

꼼꼼한 딥러닝 논문 리뷰와 코드 실습

Deep Learning Paper Review and Code Practice

나동빈(dongbinna@postech.ac.kr)

Pohang University of Science and Technology

CVPR 2018

Boosting Adversarial Attacks with Momentum

Yinpeng Dong, Fangzhou Liao, Tianyu Pang,
Hang Su, Jun Zhu, Xiaolin Hu, Jianguo Li

Tsinghua University, BNRist Lab, Intel Labs China

[배경 지식] 적대적 예제(Adversarial Examples)

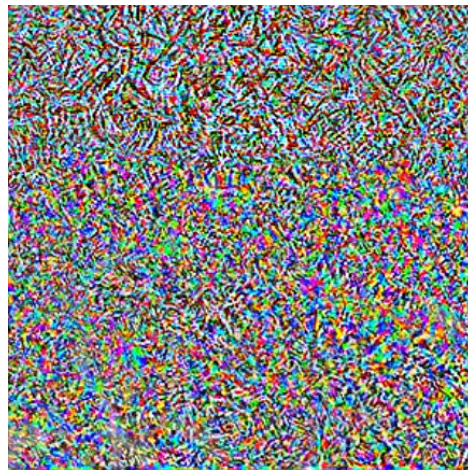
- 적대적 예제는 인간의 눈에 띄지 않게 조작된 데이터로, 딥러닝 모델의 부정확한 결과를 유도합니다.
- 많은 경우에 공격자는 데이터를 정해진 범위까지(눈에 띄지 않는 선에서) 조작할 수 있도록 설정합니다.
 - i.e., a norm-constrained perturbation is constrained below a specific constant ϵ .



x

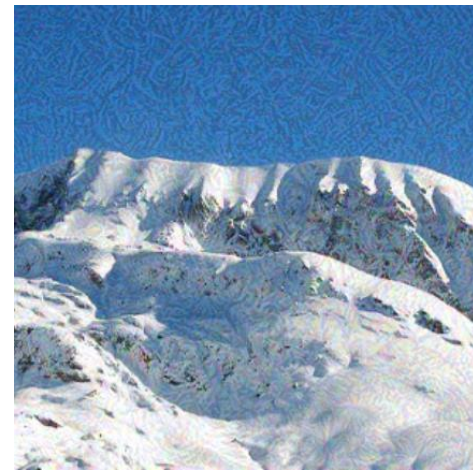
(Alps: 94.39%)

$+ \epsilon *$



Perturbation (δ)

$=$



x^*

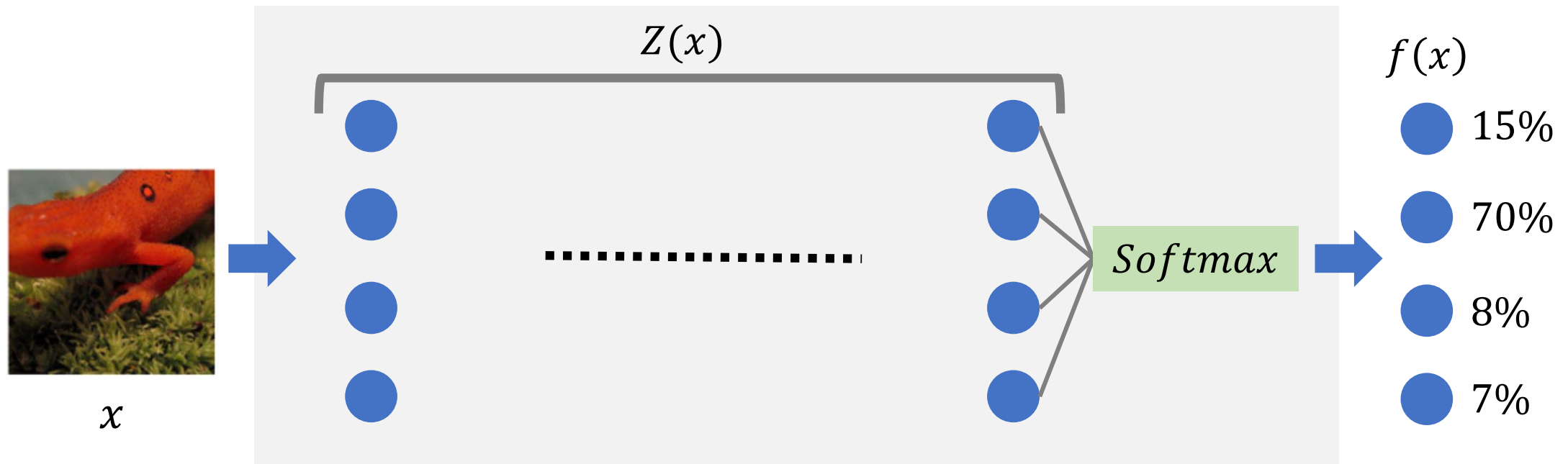
(Dog: 99.99%)

[배경 지식] Threat Model: 공격자가 어디까지 알고 있을까요?

- 화이트박스(White-box)
 - 공격자가 뉴럴 네트워크에 대한 모든 정보를 알고 있습니다.
 - 예시: 네트워크 아키텍처, 학습된 가중치, 학습 방법 등
 - 공격자는 비용(loss) 값이 낮을 때 분류 결과가 바뀌도록 비용 함수를 설정하여 공격할 수 있습니다.
- 블랙박스(Black-box)
 - 공격자가 뉴럴 네트워크에 대한 내부 정보를 알지 못합니다.
 - 예시: 입력값에 대한 네트워크의 최종 출력값
 - 공격자는 반복적으로 쿼리(query)를 날리거나 유사한 네트워크를 만들어 공격할 수 있습니다.

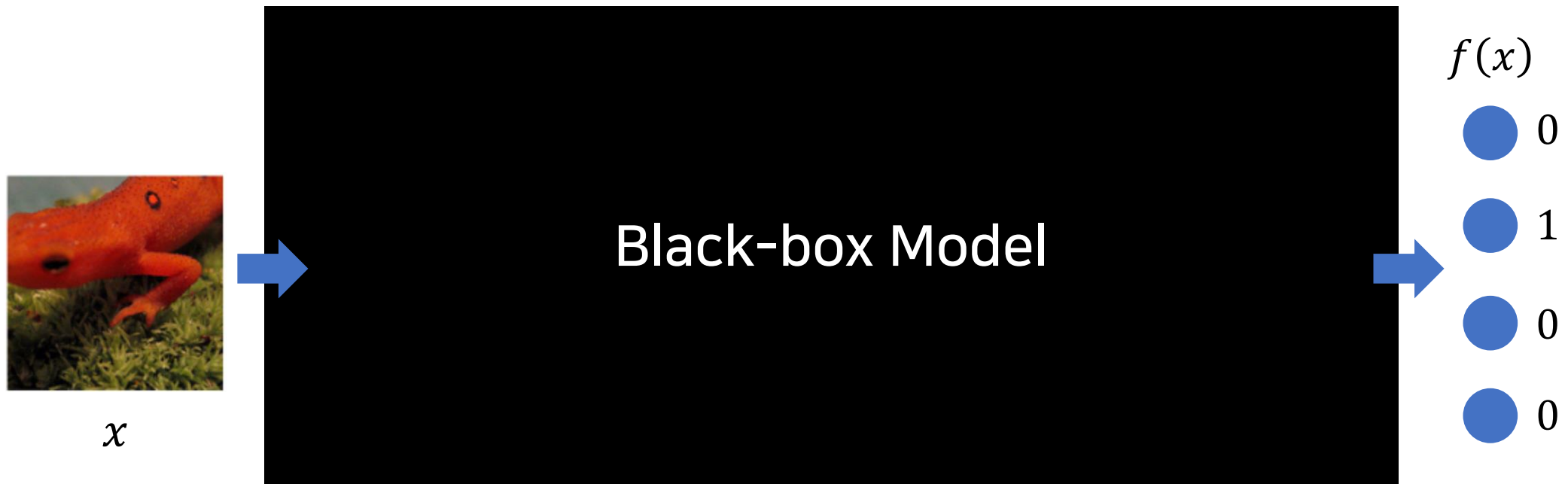
[배경 지식] Threat Model: White-box Setting

- Model information including network structure and weights is revealed to the attacker.
 - The gradient of input can be computed by back-propagation.
 - Attacker minimizes the loss function by gradient descent.



[배경 지식] Threat Model: Hard-label Black-box Setting

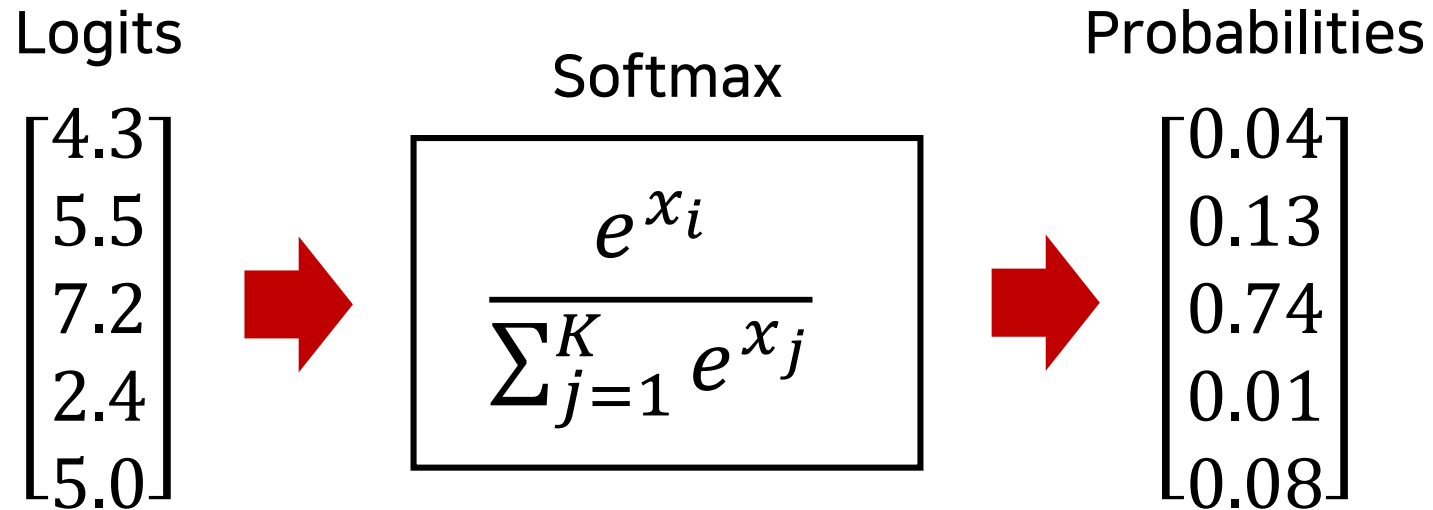
- The model is not known to the attacker.
 - The attacker can make a query and observe a hard-label multi-class output.
 - The attacker is not able to compute the gradient of input x by back-propagation.



화이트박스(White-box) 공격

소프트맥스 함수(Softmax Function)

- 다중 클래스 분류 모델에서 일반적으로 마지막 레이어에 사용되는 함수입니다.
 - Logits 레이어 $Z(x)$ 이후에 소프트맥스(Softmax)를 취합니다.
- 소프트맥스(Softmax)를 사용할 때 각 클래스에 대한 모델의 확률 값을 모두 합하면 1이 됩니다.



크로스 엔트로피(Cross-entropy) 비용 함수

- 마지막 레이어에서 소프트맥스(Softmax)를 사용하는 분류 문제에서 일반적으로 사용하는 비용 함수입니다.

S

L

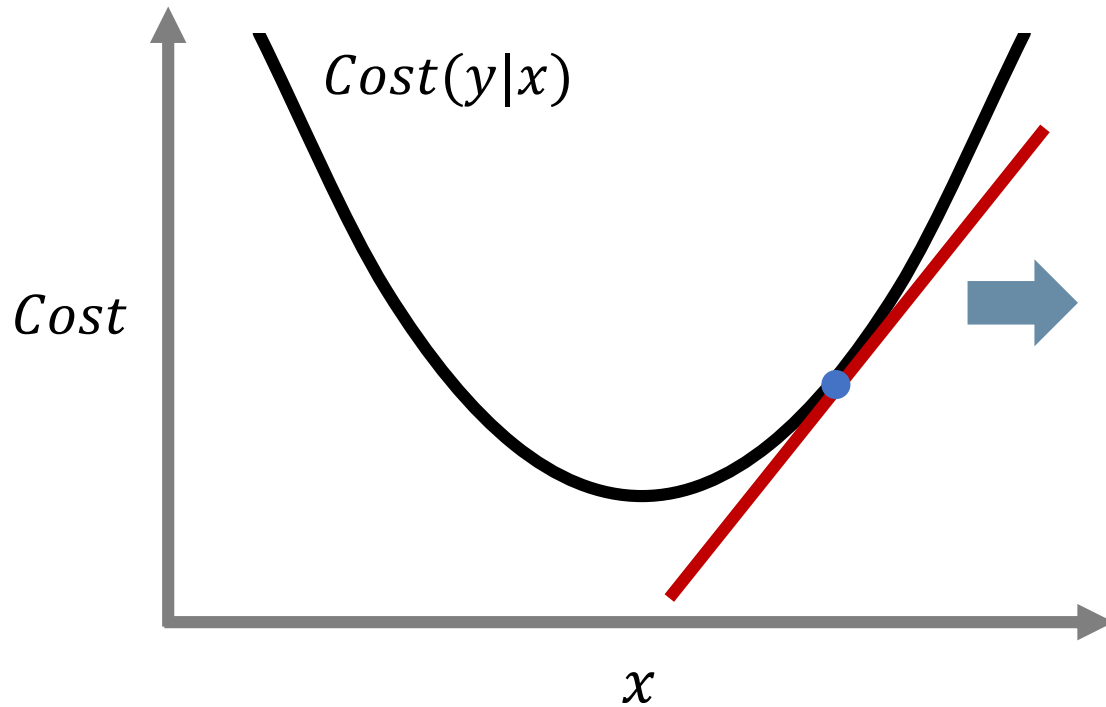
$$CrossEntropy(S, L) = - \sum_i L_i \log(S_i)$$

Diagram illustrating the Cross-Entropy loss function calculation for a specific example:

- Input vector S (Predicted probabilities): $\begin{bmatrix} 0.04 \\ 0.13 \\ 0.74 \\ 0.01 \\ 0.08 \end{bmatrix}$
- Target vector L (Ground truth labels): $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$
- The formula shows the loss is calculated as $- \sum_i L_i \log(S_i)$.
- Arrows indicate the calculation for the third element ($i=3$), where $L_3 = 1$ and $S_3 = 0.74$.

경사 하강(Gradient Descent)을 활용한 적대적 예제 생성

- 경사 하강은 가중치뿐 아니라 입력 데이터에 대해서도 수행할 수 있습니다.
- 비용(손실)을 늘리는 방향으로 입력 데이터를 조금씩 **업데이트**하면 어떻게 될까요?



“현재 기울기가 양수(+)구나?
입력값을 양수(+) 방향으로 이동시키자!”

FGSM (Fast Gradient Sign Method)

- 고차원 공간에서의 선형적인 행동(linear behavior)은 적대적 예제를 만들기에 충분합니다.
- 입력에 대한 비용 함수의 기울기(gradient)를 계산해 한 번 업데이트(single-step)를 수행합니다.
 - 각 입력 뉴런(픽셀)에 대하여 비용이 증가하는 방향으로 입실론(ϵ)만큼 업데이트합니다.



x

(Tabby Cat)

$+ \epsilon *$



$\text{sign}(\nabla_x J(x, y))$

$=$



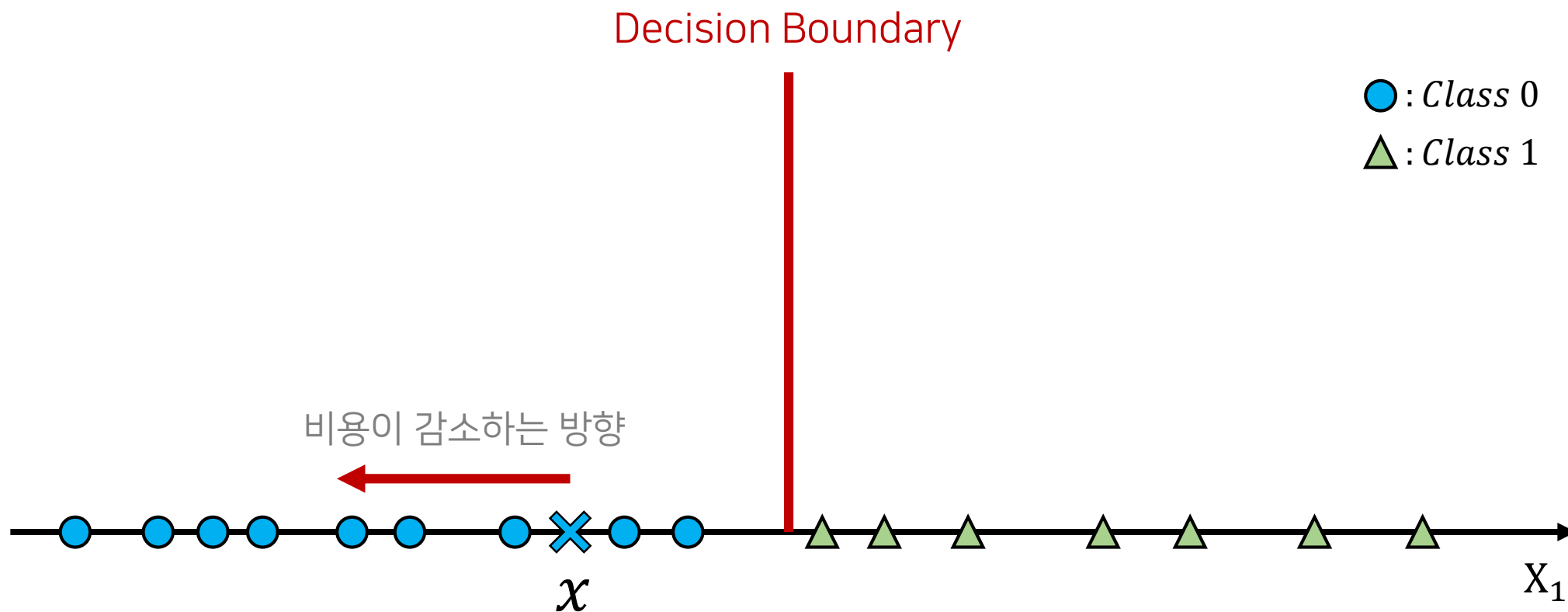
x^*

(Guacamole)

*Explaining and Harnessing Adversarial Examples (ICLR 2015)

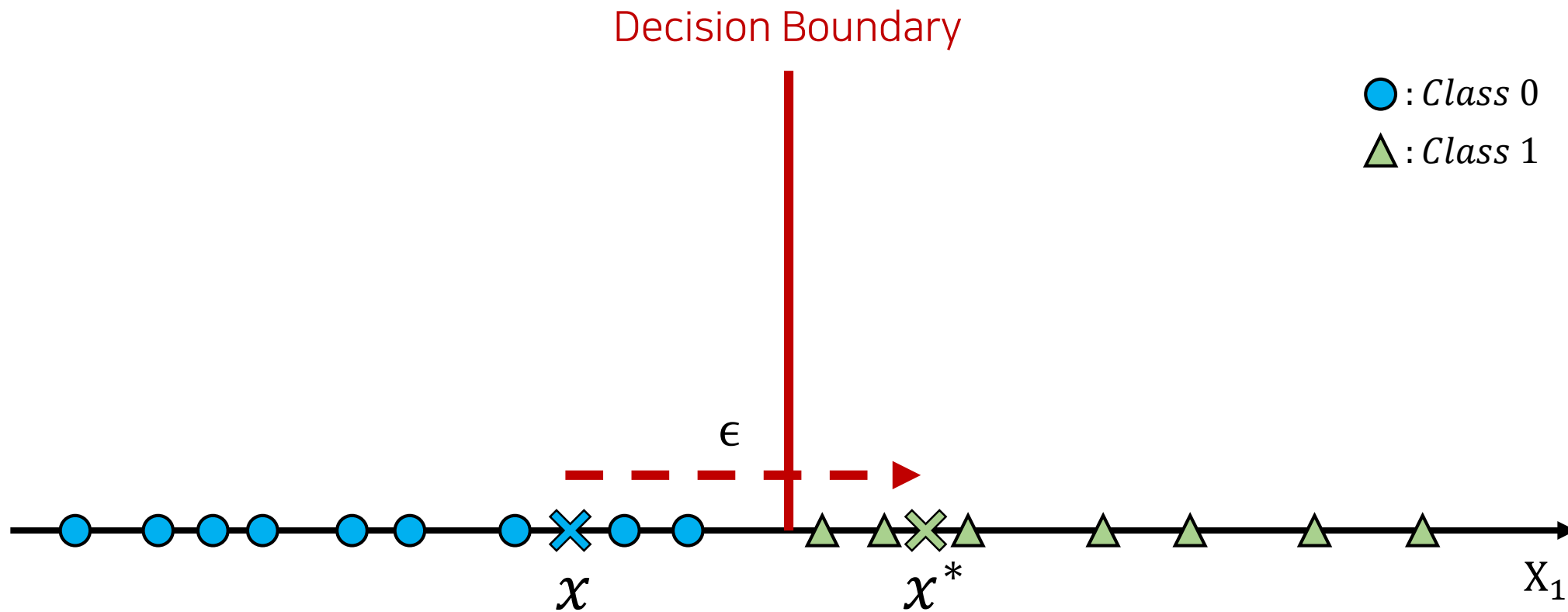
FGSM (Fast Gradient Sign Method) 쉽게 이해하기

- 1차원 데이터에 대하여 적대적 예제를 만드는 경우를 이해해 봅시다.



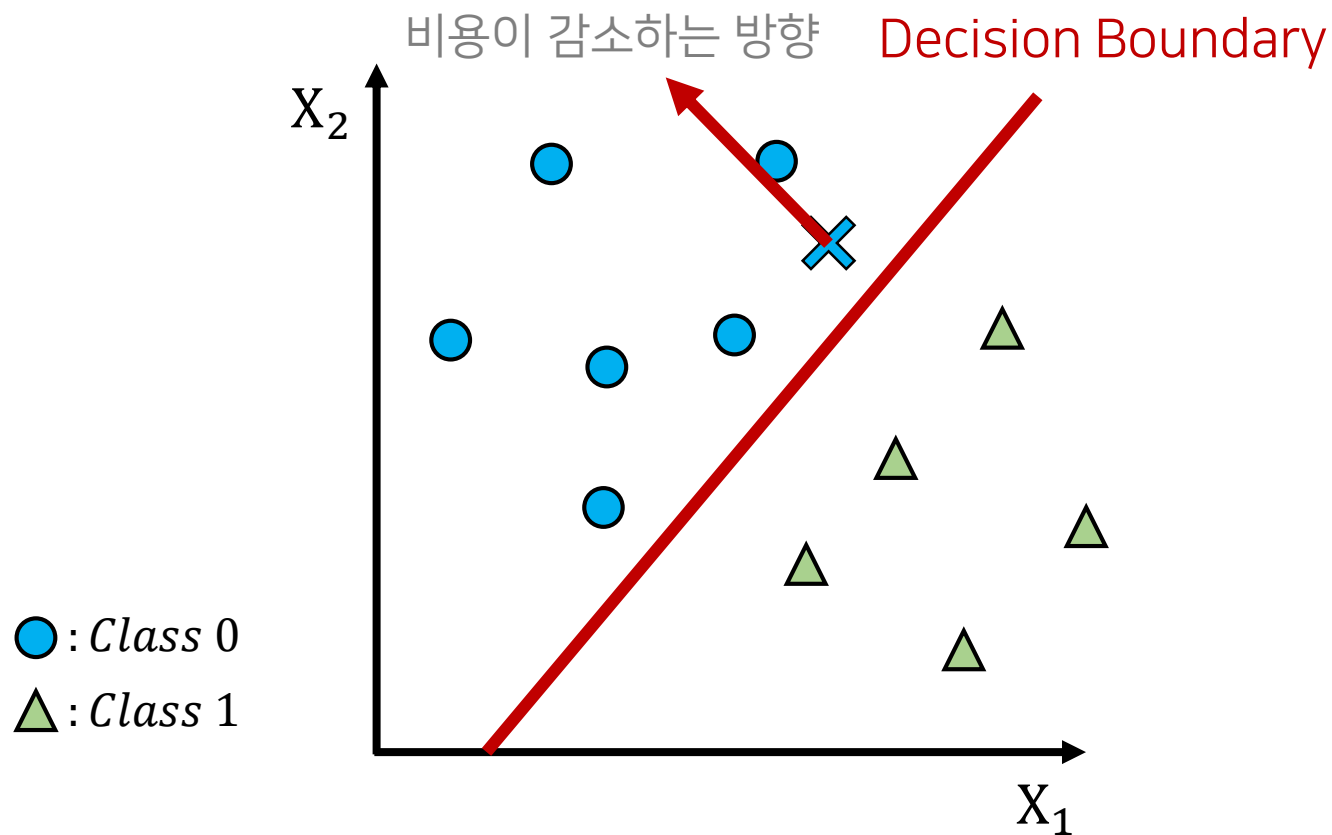
FGSM (Fast Gradient Sign Method) 쉽게 이해하기

- 1차원 데이터에 대하여 적대적 예제를 만드는 경우를 이해해 봅시다.



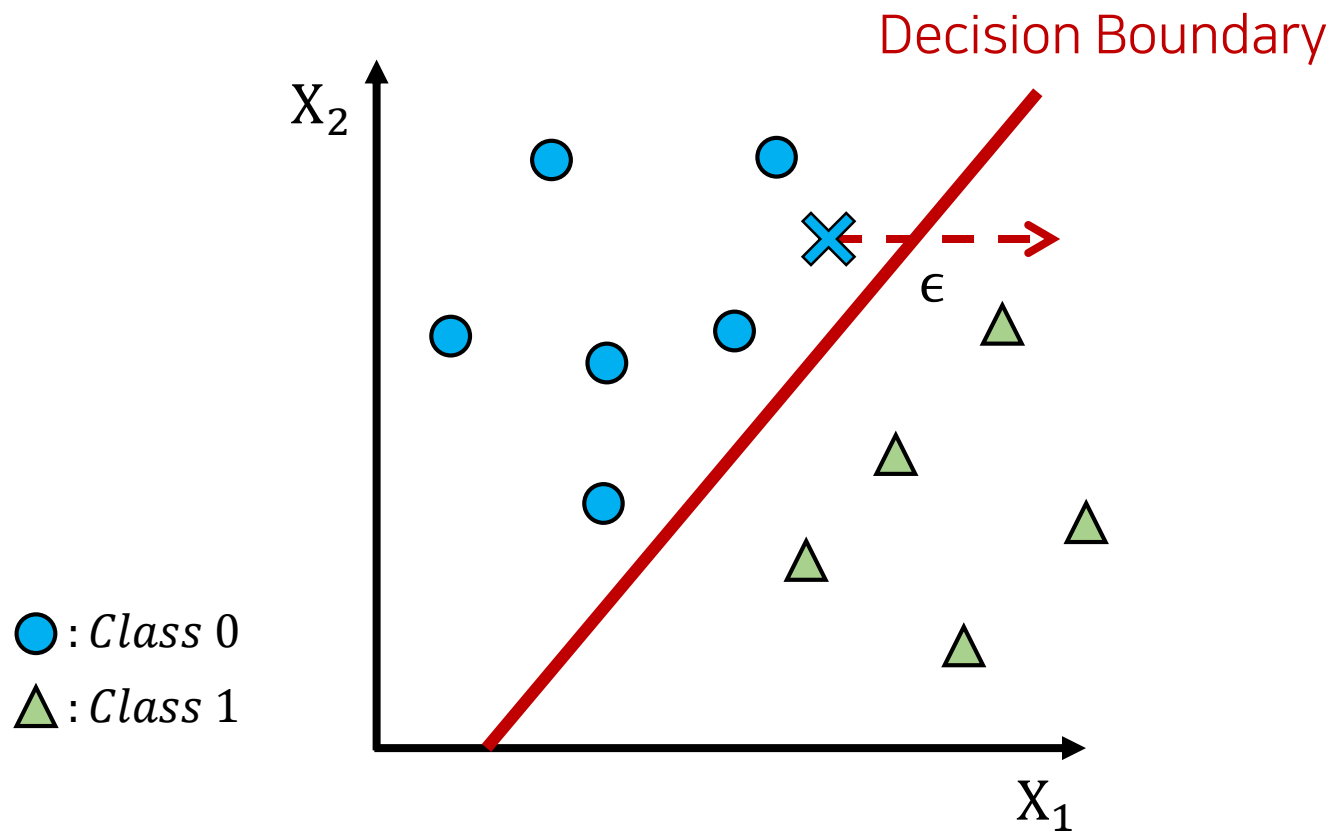
FGSM (Fast Gradient Sign Method) 쉽게 이해하기

- 2차원 데이터에 대하여 적대적 예제를 만드는 경우를 이해해 봅시다.



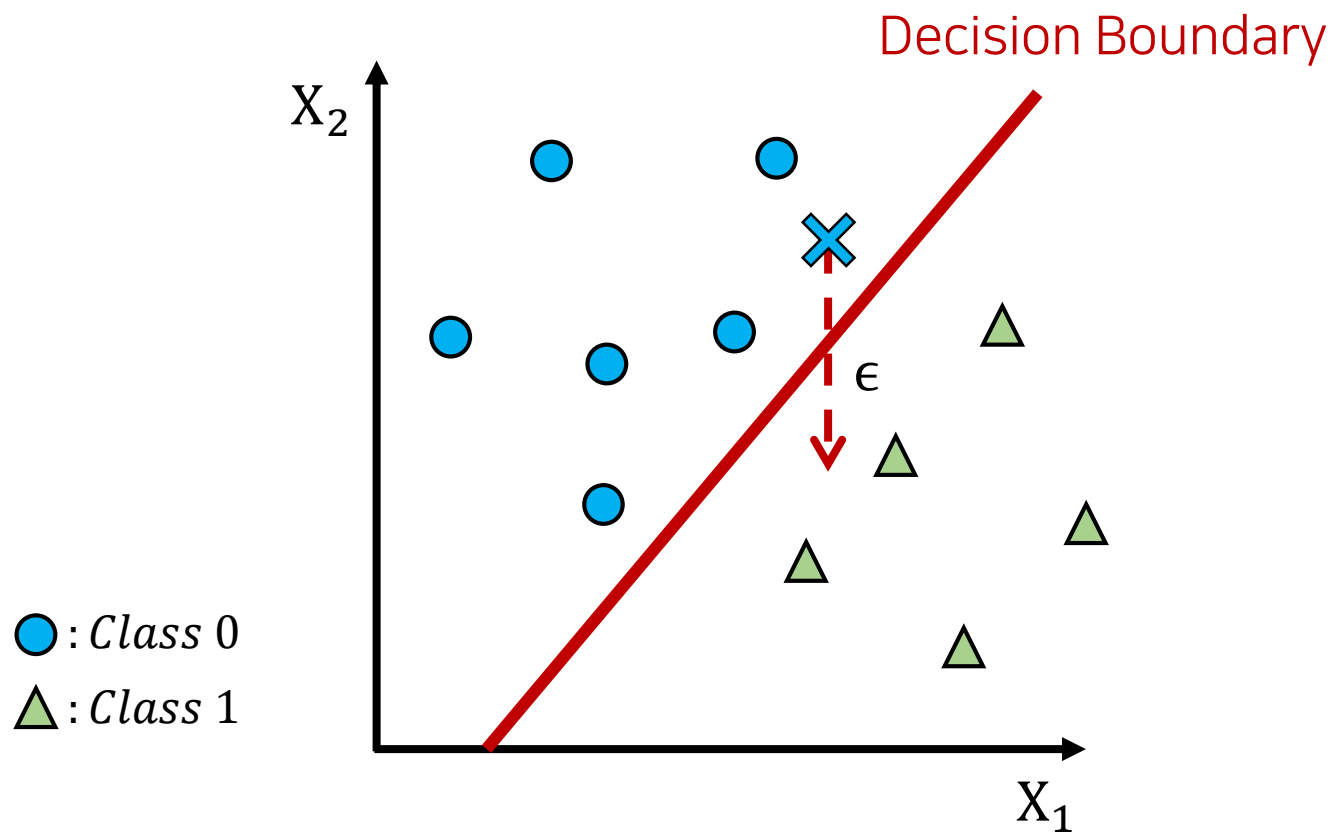
FGSM (Fast Gradient Sign Method) 쉽게 이해하기

- X_1 축에 대하여 비용이 감소하는 방향이 음수(-)이므로 양수 방향으로 ϵ 만큼 업데이트합니다.



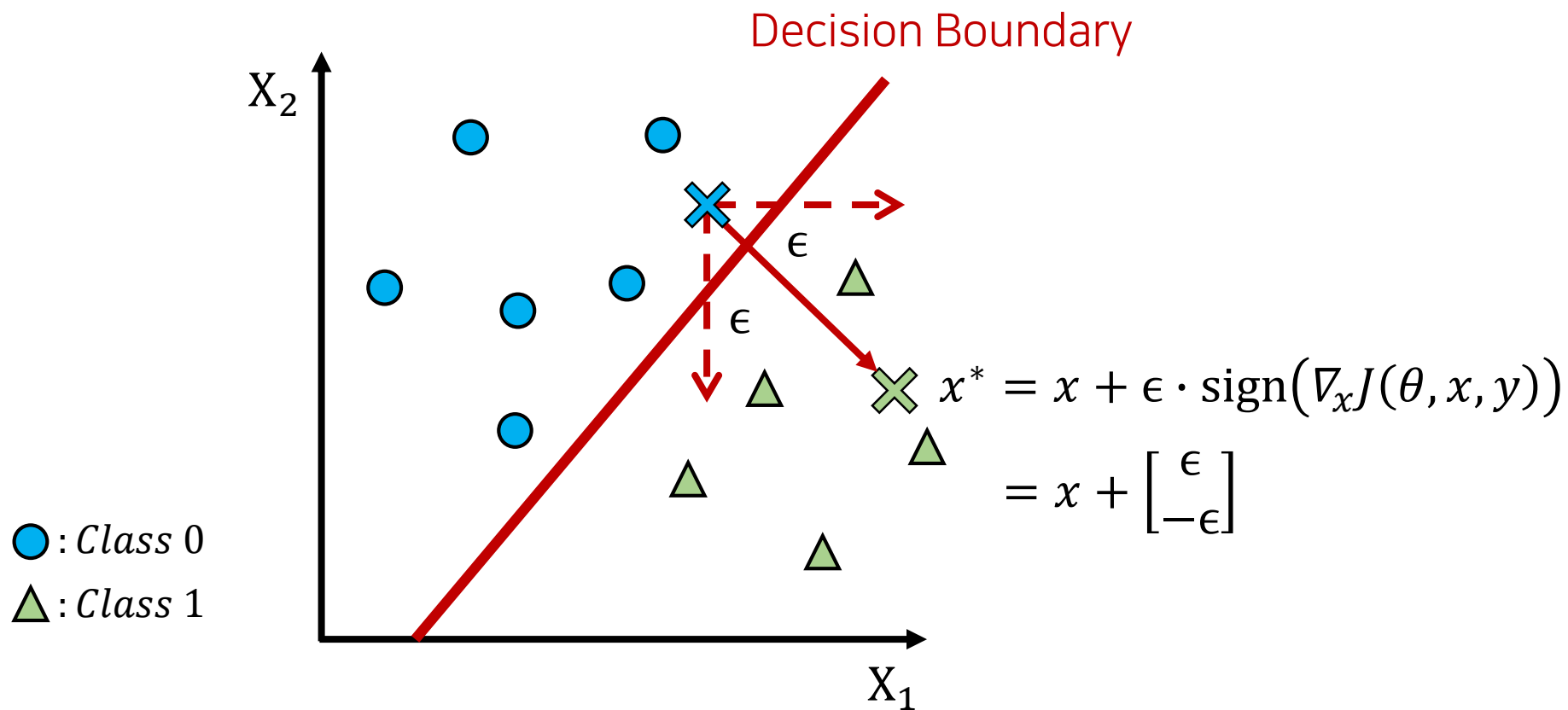
FGSM (Fast Gradient Sign Method) 쉽게 이해하기

- X_2 축에 대하여 비용이 감소하는 방향이 양수(+)이므로 음수 방향으로 ϵ 만큼 업데이트합니다.



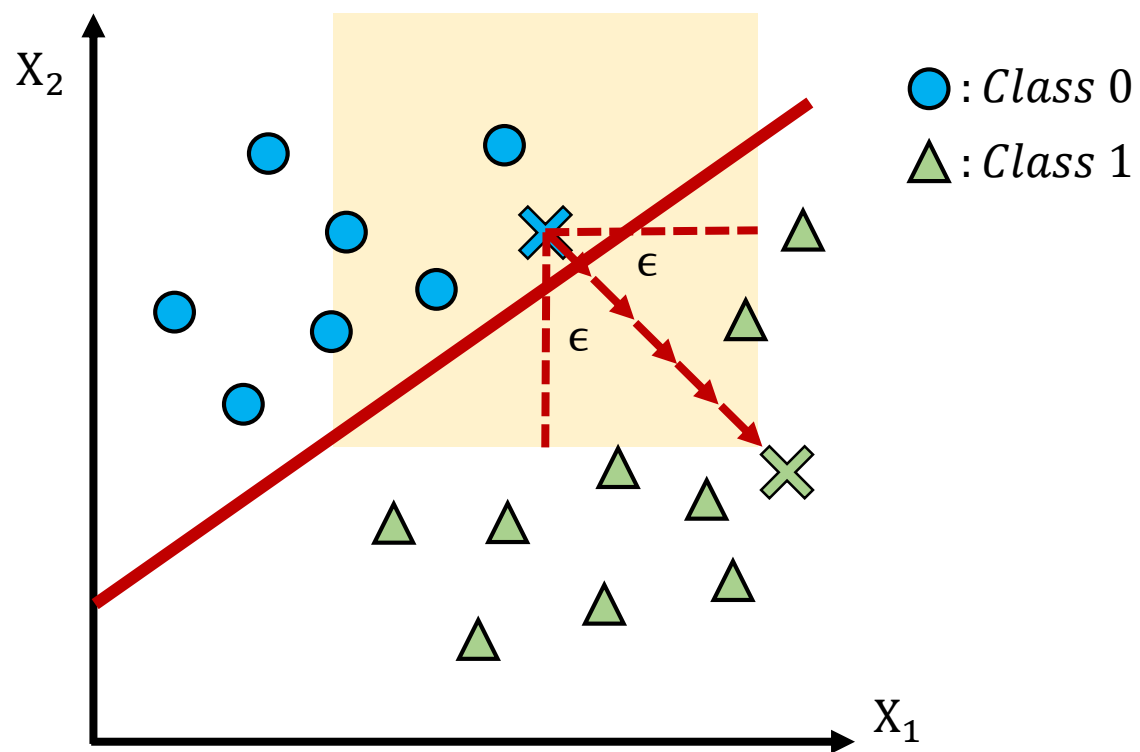
FGSM (Fast Gradient Sign Method) 쉽게 이해하기

- 최종적인 적대적 예제는 다음과 같습니다.



더 강력한 공격: Projected Gradient Descent (PGD)

- PGD attack: $x^{t+1} = \Pi_{x+S} (x^t + \alpha * \text{sign}(\nabla_x L(x, y)))$
- $L_{infinity}$ 공간의 PGD는 Iterative FGSM (I-FGSM)과 유사하며 랜덤 노이즈에서 시작합니다.



θ : the parameters of a model

x : the input to the model

y : the targets associated with x

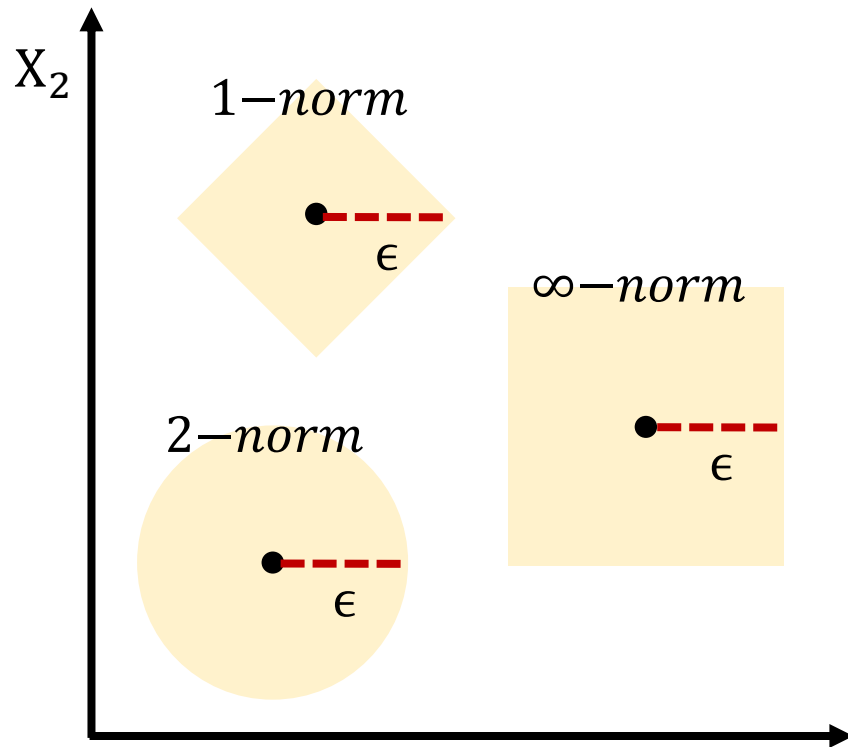
$J(\theta, x, y)$: the cost used to train the neural network

Constraint of perturbation: $\|\delta\|_{\infty} = \max_i |\delta_i| \leq \epsilon$

*Towards Deep Learning Models Resistant to Adversarial Attacks (ICLR 2018)

Metric: p-norm

- 특정한 벡터의 크기(size)를 판단하는 기준으로 사용할 수 있습니다.
- 노이즈(perturbation)의 크기를 p-norm을 이용하여 제한할 수 있습니다.
 - p-norm이란? $\|x\|_p = (|x_1|^p + |x_2|^p + \dots + |x_n|^p)^{1/p}$



$$\|x\|_0 = |x_1|^0 + |x_2|^0 + \dots + |x_n|^0 \leq \epsilon$$

$$\|x\|_1 = |x_1| + |x_2| + \dots + |x_n| \leq \epsilon$$

$$\|x\|_2 = (x_1^2 + x_2^2 + \dots + x_n^2)^{1/2} \leq \epsilon$$

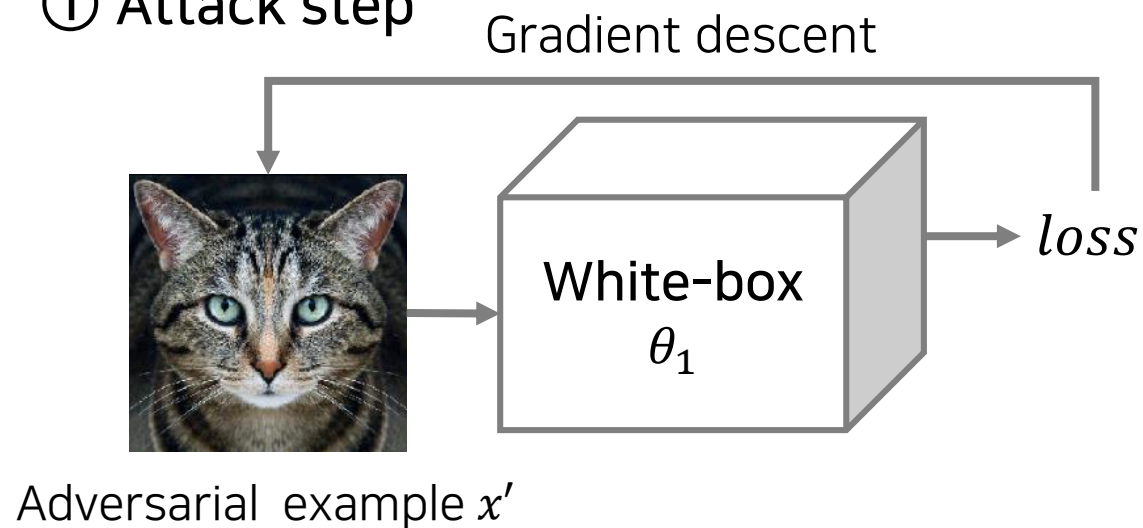
$$\|x\|_\infty = \max\{|x_1|, |x_2|, \dots, |x_n|\} \leq \epsilon$$

블랙박스(Black-box) 공격

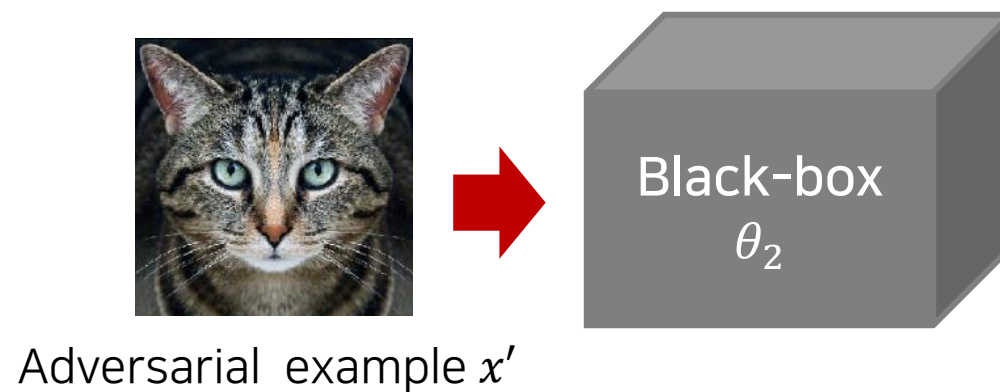
(배경 지식) Transfer-based attack

- Adversarial example은 모델 사이에서 전송 가능한(transferable) 특성이 있습니다.
- 이러한 transferability를 이용한 공격 방법은 다음과 같습니다.
 1. 공격자가 개인적으로 공격 대상 모델(black-box)과 유사한 대체 모델(substitute model)을 학습합니다.
 2. 자신의 대체 모델에 대하여 white-box 공격을 수행해 adversarial example을 생성합니다.
 3. 해당 adversarial example을 공격 대상인 black-box 모델에 넣어 최종적으로 공격을 수행합니다.

① Attack step



② Transfer step



(배경 지식) Transfer-based attack

- Transfer-based attack이 가능한 이유는 무엇일까요?
 - 유사한 학습 데이터 세트(training dataset)로 학습한 모델은 유사한 decision boundary를 가집니다.
 - 따라서 특정한 모델 θ_1 에 대한 adversarial example은 θ_2 에 대해서도 adversarial할 수 있습니다.
- 현실적인 공격 기법 (ASIA CCS 2017)
 - 추가적인 쿼리(query)를 날려 black-box 모델과 더욱 유사한 surrogate 모델을 만들어 공격합니다.
- 다른 관점에서의 분석 (NIPS 2019)
 - Adversarial perturbation을 non-robust feature로 이해할 수 있습니다.
 - 모델들이 유사한 generalized non-robust feature를 학습하므로 transferability가 존재합니다.
- 대표적인 방어 기법 (ICLR 2018)
 - Ensemble adversarial training은 transfer-based attack에 대하여 높은 방어율을 보입니다.

*Practical Black-Box Attacks against Machine Learning (ACM CCS 2017)

*Adversarial Examples Are Not Bugs, They Are Features (NIPS 2019)

Boosting Adversarial Attacks with Momentum (CVPR 2018)

본 논문에서 제안한 메서드: MI-FGSM (Momentum Iterative Fast Gradient Sign Method)

- Non-targeted 공격을 위한 목적 함수

$$\arg \max_{\mathbf{x}^*} J(\mathbf{x}^*, y), \text{ s.t. } \|\mathbf{x}^* - \mathbf{x}\|_{\infty} \leq \epsilon$$

- g_t 는 처음부터 t 개의 기울기(gradient) 정보를 가지고 있습니다. (Momentum)
 - 이전까지의 기울기 정보를 활용하여 poor local maxima에 빠지지 않도록 합니다.
- 만약 μ 값이 0이라면 일반 I-FGSM과 같습니다.
- 공격이 수행되는 과정에서 기울기 벡터의 크기 (scale)는 다양하게 존재할 수 있으므로 L_1 거리로 정규화(normalization)합니다.

Algorithm 1 MI-FGSM

Input: A classifier f with loss function J ; a real example \mathbf{x} and ground-truth label y ;

Input: The size of perturbation ϵ ; iterations T and decay factor μ .

Output: An adversarial example \mathbf{x}^* with $\|\mathbf{x}^* - \mathbf{x}\|_{\infty} \leq \epsilon$.

1: $\alpha = \epsilon/T$;

2: $\mathbf{g}_0 = 0$; $\mathbf{x}_0^* = \mathbf{x}$;

3: **for** $t = 0$ to $T - 1$ **do**

4: Input \mathbf{x}_t^* to f and obtain the gradient $\nabla_{\mathbf{x}} J(\mathbf{x}_t^*, y)$;

5: Update \mathbf{g}_{t+1} by accumulating the velocity vector in the gradient direction as

$$\mathbf{g}_{t+1} = \mu \cdot \mathbf{g}_t + \frac{\nabla_{\mathbf{x}} J(\mathbf{x}_t^*, y)}{\|\nabla_{\mathbf{x}} J(\mathbf{x}_t^*, y)\|_1}; \quad (6)$$

6: Update \mathbf{x}_{t+1}^* by applying the sign gradient as

$$\mathbf{x}_{t+1}^* = \mathbf{x}_t^* + \alpha \cdot \text{sign}(\mathbf{g}_{t+1}); \quad (7)$$

7: **end for**

8: **return** $\mathbf{x}^* = \mathbf{x}_T^*$.

Momentum 적용

Iterative 공격 수행

본 논문에서 제안한 메서드: MI-FGSM for Ensemble of Models

- Ensemble in logits 메서드를 제안합니다.
 - Logits 값의 가중치 합을 계산합니다.
 - 이후에 softmax cross-entropy loss를 이용해 전체 loss value를 계산합니다.

- Non-targeted 공격을 위한 목적 함수

$$\arg \max_{\mathbf{x}^*} J(\mathbf{x}^*, y), \text{ s.t. } \|\mathbf{x}^* - \mathbf{x}\|_{\infty} \leq \epsilon$$

$$\mathbf{l}(\mathbf{x}) = \sum_{k=1}^K w_k \mathbf{l}_k(\mathbf{x})$$

$$J(\mathbf{x}, y) = -\mathbf{1}_y \cdot \log(\text{softmax}(\mathbf{l}(\mathbf{x})))$$

Algorithm 2 MI-FGSM for an ensemble of models

Input: The logits of K classifiers $\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_K$; ensemble weights w_1, w_2, \dots, w_K ; a real example \mathbf{x} and ground-truth label y ;

Input: The size of perturbation ϵ ; iterations T and decay factor μ .

Output: An adversarial example \mathbf{x}^* with $\|\mathbf{x}^* - \mathbf{x}\|_{\infty} \leq \epsilon$.

- 1: $\alpha = \epsilon/T$;
- 2: $\mathbf{g}_0 = 0$; $\mathbf{x}_0^* = \mathbf{x}$;
- 3: **for** $t = 0$ to $T - 1$ **do**
- 4: Input \mathbf{x}_t^* and output $\mathbf{l}_k(\mathbf{x}_t^*)$ for $k = 1, 2, \dots, K$;
- 5: Fuse the logits as $\mathbf{l}(\mathbf{x}_t^*) = \sum_{k=1}^K w_k \mathbf{l}_k(\mathbf{x}_t^*)$;
- 6: Get softmax cross-entropy loss $J(\mathbf{x}_t^*, y)$ based on $\mathbf{l}(\mathbf{x}_t^*)$ and Eq. (9);
- 7: Obtain the gradient $\nabla_{\mathbf{x}} J(\mathbf{x}_t^*, y)$;
- 8: Update \mathbf{g}_{t+1} by Eq. (6);
- 9: Update \mathbf{x}_{t+1}^* by Eq. (7);
- 10: **end for**
- 11: **return** $\mathbf{x}^* = \mathbf{x}_T^*$.

Logits 가중치 합 ←

[실험 결과] Non-targeted Adversarial Attacks: Attacking a Single Model

- White-box 상황에서 하나의 모델에 대하여 100%에 가까운 공격 성공률을 보입니다.
- Black-box 상황에서 하나의 모델에 대하여 좋은 공격 성공률을 보입니다.
- Black-box 상황에서 ensemble adversarial training에 대해서는 낮은 공격 성공률을 보입니다.

$$\mu = 1, \epsilon = 16/255$$

Number of iterations = 10

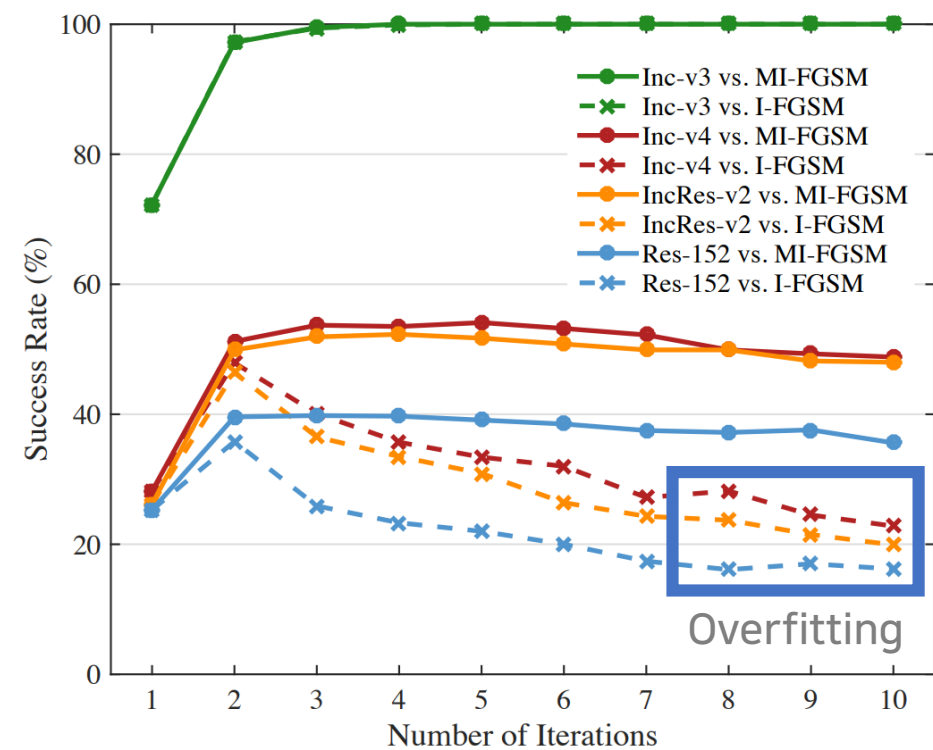
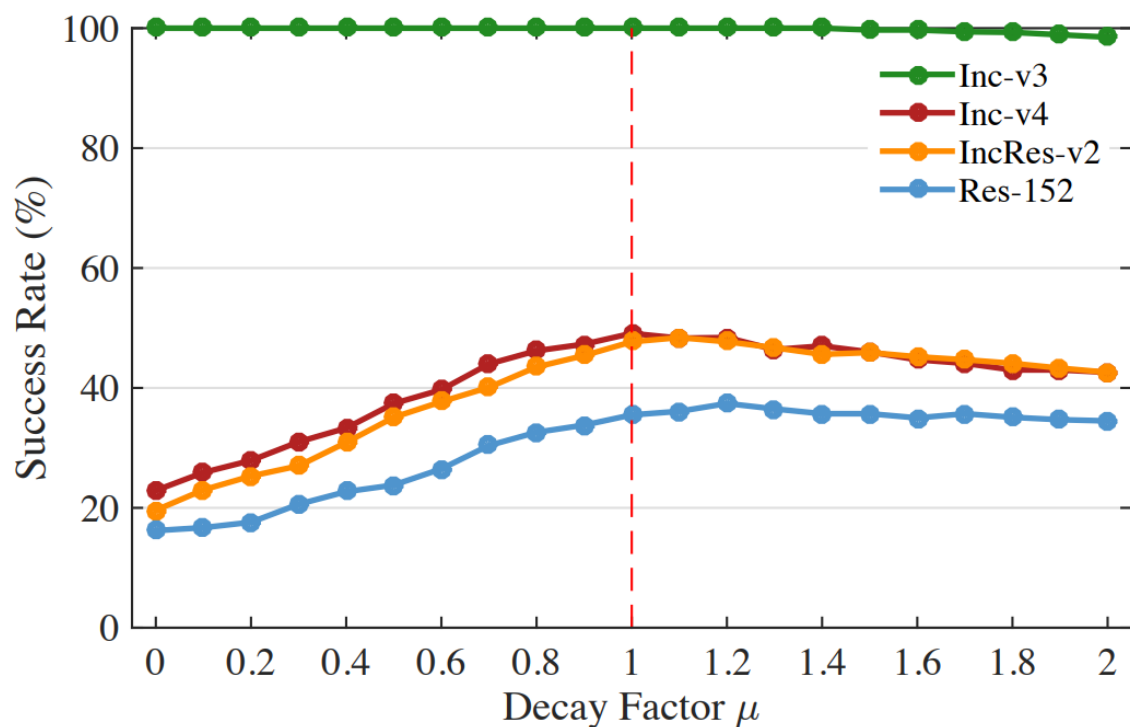
Perturbation 제작 목적의 모델 (1개)		White-box 공격 성공률				Ensemble adversarial training 모델		
	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3 _{ens3}	Inc-v3 _{ens4}	IncRes-v2 _{ens}
Inc-v3	FGSM	72.3*	28.2	26.2	25.3	11.3	10.9	4.8
	I-FGSM	100.0*	22.8	19.9	16.2	7.5	6.4	4.1
	MI-FGSM	100.0*	48.8	48.0	35.6	15.1	15.2	7.8
Inc-v4	FGSM	32.7	61.0*	26.6	27.2	13.7	11.9	6.2
	I-FGSM	35.8	99.9*	24.7	19.3	7.8	6.8	4.9
	MI-FGSM	65.6	99.9*	54.9	46.3	19.8	17.4	9.6
IncRes-v2	FGSM	32.6	28.1	55.3*	25.8	13.1	12.1	7.5
	I-FGSM	37.8	20.8	99.6*	22.8	8.9	7.8	5.8
	MI-FGSM	69.8	62.1	99.5*	50.6	26.1	20.9	15.7
Res-152	FGSM	35.0	28.2	27.5	72.9*	14.6	13.2	7.5
	I-FGSM	26.7	22.7	21.2	98.6*	9.3	8.9	6.2
	MI-FGSM	53.6	48.9	44.7	98.5*	22.1	21.7	12.9

[실험 결과] Non-targeted Adversarial Attacks: Attacking a Single Model

- MI-FGSM의 decay factor μ 를 1로 설정할 때 경험적으로 우수한 공격력을 보입니다.
- MI-FGSM의 경우 많은 수의 반복(iteration)을 거쳐도 높은 공격 성공률을 보입니다.
- 아래 두 그래프는 Inc-v3에 대한 white-box 설정으로 공격을 수행한 결과입니다.

$$\mu = 1, \epsilon = 16/255$$

Number of iterations = 10

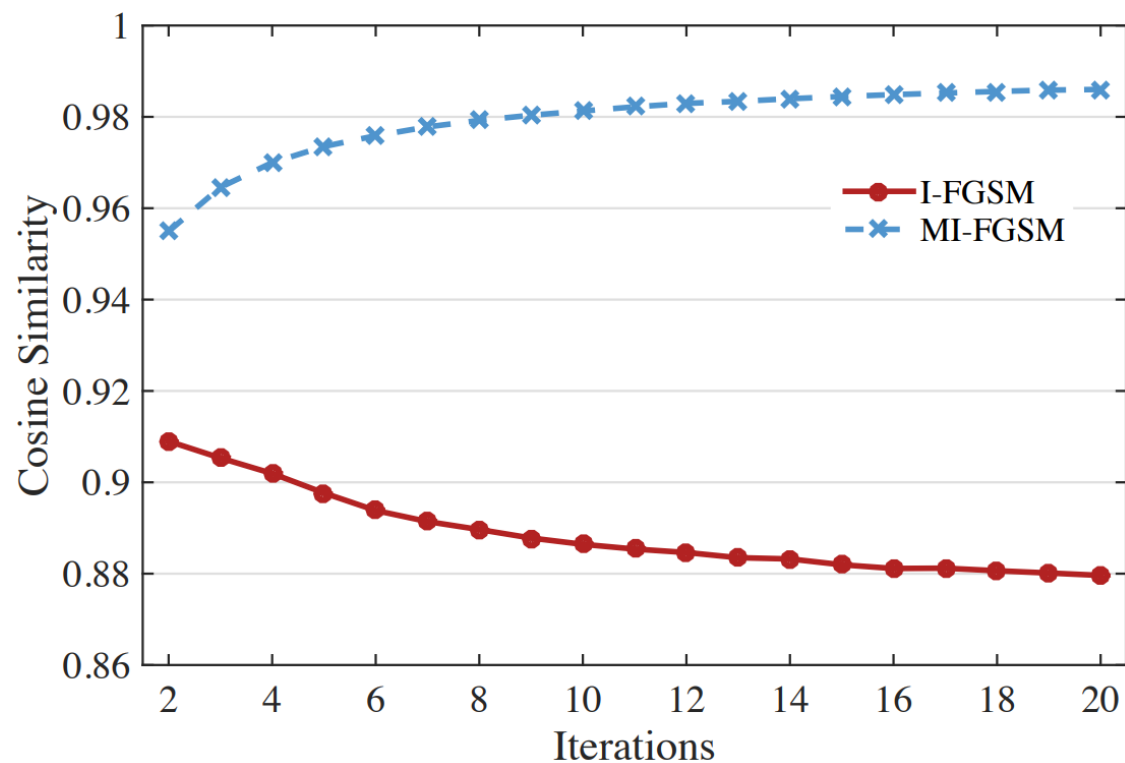


[핵심 내용] Momentum Iterative Fast Gradient Sign Method의 장점

- 기본적인 FGSM은 공격 대상 모델에 대하여 under-fitting되는 특징이 있습니다.
 - 어느 정도의 transferability를 보이지만, white-box 상황에서 충분히 강력하지 못합니다.
- 반면에 I-FGSM (Iterative FGSM)은 과하게 over-fitting되며 poor local maxima에 빠질 수 있습니다.
 - 오히려 일반적인 FGSM보다 transfer-based attack에서 좋은 성능을 내지 못합니다.
- 본 논문에서 제안한 Momentum을 활용한 **MI-FGSM**은 poor local maxima에 빠지지 않는 경향이 있습니다.
 - 결과적으로 좋은 transferability를 보이는 장점이 있습니다.
- **MI-FGSM**은 white-box 공격과 black-box 공격에서 모두 우수한 성능을 보입니다.
 - White-box 상황에서 I-FGSM만큼 강력합니다.
 - Black-box 상황(transfer-based attack)에서 FGSM보다 훨씬 강력합니다.

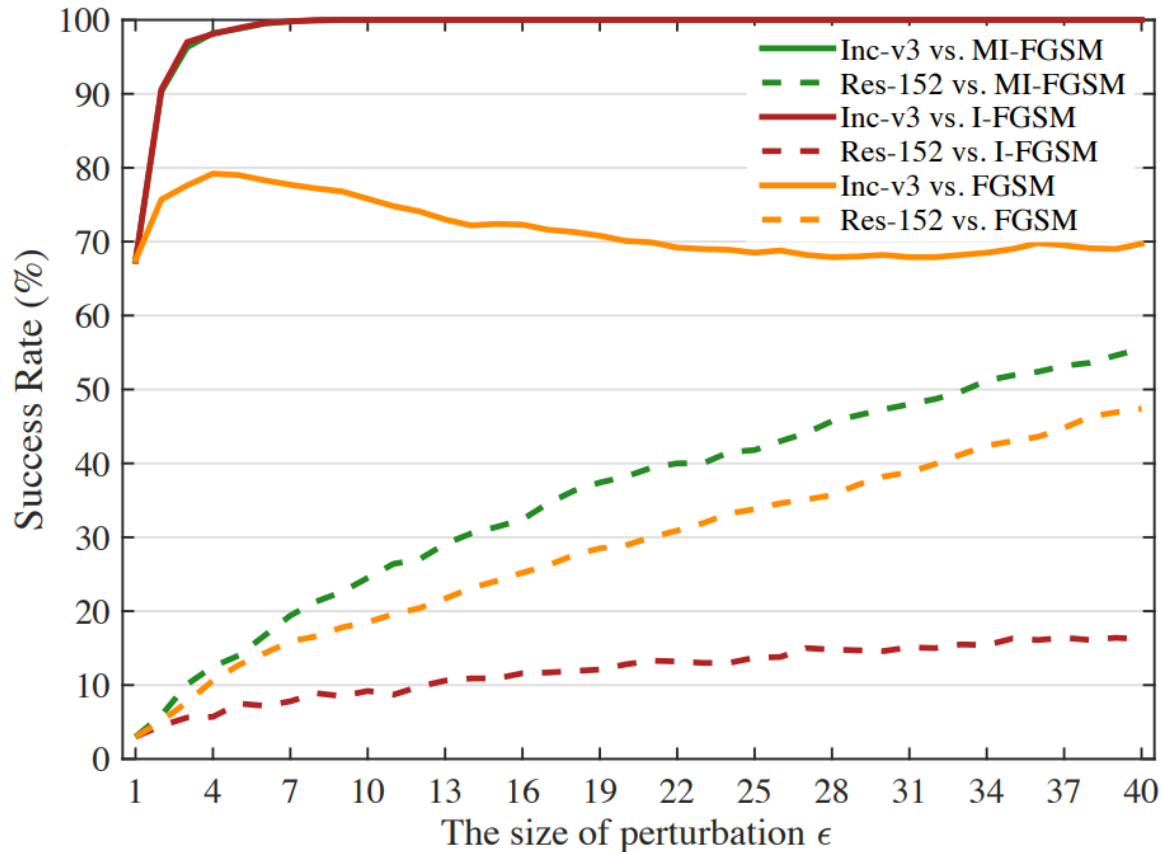
[핵심 내용] Momentum Iterative Fast Gradient Sign Method의 장점

- MI-FGSM을 이용할 때 만들어지는 perturbation들의 코사인 유사도(cosine similarity)가 높습니다.
- I-FGSM의 경우 해당 모델에 over-fitting되어 exceptional decision region에 빠질 확률이 높습니다.
- MI-FGSM은 안정적인 업데이트를 보이며, 쉽게 exceptional decision region에 빠지지 않습니다.
 - 결과적으로 높은 transferability를 보입니다.



[실험 결과] Non-targeted Adversarial Attacks: Attacking a Single Model

- Adversarial perturbation 크기에 따른 공격 성공률은 다음과 같습니다.
- 아래의 그래프는 Inc-v3에 대하여 white-box 설정으로 공격을 수행한 결과입니다.



■ : MI-FGSM 공격 수행 결과

- MI-FGSM은 white-box와 black-box 상황에서 모두 좋은 공격 성공률을 보입니다.
- FGSM은 transferability가 높지만, MI-FGSM 보다는 낮습니다. 특히 White-box 공격 성능은 MI-FGSM보다 현격히 낮습니다.

[실험 결과] Non-targeted Adversarial Attacks: Attacking an Ensemble of Models

- White-box 상황에서 하나의 모델에 대하여 100%에 가까운 공격 성공률을 보입니다. (만들 때는 ensemble 이용)
- Black-box 상황에서 하나의 모델에 대하여 좋은 공격 성공률을 보입니다. (만들 때는 ensemble 이용)

공격 대상 → 모든 모델을 이용한 white-box 공격 나머지 3개의 ensemble을 이용한 black-box 공격

	Ensemble method	FGSM		I-FGSM		MI-FGSM	
		Ensemble	Hold-out	Ensemble	Hold-out	Ensemble	Hold-out
-Inc-v3	Logits	55.7	45.7	99.7	72.1	99.6	87.9
	Predictions	52.3	42.7	95.1	62.7	97.1	83.3
	Loss	50.5	42.2	93.8	63.1	97.0	81.9
-Inc-v4	Logits	56.1	39.9	99.8	61.0	99.5	81.2
	Predictions	50.9	36.5	95.5	52.4	97.1	77.4
	Loss	49.3	36.2	93.9	50.2	96.1	72.5
-IncRes-v2	Logits	57.2	38.8	99.5	54.4	99.5	76.5
	Predictions	52.1	35.8	97.1	46.9	98.0	73.9
	Loss	50.7	35.2	96.2	45.9	97.4	70.8
-Res-152	Logits	53.5	35.9	99.6	43.5	99.6	69.6
	Predictions	51.9	34.6	99.9	41.0	99.8	67.0
	Loss	50.4	34.1	98.2	40.1	98.8	65.2

[실험 결과] Non-targeted Adversarial Attacks: Attacking an Ensemble of Models

- White-box 상황에서 ensemble adversarial training에 대하여 100%에 가까운 공격 성공률을 보입니다.
- Black-box 상황에서 ensemble adversarial training에 대하여 좋은 공격 성공률을 보입니다.
 - (기본 설정) decay factor $\mu = 1.0$, 반복 횟수(iterations) = 20, $\epsilon = 16/255$

모든 모델을 이용한 white-box 공격 ← → 나머지 6개의 ensemble을 이용한 black-box 공격

	Attack	Ensemble	Hold-out
-Inc-v3 _{ens3}	FGSM	36.1	15.4
	I-FGSM	99.6	18.6
	MI-FGSM	99.6	37.6
-Inc-v3 _{ens4}	FGSM	33.0	15.0
	I-FGSM	99.2	18.7
	MI-FGSM	99.3	40.3
-IncRes-v2 _{ens}	FGSM	36.2	6.4
	I-FGSM	99.5	9.9
	MI-FGSM	99.7	23.3

[실제 대회 성적]

본 논문의 저자들은 다수 모델의 ensemble에 대해 공격을 수행한 뒤에 transfer-based attack을 하여 NIPS 2017 당시 adversarial attack 대회에서 우승할 수 있었습니다.

(참고) 원본 Ensemble adversarial training 논문에서는 적은 iteration의 white-box 공격 실험 결과를 보입니다. $\epsilon = 16/255$ 설정에서 강한 white-box 공격에는 방어가 어렵습니다.

*Ensemble Adversarial Training: Attacks and Defenses (ICLR 2018)

본 논문이 제안한 메서드: Extensions

- Momentum iterative method는 다양한 공격 설정(setting)에 대하여 적용할 수 있습니다.
- 예를 들어 targeted 공격을 위한 기본적인 기울기(gradient) 계산 공식은 다음과 같습니다.

$$\mathbf{g}_{t+1} = \mu \cdot \mathbf{g}_t + \frac{J(\mathbf{x}_t^*, y^*)}{\|\nabla_{\mathbf{x}} J(\mathbf{x}_t^*, y^*)\|_1}$$

- Targeted MI-FGSM with an L_∞ norm bound:

$$\mathbf{x}_{t+1}^* = \mathbf{x}_t^* - \alpha \cdot \text{sign}(\mathbf{g}_{t+1})$$

- Targeted MI-FGM with an L_2 norm bound:

$$\mathbf{x}_{t+1}^* = \mathbf{x}_t^* - \alpha \cdot \frac{\mathbf{g}_{t+1}}{\|\mathbf{g}_{t+1}\|_2}$$

Conclusion

- The authors propose **momentum-based iterative methods** to boost adversarial attacks.
 - The proposed method can effectively fool white-box models as well as black-box models.
- The proposed methods outperform one-step gradient-based methods and vanilla iterative methods in a black-box manner.
- To further improve the transferability of the generated adversarial examples, the authors propose to attack **an ensemble of models whose logits are fused together**.