딥러닝 모델에 대한 적대적 예제 공격 기술 동향 연구
– 공격 기법의 노이즈 삽입 패턴 비교분석을 중심으로 –

김민지, 이륜건, 조영호(교신저자)
국방대학교 관리대학원 국방과학학과 컴퓨터공학/사이버전 협동전공
e-mail: minji11294@gmail.com, lrg1393@gmail.com, youngho@kndu.ac.kr

A Study of Research Trends in Adversarial Example Attacks on Deep Learning Models
- Focusing on Comparing Perturbation Insertion Patterns -

Minji Kim, Ryungeon Lee, Youngho Cho
Dept. of Defense Science (Computer Engineering and Cyberwarfare Major),
Korea National Defense University

요 약

딥러닝은 데이터 분석, 이미지 및 음성 인식, 자연어 처리 등 다양한 분야에서 높은 성능을 보이고 있으나, 적대적 예제 공격에 대한 취약성이 지적되고 있다. 적대적 예제 공격이란 입력 이미지에 미세한 노이즈를 삽입하여 이미지 분류 모델이 오분류를 일으키도록 하는 기술을 말하며, 딥러닝 모델의 신뢰성과 안정성에 심각한 영향을 미칠 수 있다. 따라서 본 연구에서는 적대적 예제 공격의 정의와 분류, 대표적인 공격 기법 및 최신 기술 동향을 살펴본다. 특히, 공격 기법의 노이즈 삽입 패턴은 적대적 예제를 생성할 때 활용한 거리 지표에 따라 달라진다. 따라서 이를 중점적으로 비교분석하여, 각 공격 기법이 생성한 적대적 예제 간의 차이점을 확인한다.

1. 서론

딥러닝(Deep Learning)은 인공지능 분야에서 수년간 해결하지 못했던 한계들을 극복하며 급속한 발전을 이루고 있다. 데이터 분석은 물론 이미지 및 음성 인식, 자연어 처리 등 광범위한 영역에 걸쳐 이미 압도적인 성능을 입증했으며, 이에 따라 학문, 비즈니스, 공공 등 거의 모든 분야에서 딥러닝 기술을 활용하려는 시도가 활발하다[1, 2].

한편, 딥러닝 모델을 대상으로 하는 공격 기술도 함께 발전하고 있다[3, 4, 5]. 특히, 적대적 예제 공격(Adversarial Example Attacks)은 입력 이미지에 미세한 노이즈(Perturbation)를 삽입하여 학습이 끝난 이미지 분류 모델이 오분류를 일으키도록 하는 방법이다[3]. 이러한 공격 방식은 딥러닝 모델뿐만 아니라 인간의 시각체계로도 인지하기 어렵다는 점을 고려하면, 매우 현실적인 보안 위협이라고 할 수 있다.

따라서 본 논문에서는 딥러닝 모델에 대한 대표적인 적대적 예제 공격 기법 및 최신 기술 동향을 살펴보고자 한다. 특히, 공격 기법의 노이즈 삽입 패턴은 적대적 예제를 생성할 때 활용한 거리 지표에 따라 달라진다. 각 공격 기법이 생성한 적대적 예제 간의 차이점을 확인하기 위하여, 이를 중점적으로 비교분석한다.

2. 적대적 예제 공격 소개

2.1 적대적 예제 공격의 정의

딥러닝 모델은 컴퓨터 비전 분야에서의 다양한 작업을 높은 정확도로 수행할 수 있지만, Szegedy 등[2]은 입력값에 미세한 조작을 가하여 모델의 예측 오류를 최대화할 수 있는 취약점이 있다는 것을 처음으로 발견했다. 특히, 적대적 예제(Adversarial Example)란 입력 이미지에 인간의 시각체계로도 인지하기 어려운 미세한 노이즈를 삽입하여 생성한 이미지를 말하는데, 딥러닝 기반 이미지 분류 모델은 이러한 적대적 예제를 정확하게 분류하지 못한다. 따라서 공격자가 적대적 예제 공격을 수행할 경우, 자신의 의도에 따라 딥러닝 분류 모델이 오분류를 일으키도록 유도하는 것이 가능하다.

Szegedy의 연구는 딥러닝 분류 모델을 대상으로 하는 공격에 대한 학계의 광범위한 관심을 촉발하는 계기가 되었다. 이에 따라, FGSM[3]을 시작으로 하여 JSMA[4], DeepFool[5] 등 다양한 공격 기법들이 고안되었다. 최근에는 분류 모델이 오분류를 일으킬 확률을 유지하면서 입력 이미지와 적대적 예제 간 유사성을 더욱 높이기 위한 연구가 활발히 진행 중이다[7, 9].

2.2 적대적 예제 공격의 분류

2.2.1 공격자가 가진 정보의 양에 따른 분류

우선, 블랙박스 공격(Black-box Attacks)은 공격자가 공격 대상 딥러닝 모델에 대한 지식이 없는 상태에서 적대적 예제를 생성해내는 것을 말한다. 즉, 모델의 입력과 출력만을 관찰하고 이를 기반으로 공격을 수행한다. 대부분의 실제 공격 시나리오가 블랙박스 공격에 해당한다는 점을 고려하면, 이러한 공격 기법들은 실제 실행 가능성이 큰 보안 위협이다.

반면에 화이트박스 공격(White-box Attacks)은 공격자가 공격대상 딥러닝 모델의 파라미터, 아키텍처, 학습방법 등을 완전히 이해하고 있다고 가정한다. 이러한 지식을 활용하여, 공격자는 더욱 효과적인 적대적 예제를 생성하는 것이 가능하다. 화이트박스 공격은 방어자 입장에서 모델의 보안 취약점을 파악하고 해결하기 위해 활용될 수 있다.

2.2.2 공격 목표에 따른 분류

표적 공격(Targeted Attacks)은 공격자가 입력 데이터를 조작하여 모델의 출력을 특정한 방향으로 의도적으로 유도하는 것을 말한다. 즉, 이 공격은 공격자가 의도하는 특정한 클래스로 오분류를 유도하기 위해 사용된다. 따라서 공격자는 오분류를 유도할 목표 클래스를 사전에 결정해야 하며, 이를 위해 클래스에 대한 지식이 있어야 한다.

무표적 공격(Non-targeted Attacks)은 입력 데이터를 특정한 클래스가 아니라 임의의 다른 클래스로 오분류하기 위한 공격이다. 따라서, 목표 클래스에 대한 지식이 없어도 수행할 수 있으며, 일반적으로 무표적 공격이 표적 공격보다 쉬운 공격이라 할 수 있다.

2.2.3 거리 지표(Distance Metrics)에 따른 분류

적대적 예제 공격에서는 입력 데이터(예: 이미지)와 적대적 예제 간의 유사성을 정량화하기 위해 거리 지표를 사용한다. 거리 지표는 적대적 예제가 입력 이미지와 일정한 거리 내에서만 변형되도록 하는 제약조건(Constraint)의 역할을 한다. 즉, 거리 지표는 노이즈의 크기, 형태 등을 결정짓는 가장 중요한 요소라고 할 수 있다. 거리 지표는 측정 방식에 따라 ℓ_∞, ℓ_0, ℓ_2 Norm 등이 있으며 이는 ℓ_p Norm으로 일반화된다.

ℓ_∞ Norm 공격은 적대적 예제 생성 시 변경된 픽셀 중 그 변화량의 최댓값만 측정한다. 즉, ℓ_∞ Norm으로 변화량의 크기를 제한하면 다른 픽셀들도 그 제한범위 안에서만 변할 수 있으며, 이에 따라 육안으로 식별하기 어려운 노이즈가 생성된다.

ℓ_0 Norm 공격은 적대적 예제 생성 간 변경된 픽셀의 수를 측정한다. 즉, 변경하는 픽셀의 수를 제한하여 사람의 눈에 잘 띄지 않도록 한다. 이에 따라, ℓ_0 Norm을 사용하는 공격 기법은 모든 픽셀이 아니라 특정 픽셀에만 노이즈를 삽입하는 특성이 있다.

ℓ_2 Norm 공격은 입력 이미지와 적대적 이미지 간의 유클리드(Euclidean) 거리, 즉 최단 거리를 측정한다. 이는 가장 일반적인 거리 계산 방식으로, 픽셀 간 변화량의 제곱합제곱근(Square Root of Sum of Square)으로 계산한다. 그리고 이 값을 작게 만듦으로써 노이즈를 매우 미세한 크기로 제한한다.

3. 적대적 예제 공격 기법 비교

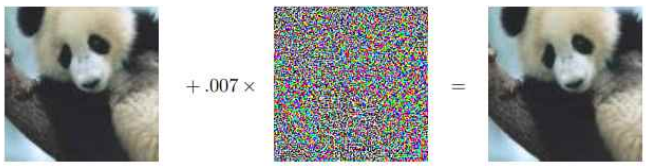
거리 지표는 적대적 예제 생성 간 노이즈의 삽입 패턴에 결정적인 영향을 미친다. 적절한 거리 지표를 선정한다면, 모델이 오분류를 일으킬 확률을 유지하면서도 입력 이미지와 더욱 유사한 적대적 예제를 생성할 수 있다. 따라서, 공격을 분류하는 세 가지 기준 중 거리 지표에 따라 각 공격 기법을 비교해보았다.

대표적인 적대적 예제 공격 기법인 FGSM, JSMA, DeepFool은 순서대로 ℓ_∞, ℓ_0, ℓ_2 Norm을 사용한다. 각 공격 기법의 원리와 알고리즘을 살펴본 다음, 노이즈 삽입 패턴을 이미지로 출력하여 직접 비교분석한다. 이후에는 최신 거리 지표를 사용한 새로운 공격 기법인 JND-based Attack, GMM에 대해 다루고, ℓ_p Norm 공격 기법과의 차이점을 확인한다.

3.1 대표적인 적대적 예제 공격 기법

3.1.1 FGSM(Fast Gradient Sign Method)

Goodfellow 등[3]은 딥러닝 모델이 입력 데이터에 작은 변화만 주어지더라도 결과값이 선형적으로 반응한다는 것을 발견했다. FGSM은 이와 같은 딥러닝 모델의 취약점을 이용한 ℓ_∞ Norm 공격 기법으로, 최대 크기가 ϵ로 제한된 노이즈를 입력 이미지에 추가하여 적대적 예제를 생성한다. FGSM의 원리는 [그림 1]과 같다.

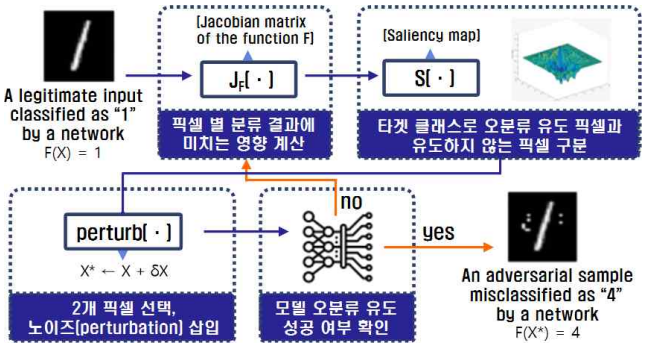


[그림 1] FGSM의 원리[3]

FGSM은 입력 이미지 x, 예측 클래스 y, 모델의 파라미터 θ를 이용하여 모델의 손실함수 $J(\theta,x,y)$를 계산한다. 이후 손실함수를 x로 미분한 기울기인 $\nabla_x J(\theta,x,y)$를 구하고, 기울기의 부호(Sign) 방향으로 ϵ 크기의 작은 노이즈를 추가하여 손실함수를 극대화한다. 이와 같은 방식으로 생성된 적대적 예제 $\tilde{x} = x + \epsilon(\nabla_x J(\theta,x,y))$는 기존의 예측 클래스인 y가 아닌 다른 클래스로 오분류된다. 이와 같은 간단한 방식을 통해, FGSM은 빠르고 효율적으로 대량의 적대적 예제를 쉽게 생성하는 것이 가능하다.

3.1.2 JSMA(Jacobian-based Saliency Map Attack)

JSMA[4]는 입력 이미지의 어떤 입력 요소(Input Feature)를 변화시켜야 딥러닝 모델의 분류 결과가 달라지는지 찾아내는 것을 목표로 한다. JSMA는 대표적인 ℓ_0 Norm 공격 기법으로, 분류 모델의 오분류를 유도하기 위해 이미지 전체가 아니라 최소의 픽셀에만 노이즈를 삽입한다. JSMA의 알고리즘은 [그림 2]과 같다.

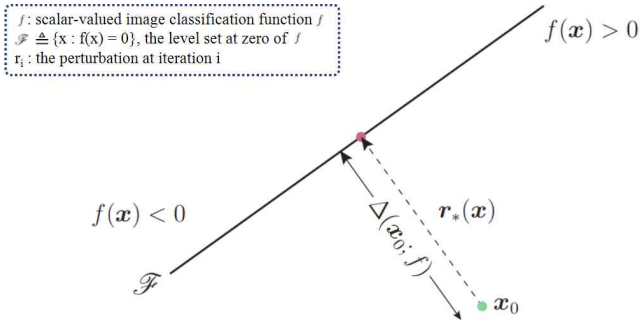


[그림 2] JSMA의 알고리즘

먼저, 입력 이미지에 대한 모델의 예측 결과를 출력한다. 다음으로, 모델의 출력값을 각 입력 픽셀로 미분한 결과를 행렬 형태로 표현한 자코비안 행렬(Jacobian Matrix)을 계산한다. 이를 통해 각 픽셀이 모델의 분류 결과에 미치는 영향을 알 수 있다. 이후 자코비안 행렬은 돌출 맵(Saliency Map)으로 매핑되며, 돌출 맵에서 가장 큰 값을 가진 픽셀을 선택해서 노이즈를 삽입한다. 이러한 과정은 오분류 유도에 성공하거나 변경 가능한 픽셀의 최대 개수에 도달할 때까지 반복된다.

3.1.3 DeepFool

Deepfool[5]은 입력 데이터를 원래의 클래스와 가장 가까운 다른 클래스로 오분류하도록 만드는 ℓ_2 Norm 기반 무표적 공격이다. 이를 위해 모델의 예측 결과와 가장 가까운 다른 클래스와의 거리를 계산하고, 해당 거리가 최소화되도록 입력 데이터에 노이즈 벡터를 추가한다. DeepFool의 원리는 [그림 3]과 같다.

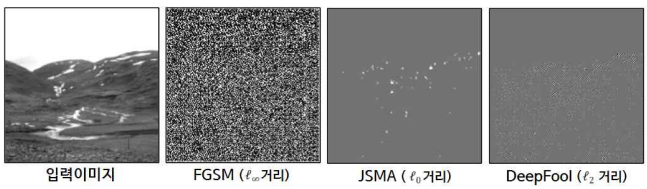


[그림 3] DeepFool의 원리[5]

DeepFool은 입력 이미지 x_0에서 시작하여 모델이 분류한 클래스와 다른 클래스에 속하는 가장 가까운 초평면(Hyperplane)을 찾는다. 이 초평면은 모델의 클래스를 나누는 경계(Boundary)를 의미한다. 다음으로, 초평면과 x_0를 잇는 벡터 r의 크기를 측정하고, 이 벡터를 초평면과 수직인 방향으로 최대한 이동시켜 초평면과 만나는 새로운 지점을 찾는다. 이 과정은 다시 새로운 지점에서 반복 수행되며, 벡터의 크기는 각 단계마다 줄어들게 된다. 이를 통해서 최소한의 크기로 모델의 오분류를 유도하는 노이즈 벡터를 구할 수 있다.

3.1.4 노이즈 삽입 패턴 비교분석

각 ℓ_p Norm에 따른 노이즈를 이미지로 출력해서 비교하면 [그림 4]와 같다. 해당 이미지는 512*512 크기의 흑백이미지 데이터셋인 BossBase v.1.01[6]에서 1장을 256*256으로 리사이징하여 사용했다.



[그림 4] 각 공격 기법에 따른 노이즈 삽입 패턴

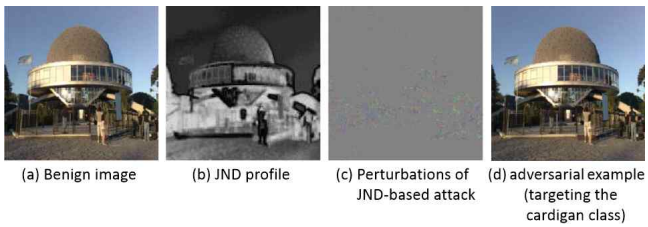
노이즈 삽입 패턴 출력을 통해, ℓ_∞ Norm을 사용하는 FGSM은 분류 모델의 오분류를 유도하기 위하여 입력 이미지의 모든 픽셀에 노이즈를 삽입했다는 것을 확인할 수 있다. 반면 ℓ_0 Norm을 사용하는 JSMA는 입력 이미지의 0.2%(117 픽셀)에만 노이즈가 삽입되었다. 평균적으로, JSMA는 입력 이미지의 4.02%에만 노이즈를 삽입하여 오분류를 달성한다[4]. ℓ_2 Norm을 사용하는 DeepFool은 ℓ_∞ Norm과 마찬가지로 모든 픽셀에 노이즈를 삽입했지만, 각 노이즈는 매우 미세한 크기로 삽입되었다. 이는 ℓ_2 Norm의 경우 입력 이미지와 적대적 예제 간 최단 거리를 측정하기 때문이다. [그림 4]에서 DeepFool의 노이즈 크기는 평균 0.002 화소값(Pixel Value)이었다. 일반적으로, DeepFool의 노이즈 크기는 FGSM의 1/5 수준이다[5]. 이와 같은 비교분석을 통해, 각 공격 기법이 삽입한 노이즈 패턴 간의 차이점을 확인할 수 있었다.

3.2 최신 적대적 예제 공격 기법

최근 연구에서는 ℓ_p Norm이 실제 사람의 시각체계(Human Visual System)로 인지하기 어려운 노이즈를 나타내는데 효과적이지 않다고 말하고 있다. 이는 ℓ_p Norm의 경우 이미지의 각 픽셀을 동등하게 취급하기 때문이다. 이에 따라, ℓ_p Norm 대신 다양한 지각적 거리(Perceptual Distance)를 적용하여 적대적 예제를 생성하려는 시도가 이루어지고 있다.

3.2.1 JND-based Attack

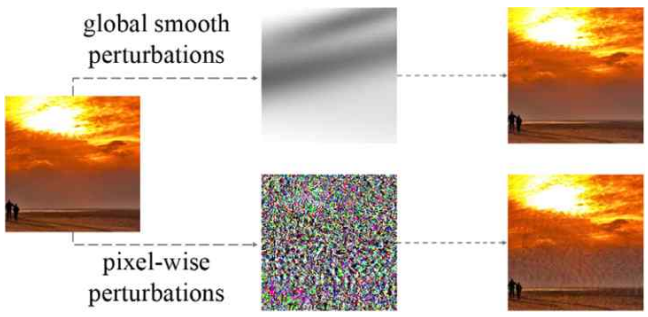
2021년에 발표된 JND-based Attack[7]은 거리 지표로 JND_p(Just Noticeable Distortion)를 사용한다. JND_p의 핵심은 같은 양의 노이즈를 삽입하더라도 RGB 채널과 픽셀의 위치에 따라 사람의 시각체계에서는 다른 효과를 준다는 것이다. 이에 착안하여, JND_p는 시각적으로 인지할 수 없는 색상 변화의 임계값을 측정한다. 또한, JND-based Attack은 이미지의 형태를 분석하여 JND 프로파일을 만들고, 이를 기반으로 노이즈를 삽입하는 방식을 통해 시각적인 차이를 더욱 최소화한다.



[그림 5] JND-based Attack의 노이즈 삽입 과정[7]

3.2.2 GMM(Gaussian Mixture Model)

2022년에 발표된 GMM[9]은 거리 지표로 LPIPS(Learned Perceptual Image Patch Similarity)[8]를 이용한다. LPIPS는 인지적으로 중요한 이미지 패치(Patch) 간의 유사도를 측정한다. 이와 함께, GMM은 픽셀 단위(Pixel-wise) 노이즈가 아니라 인접 픽셀과의 연관성를 고려하여 이미지 전체적으로 부드러운(Global Smooth) 노이즈를 삽입한다.



[그림 6] GMM(상단)과 FGSM(하단)의 노이즈 삽입 패턴 비교[9]

4. 결론

본 논문에서는 적대적 예제 공격의 정의와 분류에 대해 알아보고, 대표적인 공격 기법과 최신 공격 기법들의 원리 및 노이즈 삽입 패턴을 비교해보았다. 이를 통해, 각 공격 기법이 사용한 거리 지표에 따라서 노이즈 삽입 패턴에 어떠한 차이점이 있는지 확인해볼 수 있었다. 특히, 최근 연구에서는 ℓ_p Norm 대신 다양한 지각적 거리를 적용하여 인간의 시각체계에서 인지될 가능성을 더욱 낮추려는 시도가 이루어지고 있다.

본 연구를 바탕으로 향후에는 적대적 예제 공격 기술의 동향에 대해 더욱 심도 있는 연구를 수행하고자 한다. 거리 지표와 각 공격 기법의 원리, 노이즈 삽입 패턴 비교분석을 상세하게 다루고, 최신 공격 기법들을 추가적으로 살펴볼 예정이다.

참고문헌

[1] Y. LeCun et al., “Deep learning,” Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[2] C. Szegedy et al., “Intriguing properties of neural networks,” arXiv preprint arXiv:1312.6199, 2013.

[3] I. J. Goodfellow et al., “Explaining and Harnessing Adversarial Examples,” arXiv preprint arXiv:1412.6572, 2014.

[4] N. Papernot et al., “The Limitations of Deep Learning in Adversarial Settings,” in 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrucken, Germany, 2016, pp. 372-387.

[5] S. Moosavi-Dezfooli et al., “DeepFool: a simple and accurate method to fool deep neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574-2582, 2016.

[6] P. Bas et al., “Break our steganographic system – the ins and outs of organizing BOSS,” in 13th International Workshop on Information Hiding, vol. LNCS 6958, Springer Berlin Heidelberg, 2011, pp. 59-70.

[7] Z. Wang et al., “Invisible Adversarial Attack against Deep Neural Networks: An Adaptive Penalization Approach,” in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 3, pp. 1474-1488, 2021.

[8] R. Zhang et al., “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 586-595.

[9] Y. Liu et al, “Transferable adversarial examples based on global smooth perturbations,” Computers & Security, vol. 121, 102816, 2022.