https://doi.org/10.33778/kcsa.2020.20.2.111

딥뉴럴네트워크 상에 신속한 오인식 샘플 생성 공격★

권 현*, 박 상 준**, 김 용 철***

요 약

딥뉴럴네트워크는 머신러닝 분야 중 이미지 인식, 사물 인식 등에 좋은 성능을 보여주고 있다. 그러나 딥뉴럴네트워크는 적대적 샘플(Adversarial example)에 취약점이 있다. 적대적 샘플은 원본 샘플에 최소한의 noise를 넣어서 딥뉴럴네트워크가 잘못 인식하게 하는 샘플이다. 그러나 이러한 적대적 샘플은 원본 샘플간의 최소한의 noise을 주면서 동시에 딥뉴럴네트워크가 잘못 인식하도록 하는 샘플을 생성하는 데 시간이 많이 걸린다는 단점이 있다. 따라서 어떠한 경우에 최소한의 noise가 아니더라도 신속하게 딥뉴럴네트워크가 잘못 인식하도록 하는 공격이 필요할 수 있다. 이 논문에서, 우리는 신속하게 딥뉴럴네트워크를 공격하는 것에 우선순위를 둔 신속한 오인식 샘플 생성 공격을 제안하고자 한다. 이 제안방법은 원본 샘플에 대한 왜곡을 고려하지 않고 딥뉴럴네트워크의 오인식에 중점을 둔 noise를 추가하는 방식이다. 따라서 이 방법은 기존방법과 달리 별도의 원본 샘플에 대한 왜곡을 고려하지 않기 때문에 기존방법보다 생성속도가 빠른 장점이 있다. 실험데이터로는 MNIST와 CIFAR10를 사용하였으며 머신러닝 라이브러리로 Tensorflow를 사용하였다. 실험결과에서, 제안한 오인식 샘플은 기존방법에 비해서 MNIST와 CIFAR10에서 각각 50%, 80% 감소된 반복횟수이면서 100% 공격률을 가진다.

Rapid Misclassification Sample Generation Attack on Deep Neural Network

Hyun Kwon*, Sangjun Park**, Yongchul Kim***

ABSTRACT

Deep neural networks (DNNs) provide good performance for machine learning tasks such as image recognition and object recognition. However, DNNs are vulnerable to an adversarial example. An adversarial example is an attack sample that causes the neural network to recognize it incorrectly by adding minimal noise to the original sample. However, the disadvantage is that it takes a long time to generate such an adversarial example. Therefore, in some cases, an attack may be necessary that quickly causes the neural network to recognize it incorrectly. In this paper, we propose a fast misclassification sample that can rapidly attack neural networks. The proposed method does not consider the distortion of the original sample when adding noise. We used MNIST and CIFAR10 as experimental data and Tensorflow as a machine learning library. Experimental results show that the fast misclassification sample generated by the proposed method can be generated with 50% and 80% reduced number of iterations for MNIST and CIFAR10, respectively, compared to the conventional Carlini method, and has 100% attack rate.

Key words : Adversarial example, Machine learning, Evasion Attack, Rapid misclassification sample

접수일(2020년 03월 24일), 수정일(2020년 06월 12일), 게재확정일(2020년 06월 22일)

★ 본 논문은 화랑대연구소의 2020년도(20-군학-18) 저술활동비 지원을 받아 연구되었음.

* 육군사관학교 전자공학과 조교수(주저자)

** 육군사관학교 전자공학과 조교수(공동저자)

*** 육군사관학교 전자공학과 교수(교신저자)

1. 서 론

최근 딥뉴럴네트워크 모델[1]은 이미지 분류[2], 얼굴 인식 분야[3], 음성 인식 분야[4] 등 다양한 영역에서 좋은 성능을 보여주고 있다. 그러나 딥뉴럴네트워크에 대한 취약점이 있다. 이러한 딥뉴럴네트워크의 취약점을 이용한 대표적인 공격으로써, 적대적 샘플(Adversarial example)[5]이 있다. 적대적 샘플은 원본 샘플에 약간의 노이즈를 추가함으로써 사람은 보기에는 식별하기 어려우나 딥뉴럴네트워크는 잘못 인식하게 하는 공격 샘플을 의미한다.

적대적 샘플을 이용한 회피공격은 공격자에 의한 테스트 데이터 조작을 의미한다. 이러한 적대적 샘플에 의한 공격은 딥뉴럴네트워크를 적용한 다양한 분야에 심각한 위험이 될 수 있다. 예를 들어, 자율주행차량에서 딥뉴럴네트워크를 이용하여 사물이나 도로표지판 등을 인식한다. 만약 어떤 악의적인 공격자가 도로표지판에 적대적 샘플 공격을 적용하면, 좌회전인 도로표지판이 사람이 보기에는 좌회전이지만 딥뉴럴네트워크가 장착된 자율주행차량이 인식하기에 우회전으로 잘못 인식하게 만들 수 있다. 또한, 딥뉴럴네트워크는 환자 CT 촬영 결과에 대하여 진단하고 종류를 식별하는 데 이용되기도 한다. 그러나 악의적인 공격자가 환자의 데이터에 조작을 하는 경우, 사람은 식별하기 어렵지만 딥뉴럴네트워크 모델이 환자에 대해서 잘못된 처방을 내려 환자의 건강에 위협이 될 수도 있다. 따라서 인공지능학회나 보안학회에서 이러한 적대적 샘플에 대한 다양한 공격방법 및 방어방법에 대해서 연구하고 있다.

적대적 샘플을 이용한 공격방법은 원본 샘플간의 최소한의 왜곡과 딥뉴럴네트워크가 오인식하는 2가지 조건을 만족해야 한다. 따라서 생성과정에서 2가지 조건을 만족해야 하기 때문에 적대적 샘플을 생성할 때 상당한 시간과 반복이 요구된다. 그러나 경우에 따라서 다소 왜곡이 발생하더라도, 원본 샘플의 최소한의 왜곡에 대한 고려없이 신속하게 딥뉴럴네트워크가 오인식하는 것이 필요할 수 있다. 예를 들어, 도로표지판에 공격자가 신속하게 공격할 필요가 있을 경우, 다소 사람이 보기에도 식별되는 noise가 들어간 이미지이지만 자율주행차량이 잘못인식해서 오작동을 일으키는데 적용할 수도 있다. 또는 음성 데이터의 경우, 공격자가 신속하게 공격해야하는 경우 사람이 듣기에도 잡음 등의 noise가 있지만 신속하게 적대적 샘플을 생성하여 딥뉴럴네트워크가 오작동이 되도록 만들 필요가 있을 수 있다. 특히, 군사적인 상황에서 적군과 아군이 혼재되어 있을 때, 적군의 오판을 유도하기 위해서 신속하게 오인식을 일으키는 샘플을 생성할 필요성이 있다.

이 논문에서는 딥뉴럴네트워크가 오작동하는 것에 우선순위를 둔 신속한 오인식 샘플 생성 공격(fast misclassification sample)을 제안한다. 제안방법은 원본 샘플에 대한 왜곡에 대한 고려없이 신속하게 딥뉴럴네트워크가 오인식이 되는 최적화된 noise를 생성하여 오인식 샘플을 생성한다. 따라서 이 논문에서 제안하는 방법은 기존방법과 달리 별도의 원본 샘플에 대한 왜곡을 고려하지 않기 때문에 기존방법보다 생성속도가 빠른 장점이 있다. 이에 대해서 본 논문에서는 제안방법에 대한 시스템적인 구성과 최적화된 noise을 생성하여 제안샘플의 생성과정 원리에 대해서 설명을 하였다. 다음으로 제안방법에 대하여 성능비교를 위하여 기존 최신방법인 Carlini method[6]와 반복횟수와 공격성공률에 대하여 비교하였다. 또한, 제안한 샘플 공격에 대한 이미지에 대해서 목표공격과 비목표공격에 대해서도 분석을 하였다. 마지막으로 제안방법의 성능을 보이기 위해서 MNIST[7]와 CIFAR10[8] 데이터셋을 이용하여 제안방법의 성능을 입증하였다.

이 장의 나머지 구성은 다음과 같다. 2장에서는 관련연구에 대한 소개를 하고 3장에서는 문제정의와 제안방법에 대한 구조와 생성과정에 대한 설명을 한다. 4장에서는 제안방법에 대한 실험, 평가, 논의를 한다. 마지막으로 5장은 결론으로 구성되

어 있다.

2. 관련연구

딥뉴럴네트워크에 대한 보안 취약점에 대해서 Barreno et al. 연구진[9]에 의해서 처음으로 제시하였다. 특히, 제시한 취약점 중 exploratory attack인 적대적 샘플에 대하여 언급하였다. 적대적 샘플은 원본 샘플에 최소한의 왜곡을 추가하여 오인식을 일으키는 방법이다. 이 방법은 딥뉴럴네트워크의 취약점을 이용한다. 딥뉴럴네트워크의 기본적인 개념측면에서, 딥뉴럴네트워크[1]는 소프트맥스(Softmax)층에서 분류된 값과 실제 라벨값이 일치가 되도록 많은 학습데이터를 학습함으로써 가장 최적의 파라미터를 설정하는 학습과정(training process)을 갖는다. 하지만 이러한 딥뉴럴네트워크는 학습데이터에 최적화가 되었기 때문에 학습데이터의 분포와 차이가 나는 데이터에 대해서 잘못 인식되는 점이 있다. 왜냐하면 딥뉴럴네트워크[1]는 새로운 테스트 데이터가 들어올 경우, 소프트맥스(Softmax)층에서 가장 높은 신뢰값(Confidence value)를 가진 클래스로 인식을 하게 된다. 그러나 공격자가 테스트 데이터(Test data)에 대해서 최적화된 왜곡을 주게 되면 사람은 그 차이를 인지하지 못하지만 딥뉴럴네트워크에서 공격자가 의도한 잘못된 클래스가 가장 높은 신뢰값을 얻도록 오인식 공격을 할 수 있기 때문이다.

이 장에서는 이러한 적대적 샘플을 이용한 관련연구에 대해서 소개하고자 한다. 이 장의 구성은 다음과 같다. Ⅱ.1에서부터 Ⅱ.5까지 적대적 샘플에 대한 목표모델에 대한 정보의 양, 공격목표, 왜곡 측정, 생성방법, 응용방법 순으로 구성되어 있다.

2.1 목표모델 정보량에 따른 분류

적대적 샘플이 목표 모델에 대한 정보양에 따라서 화이트박스 공격[6,10]과 블랙박스 공격으로 구분될 수 있다. 화이트박스 공격은 목표모델에 대한 모든 정보에 대해서 공격자가 알고 있는 상황을 가정한다. 이 경우에 모델의 파라미터, 모델의 구조, 소프트맥스의 수치 등에 대한 모든 정보를 알고 있다는 가정 하에 공격을 의미한다. 반면에 블랙박스 공격[11, 12]은 목표모델에 대한 정보가 없이 입력값에 대한 결과값만 아는 상황에서의 공격을 의미한다. 이 연구에서는 화이트박스를 가정하여 목표 모델에 대한 소프트맥스의 수치를 안다는 가정 하에 제안샘플을 생성한다.

2.2 공격목표에 따른 분류

적대적 샘플[6]의 공격목표에 따라서 목표-적대적 샘플(Targeted adversarial example)과 비목표-적대적 샘플(Untargeted adversarial example)로 구분된다. 목표-적대적 샘플의 경우, 적대적 샘플에 대해서 공격자가 선택한 특정 클래스로 잘못 인식하게 하는 공격을 의미한다. 반면에 비목표-적대적 샘플의 경우, 원본 클래스가 아닌 임의의 잘못된 클래스로 잘못 공격하는 방법을 의미한다. 따라서 상대적으로 비목표-적대적 샘플이 생성하기가 쉽고 빠른 장점을 가진다. 이 연구에서는 목표-적대적 샘플과 비목표-적대적 샘플에 대한 2가지 모두에 대하여 공격성공률과 반복횟수를 기존방법과 비교분석을 하였다.

2.3 왜곡지수에 따른 분류

왜곡 지수를 설정하는 방법[6]은 L_p와 같은 식으로 많이 사용된다.

$$L_p = \sqrt[p]{(x - x^*)^p},$$

여기서 L_p는 왜곡측정방법을 의미하고, x는 원본 샘플, x^*는 적대적 샘플을 의미한다. L_p의 값이 적어질수록 원본샘플간의 유사성이 증가되게 된다. 이 연구에서 사용한 왜곡측정방법으로 L_2을 적용하였으며, 이는 원본샘플간의 픽셀값의 차이의 제곱의 합의 루트값을 의미한다. 이 연구에서는 p값이 2인 L_2 방식을 이용하여 원본샘플과 제안샘플의 왜곡을 측정하였다.

2.4 생성방법에 따른 분류

적대적 샘플을 생성하는 대표적인 방법으로는

먼저 Fast gradient sign method (FGSM)[13]가 있다. 이 방법은 딥뉴럴네트워크에서 미분한 후에 gradient(기울기) 값이 0이 되는 지점을 찾아 최적의 공격할 수 있는 적대적 샘플을 생성한다. 이 방법은 one-step 방법이기 때문에 생성속도가 빠르다는 장점이 있지만 모델에 대한 피드백이 적기 때문에 공격성공률이 낮고 공격자가 선택한 클래스로 잘못 인식하게 하는 목표-적대적 샘플 생성이 제한적인 단점을 갖는다. 두 번째로 Jacobian-based Saliency map Attack (JSMA) 방법[14]으로 목표-적대적 샘플을 생성하며 여러 번의 반복과정을 통하여 생성한다. 이 방법은 원본 샘플간의 왜곡을 최소화하기 위해서 적대적 샘플의 saliency value를 줄일 수 있는 요소를 찾는 방법을 적용한다. 하지만 이 방법은 생성하는 데 시간이 많이 걸리는 단점이 있다. 세 번째로 DeepFool 방법[10]은 FGSM에 비해 개선된 방법으로 선형근사방법을 접근하여 딥뉴럴네트워크상에 적대적 샘플을 생성한다. 그러나 이 방법도 많은 반복횟수가 요구된다는 단점이 있다. 마지막으로 Carlini method[6]는 최신 방어방법에 대해서 100% 공격성공률을 가지는 방법으로 FGSM 방법보다 우수하다. 특히 distillation 방어시스템에 대해서도 100% 공격이 가능한 점이 있다. 이 연구에서는 Carlini method를 응용 및 개선하여 적용하였다. 또한 기존방법인 Carlini 방법과 비교 실험하였다. 이에 대한 자세한 내용은 4장과 5장에서 설명하였다.

2.5 응용방법에 따른 분류

다중 딥뉴럴네트워크 환경 하에서 적대적 샘플 공격을 응용한 다양한 연구가 진행되고 있다. 예를 들어, 전시 상황에서는 적군과 아군이 혼재되어 있는 상황에 있다. 이러한 상황에서 아군은 보호하면서 적군만 오인식하는 Friend-safe adversarial example (아군 친화적 적대적 샘플)[15,16]에 대한 연구가 소개되었다. 또한, 아군 친화적 적대적 샘플 방법을 확장하여 한 개의 적대적 샘플을 토대로 여러 개의 모델을 각각 다른 클래스로 오인식하게 하는 다중 목표-적대적 샘플 공격방법[17]도 소개되었다. 예를 들어, 다중 목표-적대적 샘플의 경우 A 모델은 왼쪽으로 오인식하게 하고 B 모델은 오른쪽으로 오인식하게 하고 C 모델은 유턴으로 오인식하게 하는 등 공격자가 원하는 각각의 모델을 원하는 클래스로 오인식하게 하는 방법이다.

공격 목표 측면에서, 비목표-적대적 샘플의 패턴 취약점을 제거한 무작위 비목표-적대적 샘플 공격 방법[18]도 있다. 이 방법은 원본 클래스와 유사한 클래스로 오인식하는 패턴적인 취약점을 제거하고 임의의 클래스로 잘못 인식하게 만드는 방법이다. 또한, 목표 공격과 비목표 공격과 다르게, 선택한 특정영역을 제외한 나머지로 오인식하는 선택적 비목표공격 방법[19]에 대한 연구도 진행되었다. 이 방법은 공격자가 지정한 특정영역을 제외한 범위 내에서 모델이 오인식하게 하는 공격방법이다.

변조하는 영역 측면에서, 기존방법과 다르게 특정영역만 변조하여 오인식하는 방법들이 소개되었다. 먼저 한 개의 픽셀만 변경하여 모델이 오인식하는 방법[20]은 상대적으로 공격 성공률은 낮지만 한 개 픽셀만 변경하여 공격하는 장점이 있다. 또한 전체 영역 중에 공격자가 지정한 영역만 변조하여 오인식하는 제한적 적대적 샘플 공격방법[21]도 제안되었다.

3. 제안방법

3.1 문제정의

적대적 샘플을 생성할 때, 일반적으로 최적화문제(Optimal problem)로써 2가지 조건인 원본 샘플의 왜곡 최소화와 딥뉴럴네트워크의 오인식을 만족하는 목표함수 F를 설정한다.

$$F = loss_D(x, x^*) + c \times loss_A(x^*),$$

여기서, $loss_D(x, x^*)$는 원본 샘플 x와 적대적 샘

플 x^*와의 거리를 나타내는 왜곡 loss function을 의미한다. $loss_A(x^*)$는 딥뉴럴네트워크가 x^*에 대해서 잘못 인식하는 공격 loss function을 의미한다. c는 가중치 값으로 왜곡 loss function과 공격 loss function에 대한 가중치를 조절해주는 역할을 한다. 따라서 기본적인 방법은 목표함수 F를 최소화함으로써 공격성공률을 높이면서 동시에 최소한의 왜곡을 만족하는 적대적 샘플을 찾는 것을 목표로 한다. 그러나 이 방법은 두가지 조건을 동시에 만족하도록 적대적 샘플을 생성해야하기 때문에 상대적으로 많은 계산과 시간이 소요된다.

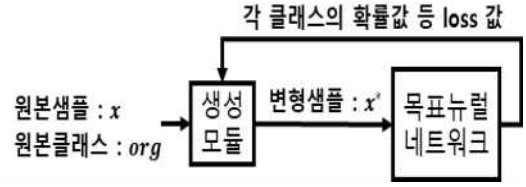
여기서 제안한 방법은 신속하게 딥뉴럴네트워크가 오인식을 일으키는 샘플을 생성하기 위해 왜곡 loss function을 삭제하여 조건식을 1개로 줄였다. 제안한 목표함수 F_p는 다음과 같다.

$$F_p = c \times loss_A(x^*)$$

이렇게 식을 변경함으로써 제한조건이 1개로 줄어들었기 때문에, 적은 반복횟수로 딥뉴럴네트워크가 오인식하는 샘플을 생성할 수 있다. 구체적으로, c값을 binary search로 증가시키면서 공격성공률이 100%가 되는 c값을 찾아 제안샘플을 생성한다. 이 방법을 사용하면 기존 loss function에서 왜곡에 대한 부분이 없기 때문에 상대적으로 빠르게 딥뉴럴네트워크가 오인식을 일으킬 수 있는 샘플을 생성할 수 있다.

3.2 제안방법 구성

제안방법에 대한 시스템 구성은 그림 1과 같이 목표딥뉴럴네트워크(목표모델)과 생성모듈로 구성되어 있다.



<그림 1> 제안방법의 구성

제안방법이 오인식 샘플을 생성하기 위해서, 생성모듈은 원본샘플 x와 noise w을 더한 변형샘플 x^*을 다음과 같이 생성한다.

$$x^* = x + w$$

여기서, w은 최적화된 noise를 의미한다. 이렇게 생성된 변형샘플 x^*을 목표모델에 입력값으로 제공하고 이에 대한 피드백으로 loss 값인 각 클래스에 확률값 등을 제공받는다. 제공받은 피드백을 이용하여 생성모듈은 다음과 같은 손실목표함수 F_p을 계산한다.

$$F_p = c \times loss_A(x^*),$$

여기서 c는 가중치로써 binary 식으로 증가하게 되고, 초기값은 1이다. $loss_A(x^*)$는 공격 loss function 값을 의미하고 목표-적대적 샘플과 비목표-적대적 샘플로 구분된다. 먼저 목표-적대적 샘플의 경우 $loss_A(x^*)$는 아래와 같이 표현된다.

$$loss_A(x^*) = \max\{Z(x^*)_i : i \neq t\} - Z(x^*)_t,$$

여기서 t는 공격자가 선택한 목표 클래스를 의미하고, $Z(\bullet)$[6, 14]는 목표모델로부터 각 클래스의 확률값을 의미한다. $loss_A(x^*)$를 최소화할수록 목표모델이 x^*를 공격자가 선택한 목표 클래스가 가장 큰 확률값이 되도록 인식한다.

반면에 비목표-적대적 샘플의 경우, $loss_A(x^*)$는 아래와 같이 표현된다.

$$loss_A(x^*) = Z(x^*)_{org} - \max\{Z(x^*)_i : i \neq org\}$$

여기서 org는 원본 클래스를 의미한다. $loss_A(x^*)$를 최소화할수록 목표모델이 원본 클래스가 아닌 임의의 클래스가 가장 큰 확률값이 되도록 인식한다. 이 과정을 주어진 횟수만큼 손실목표함수 F_p를 최소화하는 과정을 반복하면서 공격성공률이 100%인 변형샘플인 fast misclassification sample을 생성한다.

<표 1> 딥뉴럴네트워크의 구조

각 층의 제원	MNIST	CIFAR10
Convolutional + ReLU	3×3×32	3×3×64
Convolutional + ReLU	3×3×32	3×3×64
Max pooling	2×2	2×2
Convolutional + ReLU	3×3×64	3×3×128
Convolutional + ReLU	3×3×64	3×3×128
Max pooling	2×2	2×2
Fully connected + ReLU	200	256
Fully connected + ReLU	200	256
Softmax	10	10

4. 실험 및 평가

제안한 샘플을 생성하기 위한 실험환경으로 Tensorflow 머신러닝 라이브러리[22]를 사용하였으며, 서버는 Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz와 GPU는 GeForce GTX 1050을 사용하였다.

4.1 데이터셋

데이터셋은 MNIST[7]와 CIFAR10[8]을 사용하였다. MNIST은 대표적인 손글씨 이미지셋으로 0부터 9까지의 이미지로 구성되어 있다. MNIST는 6만개의 훈련데이터와 1만개의 테스트 데이터를 가진다. 반면에 CIFAR10은 컬러이미지로 비행기, 차, 새, 고양이, 사슴, 개, 개구리, 말, 보트, 트럭 순으로 10가지 사물이미지로 구성되어 있으며, 5만개의 훈련데이터와 1만개의 테스트 데이터를 가진다.

<표 2> 딥뉴럴네트워크 파라미터

제 원	MNIST	CIFAR10
Learning rate	0.1	0.1
Momentum	0.9	0.9
Delay rate	-	10 (decay 0.0001)
Dropout	0.5	0.5
Batch size	128	128
Epochs	50	200

4.2 딥뉴럴네트워크

공격의 대상이 되는 딥뉴럴네트워크는 MNIST와 CIFAR10의 경우에 합성곱 딥뉴럴네트워크(Convolutional neural network)[23]로 구성하였다. 이에 대한 아키텍처와 파라미터는 표1과 표2와 같이 구성되어 있다. MNIST의 경우, 학습데이터 6만개를 이용하여 목표 딥뉴럴네트워크를 학습하였고 테스트데이터 1만개에 대한 정확도는 99.2%를 가진다. 반면에 CIFAR10의 경우, 학습데이터 5만개를 이용하여 목표 딥뉴럴네트워크를 학습하였고, 테스트 1만개에 대한 정확도는 80.14%를 가진다

4.3 제안한 오인식 샘플 생성

생성모듈의 경우에 제안한 오인식 샘플을 생성하기 위해서, 최적화 알고리즘인 Adam 알고리즘을 사용하였다. MNIST의 경우, 학습률은 0.1이고 초기 상수값은 0.1로 설정하였다. CIFAR10의 경우, 학습률은 0.01이고 초기 상수값은 0.1로 설정하였다. 테스트 데이터를 이용하여 임의의 1000개의 제안한 오인식 샘플을 생성하여 공격성공률과 반복횟수 등에 대하여 분석을 하였다.

4.4 실험결과

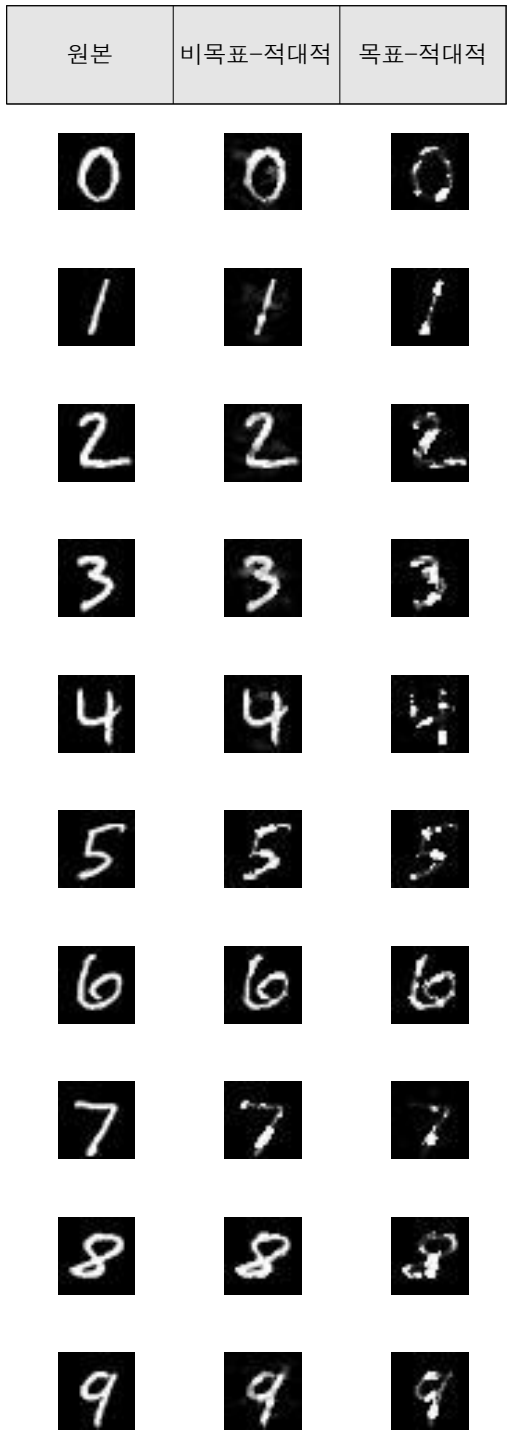
그림2와 그림3은 제안방법으로 비목표-적대적

샘플과 목표-적대적 샘플을 MNIST와 CIFAR10에 대하여 생성한 이미지 예시를 보여준다. 그림 2과 그림 3에서 목표-적대적 샘플의 경우에 목표 클래스를 +1을 한 값으로 하였다. 예를 들어, MNIST의 경우, 원본 샘플이 “0”인 경우에 “1”로 오인식하게 하였으며, 원본 샘플이 “1”인 경우에는 “2”로 오인식하게 만들었다. 마찬가지로, CIFAR10의 경우, 원본샘플이 “비행기(0)”인 경우에 “자동차(1)”로 오인식하게 하였으며, “자동차(1)”의 경우에는 “새(2)”로 오인식하게 만들었다. 그림 2에서 목표-적대적 샘플의 경우, 상대적으로 원본 샘플간의 최소화에 대한 조건이 없이 공격자가 원하는 클래스로 잘못 인식하게 해야하기 때문에 이미지 왜곡이 많이 되는 것을 볼 수 있다. 그렇지만 비목표-적대적 샘플의 경우에는 원본 클래스가 아닌 임의의 클래스로 오인식하는 것이기에 목표-적대적 샘플보다 원본 샘플간의 왜곡이 적은 것을 볼 수 있다. 한편, CIFAR10에 해당되는 그림 3을 보면 원본샘플과 제안방법의 이미지 차이가 적은 것을 볼 수 있다. MNIST와 달리 컬러이미지 이기 때문에 그 노이즈가 상대적으로 식별하기 어렵고 공격할 수 있는 픽셀의 경우수도 많기 때문에 CIFAR10에서는 사람의 인식률마저도 좋은 성능을 가지는 것을 볼 수 있다.

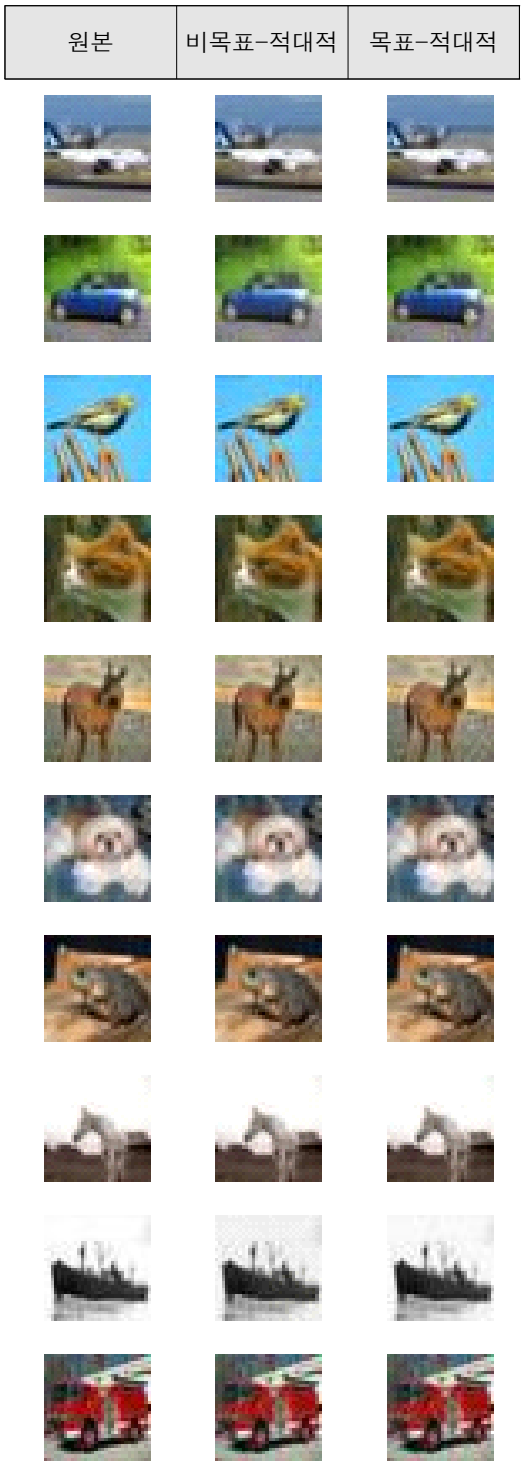
표 3은 임의의 1000개 샘플에 대하여 제안방법과 기존방법에서 필요한 반복횟수, 공격성공률, 왜곡평균을 보여준다. 표에서 보면 제안방법은 기존 Carlini 방법과 비교하여 100% 성공률을 유지하면서 동시에 상대적으로 적은 반복횟수로 가능한 것을 볼 수 있었다.

<표 3> 제안방법의 반복횟수, 성공률, 왜곡평균

제 원		MNIST		CIFAR10	
		비목표	목표	비목표	목표
반복 횟수	기존	400	500	6000	10000
	제안	200	250	700	2000
성공률	기존	100%	100%	100%	100%
	제안	100%	100%	100%	100%
왜곡 평균	기존	1.36	2.12	27.37	46.93
	제안	3.1	3.9	71.4	189.2



<그림2> MNIST에서 생성한 제안샘플.



<그림3> CIFAR10에서 생성한 제안샘플.

이는 기존방법보다 이미지 왜곡에 대한 가중치를 없앰으로써 상대적으로 적은 반복횟수로 공격이 가능하게 하였다. 하지만 왜곡측면에서 상대적으로 기존 Carlini 방법보다 다소 높은 점이 있지만 CIFAR10과 같이 컬러이미지의 경우, 사람의 눈으로 그 noise를 식별하기가 쉽지 않기 때문에, 컬러이미지의 경우 제안방법이 유용한 측면이 있다.

4.5 분석 및 논의

딥뉴럴네트워크 공격목표 측면에서, 제안방법에서 목표-적대적 샘플과 비목표-적대적 샘플에 대해서 각각 성능을 분석하였다. 목표-적대적 샘플은 공격자가 정한 class로 오인식 하는 방법으로 좀 더 많은 왜곡이 필요하다. 반면에 비목표-적대적 샘플은 임의의 잘못된 class로 오인식 하는 방법이기 때문에 적은 왜곡으로도 생성이 가능하다. 제안방법은 목표-적대적 샘플과 비목표-적대적 샘플 두가지 경우에 기존방법에 비해서 적은 반복횟수로 100% 공격성공률을 가지는 것을 볼 수 있었다.

왜곡 측면에서, 제안방법은 기존방법에 비해서 다소 왜곡이 증가하는 것을 볼 수 있었다. 왜냐하면 원본 샘플의 왜곡을 고려하지 않고 오인식에 중점을 두었기 때문이다. 하지만 CIFAR10과 같이 컬러이미지의 경우에는 사람의 눈으로 그 noise를 식별하기 어렵기 때문에 좋은 성능을 가지는 것을 볼 수 있었다.

생성측면에서, 제안방법은 실시간적으로 적대적 샘플 공격이 필요할 때 유용할 수 있다. 예를 들어, 군사적인 상황에서 적군 딥뉴럴네트워크가 오인식하도록 신속하게 오인식 샘플 생성이 필요한 경우, 제안방법을 이용하게 되면 기존방법에 비해서 50% 이상 적은 반복횟수로도 생성 및 공격이 가능한 장점이 있다.

적대적 샘플에 대한 방어방법으로 딥뉴럴네트워크를 강건하게 만드는 방법[24][25]과 데이터를 조작하는 방법[26][27]으로 구분할 수 있다. 딥뉴럴네트워크를 강건하게 만드는 방법은 적대적 샘플 학습하는 방법[24]이나 distillation method[25]로 적대적 샘플을 생성하는 것을 어렵게 구성하는 방법 등이 있다. 반

면에 데이터를 조작하는 것은 적대적 샘플에 있는 noise를 제거하는 filtering 방법[26][27] 등을 이용하여 적대적 샘플의 공격효과를 줄이는 방법들이 있다. 이러한 기존 방어방법을 응용하여 제안방법의 방어연구도 흥미로운 주제가 될 것이다.

5. 결 론

이 논문에서는 딥뉴럴네트워크를 오인식하는데 보다 적은 반복횟수로 공격하는 방법을 제안하였다. 제안한 방법은 원본 샘플간의 왜곡성을 가중치를 제거하고 딥뉴럴네트워크를 오인식하는 fast misclassification sample을 생성한다. 제안방법은 기존방법과 달리 별도의 원본 샘플에 대한 왜곡을 고려하지 않기 때문에 기존방법보다 생성속도가 빠른 장점이 있다. 실험결과에서, 제안방법은 기존 Carlini 방법보다 MNIST의 경우는 목표-적대적 샘플과 비목표-적대적 샘플에 대해서 각각 50%, 50% 감소된 반복횟수로 생성이 가능하고, CIFAR10의 경우에 목표-적대적 샘플과 비목표-적대적 샘플에 대해서 각각 80%, 88% 감소된 반복횟수로 신속히 생성이 되며, 공격성공률은 모두 100%가 되는 것을 볼 수 있었다. 특히, 컬러이미지의 경우에 사람의 인식률도 거의 유지되는 것을 볼 수 있었다. 이처럼 제안방법은 실시간적으로 신속하게 적대적 샘플을 생성할 수 있는 장점이 있다. 특히, 실시간 딥뉴럴네트워크가 이용되는 자율주행차량 등에 대하여, 적은 반복횟수로 신속하게 적대적 샘플을 생성하기 때문에 실시간적으로 딥뉴럴네트워크가 장착된 자율주행차량을 공격할 때 장점이 있다. 또한, 향후 연구로 제안방법은 이미지 뿐만 아니라 음성, 비디오 등의 데이터 연구로 확장이 가능하며, 제안방법에 대하여 딥뉴럴네트워크를 강건하게 만들거나 데이터를 조작하여 방어하는 방법에 대해서도 흥미로운 주제가 될 것이다.

참고문헌

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

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, May 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[3] Sun, Xudong, Pengcheng Wu, and Steven CH Hoi. "Face detection using deep learning: An improved faster RCNN approach." Neurocomputing 299 (2018): 42-50.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. M. N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Process. Mag., vol. 29, no. 6, pp. 82 - 97, Nov. 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] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in Proc. IEEE Symp. Secur. Privacy (SP), May 2017, pp. 39 - 57.

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

[8] A. Krizhevsky, V. Nair, and G. Hinton. (2014). The Cifar-10 Dataset. http://www.cs.toronto.ed/kriz/cifar.html

[9] Barreno M, Nelson B, Joseph AD, Tygar J. The security of machine learning. Mach Learn 2010; 81(2):121 - 48.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern

Recognit., Jun. 2016, pp. 2574 - 2582.

[11] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into transferable adversarial examples and black-box attacks,” in Proc. 5th Int. Conf. Learn. Represent. (ICLR), Toulon, France, Apr. 2017.

[12] Kwon, Hyun, et al. "Advanced ensemble adversarial example on unknown deep neural network classifiers." IEICE TRANSACTIONS on Information and Systems 101.10 (2018): 2485-2500.

[13] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in Proc. 5th Int. Conf. Learn. Represent. (ICLR), Toulon, France, Apr. 2017.

[14] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P), Mar. 2016, pp. 372 - 387.

[15] Kwon, Hyun, et al. "Friend-safe evasion attack: An adversarial example that is correctly recognized by a friendly classifier." Computers & Security 78 (2018): 380-397.

[16] Kwon, Hyun, et al, “Selective Audio Adversarial Example in Evasion Attack on Speech Recognition System ”, IEEE Transactions on Information Forensics & Security, 2019. DOI:10.1109/TIFS.2019.2925452

[17] Kwon, Hyun, et al. "Multi-targeted adversarial example in evasion attack on deep neural network." IEEE Access 6 (2018): 46084-46096.

[18] Kwon, Hyun, et al. "Random untargeted adversarial example on deep neural network." Symmetry 10.12 (2018): 738.

[19] Kwon, Hyun, et al. "Selective Untargeted Evasion Attack: An Adversarial Example That Will Not Be Classified as Certain Avoided Classes." IEEE Access 7 (2019): 73493-73503.

[20] Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." IEEE Transactions on Evolutionary Computation (2019).

[21] Kwon, Hyun, Hyunsoo Yoon, and Daeseon Choi. "Restricted Evasion Attack: Generation of Restricted-Area Adversarial Example." IEEE Access 7 (2019): 60908-60919.

[22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, “TensorFlow: A system for largescale machine learning,” in Proc. OSDI, vol. 16, 2016, pp. 265 - 283

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278 - 2324, Nov. 1998.

[24] I. Goodfellow, J. Shlens, and C. Szegedy, “Expl aining and harnessing adver sarial examples,” in International Conference on Learning Repres entations, 2015.

[25] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to advers arial perturbations against deep neural network s,” in Security and Privacy (SP), 2016 IEEE S ymposium on, pp. 582 - 597, IEEE, 2016.

[26] A. Fawzi, O. Fawzi, and P. Frossard, “Analysi s of classifiers’ robustness to aversarial pertur bations,” Machine Learning, pp. 1 - 28, 2015.

[27] Jin, Guoqing, et al. "APE-GAN: Adversarial pe rturbation elimination with GAN." ICASSP 20 19-2019 IEEE International Conference on Aco ustics, Speech and Signal Processing (ICASS P). IEEE, 2019.

〔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@kma.ac.kr

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