신뢰실행환경을 활용한 딥러닝 추론에 관한 연구

주유연[1], 백윤흥[1]
[1]서울대학교 전기·정보공학부, 서울대학교 반도체 공동연구소
yyjoo@sor.snu.ac.kr, ypaek@snu.ac.kr

A Study on Deep Learning Inference using Trusted Execution Environment

You-yeon Joo[1], Yun-heung Paek[1]
[1]Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center(ISRC), Seoul National University

요 약

딥러닝 원격 컴퓨팅 서비스(Deep Learning as a Service, DLaaS)가 널리 활용되면서 클라우드에서의 개인 정보 보호에 대한 우려가 커졌다. 신뢰실행환경(Trusted Execution Environment, TEE)는 운영체제의 접근까지 차단한 메인 프로세서의 보안 영역으로 DLaaS 환경에서의 개인 정보 보호 기법으로 채택되고 있다. 사용자의 데이터를 보호하면서 고성능 클라우드 환경을 활용하기 위해 신뢰실행환경을 활용한 딥러닝 모델 추론 연구들을 살펴보고자 한다.

1. 서론

다양한 문제를 해결하는 데에 있어 딥러닝이 널리 활용되고 있다. 일반적으로 모델의 구조가 깊거나 학습 데이터가 많을수록 모델의 정확도가 높아진다. 그러나 사용자의 개인 컴퓨터로 대용량의 데이터를 수집하거나 깊은 모델을 구축하여 학습하기에는 한계가 있다. 이에 아마존, 구글과 같은 클라우드 서비스업체는 딥러닝 원격 컴퓨팅 서비스(Deep Learning as a Service, DLaaS)를 제공하고 있다.

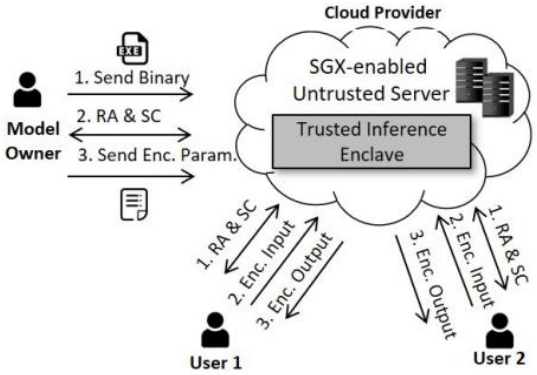
고성능 클라우드 환경을 활용하는 것은 유용하지만 근본적인 개인 정보 보호에 대한 우려가 있다. DLaaS 환경에서 사용자는 클라우드에 데이터를 보내고 그에 대한 모델 추론 결과를 돌려 받는다. 이때 클라우드 내의 공격자가 있다면 악의적인 의도로 사용자의 데이터를 탈취할 위험이 있다. 이러한 위협으로부터 사용자의 데이터를 보호하기 위한 방법으로 동형암호(Homomorphic Encryption, HE), 차분 프라이버시(Differential Privacy, DP), 다자간 연산(Multi-Party Computation, MPC), 신뢰실행환경(Trusted Execution Environment, TEE) 등이 활용되고 있다.

신뢰실행환경은 타 프로세스, 심지어는 운영체제의 접근을 차단하여 내부에 로드되는 사용자의 코드와 데이터를 보호하는 메인 프로세서의 보안 영역이다. 이중 가장 널리 활용되고 있는 Intel SGX[1]는 Enclave Page Cache(EPC)라는 메모리 영역에 생성한 enclave 에 사용자의 코드와 데이터를 저장한다. 그러나 EPC 는 최대 128MB 의 데이터만 저장할 수 있어서 이를 넘을 경우 내부 데이터의 일부를 암호화하여 EPC 밖의 메모리 영역에 저장하는데, 이 과정에서 많은 페이징이 요구되어 성능이 크게 저하된다. 또한 신뢰실행환경에서 실행되는 프로그램은 일반 CPU 에서 실행될 때보다 성능이 저하된다. 따라서 신뢰실행환경을 활용할 때는 EPC 의 메모리 한계와 성능 저하를 고려하여 프로그램을 설계해야 한다.

본 논문에서는 신뢰실행환경, 특히 SGX 를 활용한 딥러닝 추론에 대한 연구 동향을 서술한다.

2. 신뢰실행환경을 활용한 딥러닝 추론 연구

2-1. Privado[2]



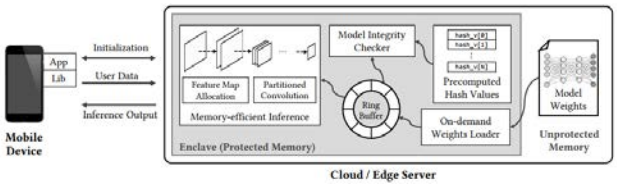
(그림 1) Privado 구조도[2]

본 연구에서는 enclave 메모리 접근 패턴을 통한 부채널 공격을 제시하고, 이에 대한 방어 프레임워크인

Privado 를 제안한다. 모델 소유자가 학습된 모델을 바이너리화하여 입력하면 이를 ONNX[3]로 해석하여 enclave 가 실행할 수 있는 형태의 바이너리로 변환하고 암호화한 모델의 파라미터를 생성한다. 모델 소유자는 생성된 바이너리와 암호화된 파라미터를, 사용자는 암호화한 입력 데이터를 클라우드의 enclave 에 업로드한다. Enclave 내에서는 모델 파라미터와 입력 데이터를 복호화하여 모델 추론을 하고, 추론 결과를 재암호화하여 사용자에게 돌려준다.

Privado 는 암호화/복호화 과정을 포함한 모델 추론 과정에 있어 10 종류의 모델에 대해 순수 CPU 대비 EPC 한계를 넘지 않는 경우(LeNet, VGG19 등)에는 1.01-1.27 배의 성능 향상을 보였고, EPC 한계를 넘는 경우(AlexNet, ResNet50 등)에는 0.15-0.9 배의 성능 저하를 보였다.

2-2. Occlumency[4]



(그림 2) Occlumency 구조도[4]

본 연구는 SGX enclave 의 제한적인 자원을 효율적으로 사용하기 위해 enclave 내에 최적화 모듈을 구현한 연구이다. 대다수의 딥러닝 모델에 대해 enclave 내의 연산 최적화를 하지 않을 경우 추론 시간은 순수 CPU 대비 enclave 가 6.4 배가량 느리며, 학습된 파라미터들은 EPC 한계를 넘는 그 크기로 인해 enclave 에 모두 로드할 수 없음을 문제로 제기했다.

Occlumency 는 enclave 내에 고정 메모리 영역을 두고 외부에 저장되어 있는 모델 파라미터를 해당 영역에 동적으로 불러오는 on-demand weights loader 와 모델 파라미터의 해시를 활용하여 무결성을 검증하는 model integrity checker 를 설계했다. 또한 추론 엔진을 포함한 모듈들을 파이프라이닝하여 효율적으로 실행되게 하기 위한 링 버퍼를 두었다.

모델 추론에는 Caffe[5] 프레임워크를 이용했는데, 순수 CPU 대비 Occlumency 는 AlexNet 과 VGG16/19 등의 모델에 대해서는 0.79-1.13 배의 추가 추론 시간이 필요했고, GoogLeNet, ResNet 에서는 그보다 적은 22%가량의 추가 추론 시간이 필요했다. 메모리는 실제 모델의 전체 파라미터 크기인 27.3-238.1MB 보다 훨씬 적은 공간인 17.1-23MB 활용했고, 에너지 측면에서도 소형 디바이스에서 추론했을 때보다 2.1-11.7 배 효율적이었다.

2-3. Slalom[6]

본 연구는 기존 연구들이 CPU 에서만 모델 추론을 한 것과는 달리 빠른 연산을 위해 GPU 를 활용했다.

Slalom with integrity & privacy

$\mathbf{TEE}(F, x_1)$ $\qquad\qquad$ $\mathcal{S}(F)$

Preproc: $\mathbf{for}\ i \in [1,n]\ \mathbf{do}\ r_i \xleftarrow{R} \mathbb{F}^{m_i}, u_i = r_i W_i$

$\mathbf{for}\ i \in [1,n]\ \mathbf{do}$

$\tilde{x}_i = x_i + r_i$ $\xrightarrow{\tilde{x}_i}$

$\xleftarrow{\tilde{y}_i}$ $\qquad\qquad$ $\tilde{y}_i = \tilde{x}_i W_i$

$y_i = \tilde{y}_i - u_i$

$\mathbf{assert}\ \text{Freivalds}(y_i, x_i, W_i)$

$x_{i+1} = \sigma(y_i)$

$\mathbf{return}\ y_n$

(그림 3) Slalom 알고리즘[6]

CPU 외부 장치인 GPU 로 연산을 하기 위해 프리발즈 알고리즘[7]을 채택했다. 프리발즈 알고리즘은 AxB=C 를 만족하는 행렬 A, B, C 가 있을 때, 정규분포에서 추출하여 생성한 행렬 v 와 B 를 행렬곱한다. GPU 에서 연산한 A 와 B· v 의 행렬곱 연산결과를 돌려받은 후 C· v 와 같은지를 검증한다. 사용자의 데이터가 B 라고 할 때, 사용자의 데이터는 랜덤값과 곱하여 숨길 수 있고, 연산 결과를 검증할 수 있어 GPU 의 연산을 신뢰할 수 있는 형태로 만들어준다.

Slalom 은 사용자가 클라우드의 enclave 에 입력 데이터를 전송하고, enclave 에서 프리발즈 알고리즘을 활용해 GPU 에서 연산을 수행하여 데이터를 보호하면서 빠른 연산 속도를 얻고자 했다. GPU 로 데이터를 보내기 전후로 추가 연산이 요구됨에도 VGG16 과 MobileNet 에 대해 순수 SGX 대비 각 10.4 배, 2.7 배의 성능 향상을 보였다. 또한 에너지 측면에서도 순수 SGX 보다 3.4-17.1 배 더 효율적이었다.

3. 결론

본 논문에서는 신뢰실행환경을 활용한 딥러닝 추론 연구 동향을 살펴보았다. 신뢰실행환경의 제한적인 자원을 효율적으로 활용하기 위해 신뢰실행환경 내부에 저장될 데이터를 최소화하거나 필요시 외부 장치를 이용하되 데이터를 보호하는 기술을 추가 적용했음을 알 수 있었다. 신뢰실행환경의 한계 및 딥러닝 연산의 특성을 주어진 요구에 맞춰 고려하면 안전하면서도 효율적인 딥러닝 추론이 가능할 것으로 기대된다.

4. ACKNOWLEDGEMENT

이 논문은 2022 년도 BK21 FOUR 정보기술 미래인재 교육연구단에 의하여 지원되었음.

참고문헌

[1] V. Costan and S. Devadas, “Intel sgx explained,” Cryptology ePrint Archive, 2016.

[2] Grover, K., Tople, S., Shinde, S., Bhagwan, R., & Ramjee, R. “Privado: Practical and secure DNN inference with enclaves.” arXiv preprint arXiv:1810.00602. 2018.

[3] Onnx, “Open neural network exchange format,” https: //onnx.ai/

[4] Lee, T., Lin, Z., Pushp, S., Li, C., Liu, Y., Lee, Y., ... & Song, J. “Occlumency: Privacy-preserving remote deep-learning inference using SGX.” In The 25th Annual International Conference on Mobile Computing and Networking, pp. 1-17, 2019.

[5] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. “Caffe: Convolutional Architecture for Fast Feature Embedding.” In Proceedings of the 22Nd ACM International Conference on Multimedia (MM ’14). ACM, New York, NY, USA, 675–678. 2014.

[6] F. Tramer and D. Boneh, “Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware,” in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.

[7] Freivalds, R. “Fast probabilistic algorithms.” In International Symposium on Mathematical Foundations of Computer Science (pp. 57-69). Springer, Berlin, Heidelberg. 1979.