C-017

準同型暗号を用いた残差ネットワークの並列処理

Parallelization of Residual Network Using Homomorphic Encryption

地引 壮悟 †　　吉田 明正 †
Sogo Jibiki　　Akimasa Yoshida

1　はじめに

準同型暗号は，暗号化したデータを復号せず，暗号化したまま加算や乗算ができる暗号方式である．この特性を活かして，機密データを暗号化したまま扱うことができ，機械学習への応用が研究されている [1][2]．画像分類の分野では，従来の CNN に残差接続を組み込んだ残差ネットワークがよく使用されている．本稿では，残差ネットワークの推論処理を C++実装し，Microsoft SEAL ライブラリ [3] で準同型暗号化を行い．そのコードに OpenMP による並列処理を適用することで性能向上を図った．32 コアの Xeon Platinum 上で，MNIST データセットを用いて行った性能評価を行い，提案手法の有効性を確認する．

2　SEAL による CKKS 方式

準同型暗号は暗号化されたデータを復号化することなく，加算や乗算が行える暗号方式である．その一つである CKKS 方式 [4] は他の BGV 方式や BFV 方式とは異なり，近似を用いることで，複素数のデータを扱うことができ，機械学習の分野での利用に適している．一方で乗算回数に制限があるので注意しなければならない．本暗号は Microsoft が提供するオープンソースライブラリである SEAL で容易に扱うことができる．

3　残差ネットワークにおける準同型暗号化と並列処理

本章では，対象とする残差ネットワーク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　残差ネットワーク

残差ネットワークは，ディープニューラルネットワークの学習を容易にするために開発されたアーキテクチャで，残差接続と呼ばれる構造が組み込まれている [5]．一般にニューラルネットワークモデルを深くすると表現力が増すが，勾配消失により学習が停滞することが分かっている．そこで通常の層に対して恒等写像でいくつかの層をスキップし，接続（加算）する．この時，スキップされた層を 1 つにまとめて残差ブロックという．この残差接続により学習時に，勾配が深い層においても消失することなく，伝わっていく．また，乗算が使用されないのでこの構造は CKKS 方式と相性がよい．残差ネットワークの一部を図 1 に示す．

3.2　対象とする残差ネットワーク

本稿では，表 1 に示す 10 層の残差ネットワークモデルを使用する．各入出力は (C,H,W) で表現する．暗号化したデータでは値の大小比較が難しいので，本モデルにおいては活性関数に 2 乗関数を適用し，プーリング層では合計プーリングを使用する．また，残差ブロックは畳み込み層が 2 つ，活性化層が 1 つで構成されている．このような残差ネットワークモデルを Python にて PyTorch で実装し，MNIST の 60000 枚の訓練画像を用いて訓練を行う．重みとバイアス，準同型暗号化したモデルにおける推論で使用するテストデータを事前に保存しておく．SEAL ライブラリは主に C++で使用されていることから，C++で同推論モデルを構築する．そしてこのモデルを準同型暗号化し，重みとバイアス，テストデータ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．

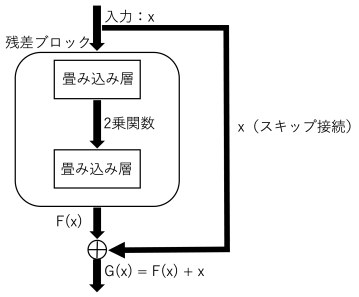

図 1　残差ネットワークの一部.

表 1　残差ネットワークの構成.

層の種類	入力	出力
畳み込み層	(1, 28, 28)	(16, 28, 28)
活性化層	(16, 28, 28)	(16, 28, 28)
残差ブロック 1 ※	(16, 28, 28)	(32, 14, 14)
残差ブロック 2 ※	(32, 14, 14)	(64, 7, 7)
プーリング層	(64, 7, 7)	(64, 1, 1)
全結合層	64	10

※残差ブロックは畳み込み層→活性化層→畳み込み層で構成.

3.3　残差ネットワークにおける準同型暗号化

CKKS 方式ではデータを平文空間に変換する際，scale というパラメータをかける．加算，乗算においてはそのパラメータが暗号文同士で一致している必要がある．また，乗算を繰り返すことでスケール値は指数的に増加するので，適切に処理しなければならない．そのため，rescale という処理でスケール値を適切に処理する．なお，rescale が可能な回数は乗算可能な回数と同義で，それは coeff_modulus の数値の個数でもある．畳み込み層，活性化層と全結合層の各層で暗号文に対して乗算は 1 度行われるので，その分だけ rescale が可能であるよう coeff_modulus を設定する．

また，演算の際には暗号文が持つ parms_id というパラメータが一致している必要もある．本稿の推論で使用する MNIST の画像は 1 ピクセルずつ Ciphertext 型に変換し，重みとバイアスは秘匿化する必要はないが画像

†明治大学総合数理学部ネットワークデザイン学科
Department of Network Design, School of Interdisciplinary Mathematical Sciences, Meiji University

との演算の都合上 Plaintext 型へ変換する．本稿で使用した ckks 方式の各パラメータを表 2 に示す．

表 2　CKKS のパラメータ．

パラメータ名	パラメータ値
poly_modulus_degree	$8192 * 2$
scale	2^{35}
coeff_modulus	$60, 35, 35, 35, 35, 35, 35, 35, 35, 35, 60$

3.4　準同型暗号化残差ネットワークのループ並列処理

本稿の対象とする残差ネットワークの各層の C++コードは for 文で記述され，主に最外側から出力マップのチャンネル，高さ，幅などを誘導変数とした for 文である．したがって本稿では最外側の for 文に OpenMP の指示文 #pragma omp parallel for を適用し，モデル並列性を利用する．加えて，高速化のために collapse 句が適用可能な畳み込み層や活性化層などは#pragma omp parallel for collapse(3) を適用し，3 階層の並列性を利用して高速化を図る．

4　マルチコア上での準同型暗号化残差ネットワークの性能評価

本性能評価では，3.2 節の残差ネットワークに 3.3 節の準同型暗号化と 3.4 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を行う．

4.1　性能評価環境

本性能評価には，表 3 に示す Xeon サーバを利用し，データセットとして MNIST データセットを使用した．

表 3　性能評価環境．

プロセッサ	Intel Xeon Platinum 8358
コア	32cores, 2.60GHz
メモリ	256GB
OS	Ubuntu 20.04LTS
gcc/g++	9.4.0
SEAL	4.1.2

表 4　準同型暗号化した残差ネットワークの並列処理時間（collapse 句有）．

スレッド数	実行時間 [s]	速度向上率 [倍]
1	21922	1.000
4	5615	3.904
8	2869	7.641
16	1580	13.875
32	1061	20.662

4.2　Xeon 上での準同型暗号化された残差ネットワークの評価

本節では，3.4 節のコードを用いて MNIST の推論用画像の 1 画像を対象とし，準同型暗号化残差ネットワークの推論処理時間を測定した．また，出力データを復号したところ，推論画像のラベルと一致した．Xeon サーバ上で OpenMP による 1 スレッド，4 スレッド，8 スレッド，16 スレッド，32 スレッドの実行結果は，表 4 に示す通りであり，32 スレッドの場合には，1 スレッド比で 20.662 倍の速度向上が得られた．

また，32 スレッドにおける各層の実行時間の測定も行なった．実行結果は表 5 に示す通りであり，準同型暗号化を用いた残差ネットワークの推論においては畳み込み層に要する時間が長く，残差ネットワークの特徴である残差接続に要する時間は比較的短い．

最後に OpenMP の collapse 句の有無による実行時間の比較を行なった．実行結果は，表 6 に示す通りであり，collapse 句を用いると 32 スレッドの時，collapse 句無に対して 0.950 倍に実行時間が短縮された．

表 5　32 スレッド並列処理による各層の実行時間．

層の種類	実行時間 [s]	比率 [%]
畳み込み層	40.779	3.843
活性化層	18.741	1.766
畳み込み層（残差ブロック 1）	249.626	23.527
活性化層（同上）	6.490	0.612
畳み込み層（同上）	347.810	32.781
Downsampling（同上）	27.180	2.562
残差接続（同上）	0.680	0.064
畳み込み層（残差ブロック 2）	142.029	13.386
活性化層（同上）	1.316	0.124
畳み込み層（同上）	156.398	14.741
Downsampling（同上）	16.416	1.547
残差接続（同上）	0.088	0.008
プーリング層	0.016	0.002
全結合層	0.102	0.010
その他	53.329	5.026
総時間	1061.000	100.000

表 6　OpenMP の collapse 句の有無の比較．

スレッド数	有 [s]	無 [s]	無に対する有の比率
4	5615	5824	0.964
8	2869	2984	0.961
16	1580	1639	0.964
32	1061	1117	0.950

5　おわりに

本稿では，C++で実装した残差ネットワークの推論において，準同型暗号化および OpenMP によるモデル並列化を行なった．MNIST データセットを用いて Xeon サーバで行なった性能評価の結果，推論の実行時間は逐次実行時と比べ最大 20.662 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] 光永 茉弥．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階層的並列処理，情報処理学会第 87 回全国大会，2J-01，2025.

[2] Lee JOON-WOO, Kang HYUNGCHUL, et al. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network, IEEE Access, Vol.10, pp.30039–30054, 2022.

[3] Microsoft, Microsoft SEAL(release 4.1.2), https://github.com/microsoft/SEAL, 2025.

[4] Jung Hee Cheon, Andrey Kim, Miran Kim, Yongsoo Song. Homomorphic Encryption for Arithmetic of Approximate Numbers, Lecture Notes in Computer Science, Vol.10624, pp.409–437, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition, arXiv preprint, arXiv:1512.03385, 2015.