摘要

基于神经网络的机器学习技术在各种领域中取得了显着的成果。 通常，模型的培训需要大型的，有代表性的数据集，这些数据集可能是众包的并且包含敏感信息。 模型不应暴露在这些数据集中的私人信息。为实现这一目标，我们开发了新的算法技术，以便在差异隐私框架内对隐私成本进行精细分析。 我们的实施和实验表明，我们可以在适度的隐私预算下培训具有非凸目标的深度神经网络，并且在软件复杂性，培训效率和模型质量方面具有可控制的成本。

1.引言

神经网络的最新进展已经在广泛的应用中取得了令人瞩目的成功，包括图像分类，语言表示，围棋(?)的移动选择等等（例如，[55,30,57,40,15]）。这些进步部分原因是可获得用于训练神经网络的大型且有代表性的数据集。这些数据集通常是众包的，可能包含敏感信息。它们的使用需要符合的技术应用程序的需求，同时提供原则和严格的隐私保证。

在本文中，我们将最先进的机器学习方法与先进的隐私保护机制相结合，在适度（“单个数字”）隐私预算内训练神经网络。我们处理具有非凸目标，多个层以及数万到数百万个参数的模型。 （相比之下，以前的工作在参数数量较少的凸模型上获得了很好的结果，或者处理复杂的神经网络但隐私损失很大。）为此，我们开发新的算法技术，在差异隐私框架内对隐私成本进行精细分析，并制定谨慎的实施策略：

1. 我们证明，通过跟踪隐私损失的详细信息（更高的时刻 ？），我们可以渐进地和经验地获得对整体隐私损失的更严格的估计。
2. 我们通过引入新技术来提高差异隐私训练的计算效率。这些技术包括用于计算各个训练样例的梯度的有效算法，将任务细分为较小批次以减少内存占用，以及在输入层应用差异隐私主投影。
3. 我们建立在机器学习框架TensorFlow[3]上，用于训练具有差异隐私的模型。我们在两个标准图像分类任务MNIST和CIFAR-10上评估我们的方法。我们选择这两项任务是因为它们基于公共数据集，并且在机器学习中作为基准有很长的历史。我们的经验表明，深度神经网络的隐私保护可以在软件复杂性，培训效率和模型质量方面以适度的成本实现。

机器学习系统通常包括有助于保护其训练数据的元素。 特别地，旨在避免过度拟合用于训练的示例的正则化技术可能隐藏那些示例的细节。 另一方面，解释深度神经网络中的内部表示是众所周知的困难，并且它们的大容量需要这些表示可能潜在地编码至少一些训练数据的精细细节（？）。 在某些情况下，确定的对手可能能够提取部分训练数据。 例如，Fredrikson等人演示了一种模型反转攻击，可以从面部识别系统中恢复图像[26]。

虽然模型反转攻击只需要“黑盒”访问训练模型（即通过输入和输出与模型交互），但我们认为对手具有额外的功能，就像Shokri和Shmatikov [52]。我们的方法提供保护对抗强大的充分了解培训机制和访问模型的参数的对手。这种保护特别适用于移动电话，平板电脑和其他设备上的机器学习应用。在设备上存储模型可以实现节能，低延迟的推理，并且可以有助于隐私，因为推理不需要将用户数据传送到中央服务器;另一方面，我们必须假设模型参数本身可能会受到敌意检查。此外，当我们关注保护训练数据中一条记录的隐私时，我们允许对手控制一些甚至所有其余训练数据的可能性。在实践中，不能总是排除这种可能性，例如当数据是众包时。

下一节将回顾深度学习和差异隐私的背景知识。第3节和第4节解释了我们的方法和实施。第5节描述了我们的实验结果。第6节讨论相关工作，第7节总结。延期证明出现在论文的完整版本中[4]。

2.背景

在本节中，我们简要回顾一下**差分隐私的定义**，介绍**高斯机制**和**组成定理**，以及**深度学习的基本原理**概述。

2.1差异隐私

差异隐私[21,18,22]构成了聚合数据库上算法的隐私保证的强标准。 它是根据相邻数据库的特定于应用程序的概念定义的。 例如，在我们的实验中，每个训练数据集都是一组图像标签对; 我们说如果它们在单个条目中不同，则这两个集合是相邻的，也就是说，如果一个图像标签对存在于一个集合中而在另一个集合中不存在。

*定义1*.随机机制M：D→R，域D和范围R满足（ε，δ） - 差分隐私，如果对于任何两个相邻输入d，d’∈D和任何子集的输出S⊆R满足以下公式：



ε-差别隐私的原始定义不包括附加项δ。 我们使用Dwork等人介绍的变体 [19]，其允许以概率δ（优选地小于1 / | d |）破坏普通ε-差别隐私的可能性。

差异隐私有几个属性，使其在我们的应用程序中特别有用：可组合性，组隐私和辅助信息的稳健性。可组合性实现了机制的模块化设计：如果机制的所有组件都是差异私有的，那么它们的组成也是如此。 如果数据集包含相关输入（例如由同一个人提供的输入），则组隐私意味着隐私保证的优雅降级。对辅助信息的稳健性意味着隐私保证不受对手可用的任何辅助信息的影响。

用于近似具有差分私有机制的确定性实值函数f：D→R的常见范例是通过校准到f的灵敏度的加性噪声，被定义为绝对距离的最大值 其中d和d’是相邻的输入。（对实值函数的限制旨在简化此评论，但不是必需的。）例如，高斯噪声机制由以下定义：



其中是正态（高斯）分布，平均值为0，标准差为。 如果和 [22，定理3.22]，单次应用高斯机制对灵敏度的函数f满足-差分隐私。 注意，该机制的这种分析可以在事后应用，并且特别地，存在满足该条件的无限多对。

加性噪声机制的重复应用的差分隐私来自基本组成定理[19,20]，或来自高级组合定理及其改进[24,34,23,11]。 在执行复合机制的过程中跟踪累积的隐私损失以及实施适用的隐私政策的任务可以由McSherry [42]引入的*隐私会计师*执行。

设计实现给定功能的差分私有加性噪声机制的基本蓝图包括以下步骤：通过有界敏感函数的顺序组合来近似功能; 选择加性噪声参数; 并对所得机制进行隐私分析。 我们在第3节中遵循这种方法。

2.2深度学习

深度神经网络对于许多机器学习任务非常有效，它将输入到输出的参数化函数定义为多层基本构建块的组合，例如仿射变换和简单的非线性函数。 后者的常用示例是sigmoids和整流线性单元（ReLU）。 通过改变这些块的参数，我们可以“训练”这样的参数化函数，目的是拟合任何给定的有限输入/输出示例集。

更准确地说，我们定义了一个损失函数L，它代表了训练数据错误匹配的代价。参数θ上的损失是训练样本{x 1，...，x N}上的损失的平均值，因此。 训练在于找到产生可接受的小损失的θ，希望是最小的损失（尽管在实践中我们很少期望达到确切的全局最小值）。

对于复杂网络，损失函数L通常是非凸的并且难以最小化。 在实践中，最小化通常通过小批量随机梯度下降（SGD）算法来完成。 在该算法中，在每个步骤中，形成随机示例的批次B并且计算作为对梯度的估计。 然后沿梯度方向朝向局部最小值更新θ。

已经构建了几个系统来支持神经网络的定义，实现有效的训练，然后执行有效的推理（执行固定参数）[31,13,3]。 我们的工作基于TensorFlow，这是由Google发布的开源数据流引擎[3]。 TensorFlow允许程序员从基本运算符定义大型计算图，并在异构分布式系统中分配它们的执行。 TensorFlow自动创建梯度的计算图形; 它还使批量计算变得容易。

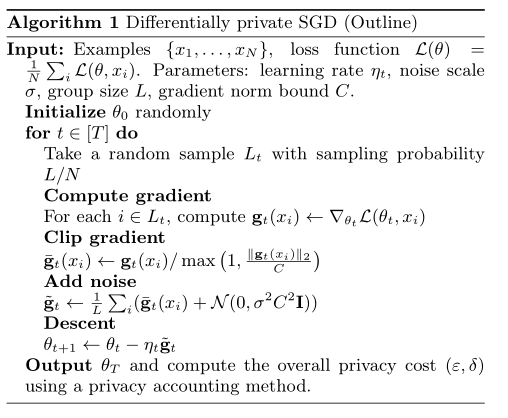
3.我们的方法

本节描述了我们的神经网络差分隐私训练方法的主要组成部分：差分隐私随机梯度下降（SGD）算法，时刻会计和超参数调整。

3.1差分隐私SGD算法

有人可能会尝试通过仅处理训练过程产生的最终参数来保护培训数据的隐私，将此过程视为黑盒子。不幸的是，一般来说，人们可能没有对这些参数对训练数据的依赖性进行有用，严格的描述;在参数中加入过于保守的噪声，其中噪声根据最坏情况分析选择的，会破坏学习模型的效用。因此，我们更倾向于采用更复杂的方法，其中我们的目标是在训练过程中控制训练数据的影响，特别是在SGD计算中。在以前的工作中遵循了这种方法（例如，[53,8]）;我们进行了一些修改和扩展，特别是在我们的隐私会计中。

算法1概述了通过最小化经验损失函数来训练具有参数θ的模型的基本方法。在SGD的每一步，我们计算一个随机子集的梯度​​，剪切每个梯度的l2范数，计算平均值，添加噪声以保护隐私，并在这个平均噪声梯度的相反方向迈出一步。最后，除了输出模型之外，我们还需要根据隐私会计师维护的信息计算机制的隐私损失。接下来，我们将更详细地描述此算法的每个组件以及我们的改进。



**规范裁剪**：证明算法1的差分隐私保证需要限制每个单独示例对 的影响。由于梯度的大小没有先验界限，我们将每个梯度剪切为l2范数;即，对于剪切阈值C，梯度向量g由替换。此剪切确保如果≤ C，然后g被保留，而如果> C，它被缩小到标准C.我们注意到这种形式的渐变剪辑是用于非隐私原因的深度网络的SGD的流行成分，尽管如此设置它通常足以在平均后剪辑。（梯度有上限C）

**每层和时间相关参数**：算法1的伪代码将所有参数组织成为损失函数L(·)的单个输入θ。对于多层神经网络，我们分别考虑每个层，这允许为不同的层设置不同的限幅阈值C和噪声标度σ。另外，clipping和noise parameters 可以随训练步骤数t而变化。在第5节中给出的结果中，我们使用C和σ的常数设置。

**Lots**：与普通的SGD算法一样，算法1通过计算一组示例中的损失梯度并取平均值来估计L的梯度。该平均值提供无偏估计，其方差随着组的大小而迅速减小。我们将这样一个组称为*lot*，以区别于通常称为batch的计算分组。为了限制内存消耗，我们可以将batch大小设置为远小于lot大小L，这是算法的一个参数。我们分批执行计算，然后将几个批次分组到一个lot中以增加噪声。实际上，为了提高效率，batches和lots的构造是通过随机置换实例然后将它们分成适当大小的组来完成的。然而，为了便于分析，我们假设通过以概率q = L / N独立地挑选每个示例来形成每个批次，其中N是输入数据集的大小。

如文献中常见的那样，我们通过将训练算法的运行时间表示为时期数来规范化训练算法的运行时间，其中每个时期是处理N个示例所需的（预期）批次数。在我们的符号中，一个时代由N / L批次组成。

**隐私会计**：对于不同的私人SGD，一个重要的问题是计算培训的整体隐私成本。 差异隐私的可组合性允许我们实施“会计”程序，该程序计算每次访问训练数据时的隐私成本，并随着训练的进展累积该成本。 每个培训步骤通常需要多层的梯度，会计师累积与所有这些相对应的成本。

**时刻会计师**：许多研究致力于研究特定噪声分布的隐私损失以及隐私损失的构成。对于我们使用的高斯噪声，如果我们选择算法1中的σ为，那么通过标准参数[22]，每个步骤对应的**lot是-差分隐私**。由于lot本身是来自数据库的随机样本，隐私放大定理[35,9]暗示**每个步骤相对于完整数据库是-差分隐私**的，其中是每一批采样率。文献中产生最佳整体界限的结果是强组合定理[24]。

然而，强组合定理可能是松散的，并且没有考虑所考虑的特定噪声分布。在我们的工作中，我们发明了一种更强大的会计方法，我们称之为会计时刻。它允许我们证明算法1对于噪声标度和限幅阈值的适当选择设置而言是-差分隐私的。与通过强组合定理得到的结果相比，我们的界限在两个方面更紧密：它在ε部分中保存了因子，在δ部分中保存了Tq因子。既然我们期望δ很小而T>>1/q（即，每个例子被多次检查），由我们的界限提供的保存是非常重要的。这个结果是我们的主要贡献之一。

**定理1**.存在常数c1和c2，因此给定采样概率q=L/N和步数T，对于任何，算法1对于任何是-差分隐私的，如果我们选择。

如果我们使用强组合定理，那么我们将需要选择。 请注意，我们在我们的渐近界保存了一个因子。 时刻会计在理论上是有益的，因为这个结果表明，并且在实践中，如第4节中的图2所示。例如，L = 0.01N，σ= 4，δ=，和T = 10000，使用会计时刻我们有ε≈1.26。 作为比较，我们将使用强组合定理得到更大的ε≈9.34。

3.2分钟会计师：详情

时刻会计师跟踪隐私损失随机变量的时刻的界限（在下面的等式（1）中定义）。它推广了跟踪的标准方法并使用了强组合定理。虽然之前已知这种改进用于组成高斯机制，但我们表明它也适用于组成使用随机采样的高斯机制，并且可以提供更严格的算法1隐私损失估计。

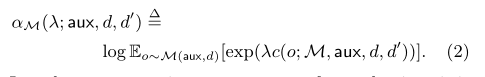
**隐私损失**是随机变量，取决于添加到算法中的随机噪声。机制M是- 差分隐私的等同于一个M的隐私损失随机变量上的某个尾界。虽然尾部绑定是关于分布的非常有用的信息，但直接从它组成可能会导致非常宽松的边界。我们改为计算隐私损失随机变量的 对数矩/ log moments，它是线性组成的。然后，我们使用时间限制以及标准马尔可夫不等式来获得尾部界限，即差异隐私意义上的隐私损失。

更具体地说，对于相邻数据库 ，机制M，辅助输入aux和结果o∈R，将o处的隐私损失定义为：



我们在本文中广泛使用的一种常见设计模式是通过顺序应用差分隐私机制来更新状态。这是*自适应合*成的一个实例，我们通过让所有先前机制的输出为第k个机制的辅助输入来进行建模。

对于给定的机制M，我们将第λ个时刻定义为在值λ处评估的时刻生成函数的对数值：

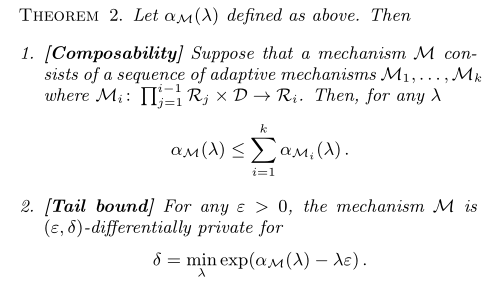


为了证明机制的隐私保证，绑定所有可能的是有用的。 我们定义



其中最大值取自所有可能的辅助数据库和所有相邻数据库d,d’。

我们陈述了我们用于会计时刻的α的属性。



特别是，当基于先前机制的（公共）输出选择机制本身时，定理2.1成立。

根据定理2，在每一步计算或约束就足够了，并将它们相加以约束整个机制的**moments**。 然后我们可以使用尾部绑定来转换绑定到- 差分隐私保证的moments。

剩下的主要挑战是限制每个步骤的值。 在具有随机采样的高斯机制的情况下，足以估计以下时刻。 设表示的概率密度函数（pdf），表示的pdf。 设是两个高斯的混合。 然后我们需要计算其中



在会计时刻的实现中，我们进行数值积分来计算。 另外，我们可以显示渐近界



与定理2一起，上述约束意味着我们的主要定理1.详细信息可以在论文[4]的完整版本中找到。

3.3超参数调整

我们确定与隐私相关的模型的特征，特别是我们可以调整的超参数，以平衡隐私，准确性和性能。特别地，通过实验，我们观察到模型精度对诸如批量大小和噪声水平之类的训练参数比对神经网络的结构更敏感。

如果我们为超参数尝试多种设置，我们可以通过会计时刻轻松累计所有设置的隐私成本。但是，由于我们只关心为我们提供最准确模型的设置，因此我们可以做得更好，例如应用Gupta等人的成果版本[29]（详见论文的完整版[4]）。

我们可以使用理论上的见解来减少需要尝试的超参数设置的数量。虽然使用小到1的batch大小可以最好地实现凸目标函数的差分隐私优化，但是非凸性学习（其本身不太稳定）可以从聚合到更大batches中获益。同时，定理1表明批量过大会增加隐私成本，合理的权衡是将每个时期的batches数与所需的时期数量相同。随着模型收敛到局部最优，非private训练的学习率通常会小心地向下调整。相比之下，我们永远不需要将学习率降低到非常小的值，因为差分隐私训练永远不会达到合理的制度。另一方面，在我们的实验中，我们发现从相对较大的学习速率开始，然后在几个时期内将其线性衰减到较小的值，并在之后保持不变，有一个小的好处。

4.实施

我们在TensorFlow中实现了差异私有SGD算法。为了保护隐私，我们需要在使用梯度之前对其进行“sanitize”以更新参数。此外，我们需要根据sanitization处理的方式跟踪“隐私支出”。因此，我们的实施主要包括两个部分：sanitizer，它预处理梯度以保护隐私;以及privacy\_accountant，它跟踪训练过程中的隐私支出。

图1包含DPSGD\_Optimizer的TensorFlow代码片段（在Python中），它使用差分隐私SGD和DPTrain最小化损失函数，DPTrain迭代调用DPSGD\_Optimizer，其使用隐私会计师来限制总隐私丢失。

在许多情况下，神经网络模型可以受益于通过将输入投影在主方向（PCA）上或通过卷积层馈送来的输入的处理。我们实施差异私有PCA并应用预先训练的卷积层（在公共数据上学习）。

**Sanitizer**。为了实现隐私保护，Sanitizer需要执行两个操作：（1）通过剪切每个示例的梯度的范数来限制每个单独示例的灵敏度; （2）在更新网络参数之前，将噪声添加到batch的梯度中。

在TensorFlow中，出于性能原因对梯度计算进行批处理，为批处理B的训练示例生成。为了限制更新的灵敏度，我们需要访问每个个体。为此，我们在TensorFlow中实现了per\_example\_gradient运算符，如Goodfellow [27]所述。该运算符可以计算一批个别。通过这种实施，即使对于较大的批量，训练也只有适度的减速。我们当前的实现支持损失函数L的批量计算，其中每个单独连接到L，允许我们处理大多数隐藏层但不包括例如卷积层。

一旦我们可以访问每个示例渐变，就很容易使用TensorFlow运算符来剪切其范数并添加噪声。

**隐私会计师**。我们实施的主要组成部分是PrivacyAccountant，它跟踪培训过程中的隐私支出。正如第3节所讨论的那样，我们实施了**moments accountan**t，它可以累加累积每一步隐私损失时刻的记录。根据噪声分布，可以通过应用渐近边界，评估闭合形式表达式或应用数值积分来计算α（λ）。第一个选项将恢复通用的高级组合定理，后两个选项可以更准确地计算隐私损失。

对于我们使用的高斯机制，α（λ）根据方程式（3）和（4）定义。在我们的实现中，我们进行数值积分以计算这些方程中的E1和E2。我们对不同范围的λ计算α（λ），因此我们可以使用定理2.2计算最佳可能（ε，δ）值。我们发现对于我们感兴趣的参数，足以计算λ≤32的α（λ）。

在训练期间的任何时候，人们都可以使用定理2.2来解释在更可解释的（ε，δ）隐私概念中的隐私损失。罗杰斯等人[49]指出与自适应选择隐私参数相关的风险。我们通过提前修复迭代次数和隐私参数来避免攻击和负面结果。隐私会计师的更一般实现必须正确区分两种操作模式 - 作为隐私里程表或隐私过滤器（更多细节见[49]）。

**差异隐私PCA**。主成分分析（PCA）是捕获输入数据主要特征的有用方法。我们实现差异私有PCA算法，如[25]中所述。更具体地，我们采用训练示例的随机样本，将它们视为向量，并将每个向量归一化为单位l2范数以形成矩阵A，其中每个向量是矩阵中的行。然后，我们将高斯噪声添加到协方差矩阵并计算噪声协方差矩阵的主方向。然后，对于每个输入示例，我们将投影应用于这些主要方向，然后将其投入神经网络。

由于运行PCA，我们会产生隐私费用。然而，正如我们对MNIST数据的实验所表明的，我们发现它对提高模型质量和缩短训练时间都很有用。详见第4节。

**卷积层**。卷积层对深度神经网络很有用。然而，卷积层的有效的每个示例梯度计算仍然是TensorFlow框架内的挑战，这促使创建单独的工作流。例如，最近的一些研究认为即使是随机卷积也足够[48,14,51,56,16]。

或者，我们探讨了在公共数据上学习卷积层的想法，遵循Jarrett等人的观点[32]。这样的卷积层可以基于GoogLeNet或AlexNet特征[55,37]用于图像模型，或者基于预训练的word2vec或GloVe嵌入语言模型[43,46]。

5.实验结果

本节报告我们对会计时刻的评估，以及两个流行图像数据集的结果：MNIST和CIFAR-10。

5.1应用时刻会计

如定理1所示，与通用强组合定理相比，时刻会计对隐私损失提供了更严格的约束。这里我们使用一些具体的值来比较它们。整体隐私损失可以从噪声水平σ，每lot的采样率q = L/N（所以每个时期由1/q批次组成）和时期数E（所以步数是T=E/q）。我们确定目标，这是我们的MNIST和CIFAR实验所用的值。

在我们的实验中，我们设置q = 0.01，σ= 4和δ=，并计算ε的值作为训练时期E的函数。图2显示的两条曲线分别对应于使用强组合物的定理和时刻会计。我们可以看到，通过使用时刻会计，我们可以更加严格地估计隐私损失。例如，当E = 100时，值分别为9.34和1.26，对于E = 400，值分别为24.22和2.55。也就是说，使用时间限制，我们实现（2.55,） - 差异隐私，而以前的技术仅获得（24.22,）的明显更差的保证。

5.2 MNIST

我们对标准的MNIST数据集进行了手写数字识别实验，包括60,000个训练样例和10,000个测试实例[38]。 每个示例是28×28大小的灰度图像。我们使用简单的前馈神经网络，其具有ReLU单位和10类（对应于10位数）的softmax，具有交叉熵损失和可选的PCA输入层。

***基线模型。***

我们的基线模型使用60维PCA投影图层和具有1,000个隐藏单位的单个隐藏图层。 使用600的批量/lot，我们可以在大约100个时期达到98.30％的准确度。 这个结果与使用香草神经网络可以实现的结果一致[38]。

***差分隐私模型***。

对于***差分隐私***版本，我们尝试使用相同的架构，具有60维PCA投影层，单个1,000个单元的ReLU隐藏层，以及600的批量大小。为了限制灵敏度，我们剪切每个层的梯度范数我们报告了噪声标度的三种选择的结果，我们称之为小（σ= 2，σp= 4），中等（σ= 4，σp= 7）和大（σ= 8，σp= 16）。这里σ表示训练神经网络的噪声水平，σp表示PCA投影的噪声水平。学习率初始设定为0.1，并且在10个时期内线性降低至0.052，然后固定为0.052。我们还尝试了多隐藏层网络。对于MNIST，我们发现一个隐藏层与PCA结合比双层网络更好。

图3显示了不同噪声水平的结果。在每个图中，我们将训练和测试精度的演变显示为时期数量和相应的δ值的函数，保持ε固定。我们分别对（0.5,10 -5），（2,10 -5）和（8,10 -5） - 差异隐私实现90％，95％和97％的测试集精度。

应用差异私有SGD的一个有吸引力的结果是模型在训练和测试集上的准确性之间的微小差异，这与差分隐私训练泛化性好的理论论证是一致的[7]。相反，非私人训练中训练和测试准确性之间的差距，即过度拟合的证据，随着时期的数量而增加。

通过使用时刻会计，我们对任何给定ε可以获得一个δ值。我们记录了图4中不同（ε，δ）对的精度。在图中，每条曲线对应于固定δ的最佳精度，因为它在10 -5和10 -2之间变化。例如，我们可以达到ε= 0.25和δ= 0.01的90％准确度。从图中可以看出，对于固定的δ，改变ε的值会对精度产生很大的影响，但对于任何固定的ε，不同δ值的差异较小。

***参数的影响***。

分类精度由多个因素决定，必须仔细调整才能获得最佳性能。这些因素包括网络的拓扑结构，PCA维度的数量和隐藏单元的数量，以及训练过程的参数，例如lots大小和学习率。一些参数特定于隐私，例如梯度范数剪切界限和噪声级别。

为了证明这些参数的影响，我们单独操作它们，保持其余参数不变。我们将参考值设置如下：60个PCA维度，1,000个隐藏单位，600个批量大小，梯度范数界限为4，初始学习率为0.1，在10个时期内降至最终学习率0.052，噪声σ等于4和7分别用于训练神经网络参数和用于PCA投影。对于每个值的组合，我们训练直到违反（2,10 -5） - 差异隐私的点（因此，例如，更大的σ允许更多的训练时期）。结果如图5所示。

**PCA投影**。在我们的实验中，精度作为PCA维度的函数是相当稳定的，并且获得了60的最佳结果。（不进行PCA会使精度降低约2％。）虽然原则上PCA投影层可以被额外的隐藏层替换，我们通过单独训练PCA层来实现更高的准确性。通过将输入大小从784减少到60，PCA导致训练时间减少近10倍。结果在PCA投影的大范围噪声水平上相当稳定，并且始终优于使用随机投影的精度，其为约92.5％并且在图中显示为水平线。

**隐藏单位数量**。包括更多隐藏单元使得更容易适应训练集。对于非私人培训，通常最好使用更多单元，只要我们采用技术来避免过度拟合。然而，对于差分私人训练，如果更多隐藏单元提高准确度，则不是先验清楚，因为更多隐藏单元增加了梯度的灵敏度，这导致在每次更新时添加更多噪声。

有点违反直觉，增加隐藏单元的数量不会降低训练模型的准确性。需要进一步分析的一个可能的解释是更大的网络更容忍噪声。这个属性非常令人鼓舞，因为在实践中使用非常大的网络是很常见的。

**Lot size**。根据定理1，我们可以运行N / L时期，同时保持不变的隐私预算。选择lot size必须平衡两个相互冲突的目标。一方面，较小的lot允许运行更多的时期，即，传递数据，提高准确性。另一方面，对于更大的lot，增加的噪声具有较小的相对效果。

我们的实验表明，lot size对准确性的影响相对较大。根据经验，最佳手数大致为√N，其中N是训练样本的数量。

**学习率**。学习率的稳定性在[0.01,0.07]范围内稳定，峰值在0.05，如图5（4）所示。但是，如果学习率太大，准确度会显着下降。一些额外的实验表明，即使对于较大的学习率，我们也可以通过降低噪音水平达到相似的准确度，因此，通过减少训练，以避免耗尽隐私预算。

**Clipping bound**。限制梯度范数有两个相反的作用：clipping破坏梯度估计的无偏性，并且如果该限幅参数太小，削波破坏了梯度估计的无偏性，并且如果削波参数太小，则平均削波梯度可能指向与真实梯度非常不同的方向。另一方面，增加范数界限C迫使我们向梯度（以及因此参数）添加更多噪声，因为我们基于σC添加噪声。在实践中，选择C值的一种好方法是在训练过程中取出未剪切梯度的范数中值。

**噪音水平**。通过添加更多噪声，每步隐私损失成比例地减小，因此我们可以在给定的累积隐私预算内运行更多的时期。在图5（5）中，x轴是噪声水平σ。此值的选择会对准确性产生很大影响。

从实验中，我们观察到以下情况。

1. PCA预测提高了模型精度和训练性能。对于投影尺寸和PCA阶段中使用的噪声水平

的大范围选择，精度非常稳定。

2.对不同网络规模的准确性相当稳定。当我们只能运行较少数量的时期时，使用更大

的网络更有利。

3.训练参数，特别是批量大小和噪声标度σ，对模型精度有很大影响。它们都确定了

sanitized gradients的“噪声 - 信号”比率以及在达到隐私限制之前我们能够通过数据的时期数量。

我们的框架允许自适应控制训练参数，例如批量大小，梯度范数界限C和噪声水平σ。我们在训练过程中降低噪声的初步实验没有显示出显着的改进，但考虑更复杂的方案来自适应地选择这些参数是有趣的。

5.3 CIFAR

我们还对CIFAR-10数据集进行了实验，该数据集由分类为10类（如船只，猫和狗）的彩色图像组成，并划分为50,000个训练样本和10,000个测试示例[1]。每个示例是具有三个通道（RGB）的32×32图像。对于这个学习任务，几乎所有成功的网络都使用卷积层.CIFAR-100数据集具有相似的参数，除了图像被分类为100个类;示例和图像类与CIFAR-10不同。

我们使用来自TensorFlow卷积神经网络教程的网络架构[2]。通过取中心贴片，首先将每个32×32图像裁剪为24×24图像。网络架构由两个卷积层组成，后跟两个完全连接的层。卷积层使用5×5圈数和步幅1，然后是ReLU和2×2最大池，每个池有64个通道。从而第一个卷积为每个图像输出12×12×64张量，第二个卷积输出6×6×64张量。后者被压平为矢量，该矢量被馈送到具有384个单元的完全连接层，以及另一个具有相同尺寸的层。

这种非私人体系结构可以在500个时代内达到约86％的准确率。它的简洁使它成为我们工作的有吸引力的选择。然而，我们应该注意到，通过使用具有不同非线性和其他先进技术的更深层网络，可以获得明显更好的准确度，其中现有技术大约为96.5％[28]。

作为此类图像数据集的标准，我们在训练期间使用数据增强。对于每个训练图像，我们通过从图像中随机选取24×24补丁，沿左右方向随机翻转图像，并随机扭曲图像的亮度和对比度来生成新的失真图像。在每个时代，这些扭曲都是独立完成的。我们将读者引用到TensorFlow教程[2]以获取更多详细信息。

由于卷积层具有共享参数，因此计算每个示例梯度具有更大的计算开销。以前的工作表明，卷积层通常是可转移的：从一个数据集中学习的参数可以在另一个数据集上使用而无需重新训练[32]。我们将CIFAR-100数据集视为公共数据集，并使用它来训练具有相同体系结构的网络。我们使用从训练此数据集中学到的卷积。仅使用此架构重新训练完全连接的层约250个时期，批量大小为120，这使我们的准确率达到约80％，这是我们的非私人基线。

**差异私人版本。**

对于差异私有版本，我们使用相同的体系结构。如上所述，我们使用预先训练的卷积层。完全连接的层也从预训练的网络初始化。我们训练softmax层，以及顶部或两个完全连接的层。基于观察梯度范数，softmax层梯度大约是其他两层的两倍，当我们尝试在3到10之间的几个不同值处进行clipping时，我们保持这个比率。lot大小是我们的另一个旋钮调：我们尝试了600,2,000和4,000。通过这些设置，每个历元的训练时间从大约40秒增加到180秒。

在图6中，我们展示了精确度和隐私成本的演变，作为几个不同参数设置的时期数的函数。

各种参数会影响其准确性，其方式与MNIST实验没有太大差别.600的批量lot导致该数据集的结果不佳，我们需要将其增加到2,000或更多，如图6所示。

与MNIST数据集相比，非私人基线与私人模型之间的准确度差异约为1.3％，我们的CIFAR-10实验的准确度相应下降幅度更大（约7％）。我们将这一差距作为对差异私人机器学习的未来研究的有趣测试。

6.相关工作

自90年代末以来，隐私保护数据挖掘或机器学习的问题一直是几个研究团体积极工作的焦点[6,39]。现有文献可以大致分为几个轴：模型类，学习算法和隐私保证。

**隐私保障**。早期的隐私保护学习工作是在**安全功能评估（SFE）和安全多方计算（MPC）**的框架内完成的，其中输入在两方或多方之间分配，重点是最小化在此期间泄露的信息。联合计算一些商定的功能。相反，我们假设数据集中保存，我们关注功能输出（即模型）的泄漏。

另一种方法，k-匿名和密切相关的概念[54]，试图通过推广和抑制某些识别属性来为基础数据提供一定程度的保护。该方法具有很强的理论和经验限制[5,10]，使得它几乎不适用于高维度，多样化的输入数据集的去匿名化。除了追求输入清理之外，我们保持基础原始记录完整并扰乱派生数据。

差异隐私理论为我们的工作提供了分析框架，已经应用于大量的机器学习任务，这些任务在培训机制或目标模型中与我们不同。

会计时刻与R'enyi差异隐私[44]的概念密切相关，后者提出（缩放）α（λ）作为量化隐私保障的手段。在并行和独立的工作中，Bun和Steinke [11]引入了通过α（λ）上的线性上界定义的差异隐私的放宽（推广Dwork和Rothblum [22]的工作）。总之，这些工作表明，会计时刻是一种有用的技术，可用于复杂的隐私保护算法的理论和实证分析。

**学习算法**。隐私学习的一个共同目标是一类适用于各种技术的凸优化问题[20,12,36]。在同时工作中，吴等人。通过凸经验风险最小化实现MNIST的准确率达到83％[58]。训练多层神经网络是非凸的，并且通常通过SGD的应用来解决，SGD的理论保证很难理解。

对于CIFAR神经网络，我们结合了PCA投影矩阵的差分私有训练[25]，用于降低输入的维数。

**模型类**。第一个端到端的差异私有系统是在Netflix Prize数据集[41]上进行评估的，这是一种协同过滤问题的版本。虽然这个问题与我们的高维输入，非凸目标函数有许多相似之处 - McSherry和Mironov采用的方法差别很大。他们确定了学习任务的核心，即有效的足够的统计数据，可以通过高斯机制以差异私有的方式计算。在我们的方法中，没有足够的统计数据。

在最近的一项工作中，Shokri和Shmatikov [52]设计并评估了一种用于深度神经网络的分布式训练的系统。密切关注数据的参与者将清理过的更新传达给中央机构。消毒依赖于基于灵敏度估计的加性噪声​​机制，可以将其改进为硬灵敏度保证。他们计算每个参数的隐私损失（不是整个模型）。根据我们的首选衡量标准，MNIST数据集上每位参与者的总隐私损失超过数千。

Phan等人探讨了一种不同的近期差异私人深度学习方法[47]。这项工作的重点是学习自动编码器。隐私是基于扰乱这些自动编码器的目标功能。

7.结论

我们展示了具有差异隐私的深度神经网络的训练，产生适度的完全隐私损失，在具有许多参数的整个模型上进行计算。在我们的MNIST实验中，我们达到了97％的训练精度，对于CIFAR-10，我们达到了73％的准确度，两者都具有（8,10 -5） - 差异隐私。我们的算法基于随机梯度下降的差异私有版本;它们在TensorFlow软件库上运行，用于机器学习。由于我们的方法直接应用于梯度计算，因此它可以适应许多其他经典和更新的一阶优化方法，如NAG [45]，Momentum [50]，AdaGrad [17]或SVRG [33]。

一种新的工具，可能是独立的利益，是一种跟踪隐私损失的机制，即会计时刻。它允许对复杂复合机构的隐私损失进行严格的自动化分析，这种复杂机制目前超出了先进的组合定理的范围。

许多进一步工作的途径都很有吸引力。特别是，我们想考虑其他类别的深度网络。我们对MNIST和CIFAR-10的经验应该会有所帮助，但我们看到了许多新研究的机会，例如将我们的技术应用于用于语言建模任务的LSTM。此外，我们希望获得更高的准确性。许多训练数据集远大于MNIST和CIFAR-10;准确性应从其规模中受益。