2J-03

準同型暗号を用いた畳み込みニューラルネットワークの並列処理

Parallization of Convolutional Neural Network Using Homomorphic Encryption

光永 茉弥 †　　吉田 明正 †
Mahiro Mitsunaga　　Akimasa Yoshida

1　はじめに

近年，クラウドサーバで機械学習モデルの学習・推論が広く行われているが，機密データを取り扱う場合，プライバシーや情報漏洩の観点から，準同型暗号を用いた推論が期待されている．準同型暗号とは，暗号化したデータに対し復号化せずに加算や乗算の処理を行うことができる技術である．ユーザーが暗号化したデータでクラウドサービスを利用し，そこで得られた演算結果に対してそのユーザーのみが秘密鍵を使用して復号化を行うことが可能である．しかしながら，暗号化したデータは暗号化されていないデータよりもデータサイズが大きく，演算に時間を要するため，実用化に向けて処理速度の向上が求められる．

準同型暗号を用いた関連研究として，完全準同型暗号ライブラリ HElib 上での行列積の高速化 [1], 準同型暗号と隔離実行環境（TEE）を組み合わせ実行性能比較 [2], MNIST を用いた画像推論に対して準同型暗号化を行い画像間の並列性を利用した CryptoNets[3] がある．

本稿では，畳み込みニューラルネットワーク（CNN）の推論処理において，Microsoft SEAL ライブラリによる準同型暗号を C++で実装し，そのコードに OpenMP を用いたループ並列化を適用した．性能評価においては，MNIST のデータセットを使用して学習したモデルを対象に，準同型暗号を伴う推論処理の畳み込み層の並列実行し，提案手法の有効性を確認する．

2　SEAL による準同型暗号

準同型暗号は，公開鍵で暗号化した入力データに対し演算を行い，秘密鍵で復号することで出力データを得ることができる．Microsoft が提供するオープンソースライブラリである SEAL[4] には，BGV および BFV 方式と CKKS 方式が用意されているが，本稿では CKKS 方式を用いる．CKKS 方式の特徴としては近似を用いることにより実数や複素数の演算が可能である点が挙げられる．そのため，整数のみを扱う BGV および BFV 方式と比べ，CKKS 方式の方が機械学習での利用に適している．なお，CKKS 方式での準同型暗号を用いた演算の流れは，図 1 に示す通りである．

3　CNN における準同型暗号化と並列処理

本章では，対象とする CNN モデル及び準同型暗号化と並列処理について述べる．

3.1　対象とする CNN モデル

本稿では表 1 に表す 6 層の CNN モデルを使用する．準同型暗号化したデータでは値の大小比較が困難であるため，本稿における CNN モデル内では，活性化関数に

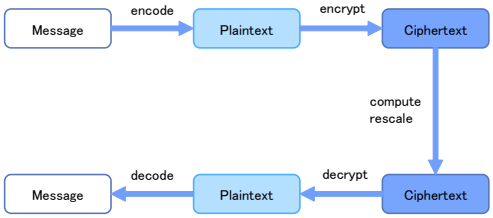

図 1　CKKS 方式の概要.

表 1　CNN の構成.

層の種類	説明	入力	出力
畳み込み層		28*28	30*24*24
活性化	Square 関数	30*24*24	30*24*24
プーリング層	Sum Pooling	30*24*24	30*12*12
全結合層		30*12*12	100
活性化	Square 関数	100	100
全結合層		100	10

Square 関数を適用し，プーリング層では Sum Pooling を使用している．なお，同様の理由から，文献 [5] では，近似活性化関数を用いた準同型暗号の CNN が提案されている．

このような CNN モデルを C++実装し，MNIST の 5000 枚の訓練用画像を用いて訓練を行い，重みとバイアスを事前にファイル保存しておく．準同型暗号化した CNN コードにおいては，事前学習した重みとバイアスをファイルから読み込み，準同型暗号を伴う推論処理を行う．

3.2　CNN における準同型暗号化

本稿で使用した CKKS 方式の Microsoft SEAL の各パラメータを表 3 に示す．SEAL では Plaintext 型へ変換されるとき，scale というパラメータを用いる．その値が一致しているデータ同士において乗算が可能であり，乗算のたびに scale の値は増えていく．そのため，初期設定値に近づくように rescale という処理を行わなければならない．なお，rescale が可能な回数は coeff_modulus に依存する．畳み込み層では，準同型暗号化し乗算処理を行った一つのデータに対し乗算は一度であるため，rescale 回数も一回を想定し，それが可能な coeff_modulus の値を設定している．

また，加算する場合，暗号化した際にそれぞれのデータに与えられる parms_id という暗号化パラメータが一致している必要がある．

本稿の推論で使用する MNIST の画像，重み，バイアスに対し，表 2 に示す SEAL のメソッドを用いて，畳み込み層の準同型暗号化を実装した．MNIST の推論用画像については，事前に 1 画像ずつ読み込み，その各ピクセルを Ciphertext 型へと準同型暗号化している．ま

†明治大学大学院先端数理科学研究科ネットワークデザイン専攻
Network Design Program, Graduate School of Advanced Mathematical Sciences, Meiji University

表 2　実装に使用した SEAL のメソッド.

メソッド名	説明
encode(double, scale, Plaintext)	Plaintext 型へ変換
encrypt(Plaintext, Ciphertext)	Plaintext 型から Ciphertext 型へ変換
multiply_plain(Ciphertext, Plaintext, Ciphertext)	Ciphertext 型と Plaintext 型の乗算
rescale_to_next_inplace(Ciphertext)	リスケール機能
add_plain(Ciphertext, Plaintext, Ciphertext)	Ciphertext 型と Plaintext 型の加算
mod_switch_to_inplace(Ciphertext, parms_id)	暗号パラメータの一致

表 3　CKKS 方式 SEAL のパラメータ.

パラメータ名	値
poly_modulus_degree	8192
scale	2^{40}
coeff_modulus	{60, 40, 40, 60}

表 4　性能評価マシン.

プロセッサ	Intel Xeon2265
コア	12cores, 3.50GHz
メモリ	128GB
SEAL	3.7.2

た，重みおよびバイアスにおいては秘匿化する必要は無いが，Ciphertext 型との演算を行うため，Plaintext 型への変換を行っている．本研究の方針としては，表 1 の 6 層の CNN 全体を準同型暗号化して性能評価することであるが，本稿では予備評価として，畳み込み層のみに準同型暗号化と並列化を適用して性能評価を行った．

3.3　準同型暗号化 CNN のループ並列処理

本稿の対象とする CNN の畳み込み層のコードは，6 重の for 文で記述される．即ち，最外側の for 文から順に，画像番号, フィルタ番号, 出力の行, 出力の列, フィルタの行, フィルタの列を誘導変数とする 6 個のネストされた for 文により構成される．

本稿では，画像間の並列性を利用するのはなく，フィルタ間の並列性を利用しており，フィルタ番号（0〜29）を誘導変数とする for 文に対して，OpenMP の指示文 #pragma omp parallel for を挿入して並列化を行う．

4　マルチコア上での準同型暗号化 CNN の性能評価

本性能評価では，3.1 節の CNN モデルに対して，3.2 節の準同型暗号化と 3.3 節の並列化を実装し，OpenMP を用いたループ並列処理の性能評価について述べる．

4.1　性能評価環境

本性能評価には，表 4 に示す Xeon サーバを利用する．また，データセットとしては，0 から 9 の手書き数字をグレースケール 28*28 ピクセルで表現された MNIST データセットを使用した．

4.2　Xeon 上での準同型暗号化された畳み込み層の評価

本節では，3.2 節の準同型暗号化と 3.3 節の並列化を実装したコードを用いて，MNIST の推論用画像の 1 画像を対象とし，準同型暗号化を伴う推論の畳み込み層の処理時間を測定した．畳み込み層では，28*28 の画像データ（暗号文）を入力とし，30 枚のフィルタにより畳み込み処理を行った後，30*24*24 のデータ（暗号文）が出力される．出力データを復号したところ，平文で行った場合と同じ結果が得られている．

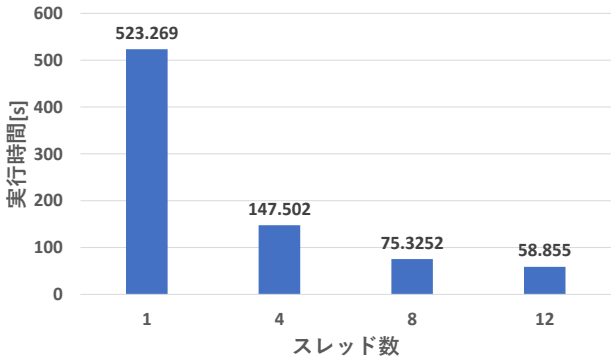

図 2　準同型暗号化した畳み込み層における並列処理の性能評価.

Xeon サーバ上での OpenMP による 1 スレッド，4 スレッド，8 スレッド，12 スレッドの実行結果は，図 2 の通りであり，12 スレッドの場合には，1 スレッド比で 8.89 倍の速度向上が得られた．

5　おわりに

本稿では，C++で実装した畳み込みニューラルネットワークの推論において，準同型暗号化および OpenMP によるループ並列化を行った．本手法では，Microsoft SEAL ライブラリの CKKS を用いて準同型暗号化を畳み込み層に対して行い，1 画像におけるフィルター間でのループ並列処理を実現した．MNIST データセットを用いて Xeon サーバーで行った性能評価の結果，畳み込み層における実行時間が，逐次実行時と比べ，12 スレッド実行時に 8.89 倍の速度向上が得られ，提案手法の有効性が確認された．

参考文献

[1] 宍戸 哲平, 西 将暉, 李 欣怡, 木村 啓二. 演算ビット数削減による準同型暗号ライブラリ SEAL の高速化，IEICE Technical Report, CPSY2021-60, 2022.

[2] 大西 隆太郎, 鈴木 拓也, 山名 早人. 準同型暗号と隔離実行環境を用いたプライバシ保護畳み込みニューラルネットワーク，DEIM Forum, 2023.

[3] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, et al. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy, Proc. The 33rd International Conference on Machine Learning, 2016.

[4] Microsoft Research, Redmond, WA. Microsoft SEAL (release 3.7), *https://github.com/Microsoft/SEAL*, 2021.

[5] 石山 琢己, 森澤 竣, 鈴木 拓也, 石巻 優, 山名 早人. 準同型暗号上での近似活性化関数を用いた畳み込みニューラルネットワーク推論の検討－精度改善に向けて－, DEIM Forum, E2-3, 2020.