https://doi.org/10.33778/kcsa.2021.21.2.057

딥뉴럴네트워크에서의 적대적 샘플에 관한 앙상블 방어 연구★

권 현*, 윤 준 혁**, 김 준 섭***, 박 상 준****, 김 용 철*****

요 약

딥뉴럴네트워크는 이미지 인식, 음성 인식, 패턴 인식 등에 좋은 성능을 보여주고 있는 대표적인 딥러닝모델 중에 하나이다. 하지만 이러한 딥뉴럴네트워크는 적대적 샘플을 오인식하는 취약점이 있다. 적대적 샘플은 원본 데이터에 최소한의 노이즈를 추가하여 사람이 보기에는 이상이 없지만 딥뉴럴네트워크가 잘못 인식 하게 하는 샘플을 의미한다. 이러한 적대적 샘플은 딥뉴럴네트워크를 활용하는 자율주행차량이나 의료사업에서 차량 표지판 오인식이나 환자 진단의 오인식을 일으키면 큰 사고가 일어나기 때문에 적대적 샘플 공격에 대한 방어연구가 요구된다. 본 논문에서는 여러 가지 파라미터를 조절하여 적대적 샘플에 대한 앙상블 방어방법을 실험적으로 분석하였다. 적대적 샘플의 생성방법으로 fast gradient sign method, DeepFool method, Carlini & Wanger method을 이용하여 앙상블 방어방법의 성능을 분석하였다. 실험 데이터로 MNIST 데이터셋을 사용하였으며, 머신러닝 라이브러리로는 텐서플로우를 사용하였다. 실험방법의 각 파라미터들로 3가지 적대적 샘플 공격방법, 적정기준선, 모델 수, 랜덤노이즈에 따른 성능을 분석하였다. 실험결과로 앙상블 방어방법은 모델수가 7이고 적정기준선이 1일 때, 적대적 샘플에 대한 탐지 성공률 98.3%이고 원본샘플의 99.2% 정확도를 유지하는 성능을 보였다.

Detecting Adversarial Example Using Ensemble Method on Deep Neural Network

Hyun Kwon*, Joonhyeok Yoon**, Junseob Kim***, Sangjun Park****, Yongchul Kim*****

ABSTRACT

Deep neural networks (DNNs) provide excellent performance for image, speech, and pattern recognition. However, DNNs sometimes misrecognize certain adversarial examples. An adversarial example is a sample that adds optimized noise to the original data, which makes the DNN erroneously misclassified, although there is nothing wrong with the human eye. Therefore studies on defense against adversarial example attacks are required. In this paper, we have experimentally analyzed the success rate of detection for adversarial examples by adjusting various parameters. The performance of the ensemble defense method was analyzed using fast gradient sign method, DeepFool method, Carlini & Wanger method, which are adversarial example attack methods. Moreover, we used MNIST as experimental data and Tensorflow as a machine learning library. As an experimental method, we carried out performance analysis based on three adversarial example attack methods, threshold, number of models, and random noise. As a result, when there were 7 models and a threshold of 1, the detection rate for adversarial example is 98.3%, and the accuracy of 99.2% of the original sample is maintained.

Key words : Machine learning, Evasion attack, Neural network

접수일(2021년 04월 15일), 수정일(2021년 06월 18일), 게재확정일(2021년 06월 28일)

★ 본 논문은 화랑대연구소의 2021년도(21-군학-3) 저술활동비 지원을 받아 연구되었음.

* 육군사관학교 전자공학과 조교수(주저자)

** 서울대학교 전기정보공학부 박사과정(공동저자)

*** 육군사관학교 전자공학과 전임강사(공동저자)

**** 육군사관학교 전자공학과 조교수(공동저자)

***** 육군사관학교 전자공학과 교수(교신저자)

1. 서 론

최근 GPU의 등장으로 병렬적인 컴퓨팅 계산능력이 향상이 되고 클라우드 환경과 같이 빅데이터 저장이 가능하게 되면서 딥러닝 모델을 이용한 방법들은 좋은 성능을 보여주고 있다. 특히, 딥러닝 모델 중에 딥뉴럴네트워크[1]는 이미지 인식이나 패턴 분석 등의 예측 및 분류에 있어서 사람보다 더 좋은 성능을 보여주고 있다. 이러한 성능이 좋은 딥뉴럴네트워크를 이용하여 표지판을 인식해야 하는 자율주행차량이나 사람의 눈 등을 진단해야 하는 의료사업 [2]에서 딥뉴럴네트워크를 활용한 연구들이 활발히 진행되고 있다.

그러나 이러한 딥뉴럴네트워크는 보안상 취약점이 존재한다. Barreno et al 연구진[3]에서 딥뉴럴네트워크의 취약점을 이용한 머신러닝 공격방법은 exploratory attack과 causative attack으로 분류하였다. exploratory attack은 공격자가 딥러닝 모델이 학습하는 과정에서 학습 데이터를 조작하여 딥러닝 모델의 성능을 저하시키는 방법이다. 이러한 exploratory attack의 대표적인 예로 중독공격(poisoning attack)[4]이 있다. 이 공격방법은 딥러닝 모델이 학습하는 학습데이터에 접근할 수 있는 전제조건이 필요하다. 반면에 causative attack은 딥러닝 모델이 학습하는 과정에 영향을 미치지 않고 이미 학습이 끝난 딥러닝 모델이 테스트 데이터를 인식할 때, 테스트 데이터를 조작하여 딥러닝 모델이 오인식을 하게 만드는 공격이다. 이러한 causative attack의 대표적인 예로 적대적 샘플(adversarial example)[5]이 있다. 이 공격방법은 테스트 데이터를 조작하기 때문에 exploratory attack보다 현실성이 있다. 본 논문에서는 적대적 샘플 공격에 대하여 다루고자 한다.

적대적 샘플 공격은 원본 데이터에 최소한의 노이즈를 추가하여 사람이 보기에는 노이즈를 식별할 수 없지만 딥뉴럴네트워크는 오인식을 일으키게 하는 공격이다. 이 적대적 샘플 공격은 이미 학습을 마친 딥뉴럴네트워크에 대해서 테스트 데이터를 조작하여 오인식을 일으키는 공격이기 때문에 실현 가능한 실제적인 공격방법이다. 예를 들어, 딥뉴럴네트워크가 장착된 자율주행차량에 대하여 공격자가 도로표지판에 조작을 하여 자율주행차량으로 하여금 오인식하게 만들 수 있다. 좌회전 도로표지판을 공격자가 우회전으로 오인식하게 하도록 최소화된 노이즈를 도로표지판에 추가하면, 사람이 보기에는 좌회전이지만 자율주행차량은 우회전으로 오인식하게 만들 수 있다. 이러한 적대적 샘플은 인공지능 분야와 보안 분야에서 다양한 공격연구들과 방어연구들에 이뤄지고 있다.

적대적 샘플에 대한 방어연구들[6][7][8]은 주로 테스트 데이터를 조작하거나 강건한 딥뉴럴네트워크를 설계하여 방어하는 연구들이 있다. 테스트 데이터를 조작할 경우, 별도의 모듈이 필요하고 훈련하는 과정이 요구된다. 또한, 강건한 딥뉴럴네트워크 설계의 경우, 적대적 샘플을 학습하는 과정이 필요하거나 별도의 뉴럴네트워크가 요구된다. 하지만 이 방어방법은 적대적 샘플을 학습하는 과정에서 원본 샘플의 성능이 저하되거나 화이트박스 공격 등에 취약점이 있다. 이러한 부분을 보완하기 위해서 기존에 갖고 있는 여러 개의 딥뉴럴네트워크를 활용하여 각 클래스의 평균 순위 점수를 비교하여 적대적 샘플을 탐지하는 앙상블 방어방법이 하나의 솔루션이 될 수 있다.

본 논문에서는 적대적 샘플에 대한 앙상블 방어방법에 대하여 분석하였다. 이 방어방법은 원본 샘플과 적대적 샘플간에 대하여 여러개(앙상블)의 딥뉴럴네트워크에서 각각 인식되는 클래스의 순위 점수를 측정하여 가장 낮은 평균 순위 점수인 샘플은 적대적 샘플로 탐지하는 방법이다. 이 방법을 적용하기 위해서 여러 딥뉴럴네트워크는 학습단계에서 random noise가 포함된 샘플을 추가적으로 학습하는 프로세스를 가진다. 본 논문의 공헌점은 다음과 같다. 첫째, 여러 딥뉴럴네트워크를 이용한 방어방법에서 여러 파라미터를 조절하여 적대적 샘플의 탐지율을 분석하였다. 또한, 앙상블 방법에서 평균 순위 점수에 의한 원리와 시스템적인 체계에 대하여 제시하였다. 두 번째, 적대적 샘플을 생성하는 fast gradient sign method (FGSM)[9], DeepFool[10], Carlini and Wagner (C&

W) 방법[11]에 대해서 적대적 샘플에 대한 탐지율을 측정하였다. 세 번째로 MNIST 데이터셋[12]을 이용하여 제안 방법의 성능을 보였다. 또한, 방어방법에 사용되는 파라미터인 딥뉴럴네트워크의 수, 적정기준선(threshold) 수치, 랜덤노이즈(random noise) 수치 등에 의한 제안 방법의 성능을 다각적으로 분석하였다.

이 장의 나머지 구성은 다음과 같다. 2장에서는 관련연구에 대한 소개를 하고 3장에서는 제안방법에 대한 구조와 설명을 한다. 4장에서는 실험 및 분석을 하고 5장은 결론으로 구성되어 있다.

2. 관련연구

이 장에서는 적대적 샘플 공격방법과 방어방법에 대하여 소개하고자 한다. 공격방법과 방어방법을 설명하기 앞서서 적대적 샘플은 크게 공격 목적에 따른 분류와 공격하고자 하는 딥뉴럴네트워크에 대한 정보의 양으로 크게 구분할 수 있다. 먼저, 적대적 샘플은 공격 목적에 따라 목표 적대적 샘플(Targeted adversarial example)과 비목표 적대적 샘플(Untargeted adversarial example)이 있다. 목표 적대적 샘플은 공격자가 정한 특정 클래스로 딥뉴럴네트워크가 오인식하게 하는 샘플을 의미한다. 반면에 비목표 적대적 샘플은 원본 클래스가 아닌 임의의 클래스로 오인식하게 하는 샘플을 의미한다. 목표 적대적 샘플은 비목표 적대적 샘플보다 공격자가 오인식할 클래스를 선택해야 하기 때문에 좀 더 정교한 공격샘플이지만 반면에 생성과정이 오래 걸리고 왜곡이 좀 더 많은 특징이 있다.

두 번째로, 공격하고자 하는 딥뉴럴네트워크에 대해서 화이트 박스 공격과 블랙 박스 공격으로 구분된다. 화이트 박스 공격은 공격자가 딥뉴럴네트워크에 대한 모든 정보를 알고 있는 환경에서의 공격을 의미한다. 반면에 블랙 박스 공격은 공격자가 딥뉴럴네트워크에 대한 정보가 없는 상황에서의 공격을 의미한다. 본 연구에서는 왜곡이 적은 비목표 적대적 샘플와 블랙 박스 공격으로 생성된 적대적 샘플에 대하여 방어연구를 하였다.

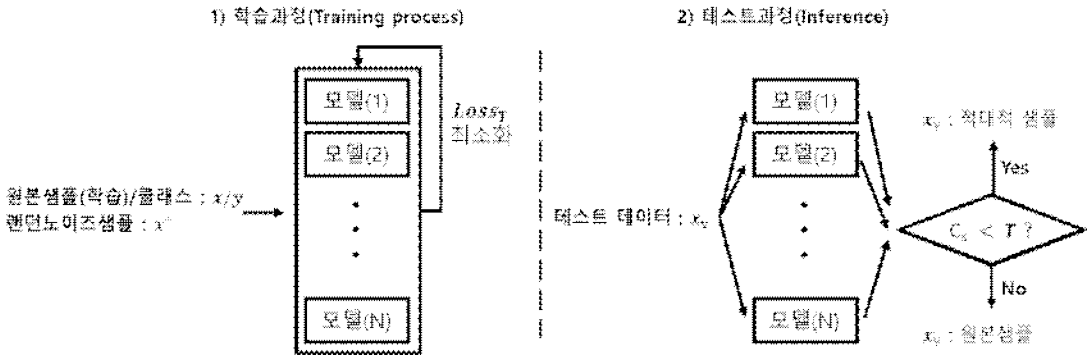
2.1 적대적 샘플 공격 연구

적대적 샘플을 생성하는 대표적인 방법으로 fast gradient sign method (FGSM)[9], DeepFool[10], Carlini and Wagner (C&W)[11]이 있다. 먼저, FGSM 방법은 딥뉴럴네트워크에서 gradient을 계산하여 최적으로 공격할 수 있는 적대적 샘플을 찾는 방법이다. 생성속도가 빠른 장점이 있지만 비목표 적대적 샘플공격만 가능한 특징이 있다. 두 번째로, DeepFool 방법은 FGSM보다 개선된 방법이지만 선형근사방식으로 적대적 샘플을 생성하기 때문에 많은 반복횟수가 요구되는 단점이 있다. 세 번째로, C&W 방법은 공격성공률과 최소한의 왜곡을 조절하여 최적의 왜곡으로 100% 공격성공률을 생성하는 방법이다. 본 논문에서는 FGSM, DeepFool, C&W 방법에 대하여 적대적 공격샘플을 생성하여 실험에 적용하였다.

2.2 적대적 샘플 방어연구

적대적 샘플의 방어연구는 크게 입력데이터를 조작 또는 검사하는 방법과 딥뉴럴네트워크를 강건하게 만드는 두가지 측면에서 볼 수 있다. 먼저, 이미 학습이 끝난 딥뉴럴네트워크가 테스트 데이터를 입력값으로 받을 때, 테스트 데이터를 조작하거나 검사를 하여서 탐지하는 방법[6]이 있다. 이 방법은 딥뉴럴네트워크가 입력값으로 받기 전에 사전에 특정 모듈을 거치게 하여서 적대적 샘플의 노이즈를 제거하거나 패턴을 분석하거나 feature squeeze [7]등으로 일부 조작 등을 하여서 딥뉴럴네트워크가 제대로 인식하게 하는 방법이다. 이러한 방법은 별도의 모듈을 따로 학습하는 과정이 필요한 특징이 있다.

두 번째로, 딥뉴럴네트워크를 강건하게 만드는 방법들이 있다. 이 방법은 적대적 샘플 학습방법(adversarial training)[8], distilation 방법[13] 등을 통해서 적대적 샘플에 대해서 강건하게 만든다. 적대적 샘플 학습방법은 딥뉴럴네트워크에 원본 학습 데이터 뿐만 아니라 임의로 만든 적대적 샘플을 학습하여 적대적 샘플에 대하여 강건하게 만드는 방법이다. 간단한 방어방법이지만 딥뉴럴



(그림 1) 적대적 샘플에 관한 앙상블 방어방법

네트워크의 성능이 저하될 수 있기 때문에 다소 한계점이 있다. 반면에 distilation 방법은 두 개의 딥뉴럴네트워크를 이용하여 적대적 샘플을 방어하는 방법이다. 1차 딥뉴럴네트워크를 통해서 테스트 데이터에 대한 확률값을 계산 한 후에 2차 딥뉴럴네트워크에서는 테스트 데이터에 대한 확률값을 라벨(label)로 사용한다. 이러한 구조적인 특징은 적대적 샘플을 생성할 때 실제 라벨이 아닌 확률값인 라벨이기 때문에 gradient (기울기) 계산을 어렵게 하여 적대적 샘플을 생성하는 것을 제한시키는 방법이 된다. 하지만 이 distilation 방법도 화이트 박스 공격인 C&W 방법에 무력화되는 한계점이 있다.

위의 언급된 방법과 달리, 테스트 데이터를 조작하는 별도의 모듈이 없고 적대적 샘플 학습방법처럼 원본 샘플에 대한 성능 저하가 일어나지 않는 앙상블 방어방법을 본 논문에서는 적용하였다. 즉, 여러 딥뉴럴네트워크들을 이용하여 앙상블 방어방법을 이용하여 적대적 샘플을 탐지하는 방법을 적용하였다. 논문에서 사용하고자 하는 앙상블 방어방법은 여러 딥뉴럴네트워크에서 나온 각 클래스의 평균 순위점수를 비교하여 평균 순위점수가 작은 경우, 적대적 샘플을 간주하여 탐지한다. 자세한 내용은 3장에 설명되어 있다.

3. 연구방법

이 논문의 연구방법으로써, 적대적 샘플에 대한 앙상블 방어방법은 (그림 1)과 같이 크게 2가지로 학습과정과 테스트과정이 이뤄진다. 먼저 학습하는 과정에서 여러 딥뉴럴네트워크인 N개 모델들이 원본 샘플(original sample)을 원본 클래스(original class)로 제대로 인식하는 것과 랜덤한 노이즈가 추가된 샘플(random perturbation sample)에 대해서 불일치율이 높게 나오도록 손실함수(loss function)를 구성하여 학습을 한다. 그리고 난 후, 테스트과정에서는 임의의 테스트 데이터에 대하여 N개 모델에서 나온 각 클래스의 평균 순위점수를 보고 가장 일치율(agreement)이 작은 것은 적대적 샘플로 간주하여 적대적 샘플로 탐지하고, 가장 일치율이 높은 것은 원본 샘플로 간주한다.

앙상블 방법에 대해서 구체적으로 살펴보면, 먼저 학습과정에서, 원본 샘플(x)과 원본 클래스(y)가 주어졌을 때, 전체 손실함수(loss function, $Loss_T$)를 최소화하여 학습을 한다. 전체 손실함수($Loss_T$)는 원본 샘플을 제대로 인식할 수 있는 인식 손실함수($Loss_r$)와 적대적 샘플에 대하여 불일치를 증가시킬 수 있는 일치 손실함수($Loss_a$)로 구성이 된다.

$$Loss_T = Loss_r + Loss_a$$

인식 손실함수($Loss_r$)는 원본 샘플과 원본 클래스가 cross-entropy loss[14]를 통해서 원본 샘플과 유사한 샘플이 왔을 때, 원본 클래스가 높은 class confidence로 나오도록 한다.

$$Loss_r = J_r(x,y,\sigma(x)) = -\frac{1}{N}\sum_{i=1}^{N}\ln(\sigma_n^y(x)),$$

여기서, N은 모델 수를 의미하고 $J_r(\bullet)$는 기본적인 cross-entropy error를 의미하고, $\sigma(\bullet)$는 입력값에 대한 softmax layer에서 출력되는 벡터값이다. softmax layer 함수를 수식적으로 표현하면 아래와 같다.

$$\sigma(x)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad (j=1,...,K)$$

일치 손실함수($Loss_a$)는 랜덤노이즈샘플(x^*)에 대하여 임의의 두 개의 모델에서 나온 평균 일치율을 나타내며, 일치 손실함수를 줄일수록 랜덤노이즈샘플과 유사한 샘플은 일치율이 적게 나오도록 역할을 한다.

$$Loss_a = \lambda J_a(x^*,\sigma(x^*))$$
$$= \lambda\binom{N}{2}^{-1}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sigma_i(x^*)^T\sigma_j(x^*),$$

여기서, λ는 원본 샘플의 정확도(Accuracy)와 랜덤노이즈 샘플의 불일치율(Disagreement) 사이의 trade-off를 조절하는 파라미터이고 실험적으로 0.27일 때 성능이 좋았다. $J_a(\bullet)$는 임의의 두 개 모델에서 softmax 출력 벡터들의 곱을 통해서 전체 평균 일치율을 계산하는 함수이고, x^*는 랜덤노이즈샘플로 원본샘플 x와 노이즈 w를 더한 값이다.

$$x^* = x + w,$$

여기서, 노이즈 w는 uniformly random 노이즈로 노이즈 벡터를 조절할 수 있다.

각 모델에서 전체 손실함수($Loss_T$)를 최소화함으로써, 여러 개의 모델은 원본 샘플을 원본 클래스로 제대로 인식하면서 랜덤노이즈샘플에 대하여 불일치율이 증가가 되도록 학습이 된다.

두 번째로, 테스트과정에서 새로운 데이터가 입력값으로 들어왔을 때, 여러 딥뉴럴네트워크가 도출한 각 클래스의 평균 순위 점수를 토대로 평균 순위 점수가 가장 높은 클래스의 평균 순위점수(C_s)을 구한다. 예를 들어, 7개 딥뉴럴네트워크마다 각각 0부터 9 클래스 중 softmax layer에서 높은 확률의 값을 갖는 클래스 순으로 10부터 1순으로 순위 점수를 부여한다. 그 후에 7개의 딥뉴럴네트워크마다 각 클래스마다의 평균 순위 점수를 구한다. 각 클래스의 평균 순위 점수 중 가장 높은 순위점수의 평균값(C_s)를 구한다. C_s값이 높은 것은 원본샘플로 간주하고 C_s값이 적은 것은 적대적 샘플로 간주한다. 이 때, 원본 샘플의 정확도를 높이면서 적대적 샘플을 제대로 탐지하는 적정기준선을 선정해야 한다. 적정기준선 및 적대적 샘플에 대한 탐지율 등 앙상블 방어방법의 성능에 대하여 4장에서 결과를 보여준다.

4. 검증 및 결과

제안방법의 앙상블 순위 점수에 의한 탐지 성능을 보여주기 위하여, 실험환경은 대표적으로 많이 사용되고 있는 텐서플로우 머신러닝 라이브러리[15]를 사용하였으며, 서버는 Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz와 GPU는 GeForce GTX 1050을 사용하였다.

4.1 데이터셋

데이터셋은 MNIST[7]를 사용하였다. MNIST은 대표적인 손글씨 이미지셋으로 0부터 9까지의 흑백 이미지로 구성되어 있다. 픽셀 구성은 행렬식 구조로 (1×28×28)로 구성되어 있으며 총 784개의 픽셀로 구성되어 있다. MNIST의 데이터양은 6만개의 학습데이터(Training data)와 1만개의 테스트 데이터(Test data)를 가진다.

4.2 딥뉴럴네트워크

여러 뉴럴네트워크는 fully connected network [16]으로 1개의 입력층(input layer)는 28 뉴런으로 되어 있으며, 1개의 은밀층(hidden layer)은 128 뉴런으로 구성되어 있고 결과층(output layer)는 10개 뉴런인 softmax로 10개 클래스로 구성되어 있다. 활성화 함수(Activation function)는 ReLU를 사용하였다. 표1과 같이, 7개의 모델은 같은 뉴럴네트워크를 구조를 갖지만 학습데이터를 달리

비율을 조정하여 각각 서로 다르게 구성하였다. 최적화 알고리즘은 Adam 알고리즘을 사용하였고 여러 딥뉴럴네트워크에 대한 파라미터는 표2와 같이 구성되어 있다.

<표 1> 각 뉴럴네트워크의 훈련데이터의 수

제 원	훈련데이터의 수
모델 1	0~55000
모델 2	5000~60000
모델 3	0~60000
모델 4	0~50000
모델 5	500~55000
모델 6	1000~60000
모델 7	0~45000

<표 2> 각 뉴럴네트워크의 파라미터

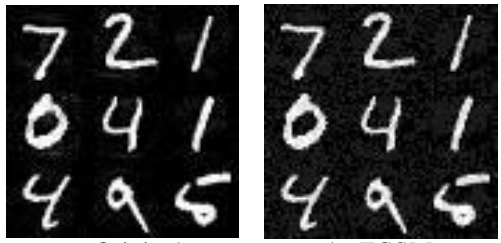
제 원	MNIST
Learning rate	0.1
Momentum	0.9
Delay rate	-
Dropout	0.5
Batch size	128
Epochs	50

4.3 적대적 샘플 생성

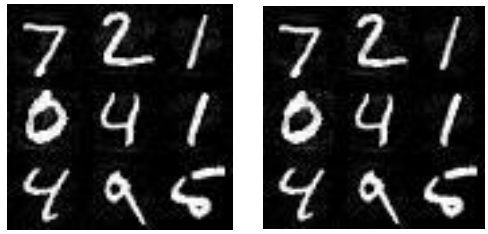
적대적 샘플을 생성하는 방법으로 FGSM, DeepFool, C&W 방법을 사용하였다. FGSM의 경우는 학습률을 0.15로 설정하였고, epsilon의 값을 0.05로 설정하였다. DeepFool와 C&W 방법의 경우, 최적화 알고리즘으로 Adam 알고리즘을 사용하였으며, 학습률 0.1과 상수값 0.01로 설정하였다. 각각의 테스트 데이터를 이용하여 임의의 500개의 적대적 샘플을 생성하여 앙상블 방어방법에 대한 성능을 분석하였다.

4.4 실험결과

정확도(accuracy)는 모델이 어떤 입력 샘플에 대하여 원본 클래스로 제대로 인식되는 비율을 의미한다. 예를 들어, 100개 샘플 중에 90개가 원본 클래스로 제대로 인식되었다고 한다면, 정확도는 90%가 된다. 탐지율(Detection rate)는 적대적 샘플이 들어갔을 때, 제안 방법이 적대적 샘플로 제대로 분류한 비율을 의미한다. 예를 들어, 100개 적대적 샘플 중에 95개를 적대적 샘플로 간주하여 탐지하였을 경우, 탐지율은 95%가 된다.



a. Original b. FGSM



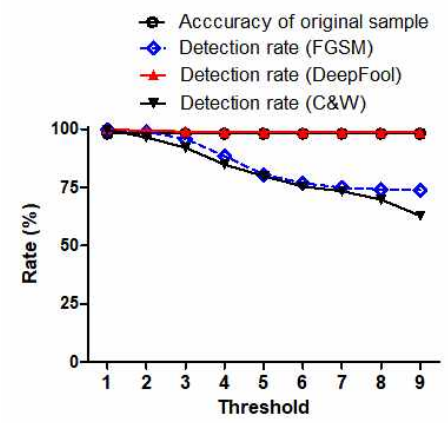
c. DeepFool d. C&W

(그림 2) FGSM, DeepFool, C&W 방법으로 생성한 적대적 샘플 예시

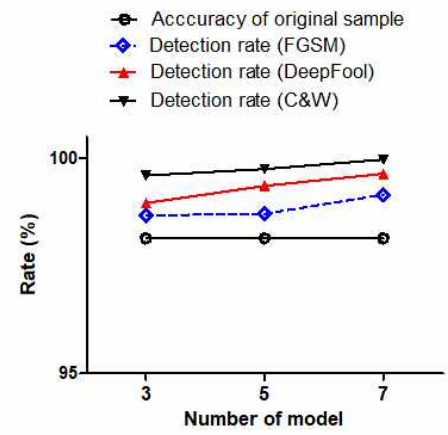
(그림 2)는 FGSM, DeepFool, C&W 방법으로 생성한 적대적 샘플의 예시를 보여준다. 그림에서, 각각의 적대적 샘플은 사람의 눈으로 식별이 어려울 정도에 작은 노이즈가 추가된 것을 볼 수 있다. 특히, DeepFool이나 C&W 방법은 아주 작은 노이즈이기 때문에 FGSM보다 더 원본 샘플과 유사한 것을 볼 수 있다. 이처럼 각각의 방법으로 생성된 적대적 샘플은 딥뉴럴네트워크에 의해 잘못 인식하게 되어 다른 클래스로 인식하게 된다. 예를 들어, <그림 2>의 b에서 숫자 “7”은 사람이 보기에는 숫자 “7”로 보이지만 딥뉴럴네트워크는 “3”으로 잘못 인식하게 된다.

(그림 3)는 앙상블 방어방법을 통해서, 적정기준선(Threshold)에 따른 FGSM, DeepFool, C&W 방법에서 생성된 적대적 샘플의 탐지율과 원본 샘플의 정확도를 보여준다. 여기서, 모델 수는 5개로 선정하였고 딥뉴럴네트워크에 적용되는 랜덤노이즈는 0.1로 하였다. 그림에서 보면 앙상블 방법

을 통해서 원본샘플을 98%이상의 정확도를 유지하면서 FGSM, DeepFool, C&W 방법에서 생성된 적대적 샘플을 탐지하는 것을 볼 수 있다. 특히, 적정기준선이 증가할수록 허용되는 평균 순위점수가 증가되어 원본 샘플을 적대적 샘플로 잘못 탐지하는 비율이 증가하기 때문에 적대적 샘플에 대한 탐지율이 감소하는 것을 볼 수 있다.



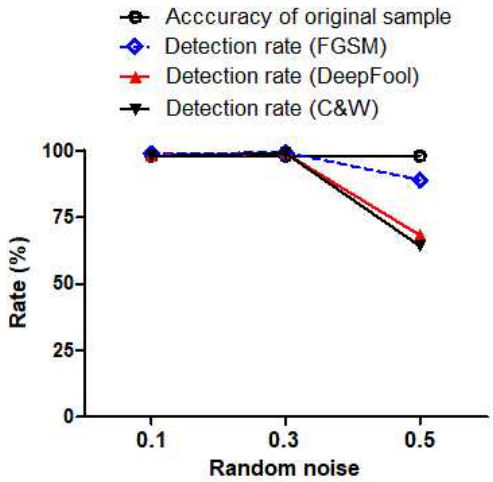
(그림 3) 적정기준선(Threshold)에 따른 FGSM, DeepFool, C&W 방법에서 생성된 적대적 샘플의 탐지율과 원본 샘플의 정확도



(그림 4) 여러 딥뉴럴네트워크(모델) 수에 따른 FGSM, DeepFool, C&W 방법에서 생성된 적대적 샘플의 탐지율과 원본 샘플의 정확도

또한, 노이즈를 컨트롤 하여 공격성공률이 높게 조절할 수 있는 C&W 방법에서 적대적 샘플의 경우, 다른 DeepFool이나 FGSM 방법에서 생성된 적대적 샘플보다 탐지하는 것이 좀 더 어려운 것을 볼 수 있다.

(그림 4)는 여러 딥뉴럴네트워크(모델) 수에 따른 FGSM, DeepFool, C&W 방법에서 생성된 적대적 샘플의 탐지율과 원본 샘플의 정확도를 보여준다. 여기서 랜덤노이즈는 0.1로 하였고 적정기준선은 2로 하였다. 그림에서 보면, 모델수가 증가할수록 적대적 샘플에 대한 탐지율이 거의 비슷하거나 약간씩 증가하는 것을 볼 수 있다. 또한, 원본 샘플의 정확도는 유지가 되는 것을 볼 수 있다.



(그림 5) 랜덤노이즈에 따른 FGSM, DeepFool, C&W 방법에서 생성된 적대적 샘플의 탐지율과 원본 샘플의 정확도

(그림 5)는 딥뉴럴네트워크에 적용되는 랜덤노이즈에 따른 FGSM, DeepFool, C&W 방법에서 생성된 적대적 샘플의 탐지율과 원본 샘플의 정확도를 보여준다. 모델 수는 5로 선정하였고 적정기준선은 2로 하였다. 그림에서보면 딥뉴럴네트워크가 자체적으로 랜덤노이즈샘플을 학습할 때 적용하는 노이즈 범위가 0.1과 0.3일 때에는 앙상블 방어방법이 98%이상의 성능을 보여주었다. 하지만 0.5의 노이즈가 적용되었을 때에는 오히려 적대적 샘플에 대한 탐지율이 저하 되었다. 이는 적대적 샘플이 최소화된 노이즈가 추가된 방법이기 때문에 과도하게 노이즈가 추가된 샘플을 학습하게 되면 상대적으로 적대적 샘플에 대한 탐지율이 떨어지게 된다.

딥뉴럴네트워크 방어 측면에서, 앙상블 방어방법을 적용할 경우, 원본 샘플에 대한 정확도를 유지하면서 적대적 샘플에 대한 상당히 높은 탐지율을 갖는 것을 볼 수 있다. 특히, 적정기준선이 1일 때, F

GSM, DeepFool, C&W 방법에서 생성된 적대적 샘플에 대한 탐지율이 98% 이상을 갖는 것을 볼 수 있었다. 이는 사람의 눈으로 거의 식별이 어려운 적대적 샘플을 앙상블 방어방법을 통해서 적대적 샘플을 탐지할 수 있다.

적정기준선 측면에서, 적정기준선을 작게 잡을수록 적대적 샘플에 대한 탐지가 높은 것을 볼 수 있었다. 왜냐하면 여러 딥뉴럴네트워크가 적대적 샘플로 간주되는 것을 가장 낮은 순위점수를 매기기 때문에 순위 점수를 높이게 되면 원본 샘플을 적대적 샘플로 잘못 탐지하는 비율이 증가하기 때문이다.

모델 수 측면에서, 대체로 여러 딥뉴럴네트워크의 수가 증가할수록 탐지율이 증가하는 것을 볼 수 있었다. 하지만 여러 딥뉴럴네트워크가 증가하더라도 큰 폭의 성능향상이 일어나지 않기 때문에 이를 고려하여 모델 수를 정할 필요가 있다.

랜덤 노이즈 측면에서, 여러 딥뉴럴네트워크가 원본 샘플과 랜덤노이즈샘플을 학습할 때, 랜덤 노이즈의 적정수준의 노이즈를 정해야한다. 실험에서 보면 노이즈의 값이 0.1이거나 0.3일 경우 최소한의 노이즈이기 때문에 적대적 샘플에 대한 탐지율이 좋게 유지되는 것을 볼 수 있다. 하지만 노이즈의 값이 0.5로 증가 되면 상당히 많은 노이즈가 들어간 샘플을 학습하기 때문에 최소한의 노이즈가 반영된 적대적 샘플에 대한 탐지가 줄어들게 된다. 따라서 랜덤노이즈에 대한 적정한 선정이 중요하다.

5. 결 론

본 논문에서는 여러 딥뉴럴네트워크를 앙상블 방식으로 순위점수를 통하여 적대적 샘플을 탐지하는 방법을 분석하였다. 이 방법은 여러 딥뉴럴네트워크가 학습하는 과정에서 사전 원본 샘플과 더불어 랜덤노이즈샘플을 추가적으로 학습함으로써, 여러 딥뉴럴네트워크에서 낮은 평균 순위점수가 나올 경우 적대적 샘플로 간주하여 탐지하는 방법이다. 이는 기존 방어방법과 달리 별도의 모듈이 요구되지 않고 원본샘플의 정확도를 유지한다. 실험결과에서 앙상블 방어방법은 모델수가 7이고 적정기준선이 1일 때, 적대적 샘플에 대한 탐지 성공률 98.3%이고 원본샘플의 99.2% 정확도를 유지한다.

향후 연구로는 이미지 [17] 뿐만 아니라 음성 [18], 텍스트 [19]. 비디오, 백도어 [20] [21] 등의 데이터 연구로 확장하는 연구가 가능하고 여러 가지 방어방법이 혼합된 앙상블 방법에 대한 연구도 흥미로운 주제가 될 것이다. 마지막으로 제안방법은 딥페이크 방어연구로 확장 할 수가 있다.

참고문헌

[1] J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural Netw., vol. 61, pp. 85 - 117, Jan. 2015.

[2] Kleesiek, Jens, et al. "Deep MRI brain extraction: A 3D convolutional neural network for skull stripping." NeuroImage 129 (2016): 460-469.

[3] Barreno, Marco, et al. "The security of machine learning." Machine Learning 81.2 (2010): 121-148.

[4] Biggio, Battista, Blaine Nelson, and Pavel Laskov. "Poisoning attacks against support vector machines." arXiv preprint arXiv:1206.6389 (2012).

[5] C. Szegedy,W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” in Proc. 2nd Int. Conf. Learn. Represent. (ICLR), Banff, AB, Canada, Apr. 2014.

[6] He, Warren, et al. "Adversarial example defense: Ensembles of weak defenses are not strong." 11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17). 2017.

[7] Xu, Weilin, David Evans, and Yanjun Qi. "Feature squeezing: Detecting adversarial examples in deep neural networks." arXiv preprint arXiv:1704.01155 (2017).

[8] Tramèr, Florian, et al. "Ensemble adversarial

training: Attacks and defenses." arXiv preprint arXiv:1705.07204 (2017).

[9] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial machine learning at scale." arXiv preprint arXiv:1611.01236 (2016).

[10] Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[11] Carlini, Nicholas, and David Wagner. "Towards evaluating the robustness of neural networks." 2017 ieee symposium on security and privacy (sp). IEEE, 2017.

[12] Y. LeCun, C. Cortes, and C. J. Burges. (2010). Mnist Handwritten Digit Database. AT&T Labs. [Online]. Available: http://yann. lecun.com/exdb/mnist

[13] Papernot, Nicolas, et al. "Distillation as a defense to adversarial perturbations against deep neural networks." 2016 IEEE Symposium on Security and Privacy (SP). IEEE, 2016.

[14] Nasr, George E., E. A. Badr, and C. Joun. "Cross entropy error function in neural networks: Forecasting gasoline demand." FLAIRS conference. 2002.

[15] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 2016.

[16] Li, Jiahao, et al. "Fully connected network-based intra prediction for image coding." IEEE Transactions on Image Processing 27.7 (2018): 3236-3247.

[17] Kwon, Hyun, et al. "Classification score approach for detecting adversarial example in deep neural network." Multimedia Tools and Applications 80.7 (2021): 10339-10360.

[18] Kwon, Hyun, et al. "Selective audio adversarial example in evasion attack on speech recognition system." IEEE Transactions on Information Forensics and Security 15 (2019): 526-538.

[19] Kwon, Hyun. "Friend-Guard Textfooler Attack on Text Classification System." IEEE Access (2021).

[20] Kwon, Hyun. "Detecting Backdoor Attacks via Class Difference in Deep Neural Networks." IEEE Access 8 (2020): 191049-191056.

[21] Kwon, Hyun, Hyunsoo Yoon, and Ki-Woong Park. "Multi-targeted backdoor: Indentifying backdoor attack for multiple deep neural networks." IEICE Transactions on Information and Systems 103.4 (2020): 883-887.

〔저자소개〕

권 현 (Hyun Kwon)
2010년 2월 육군사관학교 수학(운영분석) 학사 졸업
2015년 8월 한국과학기술원 전산학부 석사 졸업
2020년 2월 한국과학기술원 전산학부 박사 졸업
email : hkwon.cs@gmail.com

박 상 준 (Sangjun Park)
2000년 2월 육군사관학교 독일어 학사 졸업
2010년 2월 한국과학기술원 정보통신공학 석사 졸업
2020년 3월~현재 아주대학교 박사과정
email : sigpsj13438@naver.com

윤 준 혁 (Joonhyoek Yoon)
2012년 2월 육군사관학교 전자공학과 학사 졸업
2018년 8월 퍼듀 전자공학과 석사 졸업
2021년 2월~현재 서울대학교 전기정보공학부 박사과정
email : yjh9001@gmail.com

김 용 철 (Yongchul Kim)
1998년 2월 육군사관학교 전자공학 학사 졸업
2001년 11월 University of Surrey 전자공학과 석사 졸업
2012년 1월 North Carolina State University 전자공학과 박사 졸업
email : kyc6454@mnd.go.kr

김 준 섭 (Junseob Kim)
2016년 2월 육군사관학교 전자공학과 학사 졸업
2020년 8월 텍사스 A&M 대학교 전자공학과 석사 졸업
email : bestsoldier64@gmail.com