Towards Deep Learning Models Resistant to Adversarial

2018 ICLR

Abstract

Abstract

Recent work has demonstrated that deep neural networks are vulnerable to adversarial examples—inputs that are almost indistinguishable from natural data and yet classified incorrectly by the network. In fact, some of the latest findings suggest that the existence of adversarial attacks may be an inherent weakness of deep learning models. To address this problem, we study the adversarial robustness of neural networks through the lens of robust optimization. This approach provides us with a broad and unifying view on much of the prior work on this topic. Its principled nature also enables us to identify methods for both training and attacking neural networks that are reliable and, in a certain sense, universal. In particular, they specify a concrete security guarantee that would protect against *any* adversary. These methods let us train networks with significantly improved resistance to a wide range of adversarial attacks. They also suggest the notion of security against a *first-order adversary* as a natural and broad security guarantee. We believe that robustness against such well-defined classes of adversaries is an important stepping stone towards fully resistant deep learning models. [1]

- Neural network의 adversarial robustness를 robust optimization 측면에서 해석
- Saddle point formulation 을 이용하여 robustness를 guarantee 할 수 있으며 이 formulation은 attack과 defense를 모두 다룬다.
- 특히 adversarial training을 통해 가장 최적인 지점(saddle point)를 찾아 다른 attack에 대해서도 robust 할 수 있음을 이야기 한다.

Contribution

1. Saddle point formulation에 잘 optimized 될 수 있는 모델을 만드는 법을 제시한다.
   (First-order method인 PGD attack을 통해 해결)

2. Adversarial robustness를 위해서는 capacity가 큰 모델이 필요함을 밝힘.

3. MNIST 와 CIFAR-10 에 대해서 훈련한 모델이 광범위한 adversarial attac에 대해 robust 하도록 train 할 수 있다.

Optimization View on Adversarial Robustness

$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{S}} L(\theta, x+\delta, y) \right] . \tag{2.1}$$

Inner maximization problem : 주어진 data x 가 가장 높은 loss를 가지는 adversarial version을 찾는 것

Outer maximization problem : Inner attack problem에 의해 주어진 "adversarial loss"가 최소화 될 수 있는 model parameter를 찾는것

Optimization View on Adversarial Robustness

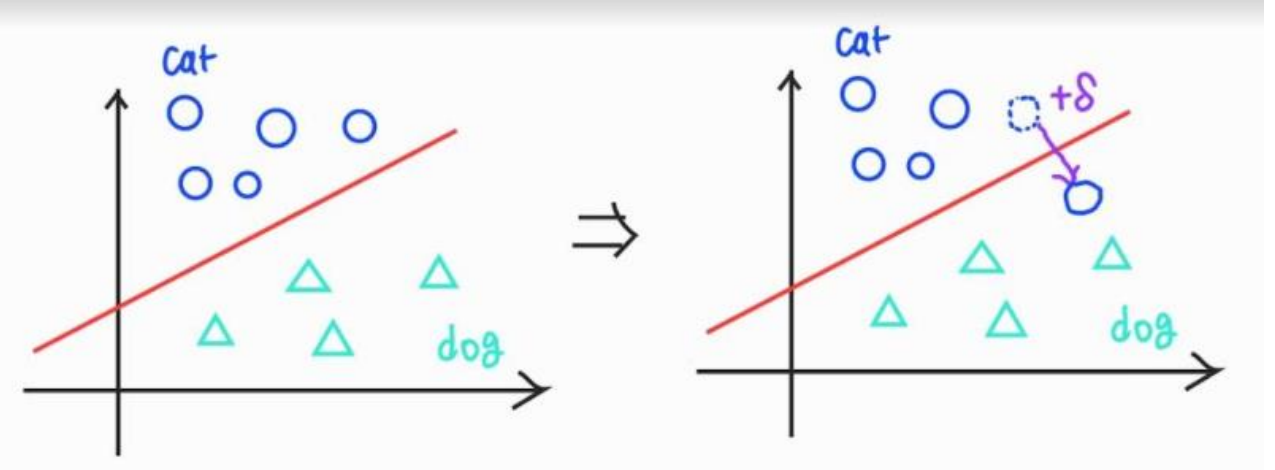
$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{S}} L(\theta, x+\delta, y) \right] . \tag{2.1}$$

Inner maximization problem : 주어진 data x 가 가장 높은 loss를 가지는 adversarial version을 찾는 것

Outer maximization problem : Inner attack problem에 의해 주어진 "adversarial loss"가 최소화 될 수 있는 model parameter를 찾는것

Optimization View on Adversarial Robustness

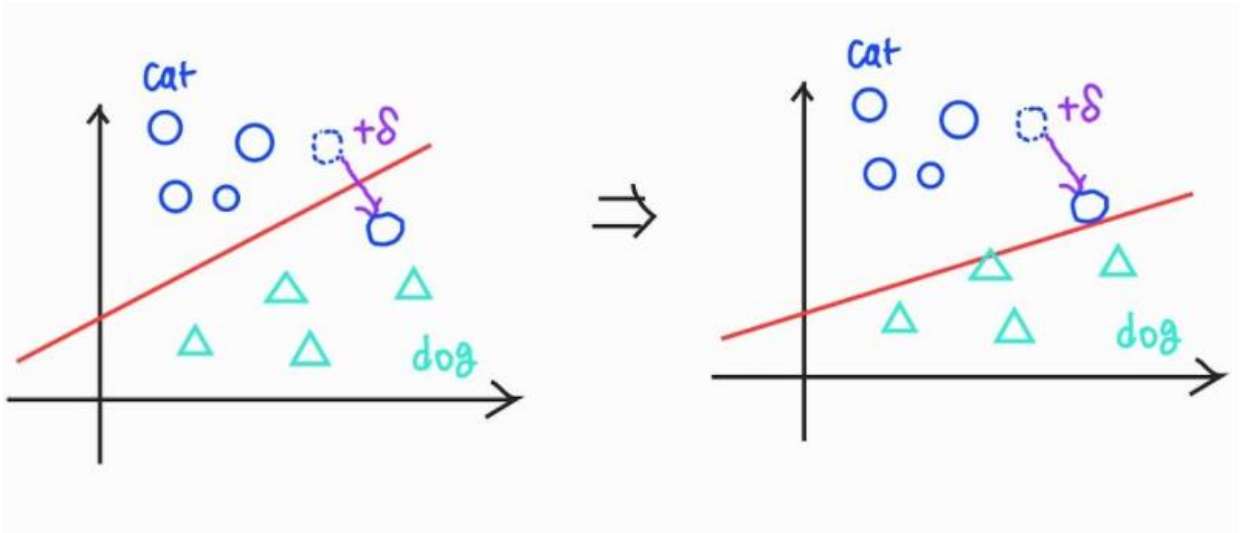
$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in \mathcal{S}} L(\theta, x+\delta, y) \right]. \tag{2.1}$$



Inner maximization problem : 주어진 data x 가 가장 높은 loss를 가지는 adversarial version을 찾는 것

Optimization View on Adversarial Robustness

$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{S}} L(\theta, x+\delta, y) \right] . \tag{2.1}$$



Outer maximization problem : Inner attack problem에 의해 주어진 "adversarial loss"가 최소화 될 수 있는 model parameter를 찾는것 (이 때 adversarial example 만을 가지고 학습을 한다는 점에 주목해야 함)

A Unified View on Attacks and Defenses

Attack : 어떻게 하면 적은 perturbation이 들어간 strong attack을 만드는가?

Defense : 어떻게 하면 Adversarial examples가 없거나 찾기 힘들게 모델을 train 할 수 있는가?

$$x + \varepsilon \operatorname{sgn}(\nabla_x L(\theta, x, y)).$$

FGSM

Loss 값에 대해 gradient를 계산, Gradient의 부호 방향으로 epsilon 만큼 이미지를 변경 (L∞ attack)

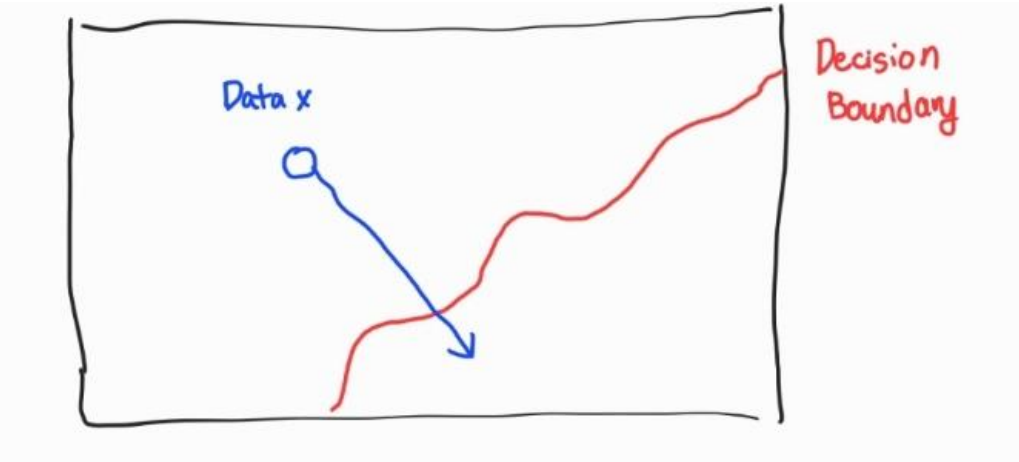
$$x^{t+1} = \Pi_{x+\mathcal{S}} \left(x^t + \alpha \operatorname{sgn}(\nabla_x L(\theta, x, y)) \right).$$

PGD Attack

같은 L∞ attack 으로 간단히 말하면 FGSM을 step 단위로 나눈 공격 방법 (+random perturbation)

Projected Gradient Descent

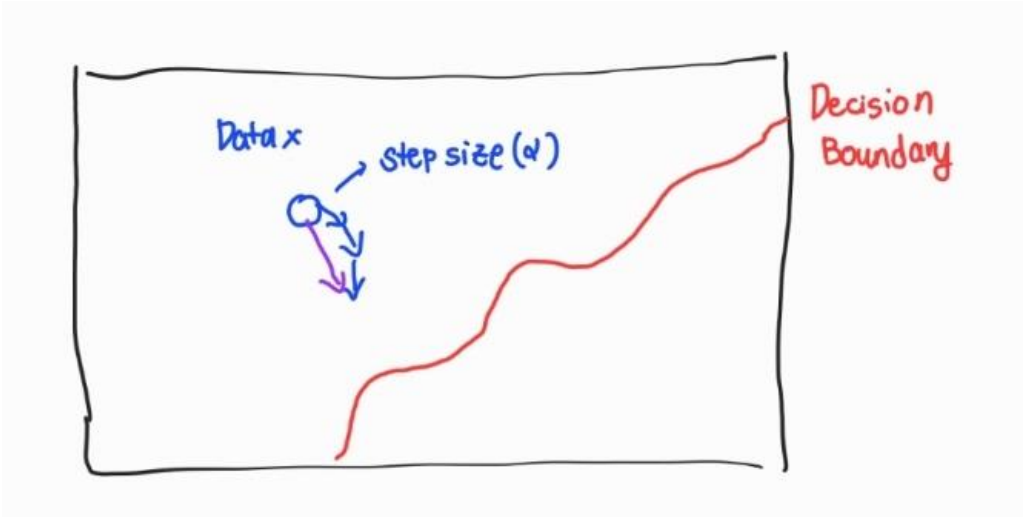
$$x^{t+1} = \Pi_{x+\mathcal{S}}\left(x^t + \alpha \operatorname{sgn}(\nabla_x L(\theta, x, y))\right).$$



Loss gradient의 부호를 계산해서 perturbation의 방향을 결정

Projected Gradient Descent

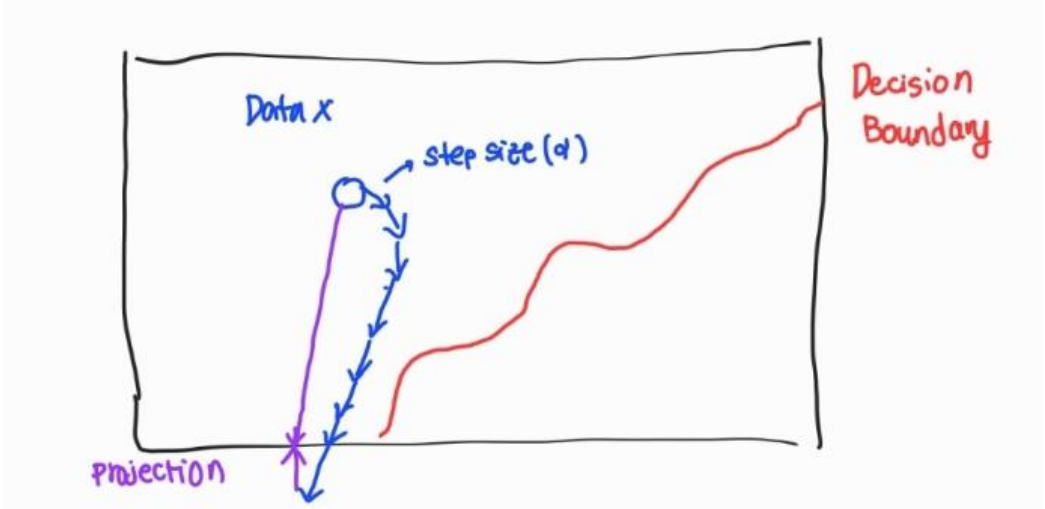
$$x^{t+1} = \Pi_{x+\mathcal{S}}\left(x^t + \alpha\,\mathrm{sgn}(\nabla_x L(\theta, x, y))\right).$$



작은 step 크기로 나누어 반복적으로 x^{t+1}update

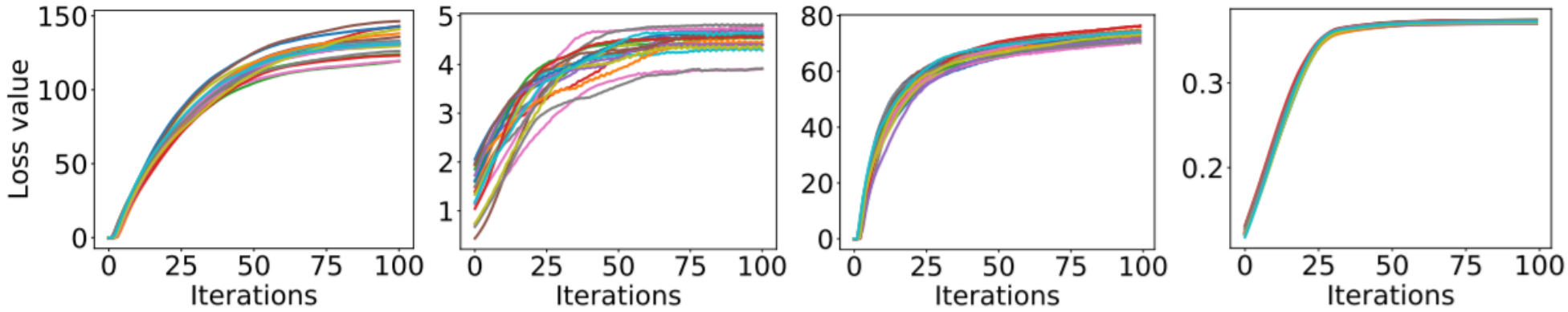
Projected Gradient Descent

$$x^{t+1} = \Pi_{x+\mathcal{S}}\left(x^t + \alpha\,\mathrm{sgn}(\nabla_x L(\theta, x, y))\right).$$



만약 bound를 벗어나게 되면 projection 수행

The Landscape of Adversarial Examples



(a) MNIST Standard training (b) MNIST Adversarial training (c) CIFAR10 Natural training (d) CIFAR10 Adversarial training

각각 MNIST와 CIFAR-10 에 대해 adversarial training 했을 때의 loss value를 standard training 했을 때와 비교

Random point에서 시작을 해도 결국 만들어진 AEs 는 비슷한 loss값을 지니게 됨

Adversarial training이 된 model의 loss value가 standard training에 비해 값이 월등히 작음

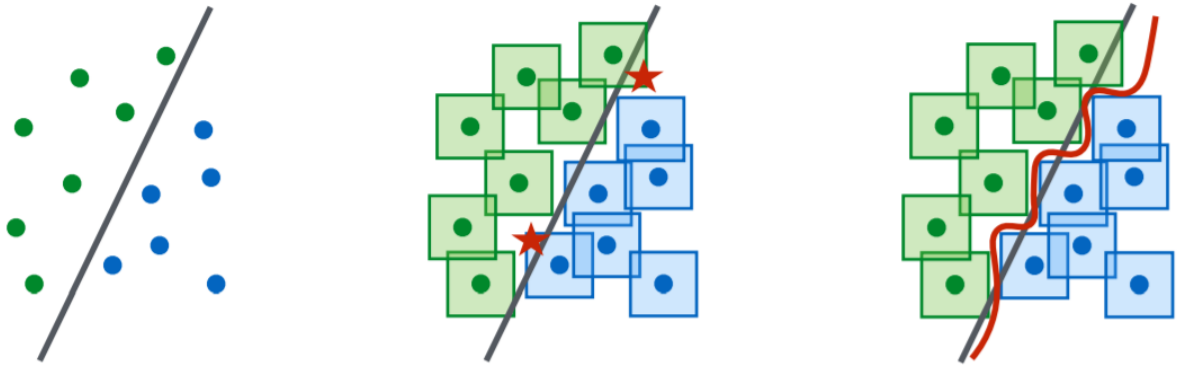
First-Order Adversaries

PGD로 만들어진 대부분의 example들은 비슷한 loss값을 지니며 PGD를 통해 만들어진 model은 대부분의 first-order adversary에 대해 robust하다고 할 수 있다.
실제로 실험결과 PGD attack 보다 더 나은 local maxima를 만드는 공격을 찾기 어려웠다.
즉, PGD에 대해서 충분히 robust 하다면 다른 attack들에 대해서도 충분히 robust하다.

Descent Directions for Adversarial Training

일반적으로는 SGD(Stochastic Gradient Descent)에 기반한 minimization 기법을 이용.
Saddle point problem에 대해서는 Danskin's theorem에 따라서 연속적으로 미분 가능한 함수에서는 학습이 잘 될 수 있을 것이라고 함.

Network Capacity and Adversarial Robustness

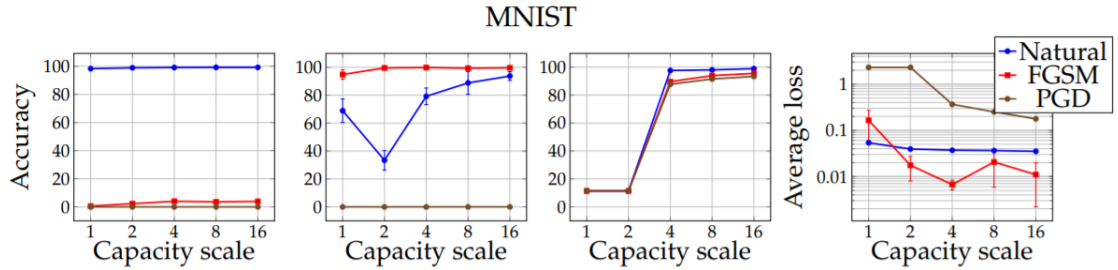


Standard network의 decision boundary의 경우 빨간색 별표 쳐진 부분처럼 l_∞ ball을 완전히 구분짓지 못함

Adversarially trained network의 decision boundary의 경우 조금 더 복잡하며 l_∞ ball을 완전히 구분지음

Adversarial trainin을 통해서 epsilon 값에 맞춰 model이 non-linear하게 학습됨을 관찰할 수 있다.

Experiments



	Simple	Wide
Natural	92.7%	95.2%
FGSM	27.5%	32.7%
PGD	0.8%	3.5%

(a) Standard training

	Simple	Wide
Natural	87.4%	90.3%
FGSM	90.9%	95.1%
PGD	0.0%	0.0%

(b) FGSM training

	Simple	Wide
Natural	79.4%	87.3%
FGSM	51.7%	56.1%
PGD	43.7%	45.8%

(c) PGD training

	Simple	Wide
Natural	0.00357	0.00371
FGSM	0.0115	0.00557
PGD	1.11	0.0218

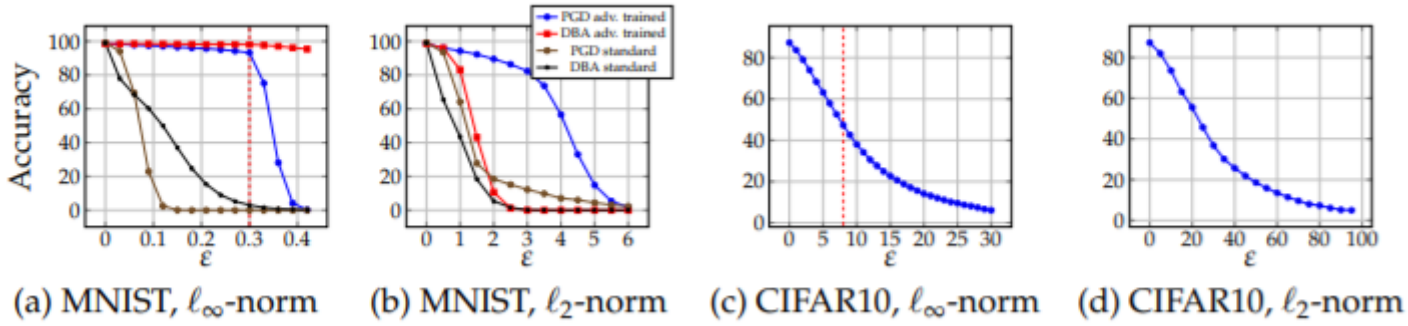
(d) Training Loss

Capacity가 robustness에 있어서 중요한 요소이며, 강력한 adversary에 대해서 train하는 데에도 중요함을 보여줌

MNIST dataset에 대해 아주 간단한 Convolutional network 에 대해 capacity 를 2배씩 키워나가며 결과를 관찰

CIFAR-10 에 대해서는 ResNet을 이용, data augmentation을 이용하였으며 capacity를 늘리기 위해 layer를 10의 배수로 결합시키도록 변환(Wide)

Experiments



Epsilon 값을 바꿔가며 실험을 진행 + L2 attack 에 대해서도 실험 진행

Epsilon의 경우 값이 조금만 증가해도 정확도가 많이 떨어짐

l_∞ attack 에 robust 하다면 l_2 attack 에 대해서도 어느 정도 robust 함을 알 수 있음

Experiments

Method	Steps	Restarts	Source	Accuracy
Natural	-	-	-	98.8%
FGSM	-	-	A	95.6%
PGD	40	1	A	93.2%
PGD	100	1	A	91.8%
PGD	40	20	A	90.4%
PGD	100	20	A	**89.3%**
Targeted	40	1	A	92.7%
CW	40	1	A	94.0%
CW+	40	1	A	93.9%
FGSM	-	-	A′	96.8%
PGD	40	1	A′	96.0%
PGD	100	20	A′	**95.7%**
CW	40	1	A′	97.0%
CW+	40	1	A′	96.4%
FGSM	-	-	B	**95.4%**
PGD	40	1	B	96.4%
CW+	-	-	B	95.7%

Method	Steps	Source	Accuracy
Natural	-	-	87.3%
FGSM	-	A	56.1%
PGD	7	A	50.0%
PGD	20	A	**45.8%**
CW	30	A	46.8%
FGSM	-	A′	67.0%
PGD	7	A′	**64.2%**
CW	30	A′	78.7%
FGSM	-	A_{nat}	85.6%
PGD	7	A_{nat}	86.0%

Adversarial trainin을 했을 때의 (좌) : MNIST 실험결과 (우) : CIFAR-10 실험결과

A: white box attack

A′ : 모델 구조는 같은데 weigh는 다른 모델

B : black box attack