효율적인 프라이버시 보존형 순환신경망을 위한 활성화함수의 cell-wise 근사

주유연[1], 남기빈[1], 하승진[1], 백윤흥[1]
[1]서울대학교 전기정보공학부, 서울대학교 반도체공동연구소
{yyjoo, kvnam, sjha}@sor.snu.ac.kr, ypaek@snu.ac.kr

A Cell-wise Approximation of Activation Function for Efficient Privacy-preserving Recurrent Neural Network

Youyeon Joo[1], Kevin Nam[1], Seungjin Ha[1], Yunheung Paek[1]
[1]Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center(ISRC), Seoul National University

요 약

원격 환경에서의 안전한 데이터 처리를 위한 기술 중 동형암호는 암호화된 데이터 간의 연산을 통한 프라이버시 보존형 연산이 가능하여 최근 딥러닝 연산을 동형암호로 수행하고자 하는 연구가 활발히 진행되고 있다. 그러나 동형암호는 신경망에 존재하는 비산술 활성화함수를 직접적으로 연산할 수 없어 다항함수로 대체하여 연산해야만 하는데, 이로 인해 모델의 정확도가 하락하거나 과도한 연산 부하가 발생하는 등의 비효율성 문제가 발생한다. 본 연구에서는 모델 내의 활성화함수를 서로 다르게 근사하는 접근을 순환신경망(Recurrent Neural Network, RNN)에 적용하여 효율적인 동형암호 연산을 수행하는 방법을 제안하고자 한다.

1. 서론

원격 기반 머신러닝 서비스는 모델 연산을 위해서는 사용자의 데이터를 원격 서버에 전송하여 추론을 수행해야 하는데, 일반적으로는 원격 서버에서 사용자의 데이터는 복호화되어 평문 상태로 올라가게 된다. 이때 원격 서버 내의 악의적인 사용자에 의한 데이터 유출 우려가 있어 이를 방지하기 위해 다양한 프라이버시 보존 기술이 등장했다. 그중 동형암호는 암호화된 데이터 간의 덧셈, 곱셈 연산과 이들의 조합으로 이루어진 연산을 수행할 수 있어 원격 환경에서도 복호화없이 안전하게 데이터를 처리할 수 있기에 원격 환경에서의 프라이버시 보존형 머신러닝 연산을 가능하게 한다. 동형암호를 원격 연산 환경에 적용한다면, 서비스를 위한 위치정보나 얼굴 등의 개인의 민감한 정보는 개인 디바이스를 떠나는 순간부터 원격 서버에서의 모델 추론 및 그 결과를 되받는 순간까지 모든 과정에서 암호화되어 원격 서버로부터 완전히 안전하게 보호할 수 있는 프라이버시 보존형 컴퓨팅이 가능하게 되어 프라이버시 보존 머신러닝(PPML)이 가능해진다.

동형암호는 그 수학적 성질으로 인해 암호문 간의 덧셈, 곱셈만 직접 지원한다. 이는 딥러닝 모델 내에 존재하는 비산술 연산인 활성화함수를 동형암호로 직접적인 연산이 어려워 모델 연산을 완전히 동형암호로 변환하기 위해서는 활성화함수의 다항함수로의 근사는 불가피하다는 것을 의미한다.

본 연구에서는 모델 내 모든 활성화함수를 하나의 함수로 근사하지 않고, layer-wise 근사식을 활용하는 최근 연구 동향에 맞추어 순환신경망의 활성화함수인 tanh를 cell-wise 근사하여 효율적인 HE 연산을 시도하고, 이를 실험으로써 보이고자 한다.

2. 관련 연구 및 Motivation

이미지 처리를 위한 합성곱신경망(CNN) 내의 ReLU = max(x, 0) 함수에 대한 근사 연구는 다양한 접근으로 활발하게 이루어졌다. Cryptonets[1]는 $y = x^2$을 ReLU 대신에 사용하여 모델 연산을 수행했고, Faster cryptonet[2]은 $y = \frac{1}{8}x^{2+}\frac{1}{2}x + \frac{1}{4}$를 활성화함수로 사용하였다. 이후에도 N차 다항식을 ReLU의 근사함수로 대체하여 추론연산을 구현한 연구들이 있었으나 대부분 얕은 모델에 대해서는

ReLU 기반 모델과 유사한 수준의 정확도를 유지하였으나 최근 딥러닝 응용에서 채택되는 ResNet 수준의 깊이를 가지는 모델에서는 30% 이상 하락하는 등 심각한 정확도 하락을 보이는 한계가 있었다. 깊은 모델에서도 정확한 근사를 하는 연구로는 MPCNN[3]이 있는데, 이 연구는 sign 함수에 대한 k개의 근사함수를 합성하여 ReLU 함수를 근사하였다. 이는 100개 이상의 레이어를 갖는 ResNet-110에서도 1-2퍼 이내의 정확도 하락만 보이는 등 매우 정확한 근사를 수행했다. 그러나 여러 함수를 합성하는 접근으로 인해 동형암호 연산의 복잡성이 증가하여 모델의 추론 시간이 오래 걸린다.

한편 빠르지만 부정확한 저차식과 정교하지만 매우 큰 성능부하를 일으키는 고차식 근사 접근 사이에서 모델 내 모든 활성화함수를 단 하나의 근사함수로 대체해야만 하는가에 대한 의문이 발생한다. DeepReduce[4]는 모델 내의 모든 활성화함수가 동일한 수준의 중요성을 갖지 않음을 실험으로써 보여, 모델의 정확도에 큰 영향을 미치는 활성화함수가 있고, 그렇지 않은 활성화함수가 있다는 것을 보였다. 이에 최근 모델 내 활성화함수의 서로 다른 근사 방법을 통한 효율적인 동형암호 모델 설계에 대한 연구가 진행되고 있다. SAFENet[5]는 모델의 활성화함수를 2차식과 3차식을 섞어 channel-wise 활성화함수의 효율성을 보였다. AutoFHE[6]는 CNN의 layer-wise 활성화함수 근사를 위해 유전 알고리즘을 활용한 모델 파인튜닝을 제안했다. AutoFHE는 2차식부터 최대 6개의 근사함수의 합성함수를 모두 고려하여 유전 알고리즘을 구성하여 CNN 모델에 관해 기존 연구와 대비하여 더욱 효율적이면서 비슷한 수준의 정확도를 달성했다. <표 2>는 활성화 함수 근사 접근에 따라 기존연구를 분류한 표이다. 동형암호 연산을 위해 다항식 근사를 활용하되, 이를 어느 시점에 다항식으로 변환할지와 모든 활성화함수를 하나의 함수로 근사하거나 각기 다른 함수를 활용하는 접근을 나누어 분류했다.

<표 2> 활성화 함수 근사 접근에 따른 분류

근사 적용 시점	Model-wise	Layer-wise
학습	AESPA[7], MatHEAAN[8]	SAFENet[5], AutoFHE[6]
추론	MPCNN[3]	J Lee.[9]

<표 1> MNIST CNN/RNN PPML 정확도 예시

	평문	암호문
LeNet-5 (CNN)	98.90%	98.20% (-0.70%)
GRU (RNN)	98.23%	94.20% (-4.03%)

그러나 해당 연구들은 CNN의 대표 활성화 함수인 ReLU를 타겟하여 근사를 시도한 연구로, RNN에서 주로 활용하는 활성화함수인 또다른 비산술 연산인 tanh, sigmoid에는 ReLU의 비교 연산이 아닌, 삼각, 지수 함수로 구성되어 있다. 결국 이들을 근사하기 위해서는 다항식 근사를 활용해야 하는데, RNN에서는 CNN만큼 충분히 효율적이지 않았다. <표 1>은 MNIST 데이터셋에 대한 간단한 CNN, RNN 모델의 정확도 예시이다. 두 모델 모두 동일한 tanh를 활성화함수로 학습하였으며, 추론 과정에서 근사다항식으로 대체하여 암호문의 정확도를 시뮬레이션한 결과이다. CNN에 비해 RNN이 더 많은 근사함수를 수행함에 따른 원함수와 근사함수의 편차 누적이 심해서 더 큰 정확도 하락이 발생한 것을 확인할 수 있다. 따라서 RNN의 편차 누적을 완화하기 위해 cell-wise 다항식 적용이 필요하며, 본 연구는 재학습없이 추론단계에서의 근사를 목표로 한다.

3. Cell-wise 활성화함수를 적용한 순환신경망

(1) **실험 환경** 본장의 실험은 Intel Xeon Gold 6326 2.9GHz CPU 2개와 RTX A6000 GPU 및 1TB 용량의 메모리가 장착되어있는 서버에서 진행했다. 모델 학습은 GPU를 사용했고, 근사 모델 탐색은 총 64개의 쓰레드를 이용해 CPU를 사용했다. 동형암호 연산 시간은 싱글 쓰레드 CPU로 측정하였다. 딥러닝 라이브러리는 tensorflow 2.13.1를 사용하였으며, 동형암호 시뮬레이션은 HEaaN 라이브러리의 CPU 버전을 이용했으며, 암호 파라미터는 <표 3>의 bootstrapping이 가능한 HEaaN의 FGb 파라미터를 사용하였다. FGb는 최대 9의 multiplicative depth를 허용하는 파라미터로, $Q_{\min}$은 최소 레벨에서의 Q 값으로 추가 곱셈 연산을 위해서는 bootstrapping이 요구되며, $Q_{\max}$는 bootstrapping 직후의 Q 값을 의미한다. scaling factor는 42 bit prime을 사용하였다.

<표 3> HEaaN FGb 파라미터

N	$\log_2(Q)$	$\log_2(Q_{\min})$	$\log_2(Q_{\max})$
2^{16}	1258	184	562

(2) baseline 모델 학습 본 실험에서는 MNIST 데이터셋을 이용하여 학습한 Vanilla RNN 모델을 타겟으로 하였다. 이는 기존 다항식 근사 접근을 활용한 CNN 연구와 유사한 수준의 근사함수 수를 갖는 RNN 모델에 대한 cell-wise 활성화함수의 효율성과 직관성을 보이기 위해 한 cell 내에 하나의 활성화함수를 갖는 Vanilla RNN을 사용하였다.

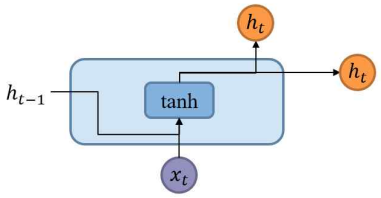

그림 1 Vanilla RNN Cell

Vanilla RNN의 baseline 모델은 다음의 하이퍼파라미터를 사용하여 학습하였다. 먼저 모델의 파라미터로 SimpleRNN의 hidden unit은 32로 설정하고 활성화함수로는 default인 tanh를 사용하였으며, 바로 이어 Dense 레이어를 위치하게 하여 다른 레이어 연산으로 인한 정확도 영향을 최소화하였다. 모델 학습 하이퍼파라미터는 배치 사이즈 128, 에포크 100, Adam optimizer를 사용하였으며, 학습 과정의 과적합을 막기 위해 에포크 10 단위 EarlyStopping을 적용하고 tensorflow의 ReduceLROnPlateau를 learning rate 스케줄러로 사용하여 learning rate를 10e-2부터 10e-5 수준까지 설정하였다.

(3) cell-wise 근사다항식 탐색 기존 연구[8]는 tanh에 대한 근사 범위를 [-8, 8]으로 설정하고, 이에 대한 최소제곱법을 이용한 근사다항식을 얻었다. 본 연구에서는 동일한 근사 접근을 택하고, tanh의 3차, 7차, 17차 근사 다항식을 후보 다항식으로 두었다. 동형암호로 연산시 N차 다항식은 log(N)만큼의 multiplicative depth를 소모하는데, 3차와 17차 다항식은 2개만큼의 depth 차이가 있기에 최대한 많은 수의 활성화함수를 낮은 차수로 연산하여 depth 소모를 최소화하는 것을 목적으로 한다. 이에 먼저 모든 활성화함수를 17차 근사다항식으로 대체한 모델을 최초 모델으로 초기화하였다. 이후 각 time-step의 활성화함수를 3차식, 7차식으로 대체하여 같은 방법으로 반복 측정하였다. 각 탐색 시점에서의 모델 정확도 하락 허용치는 직전 상태 모델의 정확도의 2% 이내 수준으로 설정하였다. 허용 수준을 넘은 근사 cell에 대하여 또 다른 time-step을 계속하여 근사하는 깊이우선탐색(Depth-First Search, DFS)을 수행하며 모델을 탐색했다.

4. 실험 결과

(1) 근사 다항식 모델 탐색 cell-wise 근사다항식 탐색의 search space는 약 2^{44}인데, 허용 정확도에 따른 차이는 발생하지만 본 실험의 세팅에서는 전체 2^{11} 수준의 search를 진행하여 총 7.49시간이 소요되어 허용 정확도 2% 범위의 총 525개의 근사 모델을 찾았다. 각 근사 모델은 정확도 오차 범위 내에서 최대 4개의 17차 근사다항식을 3차 또는 7차로 대체하였다. <그림 2>와 <그림 3>은 각각 최초 모델에서 3, 7차식을 활용해 근사한 모델과 7차식만 활용해 근사한 모델들에서의 각 셀에 대한 활성화함수의 분포를 나타낸 그래프이다.

그림 2 3차, 7차 근사식의 분포 (29 samples)

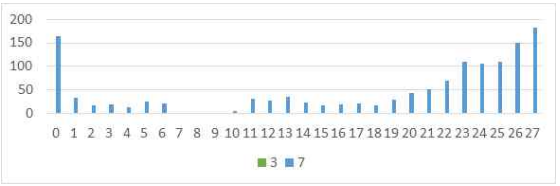

그림 3 7차 근사식의 분포 (496 samples)

전체 28개 cell 내의 활성화함수 중에서 3차 또는 7차식으로 대체된 활성화함수는 초반 cell 또는 후반부에 집중되어 있는 것을 확인할 수 있다. 이는 DeepReduce[4]의 실험적 결과와 유사한 결과로, 모델의 초반 또는 후반부에 위치한 활성화함수가 중앙부에 비해 상대적으로 정확도에 미치는 영향이 크지 않음을 의미한다. <그림 4>는 각 근사 경우의 수에서의 최고 정확도 모델의 활성화함수 분포이다. 레이블의 숫자는 해당 모델에서의 3차, 7차, 17차 근사다항식의 개수를 의미한다.

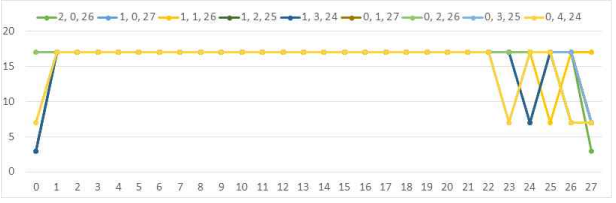

그림 4 활성화함수 근사 차수 분포

(2) 동형암호 수행시간 본 시뮬레이션에서 사용한 암호 파라미터는 <표 3>에서 확인할 수 있다. 앞서 서술했듯이, 동형암호 연산 시 활성화함수를

가능한 낮은 차수로 근사해야 효율적인 연산이 가능하다. <표 4>는 MNIST SimpleRNN 모델을 동형암호로 연산했을 때의 정확도 및 예상 연산 시간이다. 근사 모델 중 근사한 활성화함수의 개수가 가장 많은 (0, 4, 24) 케이스에 대해 측정한 결과이다.

<표 4> SimpleRNN 동형암호 시뮬레이션

	정확도	연산 시간
baseline	91.49%	-
All 17	91.19%	628.03 초
(0, 4, 24)	90.03%	581.23 초

활성화함수를 cell-wise로 근사하여 다항식을 배치했을 때, 모두 같은 차수로 근사했을 때보다 정확도는 보다 낮지만 30초 가량 더 빠른 추론 연산이 가능하다. 이는 CNN의 layer-wise 근사 접근과 동일하게 비교적 연산 편차 영향이 큰 RNN에서도 cell-wise 근사 접근이 유효함을 보여준다. 실험에서 허용한 2%보다 큰 정확도 오차를 허용한다면, 더 많은 cell을 낮은 차수로 근사할 수 있어 더 빠른 추론 연산도 가능하다.

4. 결론

본 연구에서는 모델 내의 활성화함수를 서로 다르게 근사하는 CNN의 layer-wise 근사 접근을 순환신경망(Recurrent Neural Network, RNN)에 적용하여 cell-wise 근사의 효율성을 실험적으로 보였다. 실험 결과를 기반으로 RNN에서도 서로 다른 활성화함수 적용이 유효함을 확인하여 더욱 효율적인 연산을 할 수 있음을 보였다. 또한 사용자가 허용하는 오차 범위를 설정하는 사용자화도 가능하여 본 연구의 접근을 확장한다면 보다 효율적인 동형암호 모델 연산이 가능할 것이라 기대된다.

5. ACKNOWLEDGEMENT

이 논문은 2024년도 BK21 FOUR 정보기술 미래인재 교육연구단에 의하여 지원되었음. 이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (RS-2023-00277326). 이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (IITP-2023-RS-2023-00256081)

참고문헌

[1] Gilad-Bachrach, Ran, et al., "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy.", International conference on machine learning. PMLR, 2016.

[2] Chou, Edward, et al., "Faster cryptonets: Leveraging sparsity for real-world encrypted inference." arXiv preprint arXiv:1811.09953 (2018).

[3] Lee, Eunsang, et al., "Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions." International Conference on Machine Learning. PMLR, 2022.

[4] Jha, Nandan Kumar, et al., "Deepreduce: Relu reduction for fast private inference." International Conference on Machine Learning. PMLR, 2021.

[5] Lou, Qian, et al., "Safenet: A secure, accurate and fast neural network inference." International Conference on Learning Representations. 2020.

[6] Ao, Wei, and Vishnu Naresh Boddeti., "Autofhe: Automated adaption of cnns for efficient evaluation over fhe." arXiv preprint arXiv:2310.08012 (2023).

[7] Park, Jaiyoung, et al., "AESPA: Accuracy preserving low-degree polynomial activation for fast private inference." arXiv preprint arXiv:2201.06699 (2022).

[8] Jang, Jaehee, et al. "Privacy-preserving deep sequential model with matrix homomorphic encryption." Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. 2022.

[9] Lee, Junghyun, et al., "Optimizing layerwise polynomial approximation for efficient private inference on fully homomorphic encryption: a dynamic programming approach." arXiv preprint arXiv:2310.10349 (2023).