連合学習におけるローカル差分プライバシーメカニズムのハイパーパラメータ調整に関する一考察

前田 若菜 [1]　長谷川 聡 [2]　高橋 翼 [3]

ローカル差分プライバシー (LDP) を満たす連合学習では，モデルの更新情報に対してノイズを加算するメカニズム (LDP メカニズム) を各クライアントに導入する．このとき，LDP メカニズムでは，モデルの更新情報のノルムをある定数以下に制限した上で，この定数 (クリップサイズ) を考慮したノイズの加算が求められる．クリップサイズの設定が不適切な場合，ノイズによって学習が発散する，または学習の進行が阻害される，等の不都合が生じることが知られている．本研究では，LDP 下での連合学習を効率よく進行することを目的として，クリップサイズの設定方法や，適応的に変更していく方法に関して，いくつかの実験的な取り組みについて報告する．

1　はじめに

連合学習 [19] は，複数のクライアントとサーバからなる分散型機械学習である．この方式では，クライアントはサーバの持つグローバルモデルを手元の学習データで更新し，サーバはその更新情報を受け取ってグローバルモデルを更新する．サーバは学習データを受け取らないものの，受信した更新情報から元の学習データを復元できてしまう恐れが文献 [9] で指摘されている．

この対策として差分プライバシー (DP: Differential Privacy) [6] の導入が考えられる．DP とはプライバシー漏洩を制限，定量化するための概念である．DP 下の機械学習では，ノイズを加算するメカニズムを導入してプライバシーを保証する．

クライアント側でノイズを加算するローカル差分プライバシー (LDP: Local Differential Privacy) [13] 下の連合学習では，モデルの更新情報に対してノイズを加算するメカニズム (LDP メカニズム) をクライアントに導入する．このとき，LDP メカニズムでは，モデルの更新情報のノルムをある定数以下に制限した上で，この定数 (クリップサイズ) を考慮したノイズの加算が求められる．クリップサイズの設定が不適切な場合，ノイズによって学習が発散する，または学習の進行が阻害される，等の不都合 [2] が LDP 下の連合学習でも生じると考えられる．

本稿では，LDP 下での連合学習を効率よく進行することを目的として，クリップサイズの設定方法や，適応的に変更する方法に関して，以下のリサーチクエスチョン (RQ) の検証を行う．

- RQ1: クリップサイズの値によって学習はどのように変化するか
- RQ2: クリップサイズを学習途中で減衰していくことで学習を効率化できるか
- RQ3: クリップサイズを適応的に増幅・減衰していくことで学習を効率化できるか

実験結果より，次のことがわかった．

- 学習初期においてはある程度の大きさのクリップサイズを用いることで学習を効率化できること
- 学習が効率よく進んでいる際にクリップサイズを減衰することで，ノイズによる loss の増加を抑制することができ，結果として学習の阻害を防ぐことができること
- 学習初期のクリップサイズが小さい場合，クリップサイズを増幅することで loss の減少を促進し，学習を効率化できること
- LDP を想定していないクリップサイズの設定方針や適応的更新手法は LDP 下の連合学習では有効でない可能性があること

本稿の貢献は，LDP を満たす連合学習におけるクリップサイズの調整に関する知見と，LDP 下の連合学習を想定したクリップサイズの適応的更新手法の必要性を示したことにある．

2　関連研究

連合学習に差分プライバシー (DP) を適用することでプライバシー保護を強化することを目指す研究が進められている [8] [10] [11] [15] [17] [18]．連合学習での DP の適用には，セントラル差分プライバシー (CDP) に基づく方法とローカル差分プライバシー (LDP) に基づく方法が考えられる．CDP に基づく方法は，クライアントから集めた更新情報にサーバでノイズを加算することでプライバシー保護を達成する．LDP に基づく方法は，クライアントで更新情報にノイズを加算することでプライバシー保護を達成する．LDP に基づく方法はサーバに対してもプライバシー保護をするため，更新情報にノイズを加算することから，CDP に基づく方法と比べ，加算されるノイズの総量が大きくなる傾向がある．近年，CDP に基づくアプローチにおいて，セキュアアグリゲーションと組み合わせる方式も提案されている [2] [14] [22]．セキュアアグリゲーションとは個々の入力データを開示せずに集め，そのデータの集約値を計算をするもので，実現方法の一つに Trusted Execution Environment(TEE) [24] に基づくものがある．これは，ハードウェア上に通常の実行環境から分離された安全な実行環境を構成し，処理するデータとプログラムに対して機密性と完全性を保証するものである．

DP を満たす連合学習の手法として，機械学習における DP を満たすための手法である Differentially Private Stochastic Gradient Descent (DP-SGD) [1] アルゴリズムに基づくものや，これを拡張したものがある [2] [10] [17] [18]．DP-SGD では，勾配に対して二つの処理を行う．一つは勾配ノルムをある定数（クリップサイズ）以下になるよう制限する処理（クリッピング）であり，もう

[1] 非会員　LINE ヤフー株式会社
wakana.maeda@lycorp.co.jp
[2] 非会員　LINE ヤフー株式会社
satoshi.hasegawa@lycorp.co.jp
[3] 正会員　LINE ヤフー株式会社
tsubasa.takahashi@lycorp.co.jp

一つはクリップサイズとノイズスケールに基づいて生成されたガウスノイズをクリッピングされた勾配に加算する処理である．このクリップサイズが小さすぎると，勾配に含まれる情報の多くが破棄されるためバイアスが大きくなる可能性がある．一方で，クリップサイズが大きすぎるとガウスノイズの量が大きくなり，学習を阻害する可能性がある．そのため，適切なクリップサイズを設定する必要がある．

では，適切なクリップサイズをどう設定すればいいだろうか．連合学習における先行研究では，更新情報のノルムの分布の中央値を推奨し，クリップサイズを適応的に更新する手法の提案 [2] がある．ただし，これは CDP 下を想定しており，LDP 下でも有効かはわからない．LDP を満たす連合学習における先行研究 [8] [11] [15] においては，クリップサイズと学習の関係については議論されておらず，クリップサイズの適応的更新手法も提案されていない．一方，連合学習を想定していない先行研究においては，クリップサイズを途中で小さくすることでモデルの精度が向上した報告 [20] や，勾配ノルムの分布の中央値を推奨するもの [1] がある．これらは，LDP 下の連合学習においても有効かはわからない．さらに，クリップサイズを適応的に更新する手法の提案 [12] [23] [25] などがあるものの，LDP 下の連合学習にはそのまま適用できない．

本研究では，文献 [20] で報告されたようなクリップサイズの減衰による学習の効率化が LDP 下の連合学習でも有効かを検証する．さらに，文献 [2] のクリップサイズの適応的更新手法や文献 [1] が推奨する勾配ノルムの分布の中央値をクリップサイズに用いる方針が LDP 下の連合学習でも有効かを検証する．

3 準備

初めに差分プライバシーを定義し，次に連合学習と差分プライバシーを適用した連合学習を説明する．最後に，クリップサイズの適応的更新手法を紹介する．

3.1 差分プライバシー

差分プライバシー (DP) の 2 つのプライバシーモデルを定義したのち，ノイズの設計とプライバシー消費の管理について述べる．

3.1.1 プライバシーモデル

DP [6] の二つのプライバシーモデルを定義する．一つはセントラル差分プライバシー (CDP) で，サーバが収集したデータから算出した統計量を公開する際に，統計量に対してノイズを加えることでプライバシーを保護する．もう一方はローカル差分プライバシー (LDP) [13] で，サーバに送信するデータにノイズを加えることでプライバシーを保護する．これらを以下のように定義する．

定義 1. ((ϵ,δ)-CDP [6]). $\epsilon \geq 0$ と $0 \leqq \delta < 1$ が与えられたとき，あるランダム化メカニズム $\mathcal{M}: \mathcal{D} \to \mathcal{S}$ が (ϵ,δ)-CDP を満たすとは，あらゆる二つの隣接するデータベース $D, D' \in \mathcal{D}$，および任意の出力の部分集合 $S \subseteq \mathcal{S}$ について，次式が成り立つときである．

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

定義 2. ((ϵ,δ)-LDP [13]). $\epsilon \geq 0$ と $0 \leqq \delta < 1$ が与えられたとき，あるランダム化メカニズム $\mathcal{M}: \mathcal{X} \to \mathcal{S}$ が (ϵ,δ)-LDP を満たすとは，あらゆる入力の組 $x, x' \in \mathcal{X}$，および任意の出力の部分集合 $S \subseteq \mathcal{S}$ について，次式が成り立つときである．

$$\Pr[\mathcal{M}(x) \in S] \leqq e^{\epsilon} \cdot \Pr[\mathcal{M}(x') \in S] + \delta \tag{2}$$

LDP は，直感的に説明すれば，メカニズム $\mathcal{M}$ に x, x' を入力したとしてもそれら出力がどちらの入力から得られたものか識別できないことを保証している．そのため，LDP を満たすメカニズムの出力の元である入力が何であったかを推測することは難しい．

3.1.2 ノイズの設計

(ϵ,δ)-CDP を満たすメカニズムとしてガウスメカニズムがある．ガウスメカニズムでは，平均 0，分散 $\Delta_f^2 \cdot \sigma^2$ で設計されたガウス分布 $\mathcal{N}(0, \Delta_f^2 \cdot \sigma^2)$ からノイズをサンプリングし，f の出力に加算する．σ はノイズスケールであり，$\epsilon, \delta \in (0,1)$ が与えられたとき，$\sigma = \sqrt{2\log(1.25/\delta)}/\epsilon$ であるガウスメカニズムは (ϵ, δ)-CDP を満たす [4]．なお，$\epsilon > 1$ を含む一般的なケースは [4] を参照されたい．Δ_f は以下に定義するセンシティビティである．

定義 3. (センシティビティ). あらゆる二つの隣接するデータベース $D, D' \in \mathcal{D}$ に対する関数 f のセンシティビティ Δ_f は以下のように表される．

$$\Delta_f = \sup_{D,D'} ||f(D) - f(D')||_{\mathrm{p}} \tag{3}$$

ガウスメカニズムを用いて (ϵ,δ)-LDP を満たす場合には，CDP の 2 倍のセンシティビティを用いる必要がある[*1]．

3.1.3 プライバシー消費の管理

DP では，メカニズムを通してデータにアクセスする度にプライバシーを消費すると考える．そのため，全体を通してどのくらいのプライバシー消費があったかを求めるために，消費したプライバシーを合算する必要がある．

定理 1. (直列合成定理 [6] [7]). メカニズム列 $\{M_i\}_{i=1}^{K}$ がそれぞれ (ϵ_i, δ_i)-DP を満たすとき，メカニズム列は $(\Sigma_{i=1}^{K}\epsilon_i, \Sigma_{i=1}^{K}\delta_i)$-DP を満たす．

なお，直列合成定理は最も単純な合算方法であり，より厳格に消費を見積もる方法が研究されている [3] [21].

3.2 連合学習

本稿は連合学習のうち Federated stochastic gradient descent(FedSGD) [19] に関して議論する．この方式のアルゴリズムを Algorithm 1，2 に示す．サーバ側は t 回目の学習において，m クライアントからなる集合 S_t を取得し，それらにモデルのパラ

[*1] これは，CDP と LDP 間で隣接データベースの考え方が異なることに起因している．CDP では隣接データベースの概念を，追加と削除の処理による違いでデータベース間の距離を考える．そのため，一つの削除あるいは追加によるデータベース間の差分は 1 距離である．一方，LDP における隣接データベースの概念は，置換の処理による違いで距離を考える．置換の処理は，削除し新たに追加する 2 処理とみなすことができるため，LDP における隣接データベースの 1 距離は，CDP の観点では 2 距離である．したがって，CDP を想定して考えられたメカニズムを LDP に適用する場合，CDP の 2 倍のセンシティビティを用いる必要がある．

Algorithm 1: FedSGD; server-side

Input: learning rate η

Initialize model θ_0

for each round $t = 0, 1, \ldots$ **do**

  $S_t \leftarrow$ (sample m clients randomly)

  for each client $k \in S_t$ **do**

    $g_t^{(k)} \leftarrow \text{ClientUpdate}(\theta_t)$

  end

  $\theta_{t+1} \leftarrow \theta_t - \eta \frac{1}{m} \sum_{k \in S_t} g_t^{(k)}$

end

Algorithm 2: FedSGD; client-side

Require: training data B, loss function ℓ

Function `ClientUpdate(`θ`)`**:**

  $g \leftarrow \nabla \ell(B; \theta)$

  return g

メータ θ_t を送信する．クライアント $k \in S_t$ は，受信したパラメータ θ_t と手元の n 個の d 次元の実数ベクトルからなる学習データ $B = \{x_i\}_{i=1}^n$ ($x_i \in \mathbb{R}^d$) を用いて loss 関数 ℓ の勾配 $g_t^{(k)}$ を計算し，これをサーバへ送信する．サーバは，クライアント集合 S_t から得た勾配から平均値 $\frac{1}{m}\Sigma_{k \in S_t} g_t^{(k)}$ を計算し，それに学習率 η を乗算したものを使ってパラメータを θ_{t+1} へ更新する．

3.3 ローカル差分プライバシーを満たす連合学習

CDP を満たす機械学習にて提案された Differentially private stochastic gradient descent（DP-SGD) [1] を LDP を満たす連合学習向けに拡張する．この拡張したものを Federated LDP-SGD と呼ぶことにする．Federated LDP-SGD は，クライアントの勾配の計算にガウスメカニズムを適用することで DP を満たすことを基本とする．ただし，一般的に勾配は $[0, \infty)$ の値を取りうるため，センシティビティが無限になってしまうことから，勾配をクリップサイズ C 以下に制限 (クリッピング) することでセンシティビティを C に抑える．

アルゴリズムは，サーバ側は通常の連合学習と同様に Algorithm 1，クライアント側は Algorithm 3 に示したものになる．クライアントは勾配 g を計算した後，勾配 g の ℓ_2 ノルムが最大でもクリップサイズ C になるようにクリッピングを行い，クリッピングされた勾配 $\bar{g}$ を得る．次に，クリッピングされた勾配 $\bar{g}$ にガウスノイズを加算して勾配のランダム化を行い，勾配 $\tilde{g}$ を得る．このとき，ガウスノイズのスケールは，クリップサイズ C とノイズスケール σ を用いて，$C\sigma$ とする．クライアントはランダム化された勾配 $\tilde{g}$ をサーバに送信する．

3.4 クリップサイズの適応的更新

まず，DP を適用しない時の adaptive quantile clipping [2] を示し，その後 CDP の満たし方を示す．この手法では，クリップサイズ C を更新情報 (FedSGD の場合は勾配) のノルムの分布の γ パーセンタイルに収束するよう更新する．例えば，5 つの更新情報のノルムが $1, 2, 3, 4, 5$ で $\gamma = 0.5$ の時，$C = 3$ に収束するよう

Algorithm 3: Federated LDP-SGD; client-side

Require: noise scale σ, training data B, clipping size C, loss function ℓ

Function `ClientUpdate(`θ`)`**:**

  $g \leftarrow \nabla \ell(B; \theta)$

  $\bar{g} \leftarrow g \cdot \min(1, \frac{C}{||g||_2})$ ▷ Clip gradient

  $\tilde{g} \leftarrow \bar{g} + \mathcal{N}(0, (C\sigma)^2 I)$. ▷ Add noise

  return Noisy gradient $\tilde{g}$

更新する．

サーバはクリップサイズの学習率 η_C とクリップサイズとして指定したい更新情報のノルムの分布のパーセンタイル $\gamma \in [0, 1]$ を指定する．中央値を指定するのであれば，$\gamma = 0.5$ とする．学習 t 回目で抽出されたクライアント $k \in S_t$ は，更新情報に加えて，更新情報のノルムをクリップしたか否かを表す $b_t^{(k)}$ を送信する．クリップしてないならば $b_t^{(k)} = 1$，クリップしたならば $b_t^{(k)} = 0$ とする．サーバは各クライアントから $b_t^{(k)}$ を収集し，この総和 $\Sigma_{k \in S_t} b_t^{(k)}$ を抽出されたクライアント数 $|S_t|$ で割ることで，クリップされていない割合 $\bar{b_t}$ を求める．そして，次式のようにクリップサイズ C を更新する．

$$C_{t+1} = C_t \cdot \exp(-\eta_C(\bar{b_t} - \gamma)) \tag{4}$$

なお，CDP を満たすようにする場合は，$\bar{b_t}$ の計算時にノイズを加算する．

4 実験

本研究では，ローカル差分プライバシー (LDP) を満たす連合学習において適切なクリップサイズを設定できるようにするために以下のリサーチクエスチョン (RQ) の検証を行う．

- RQ1: クリップサイズの値によって学習はどのように変化するか（実験 1: クリップサイズと学習との関係の探索的実験）
- RQ2: クリップサイズを学習途中で減衰していくことで学習を効率化できるか（実験 2: クリップサイズの単純減少実験）
- RQ3: クリップサイズを適応的に増幅・減衰していくことで学習を効率化できるか（実験 3: クリップサイズの適応的更新実験）

以下，基本的な実験設定とプライバシーモデルを説明した後，各実験の詳細設定と結果を述べる．

4.1 基本的な実験設定

データセット 実験データとして手書き数字のデータセット MNIST [5] を用いる．これは数字 0〜9 に分類する 10 クラス分類タスクである．クライアントが持つ訓練データの数をクライアントによらず一律でそれぞれ 5 つとし，テストデータは 1 つとし，これらを復元抽出によって得る．

学習モデル CNN モデルを用い，そのアーキテクチャは文献 [8] の TABLE XIII を参考にした．モデルの精度評価として，分類精度 (accuracy) を用いる．

パラメータ 全クライアント数を 10,000,000 とし，モデルの

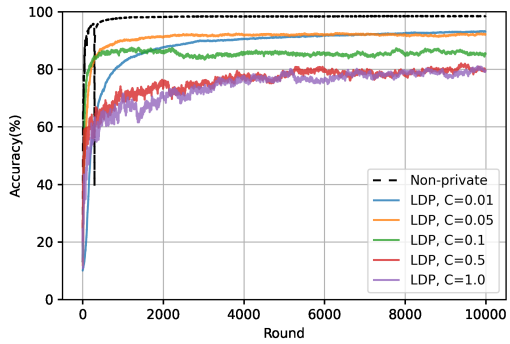

図 1　クリップサイズ C における分類精度の推移

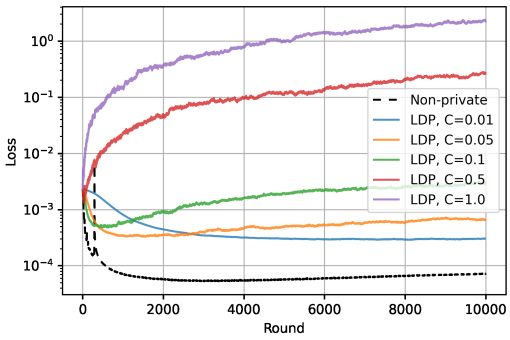

図 2　クリップサイズ C における loss の推移

更新 1 回につき参加するクライアント数を 1,000 とする．また，モデルの更新回数 (round) を 10,000 とする．学習率は DP 非適用時 (non-private) と LDP 下でそれぞれ 0.1，1 とする．プライバシーパラメータは，(ϵ, δ)=$(8, 10^{-7})$ とする．

4.2　プライバシーモデル

実験 1，2 と実験 3 ではプライバシーモデルがやや異なる．実験 1，2 では LDP モデルであり，実験 3 では LDP モデルとセントラル差分プライバシー (CDP) モデルの混合型となっている．

プライバシーモデルが異なる理由は，クライアントから取得する情報の違いにある．実験 1，2 ではクライアントから勾配のみを取得するが，実験 3 ではクリップサイズの適応的更新をするために必要な補助情報を追加で取得する．

補助情報取得に対して，文献 [2] に合わせて，CDP モデルを適用する．なお，実験 3 でも実験 1，2 と同様に勾配には LDP モデルを適用する．

4.3　実験 1：クリップサイズと学習との関係の探索的実験

実験 1 ではクリップサイズ C={0.01, 0.05, 0.1, 0.5, 1.0} によって分類精度や loss がどのように変化するかを観測することで学習への影響を調査する．図 1，2 はクリップサイズ C を変えた時の分類精度と loss の推移を示したものである．黒の波線が non-private の結果で，残りが LDP の結果である．

図 1 では，C = 0.01，0.05，0.1 の順で non-private の分類精度に近い値となっている．ただし，これらのうち学習初期においては，C = 0.01 は分類精度の向上が遅く，最も早いのは C = 0.05 だとわかる．C = 0.5，1.0 については，いずれも同じような推移をしており，学習全体を通して分類精度が低かった．

図 2 では，C = 0.01 は loss の減少が進んでいる．一方で，C = 0.05，0.1 については loss が途中で増加に転換している．転換の早さは C = 0.1 の方が C = 0.05 より早い．残りの C = 0.5, 1.0 については学習初期から loss の増加が続いており，うまく学習できていないことがわかる．

クリップサイズについて以下のことがわかった．

- 学習初期においてはクリップサイズが小さいと学習の進みが遅く，クリップサイズがある程度の大きさだと学習の進みが早い．
- クリップサイズが小さいと学習が進んでも loss の増加がなく，分類精度が高い．
- ある程度の大きさのクリップサイズでは学習初期から，もしくは，学習途中から loss が増加し学習がうまく進まない．

4.4　実験 2：クリップサイズの単純減少実験

実験 1 では，クリップサイズの大きさによって学習初期とその後で分類精度，loss の推移に違いがあることがわかった．学習初期の学習効率を高めつつその後も継続して学習をうまく進める方法として，クリップサイズを学習初期はある程度の大きさにし，その後にクリップサイズを減衰する方法が考えられる．

実験 2 では，クリップサイズを学習途中で減衰していくことで継続的に学習がうまく進行するかを検証する．

4.4.1　クリップサイズ調整方法

クリップサイズを固定にする方法と更新する方法を比較する．更新方式は 2 種類あり，一つは学習の一時点でのみクリップサイズを小さくする方式 (一時点切替) と，もう一つは学習の度にクリップサイズを減衰する方式 (減衰) である．

ベースライン　クリップサイズは固定にし，値として 0.01, 0.05 を用いる．それぞれを clip0.01, clip0.05 とする．

一時点切替　初期値として実験 1 で初期の分類精度が高かったクリップサイズ 0.05 を用い，最終的な分類精度が高かったクリップサイズ 0.01 に途中で切り替える方法を考える．切り替えるタイミングとして，クリップサイズ 0.05 と 0.01 の loss の高低が逆転した 2,000 回目と分類精度の高低が逆転した 6,000 回目を用いる．2,000 回目と 6,000 回目に切り替える方法をそれぞれ 0.05to0.01(2000), 0.05to0.01(6000) とする．

減衰　クリップサイズの減衰方法として，学習率減衰方法の poly learning rate policy [16] [26] を次式のように応用する．

$$C_{t+1} = C_0 \times (1 - \frac{t}{T})^{power} \tag{5}$$

C_t は t 回目のクリップサイズ，T は最大ラウンド数である．C_0 = {0.01, 0.05}, $power$ = {0.5, 1.0, 2.0} を用いる．それぞれを初期値と $power$ を使って poly(0.01)0.5 のように表記する．図 3 はクリップサイズの減衰の様子を示したもので，$power$ = 0.5, 2.0 はラウンドが進むごとに減衰がそれぞれ，大きくなる，小さくなる．一方、$power$ = 1.0 はラウンドに関係なく一定で減衰する．

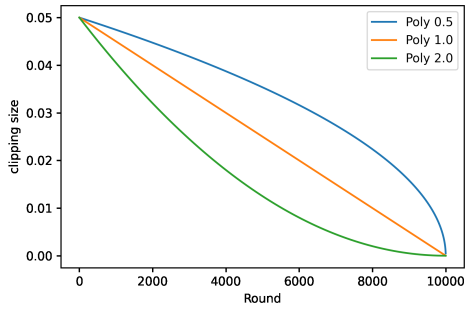

図 3　クリップサイズ C の減衰

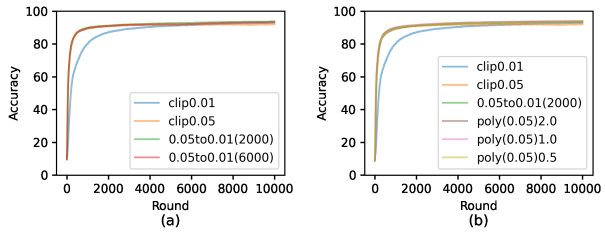

図 4　クリップサイズ調整方法と分類精度の推移

表 1　ベースラインと一時点切替の分類精度

	clip0.01	clip0.05	0.05to0.01 (2000)	0.05to0.01 (6000)
Mean	92.98	92.03	93.75	93.35
Std	0.34	0.33	0.09	0.17

表 2　減衰の分類精度

	poly(C_0)0.5	poly(C_0)1.0	pol(C_0)2.0
$C_0 = 0.01$	92.16 ± 0.20	91.40 ± 0.29	89.65 ± 0.20
$C_0 = 0.05$	93.86 ± 0.08	94.04 ± 0.24	93.56 ± 0.11

4.4.2　実験結果

クリップサイズ調整方法ごとの分類精度の推移を図 4 に示す．図 4(a) が示すように，クリップサイズの初期値を 0.05 とし，学習途中で 0.01 に切り替えることで，学習初期とその後の両方で分類精度が向上した．図 4(b) が示すように，クリップサイズの初期値を 0.05 とし，毎回減衰させる方法でも同様の結果が得られた．

クリップサイズ調整方法ごとの最大ラウンド時の分類精度の 5 試行の平均 (mean) と標準偏差 (std) を表 1，2 に示す．初期値 0.01 について，固定方式の clip0.01 と比較すると，減衰方式はいずれも分類精度は下回った．一方，初期値 0.05 について，固定方式の clip0.05 と比較すると，クリップサイズを更新した方式全ての分類精度が上回った．これらの結果からある程度小さいクリップサイズを減衰していくことは分類精度の向上に有効でなく，ある程度大きいクリップサイズを減衰していくことは分類精度の向上に有効であることがわかる．

クリップサイズ調整方法ごとの loss を図 5 に示す．図 5(a) をみると，クリップサイズを切り替えることで増加していた loss が減少に転換していることがわかる．次に，図 5(b) をみると，loss の減少が clip0.01 よりも減衰方法の方が遅く，学習の効率が落ちていることがわかる．最後に，図 5(c) をみると，減衰方法では poly(0.05)1.0，poly(0.05)0.5 が 0.05to0.01(2000) よりも loss が大きく，poly(0.05)2.0 が 0.05to0.01(2000) よりも僅かに loss が小さかった．これら結果から，クリップサイズの初期値が小さい際にクリップサイズを減少させると学習が遅くなること，クリップサイズの初期値がある程度の大きさの際に更新回数を大きくしてもクリップサイズの更新が適切でないと loss の減少への影響は大きくならないことがわかる．

これらの実験 2 の結果から，次のことがわかった

- ある程度小さいクリップサイズを減衰していくことは loss の減少の妨げになり，学習の進みを遅くすること
- ある程度大きいクリップサイズを減衰していくことは loss の減少に効果があり，学習の効率化につながること
- 更新回数を多くしてもクリップサイズの更新が適切でないと loss の減少効果は大きくならないこと

4.5　実験 3：クリップサイズの適応的更新実験

実験 2 では，初期のクリップサイズがある程度の大きさならば学習が進むにつれクリップサイズを適宜小さくすることでより分類精度を向上できるとわかった．一方で，初期のクリップサイズが小さい場合には逆効果であることもわかった．そのため，初期のクリップサイズが小さければクリップサイズを大きくするという更新方法も考える必要がある．

実験 3 では，クリップサイズを適応的に増幅・減衰といった更新をすることで学習を効率化できるかを検証する．

クリップサイズを適応的に更新するには，クライアントから補助情報を取得する必要がある．補助情報の取得に対して，文献 [2] に合わせて CDP モデルを適用する．

4.5.1　クリップサイズ調整方法

クリップサイズを固定にする方法として，実験 2 と同様に clip0.01，clip0.05 を使用する．

適応的更新方法として，adaptive quantile clipping [2] と，ヒストグラムから勾配ノルムの分布の中央値を推定してクリップサイズを更新する手法 adaptive median clipping を用いる．

Adaptive quantile clipping [2] では，デフォルト値のノイズスケール $\sigma = 5$ とクリップサイズの学習率 $\eta_C = 0.2$ を用いる．初期値 $C_0 = 0.01$，パーセンタイル $\gamma = 0.1, 0.5$ を用い，それぞれ quantile(0.01)10%，quantile(0.01)50% とする．クリップサイズは毎回更新を行い，計算過程にはガウスノイズを加算し，これは $(0.006, 10^{-7})$-CDP を満たす[*2]．

Adaptive median clipping は次節で説明する．

4.5.2　Adaptive median clipping

文献 [1] のクリップサイズの設定方針に従い，adaptive median clipping では勾配ノルムの分布の中央値へクリップサイズを更新

[*2] Rényi differential privacy による合成から計算した [3].

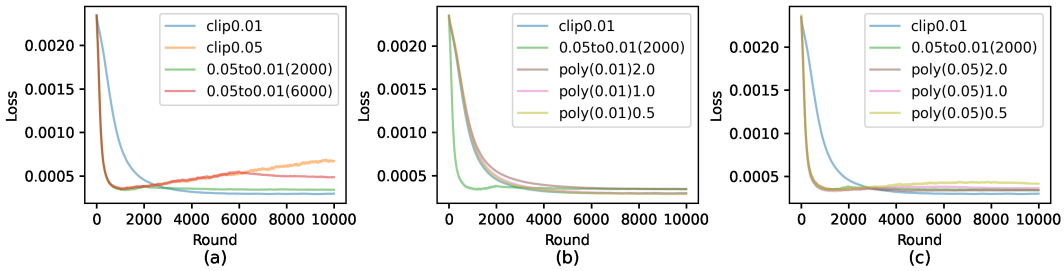

図 5　クリップサイズ調整方法と loss の推移

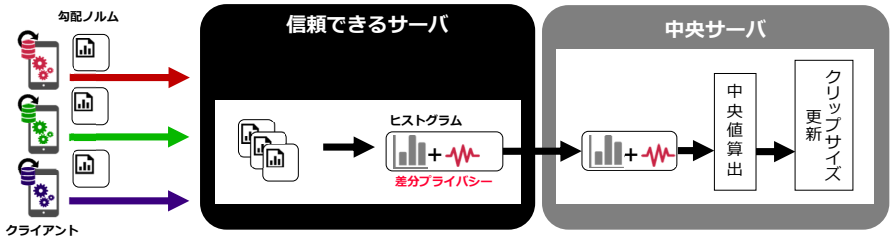

図 6　Adaptive median clipping の構成

する．手法の構成を図 6 に示す．まず，信頼できるサーバにてクライアントから勾配ノルムを集約し，CDP を満たすヒストグラムを生成する．この時，ヒストグラムのビンは事前に設定しておく．サーバは CDP を満たすヒストグラムを受け取り，これに基づいて勾配ノルムの分布の中央値を算出する．そして，得られた中央値へクリップサイズを更新する．なお，補助情報の集約は Trusted Execution Environment(TEE) [24] などを用いて実装が可能である．

実験の設定を記述する．ヒストグラムのビンとして，$\{[0, 2^{-7}], ..., [2^{-4}, 2^{-3}]\}$ の範囲をもつ 5 つのビンを設定した．2^{-3} より大きい値は全て 2^{-3} の値としてトップコーディングする．ビンの設定は実験 1 を踏まえており，一番小さい値のビンの中央値が 0.01 の約 1/10，一番大きい値のビンの中央値が約 0.1 になるように設定してある．勾配ノルムの分布の中央値として累積度数が 50% を超えたビンの中央値を用いる．ヒストグラムのランダム化メカニズムは $(0.8, 10^{-8})$-CDP を満たすとし，クリップサイズの更新は 1000 回ごとに行う．クリップサイズの初期値を 0.01, 0.05 とし，それぞれ adaptive0.01, adaptive0.05 と表記する．なお，本実験では TEE を使用していない．

4.5.3　実験結果

クリップサイズ調整方法ごとの分類精度を図 7 に，loss を図 8 に示す．図 7 にあるように，adaptive0.01 は学習初期の分類精度が clip0.01 や clip0.05 よりも向上しているものの，学習が進むにつれ adaptive 方式の分類精度は clip 方式よりも下回った．これら分類精度は，学習初期の最高値に比べて低くなっている．また，quantile 方式については学習がうまくいかなかった．図 8 では，adaptive0.01 は学習初期の loss の減少が clip0.01 と比べて早くなっているものの，途中から loss が増加に転換し，学習が阻害されていることを示している．一方，adaptive0.05 は clip0.05 よりも loss が増加しており学習がうまく進んでいない．さらに，quantile 方式では loss が早い段階で発散していた．なお，quantile 方式では 50, 10 のいずれのパーセンタイルでもクリップサイズが増加し続けていた．

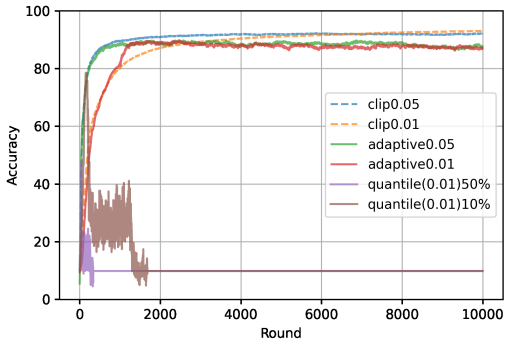

図 7　クリップサイズ調整方法ごとの分類精度の推移

図 9 に adaptive0.01 における各更新ごとの勾配ノルムのヒストグラムを示す．頻度へのノイズ加算前のものが raw，加算後が noisy とする．ノイズ加算前後でヒストグラムに大きな違いはなく，最も大きな勾配ノルムの値をもつ一番右のビンの頻度が最大かつ過半数を超えていた．このため，adaptive 方式ではクリップサイズは初回の更新から最後まで 0.1 に更新されていた．学習が進むにつれ，一番右のビン $[2^{-3}, 2^{-4}]$ の頻度は減少し，一番左のビン $[0, 2^{-7}]$ が増加しつつあるも，それでも一番右のビンが最も頻度の高いビンとなっている．そのため，勾配ノルムの分布の中

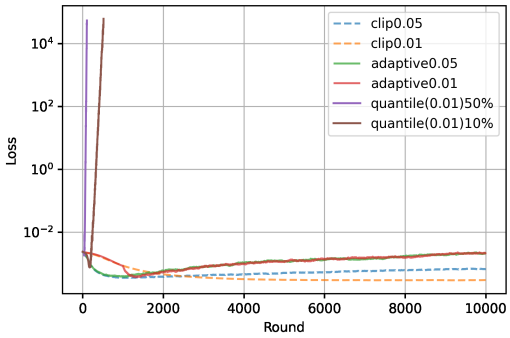

図 8　クリップサイズ調整方法ごとの loss の推移

央値を使ったクリップサイズの更新ではクリップサイズが大きくなりやすい可能性がある．

これらの結果から，学習初期のクリップサイズが小さい場合，クリップサイズを大きくすることで初期の学習を効率化可能だとわかった．ただし，その後も適切にクリップサイズを更新しなければ学習が阻害され，分類精度が学習初期よりも低下する可能性があることがわかった．

5　考察

実験 1 では，クリップサイズの大きさによって学習初期とその後で分類精度，loss の推移に違いがあることがわかった．これは，学習初期においてはクリッピングの影響が大きく，学習が進むにつれてその影響が小さくなり，ノイズの影響が相対的に大きくなるからだと考えられる．各クリップサイズ調整方式におけるクリッピングとノイズ加算前の勾配ノルムの平均値の推移を図 10 に示す．non-private の勾配ノルムは学習が進むにつれ減少する一方で，LDP の勾配ノルムは値が大きく，クリップサイズの変化がない clip0.01, 0.5 はその値が増幅し続けている．これは，学習が進むにつれてノイズの影響が大きくなり，学習がうまく進まないのを補うように勾配ノルムが増幅しているからだと考えられる．クリップサイズを途中で切り替える 0.05to0.01(2000) 方式では勾配ノルムの増幅を抑えることができており，このことからクリップサイズを調整することで勾配ノルムの増幅を抑えることができることがわかる．

実験 2 で確認した通り，学習初期においてはクリップサイズを大きくすることでクリッピングの影響を小さくして学習初期の進行を早くし，適宜クリップサイズを小さくすることでその後のノイズの影響を小さくすることができると考えられる．ただし，学習初期の進行が遅くなるようなクリップサイズの時にこれを小さくすることは，より学習の進行を遅くし，分類精度の低下をまねくことになる．このことから，クリップサイズを小さくする更新だけでなく，クリップサイズが小さい場合には大きくする更新が必要なことがわかる．

実験 3 では，学習初期のクリップサイズが小さい場合，これを大きくすることで初期の学習を効率化可能だとわかった．ただし，その後も適切にクリップサイズを更新しなければ学習が阻害されることがわかった．今回用いたクリップサイズの設定方針 [1] と適応的更新手法 [2] は，それぞれセントラル差分プライバシー (CDP) 下の機械学習，及び，連合学習で有効とされていたものだが，いずれもローカル差分プライバシー (LDP) 下の連合学習にとっては大きいクリップサイズへ更新していた．その結果，ノイズの影響が大きくなり学習が阻害されたのだと考えられる．LDP 下の連合学習は CDP 下の学習と異なり，各クライアントの勾配にノイズを加算する必要がある．それゆえ，ノイズの影響を小さくするためには，CDP 下の学習よりも小さいクリップサイズに設定する必要があると考えられる．したがって，LDP 下の連合学習においては CDP 下と比べて小さいクリップサイズになるように設計された専用の適応的更新手法が必要だといえる．

6　おわりに

本研究では，ローカル差分プライバシー (LDP) 下の連合学習を効率よく進行することを目的とし，クリップサイズの設定方法や，適応的に更新する方法に関して，いくつかの実験的な取り組みを行なった．実験から，学習初期から継続的に学習を効率よく学習を進めるためには，初期のクリップサイズはある程度大きくし，学習が進むにつれてクリップサイズを減衰することが有効であることがわかった．また，学習初期のクリップサイズが小さい場合，クリップサイズを増幅することで初期の学習を効率化可能なことがわかった．そして，LDP 下を想定していないクリップサイズの設定方針や適応的更新手法は LDP 下の連合学習では有効ではない可能性を示した．

今後の課題として，複数のデータセットを用いて今回の実験結果の一般性を確かめることと，LDP を満たす連合学習向けのクリップサイズの適応的更新手法を確立することがある．

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*, pages 308–318, 2016.

[2] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34:17455–17466, 2021.

[3] Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and renyi differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 2496–2506. PMLR, 2020.

[4] Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pages 394–403. PMLR, 2018.

[5] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[6] Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, ICALP'06, page 1–12, Berlin, Heidelberg, 2006. Springer-Verlag.

[7] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[8] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evalua-

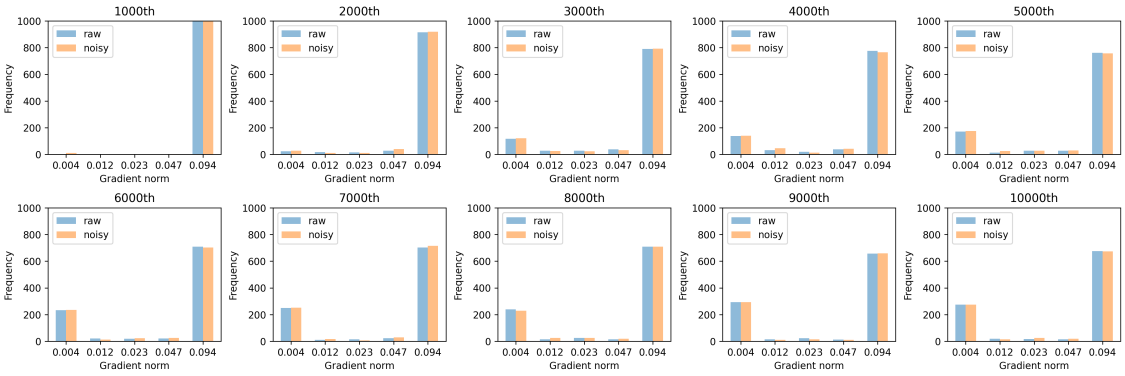

図 9　各更新ごとの勾配ノルムのヒストグラム

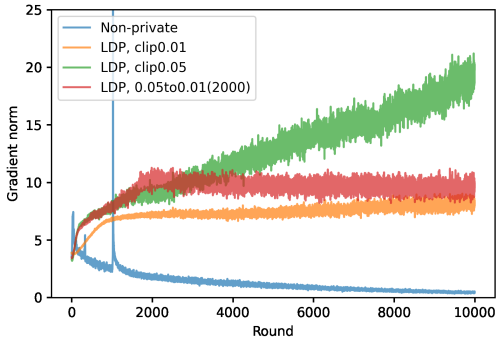

図 10　各クリップサイズ調整方式における加工前の勾配ノルムの平均値の推移

tion. *arXiv preprint arXiv:2001.03618*, 2020.

[9] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

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

[11] Antonious M Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of federated learning: Privacy, accuracy and communication trade-offs. *IEEE journal on selected areas in information theory*, 2(1):464–478, 2021.

[12] Aditya Golatkar, Alessandro Achille, Yu-Xiang Wang, Aaron Roth, Michael Kearns, and Stefano Soatto. Mixed differential privacy in computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8376–8386, 2022.

[13] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[14] Fumiyuki Kato, Yang Cao, and Masatoshi Yoshikawa. Olive: Oblivious and differentially private federated learning on trusted execution environment. *arXiv preprint arXiv:2202.07165*, 2022.

[15] Muah Kim, Onur Günlü, and Rafael F Schaefer. Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2650–2654. IEEE, 2021.

[16] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[17] Mohammad Malekzadeh, Burak Hasircioglu, Nitish Mital, Kunal Katarya, Mehmet Emre Ozfatura, and Deniz Gündüz. Dopamine: Differentially private federated learning on medical data. *arXiv preprint arXiv:2101.11693*, 2021.

[18] Virendra J Marathe and Pallika Kanani. Subject granular differential privacy in federated learning. *arXiv preprint arXiv:2206.03617*, 2022.

[19] 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.

[20] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

[21] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[22] John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. Federated learning with buffered asynchronous aggregation. In *International Conference on Artificial Intelligence and Statistics*, pages 3581–3607. PMLR, 2022.

[23] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X Yu, Sashank J Reddi, and Sanjiv Kumar. Adaclip: Adaptive clipping for private sgd. *arXiv preprint arXiv:1908.07643*, 2019.

[24] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64. IEEE, 2015.

[25] Jun Wang and Zhi-Hua Zhou. Differentially private learning with small public data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6219–6226, 2020.

[26] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.