동형암호 기반 딥러닝 기법 연구 동향

임세진[1], 김현지[1], 강예준[1], 김원웅[1], 서화정[1]
[1]한성대학교 IT융합공학부
dlatpwls834@gmail.com, khj1594012@gmail.com, etus1211@gmail.com,
dnjsdndeee@gmail.com, hwajeong84@gmail.com

Trends in deep learning techniques based on Homomorphic Encryption

Se-Jin Lim[1], Hyun-Ji Kim[1], Yea-Jun Kang[1], Won-Woong Kim[1], Hwa-Jeong Seo[1]
[1]Dept. of IT Convergence Engineering, Han-Sung University

요 약

딥러닝 기술이 발전하면서 적용되는 산업 분야가 늘어남에 따라 딥러닝 모델에서 역으로 학습 데이터를 추출하는 등 다양한 딥러닝 모델 공격 이슈가 발생하고 있다. 이러한 위협에 대응하기 위해 딥러닝 학습에 사용되는 데이터의 노출을 방지할 수 있도록 사용자 프라이버시를 보호하는 기법의 중요성이 대두되고 있다. 동형암호는 학습 데이터를 보호할 수 있는 기법 중 하나로, 복호화 과정없이 암호화된 상태로 연산, 탐색, 분석 등을 수행할 수 있는 차세대 암호 알고리즘이다. 본 논문에서는 동형암호 기반의 딥러닝 기법 연구 동향에 대해 알아본다.

1. 서론

최근 딥러닝 모델에서 역으로 학습데이터를 추출하는 등[1] 다양한 공격 기법이 연구되고 있다. 딥러닝 기술이 산업 전반에 적용되면서 딥러닝 모델에 사용되는 사용자 프라이버시와 관련된 민감한 데이터도 증가하고 있어 학습데이터에 대한 보호는 필수적이다. 학습 데이터를 보호하는 기법에는 연합학습(Federated Learning), 차분 프라이버시(Differential Privacy), 데이터 비식별화, 동형암호(Homomorphic Encryption) 등이 있다. 암호 알고리즘을 통해 데이터를 보호할 수 있지만 학습 시 데이터를 연산하기 위해서는 복호화 과정이 반드시 필요하며, 이때 복호화로 인한 데이터 유출의 위험이 존재한다. 동형암호는 이러한 문제를 해결하기 위해 등장한 암호 알고리즘으로, 복호화 과정 없이도 암호문 간 연산, 탐색, 분석 등의 작업을 수행할 수 있다. 많은 기업들이 민감한 독점 데이터를 아웃소싱할 때 발생할 수 있는 노출 위험에 대해 우려하고 있는데, 데이터를 암호화된 상태로 작업할 수 있는 동형암호는 이러한 문제를 해결할 수 있을 것으로 보여 주목받고 있다. 본 논문에서는 이러한 동형암호가 적용된 딥러닝 기법의 연구동향에 대해 알아보고자 한다.

2. 관련 연구

2.1 동형암호(Homomorphic Encryption)

동형암호는 평문과 암호문의 동형(Homomorphic) 성질로 인해 복호화 키에 액세스 하지 않고도 데이터가 암호화된 상태로 연산이 가능한 차세대 암호기술이다. 즉, 평문을 연산한 결과 값을 암호화한 것과 각각의 평문을 암호화한 후 연산한 결과가 동일한 것을 말한다. 연산의 유형, 횟수 등에 따라 부분동형암호, 준동형암호, 완전동형암호와 같이 3가지로 분류할 수 있다[2]. 부분동형암호(Partially Homomorphic Encryption, PHE)는 암호화한 상태에서 특정 한가지 연산만 수행할 수 있는 동형암호를 말한다. 덧셈만 가능하거나 곱셉만 가능한 동형암호가 이에 해당한다. 준동형암호(Somewhat Homomorphic Encryption, SHE)는 PHE에 비해 임의의 연산을 수행할 수 있으나 횟수가 제한된 동형암호를 말한다. SHE는 암호문에 노이즈를 삽입하는 방식으로 기밀성을 유지하는데, 연산을 거듭할수록 노이즈가 증가하여 원본 정보에 훼손이 발생하기 때문에 연산 횟수에 제한이 있다. 완전동형암호(Fully Homomorphic Encryption, FHE)는 모든 논리 연산을 지원하는 동형암호로써 데이터에 대해 횟수 제한없이 다양한 유형의 연산을 수행할 수 있다. 또한 FHE는 격자 기반 암호 중 하나로 양자컴퓨터의 상용화 이후에도 사용이 가능하다. FHE는 노이즈가

커진 암호문을 재부팅(bootstrapping)하는 과정을 통해 노이즈를 줄인 새로운 암호문을 생성하기 때문에 지속적인 연산수행이 가능하다. 하지만 성능 측면에서는 현저히 불리하다. 딥러닝에 동형암호 알고리즘을 적용하면 민감한 정보를 안전하게 보호하면서도 유용하게 데이터를 처리할 수 있어 사용자의 프라이버시를 보호하면서 딥러닝 학습을 수행할 수 있다. 하지만 암호화 키 크기가 증가함에 따라 데이터 크기가 매우 커진다는 단점이 있다. 또한 암호화된 데이터에 대한 연산은 원본 데이터에 대한 연산보다 매우 느리다. 따라서 암호화 데이터의 크기를 줄이고 연산 속도를 높이고자 하는 연구가 활발히 진행되고 있다. 현재 알려진 동형암호 라이브러리에는 IBM에서 개발하는 HELib, 마이크로소프트의 SEAL, 서울대학교의 HEAAN 등이 있다[3].

2.2 완전동형암호 스킴

대표적인 완전동형암호 알고리즘으로 CKKS 알고리즘 및 TFHE 알고리즘이 있다. CKKS[4]은 실수 연산이 가능한 최초의 동형암호이며 복소수 데이터에 대한 연산도 지원한다. 2017년에 제안된 신세대 동형암호 알고리즘으로 딥러닝 기술에 적합하며 활용도가 매우 높다. 하지만 안전성을 보장하는 에러를 데이터의 오차로 허용하여 연산이 증가함에 따라 데이터의 정확도가 낮아질 수 있는 한계점이 존재한다. 또한 재부팅에 많은 시간이 소요되어 깊은 연산 시 수행시간이 늘어나게 된다.

TFHE[5]는 재부팅 연산을 빠르게 할 수 있으며 비선형 함수 및 비트 연산에 강점을 가진다. 하지만 다른 동형암호 알고리즘이 가지고 있는 SIMD 성질이 제한적으로 존재하며, 게이트마다 재부팅 연산이 수행되어야하므로 한 번에 하나의 비트 연산만 가능하다는 한계점이 있다.

2.3 딥러닝에서의 동형암호

딥러닝에서 동형암호를 적용하여 데이터를 보호하는 과정은 다음과 같이 이루어진다[6]. 개인 데이터를 가진 사용자와 연산을 수행하는 서버가 있다고 하자. 사용자는 공개키, 비밀키, 비밀키로부터 생성한 평가키를 가진다. 비밀키를 제외한 키는 서버와 공유하며, 사용자의 공개키로 데이터를 암호화하기 위해 필요한 파라미터 또한 서버와 공유한다. 사용자가 공개키로 데이터를 암호화하여 서버에 보내면 서버는 일반 데이터로 학습된 모델을 사용하여 추론한다. 이때 evaluation 과정을 통해 완전연결 계층과 합성곱 계층에 필요한 동형 연산(덧셈, 곱셈)을 수행한다. 그러나 기존 신경망에서는 연산에 실수 값을 사용하고 동형암호 연산에서는 다항식 연산을 수행하므로 인코딩 과정을 통해 기존의 신경망을 수정하여 사용해야한다. 이 과정에서 원본 데이터를 다항식으로 변환하며 bias나 padding 값 또한 인코딩하여 사용된다. 이처럼 동형암호 연산을 수행하기 전에 인코딩 과정을 거치며, 이후 암호화된 상태로 추론 연산이 수행된다. 서버는 마지막 출력층에서 활성화 함수를 적용하지 않은 상태로 결과를 반환한다. 최종적으로 사용자가 암호화된 연산 결과에 활성화 함수를 적용하여 예측하며, 해당 값을 사용자의 비밀키로 복호화하여 결과를 확인하게 된다.

3. 동형암호 기반 딥러닝 기법 연구 동향

비선형 활성화 함수는 딥러닝 발전에 가장 큰 기여를 한 기술 중 하나로, 주로 사용되는 함수에는 sigmoid, ReLU 등이 있다. 이러한 비선형 함수는 평문 상에서는 연산하기 쉬운 편에 속하나, 덧셈과 곱셈만 지원하는 암호문 상에서의 정확한 연산은 불가능하다. 합성곱 신경망 연산처럼 딥러닝에서 주로 사용되는 연산은 대체로 선형 함수이기 때문에 동형 암호로 구현하는 것이 가능하다. 따라서 비선형 활성함수를 어떤 방식으로 구현하는지가 중요한 문제가 된다[7]. 또한 동형암호는 연산량이 많아 연산에 많은 시간이 걸리므로 연산 속도를 개선하는 것도 매우 중요한 문제이다.

3.1 CryptoNet[8]

2016년 마이크로소프트는 불필요한 연산을 줄이고 행렬 형태로 연산을 정리하여 한번에 처리하는 체계를 고안하였다. 비선형 함수들은 동형암호 상에서도 계산할 수 있게 모두 제곱 다항식 형태로 전환하여 적용하였다. 클라우드 환경 및 딥러닝 환경에서 활용할 수 있는 동형암호 알고리즘인 SEAL로 암호화한 MNIST 데이터를 학습시켜 정확도 99%를 달성하였다.

3.2 [9]

[9]는 Google의 Swish 활성화 함수, ReLU 활성화 함수의 다항식 근사 활성화 함수로 저차 다항식이 아닌 4차 다항식을 사용하여 분류 정확도를 개선하였다. 마이크로소프트의 SEAL에서 제공하는 동형암호

스킴 중 CKKS 스킴 기반의 SEAL을 사용하여 구현하였다. MNIST 와 CIFAR-10에 대해 각각 99.22%, 80.48%의 높은 정확도를 달성하였으며, 각각 이전 연구에 비해 0.04%, 4.11% 개선된 성능을 보인다.

3.3 HCNN[10]

[10]에서는 완전동형암호 기반의 CNN을 제안했다. 사전에 학습된 모델 상에서 암호화된 이미지를 연산하는 최초의 연구이다. pooling 대신 average pooling을 적용했으며, 정수형 동형암호를 활용하는 대신 scaling factor를 별도 저장하여 메모리 오버헤드가 증가하였다. MNIST 데이터를 학습시키는 HCNN 모델의 경우, CryptoNet[8]보다 적은 계층으로 99%의 정확도를 달성하였으며 5.16초가 소요된다. 또한 CIFAR-10 데이터셋에서는 77.55%의 정확도를 가지며 304.43초가 소요된다.

3.4 [11]

[11]은 성능 개선을 위해 ReLU 함수를 최대한 정확하게 근사하여 기존의 딥러닝 모델인 ResNet에 적용한 연구이다. minimax polynomial의 합성 기법을 사용하여 CIFAR-10과 ImageNet을 덧셈과 곱셈만으로 높은 정확도로 분류하는 데 성공하였다. 특히 [11]에서 제안한 방법은 ResNet과 같은 실제 성능이 검증된 기존의 딥러닝 모델의 구조와 사전에 학습되어 있는 파라미터를 그대로 활용할 수 있다는 장점이 있다. ReLU 함수를 정확하게 근사할수록 이미지 분류의 성능이 정확해진다.

4. 결론

딥러닝에 동형암호 알고리즘을 적용하면 민감한 정보를 안전하게 보호하면서도 데이터를 처리할 수 있어 사용자 프라이버시 보호를 위한 가장 좋은 해결책으로 보인다. 하지만 동형암호는 연산량이 많아 연산에 많은 시간이 걸린다. 또한 동형암호는 덧셈과 곱셈을 보존하지만, sigmoid나 ReLU처럼 딥러닝에서 많이 사용되는 활성화 함수 연산은 보존하지 않기 때문에 다항함수가 아닌 활성화 함수들은 다항함수로 근사하거나 다항함수로 대체해야 한다. 이때 고차 다항식으로 근사할수록 연산에 필요한 시간이 지수적으로 증가하기 때문에 연산 속도를 높이기 위한 연구가 활발히 진행되고 있다. 동형암호의 암·복호화 속도를 개선하기 위한 연구들이 많이 이루어지고 있지만 아직까지 복잡한 딥러닝 모델에 적용하기에는 어려움이 있다. 동형암호의 연산 속도 개선을 위해 연산량을 줄일 수 있는 기법, 하드웨어를 통한 연산 속도 개선 등 다양한 방면으로 연구가 더욱 필요하다.

5. Acknowledgement

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2018-0-00264, IoT 융합형 블록체인 플랫폼 보안 원천 기술 연구, 100%).

참고문헌

[1] Z. Yang, J. Zhang, E.C. Chang, and Z. Liang, “Neural Network Inversion in Adversarial Setting via Background Knowledge Alignment,” Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, ACM, pp. 225-240, 2019.

[2] G.Y.Lee, “Trends in Personal Information Protection and Management Technology for Big Data Utilization,” Information & Communications Magazine, 37(1), pp. 32-39, 2019.

[3] J.Y.Kim, N.S.Jho, and K.Y.Chang, “Trends in Data Privacy Protection Technologies with Enhanced Utilization,” Electronics and Telecommunications Research Institute, 35(6), 2020.

[4] J.H.Cheon, A.Kim, M.Kim, and Y.S.Song, “Homomorphic Encryption for Arithmetic of Approximate Numbers,” ASIACRYPT 2017, pp 409-437, 2017.

[5] I. Chillotti, N. Gama, M. Georgieva and M. Izabachène, “TFHE: Fast Fully Homomorphic Encryption Over the Torus,” Journal of Cryptology, 33, pp. 34 - 91, 2020.

[6] H.J.Kim and H.J.Seo, “Trends in deep learning technology to protect the privacy of training data,” ACK2021, pp.451-453, 2021.

[7] J.H.Lee and J.S.No, “A Study on the Performance of the Approximate Convolutional Neural Network According to Approximation Bound for Homomorphically Encrypted Data,” The 2nd Korea Artificial Intelligence Conference, pp 227-229, 2021.

[8] N. Dowlin et al., “CryptoNets: Applying

Neural Networks to Encrypted Data with High Throughput and Accuracy," Microsoft Research, 2016.
[9] T. Ishiyama, T. Suzuki and H. Yamana, "Highly Accurate CNN Inference Using Approximate Activation Functions over Homomorphic Encryption," 2020 IEEE International Conference on Big Data, Atlanta, GA, USA, 2020, pp. 3989-3995.
[10] A. Al Badawi et al., "Towards the AlexNet Moment for Homomorphic Encryption: HCNN, the First Homomorphic CNN on Encrypted Data With GPUs," IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 3, pp. 1330-1343, 2021.
[11] J.H.Lee, E.S.Lee, J.W.Lee, Y.J.Kim, Y.S.Kim and J.S.No, "Precise Approximation of Convolutional NeuralNetworks for Homomorphically Encrypted Data," arXiv preprint arXiv:2105.10879, 2021.