딥러닝 기술을 적용한 난수 생성기 연구 동향

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

Research Trends of Random Number Generators using Deep Learning

Hyun-Ji Kim[1], Se-Jin Lim[1], Hwa-Jeong Seo[1]
[1]Dept. of IT convergence engineering, Han-Sung University

요 약

암호화 프로그램에서 난수생성기는 널리 사용되며 중요한 역할을 하므로 공격의 대상이 되기 쉽고, 따라서 높은 난수성을 확보해야 한다. 최근에는 인공 신경망 기술이 발달함에 따라 난수생성기에 딥러닝 기술을 적용하는 연구들이 다수 진행되었으며, 본 논문에서는 이러한 연구 동향에 대해 알아본다. 크게 난수를 생성하는 연구와 다음에 올 수를 예측하는 예측 공격으로 나뉜다. 공통적으로는 학습해야 할 대상인 난수가 시계열 데이터이므로 대부분의 연구들이 RNN, CNN-1D 신경망을 사용한다. 난수 생성을 위해서는 분류형 신경망이 아닌, 생성형 신경망과 강화학습을 주로 사용하였다. 대부분의 연구들이 NIST SP-800 테스트를 시행하였을 때 높은 난수성을 확보할 수 있었다. 이외에도 최근 양자 컴퓨터가 개발됨에 따라 양자 하드웨어로부터의 양자 난수 생성기에 대한 예측 공격에 관한 연구도 있다. 딥러닝 기반의 난수 생성기에 대해서, 향후에는 기존의 난수생성기보다 빠른 생성 속도를 달성할 수 있는 경량 구현에 대한 연구와 그에 대한 비교 및 평가가 있어야 할 것으로 생각된다.

1. 서론

암호화 프로그램에서 난수생성기는 널리 사용되고 중요한 역할을 하기 때문에 공격의 대상이 되기 쉽다. 따라서 높은 난수성을 갖는 PRNG를 구현하는 것은 실제 응용프로그램에 있어 중요한 부분이다. 이러한 난수생성기에 딥러닝 기술을 활용하는 연구가 최근 증가하고 있으며, 본 논문에서는 이러한 연구들에 대해 알아본다.

2. 관련 연구

2.1 인공 신경망

인공 신경망은 여러 노드로 구성된 레이어가 여러 층으로 쌓인 구조를 가지며, 각 노드들의 값은 해당 노드와 연결된 이전 레이어의 노드 값 및 가중치를 곱하고 모두 더한 후 활성화 함수를 거쳐 하나의 값으로 계산된다. 입력층에 데이터를 입력하면 위와 같은 과정을 거쳐 출력층까지 연산을 수행한 후, 최종 출력 값은 손실함수에 입력되어 실제 정답과의 차이인 손실을 구한다. 이후, 해당 손실 값을 최소화하기 위한 방향으로 학습한다. 이러한 신경망은 입력 데이터에 따라 효과적인 구조가 존재하며, 대표적으로 이미지 또는 시계열 데이터 학습에 좋은 Convolutional Neural Network (CNN), 시계열 학습에 효과적인 Recurrent Neural Network (RNN), 데이터 생성을 위한 Generative Adversarial Network (GAN) 등이 있다.

2.1.1 Generative Adversarial Network (GAN)

GAN은 generator와 discriminator라는 두 모델로 구성되어 있으며, 각 모델은 하나의 신경망이다. generator는 랜덤 시드 값을 입력으로 받아 데이터를 생성해내는 부분이고, discriminator는 generator가 생성한 데이터와 실제 학습데이터를 모두 입력으로 받아 어떤 데이터가 실제 데이터이고 어떤 데이터가 generator가 생성한 데이터인지를 구별하는 역할을 한다. 따라서 discriminator가 구별을 잘 하게 될수록 generator는 이를 속이기 위해 더 진짜 같은 데이터를 생성하게 된다. 즉, 두 신경망이 서로 경쟁하며 학습하는 구조이다.

2.1.2 강화학습

강화학습은 보상 개념을 활용해서 상태를 지속적으로 탐험하고 agent가 현재 상태에서의 최적의 수를 행하도록 하는 학습법이다. 다시 말해서, 지도 및 비지도 학습은 데이터의 특징을 찾는 것이지만, 강화학습은 어떻게 행동할지를 가르쳐서 최적의 결과를 얻을 수 있도록 하는 학습기법이다.

2.2 난수 생성기

난수 생성기에는 진정한 난수 생성기 (True Random Number Generator, TRNG), 의사 난수 생성기 (Pseudo Random Number Generator, PRNG), 그리고 양자 역학적 성질을 활용한 양자 난수 생성기 (Quantum Random Number Generator, QRNG)가 있다. 진정한 난수 생성기는 하드웨어 잡음원 등에서 노이즈를 수집하여 난수를 생성하는 것으로 실제적인 난수이다. 그에 비해 의사 난수 생성기는 랜덤으로 보이는 난수열을 생성한다. 즉, 실제적인 난수는 아니지만 난수로 보이는 (난수성을 지닌) 수들을 생성하는 것이다.

3. 연구 동향

본 논문에서는 딥러닝 기술을 난수 생성기에 활용한 연구 사례에 대해 알아본다. 이러한 연구들은 딥러닝 기술을 활용하여 난수 생성기가 생성해내는 난수열에서 다음에 올 수를 예측하거나 난수를 생성해내는 것을 목표로 한다. 표 1은 연구 동향을 정리한 표이다. 크게 PRNG를 생성해내는 연구와 QRNG를 예측하는 연구로 나뉘며, 난수를 생성해내기 위해서는 데이터를 생성해낼 수 있는 신경망 모델을 사용하였다. 또한, 해당 연구들은 난수를 대상으로 하므로 시계열 데이터 처리에 적합한 RNN, CNN 1-dimension 등의 레이어를 주로 사용하였다.

표 1. 난수생성기에 대한 연구 동향

	[1]	[2]	[3]	[4]
Target	PRNG			QRNG
Task	Generation			Prediction
Method	Generative Adversarial Networks		Reinforce learning	Recurrent Convolutional Network

[1]에서는 딥러닝 모델 중 하나인 GAN을 기반으로 하는 의사 난수 생성기를 제안하였다. GAN의 generator 모델은 난수를 생성하고, discriminator 부분은 predictor로 변형되어 generator가 생성한 난수열 중 마지막 정수를 예측하도록 한다. Predictor는 generator의 출력을 입력으로 사용하며, 8개의 정수열마다 7개와 1개로 나누어 마지막 1개의 정수를 label로 사용하고, 앞의 7개의 정수를 학습 데이터로 사용한다. 이 경우, 별도의 실제 데이터가 필요하지 않다는 장점이 있다. 다시 말해서, 7개의 난수를 학습데이터로 하여 마지막 8번째 정수를 예측하는 것이며, 이때 8번째 정수의 실제 값은 generator가 생성한 값이고 predictor는 해당 값을 label로 활용하여 값을 예측하도록 학습하는 것이다. 이때 generator가 생성하는 값들은 0~65535 사이의 정수이다. 또한, generator는 predictor가 예측하는 값들을 활용하여 학습한다. 즉, predictor가 어떠한 난수는 학습을 해서 제대로 예측하고 어떤 난수는 학습하지 못하여 예측에 실패하는지를 활용해서, 예측할 수 없는 난수열을 생성해내도록 학습하는 것이다. 이를 위해서 generator와 predictor의 손실함수가 서로 다르게 설정된다. 해당 연구에서는 generator를 위해서는 Fully-connected layer, predictor를 위해서는 CNN 1-dimension layer를 사용하였고, 200000 epoch만큼 학습하였다. 이러한 방법을 통해 생성된 난수들이 실제 난수로 사용될 수 있는지에 대해 검증하기 위해 NIST의 PRNG에 대한 난수성 검사(통계적 테스트)를 수행하였으며, 여기에는 0과 1의 빈도수 검사, 특정 패턴의 등장에 대한 검사 등의 15개의 테스트가 포함되며 각 테스트는 여러 번 시행된다. 이를 테스트 인스턴스라고 하며, 특정 테스트는 입력 데이터의 길이에 따라 인스턴스의 수가 다르고 그 외 나머지 테스트는 1개 당 10번 정도 반복된다. 총 188개의 테스트 인스턴스에 대해 약 3%의 테스트 인스턴스만 실패하였으며, 전체적으로 실패한 테스트의 수는 평균적으로 4.5개를 달성하여 성공적으로 난수를 생성할 수 있음을 보였다.

[2]에서는 [1]의 연구를 기반으로 generator-predictor 구조를 사용하였다. 그러나, [1]보다 더 높은 난수성과 더 효율적인 구조를 제안하였다. 해당 연구에서는 생성하는 데이터 타입을 비트로 하였으며 CNN 1-dimension과 더불어 시계열 학습에 더 좋은 RNN을 사용하였다. 이를 통해 더 긴 길이의 데이터를 생성해낼 수 있었다. 즉, generator에 입력되는 동일한 64-bit seed에 대해 837056-bit를 더 생성할 수 있으며, 30 epoch만을 사용하여 학습하였다. 또한, 저전력 임베디드 기기에서의 추론이 가능하도록 하였고, 이를 효율적으로 활용하기 위해 predictor에 시계열 신경망을 주로 사용하여 성능을 향상시켰고, 실제 임베디드

기기에 배포하는 generator에는 fully connected layer만을 사용하였다. 해당 연구의 경우에도 NIST 난수성 테스트인 SP 800-22를 시행하였고, 그 결과 실패한 테스트 인스턴스의 비율이 1.09로 [1]에 비해 1.91% 감소하였다. 또한, 전체에서 실패한 테스트의 수는 0개 (테스트 인스턴스와 다름)를 달성하여 더 높은 난수성을 달성하였음을 보였다.

[3]에서는 강화학습 (Reinforce learning)을 통한 난수 생성기를 제안하였다. 해당 연구에서는 강화학습을 사용하였으므로 크게 두 개의 부분으로 나뉜다. 환경 부분은 agent의 행동 별 상태 공간, 보상 정책, 평가로 구성되고, agent 부분은 시계열 학습을 위한 LSTM, 해당 모델에서 생성된 특징벡터, 특징 벡터를 시간 순으로 누적하여 이미지로 학습하는 CNN, Fully connnected layer로 나뉜다. 우선, 행동별 상태 공간은 딥러닝을 통해 생성한 동작을 다음 상태로 변환하는 부분이고, 평가 함수는 다음 상태 (난수)의 임의성을 점수로 변환한 후 다음 행동의 보상으로 사용한다. 즉, 보상이 높을수록 난수성이 높음을 의미한다. agent 부분에서는 이전 데이터 패턴을 장기적으로 기억하고 해당 패턴을 고려하기에 적합한 Long short term memory (LSTM) 모델을 사용하였으며, 이에 더불어 CNN을 사용하였다. LSTM에서 추출한 특징 벡터를 누적해서 이미지를 생성하며, 이를 이미지 형태로 학습해야 하므로 CNN을 추가적으로 사용하였다. 또한, LSTM으로부터 생성된 특징벡터 1과 CNN으로부터 생성된 특징벡터 2를 합쳐 특징벡터 3을 생성하며, Fully-connected layer에 입력되어 최종적으로 800-bit의 난수열이 생성된다. 해당 연구에서도 NIST SP 800-22를 사용하였을 때, 평균적인 보상의 가치가 충분히 높으므로 난수성을 달성하였음을 보였다.

이외에도 난수를 생성하는 것이 아니라 다음에 올 수를 예측하는 양자 난수 생성기에 대한 예측 연구 [4] 등이 있으며, 해당 연구에서는 양자 하드웨어로부터 난수 데이터를 수집하고, Recurrent Convolutional Neural Network (RCNN)을 통해 생성된 난수의 시퀀스에서 잠재적 특징을 파악하여 다음에 올 난수를 예측하도록 하였다.

4. 결론

본 논문에서는 딥러닝 기술이 적용된 난수 생성기에 대한 연구 동향을 살펴보았다. 이러한 연구들은 크게 난수를 생성하는 것과 다음에 올 수를 예측하는 연구로 나뉜다. 난수 생성을 위해서는 난수가 시계열 데이터이므로 주로 RNN, CNN-1D을 사용하였으며, 난수 생성을 위한 연구에서는 분류형 신경망이 아닌, 생성형 신경망인 GAN과 agent가 최적 행동을 하도록 하는 강화학습이 주로 활용되었으며, 모든 연구가 NIST SP-800을 만족하는 높은 난수성을 확보하였다. 딥러닝 기반의 난수 생성기에 관한 연구들은 향후에는 기존의 난수생성기보다 더 빠른 속도로 난수를 생성할 수 있도록 경량화 된 신경망 구현과 그에 대한 평가가 필요할 것으로 생각된다.

5. Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2018-0-00264, Research on Blockchain Security Technology for IoT Services).

참고문헌

[1] Bernardi, Marcello De, M. H. R. Khouzani, and Pasquale Malacaria. "Pseudo-random number generation using generative adversarial networks." Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham, 2018.

[2] Kim, Hyunji, et al. "Generative Adversarial Networks-Based Pseudo-Random Number Generator for Embedded Processors." International Conference on Information Security and Cryptology. Springer, Cham, 2020.

[3] Park, Sungju, et al. "Dynamical Pseudo-Random Number Generator Using Reinforcement Learning." Applied Sciences 12.7 (2022): 3377.

[4] Truong, Nhan Duy, et al. "Machine learning cryptanalysis of a quantum random number generator." IEEE Transactions on Information Forensics and Security 14.2 (2018): 403-414.