou等人[13]考虑到神经网络梯度下降过程的实际在一个顶部低维子空间，提出将梯度矩阵投影到辅助数据集梯度来降低环境参数的影响以减小噪声量，将辅助数据集噪声梯度重建并进行参数更新。

3 算法思路与实现

3.1 梯度子空间分解加噪

在大规模深度学习场景下，经过短时间训练，梯度空间会收敛到一个较小子空间中。子空间由黑塞矩阵的几个顶部特征张量组成，也即梯度下降实际作用于一低维空间中。传统差分隐私直接对$d\times p$维梯度矩阵摄入噪声，因此受到环境参数p影响较大，尤其是隐含层拥有上千个神经元，梯度冗余现象普遍存在，例如，在某一$d\times p$梯度矩阵中，若存在行向量为零的情况，按照差分隐私概念需要注入足量噪声，但在实际攻击中只要保证攻击者无法根据非零向量统计特性推断出输入信息就能保证隐私性，因此，梯度冗余会引发不必要噪声的加入导致训练精度降低。基于窃取成员参数的攻击方式和梯度下降实际所处的低维子空

间，可以推断出实际保护隐私的噪声只作用于梯度的某一子空间。若利用PCA降维则难以重构梯度，文献[13]提出辅助数据集概念，但在实际训练中未必能找到合适的辅助数据集来实现高质量梯度重建，这会导致重构误差的累积。基于此要求，本文提出引入Funk-SVD矩阵分解提取梯度主成分向量并实现差分隐私机制。假设私有梯度为$\boldsymbol{g}\in R^{d\times p}$，利用Funk-SVD进行矩阵特征值分解求得包含输入信息主成分的特征子空间$\boldsymbol{V}\in R^{d\times k}$和环境参数子空间矩阵$\boldsymbol{H}\in R^{k\times p}$，其中$k$为分解特征向量个数，$d$为输入特征数。根据梯度矩阵，建立最优化模型，同时为防止过拟合现象，对目标函数引入正则化方法

$$\min_{V,H}:\ S=\sum_{i=1}^{d}\sum_{j=1}^{p}(g_{ij}-V_iH_j)^2+\theta\sum_{i=1}^{d}\sum_{u=1}^{k}(V_{iu})^2+\theta\sum_{u=1}^{k}\sum_{j=1}^{p}(H_{uj})^2 \tag{2}$$

其中，θ是控制正则化的超参数，本文采用普通梯度下降求解$\boldsymbol{V}$和$\boldsymbol{H}$，由于梯度矩阵规模有限不会造成过高的算法复杂度。随后可得残差矩阵$\boldsymbol{\Phi}=\boldsymbol{g}-\boldsymbol{VH}$。与传统差分隐私对$\boldsymbol{g}$扰动不同，本文提出在$\boldsymbol{V}$和$\boldsymbol{\Phi}$上添加噪声。在$\boldsymbol{V}$上实现差分隐私能有效阻止特征隐私泄露，此外为了防止攻击者通过残差推断出某些隐私信息，对$\boldsymbol{\Phi}$同样设置噪声扰动。由于$\boldsymbol{V}$和$\boldsymbol{\Phi}$同为$\boldsymbol{g}$分解后的产物，根据差分隐私理论，不涉及隐私预算的增加，只是在同一隐私预算下改变扰动对象。基于Funk-SVD矩阵分解的差分隐私算法步骤如图1所示。

3.2 平滑敏感度

Sun等人[14]提出用考虑在教师-学生模型中采用平滑敏感度代替全局敏感度以降低噪声量。然而在迁移学习中训练精度和累计隐私预算计算过程受到自定义常数影响较大并且未能改变最大局部敏感度。本文算法对梯度矩阵分解降维，因此需要重新计算特征子空间矩阵和残差矩阵敏感度。平滑敏感度基础定义在文献[15]给出，这里不再赘述。

对于给定相邻数据集D,D'，分别采用Funk-SVD得低维特征子空间$\boldsymbol{V},\boldsymbol{V}'$和残差矩阵$\boldsymbol{\Phi},\boldsymbol{\Phi}'$，根据敏感度定义和高斯机制所需的二范数敏感度推导最大局部敏感度$S(D)$

$$S_{\boldsymbol{V}}=||\boldsymbol{V}-\boldsymbol{V}'||_2=\sum_{h=1}^{d}\sum_{j=1}^{k}||\sum_{x_i\in L}V_{ij}-\sum_{x_i'\in L}V'_{ij}||_2\le 2dk|M_{\boldsymbol{V}}| \tag{3}$$

$$S_{\boldsymbol{\Phi}}=||\boldsymbol{\Phi}-\boldsymbol{\Phi}'||_2=\sum_{h=1}^{p}\sum_{j=1}^{k}||\sum_{x_i\in L}\Phi_{ij}-\sum_{x_i'\in L}\Phi'_{ij}||_2\le 2pk|M_{\boldsymbol{\Phi}}| \tag{4}$$

其中，k为特征数，L为样本集，$|M_{\boldsymbol{V}}|,|M_{\boldsymbol{\Phi}}|$分别为子空间矩阵和残差矩阵中的最大值。由于$\boldsymbol{V}$和$\boldsymbol{\Phi}$都是由梯度矩阵$\boldsymbol{g}$分解而来的，因此，$||\boldsymbol{V}(\boldsymbol{x}_i)\boldsymbol{H}(\boldsymbol{x}_i)||_2\approx||\boldsymbol{g}(\boldsymbol{x}_i)||_2,||\boldsymbol{\Phi}(\boldsymbol{x}_i)||_2\ll||\boldsymbol{g}(\boldsymbol{x}_i)||_2$。根据式(3)和式(4)，不难得到$S_{\boldsymbol{V}}+S_{\boldsymbol{\Phi}}<S_{\boldsymbol{g}}$。所以随着矩阵的分解，矩阵元素减小，全局敏感度降低，噪声规模与敏感度成正比，因此也会相应降低。在梯度重构过程中，根据矩阵乘法原则利用环境特征矩阵中的零值点与特征矩阵冗余位置噪声梯度相乘有效去除梯度冗余以使噪声扰动更加合理。即便残差梯度中存在梯度冗余现象，但由于残差矩阵敏感度相比原梯度矩阵大大降低，对训练精度影响较小。子空间矩阵和残差矩阵的平滑敏感度为

$$S^*_{\boldsymbol{V}}(f,D)\le 2nk|M_{\boldsymbol{V}}|\mathrm{e}^{-\beta},S^*_{\boldsymbol{\Phi}}(f,D)\le 2pk|M_{\boldsymbol{\Phi}}|\mathrm{e}^{-\beta} \tag{5}$$

其中，$\beta=\varepsilon/(2\ln(1/\delta))$[15]。

3.3 相关性计算

在3.1节中，将梯度矩阵$\boldsymbol{g}$分解为$\boldsymbol{V}$和$\boldsymbol{H}$，并将噪声添加到$\boldsymbol{V}$和$\boldsymbol{\Phi}$，虽然将矩阵分解，但$\boldsymbol{V}$和$\boldsymbol{\Phi}$依

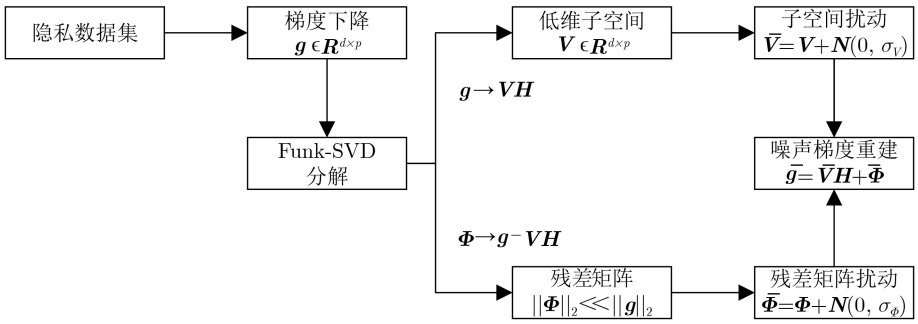


图 1 基于Funk-SVD分解的差分隐私算法

然包含d维特征向量。当特征向量采用相同隐私预算时，训练精度还是不够理想。因此，引入Layer-wise关联传播(Layer-wise Relevance Propagation, LRP)[10]对神经元影响因子进行评估，调整不同梯度神经元噪声规模。LRP相关理论基础在文献[16]中已经有了充分的介绍，这里不再赘述，计算相关性传播最后得到输入向量，隐藏层神经元和输出特征之间的相关性

$$f_{\boldsymbol{x}_i}(\omega)=\sum_{m\in h_l}R_{z\leftarrow m}^{(o,l)}(\boldsymbol{x}_i)=\cdots=\sum_{j\in[1,d]}R_j(\boldsymbol{x}_i) \tag{6}$$

其中，$R_j(\boldsymbol{x}_i)$是输入变量$\boldsymbol{x}_i$经卷积层后的特征j与模型输出层o之间的相关性。其中神经元p到神经元q的相关系数$R_{q\leftarrow p}^{(l,l+1)}$根据$\varepsilon-$rule规则[16]求解，根据式(6)可以计算不同输入特征对输出的影响因子。与输出相关性较大的特征向量分配较高的隐私预算，而相关性较小的特征向量则分配较低的隐私预算。根据输入特征对输出的相关性得分$R_j(\boldsymbol{x}_i)$得到特征j所需隐私预算大小

$$\varepsilon_j=d\times\varepsilon\times|R_j|\Big/\sum_{j=1}^{d}|R_j| \tag{7}$$

结合式(7)和3.2节中平滑敏感度可到特征j在低维子空间矩阵$\boldsymbol{V}$和残差矩阵$\boldsymbol{\Phi}$中的噪声规模

$$\left.\begin{aligned}\sigma_j^{\boldsymbol{V}}&\geq\sqrt{2\ln(1.25/\delta)}\frac{S_{\boldsymbol{V}}^{*}(f,D)}{\varepsilon_j}\\ \sigma_j^{\boldsymbol{\Phi}}&\geq\sqrt{2\ln(1.25/\delta)}\frac{S_{\boldsymbol{\Phi}}^{*}(f,D)}{\varepsilon_j}\end{aligned}\right\} \tag{8}$$

3.4 自适应梯度剪裁

在差分隐私随机梯度下降算法中，梯度剪裁对模型训练有重要影响，梯度剪裁阈值C过大或过小都会影响收敛速度。单样本最佳裁剪阈值为$|\boldsymbol{g}(\boldsymbol{x}_i)||_2$，但批样本训练最佳剪裁阈值难以确定，目前最常使用的方法是根据经验选取一个全局剪裁阈值C，但随着神经网络不断迭代，会影响收敛。为此，有学者选取少量公开数据进行训练并更新合适的C值。但实际训练过程难以采集合适的公开数据集进行预训练。

本文采用梯度矩阵降维方法加噪，实现差分隐私保护，单一选取总体梯度剪裁阈值已经不适合本文算法。针对上述问题，本文提出一种适用于分解矩阵加噪的梯度剪裁算法。为了避免超参数的使用导致模型不稳定，本文取$\boldsymbol{V}$和$\boldsymbol{\Phi}$加噪后的均值作为梯度剪裁阈值，但该取值方法的最大问题就是异常值的存在使得剪裁梯度过大而添加过量噪声。因此，本文分别计算特征矩阵$\boldsymbol{V}$、残差矩阵$\boldsymbol{\Phi}$和梯度矩阵$\boldsymbol{g}$范数均值和方差：$\mu_{\boldsymbol{V}}$, $\mu_{\boldsymbol{\Phi}}$, $\mu_{\boldsymbol{g}}$和$\sigma_{\boldsymbol{V}}$, $\sigma_{\boldsymbol{\Phi}}$, $\sigma_{\boldsymbol{g}}$对于大于$\mu+3\sigma$和小于$\mu-3\sigma$的矩阵范数做异常值剔除。最后选取非异常值的梯度范数作为该轮迭代的梯度剪裁阈值，见表1。

其中，L'为非异常样本数。值得注意的是在表1中，选择先添加噪声，再对噪声梯度进行剪裁，这与文献[6]先剪裁后加噪的方式有所不同，这是因为对噪声梯度整体剪裁的方法在实际训练中能有效提高前期收敛的速度，而模型收敛顺利也会在一定程度上提升训练效果。

3.5 算法描述

本文提出一种基于Funk-SVD的深度神经网络差分隐私算法(Deep neural networks under differential Privacy based on Funk-SVD, FSDP)，具体过程如表2所示。相比于差分隐私通用方法本文主要实现以下3点改进：第一，利用Funk-SVD算法分解梯度矩阵，扰动特征子空间和残差矩阵，去除梯度冗余造成的多余噪声，计算降维后的矩阵范数以降低全局敏感度并引入平滑敏感度进一步降低噪声规模。第二，根据输出特征和输出相关性调整隐私预算分配，将更多隐私预算分配到相关性较大的特征上。第三，提出一种与Funk-SVD算法结合的自适应梯度剪裁策略，在每轮迭代中选择最合适的梯度剪裁阈值。

下面对本文提出的基于Funk-SVD矩阵分解的噪声添加方法是否满足差分隐私给出如下证明：

表 1 自适应梯度剪裁算法

输入: 当前批次样本的梯度$\boldsymbol{S}=\{\boldsymbol{g}(\boldsymbol{x}_1),\boldsymbol{g}(\boldsymbol{x}_2),\cdots,\boldsymbol{g}(\boldsymbol{x}_L)\}$, 噪声规模$\sigma_{\boldsymbol{V}}$, $\sigma_{\boldsymbol{\Phi}}$.
输出: 自适应剪裁阈值$C_{\boldsymbol{g}}$, $C_{\boldsymbol{V}}$, $C_{\boldsymbol{\Phi}}$.

(1) $\boldsymbol{g}''(\boldsymbol{x}_i)\rightarrow\{\boldsymbol{g}(\boldsymbol{x}_i)\in\boldsymbol{S}|\mu_{\boldsymbol{g}}-3\sigma_{\boldsymbol{g}}\leq||\boldsymbol{g}(\boldsymbol{x}_i)||_2\leq\mu_{\boldsymbol{g}}+3\sigma_{\boldsymbol{g}}\}$//异常值剔除
(2) $\boldsymbol{g}''(\boldsymbol{x}_i)\rightarrow\boldsymbol{V}''(\boldsymbol{x}_i)\boldsymbol{H}''(\boldsymbol{x}_i)$, $\boldsymbol{\Phi}''(\boldsymbol{x}_i)\rightarrow\boldsymbol{g}''(\boldsymbol{x}_i)-\boldsymbol{V}''(\boldsymbol{x}_i)\boldsymbol{H}''(\boldsymbol{x}_i)$//分解梯度矩阵，计算残差矩阵
(3) $C_{\boldsymbol{V}}\rightarrow\dfrac{1}{|L'|}\sum\limits_i(||\boldsymbol{V}''(\boldsymbol{x}_i)+N(0,\sigma_{\boldsymbol{V}}^2\boldsymbol{I})||_2)$, $\overline{\boldsymbol{V}''}(\boldsymbol{x}_i)\rightarrow\boldsymbol{V}''(\boldsymbol{x}_i)+N(0,\sigma_{\boldsymbol{V}}^2\boldsymbol{I})$;
$C_{\boldsymbol{\Phi}}\rightarrow\dfrac{1}{|L'|}\sum\limits_i(||\boldsymbol{\Phi}''(\boldsymbol{x}_i)+N(0,\sigma_{\boldsymbol{\Phi}}^2\boldsymbol{I}))||_2$, $\overline{\boldsymbol{\Phi}''}(\boldsymbol{x}_i)=\boldsymbol{\Phi}''(\boldsymbol{x}_i)+N(0,\sigma_{\boldsymbol{\Phi}}^2\boldsymbol{I}))$//特征矩阵$\boldsymbol{V}$和残差矩阵$\boldsymbol{\Phi}$对应剪裁阈值
(4) $\overline{\boldsymbol{g}''}(\boldsymbol{x}_i)\rightarrow\overline{\boldsymbol{V}''}(\boldsymbol{x}_i)\boldsymbol{H}''(\boldsymbol{x}_i)+\overline{\boldsymbol{\Phi}''}(\boldsymbol{x}_i)$, $C_{\boldsymbol{g}}\rightarrow\frac{1}{|L'|}\sum\limits_i||\overline{\boldsymbol{g}''}(\boldsymbol{x}_i)||_2$//噪声梯度重建并计算梯度剪裁阈值
(5) 返回梯度剪裁阈值$C_{\boldsymbol{g}}$, $C_{\boldsymbol{V}}$, $C_{\boldsymbol{\Phi}}$

$$\frac{P\{\boldsymbol{\omega}_{t+1(L)}\}}{P\{\boldsymbol{\omega}_{t+1(L')}\}}=\frac{\prod\limits_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\prod\limits_{j=1}^{d}\exp\left(\dfrac{\varepsilon_j\dfrac{\eta}{|L|}||\sum\limits_{\boldsymbol{x}_i\in L_t}\boldsymbol{g}(\boldsymbol{x}_i)-\sum\limits_{\boldsymbol{x}_i\in L_t}(\overline{\boldsymbol{V}}(\boldsymbol{x}_i)\boldsymbol{H}(\boldsymbol{x}_i)+\overline{\boldsymbol{\Phi}}(\boldsymbol{x}_i))/\max(1,||\overline{\boldsymbol{g}_t}(\boldsymbol{x}_i)||_2/C_{\boldsymbol{g}})||_1}{\Delta\boldsymbol{\omega}_t}\right)}{\prod\limits_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\prod\limits_{j=1}^{d}\exp\left(\dfrac{\varepsilon_j\dfrac{\eta}{|L|}||\sum\limits_{\boldsymbol{x}'_i\in L_t}\boldsymbol{g}(\boldsymbol{x}'_i)-\sum\limits_{\boldsymbol{x}_i\in L_t}(\overline{\boldsymbol{V}}(\boldsymbol{x}_i)\boldsymbol{H}(\boldsymbol{x}_i)+\overline{\boldsymbol{\Phi}}(\boldsymbol{x}_i))/\max(1,||\overline{\boldsymbol{g}_t}(\boldsymbol{x}_i)||_2/C_{\boldsymbol{g}})||_1}{\Delta\boldsymbol{\omega}_t}\right)}$$

$$\leq\prod_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\prod_{j=1}^{d}\exp\left(\frac{\varepsilon_j\dfrac{\eta}{|L|}||\sum\limits_{\boldsymbol{x}_i\in L_t}\boldsymbol{g}(\boldsymbol{x}_i)-\sum\limits_{\boldsymbol{x}_i\in L_t}\boldsymbol{g}(\boldsymbol{x}'_i)||_1}{\Delta\boldsymbol{\omega}_t}\right)\leq\prod_{\omega\in\omega_t}\prod_{j=1}^{d}\exp\left(\frac{\varepsilon_j\dfrac{\eta}{|L|}}{\Delta\boldsymbol{\omega}_t}2\max_{\boldsymbol{x}_i\in L_t}||\boldsymbol{g}(\boldsymbol{x}_i)||_1\right)$$

$$\leq\prod_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\prod_{j=1}^{d}\exp\left(\frac{\dfrac{\eta}{|L|}}{\Delta\boldsymbol{\omega}_t}2d\times\varepsilon\times|R_j|/\sum_{j=1}^{d}|R_j|\right)\leq\exp\left(\varepsilon\frac{\eta}{|L|}\frac{2\sum\limits_{\omega\in\omega_t}d\times\sum\limits_{j=1}^{d}|R_j|/\sum\limits_{j=1}^{d}|R_j|}{\Delta\boldsymbol{\omega}_t}\right)\leq\exp(\varepsilon)\tag{9}$$

根据式(1)和式(9)，本文方法满足差分隐私基础定义。　　证毕

3.6 计算隐私损失

如表2算法所示，本文提出的差分隐私算法，不但基于Funk-SVD算法对矩阵实现分解加噪，而且为了提高数据可用性分别对敏感度、预算分配和梯度剪裁方法做出调整，在使用多种自适应算法情况下如何跟踪算法训练过程中的累计隐私损失是在深度学习中应用差分隐私技术的关键步骤之一。本文使用Abadi等人[6]提出的时刻统计来计算累计隐私损失，首先给出以下定义：

定义2(时刻统计)　令$M:D\to R$为一种随机扰动机制，对于给定相邻数据集D, $D'\in D^n$, M在λ时刻迭代的时刻统计为

$$\alpha_M(\lambda)=\max_{D,D'}\ln E_{O\sim M(d)}[\exp(\lambda c(M,D,D',O))]\tag{10}$$

其中，$c(M,D,D',O)=\ln\dfrac{\Pr[M(D)=O]}{\Pr[M(D')=O]}$为$M$在输出为$O\in R$处隐私损失的随机变量，$\Pr[\cdot]$由算法$M$决定。

时刻统计追踪高斯机制累计隐私损失过程中，本文对噪声添加方式、敏感度和梯度剪裁方式的设置更为先进，这与Abadi等人[6]提出的DP-SGD算法不同，无法直接使用定义2计算实际隐私损失。时刻统计量的大小由噪声规模σ，即敏感度Δf和迭

表 2　基于Funk-SVD的深度神经网络差分隐私保护算法 (FSDP)

输入: 训练数据集$D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$, 学习率$\eta$, batch样本数量$L$, 损失函数$\ell(\boldsymbol{\omega}_t,\boldsymbol{x}_i)$, batch数量$T$, 给定隐私预算$\varepsilon_0$.
输出: 参数$\boldsymbol{\omega}_t$.

(1) 计算特征j隐私预算: ε_j
(2) for $t\in[T]$:
(3) 由抽样概率L/N随机选取样本集L_t
(4) 根据表1算法计算当前批次剪裁阈值$C_{\boldsymbol{g}}$, $C_{\boldsymbol{V}}$, $C_{\boldsymbol{\Phi}}$
(5) for $i\in L_t$:
(6)　$\boldsymbol{g}_t(\boldsymbol{x}_i)\to\nabla_\omega\ell(\omega_t,\boldsymbol{x}_i)\to\boldsymbol{V}\boldsymbol{H}+\boldsymbol{\Phi}$//做Funk-SVD矩阵分解
(7)　$\sigma_j^{\boldsymbol{V}}\geq\sqrt{2\ln(1.25/\delta)}\frac{S^*_{\boldsymbol{V}}(f,D)}{\varepsilon_j}$, $\sigma_j^{\boldsymbol{\Phi}}\geq\sqrt{2\ln(1.25/\delta)}\frac{S^*_{\boldsymbol{\Phi}}(f,D)}{\varepsilon_j}$//计算噪声规模
(8)　$\boldsymbol{z}^{\boldsymbol{V}}\sim N(0,(\sigma_j^{\boldsymbol{V}}C_{\boldsymbol{V}})^2\boldsymbol{I}_{k\times k})$: $\overline{\boldsymbol{V}}\to\boldsymbol{V}+\boldsymbol{z}^{\boldsymbol{V}}$//扰动特征矩阵
(9)　$\boldsymbol{z}^{\boldsymbol{\Phi}}\sim N(0,(\sigma_j^{\boldsymbol{\Phi}}C_{\boldsymbol{\Phi}})^2\boldsymbol{I}_{p\times p})$: $\overline{\boldsymbol{\Phi}}\to\boldsymbol{\Phi}+\boldsymbol{z}^{\boldsymbol{\Phi}}$//扰动残差矩阵
(10) $\overline{\boldsymbol{g}_t}(\boldsymbol{x}_i)\to\overline{\boldsymbol{V}}(\boldsymbol{x}_i)\boldsymbol{H}(\boldsymbol{x}_i)+\overline{\boldsymbol{\Phi}}(\boldsymbol{x}_i)$//梯度重建
(11) end for
(12) $\widetilde{\boldsymbol{g}}_t(\boldsymbol{x}_i)\to\dfrac{1}{|L|}\sum\limits_{\boldsymbol{x}_i\in L_t}(\overline{\boldsymbol{g}_t}(\boldsymbol{x}_i)/\max(1,||\overline{\boldsymbol{g}_t}(\boldsymbol{x}_i)||_2/C_{\boldsymbol{g}}))$//梯度剪裁
(13) 参数更新: $\boldsymbol{\omega}_{t+1}\to\boldsymbol{\omega}_t-\eta\widetilde{\boldsymbol{g}}_t(\boldsymbol{x}_i)$
(14) 计算隐私损失: $\varepsilon>\varepsilon_0$, 则结束循环
(15) end for

代次数决定。本文算法给出一种新的加噪方法，即对低维特征矩阵$\boldsymbol{V}$和残差矩阵$\boldsymbol{\Phi}$分别扰动，重新分配特征隐私预算，将两次添加噪声和隐私预算分配过程看作独立子算法，引入时刻统计中的组合定理计算本文算法实际隐私损失。

定理1(组合定理) 若算法$M_{1:k}$由相互独立的子算法$M_1, M_2, \cdots, M_k$组成，则有：

(1)(组合性)总体时刻累计有每个子算法的时刻累计组成：$\alpha_{M_{1:k}}(\lambda) \le \sum_{i=1}^{k} \alpha_{M_i}(\lambda)$;

(2)(尾边界)对于任意的$\varepsilon > 0$，算法M满足(ε, δ)差分隐私，尾边界为：$\delta = \min_{\lambda} \exp\left(\sum_{i=1}^{k} \alpha_{M_i}(\lambda) - \lambda\varepsilon\right)$。

本文算法在每一次迭代中分别在$\boldsymbol{V}$和$\boldsymbol{\Phi}$中添加同一隐私预算下不同敏感度的噪声规模$\sigma_{\boldsymbol{V}}, \sigma_{\boldsymbol{\Phi}}$，这两次噪声规模的添加是不独立的，引入赫尔德不等式计算隐私边界

$$\alpha_M(\lambda) \le \min_{\xi_1,\xi_2\in(0,1):\xi_1+\xi_2=1} (\xi_1\alpha_V(\lambda/\xi_1) + \xi_2\alpha_\Phi(\lambda/\xi_2)) \tag{11}$$

将d维特征注入不同大小噪声看作d个互不相等的子算法，按式(11)方法需要引入d个参数，计算难度较大，因此本文利用闵可夫斯基不等式计算隐私边界：

$$\begin{aligned}
\alpha_{M_{1:d}}(\lambda) &= \max_{D,D'} \ln E_{O\sim M(D)}[\exp(\lambda c_{1:d})] \\
&= \max_{D,D'} \ln E_{O\sim M(D)}\left[\exp\left(\lambda\sum_{j=1}^{d} c_j\right)\right] \\
&= \max_{D,D'} \ln E_{O\sim M(D)}\left[\prod_{j=1}^{d} \exp(\lambda c_j)\right] \\
&\le \max_{D,D'} \ln \prod_{j=1}^{d} (E_{O\sim M(D)}[\exp(c_j\lambda/d)])^d \\
&\le \sum_{j=1}^{d} d\alpha_{M_j}(\lambda/d)
\end{aligned} \tag{12}$$

其中，$c_{1:d}$为$c(M_{1:d}, D, D', O)$，下文用c_j为$c(M_j, D, D', O_j)$，$c_{1:d} = \sum_{j=1}^{d} c_j$可用链式法则推导得，这里不再赘述，根据式(11)和式(12)得到本文算法总体时刻统计为

$$\begin{aligned}
\alpha(\lambda) \le \min_{\xi_1,\xi_2\in(0,1):\xi_1+\xi_2=1} \sum_{j=1}^{d} &(\xi_1 d\alpha_{V_j}(\lambda/d\xi_1) \\
&+ \xi_2 d\alpha_{\Phi_j}(\lambda/d\xi_2))
\end{aligned} \tag{13}$$

本文提出一种新的差分隐私算法，在神经网络训练过程中不断更新隐私损失。但如何计算$\alpha_{\boldsymbol{V}_j}$和$\alpha_{\boldsymbol{\Phi}_j}$的时刻统计是一大难点。在高斯机制中，通过以下方法计算每一步的边界值：假设$u_0(x|\sigma)$，$u_1(x|\sigma)$为两高斯概率密度函数，而$u(x|\sigma)$则为两高斯函数的混合密度，$u(x|\sigma) = (1-q)u_0(x|\sigma) + qu_1(x|\sigma)$，其中，$q = L/|D|$，$u_i(x|\sigma) = N(i,\sigma) = \frac{1}{\sqrt{2\ \ \sigma^2}} \exp\left(-\frac{(x-i)^2}{2\sigma^2}\right)$，则有

$$\alpha_{\boldsymbol{V}_j}(\lambda/d\xi_1) = \ln\max(E_1(\lambda/d\xi_1, \sigma_j^{\boldsymbol{V}}), E_2(\lambda/d\xi_1, \sigma_j^{\boldsymbol{V}})) \tag{14}$$

$$\alpha_{\boldsymbol{\Phi}_j}(\lambda/d\xi_2) = \ln\max(E_1(\lambda/d\xi_2, \sigma_j^{\boldsymbol{\Phi}}), E_2(\lambda/d\xi_2, \sigma_j^{\boldsymbol{\Phi}})) \tag{15}$$

$$E_1(\lambda/d\xi_1, \sigma_j^{\boldsymbol{V}}) = \int_{-\infty}^{+\infty} u_0(x|\sigma_j^{\boldsymbol{V}}) \cdot \left(\frac{u_0(x|\sigma_j^{\boldsymbol{V}})}{u(x|\sigma_j^{\boldsymbol{V}})}\right)^{\lambda/d\xi_1} \mathrm{d}x \tag{16}$$

$$E_2(\lambda/d\xi_2, \sigma_j^{\boldsymbol{\Phi}}) = \int_{-\infty}^{+\infty} u(x|\sigma_j^{\boldsymbol{\Phi}}) \cdot \left(\frac{u(x|\sigma_j^{\boldsymbol{\Phi}})}{u_0(x|\sigma_j^{\boldsymbol{\Phi}})}\right)^{\lambda/d\xi_2} \mathrm{d}x \tag{17}$$

计算得本文算法总体隐私统计$\alpha(\lambda)$后，根据组合定理即可得到尾部边界值。

4 实验结果及分析

4.1 实验环境及数据集介绍

本节将通过具体实验对本文提出的FSDP算法的效果进行验证和说明。实验基于python平台和Pytorch框架下实现，GPU RTX 2060Ti加速。实验采用MNIST(手写数字体)和CIFAR-10标准数据集。

对于MNIST数据集，设置两个卷积层分别为32特征和64特征并使用一个5×5，步长为1的矩阵卷积和两个1000个神经元节点隐含层的全连接层。对于CIFAR-10数据集，本文使用一个带有两个卷积层分别为128和256特征和两个1000个神经元节点隐含层的全连接层的深度神经网络进行训练。

4.2 实验结果对比与分析

本文提出一种基于Funk-SVD矩阵分解的深度差分隐私保护算法，设计了多组实验在MNIST和CIFAR-10数据集上对训练精度和隐私损失进行比对。

表3设计了一组实验，探究分解矩阵在不同特征向量下的影响，取$k = [10, 50, 200, 500, 1000]$分别进行实验。从表3不难发现，在同一隐私预算下，$k$值的选取对训练精度的影响并不十分敏感。只要$k$值的选取能使得分解矩阵包含足够的原梯度矩阵特征而又不引入过多的梯度冗余，训练就能趋于稳定。因此在下面的实验中，本文选取k=50进行实验，与其他自适应算法在同一网络环境和隐私预算下进行比对。

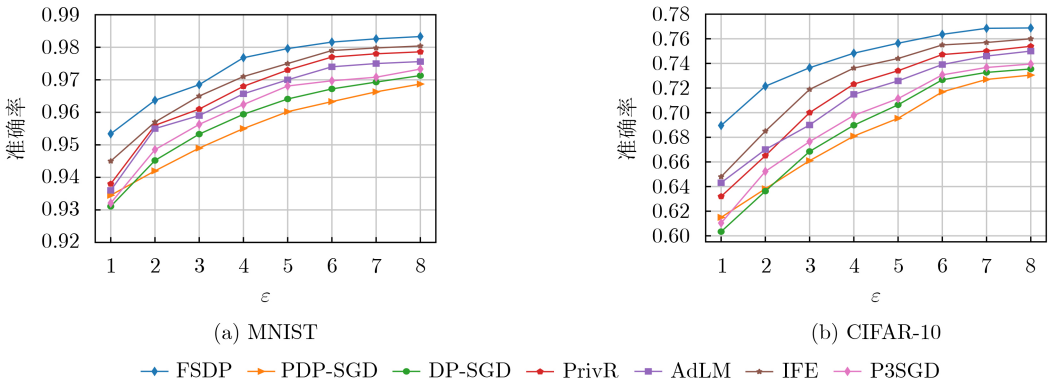
差分隐私在深度神经网络中的数据可用性一般是通过比较在相同隐私预算下的训练精度，本实验设置δ=10^{-5}，隐私预算取ε= [1, 2, 3, 4, 5, 6, 7, 8]。观察图2发现，DP-SGD算法[6]训练精度较低，这可能是因为模型噪声添加方式较为单一，考虑全局敏感度来计算梯度所需高斯噪声。PDP-SGD算法[13]首先考虑到梯度下降发生过程的内部变化，将梯度投影到辅助低维子空间后实现差分隐私机制，在高隐私要求下表现优于文献[6]，但随着噪声量降低可能会引入较多的重构误差导致精度下降。P3SGD算法[12]预先设置噪声组，采用双噪声干扰，对噪声的控制更加细致，但只利用正则化提高训练精度相比于文献[6]表现更好，但相比于其他算法略有不足。AdLM算法[9]率先提出改变噪声添加方式以及合理分配隐私预算策略，但在输入输出和仿射变换层分别添加噪声策略可能会引入过量噪声，在一定程度上影响了训练精度。PrivR[10]和IFE算法[11]在AdLM算法基础上优化，PrivR算法设置阈值，将噪声分为强弱相关噪声，并添加对应强弱噪声到相关神经元梯度，而IFE算法则直接计算神经元与输出的相关性大小，根据神经元影响因子分配隐私预算。相比基线模型，本文算法FSDP在噪声注入位置、敏感度、隐私预算分配和梯度剪裁方法均做出不同的优化，因此在训练精度方面有提升，如当ε = 2时，在MNIST和CIFAR-10标准数据集上提升了1.83%和8.52%。当然相比在无隐私条件下，在本文设置的网络环境下，MNIST和CIFAR-10数据集能达到的训练精度99.3%和83%还是有一定差距，但引入噪声后会降低数据可用性在目前的理论下是不可避免的。相比其他自适应算法，本文算法改进方法更为全面和先进，因此在相同隐私预算下，最终本文算法添加到网络中的噪声最小，训练精度有不同程度的提升，这足以证明本文算法的有效性。当然，采用更高级的网络结构也能提升训练精度，但本文算法只探究在同一网络条件和隐私性能下不同隐私算法性能。

本文提出的计算隐私损失的方法，适用于多种自适应算法，表4给出一组验证实验，计算在相同隐私预算不同算法在同一迭代次数下的实际隐私损失。为了公平起见，本文选取同样使用高斯机制的3种算法进行对比。表4中PDP-SGD算法表现出的隐私损失是略大，可能是因为算法并未考虑到残差梯度隐私泄露的问题，而P3SGD算法的隐私损失表现略低，这可能是因为该算法采用双噪声机制，在相同迭代下的隐私性更好。通过横向比较不难发现上述几种对比算法与本算法的实际隐私损失十分接近。因此，本文算法只是改进噪声添加方式和隐私预算分配方式减少了无效噪声摄入，并不会造成额外隐私损失。

为了探究本文提出了多种噪声优化算法在模型中的实际影响，表5对算法进行拆分，FSDP-M表

表 3　不同特征向量数量k的训练精度(%)

数据集	隐私预算	k				
		10	50	200	500	1000
MNIST	$(2, 10^{-5})$	96.12	96.37	96.41	96.39	96.07
	$(4, 10^{-5})$	97.33	97.68	97.72	97.69	97.35
	$(8, 10^{-5})$	98.16	98.33	98.35	98.38	97.89
CIFAR-10	$(2, 10^{-5})$	71.59	72.15	72.16	72.32	71.26
	$(4, 10^{-5})$	73.95	74.83	74.87	74.93	74.05
	$(8, 10^{-5})$	75.77	76.88	76.89	76.96	76.14



(a) MNIST　　(b) CIFAR-10

图 2 不同差分隐私条件下算法训练精度对比

表 4 隐私损失对比

数据集	ε	Epochs	DP-SGD	PDP-SGD	P3SGD	本文
MNIST	4.4	25	1.93	2.11	1.88	1.96
		50	2.71	2.87	2.65	2.82
	1.3	25	0.76	0.79	0.71	0.79
		50	1.09	1.11	1.04	1.11
CIFAR-10	4.4	25	0.72	0.89	0.63	0.79
		50	0.81	0.93	0.76	0.88
	1.3	25	0.25	0.28	0.21	0.28
		50	0.31	0.39	0.30	0.35

表 5 算法中各结构对训练精度(%)的影响

模型	MNIST			CIFAR-10		
	$(2, 10^{-5})$	$(4, 10^{-5})$	$(8, 10^{-5})$	$(2, 10^{-5})$	$(4, 10^{-5})$	$(8, 10^{-5})$
DP-SGD	94.52	95.94	97.13	63.63	68.98	73.55
FSDP-M	95.37	96.64	97.51	67.24	71.32	74.68
FSDP-S	95.63	96.98	97.85	68.74	72.49	75.54
FSDP-R	96.18	97.51	98.21	71.29	74.25	76.54
FSDP-C	95.45	96.81	97.73	67.89	71.91	75.21
FSDP	96.37	97.68	98.33	72.15	74.83	76.88

示，对梯度矩阵进行Funk-SVD分解，采用固定噪声扰动低维子空间矩阵和残差矩阵；FSDP-S代表用平滑敏感度取代全局敏感度扰动低维子空间矩阵和残差矩阵；FSDP-R代表在FSDP-S的基础上根据神经元相关性分配特征的隐私预算；FSDP-C表示在FSDP-M基础上采用文中提出的梯度自适应剪裁策略对噪声梯度进行剪裁，FSDP则是本文提出的差分隐私算法。通过与基线比较不难发现，FSDP-M对提高训练精度有显著作用，这主要是因为，算法避开了环境参数的影响，无须在冗余梯度上注入噪声，而直接在特征子空间中进行扰动，能有效降低非必要噪声的加入，此外，由于对梯度矩阵进行分解，降低了梯度矩阵范数，降低了全局敏感度，从而有效降低噪声规模，对提高训练精度有重要作用。FSDP-R进一步降低了敏感度，这对提高数据可用性有一定帮助。FSDP-R根据输入输出相关性分配隐私预算，对于输出相关性较大的特征分配较少的噪声，反之亦然，该方法对提高训练精度也有较为显著的影响。相比之下，自适应梯度剪裁策略对提升训练精度影响最小，该方法的主要贡献在于解决前期收敛缓慢的问题，使得训练稳定，这对提高训练精度也有较为积极的影响。

5 结束语

本文针对现有差分隐私直接添加全局噪声会在梯度中引入过多非必要噪声，降低数据可用性的情况提出一种基于Funk-SVD矩阵分解差分隐私算法。利用Funk-SVD将梯度矩阵分解为输入特征矩阵和环境特征矩阵，并对包含输入特征的低维子空间矩阵扰动，利用矩阵乘法和环境特征矩阵消除冗余梯度噪声，并对残差梯度扰动以保护残余隐私信息。通过计算分解矩阵范数和平滑敏感度有效降低噪声规模。利用输入特征与输出之间的相关系数合理分配隐私预算，减少大信息量特征梯度噪声。最后，根据分解矩阵特点和非异常梯度矩阵范数均值，提出一种自适应梯度阈值剪裁方法以提高收敛速度和稳定性。重新推导多种优化策略下的时刻统计计算累计隐私损失。在MNIST和CIFAR-10标准数据集上进行实验，验证了本文算法能在相同隐私预算下，有效提升分类精度，分类精度越高，包含信息量越高，数据可用性得到提升，尤其在隐私预算$(2, 10^{-5})$下，在两个数据集相比DP-SGD模型训练精度分别提升1.83%和8.52%。

参 考 文 献

[1] 刘睿瑄, 陈红, 郭若杨, 等. 机器学习中的隐私攻击与防御[J]. 软件学报, 2020, 31(3): 866–892. doi: 10.13328/j.cnki.jos.005904.

LIU Ruixuan, CHEN Hong, GUO Ruoyang, *et al*. Survey on privacy attacks and defenses in machine learning[J]. *Journal of Software*, 2020, 31(3): 866–892. doi: 10.13328/j.cnki.jos.005904.

[2] NASR M, SHOKRI R, and HOUMANSADR A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning[C]. 2019 IEEE Symposium on Security and Privacy, San Francisco, USA, 2019: 739–753. doi: 10.1109/SP.2019.00065.

[3] HITAJ B, ATENIESE G, and PEREZ-CRUZ F. Deep models under the GAN: Information leakage from collaborative deep learning[C]. The 2017 ACM SIGSAC Conference on Computer and Communications Security, New York, USA, 2017: 603–618.

[4] JUUTI M, SZYLLER S, MARCHAL S, *et al*. PRADA: Protecting against DNN model stealing attacks[C]. 2019 IEEE European Symposium on Security and Privacy (EuroS&P), Stockholm, Sweden, 2019: 512–527.

[5] 冯登国, 张敏, 叶宇桐. 基于差分隐私模型的位置轨迹发布技术研究[J]. 电子与信息学报, 2020, 42(1): 74–88. doi: 10.11999/JEIT190632.

FENG Dengguo, ZHANG Min, and YE Yutong. Research on differentially private trajectory data publishing[J]. *Journal of Electronics & Information Technology*, 2020, 42(1): 74–88. doi: 10.11999/JEIT190632.

[6] ABADI M, CHU A, GOODFELLOW I, *et al*. Deep learning with differential privacy[C]. The 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, The Republic of Austria, 2016: 308–318.

[7] XU Chugui, REN Ju, ZHANG Deyu, *et al*. GANobfuscator: Mitigating information leakage under GAN via differential privacy[J]. *IEEE Transactions on Information Forensics and Security*, 2019, 14(9): 2358–2371. doi: 10.1109/TIFS.2019.2897874.

[8] PHAN N, VU M N, LIU Yang, *et al*. Heterogeneous Gaussian mechanism: Preserving differential privacy in deep learning with provable robustness[C]. The Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 2019: 4753–4759.

[9] PHAN N, WU Xintao, HU Han, *et al*. Adaptive Laplace mechanism: Differential privacy preservation in deep learning[C]. 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, USA, 2017: 385–394.

[10] GONG Maoguo, PAN Ke, and XIE Yu. Differential privacy preservation in regression analysis based on relevance[J]. *Knowledge-Based Systems*, 2019, 173: 140–149. doi: 10.1016/j.knosys.2019.02.028.

[11] ADESUYI T A and KIM B M. Preserving privacy in convolutional neural network: An ∈-tuple differential privacy approach[C]. 2019 IEEE 2nd International Conference on Knowledge Innovation and Invention (ICKII), Seoul, South Korea, 2019: 570–573.

[12] WU Bingzhe, ZHAO Shiwan, SUN Guangyu, *et al*. P3SGD: Patient privacy preserving SGD for regularizing deep CNNs in pathological image classification[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, USA, 2019: 2094–2103.

[13] ZHOU Yingxue, WU Zhiwei, and BANERJEE A. Bypassing the ambient dimension: Private SGD with gradient subspace identification[EB/OL]. https://arxiv.org/abs/2007.03813,2020.

[14] SUN Lichao, ZHOU Yingbo, YU P S, *et al*. Differentially private deep learning with smooth sensitivity[EB/OL]. https://arxiv.org/abs/2003.00505, 2020.

[15] THAKURTA A. Beyond worst case sensitivity in private data analysis[M]. KAO M Y. Encyclopedia of Algorithms. Boston: Springer, 2016: 192–199.

[16] XU Jincheng and DU Qingfeng. Adversarial attacks on text classification models using layer-wise relevance propagation[J]. *International Journal of Intelligent Systems*, 2020, 35(9): 1397–1415. doi: 10.1002/int.22260.

周治平：男，1962年生，博士，教授，研究方向为检测技术与自动化装置、信息安全等.

钱新宇：男，1995年生，硕士生，研究方向为信息安全.

责任编辑：马秀强