差分プライベートなベイジアンニューラルネットワークのプライバシーリスク

芝原 俊樹[1,a)]　三浦 尭之[1]　紀伊 真昇[1]　市川 敦謙[1]

概要：ベイジアンニューラルネットワーク（NN）は，予測値の不確実性も出力できるため，deep neural network (DNN) の誤識別が深刻な影響を及ぼすタスクで注目されている．特に，Monte Calro (MC) dropout を用いると，NN の構造を変えずに予測時の使い方を変えるだけで，決定的な NN をベイジアン NN として使うことができる．このとき，ベイジアン NN では，より多くの情報を出力するため，教師データの情報が漏洩するリスクも高まる可能性があるが，詳細な調査は行われていない．そこで，本稿では，DNN を決定的な NN として使用した場合と， MC dropout を適用してベイジアン NN として使用した場合で，プライバシーリスクがどの程度異なるかを評価する．プライバシーリスクの評価は，ベイジアン NN の利用シーンを想定して定義した攻撃モデルに従い，ベイジアン NN の出力に基づいてリスクを評価する提案手法を用いて行った．CIFAR-10 と CNN を使用した実験で，ベイジアン NN は決定的な NN よりプライバシーリスクが高いこと，ベイジアン NN の出力を量子化することで有用性を維持しつつプライバシーリスクを低減できる可能性が高いことを示した．

Privacy Risk of Differentially Private Bayesian Neural Network

Toshiki Shibahara[1,a)]　Takayuki Miura[1]　Masanobu Kii[1]　Atsunori Ichikawa[1]

Abstract: Bayesian neural networks (NNs) that output uncertainties of predictions have attracted attention in a critical situation. Especially, Monte Calro (MC) dropout enables us to use deterministic NNs as Bayesian NNs without changing NN architectures. Since Bayesian NNs output more information than deterministic NNs, Bayesian NNs potentially have higher privacy risks. However, whether applying MC dropout increases privacy risks of deep NNs (DNNs) or not has not been studied. Therefore, we compare privacy risks of deterministic NNs to those of Bayesian NNs. For the comparison, we define an attack model against Bayesian NNs and design a method for evaluating privacy risks of Bayesian NNs. Our experiments using CIFAR-10 and CNN show that Bayesian NNs have higher privacy risks than deterministic NNs and that privacy risks can be decreased without degrading the utility by quantizing outputs of Bayesian NNs.

1. はじめに

Deep neural network (DNN) の発展に伴い，医療現場での診断や自動運転のように DNN の誤識別が深刻な影響を及ぼすタスクでの活用も検討されている．このようなタスクでは，予測値だけでなく予測値の不確実性も出力できるベイジアンニューラルネットワークが注目されている [1–4]．不確実性を算出するためには，特別な確率計算やニューラルネットワーク（NN）が必要となる手法が多いが，Monte Calro (MC) dropout [4] は，dropout [5] が適用されていれば NN の構造を変えずに予測値の不確実性を算出できるため注目されている．不確実性の算出では，ランダムに一部のニューロンを取り除く dropout の操作を予測時にも適用して複数回予測値をサンプリングし，その分散から不確実性を算出する．つまり，予測時の NN の使い方を変えるだけで，決定的な NN として使用されていた NN をベイジアン NN として使用することができる．

DNN の実応用が進むにつれて，DNN の教師データを記憶しやすい特性が，プライバシーリスクにつながることも

[1] NTT 社会情報研究所
NTT Social Informatics Laboratories
[a)] toshiki.shibahara.de@hco.ntt.co.jp

指摘されている [6,7]．具体的には，特定のデータが教師データに含まれていたかどうかを，DNN の出力から特定できることが示されている．医療データやウェブサイトの閲覧履歴など，ユーザが他人に知られたくないデータを扱う場合は，プライバシーリスクを低減させる必要がある．

DNN の学習で用いられる代表的なプライバシー保護手法は，確率的勾配降下法 (SGD) で差分プライバシーを保証した differentially private SGD (DP-SGD) である [8]．DP-SGD では，パラメータの更新時にノイズを加えることで，教師データを保護している．DP-SGD で学習した DNN は，予測時には制約なく使用することができる．つまり，決定的な NN として使用することもできるが，MC dropout を適用してベイジアン NN として使用することもできる．どちらの使い方をしても差分プライバシーは保証されているが，決定的な NN は 1 つの予測値を出力するのに対し，ベイジアン NN はサンプリングされた複数の予測値を出力するため，より多くの情報を出力していることになる．そのため，ベイジアン NN として使用することで，プライバシーリスクが高まる可能性があるが，MC dropout を適用することによるプライバシーリスクへの影響は調査されていない．

本稿では，DNN を決定的な NN として使用した場合と，MC dropout を適用してベイジアン NN として使用した場合で，プライバシーリスクがどの程度異なるかを評価する．具体的には，下記 3 つの research questions (RQs) について調査する．

RQ1 決定的な NN よりベイジアン NN はプライバシーリスクが高いか？

RQ2 MC dropout のサンプリングの回数はプライバシーリスクに影響するか？

RQ3 ベイジアン NN の有用性を維持しつつプライバシーリスクを低減できるか？

プライバシーリスクの評価には，あるデータが教師データに含まれていたか特定する攻撃が，どの程度成功するかに基づいて算出する手法 [9,10] を活用する．ただし，先行研究で提案されていた攻撃は決定的な NN が対象のため，本稿では，ベイジアン NN の利用シーンを想定した攻撃モデルの定義と，ベイジアン NN の出力に基づく攻撃の提案を行った．攻撃で使用できる出力は，決定的な NN が 1 つの予測値であるのに対し，MC dropout ではサンプリングされた複数の予測値である．そこで，サンプリングされた予測値から平均値，標準偏差，最大値，最小値を求めて，これらの値が閾値を超えるかに基づいて，特定のデータが教師データに含まれていたか判定する攻撃を提案する．

RQ を明らかにするために，差分プライバシーの評価でも用いられている CIFAR-10[*1]を教師データとして使用して，畳み込み層 2 層と全結合層 2 層で構成される DNN を DP-SGD で学習させて実験を行った．実験では，この DNN を決定的な NN と用いる場合と，MC dropout を適用してベイジアン NN として用いる場合のプライバシーリスクを比較し，プライバシーリスクが高くなる条件を分析した．本稿の主な結果は下記の通りである．**RQ1**：ベイジアン NN は決定的な NN よりプライバシーリスクが高い．**RQ2**：サンプリング回数が 2 回以上では，プライバシーリスクに大きな変化がない．**RQ3**：出力を量子化することで，有用性を維持しつつプライバシーリスクを低減できる可能性が高い．

[*1] https://www.cs.toronto.edu/ kriz/cifar.html

2. 準備

本章では，提案手法が前提としている DNN と差分プライバシーに関連する研究について説明する．

2.1 Deep Neural Network

DNN は入力 $\boldsymbol{x} \in \mathcal{X}$ からラベル $\boldsymbol{y} \in \mathcal{Y}$ を予測する関数 $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$ として定義される．DNN は複数の層で構成されており，L 層で構成される DNN の i 番目の層を $\boldsymbol{f}_i$ とすると，予測値は $\hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}_L(\boldsymbol{f}_{L-1}(\cdots \boldsymbol{f}_2(\boldsymbol{f}_1(\boldsymbol{x}))))$ となる．

DNN の学習では，入力とラベルの組の集合である教師データ $D = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{|D|}$ を用いて，$\boldsymbol{f}$ のパラメータ $\boldsymbol{\theta}$ の最適化を行う．最適化では，事前に設定された損失 ℓ を最小化する．分類タスクの代表的な損失関数はクロスエントロピーである．パラメータ $\boldsymbol{\theta}$ の DNN でデータ $(\boldsymbol{x}, \boldsymbol{y})$ を予測した際のクロスエントロピーは，予測値の対数尤度とラベルを使って定義され，$\ell(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{y}) = -\boldsymbol{y} \log \hat{\boldsymbol{y}}$ となる．

損失を最小化するための代表的な手法は，確率的勾配降下法（SGD）である．SGD では，教師データ D の部分集合をミニバッチ $B \subset D$ としてランダムに選択し，ミニバッチごとにパラメータを更新する．具体的には，損失の勾配に基づいて下記のように更新する．

$$\boldsymbol{\theta}_{i+1} \leftarrow \boldsymbol{\theta}_i - \eta \left(\frac{1}{|B|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in B} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_i; \boldsymbol{x}, \boldsymbol{y}) \right) \quad (1)$$

ここで，η は学習率，$\nabla_{\boldsymbol{\theta}}$ はパラメータでの勾配である．

2.2 Monte Carlo Dropout

Dropout を適用した NN の学習が，変分ベイズの近似とみなせることに着目した手法が MC dropout である [4]．変分ベイズでは，入力 $\boldsymbol{x}$ に対するラベル $\boldsymbol{y}$ の事後確率 $q(\boldsymbol{y}|\boldsymbol{x})$ を下記のように算出する．

$$q(\boldsymbol{y}|\boldsymbol{x}) = \int p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) q(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \quad (2)$$

ここで，$q(\boldsymbol{\theta})$ はパラメータの分布である．

Dropout では，一定の割合のニューロンをランダムに

NN から取り除いて学習を行う [5]．MC dropout では，この一部のニューロンを取り除いた NN のパラメータ $\boldsymbol{\theta}^i$ を $q(\boldsymbol{\theta})$ からのサンプリングと考え，式 2 を T 回のサンプリングで近似する．事後確率の期待値 $\mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x})}(\boldsymbol{y})$ は，$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^i)=\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}^i)$ であることを使うと，下記のように算出できる．

$$\mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x})}(\boldsymbol{y}) \approx \frac{1}{T}\sum_{i=1}^{T}\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}^i) \tag{3}$$

また，予測の不確実性の指標となる事後確率の分散は下記のように算出できる．

$$\mathrm{Var}_{q(\boldsymbol{y}|\boldsymbol{x})}(\boldsymbol{y}) \approx \beta\mathbf{I} + \frac{1}{T}\sum_{i=1}^{T}\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}^i)\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}^i)^T - \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x})}(\boldsymbol{y})\mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x})}(\boldsymbol{y})^T \tag{4}$$

ここで，β は事前に決められたパラメータである．つまり，予測時にも dropout を適用して T 回予測値を算出し，その平均と分散から事後確率の期待値と分散の近似を求めることができる．

2.3 Differentially Private SGD

DP-SGD は SGD を差分プライベートにした手法である．まず，差分プライバシーの定義について説明する．(ϵ,δ)-差分プライバシーでは，アルゴリズム $\mathcal{M}:\mathcal{D}\to\mathcal{R}$ と隣接するデータセット $D,D'\subset\mathcal{D}$ を考える．ここで，$\mathcal{D}$ はすべてのデータの集合，$\mathcal{R}$ はアルゴリズムの出力の値域である．本稿では，隣接データセットとして，1 要素のみ異なるデータセットを考える．具体的には，異なる要素を $(\boldsymbol{x}',\boldsymbol{y}')$ とすると，$D'=D\cup\{(\boldsymbol{x}',\boldsymbol{y}')\}$ となる．すべての隣接データセット，値域の部分集合 $S\subset\mathcal{R}$ に対し下記を満たすとき，アルゴリズム $\mathcal{M}$ は (ϵ,δ)-差分プライベートであるという．

$$Pr[\mathcal{M}(D)\in S] \le e^{\epsilon}Pr[\mathcal{M}(D')\in S]+\delta \tag{5}$$

この定義では，アルゴリズムが小さな ϵ を満たすとき，隣接するデータセットを明確には区別できないことを保証している．一般的に，δ は 10^{-5} などの小さな定数が使われるため，ϵ がアルゴリズムのプライバシーリスクを表す指標となる．差分プライバシーでは，どのような条件でも超えることのない ϵ，つまりプライバシーリスクの上界を求めることで安全性を保証している．

DP-SGD は，SGD の更新式に，正規分布に従うノイズを加えることで，差分プライバシーを保証する手法である．具体的には，ノイズ $Z_i\sim\mathcal{N}(0,\sigma^2C^2\mathbb{I})$ を用いて下記のようにパラメータを更新する．

$$\boldsymbol{\theta}_{i+1}\leftarrow\boldsymbol{\theta}_i-\eta\left(\frac{1}{|B|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in B}\mathbf{clip}_C(\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}_i;\boldsymbol{x},\boldsymbol{y}))+Z_i\right) \tag{6}$$

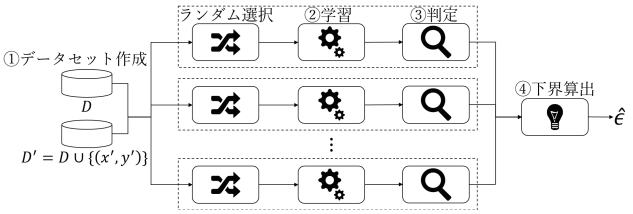


図 1: プライバシーリスクの下界算出の流れ

Algorithm 1 決定的な NN の出力に基づく判定方法

Require: モデル $\boldsymbol{f}$, データ $(\boldsymbol{x}',\boldsymbol{y}')$, 閾値 τ
1: **if** $\ell(\boldsymbol{\theta};\boldsymbol{x}',\boldsymbol{y}')<\tau$ **then**
2: return D'
3: **end if**
4: return D

ここで，$\mathbf{clip}_C$ はベクトルの L2 ノルムを最大 C に制限する関数で，下記で定義される．

$$\mathbf{clip}_C(\boldsymbol{v}) = \boldsymbol{v}\cdot\min\left(1,\frac{C}{||\boldsymbol{v}||_2}\right)$$

DP-SGD が満たす ϵ に関しては多くの研究がある [8,11]．

2.4 プライバシーリスクの下界

差分プライバシーでは，プライバシーリスクの上界を求めているが，プライバシーリスクの下界を求める研究も近年行われている [9,10]．これらの研究では，隣接するデータセット D,D' とアルゴリズム $\mathcal{M}$ が与えられたときに，アルゴリズムの出力からデータセットを区別する攻撃がどの程度成功するか評価する．実現した攻撃に相当するプライバシーリスクを $\hat{\epsilon}$ とすると，実際のアルゴリズムの ϵ は，少なくとも $\hat{\epsilon}$ 以上となる．つまり，$\hat{\epsilon}$ はプライバシーリスクの下界となっている．プライバシーリスクの下界を知ることで，差分プライバシーの理論値が厳密にリスクを評価できているかや，現実的な条件でどの程度のプライバシーリスクがあるのかを確認することができる．プライバシーリスクの下界を求める手法は，図 1 に示すように，主に 4 つのステップで構成される．

Step 1：データセット作成 最初に，攻撃対象となる隣接するデータセットを作成する．データセット D と異なる要素となるデータ $(\boldsymbol{x}',\boldsymbol{y}')$ が得られた場合は，隣接データセットは $D'=D\cup\{(\boldsymbol{x}',\boldsymbol{y}')\}$ となる．このとき，データ $(\boldsymbol{x}',\boldsymbol{y}')$ を用意する方法は，攻撃者がデータセットを操作可能かなど，攻撃者の能力に応じて，様々な作成方法が考えられる．本稿で想定する攻撃者の能力とデータセット作成の方法は 3 章で説明する．

Step 2：学習 次に，データセット D,D' からランダムに 1 つを選択し，DNN $\boldsymbol{f}$ の学習を行う．学習には，DP-SGD に代表されるプライバシー保護手法が用いられる．

Step 3：判定 学習結果に基づいて，攻撃者は D,D' のどちらが学習に用いられていたか判定する．判定も攻撃者の

利用できる情報によって様々な方法が考えられる．最もシンプルな方法は，アルゴリズム 1 に示すように，データ $(\boldsymbol{x}', \boldsymbol{y}')$ の損失に基づいて判定する方法である．この方法では，損失が閾値より小さい場合には，教師データに含まれていたと判断し，データセットは D' とする．損失が閾値 τ 以上の場合は，データが教師データに含まれていなかったと判断し，データセットは D と判定する．このとき閾値 τ は最も強い攻撃となるように，次のステップで算出するプライバシーリスクの下界 $\hat{\epsilon}$ が最も大きくなるものを採用する．

Step 4：下界算出 最後に攻撃がどの程度成功するかに基づいて，プライバシーリスクの下界を算出する．攻撃の成功率を算出する必要があるため，同一の隣接データセットで，Step 2 と Step 3 を繰り返し行う．攻撃者が達成可能な誤検知率 (FPR) と見逃し率 (FNR) から式 5 で定義される ϵ と δ を求めることができる [12]．具体的には，下記 2 式のどちらかを満たす ϵ と δ の組み合わせが下界となる．

$$FPR + e^{\epsilon} FNR \leq 1 - \delta$$
$$FNR + e^{\epsilon} FPR \leq 1 - \delta$$

δ が与えられている場合は，上式どちらかを満たす最大の ϵ が下界となり，$\hat{\epsilon}$ を下記で求めることができる．

$$\hat{\epsilon} = \max\left(\log\frac{1-\delta-FPR}{FNR}, \log\frac{1-\delta-FNR}{FPR}\right) \quad (7)$$

3. ベイジアンニューラルネットワークのプライバシーリスクの下界

本稿では，MC dropout に基づくベイジアン NN の差分プライバシーの下界を求める手法を提案する．まず，ベイジアン NN の利用シーンから，攻撃者の能力（攻撃モデル）を定義し，現実的な手法を設計する．全体の流れは，先行研究と同様に図 1 に従って行う．ただし，データセットは，攻撃モデルに従って 4 つの方法で作成する．データセットの判定も，ベイジアン NN の出力と攻撃モデルに適した方法を設計する．決定的な NN は 1 つの予測値を出力するが，MC dropout は事後分布からサンプリングされた複数の予測値を出力する．そこで，サンプリング結果に基づいた判定方法を設計した．本章では，攻撃モデル，データセット作成，ベイジアン NN の出力に基づく判定について説明する．

3.1 攻撃モデル

本稿で想定する攻撃者の能力について説明する．攻撃者としては，学習済みモデルを利用するユーザを想定する．この設定は，医療現場での診断モデルやローン審査モデルなどを利用する状況では自然である．攻撃者は，これらの学習済みモデルを利用して，教師データに特定のデータが含まれていたかを判定する．

次に，この設定に適したデータセット作成と判定方法を説明する．データセット作成に関しては，この設定では攻撃者は操作することができない．しかし，攻撃者が教師データに含まれているか特定したいデータの特性（学習の難易度，他のデータとの類似度）によって，プライバシーリスクは異なると考えられる．そこで，異なる要素となるデータの特性が異なる 4 種類の隣接するデータセット用意して実験を行った．判定では，教師データに含まれているか判定するデータの出力を攻撃者が利用することを想定する．判定対象以外のデータの出力も利用することができるが，本稿の比較対象である決定的な NN の出力に基づく判定方法 [9,10] と攻撃の条件をそろえるために，判定対象のデータの出力に基づく方法を検討した．ベイジアン NN の出力としては，サンプリングされた複数の予測値を直接受け取る方法と，平均や標準偏差などのサンプリング結果の統計値を受け取る方法が考えられる．本稿では，両者を想定して判定方法を設計した．

3.2 データセット作成

学習の難易度と他のデータとの類似度が異なるデータを，隣接するデータセット間で異なる要素 $(\boldsymbol{x}', \boldsymbol{y}')$ として用意し，4 つのデータセットを作成した．データをランダムに選択する方法と，損失の大きいデータを選択する方法で，難易度の異なるデータを用意した．選択されたデータをそのまま使用する方法と，損失を大きくする adversarial noise を加えて使用する方法で，他のデータとの類似度の異なるデータを用意した．これらの方法を組み合わせて，下記の通り 4 つのデータセットを作成した．

データセット 1：ランダム選択 学習が比較的容易なデータとして，D に含まれないデータをランダムに選択し，隣接データセットに追加する要素として使用した．

データセット 2：ランダム選択 + adversarial noise データセット 1 と同様にランダムに選択したサンプルに，adversarial noise を付加し，隣接データセットに追加する要素として使用した．Adversarial noise は，損失を大きくするノイズのため，他のデータとの類似度は低くなると想定される．ノイズ付加では，データセット D 以外のデータセット D_s で学習した shadow model $\tilde{\boldsymbol{f}}_1, \tilde{\boldsymbol{f}}_2, \cdots, \tilde{\boldsymbol{f}}_n$ を n 個用意する．この学習には，2.4 章の Step 2 と同じプライバシー保護手法で学習を行う．次に，adversarial example を作成する手法 [13] を参考に，これらの shadow model の損失が大きくなるように，少しずつデータを更新する．ランダムに選択されたサンプルを $(\boldsymbol{x}'_0, \boldsymbol{y}'_0)$ とすると更新式は下記の通りである．

$$\boldsymbol{x}'_{i+1} = \boldsymbol{x}'_i + \alpha \ \mathrm{sign}\left(\frac{1}{n}\sum_{j=1}^{n} \nabla_{\boldsymbol{x}} \ell(\tilde{\boldsymbol{\theta}}_j, \boldsymbol{x}'_i, \boldsymbol{y}'_0)\right) \quad (8)$$

ここで，sign は符号関数であり，行列の各要素を正の場

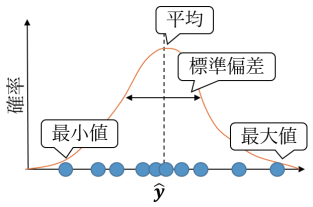


図 2: サンプリング

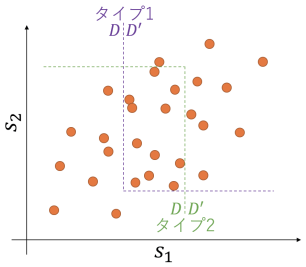


図 3: 識別タイプ

合 1，負の場合 −1 に変換する関数である．また，α は正の小さな定数である．Adversarial example とは異なり，ノイズのノルムに関する制約は設けなかった．本稿では，adversarial example 作成で用いられる設定を参考にして，更新の回数は 40，α は 0.01，shadow model の数は 10 とした．

データセット 3：損失大 このデータセットでは，まず D' を用意し，学習が難しいデータとして，下記のとおり shadow model の損失が大きいサンプルを $(\boldsymbol{x}', \boldsymbol{y}')$ として選択する．

$$(\boldsymbol{x}', \boldsymbol{y}') = \arg\max_{(\boldsymbol{x},\boldsymbol{y})\in D'} \frac{1}{n}\sum_{i=1}^{n} \ell(\tilde{\boldsymbol{\theta}}_i, \boldsymbol{x}, \boldsymbol{y})$$

選択されたデータを D' から取り除くことで，$D = D'\backslash\{(\boldsymbol{x}', \boldsymbol{y}')\}$ を用意した．

データセット 4：損失大 ＋ adversarial noise まず，データセット D'' を用意し，データセット 3 と同じ方法でデータ $(\boldsymbol{x}', \boldsymbol{y}')$ を選択する．これに式 8 の方法で adversarial noise を付加したものを $(\boldsymbol{x}'_{adv}, \boldsymbol{y}')$ とする．作成したデータセットは，$D = D''\backslash\{(\boldsymbol{x}', \boldsymbol{y}')\}$，$D' = D \cup \{(\boldsymbol{x}'_{adv}, \boldsymbol{y}')\}$ となる．

3.3 ベイジアン NN の出力に基づく判定

判定方法としては，ベイジアン NN でサンプリングされた予測値を全部使用できる場合と，平均と標準偏差が使用できる場合を検討する．サンプリング結果をすべて使用できる場合も，その中から事後分布による影響を受けやすい最小値や最大値，分布を表現する統計量である平均や標準偏差を使用することで判定しやすくなる．そこで，どちらの場合も，サンプリング結果から図 2 に示す分布を特徴づける情報を抽出して，それらの情報をもとにデータセットを判定する方法として整理する．ただし，サンプリング結果をすべて使用できる場合と，平均と標準偏差が使用できる場合では，使用できる情報の種類や数が異なる．

ベイジアン NN の i 回目のサンプリングを $\hat{\boldsymbol{y}}_i$，T 回のサンプリングを $\hat{\boldsymbol{Y}}_T = \{\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \cdots, \hat{\boldsymbol{y}}_T\}$ とする．判定に使用する情報を取得する関数を S_i，算出された情報を $s_i = S_i(\hat{\boldsymbol{Y}}_T)$ と表す．このとき，s_i は攻撃対象のデータが教師データに含まれていた場合に大きくなるように設計する．具体的には，最大値，最小値，平均は，$\hat{\boldsymbol{y}}_i$ が予測値のため教師データに含まれていた場合に大きくなると想定される．標準偏差は，予測の不確実性を表しているため，教師データに含まれていた場合に小さくなると想定される．そのため，標準偏差に関しては逆数を用いる．先行研究と同様に，取得された情報が閾値 τ_i を超えるかに基づいて判定を行う．ただし，取得する情報は複数であるため，図 3 に示す 2 つのタイプを想定する．タイプ 1 は，すべての情報がそれぞれの閾値より大きい場合に D' とする．タイプ 2 は，いずれかの情報がそれぞれの閾値より大きい場合に D' とする．アルゴリズム 2 により詳細な定義を示す．閾値は最も強い攻撃となるように，$\hat{\epsilon}$ が最大となる閾値を選択した．

Algorithm 2 ベイジアン NN の出力に基づく判定方法

Require: モデル $\boldsymbol{f}$, データ $(\boldsymbol{x}', \boldsymbol{y}')$, サンプリング回数 T, 関数 $S_1, \cdots S_k$, 閾値 $\tau_1, \cdots, \tau_k$, タイプ $type$

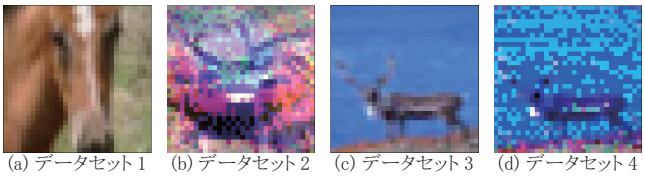
$\hat{\boldsymbol{Y}}_T \leftarrow \{\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \cdots, \hat{\boldsymbol{y}}_T\}$
$s_1, \cdots, s_k \leftarrow S_1(\hat{\boldsymbol{Y}}_T), \cdots, S_k(\hat{\boldsymbol{Y}}_T)$
if $type = 1$ **then**
  if $s_1 > \tau_1 \cdots$ **and** $s_k > \tau_k$ **then**
    return D'
  else
    return D
  end if
else if $type = 2$ **then**
  if $s_1 > \tau_1 \cdots$ **or** $s_k > \tau_k$ **then**
    return D'
  else
    return D
  end if
end if

4. 実験

決定的な NN とベイジアン NN のプライバシーリスクを比較することで，NN の使い方によってプライバシーリスクがどの程度変化するかを調査する．3.3 章の判定方法に基づく攻撃の強さは，使用する情報の種類，数，信頼性によって異なると想定されるため，プライバシーリスクが高くなる条件についても調査した．具体的には，下記 3 つの RQ を明らかにする実験を行った．**RQ1：**決定的な NN よりベイジアン NN はプライバシーリスクが高いか？ **RQ2：**サンプリングの回数はプライバシーリスクに影響があるか？ **RQ3：**有用性を維持してプライバシーリスクを低減できるか？

4.1 実験設定

実験の流れ 2.4 章の方法で算出された $\hat{\epsilon}$ は，試行ごとのばらつきが大きかったため，$\hat{\epsilon}$ の算出を 10 回行い，その平均と標準偏差を実験結果として示す．各試行では，2.4 章の Step 2 と Step 3 を 100 回繰り返して行った．予備実験では，100 回繰り返した場合と，1,000 回繰り返した場合で算出される $\hat{\epsilon}$ に大きな違いがなかったため，本稿では試行回数を増やすために，繰り返し回数を 100 回とした．



(a) データセット 1　(b) データセット 2　(c) データセット 3　(d) データセット 4

図 4: 隣接データセットに追加したデータ

表 1: CNN の構成

Layer	Type (Activation)	Patch size	Output size
1	Convolution (ReLU)	5×5	$28 \times 28 \times 64$
2	Dropout		$28 \times 28 \times 64$
3	Max pooling	2×2	$14 \times 14 \times 64$
4	Convolution (ReLU)	5×5	$10 \times 10 \times 64$
5	Dropout		$10 \times 10 \times 64$
6	Max pooling	2×2	$5 \times 5 \times 64$
7	Fully connected (ReLU)		512
8	Dropout		512
11	Fully connected (Softmax)		10

データセット　データセットは，差分プライバシーの研究でもよく使用される画像分類のデータセット CIFAR-10[*2]を用いた．CIFAR-10 は，10 クラスの 32×32 のカラー画像データセットであり，50,000 枚が教師データ，10,000 枚がテストデータである．教師データを隣接データセット作成に用い，テストデータはモデルの精度評価と shadow model の学習に用いた．

3.2 章の方法で作成した隣接するデータセット間で異なる要素 $(\boldsymbol{x}', \boldsymbol{y}')$ を図 4 に示す．Adversarial noise のノルムに制約を設けていないため，図 4(b, d) は物体の認識が難しくなっている．正解のクラスは図 4(a) は horse（馬），図 4(b–d) は deer（鹿）である．

ニューラルネットワーク　先行研究 [10] と同様に CNN を用いた．この CNN は，畳み込み層 2 層の後に，全結合層 2 層を適用する．畳み込み層の後には Max pooling，畳み込み層と最初の全結合層の後に Dropout を適用した．Dropout 率は 0.5 とした．詳細は，表 1 の通りである．

ハイパーパラメータ　ハイパーパラメータは先行研究 [10] に従って決定した．具体的には，エポック数は 60，バッチサイズは 256，学習率は 0.15 である．DP-SGD のパラメータとしては，勾配の L2 ノルムの制約 C は 1，ノイズの分散 σ^2 は 0.5，差分プライバシーの δ は 10^{-5} である．CNN の実装は PyTorch[*3]，DP-SGD の実装は Opacus[*4]を利用した．この設定で，文献 [11] に基づいて算出された DP-SGD で保証される ϵ の理論値は 25.63 である．

4.2　実験結果

RQ1：決定的な NN よりベイジアン NN はプライバシーリスクが高いか？　決定的な NN とベイジアン NN のプライバシーリスクを比較する．具体的には，決定的な NN の出力をアルゴリズム 1 で判定する方法 (Deterministic)，ベイジアン NN の出力として平均と標準偏差が得られる状況でアルゴリズム 2 で判定する方法 (Mean std)，ベイジアン NN の出力としてすべてのサンプリング結果が得られる状況で最小値と最大値を用いる方法 (Min max)，すべてのサンプリング結果が得られる状況で平均，標準偏差，最小値，最大値を用いる方法 (Min max mean std) のプライバシーリスクを算出した．MC dropout のサンプリングの回数は十分な回数として 1,000 回とした．各データセットでの評価結果を図 5 に示す．

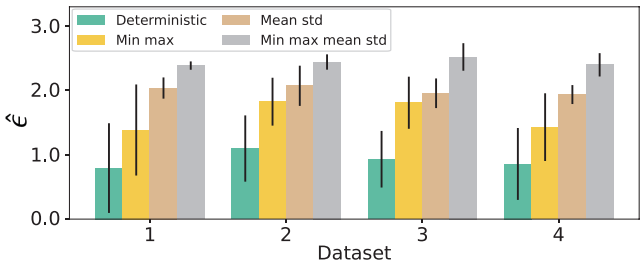


図 5: データセットごとのプライバシーリスク

決定的な NN と比較して，ベイジアン NN はデータセットと使用する情報に依存せずにリスクが高い傾向がある．決定的な NN と比較した際に最もリスクが高くなったのは，2 つの情報を使用した場合は，データセット 1 で平均と標準偏差を使用した場合に 2.6 倍，4 つの情報を使用する場合は，データセット 1 で 3.0 倍となった．これらの結果から，**ベイジアン NN は決定的な NN に比べてプライバシーリスクが高い**ことが分かる．

ベイジアン NN のプライバシーリスクを比較すると，Mean std の方が Min max より高く，標準偏差が小さい傾向がある．これは，Min max はサンプリング結果によってどの程度強い攻撃が実現できるかにばらつきがある一方，Mean std では安定して強い攻撃が実現できていたからであった．ただし，4 つの情報を使用した場合と比べると，Mean std の方がリスクが低いため，サンプリング結果をすべて公開するのではなく，平均と標準偏差のみを公開することでプライバシーリスクを低減することができる．

RQ2：サンプリングの回数はプライバシーリスクに影響があるか？　サンプル回数は判定に使用する情報の信頼性に影響すると考えられるため，サンプリング回数のプライバシーリスクへの影響を調査した．まず，サンプリング回数と情報の信頼性の関係を調べるために，サンプリング結果の平均値を用いた場合のテストデータの精度を調査した．サンプリング回数が 1 回では 0.42，2 回で 0.44，5 回で 0.46，10 回で 0.47，50 回で 0.48 となった．決定的な NN の精度は，0.48 であるため，サンプリング回数が 10 回以下だと情報の信頼性が低いことが分かる．

サンプリング回数を 1, 2, 5, 10, 50, 100, 500, 1,000 とし

[*2] https://www.cs.toronto.edu/ kriz/cifar.html
[*3] https://pytorch.org/
[*4] https://opacus.ai/

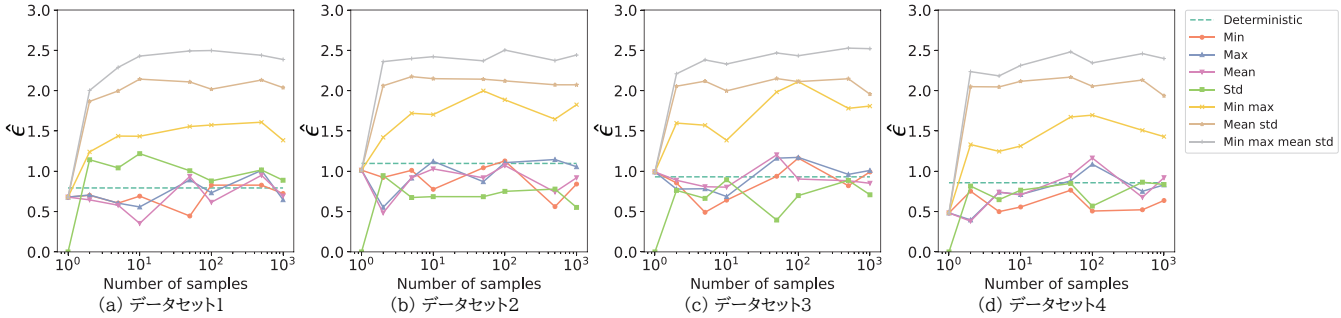
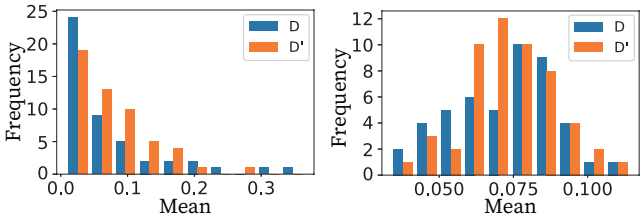
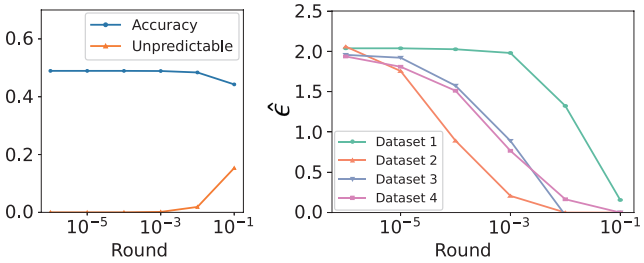


図 6: サンプリング回数のプライバシーリスクへの影響



(a) サンプリング 2 回　　**(b)** サンプリング 1,000 回

図 7: 予測値の平均の分布



(a) 有用性　　**(b)** プライバシーリスク

図 8: 量子化の効果

た時のプライバシーリスクを図 6 に示す．視認性を高めるため，リスクの平均のみを描画し，標準偏差は省いてある．2 つ以上の情報を使用する場合，サンプル数が 1 の時は複数の情報を使用するメリットがないため，決定的な NN と同等のリスクとなった．サンプル数が 2 以上の場合は，決定的な NN よりリスクが高くなっているが，サンプル数との明確な関係は見られなかった．この結果から，**サンプリングが 1 回の場合を除いて，サンプリングの回数とプライバシーリスクに明確な関係はない**と言える．

サンプル数が 2 以上の場合にプライバシーリスクが大きく変化しない理由を分析する．図 6 に示してある 1 つの情報だけ使用した場合の結果を見ると，サンプル数 1 の Std 以外はプライバシーリスクがサンプル数に依存していないことが分かる．標準偏差はサンプル数 1 のとき必ず 0 となるため，攻撃が成功せず外れ値となっている．この結果からも，情報の信頼性が低くても，プライバシーリスクが低くなっていないことが分かる．より詳細に分析するために，サンプリング回数 2 回と 1,000 回の場合の平均値の分布を図 7 に示す．サンプリング回数 2 回の時は，ばらつきが大きく，想定通り情報の信頼性が低いことが分かる．一方で，プライバシーリスクに関して平均値が小さい領域に着目すると，サンプリング回数にかかわらず D の方が D' の頻度より高くなっており，どの程度 D と D' を見分けられるかは同程度であった．これらの結果から，サンプリング回数を減らしても，有用性は低下するが，プライバシーリスクは低下せず，複数の情報を使用する場合は，情報を組み合わせる効果が出るサンプル数 2 以上では，プライバシーリスクが大きく変化しなかったことが分かる．

RQ3：有用性を維持してプライバシーリスクを低減できるか？　RQ1 の結果から，サンプリング結果をすべて出力するよりは，平均と標準偏差を出力した方がプライバシーリスクが低いことは分かっている．そこで，平均と標準偏差を出力する状況でさらにリスクを低減できるか評価する．ベイジアン NN の有用性としては，テストデータでの精度を保つことと，事後分布に影響しないことが重要である．ラベルのみを出力する対策や，softmax の温度パラメータを調整する対策では，事後分布を変化させてしまうため，本稿では，出力値を量子化する対策の効果を調査する．

量子化した平均値と標準偏差を出力する場合の有用性とプライバシーリスクの実験結果を図 8 に示す．有用性としては，テストデータでの精度を調査した．量子化すると，最も高い予測値のクラスと 2 番目に高い予測値のクラスが同一の値となってしまい，ラベルを出力できなくなる場合がある．そこで，予測ラベルを出力できないサンプルの割合と，テストデータでの精度を図 8(a) に示す．各データセットでの結果が重なってしまったため，データセット 1 の結果を示してある．横軸は量子化の単位であり，単位が 0.01 の場合は出力が 0.01 刻みの値となる．出力を 0.01 で量子化した場合でも，精度低下は 1%未満となった．図 8(b) に量子化した場合のプライバシーリスクを示す．量子化することで，平均や標準偏差が同一となることが多くなるため，データセットの区別が難しくなり，プライバシーリスクが低下している．量子化の効果が小さかったデータセット 1 でも，0.01 単位での量子化で，プライバシーリスクが 2.0 から 1.3 に低下している．これらの結果から，**量**

子化により有用性を維持しつつプライバシーリスクを低減できる可能性が高いことが分かる．

5. 議論

より強い攻撃の存在 ベイジアン NN の利用シーンを想定して攻撃を設計したが，さらに強い攻撃を実現できる可能性はある．本稿の目的は，よりタイトなプライバシーリスクの下界を見つけることではなく，ベイジアン NN と決定的な NN のプライバシーリスクを比べることである．そのため，決定的な NN の出力に基づく手法の自然な拡張となるように攻撃を設計した．

理論値との乖離 今回の実験設定では，プライバシーリスクの上界の理論値は 25.63 であった．しかし，下界として算出された値は，高くても 2.5 程度である．算出された値が低すぎるように見えるかもしれないが，先行研究 [10] でも同程度である．DP-SGD はミニバッチごとの勾配を保護しているが，攻撃は最終的なモデルの予測結果を利用しているため，理論値より低くなっていると考えられる．

6. 関連研究

差分プライベートな機械学習 差分プライバシーはデータセットに含まれるデータの安全性を保証するために定義された [14]．この定義に基づき，勾配にノイズを加えて SGD を差分プライベートにする手法は，DNN でも有効であることが確認されている [8]．ベイズ推論に基づく機械学習手法でも，十分統計量にノイズを加える手法 [15] や，事後分布からのサンプリングで差分プライバシーを保証する手法 [16] が提案されている．

DNN のプライバシーリスク DNN の教師データを記憶しやすい特性から，あるデータが教師データに含まれていたかを特定する membership inference 攻撃が可能なことが指摘されている [6]．この membership inference 攻撃を活用して，差分プライバシーで定義されるプライバシーリスクの下界を評価する研究も近年行われる始めている [9, 10]．

ベイジアン NN ベイズ推論は DNN 以前の機械学習でも研究されてきたが，DNN でも様々な手法が提案されている．代表的な手法は，変分推定に基づく手法 [1, 4]，サンプリングに基づく手法 [2]，アンサンブルに基づく手法 [3] である．本稿では，決定的な NN の構造を変えずに容易に導入可能な MC dropout [4] に着目し，NN の使い方を変えた時にプライバシーリスクが増加するかを調査した．

7. おわりに

本稿では，MC dropout を適用した際にプライバシーリスクが増加するかを調査した．そのために，ベイジアン NN の利用シーンに適したプライバシーリスクの評価手法を提案した．CIFAR-10 と CNN を用いた実験で，ベイジアン NN は決定的な NN よりプライバシーリスクが高いこと，サンプリング回数が 2 回以上ではプライバシーリスクに大きな変化がないこと，ベイジアン NN の出力を量子化することで有用性を維持しつつプライバシーリスクを低減できる可能性が高いことを示した．

参考文献

[1] Blundell, C. et al.: Weight uncertainty in neural network, *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1613–1622 (2015).

[2] Korattikara, A. et al.: Bayesian dark knowledge, *arXiv preprint arXiv:1506.04416* (2015).

[3] Lakshminarayanan, B. et al.: Simple and scalable predictive uncertainty estimation using deep ensembles, *arXiv preprint arXiv:1612.01474* (2016).

[4] Gal, Y. and Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning, *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1050–1059 (2016).

[5] Hinton, G. E. et al.: Improving neural networks by preventing co-adaptation of feature detectors, *arXiv preprint arXiv:1207.0580* (2012).

[6] Shokri, R. et al.: Membership inference attacks against machine learning models, *Proceedings of the 38th IEEE Symposium on Security and Privacy*, pp. 3–18 (2017).

[7] Yeom, S. et al.: Privacy risk in machine learning: Analyzing the connection to overfitting, *Proceedings of the 31st IEEE Computer Security Foundations Symposium*, pp. 268–282 (2018).

[8] Abadi, M. et al.: Deep learning with differential privacy, *Proceedings of the 23rd ACM Conference on Computer and Communications Security*, pp. 308–318 (2016).

[9] Jagielski, M. et al.: Auditing differentially private machine learning: How private is private sgd?, *arXiv preprint arXiv:2006.07709* (2020).

[10] Nasr, M. et al.: Adversary instantiation: Lower bounds for differentially private machine learning, *arXiv preprint arXiv:2101.04535* (2021).

[11] Balle, B. et al.: Hypothesis testing interpretations and renyi differential privacy, *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pp. 2496–2506 (2020).

[12] Kairouz, P. et al.: The composition theorem for differential privacy, *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1376–1385 (2015).

[13] Kurakin, A. et al.: Adversarial examples in the physical world, *arXiv preprint arXiv:1607.02533* (2016).

[14] Dwork, C. et al.: Calibrating noise to sensitivity in private sata analysis, *Proceedings of the 3rd Theory of Cryptography Conference*, pp. 265–284 (2006).

[15] Kulkarni, T. et al.: Differentially private Bayesian inference for generalized linear models, *Proceedings of the 38th International Conference on Machine Learning*, pp. 5838–5849 (2021).

[16] Wang, Y.-X. et al.: Privacy for free: Posterior sampling and stochastic gradient monte carlo, *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2493–2502 (2015).