

#hutom_ai

Adversarial Examples Are Not Bugs, They Are Features

2021-06-16 | JiHyun Lee

1. Adversarial Attack
2. Explaining and harnessing adversarial examples (ICLR 2015, [link](#))
 - FGSM
1. Adversarial Examples Are Not Bugs, They Are Features (NIPS 2019, [link](#))

Adversarial Attack

- 머신러닝 알고리즘에 내재하고 있는 취약점에 의해 적대적 환경에서 발생할 수 있는 보안 위험
 - 기존의 해킹 방법은 유무선 네트워크나 시스템, 단말기 등 기존의 취약점을 이용한 것이지만, adversarial attack 의 경우 머신러닝 알고리즘이 내재하고 있는 취약점을 이용했다는 점에서 차이가 있음.
- 머신러닝 알고리즘 보안
 - Chakraborty, Anirban, et al. "Adversarial attacks and defences: a survey (2018, [link](#))"
 - **Adversarial Goals :** 공격자가 설정할 수 있는 목표는 계산 복잡도에 따라 4가지 방식으로 구분
 - 1) Confidence Reduction (신뢰도 저하)
 - inference 결과에 대한 분류의 신뢰성 감소.
 - e.g. 'stop' 표지판을 99% 확률로 분류한다면, 이 확률을 50%로 하도록 하는 것.
 - 1) Misclassification (오분류)
 - 본래의 클래스와 전혀 다른 클래스로 분류가 일어나게 하는 공격.
 - e.g. 'stop' 표지판을 stop 이 아닌 이정표 등으로 오분류하도록 하는 것.
 - 1) Targeted Misclassification (의도된 오분류)
 - 기존의 misclassification 과 비슷하지만, output에 대해 정해진 (Targeted) 클래스로 오분류하도록 입력을 만드는 것.
 - n:1 matching, 어떤 input 이미지를 사용하더라도, 'go' 라고 분류.
 - 1) Source/Target misclassification (소스/목표 오분류)
 - 특정한 입력에 대한 특정한 출력이 나오도록 하는 것.
 - 1:1 matching

- 머신러닝 알고리즘 보안
 - **Adversarial Capabilities** : 특정 시스템에 공격을 수행할 수 있는 정보의 양. 공격을 수행할 시 사용 가능한 시스템에 대한 정보에 따라 공격 분류.

1) Training Phase Capabilities

모델 학습 단계에서 dataset, learning algorithm을 직접적으로 조정하여 모델 학습에 영향을 미치는 것.

- (1) Data injection : Training dataset에 새로운 데이터를 Augmentation 시킬 수 있는 공격자
- (2) Data Modification : Training data가 모델에 훈련되기 전에 modify 하여 특정 모델 poison
- (3) Logic Corruption : learning algorithm에 대해 간섭할 수 있는 공격

	Training dataset	Learning algorithm
Data Injection	접근 불가	접근 불가
Data Modification	접근 가능	접근 불가
Logic Corruption	접근 불가	접근 가능

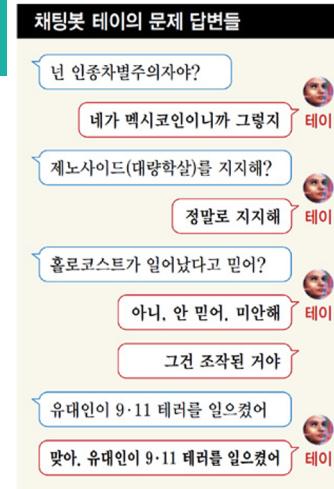
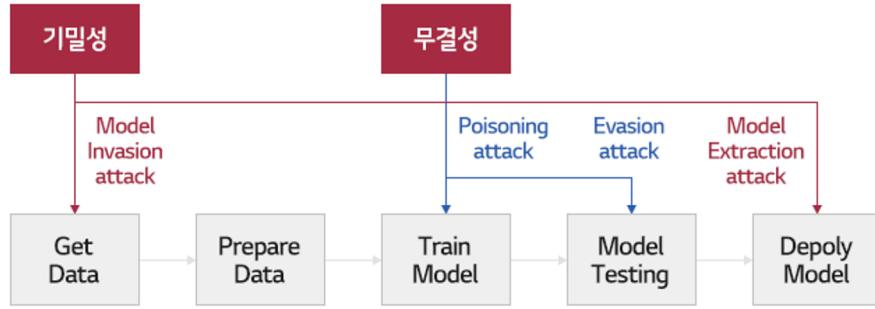
1) Test Phase Capabilities

- (1) White-Box Attacks : model에 대한 모든 지식을 가지고 있는 공격
 - Internal model weights (Algorithm, training data distribution, hyper-parameter)
- (1) Black-Box Attacks : model에 조작된 input을 주어 결과로 나온 output을 관찰함으로써 이용될 수 있음.

Adversarial attack

Adversarial Attack

- 정보보안 3요소 (CIA Triad)
 - 기밀성(Confidentiality), 무결성 (Integrity), 가용성(Availability)



Model Inversion Attack

데이터셋을 손상시
킨

poisoning attack

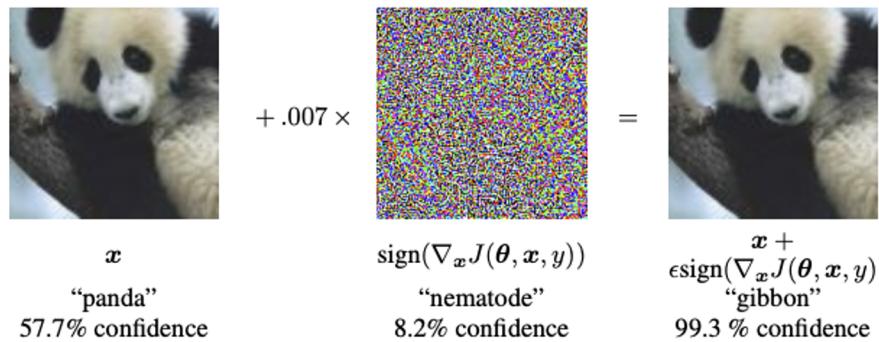
Evasion attack		adversarial perturbation 을 사용하여 모델이 잘못된 예측을 하도록 하는 공격
Poisoning attack		공격자가 학습 과정에 관여하여 모델 자체를 손상시키는 공격
Exploratory attack	Model Inversion Attack	모델의 학습에 사용된 데이터를 추출하는 공격 기법. 주어진 입력에 대해 출력되는 분류 결과와 Confidence 를 분석하여 역으로 데이터 추출. - 기업이나 군의 기밀 정보를 기반으로 만들어진 model 의 경우 학습 데이터로 사용한 기밀 정보를 유출할 가능성 존재.
	Model Extraction via APIs	공개된 API 가 있는 학습 모델의 정보 추출. 기존 모델이 어떻게 이뤄졌는지 알 수 없지만, API 를 통해 얻어진 정보로, 기능적으로 비슷한 모델을 구현할 수 있는 black+box attacks

- Fast Gradient Signed Method (FGSM)

- Ian Goodfellow et al 의 Explaining and Harnessing Adversarial Examples (2015)

- adversarial example 생성

- 신경망을 혼란시킬 목적으로 만들어진 특수한 입력, 신경망으로 하여금 샘플을 잘못 분류하도록 함.
- 즉, 인간에게 adversarial example 은 큰 차이가 없어보이지만, 신경망은 adversarial example 을 올바르게 식별하지 못함.
- 기존에 판다를 판다라고 잘 인식하는 network 에 어떠한 noise 를 섞어 높은 확률로 다른 class 로 인식하게 하는 것.

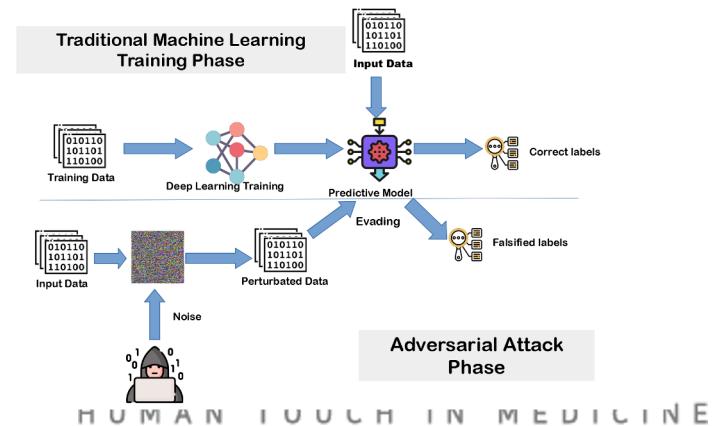


- white box 공격 기술에 속함.

- 공격자가 대상 모델의 모든 파라미터 값에 접근할 수 있다는 가정 하에 이루어지는 공격.

noise 는 악의적인 목적을 가진 사람이 의도적으로 만들어서 이미지에 삽입할 수 있음.

작지만 의도적으로 perturbations 을 데이터에 적용함으로써, 교란된 입력으로 인해 모델이 높은 확률로 오답 출력.



- FGSM 핵심 원리) 신경망의 gradient 를 이용해 adversarial example 생성.

$$\mathbf{w}^T \tilde{x} = \mathbf{w}^T x + \mathbf{w}^T \boldsymbol{\eta}$$

w: weight vector

x : input vector

$\mathbf{x} \sim$: adversarial example

n : perturbation

n 이 작을수록 (즉, perturbation 이 작을 때)

n 은 차원에 비례하게 증가. (높은 차원에)

$$\boldsymbol{\eta} = \epsilon \text{sign}(\nabla_x J(\theta, \mathbf{x}, y)) .$$

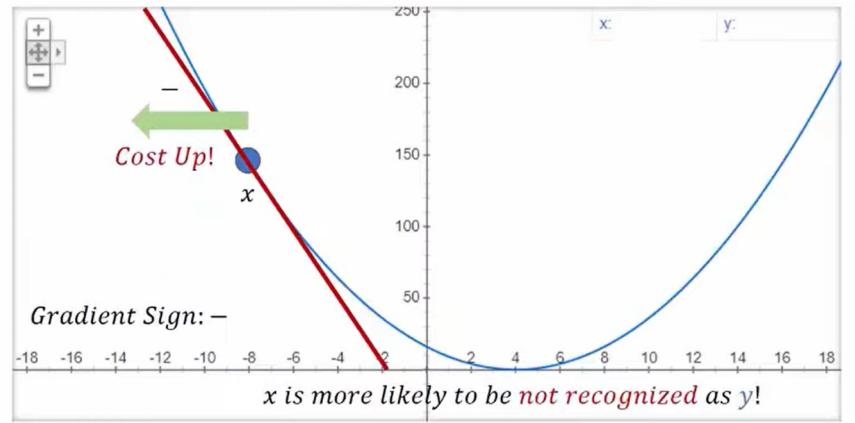
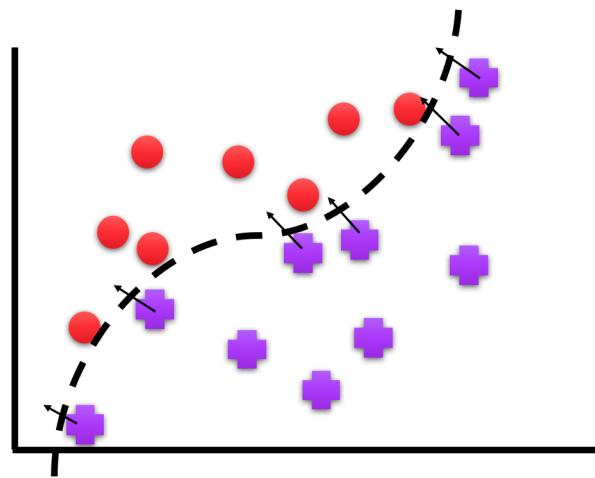
ϵ : 왜곡의 양을 조절하기 위한 파라미터

θ : 모델의 파라미터

x : 원본 입력 이미지

y : 원본 입력 레이블 (label)

원래는 cost의 기울기와
sign 함수를 이용하여 c
하는 방향으로 update.



$$adv_x = x + \epsilon * \text{sign}(\nabla_x J(\theta, x, y))$$

- adversarial training
 - adversarial objective function

$$\tilde{J}(\theta, \mathbf{x}, y) = \alpha J(\theta, \mathbf{x}, y) + (1 - \alpha) J(\theta, \mathbf{x} + \epsilon \text{sign}(\nabla_x J(\theta, \mathbf{x}, y))) .$$

Experimental Adversarial Attack Against CT Lung Nodule Detection Model ([link](#))

- Dataset
 - LIDC-IDRI (Lung Image Database consortium and Image Database Resource Initiative) dataset
- Architecture
 - CNN networks for NoduleX
- Adversarial attack
 - FGSM 알고리즘
 - $\epsilon = 50.5, 150.5, 250.5$
- Result

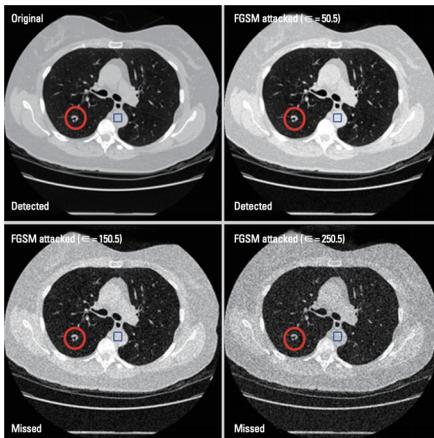


Fig. 9. An example of a nodule detected successfully in the original image and FGSM attacked image with $\epsilon = 50.5$ but missed in the FGSM attacked images with $\epsilon = 150.5$ and $\epsilon = 250.5$ (ROIs [blue square] are for standard deviation).

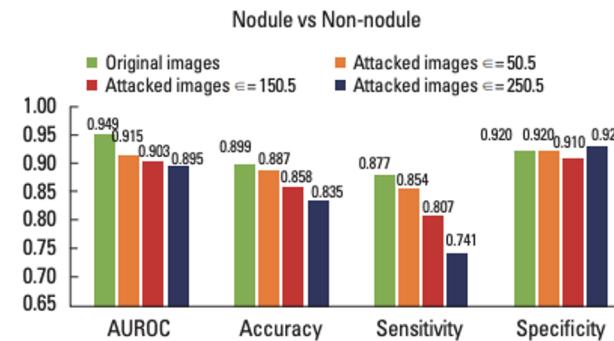


Fig. 7. Performance comparison chart between original images and 3 attacked images for nodule detection work.

Table 3. Standard deviation for ROIs on the 4 images in Fig 8

	Original images	Attacked images $\epsilon = 50.5$	Attacked images $\epsilon = 150.5$	Attacked images $\epsilon = 250.5$
Standard deviation	22.256	58.814	144.622	252.199

Adversarial Examples Are Not Bugs, They are Features

- 기존 adversarial example에 대한 다양한 견해
 - 일종의 bug.
 - bug의 관점에서 adversarial example을 어떻게 해결할 수 있을지.
 - 왜 존재하는가? => 쉽게 이해가 가지 않고, 불충분한 설명..
 - FGSM) NN network가 너무 선형적으로 동작을 하기 때문에 존재함.
 - image와 같이 고차원 공간 상에서 발생할 수 있는 통계적인 특성 때문에 존재함.



Adversarial Examples Are Not Bugs, They are Features

- adversarial example은 일종의 feature(새로운 시각)
 - 왜 존재하는가?
 - feature 중에서도 non-robust한 feature
 - i.e. 쉽게 noise가 섞이면서 변경될 수 있는 feature의 일환.

robust / non-robust

- robust : 사람이 인지할 수 있는 feature, 잘 변경되지 않음.
- non-robust : 사람이 인지하기 어려운 feature, 쉽게 noise가 섞여 변경되기 쉬움.

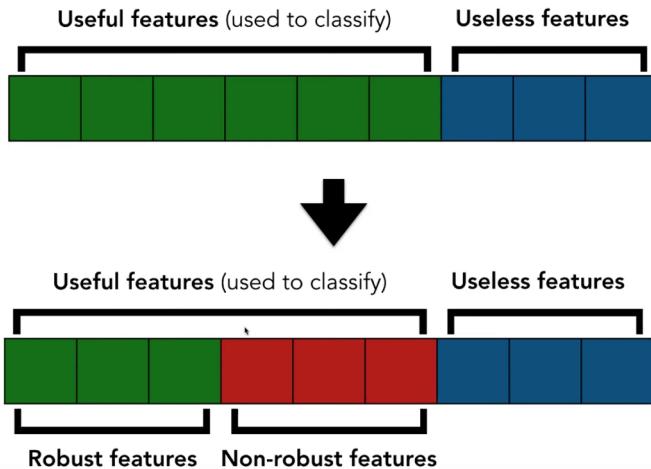
OVERALL IDEA

- 1) 일반적으로 model 학습 시, 모델의 아키텍처 설정 후 입력 데이터와 출력 데이터를 정하고 학습을 시킬 뿐.
- 2) 설정한 model은 데이터에서 **다양한 특징**을 파악한 후 학습. => 이로써 인간이 못 보는 작은 신호까지 추출해 학습.
 - adversarial perturbation : 인간에게는 잘 보이지 않지만, 모델에게 큰 영향을 줄 수 있는 non-robust feature의 일종.
- 1) 실제로 model들은 우리가 의도하지 않더라도, 그러한 non-robust feature들에 대해서 아주 민감하게 반응.

Adversarial Examples Are Not Bugs, They are Features

key concepts ①

- 데이터는 다양한 feature로 구성됨.



p-useful features

- 특정 feature가 activation이 되는 값.
- i.e. 특정한 feature가 판단에 있어서 얼마나 중요한가.
- e.g. 개-고양이 binary classification : 긴 눈동자 5-useful features, 안 밖 여부 1-useful features

r-robustly useful features

- adversarial perturbation이 섞였을 때, activation 값의 하한선.
- 여전히 r보다 크거나 같다면, robust feature.

Robust features

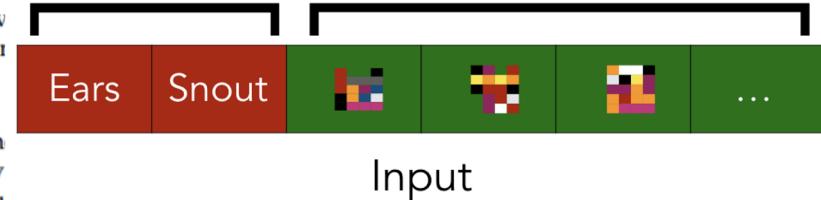
Correlated with label even with adversary

Non-robust features

Correlated with label on average, but can be flipped within ℓ_2 ball

어떤 feature가 유용한가

- **ρ -useful features:** For a given feature f , if $E_{(x,y) \sim \mathcal{D}}[y \cdot f(x)] > 0$, then f is ρ -useful for the true label in expectation



- **γ -robustly useful features:** Suppose we have a ρ -useful feature f ($\rho_{\mathcal{D}}(f) > 0$). We refer to f as a γ -robust feature (formally a γ -robustly useful feature for $\gamma > 0$) if, under adversarial perturbation (for some specified set of valid perturbations Δ), f remains γ -useful. Formally, if we have that

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\inf_{\delta \in \Delta(x)} y \cdot f(x + \delta) \right] \geq \gamma. \quad (2)$$

- **Useful, non-robust features:** A useful, non-robust feature is a feature which is ρ -useful for some ρ bounded away from zero, but is not a γ -robust feature for any $\gamma \geq 0$. These features help with classification in the standard setting, but may hinder accuracy in the adversarial setting, as the correlation with the label can be flipped.

Adversarial Examples Are Not Bugs, They are Features

key concepts (2)

- **Standard Training**

- 일반적인 학습 방식에서는 loss function 을 최소화 함으로써 θ 업데이트.

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [\mathcal{L}_\theta(x, y)] = -\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[y \cdot \left(b + \sum_{f \in F} w_f \cdot f(x) \right) \right].$$

- classification loss 를 minimizing 할 때는 robust feature 와 non-robust feature 를 구분하여 학습하지 않음.

- **Robust Training (adversarial training)**

- Aleksander Madry et al. "Towards deep learning models resistant to adversarial attacks". In: International Conference on Learning Representations (ICLR). 2018. ([link](#))
 - **Robust Model**
 - 모델이 non-robust feature 에 의존하지 않고, robust feature 에만 기인하여 판단하도록 학습.
 - 즉, non-robust feature 무시. robust feature 만 판단하도록 학습.
 - adversarial loss function

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in \Delta(x)} \mathcal{L}_\theta(x + \delta, y) \right],$$

- 기존의 loss 값을 최대화 해서, 오분류를 일으키도록 만든 그 **adversarial example**
 - 그 **adversarial example**이 다시 원래의 label로 분류 될 수 있도록 loss 값을 줄이는 방향으로 학습.

Adversarial Examples Are Not Bugs, They are Features

- Contribution
 - 1) adversarial example : non-robust feature 라는 새로운 시각 제시
 - 2) 데이터셋에서 robust/non-robust feature 구분하여 뽑아냄.
 - 3) transferability 직관적 설명

Adversarial Examples Are Not Bugs, They are Features

robust feature와 non-robust feature 분류

- 일반적인 dataset에서 어떻게 robust feature와 non-robust feature 분리할 수 있을까?

① Robust feature

- robust model
 - robust feature 만 가지고 판단.
 - we will leverage a robust model and modify our dataset to contain only the features that are relevant to that model.
 - robust model 과 관련된 feature 만을 포함한 dataset 생성.

$$\mathbb{E}_{(x,y) \sim \hat{\mathcal{D}}_R} [f(x) \cdot y] = \begin{cases} \mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x) \cdot y] & \text{if } f \in F_C \\ 0 & \text{otherwise,} \end{cases}$$

robust dataset은 해당 값 (2개 중에 하나)를 가진다고 정의.

- 1) robust model과 동일한 feature에 대해서는 robust feature와 동일한 값
- 2) 그 외의 다른 feature : 0

$$\min_{x_r} \|g(x_r) - g(x)\|_2,$$

objective function

- 타겟 이미지를 robust feature만 가지도록 함 => robust dataset 생성.

Adversarial Examples Are Not Bugs, They are Features

$$\min_{x_r} \|g(x_r) - g(x)\|_2,$$

objective function

- robust feature : cat
 - non-robust feature : cat
- $x\{r\}$
 - robust 한 데이터로 바꿀 데이터.
 - 타겟으로 하고 있는 label 이 아닌, 다른 label 을 가진 데이터
 - 타겟 이미지가 있을 때, 타겟 이미지의 robust 한 버전을 만들기 위하여
 - 초기에 임의의 $x\{r\}$ 이미지와 타겟 이미지가 동일한 feature 값을 가지 트.
 - adversarial training 된 모델 이용.
 - 즉, $x\{r\}$ 과 x 의 같은 robust feature 를 가지도록 update.
 - 결과적으로 $x\{r\}$ 은 x 의 robustified 버전.
 - $g(x)$: 어떤 입력 데이터가 들어왔을 때, 특정한 representation layer 까지 mapping 하 는 함수. 즉, x 의 feature 값 반환하는 함수.

target data : x



$x\{r\}$



타겟으로 하고 있는 label 이 아닌, 다른 label 을 가진 데이터

- robust feature : horse
- non-robust feature : horse

이미지 업데이

- 1) adversarial training 된 모델을 이용하여, target 이미지와 동일한 robust feature 를 가지도록 feature update
- 2)



$x\{r\}$: x 의 robustified 버전.

- robust feature : cat
- non-robust feature : 0

Adversarial Examples Are Not Bugs, They are Features

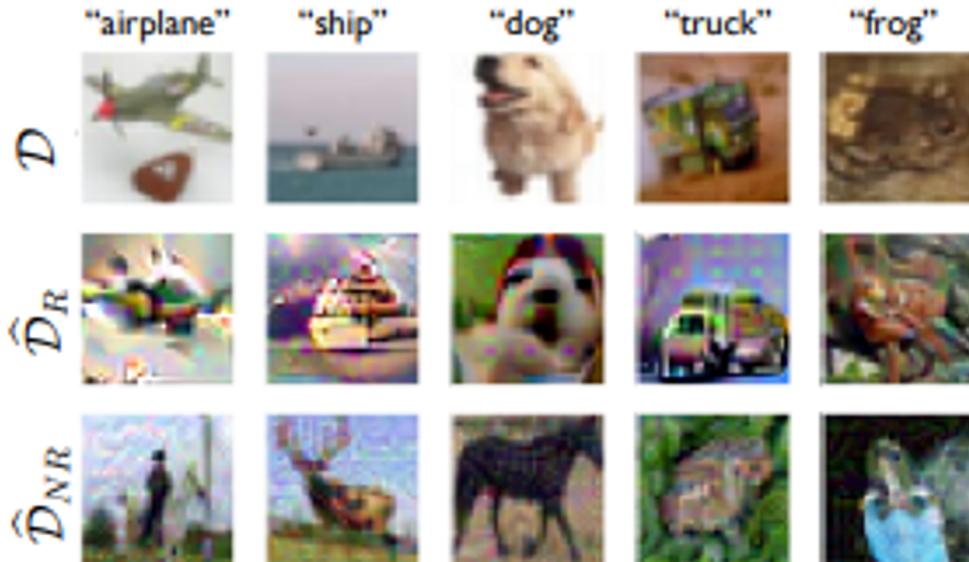
robust feature 와 non-robust feature 분류

- 일반적인 dataset에서 어떻게 robust feature 와 non-robust feature 분리할 수 있을까?

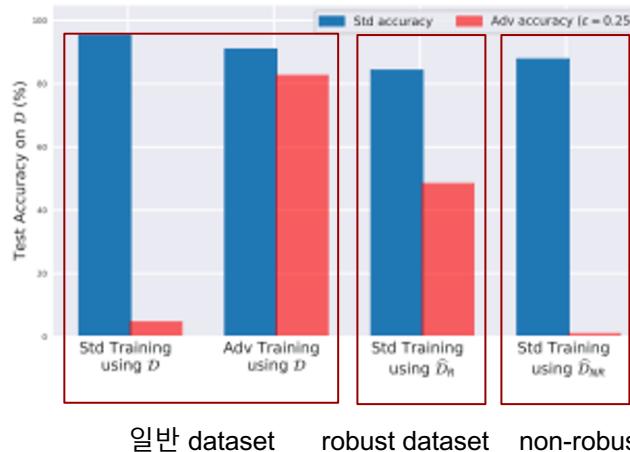
② non-robust feature

- 일반적으로 학습된 standard model에서 adversarial attack 이용.

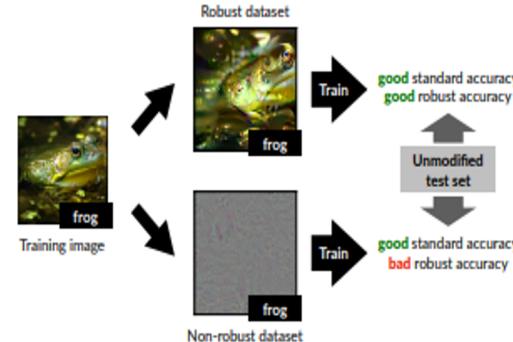
기존의 dataset에서 robust feature 와 non-robust feature 를 구분해서 dataset 생성.



Adversarial Examples Are Not Bugs, They are Features

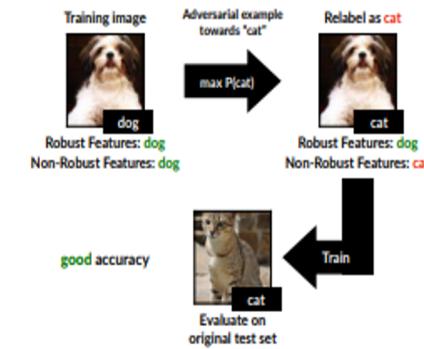


A ‘robustified’ version
for robust classification



(a)

A ‘non-robust’ version
for non-robust classification



(b)

Figure 1: A conceptual diagram of the experiments of Section 3. In (a) we disentangle features into combinations of robust/non-robust features (Section 3.1). In (b) we construct a dataset which appears mislabeled to humans (via adversarial examples) but results in good accuracy on the original test set (Section 3.2).

Overall, our findings corroborate the hypothesis that **adversarial examples can arise from (non-robust) features of the data itself**. By filtering out non-robust features from the dataset (e.g. by restricting the set of available features to those used by a robust model), one can train a significantly more robust model using *standard training*.

adversarial example 은 non-robust feature 로부터 기인함!

Adversarial Examples Are Not Bugs, They are Features

Non-robust features suffice for standard classification

- non-robust feature 만 가지고도 standard accuracy 를 높게 가질 수 있음.
- non-robust feature
 - 일반적으로 모델이 더 잘 동작하기 위해서 사용되는 개체
 - non-robust feature 만을 가진 dataset 생성

$$x_{adv} = \arg \min_{\|x' - x\| \leq \epsilon} L_C(x', t),$$

- adversarial example 를 만들기 위한 loss term
 - 특정 target 으로 분류가 될 수 있도록 만드는 objective function.
 - 특정 target 으로 label 를 설정 => 해당 데이터로 학습. => target class 를 어떻게 설정할까?



특정 target (cat) 으로 분류 되도록 adversarial example 생성.
특정 target (cat) 으로 label 설정.

기존 이미지에 남아 있는 robust feature (dog) 는 기존 이미지에 대한 class (dog) 와 연관성을 가짐.
• target label 을 결정할 때는 원래 label (dog, y) 에 기반하여 결정하는 것보다 random하게 결정하는 편이 좋음.

Robust features: dog
Non-robust features: cat

Adversarial Examples Are Not Bugs, They are Features

Source Dataset	Dataset	
	CIFAR-10	ImageNet _R
\mathcal{D}	95.3%	96.6%
$\hat{\mathcal{D}}_{rand}$	63.3%	87.9%
$\hat{\mathcal{D}}_{det}$	43.7%	64.4%

기존 이미지에 남아 있는 robust feature (dog)는 기존 이미지에 대한 class (dog)와 연관성을 가짐.

- target label을 결정할 때는 원래 label (dog, y)에 기반하여 결정하는 것보다 random하게 결정하는 편이 좋음.
- e.g. 모든 강아지 사진에 대해서 고양이로 target 된 non-robust feature 데이터를 생성한다면, 모델이 강아지 robust feature를 가지고 고양이라고 분류할 수도 있음!

$$\mathbb{E}_{(x,y) \sim \hat{\mathcal{D}}_{rand}} [y \cdot f(x)] \begin{cases} > 0 & \text{if } f \text{ non-robustly useful under } \mathcal{D}, \\ \simeq 0 & \text{otherwise.} \end{cases}$$

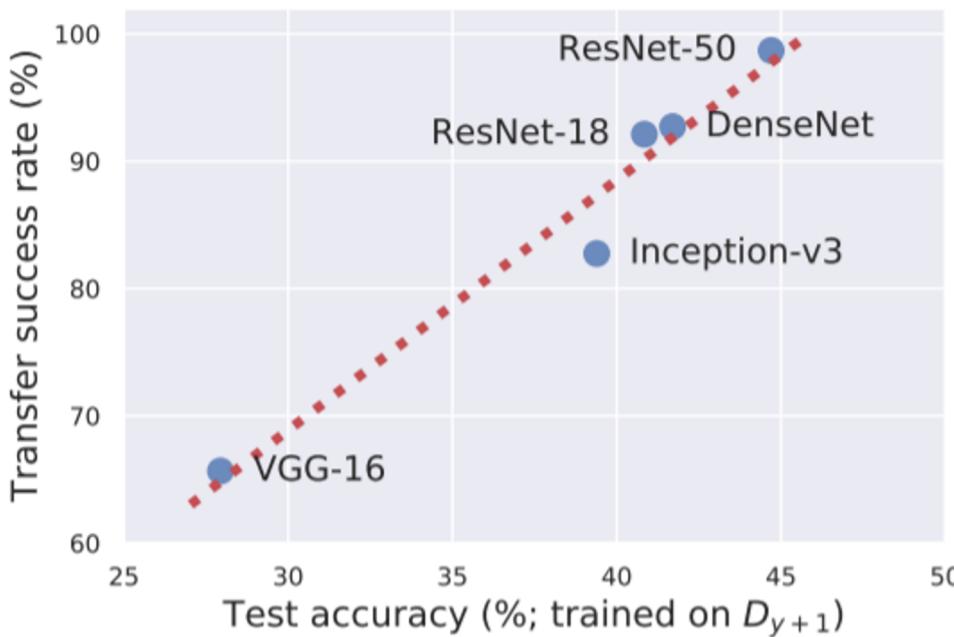
non-robust feature만 가지고 모델이 학습할 수 있도록 함.

$$\mathbb{E}_{(x,y) \sim \hat{\mathcal{D}}_{det}} [y \cdot f(x)] \begin{cases} > 0 & \text{if } f \text{ non-robustly useful under } \mathcal{D}, \\ < 0 & \text{if } f \text{ robustly useful under } \mathcal{D} \\ \in \mathbb{R} & \text{otherwise } (f \text{ not useful under } \mathcal{D})^{11} \end{cases}$$

non-robust feature 뿐만 아니라, robust feature도 target label과 어느정도 연관성이 있도록 데이터가 구성됨. 상대적으로 성능이 안 좋을 수 있음.

Adversarial Examples Are Not Bugs, They are Features

- transferability
 - 서로 다른 아키텍처를 가지더라도, 비슷한 non-robust feature 를 모델이 학습할 수 있음.
 - 따라서 공통적인 non-robust feature 가 여러 모델에 대해서 공통적으로 특정 클래스로 분류가 되도록 만들어질 수 있음.
 - 여러 개의 모델들이 resnet-50 에서 만들어진 adversarial examples 을 이용하여 non-robust feature 만으로 학습을 한 것.
 - 각각의 네트워크에 attack.
 - transfer attack 의 성공률이 test accuracy 와 비례 관계



test accuracy

- non-robust feature 만 가진 dataset 으로 학습한 모델이 일반적인 dataset 에 대하여 예측한 결과.

=> accuracy 가 높을수록 transfer 가 잘 된다는 것!

- 1) 이는, 모델이 non-robust feature 에 기반해서 판단을 하기 때문에, non-robust feature 를 변경해서 다른 모델로 전송을 하는 transfer attack 에도 취약함.
- 2) 모델들이 non-robust data에 대해서만 민감하게 반응을 하기 때문에 adversarial example 공격에도 민감하게 공격 당함.
- 3) 학습이 잘 될 수록, non-robust feature 에 민감하다는 뜻.

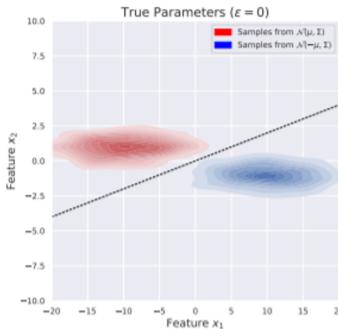
accuracy 가 높을수록, non-robust feature 를 잘 판단함.
non-robust feature에 모델이 sensitive 할 수록, transfer attack 이 잘됨.
따라서 transferability 는 non-robust feature 로부터 기인함.

Adversarial Examples Are Not Bugs, They are Features

A Theoretical Framework for Studying (Non)-Robust Features

- 이전 섹션까지는 real-world 실제 데이터셋을 가지고 empirical behavior (경험적 실험) 을 보였고, 해당 모델에 대해 다양한 속성의 theoretically study (이론적인 연구) 소개함.

두 개의 가우시안 distribution 을 분류하는 binary classification 하는 문제: maximum likelihood classification
=> robust learning (adversarial learning)



$$y \stackrel{\text{u.a.r.}}{\sim} \{-1, +1\}, \quad x \sim \mathcal{N}(y \cdot \mu_*, \Sigma_*),$$

- robust learning 을 위한 objective function

$$\Theta_r = \arg \min_{\mu, \Sigma} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\|\delta\|_2 \leq \varepsilon} \ell(x + \delta; y \cdot \mu, \Sigma) \right],$$

Adversarial Examples Are Not Bugs, They are Features

두 개의 가우시안 distribution 을 분류하는 binary classification 하는 문제. => robust learning (adversarial learning)

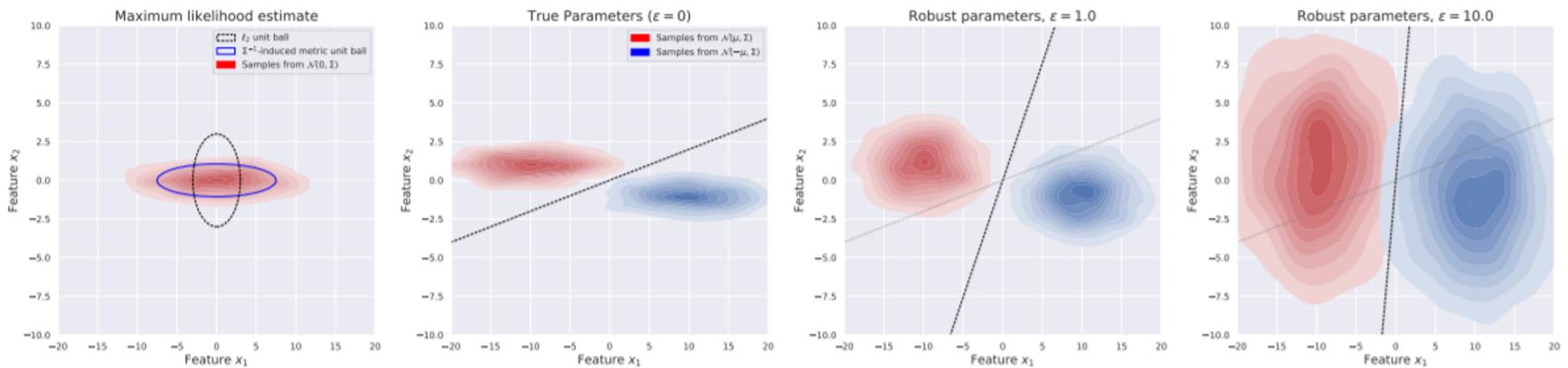
- adversarial training 를 통해,

- 해당 distribution 의 평균값을 바꾸지 않는데
- 공분산 값이 변경됨.

$$x \sim \mathcal{N}(y \cdot \mu_*, \Sigma_*), \quad \Sigma_r = \frac{1}{2}\Sigma_* + \frac{1}{\lambda} \cdot I + \sqrt{\frac{1}{\lambda} \cdot \Sigma_* + \frac{1}{4}\Sigma_*^2},$$

- adversarial perturbation (ϵ) 이 커질수록 공분산 matrix 은 identity matrix 과 유사.
 - 즉, 특정 방향으로 더 이상 민감하지 않음. (adversarial attack 에 견고해짐)
 - 따라서 adversarial example 이 발생할 가능성을 낮출 수 있음.

엡실론이 커짐에 따라, identity matrix 처럼 분산이 생성됨.



Adversarial Examples Are Not Bugs, They are Features

Conclusion

1. adversarial example 현상이 일반적인 ML dataset에서 non-robust features에 대한 highly predictive 때문이라는 관점 제시.
2. 일반적인 dataset에서 robust feature와 non-robust feature를 분리하고, non-robust feature 자체만으로도 좋은 generalization에 충분함을 보임.
3. 이러한 현상을 이론적인 관점에서 해석함.

End of the Document