Adversarial Learning

План

Adversarial Examples

Why it works

Adversarial Attacks

Adversarial Defenses

Adversarial Examples

Adversaries

Adversary – противник, враг, соперник, неприятель, ...

Соперники есть во многих областях:

- Электронная почта: спаммеры
- Распознавание лиц: люди, которые хотят остаться инкогнито
- Проверка состояния машины по фото: недобросовестные водители/таксопарки
- ...

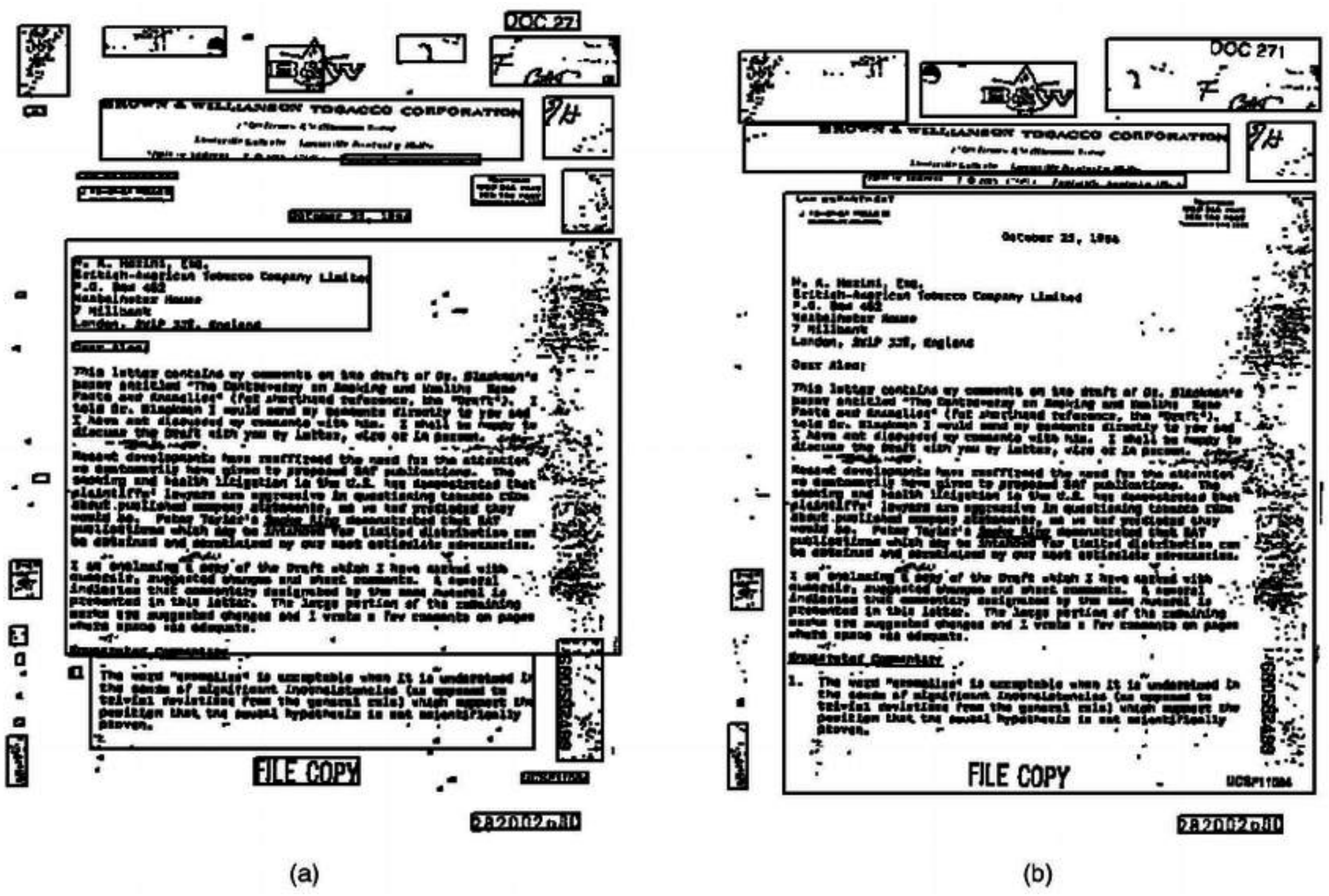
Adversaries

Помимо настоящих соперников есть еще и шум:

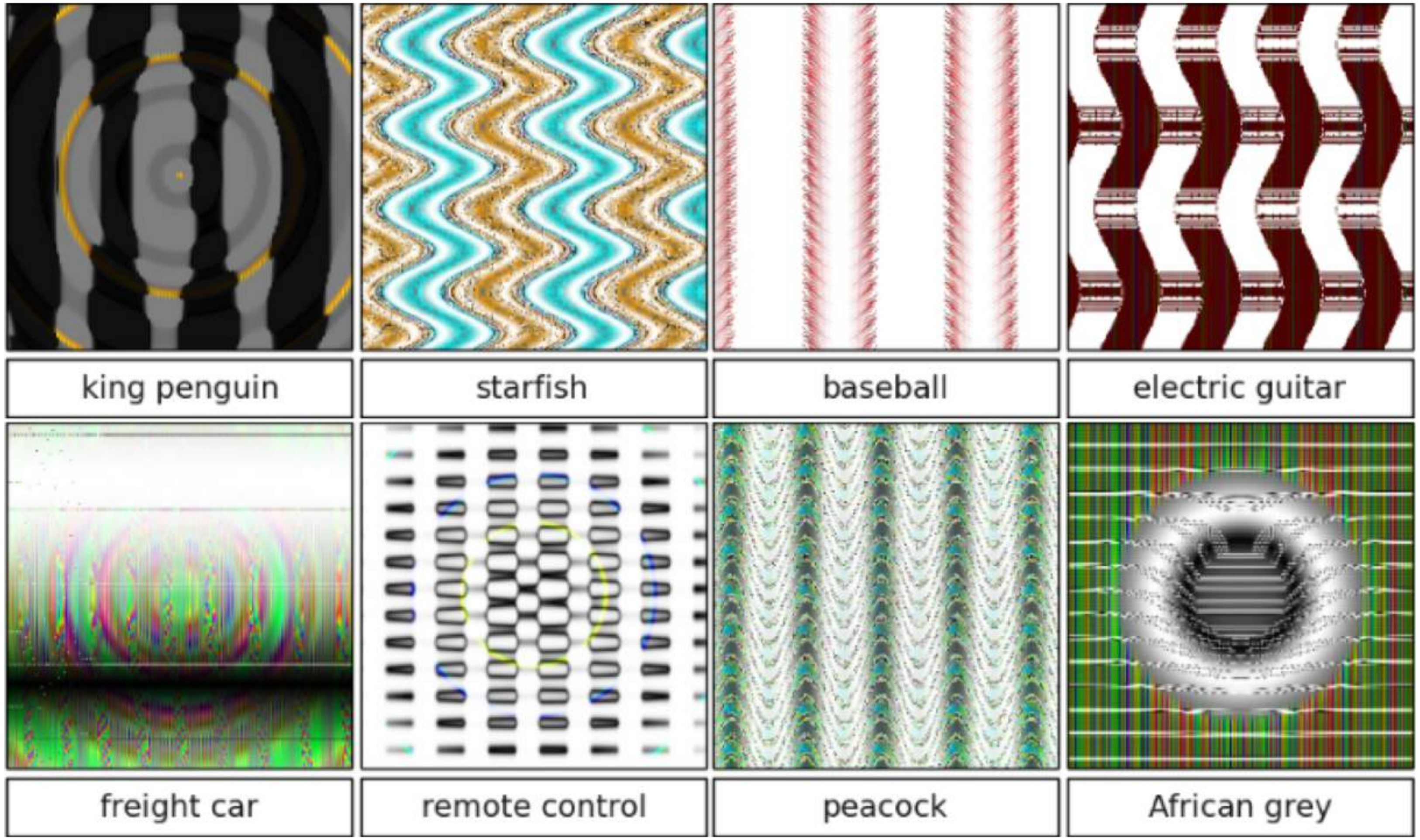
- Распознавание речи: Помехи в записи голоса
- Распознавание лиц: свет, угол обзора, …
- Распознавание текста в распечатанных документах: артефакты печати
- Распознавание дорожных знаков: граффити, стикеры, …
- …

Adversarial Examples in Real Life

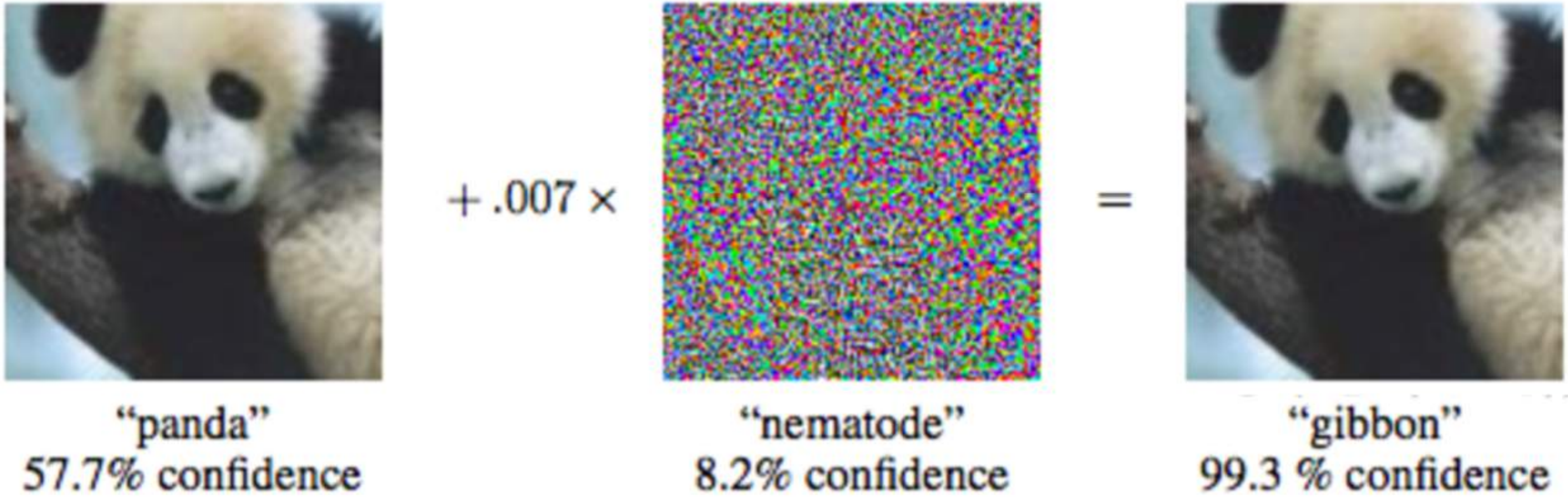
Adversarial Examples in Real Life



Adversarial Examples in Machine Learning

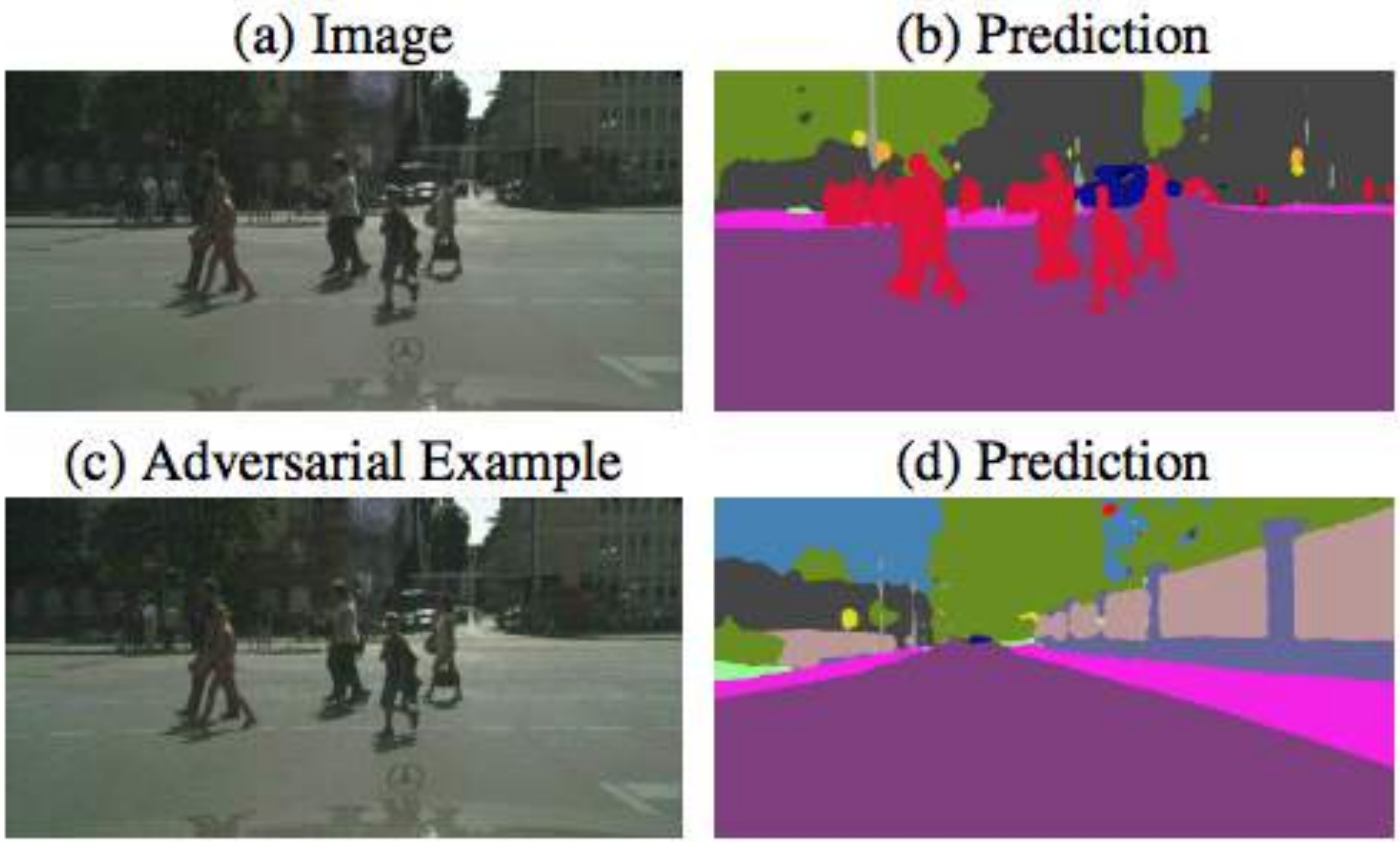


Adversarial Examples in Machine Learning

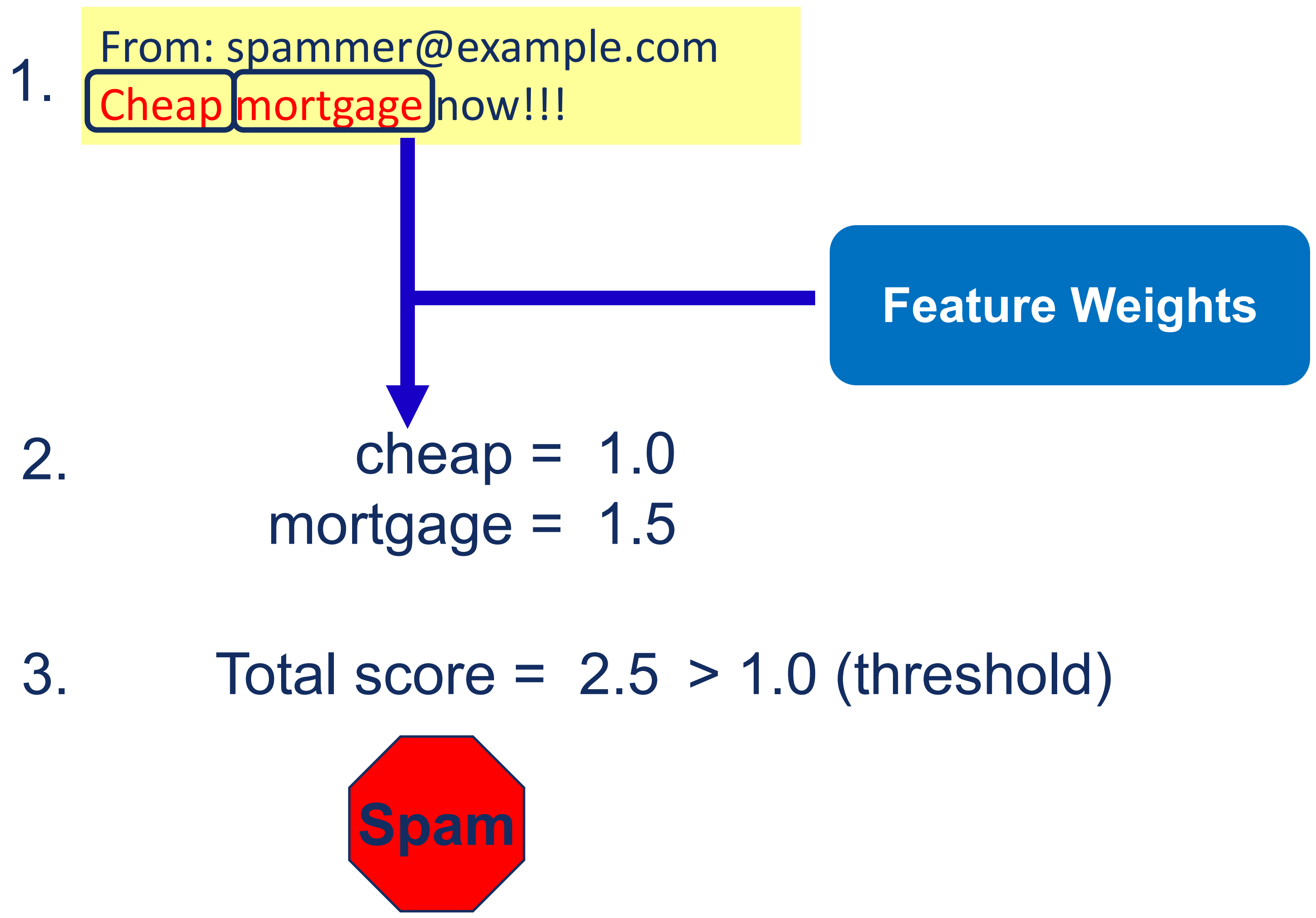


Source: Goodfellow et al. Explaining and Harnessing Adversarial Examples

Adversarial Examples in Machine Learning



Adversarial Examples in Machine Learning



Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

Adversarial Examples in Machine Learning

1. From: spammer@example.com
   Cheap mortgage now!!!
   Joy Oregon

Feature Weights

2. cheap = 1.0
   mortgage = 1.5
   Joy= -1.0
   Oregon = -1.0

3. Total score = 0.5 < 1.0 (threshold)

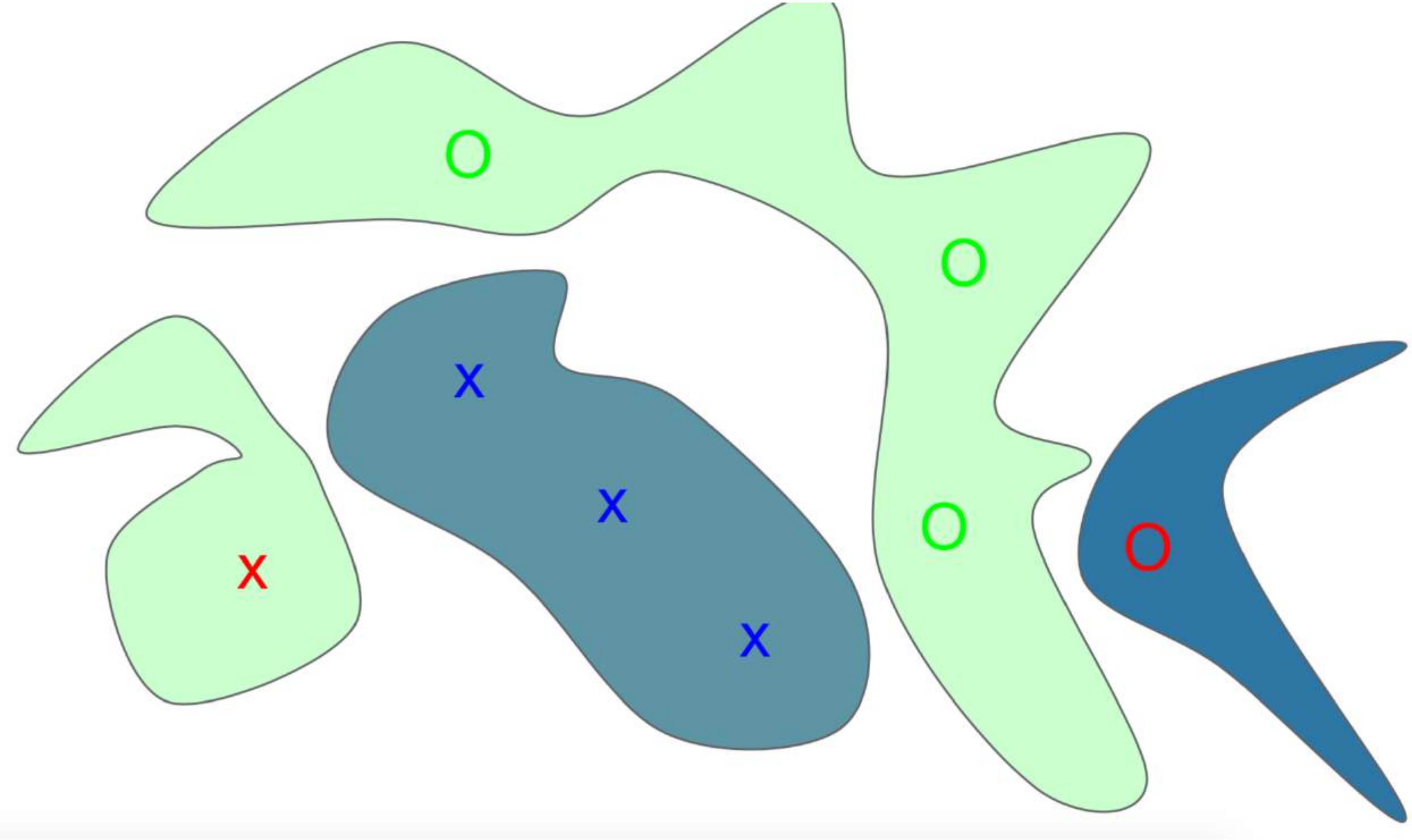
OK

Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

Why it works

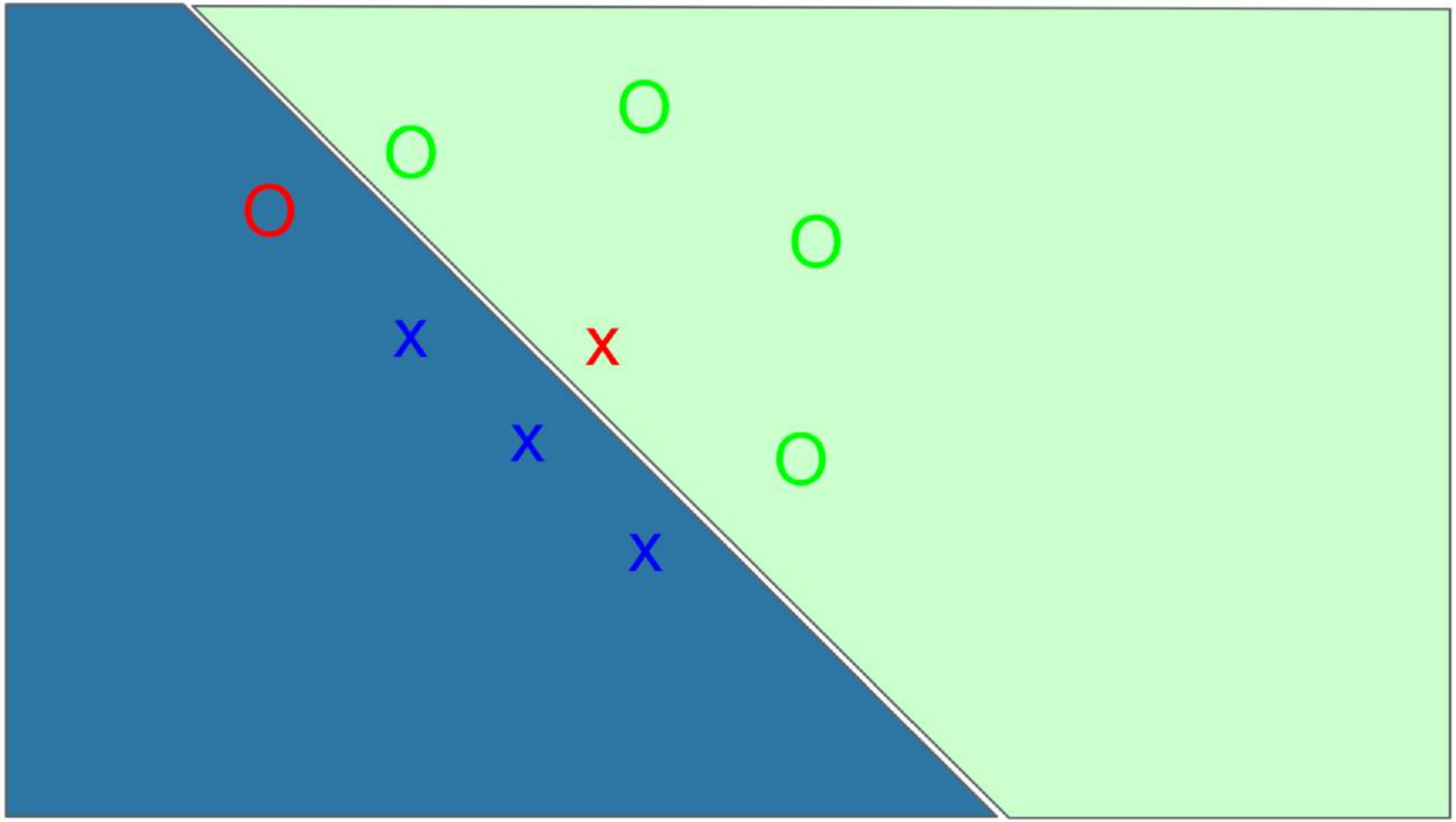
Why it works

Why it works: Overfitting?



Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Why it works: Underfitting?



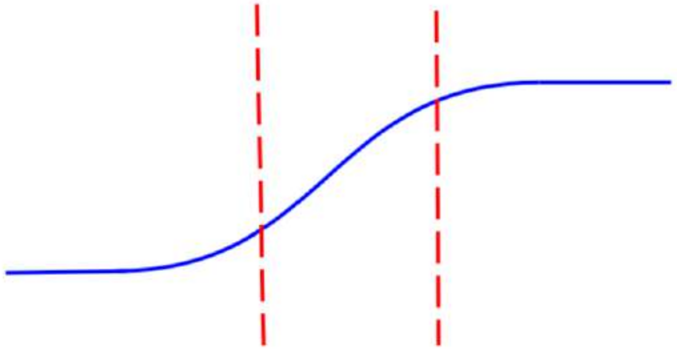
Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Why it works: Excessive Linearity

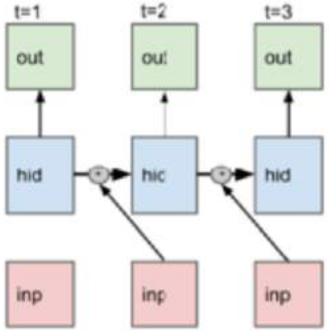
Rectified linear unit

Maxout

Carefully tuned sigmoid



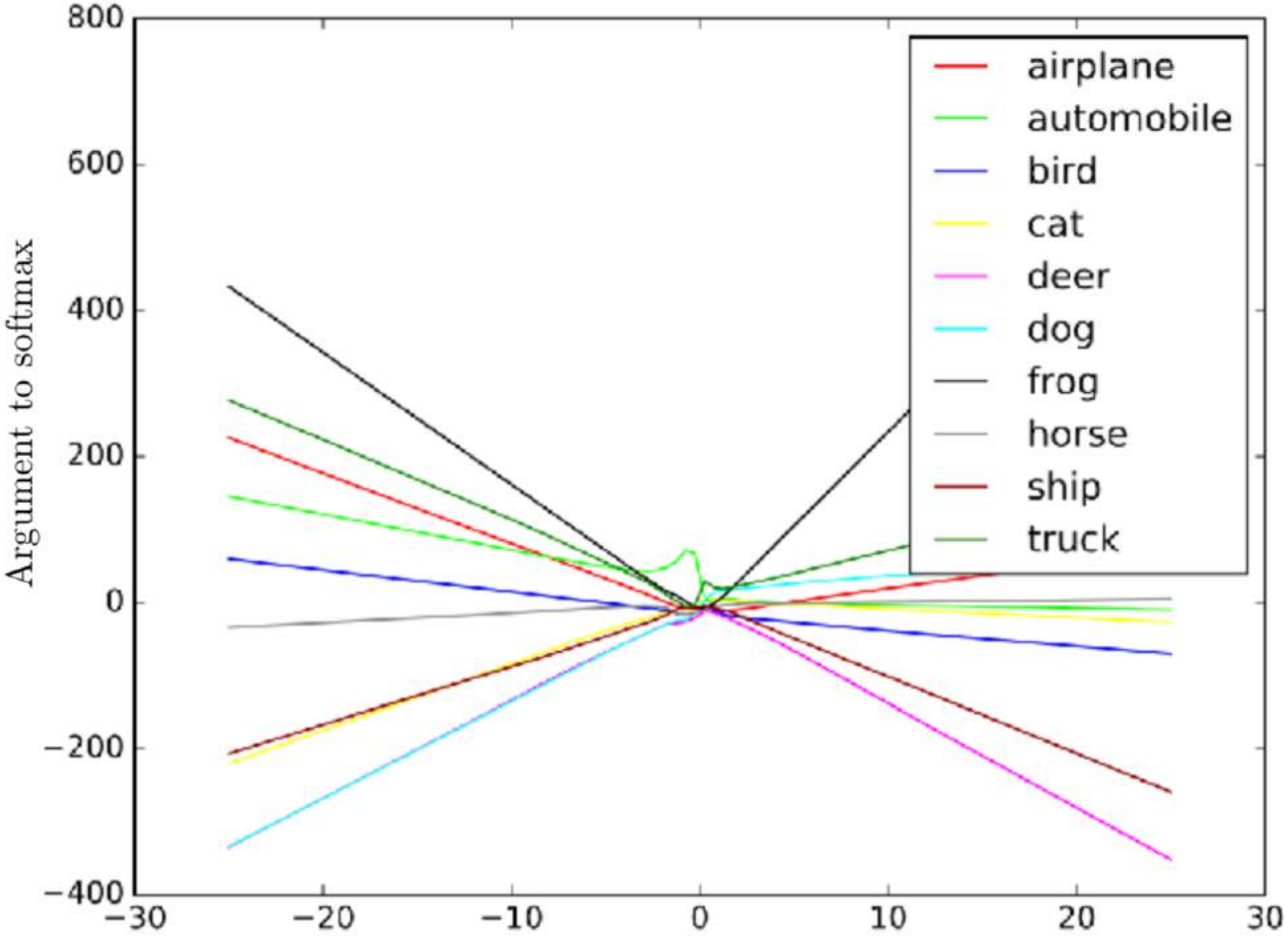
LSTM



Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Why it works: Excessive Linearity

$\epsilon = 0$

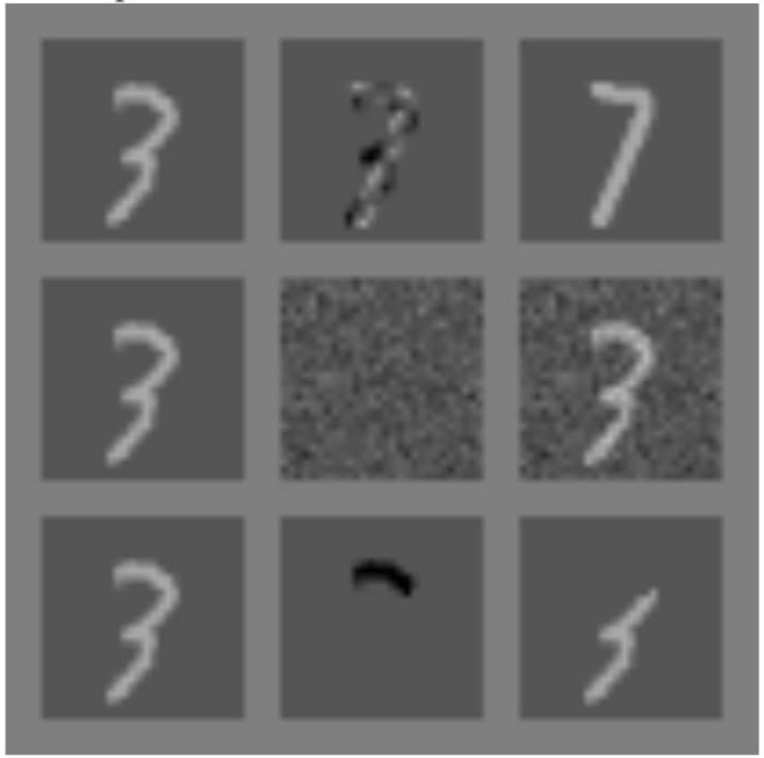


ϵ

Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Why it works: Excessive Linearity

Clean example | Perturbation | Corrupted example



Perturbation changes the true class

Random perturbation does not change the class

Perturbation changes the input to “rubbish class”

All three perturbations have L2 norm 3.96

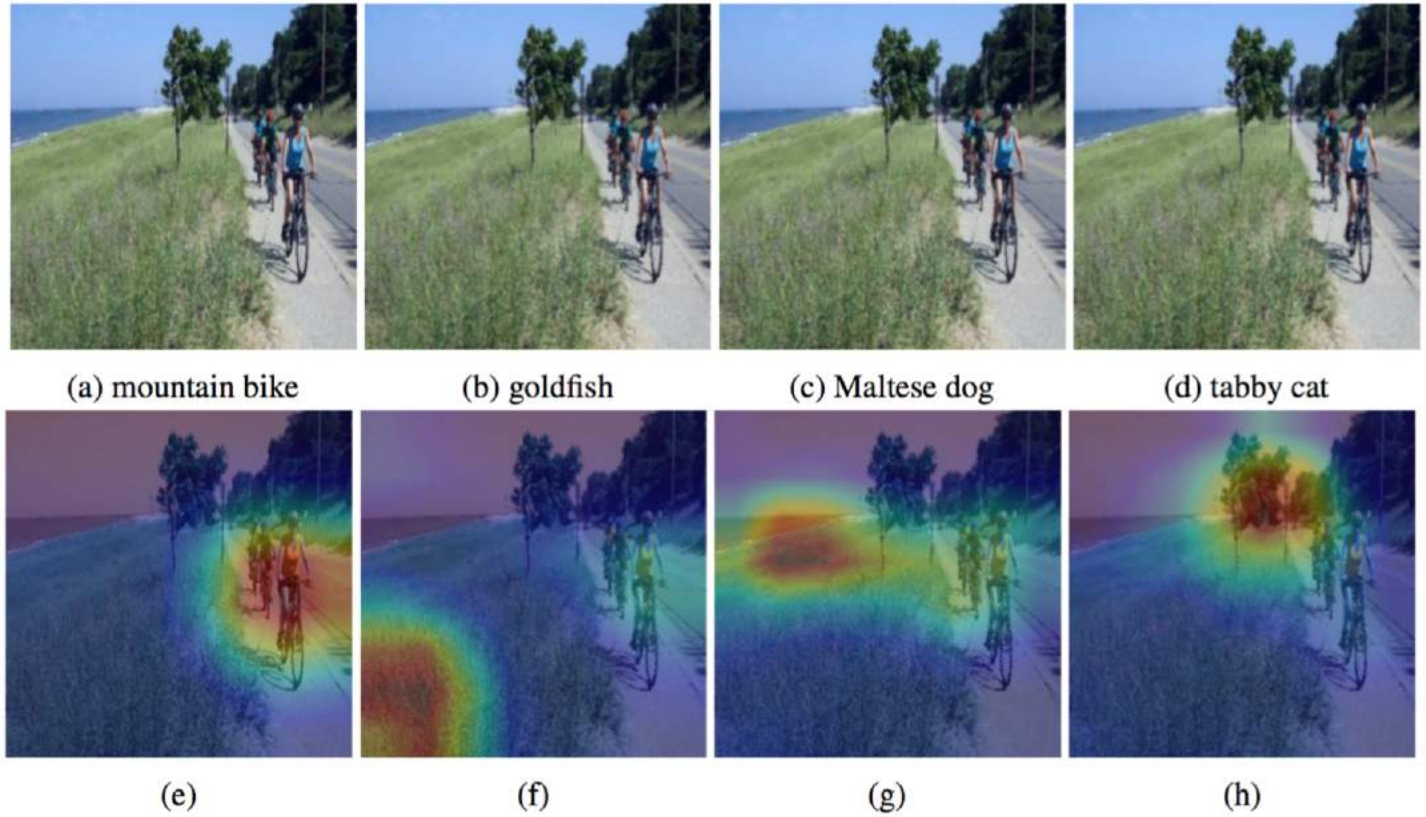
This is actually small. We typically use 7!

Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Why it works: Bottom line

Если найти правильное направление движения в пространстве изображений, то можно с помощью почти невидимых для человека изменений с высокой вероятностью обмануть свёрточную сеть.

Why it works: Bottom line



CAM attention visualization for ImageNet inception_v3 model. (a) the original image and (b)-(d) are stAdv adversarial examples targeting different classes. Row 2 shows the attention visualization for the corresponding images above.

Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

Adversarial Attacks

Adversarial Attacks: Classification

- Evasion attacks
  - Targeted/non-targeted
  - White-box, black-box
- Data Poisoning

White-box Evasion Attacks: Non-targeted FGSM

$$x^{adv} = x + \varepsilon \cdot \mathtt{sign}(\nabla_x J(x, y_{true})),$$

where

x is the input (clean) image,

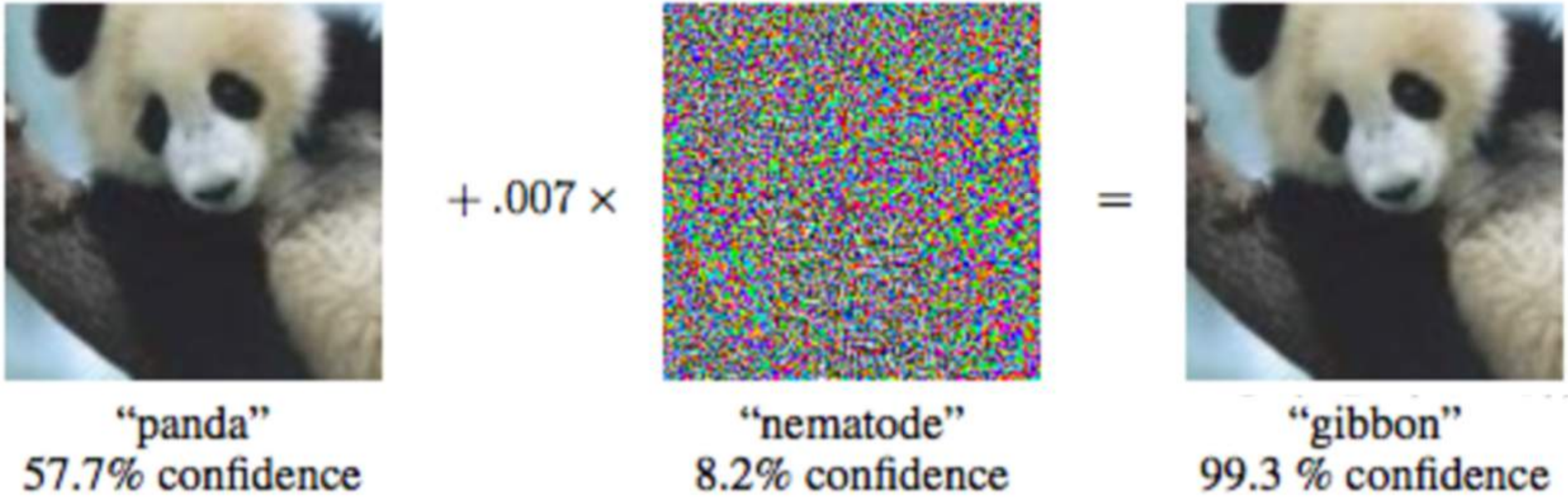
x^{adv} is the perturbed adversarial image,

J is the classification loss function,

y_{true} is true label for the input x.

Source: https://medium.com/onfido-tech/adversarial-attacks-and-defences-for-convolutional-neural-networks-66915ece52e7

White-box Evasion Attacks: Non-targeted FGSM



Source: Goodfellow et al. Explaining and Harnessing Adversarial Examples

White-box Evasion Attacks: Non-targeted I-FGSM

$$x_0^{adv} = x, \quad x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \mathtt{sign}\left(\nabla_x J(x_t^{adv}, y)\right).$$

Source: https://medium.com/onfido-tech/adversarial-attacks-and-defences-for-convolutional-neural-networks-66915ece52e7

White-box Evasion Attacks: Targeted FGSM

$$x^{adv} = x - \varepsilon \cdot \mathtt{sign}(\nabla_x J(x, y_{target})),$$

where

y_{target} is the target label for the adversarial attack.

Source: https://medium.com/onfido-tech/adversarial-attacks-and-defences-for-convolutional-neural-networks-66915ece52e7

White-box Optimization Attacks in General

$$\underset{\delta}{\operatorname{argmin}} \; \lambda||\delta||_p + J(f_\theta(x+\delta), y^*)$$

Perturbation/Noise Matrix

Adversarial Target Label

Lp norm (L-0, L-1, L-2, ...)

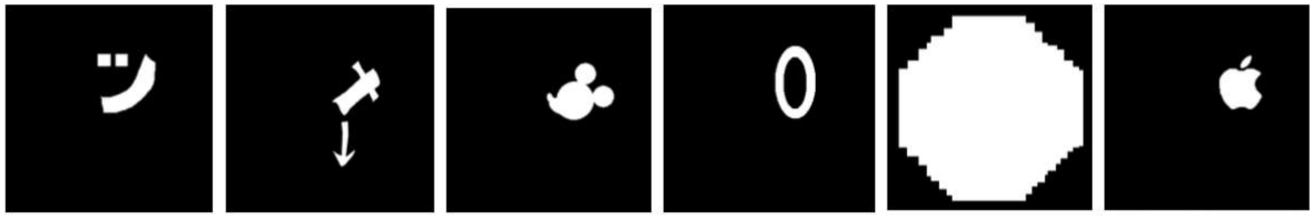
Loss Function

$$\underset{\delta}{\operatorname{argmin}} \; \lambda||\delta||_p + \frac{1}{k}\sum_{i=1}^{k} J(f_\theta(x_i+\delta), y^*)$$

Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

White-box Optimization Attacks in General

$$\underset{\delta}{\operatorname{argmin}}\ \lambda \| M_x \cdot \delta \|_p + \frac{1}{k} \sum_{i=1}^{k} J(f_\theta(x_i + M_x \cdot \delta), y^*)$$



Subtle Poster

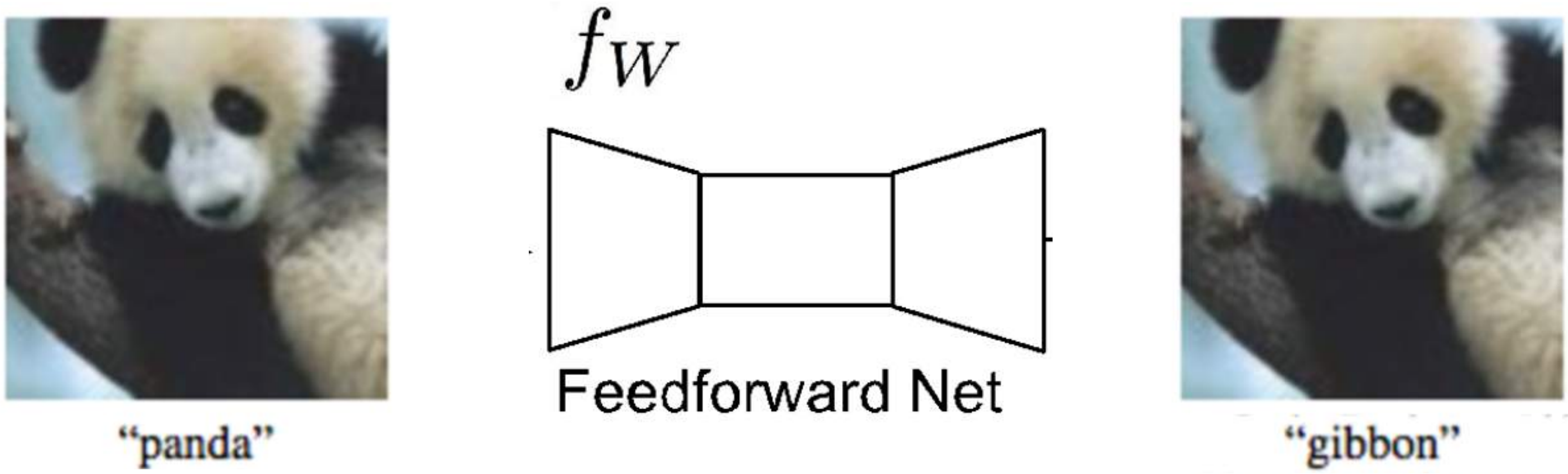
Camouflage Sticker

Mimic vandalism

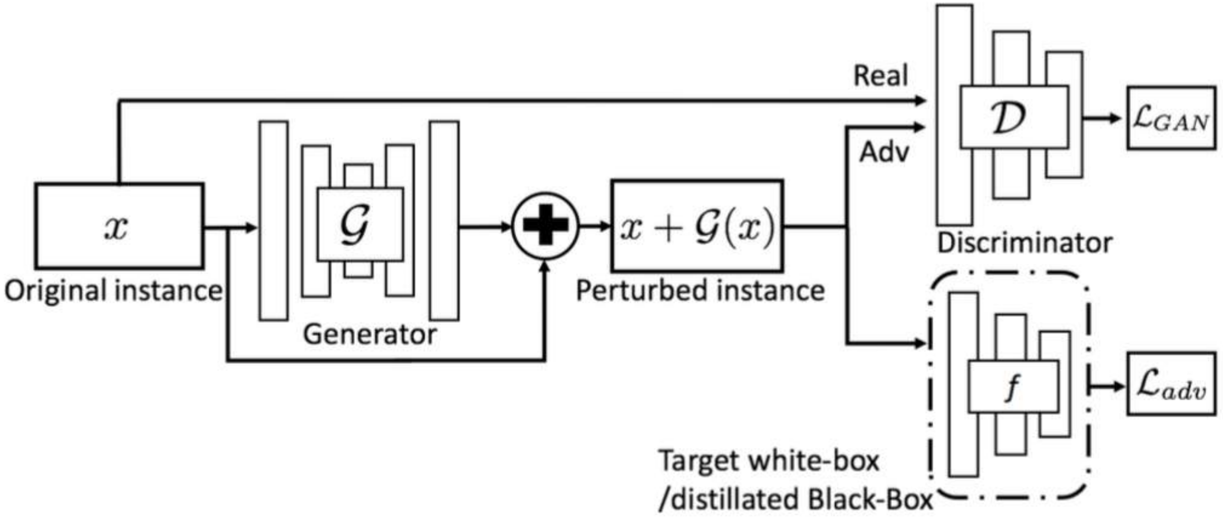
“Hide in the human psyche”

Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

Adversarial Transformation Networks



Generating Adversarial Examples with GANs



Black-box can be performed here via distillation

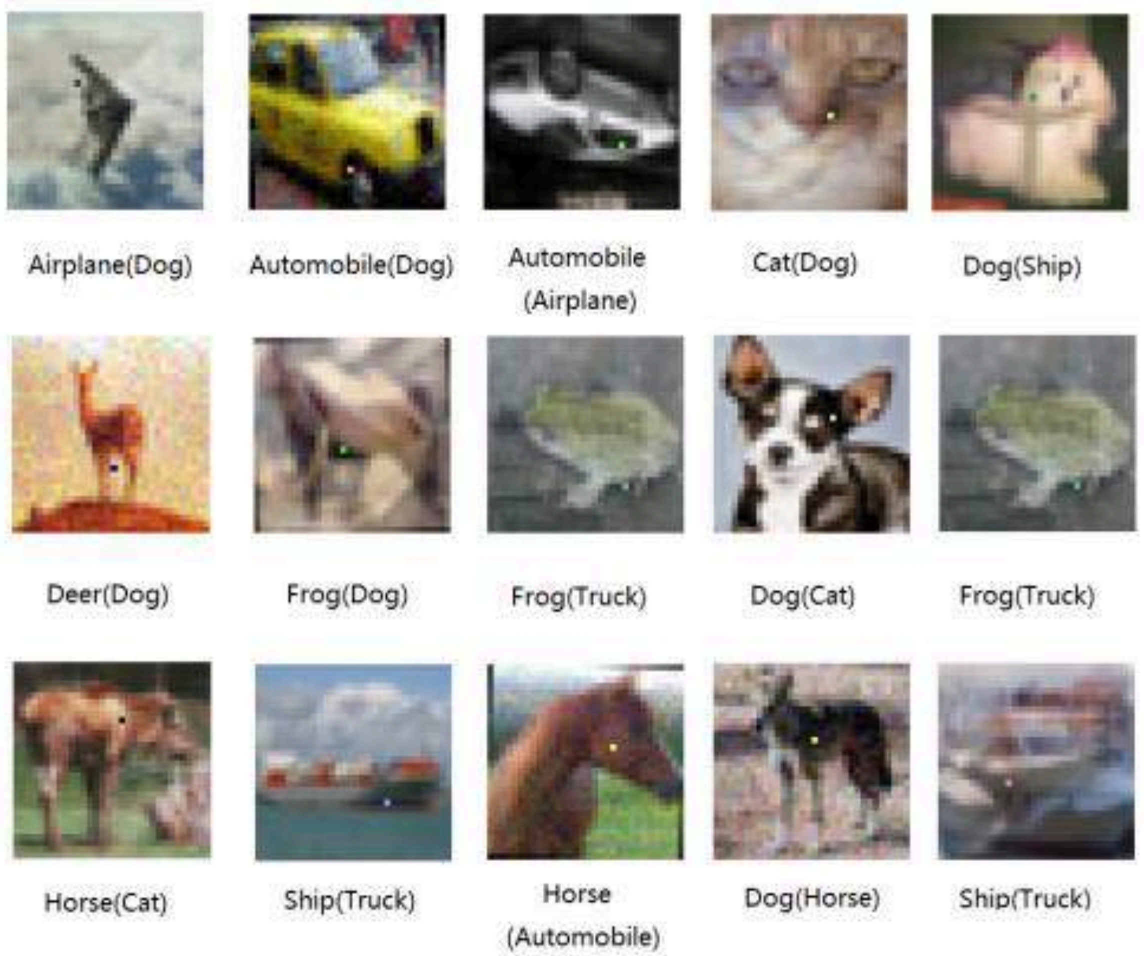
$$\mathcal{L}_{GAN} = \mathbb{E}_{x\sim\mathcal{P}_{\text{data}}(x)} \log \mathcal{D}(x) + \mathbb{E}_{x\sim\mathcal{P}_{\text{data}}(x)} \log(1 - \mathcal{D}(x + \mathcal{G}(x)))$$

$$\mathcal{L} = \mathcal{L}^{f}_{adv} + \alpha\mathcal{L}_{GAN} + \beta\mathcal{L}_{hinge}$$

The GAN loss here tries to ensure the diversity of adversarial examples

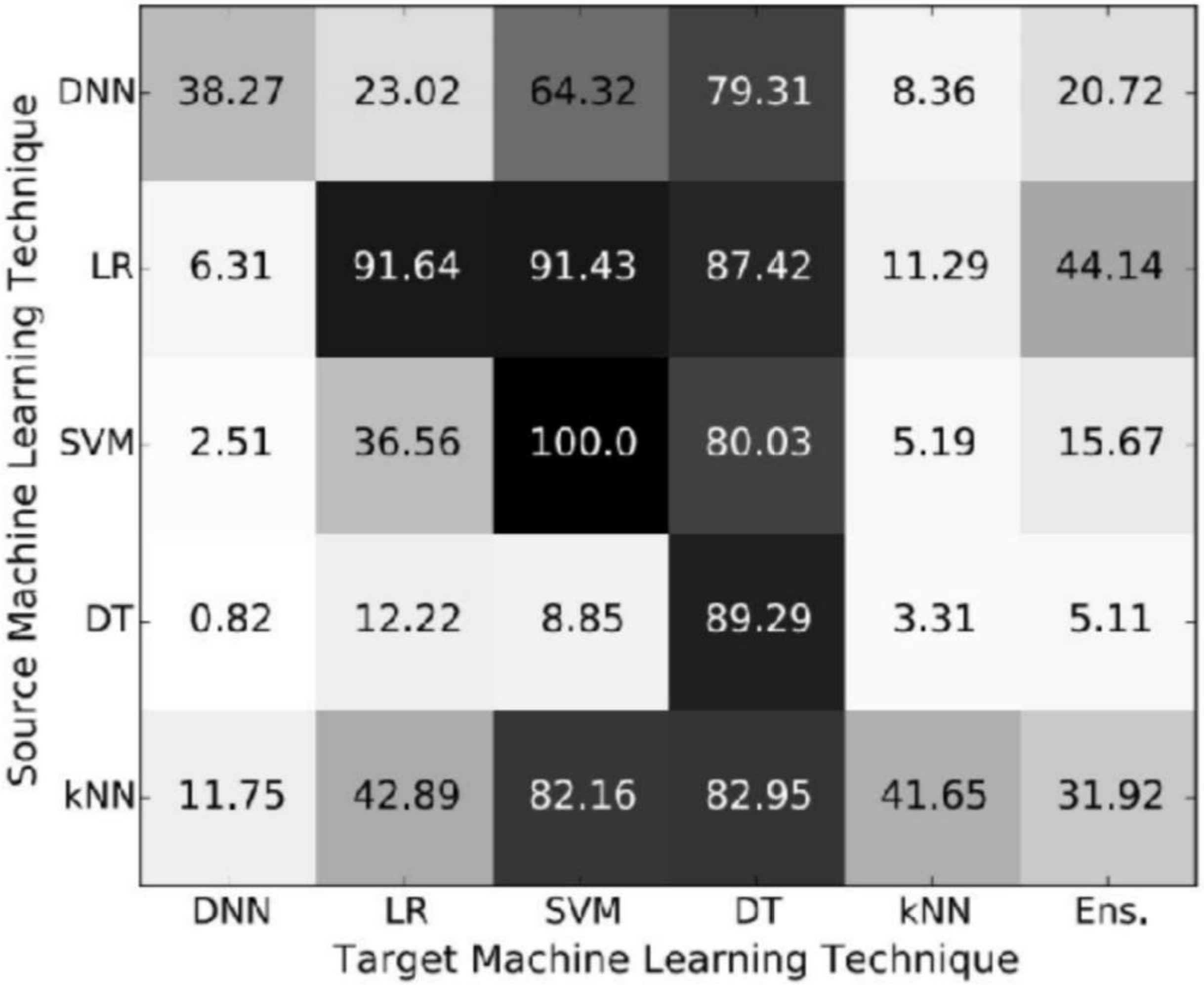
Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

One-pixel Adversarial Attack



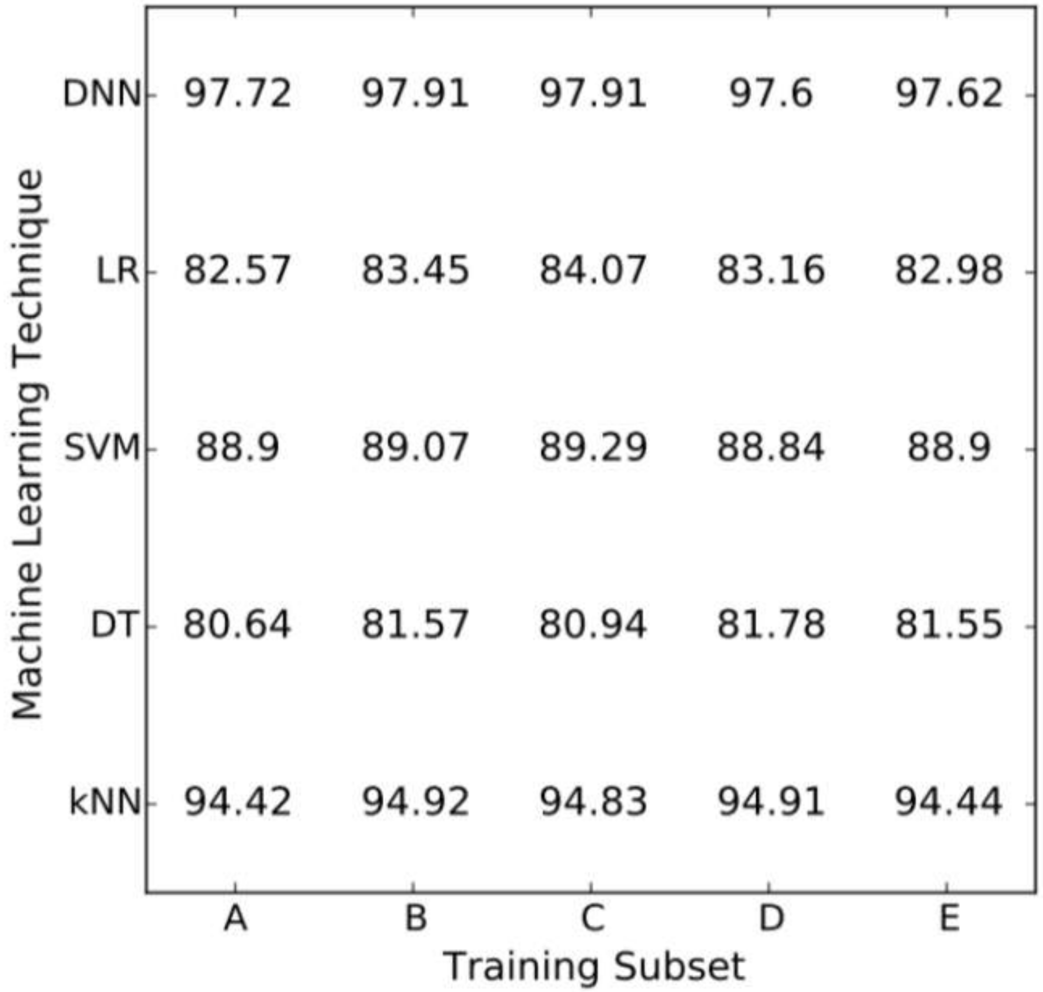
Source: Su et al. One pixel attack for fooling deep neural networks

Model2Model Transfer



Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Data2Data Transfer



Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Black-box Evasion Attacks: FD Method

Given d-dimensional vector x, we can make $2d$ queries to estimate the gradient as below

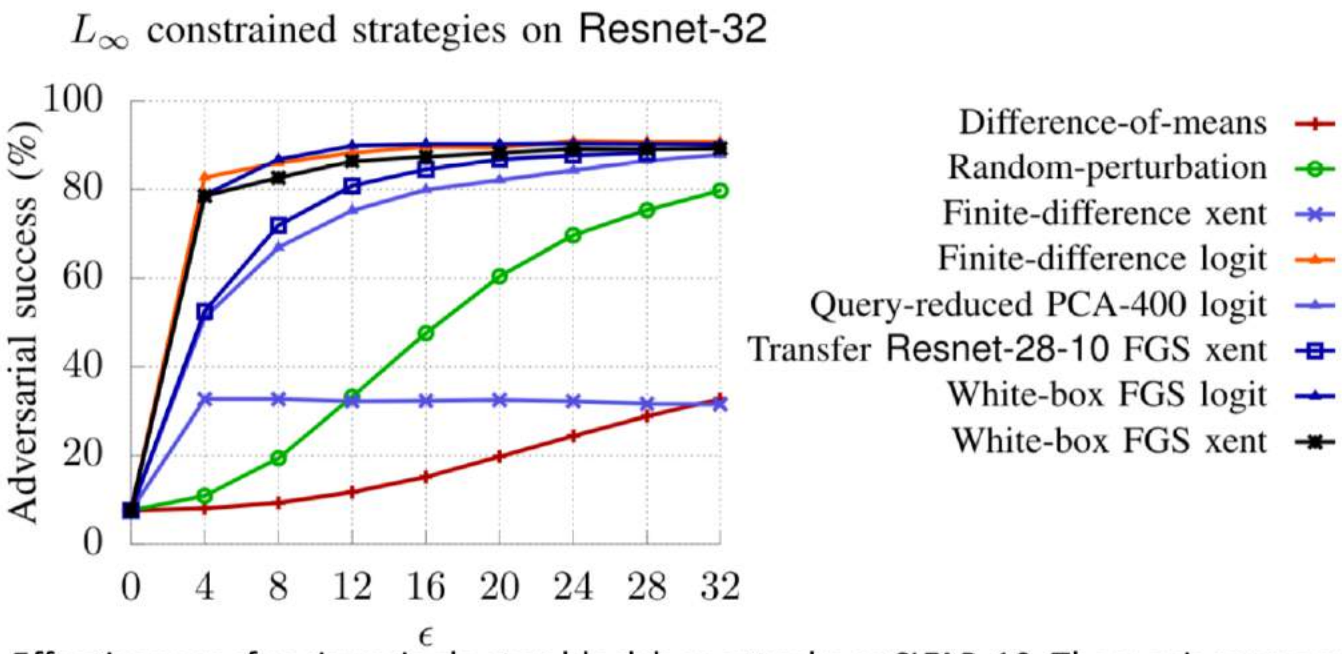
$$\mathrm{FD}_{\mathbf{x}}(g(\mathbf{x}), \delta) = \begin{bmatrix} \frac{g(\mathbf{x}+\delta \mathbf{e}_1)-g(\mathbf{x}-\delta \mathbf{e}_1)}{2\delta} \\ \vdots \\ \frac{g(\mathbf{x}+\delta \mathbf{e}_d)-g(\mathbf{x}-\delta \mathbf{e}_d)}{2\delta} \end{bmatrix}$$

An example of approximate FGS with finite difference

$$x_{adv} = \mathbf{x} + \epsilon \cdot \mathrm{sign}\left(\mathrm{FD}_{\mathbf{x}}(\ell_f(\mathbf{x}, y), \delta)\right)$$

Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

Black-box Evasion Attacks Results



Effectiveness of various single step black-box attacks on CIFAR-10. The y-axis represents the variation in adversarial success as ϵ increases.

Finite Differences method outperform other black-box attacks and achieves similar attach success rate with the white-box attack

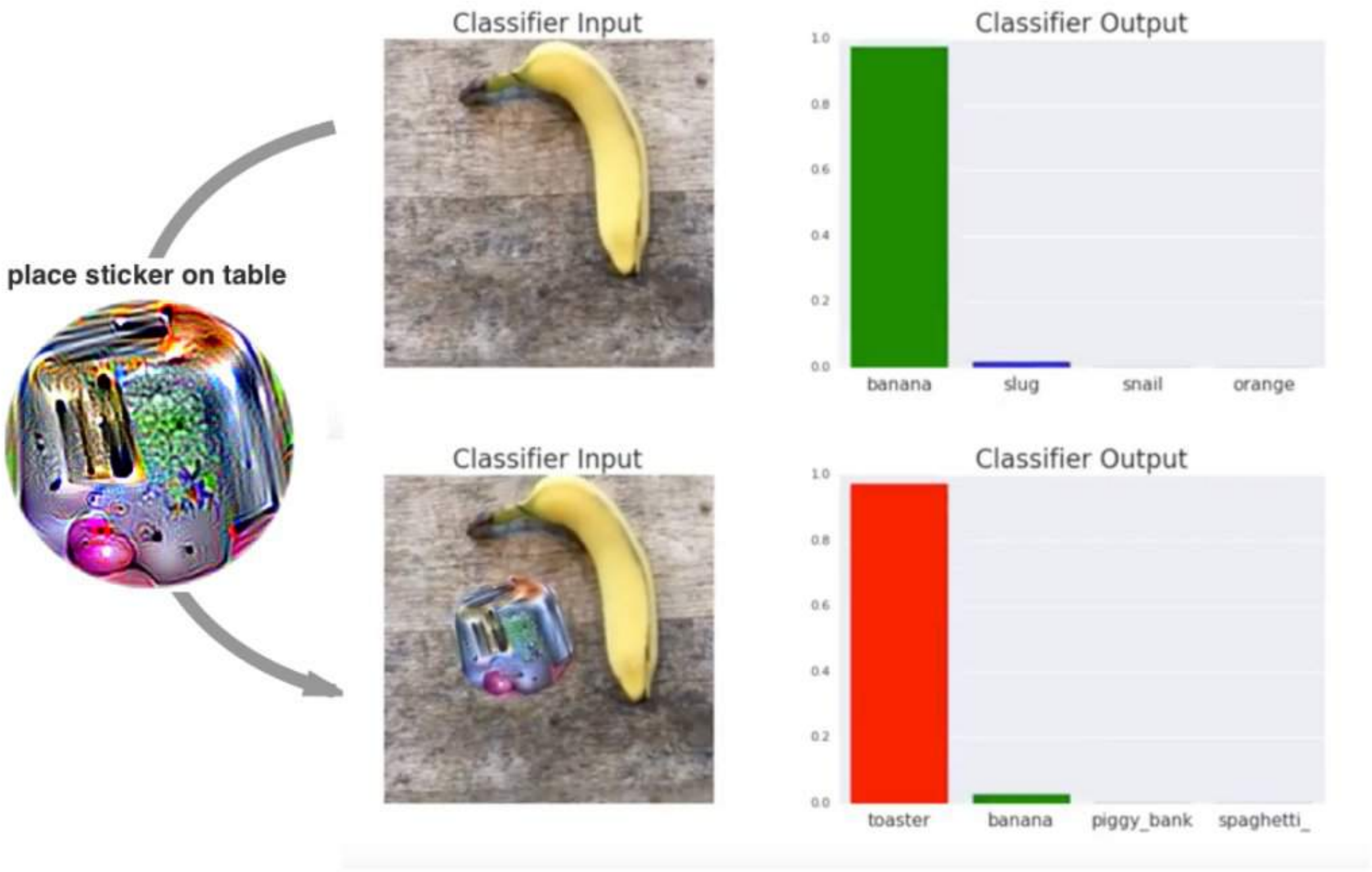
Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

Black-box Evasion Attack: Model Stealing

Если получить ответ black-box классификатора можно почти бесплатно, то можно обучить свой классификатор на ответах black-box классификатора.

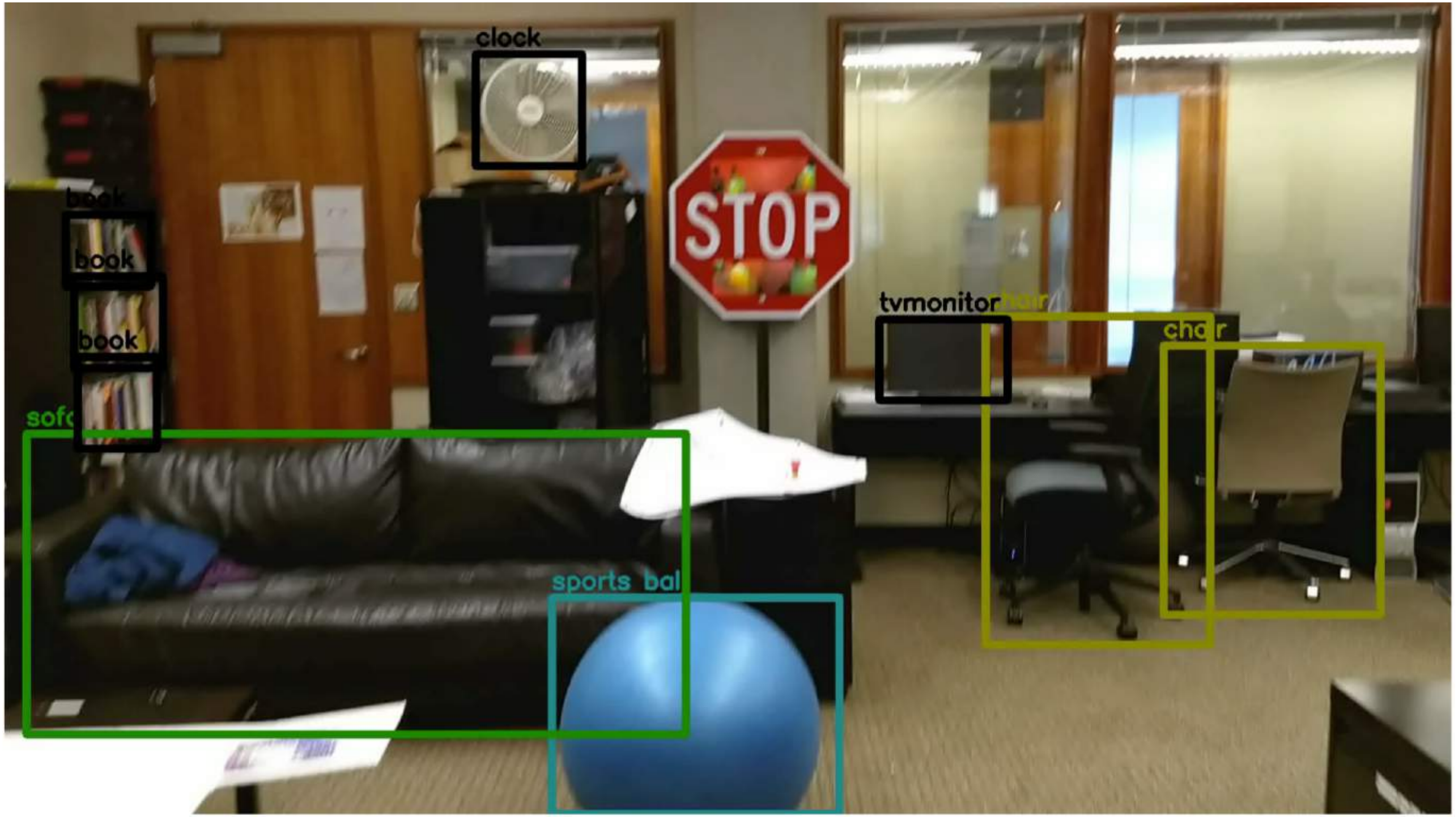
Затем можно научиться обманывать полученный классификатор, и с высокой вероятностью обман будет успешен и для black-box классификатора.

Real World Evasion Attacks: Adversarial Patch



Source: Brown et al. Adversarial Patch

Real World Evasion Attacks: Adversarial Stickers



Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

Data Poisoning

- Direct poisoning of labels
- Deleting/injecting examples
- Injecting noise/perturbations in data

Adversarial Defenses

Failed Defenses

Generative pretraining

Removing perturbation with an autoencoder

Adding noise at test time

Ensembles

Confidence-reducing perturbation at test time

Error correcting codes

Multiple glimpses

Weight decay

Double backprop

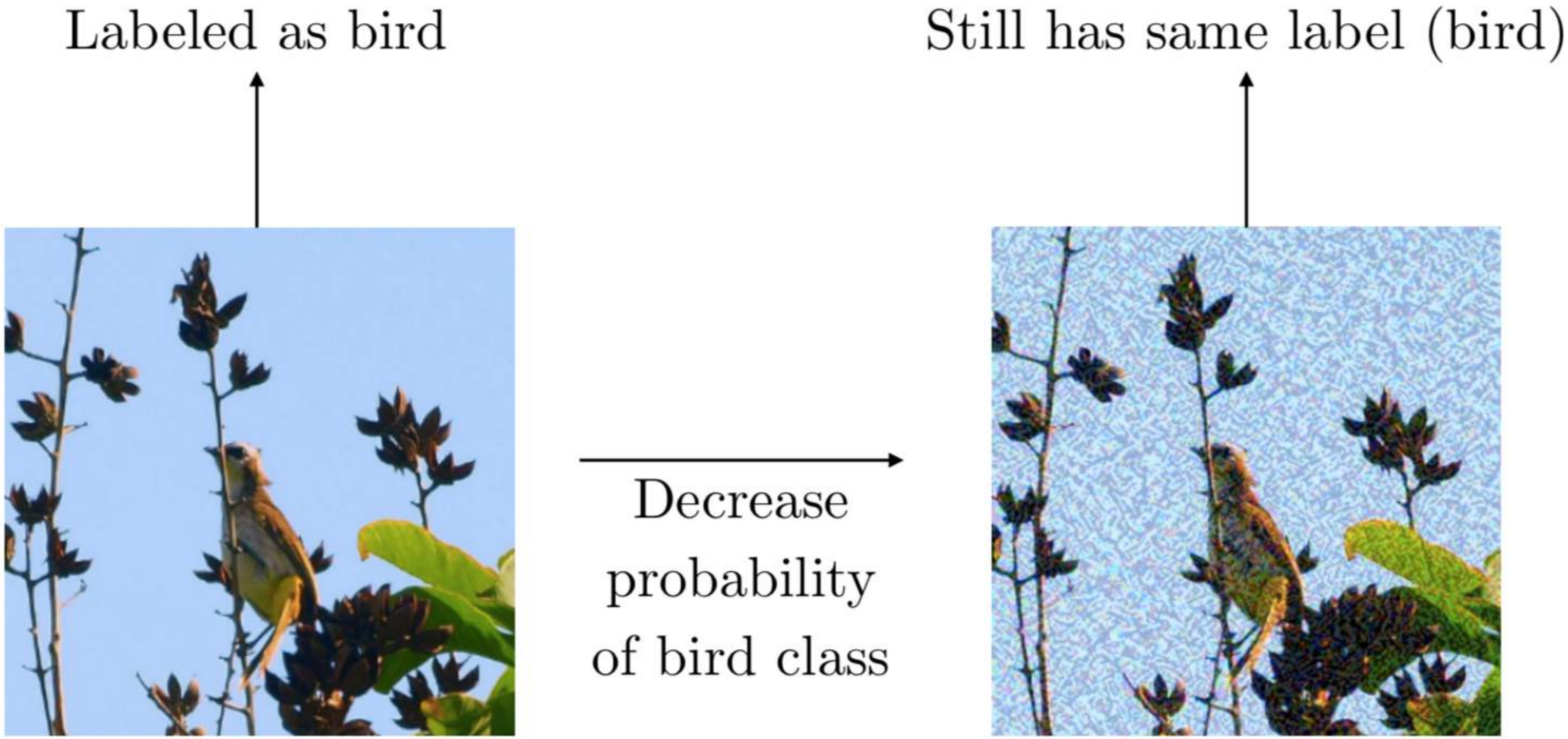
Adding noise at train time

Various non-linear units

Dropout

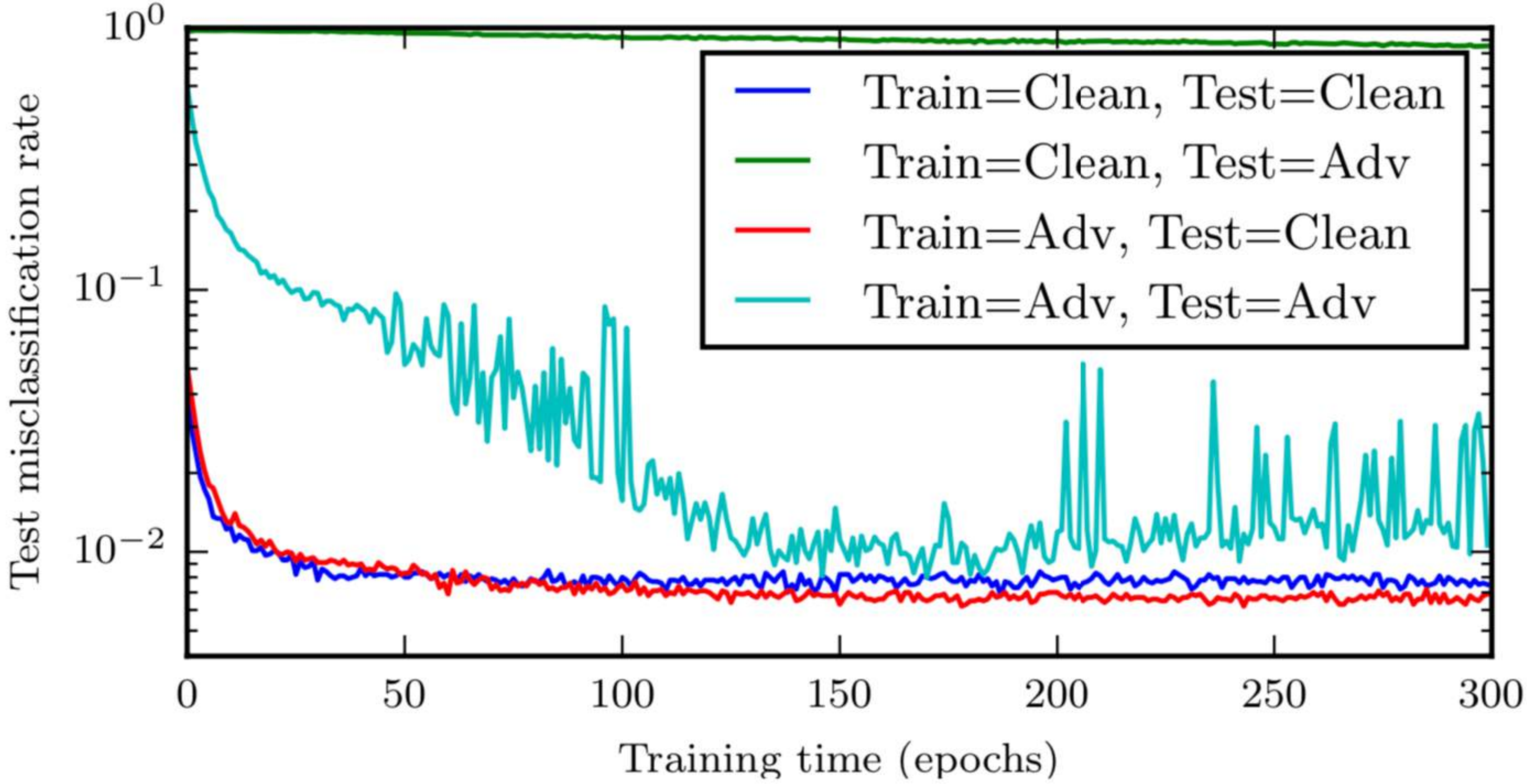
Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Adversarial Training



Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Adversarial Training

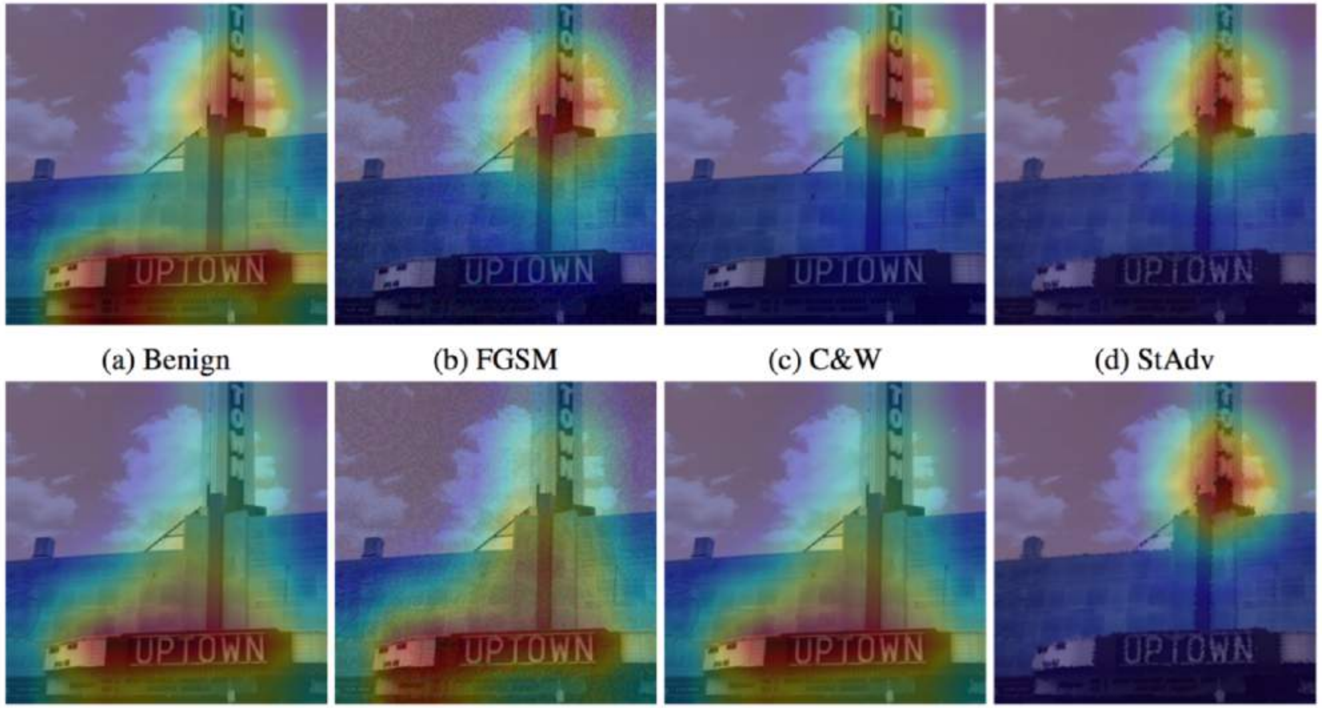


Source: cs231n Lecture 16 Adversarial Examples and Adversarial Training

Adversarial Training

inception_v3 model

Adversarial trained inception_v3 model



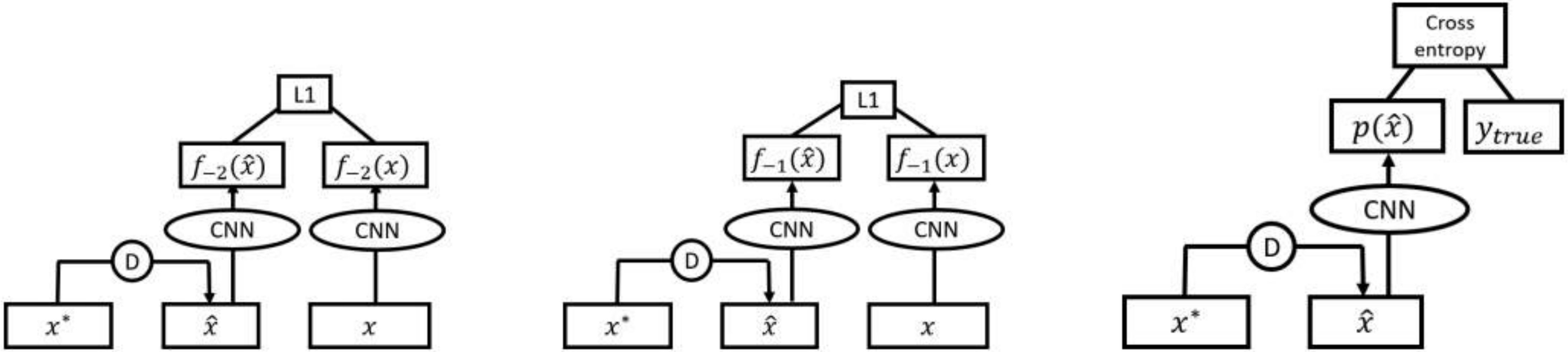
(a) Benign (b) FGSM (c) C&W (d) StAdv

(e) Benign (f) FGSM (g) C&W (h) StAdv

CAM attention visualization for ImageNet inception_v3 model. Column 1 shows the CAM map corresponding to the original image. Column 2-4 show the adversarial examples generated by different methods. (a) and (e)-(g) are labeled as the ground truth “cinema”, while (b)-(d) and (h) are labeled as the adversarial target “missile.”

Source: Adversarial Machine Learning Tutorial (https://aaai18adversarial.github.io)

High Level Representation Guided Denoiser



(a) FGD (b) LGD (c) CGD

Source: Liao et al. Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser

Papers

Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser: https://arxiv.org/abs/1712.02976

Generating Adversarial Examples with Adversarial Networks: https://arxiv.org/abs/1801.02610

Explaining and Harnessing Adversarial Examples: https://arxiv.org/abs/1412.6572

Robust Physical-World Attacks on Deep Learning Models: https://arxiv.org/abs/1707.08945

Adversarial Patch: https://arxiv.org/abs/1712.09665

One pixel attack for fooling deep neural networks: https://arxiv.org/abs/1710.08864

Materials

Adversarial Machine Learning Tutorial (tutorial presentations): https://aaai18adversarial.github.io

cs231n Lecture 16| Adversarial Examples and Adversarial Training:
https://www.youtube.com/watch?v=CIfsB_EYsVI&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv

CleverHans Library: https://github.com/tensorflow/cleverhans