Adversarial Attacks

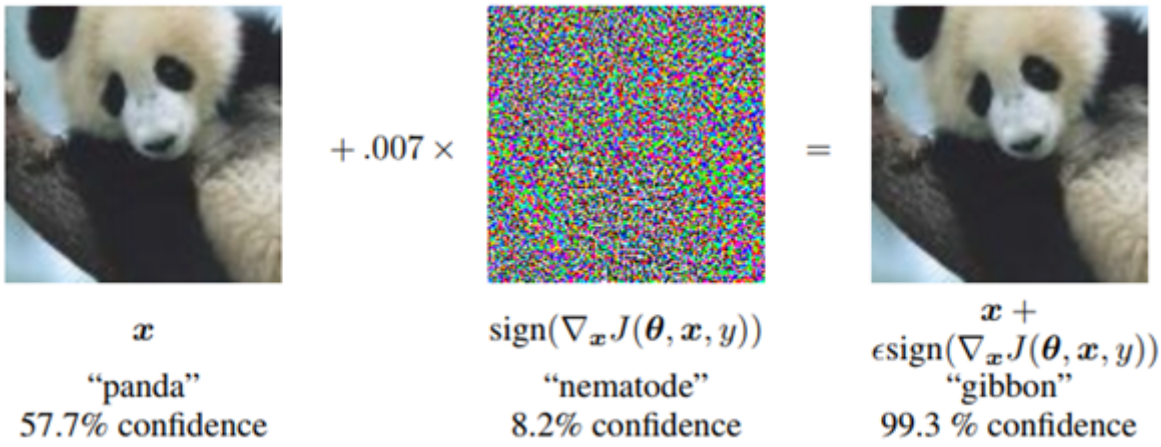
Confusing?

물론 사람은 이 사진들을 다 구분할 수 있다. 그러나 기계는 그렇지 않다!

Introduction

Adversarial Attack

- Change the result of the model by adding imperceptible(눈에 띄지 않는) noise to input
- To deceive image classification model



<Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014)>

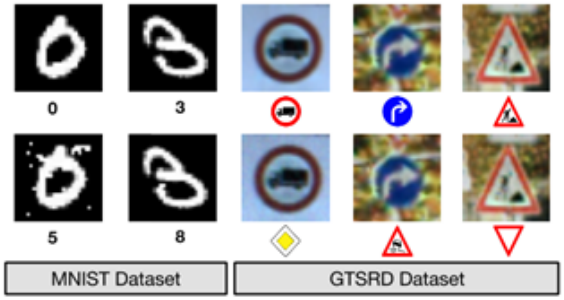
위 그림을 우리 눈으로 보면 좌, 우측에 판다가 있고 중앙에 노이즈처럼 생긴 그림이 있다. 문제는 좌측의 판다를 57.7%의 신뢰도로 '판다'라고 분류 가능한 분류기가 있다고 가정할 때, 공격자는 중앙에 보이는 것과 같은 노이즈를 주어 우측 판다 이미지를 합성하게 된다. 그 결과 우측 판다 이미지는 우리에게는 여전히 '판다' 이미지로 보이지만 이전의 동일한 분류기로 이미지 식별 결과 99.3%의 신뢰도로 '긴팔원숭이'로 판단하게 된다는 것이다.

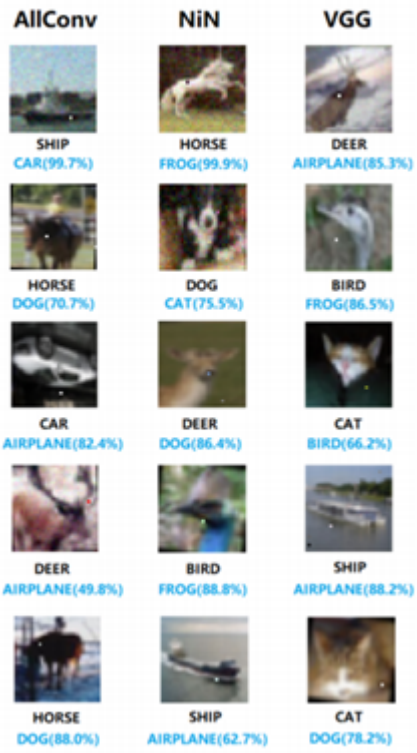
즉, 사람이 구분할 수 없는 노이즈를 이미지에 합성시켜 인공지능이 잘못된 예측을 하도록 유도할 수 있다는 의미이다.

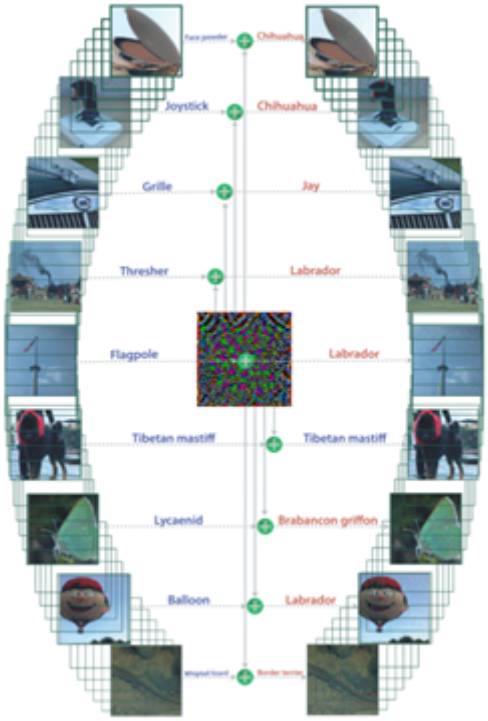
이렇게 '긴팔원숭이'로 분류되도록 만들어진 sample image를 Adversarial Examples(AEs)라고 한다.

AEs는 Training단계에서 만들어지지만 DNN이 Training된 후에는 Train 과정에 대한 변경이 필요하지 않다. 즉, **Real Image의 일부 픽셀 변화로 생성하는 것이다.**

Varieties of Adversarial Attacks



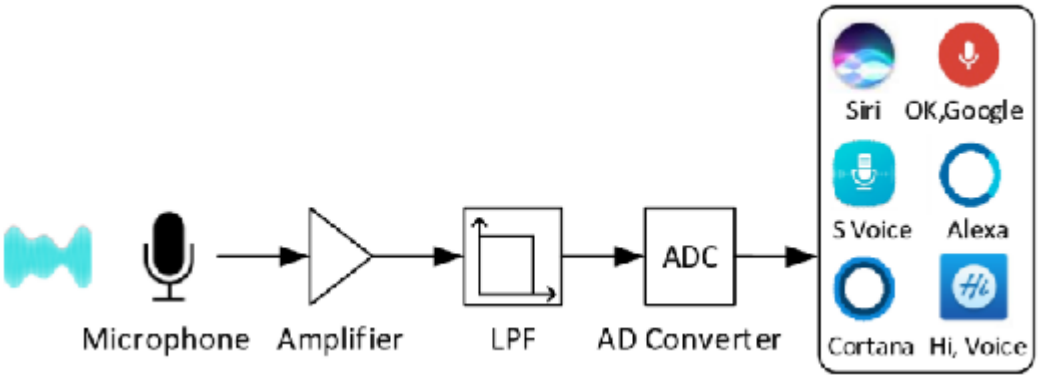




Real World

True image

Adversarial image



실제 AEs를 이용한 공격 사례에 대해서 살펴보자

워싱턴 대학의 타다요시 코노 박사 연구팀에 따르면 도로 교통 표지판에 작은 스ㅡ티커를 붙여 '정지' 신호를 '시속45마일 속도제한' 신호로 오분류하게 만드는 방법을 발견해내었다. 연굩팀은 이 사이버 공격 수법을 'Robust Physical Perturbations(RP2)'라고 명명했다. 이 공격 수법을 이용해서 특정 모델에 대한 적응형 또는 맞춤형 이미지를 생성하게 될 수 있다는 위험성을 강조하였다.

이러한 AEs가 동작하는 원인은 뭘까?

현재의 DNN기술에 기반을 둔 Classification Algorithm은 Natural Image 즉, Real World에 존재하 수 있는 이미지에 대해서는 정상적으로 동작하도록 설계되었지만 사람이 일부 변화를 가한 AEs에 대해서는 취약할 수 있다.

일반적으로 디지털 이미지들은 8bit의 Resolution을 갖고 있으며 이는 256개의 정보 차이를 구분할 수 있음을 의미한다. 하지만 256개 이하의 정보가 변경되는 경우 차이를 육안으로 구별할 수 없다. 그러나 Neural Network에서는 입력된 데이터에 가중치가 곱해지고 바이어스가 더해지게 되는데, 이때 원본 데 이터에서 1/256보다 작은 입력이 더해져도 가중치가 큰 경우 1/256의 정밀도 이상의 값이 결과에 반영될 수 있다. Layer가 복잡한 Network에서는 작은 변화에도 이러한 과정이 일반 데이터와 함께 증폭이 일어나게 되고 이런 증포이 계속되면 최종적으로 Misclassification이 발생하 수 있는 것이다.

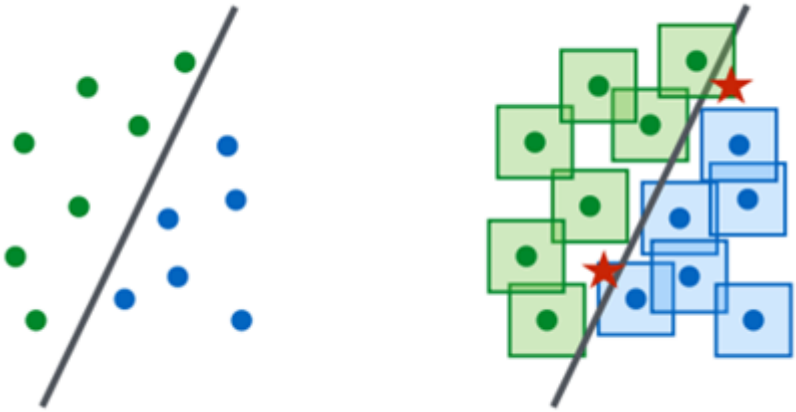
Adversarial examples poses a security concern for machine learning models

- An attack created to fool one network also fools other networks.
- Attacks also work in the physical world.
- For Deep Neural Networks, it is very easy to generate adversarial examples but this issue affects other ML classifiers.
- Many defense strategies have been proposed, they all fail against strong attacks.

Why it happens?

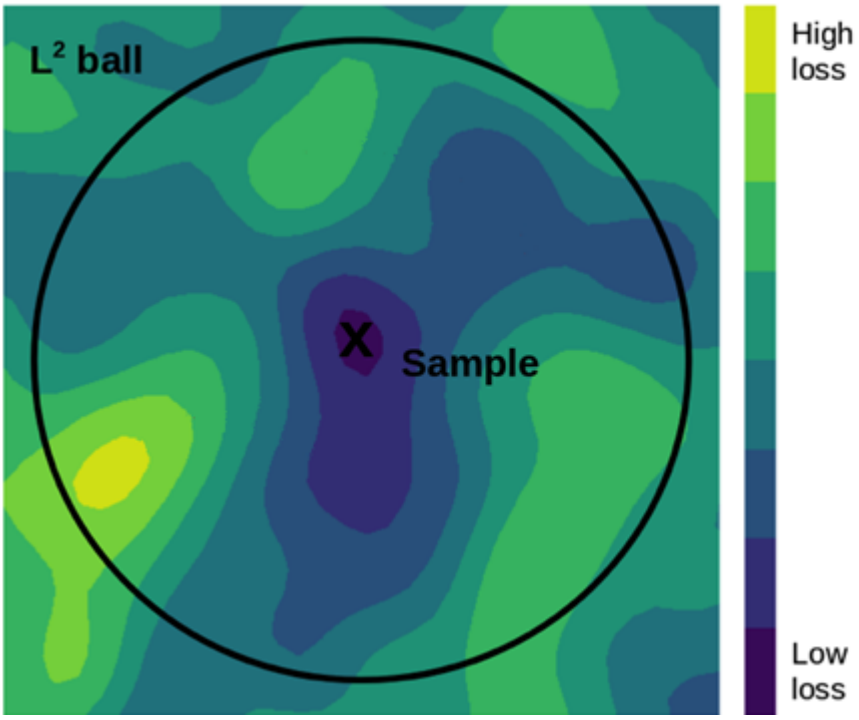
Sample cross the decision boundary

- Small differences on input pixel contribute to a dramatic difference in weights x inputs(아까 위에서 했던말)
- Simple decision boundary does not separate the l∞-balls aroound the data points



<Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).
https://towardsdatascience.com/know-your-adversary-understanding-adversarial-examples-part-1-2-63af4c2f5830
Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks." arXiv preprint arXiv:1706.06083(2017).>

위의 오른쪽 그림의 정사각형 영역은 l∞영역이다. (adversarial 분야에서는 L∞을 사용)그런데 boundary는 이걸 자르고 있다.



사람은 이 원안(L2 ball)에 있는건 동일한 x로본다.이 그림의 가로를 x1축,세로를 x2축이라고 하자. X=(x1,x2)

Adversarial Example

An example X(Tilde) is said adversarial if:

- It is close to a sample in the true distribution:

$$D(X, \tilde{X}) \leq \epsilon$$

- It is misclassified

$$\text{argmax}\, P(y|\tilde{X}) \neq y_{true}$$

y_true = argmaxP(y|X) 즉, X의 label값이다.

- It belongs to the input domain. E.g. for images

$$0 \leq \tilde{X} \leq 255$$

To measure the similarity between samples:

- A good measure between samples is still an active area or research. Commonly,researchers use:
- L2 norm(Euclidean distance):

$$\left\|\tilde{X} - X\right\|_2$$

- L∞ norm

$$\left\|\tilde{X} - X\right\|_\infty = \max_{ij}\left|\tilde{X}_{ij} - X_{ij}\right|$$

각 축 의 차의 최대값, L∞ norm을 많이 사용한다.

Distance Measure

$$\|\boldsymbol{x}\|_p = \left(\sum_{i-1}^{N} |x_i|^p\right)^{\frac{1}{p}}$$

- ℓ_0: control the number of pixels that are modified
- ℓ_1: control the total amount of pixel value changes
- ℓ_2: control the Euclidean distance of pixel value changes
- ℓ_∞: control the maximum pixel value change in any coordinate

Surprise!!

- They are not specific to a particular architecture.

- Even the misclassified classes are mostly the same across various models. (서로 다른 NN(model)일지라도 training data가 같으면 AEs를 똑같이 구분을 잘 못한다.)

Some causes

- Non-linearity of neural networks
- Insufficient regularization
- Insufficient model averaging

Threat Models

Black-box:

- Access to domain data,
- Can feed inputs to the model and obsesrve outputs.

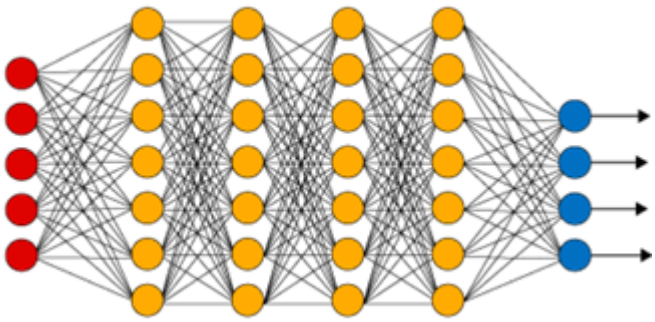
공격자는 모델의 정보를 알 수 없으며 오직 모델의 입력과 출력 패턴을 보고 공격을 하는것이다. 따라서 white box attack보다 공격이 까다롭다.

White-box:

- Full knowledge of the data, architecture and parameters.

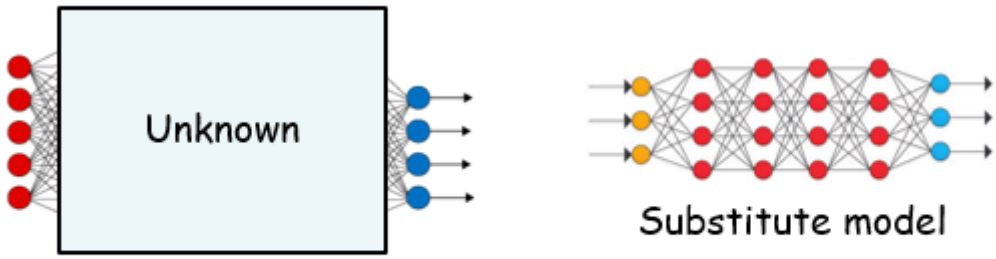
White-box Attacks

- Every parameter of the network is known
- Analyze the network and find out adversarial examples



Black-box Attacks

- parameters of the network are not known, but training data is open
- Build a substitute model using the training data
- Find out adversarial examples for the substitute model
  - The adversarial examples can also fool the original network.



Targeted vs. Untargeted Attacks

According to the attacker's goal:

Non-targeted attacks: attacker tries to fool a classifier to get any incorrect clas

$$\text{argmax}\, P(\mathbf{y}|\tilde{X}) \neq y_{\text{true}}$$

Targeted attacks: attacker tries to fool a classifier to predict a particular class

$$\text{argmax}\, P(\mathbf{y}|\tilde{X}) = y_{\text{target}}$$

untargeted 는 주어진 target을 틀리는 x를 만드는것

targeted는 주어진 target을 맞추는 새로운 x를 만드는것

Attacks

One Step Gradient Method

Untargeted : Fast Gradient Sign Method(FGSM)

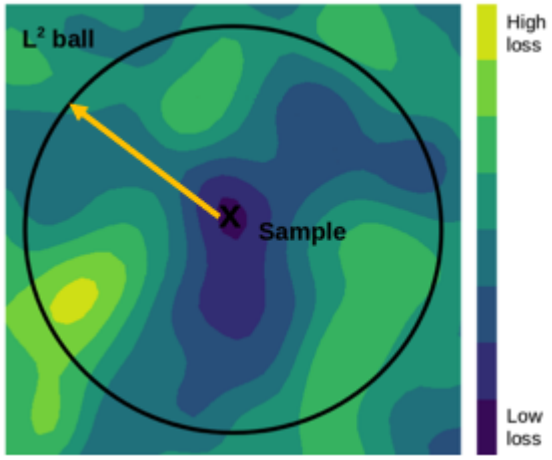
- Add noise by finding reverse gradient direction(gradient ascent) of true label
  - $$x^* = x + \epsilon \text{sign}\left(\nabla_x L(x, y_{true})\right)$$
- 공격의 품질보다는 speed를 우선시 한다.

신경망의 gradient를 이용해 적대적 샘플을 생성하는 기법이다. 만약 모델의 입력이 이미지라면, 입력 이미지에 대한 손실 함수의 gradient를 계산하여 그 손실을 최대화하는 이미지를 생성한다. 이처럼 새롭게 생성된 이미지를 적대적 이미지라고 한다.

여기서 흥미로운 사실은 입력 이미지에 대한 그래디언트가 사용된다는 점이다. 이는 손실을 최대화하는 이미지르 생성하는 것이 FGSM의 목적이기 때문이다. 요약하자면, 적대적 샘플은 각 픽셀값에 왜곡을 추가함으로써 생성할 수 있다. 각 픽셀의 기여도는 연쇄 법칙을 이용해 그래디언트를 계산하는 것으로 빠르게 파악할 수 있다. 이것이 입력 이미지에 대한 그래디언트가 쓰이는 이유이다. 또한, 대상 모델은 더이상 학습하고 있지 않기 때문에(따라서 신경망의 가중치에 대한 그래디언트는 필요하지 않다)모델의 가중치값은 변하지 않는다.
FGSM의 궁극적인 목표는 이미 학습을 마친 상태의 모델을 혼란시키는 것이다.

Targeted : One step least-likely class method

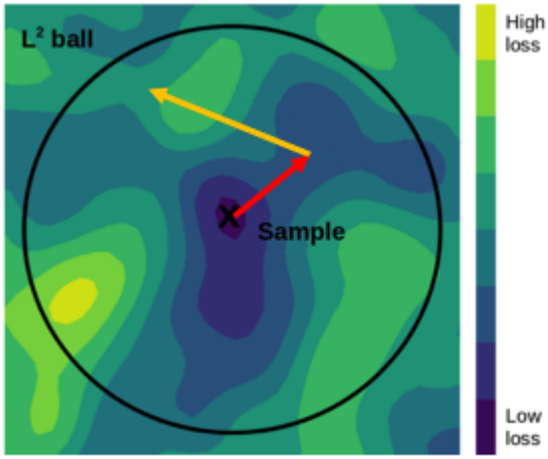
- Add noise by finding gradient direction of target
  - $$x^* = x - \epsilon \text{sign}\left(\nabla_x L(x, y_{target})\right)$$
- Mostly, least likely class is used



다음 그림은 FGSM을 나타낸다. 화살표의 방향은 sign함수를 나타낸다. 즉 방향만생각한다. 또한 입실론은 저 화살표의 길이다.

Randomized Fast Gradient Sign Method(RAND + FGSM)

- First apply a small random perturbation before using FGSM
  - $x' = x + \alpha \text{sign}(\mathcal{N}(0^n, \text{I}^n)), \alpha < \epsilon$
  - $x^* = x' + (\epsilon - \alpha)\text{sign}\left(\nabla_x L(x, y_{true})\right)$
- Effective and Fast
  - It is used for baseline, usually



Iterative Methods

Basic iterative method(iter. basic)

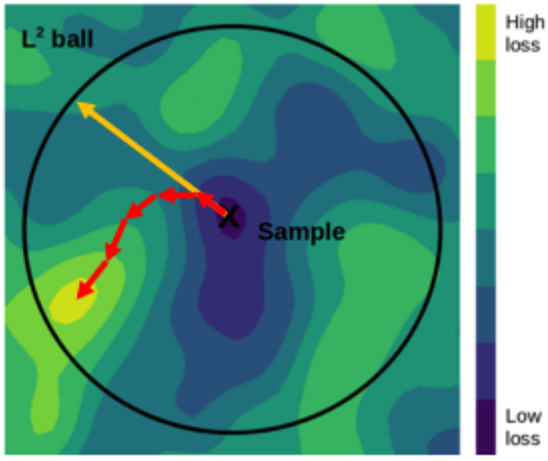
- Add noise by finding reverse gradient direction of true label, iteratively

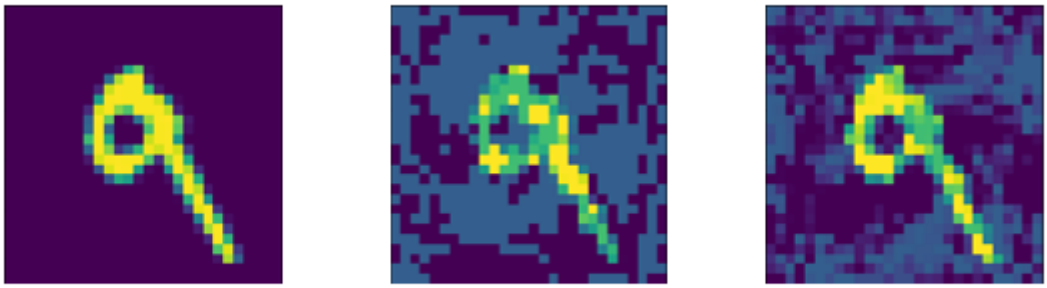
$$x_0^* = x,\ x_{N+1}^* = Clip_{x,\epsilon}\{x_N^* + \alpha \text{sign}\left(\nabla_x L(x_N^*, y_{true})\right)\}$$

Iterative least-likely class method(iter. l.l.)

- Add noise by finding gradient direction of target label, iteratively

Iterative methods are stronger than one step methods





Projected Gradient Descent(PGD)

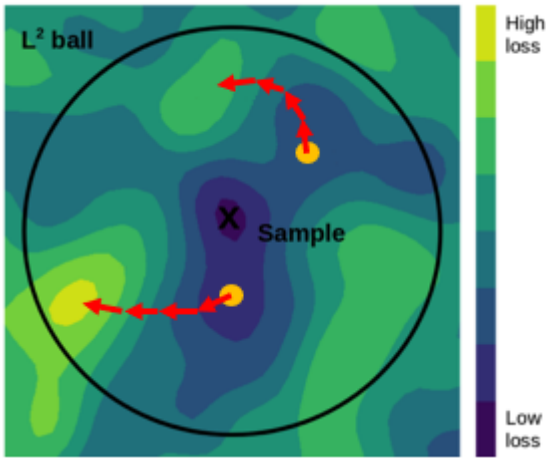
- Apply FGSM iteratively based on random starting points around data, and restart at random points if fails
- Universal first order adversary

- $x^{t+1} = \prod_{x+S} x^t + \alpha \mathrm{sgn}(\nabla_x L(\theta, x, y))$

or

- $x^{t+1} = x^t + \prod_S \alpha \text{sgn}(\nabla_x L(\theta, x, y))$

- Pros
  - Known as strongest attack
- Cons
  - Slow



전에 말한 iterative model방법을 통해 loss가 큰곳으로 이동한다. 그러나 loss가 커서 속일 확률이 높을뿐이지 속이는게 아니다 . 그래서 PGD가 나오게 된것이다.

Defense

Defense Approaches

Adversarial Training

- Train model using adversarial examples as well as natural data
- 학습→공격→공격이미지 학습 데이터에추가→다시 학습→공격→....

Filtering/Detecting

- Learning pattern of adversarial examples or perturbations
- Reject adversarial samples without classifying them using a specialized side model
- 그러나 filtering,detecting이 다 NN으로 만들어졌기 때문에 공격받을수있다.

Denoising(preprocessing)

- Reduce noise in the input using denoiser
- 이것도 NN으로 만들어졌기 때문에 공격받을수 있다.

Adversarial Training+α

- Adversarial Logit Pairing
- Defensive Distillation
- Label Smoothing

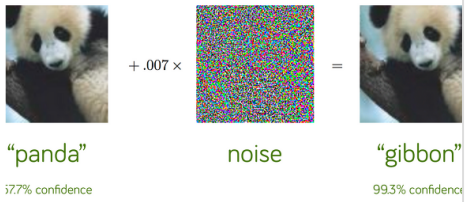
References

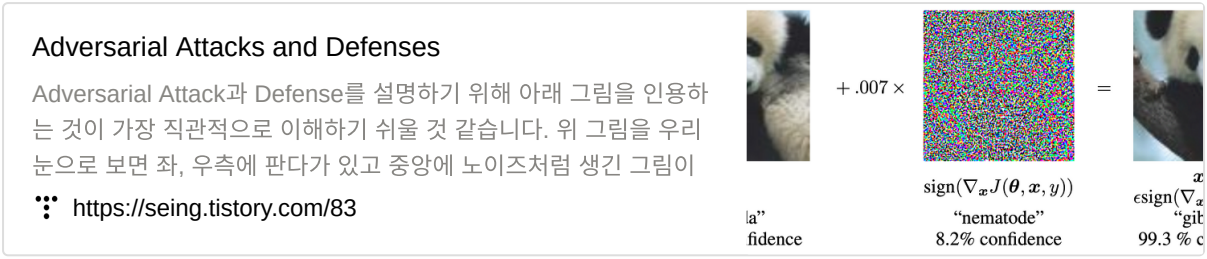
Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014)

적대적 공격 동향(Adversarial Attacks Survey)

Deep Learning 을 활용한 Neural Network 기술들이 등장하면서 Machine Learning의 다양한 공학적 접근이 개발되고 있습니다. ImageNet과 같은 거대한 데이터셋에 대해서도 효율적으로 연산할 수

https://rain-bow.tistory.com/entry/Adversarial-Attack





Papernot, Nicolas, et al. "Practical black-box attacks against machine learning." Proceedings of the 2017 ACM on Asia conference on computer and communications security. ACM, 2017.
Moosavi-Dezfooli, Seyed-Mohsen, et al. "Universal adversarial perturbations." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." IEEE Transactions on Evolutionary Computation (2019).

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).
https://towardsdatascience.com/know-your-adversary-understanding-adversarial-examples-part-1-2-63af4c2f5830
Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks." arXiv preprint arXiv:1706.06083(2017).