Adversarial Example

EE807: Recent Advances in Deep Learning

Lecture 14

Slide made by

Seokmin Youn and Kimin Lee

KAIST EE

Table of Contents

1. **Introduction**
   - What is the adversarial example?
   - Threat of the adversarial example

2. **Adversarial Attack**
   - White-box attack
   - Black-box attack

3. **Adversarial Defense**
   - Adversarial training
   - Input pre-processing
   - Robust network construction

Table of Contents

1. Introduction

- What is the adversarial example?
- Threat of the adversarial example

2. Adversarial Attack

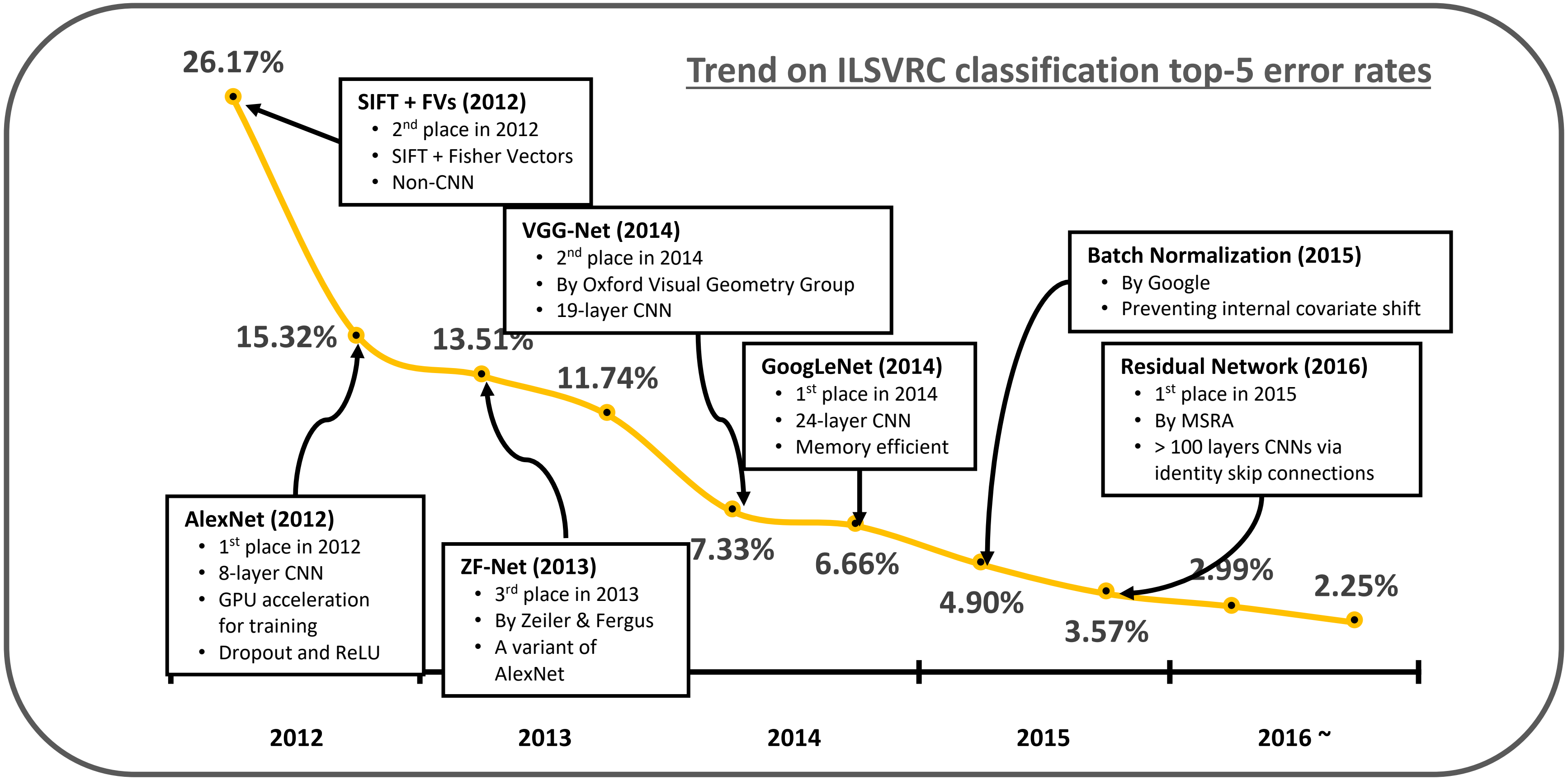
- White-box attack
- Black-box attack

3. Adversarial Defense

- Adversarial training
- Input pre-processing
- Robust network construction

Back Side of The Neural Network Success

- Nowadays, **Convolutional Neural Network** shows impressive performance
  - The problem is that a neural network is highly vulnerable to a small perturbation of an input



Trend on ILSVRC classification top-5 error rates

SIFT + FVs (2012)
- 2nd place in 2012
- SIFT + Fisher Vectors
- Non-CNN

VGG-Net (2014)
- 2nd place in 2014
- By Oxford Visual Geometry Group
- 19-layer CNN

Batch Normalization (2015)
- By Google
- Preventing internal covariate shift

GoogLeNet (2014)
- 1st place in 2014
- 24-layer CNN
- Memory efficient

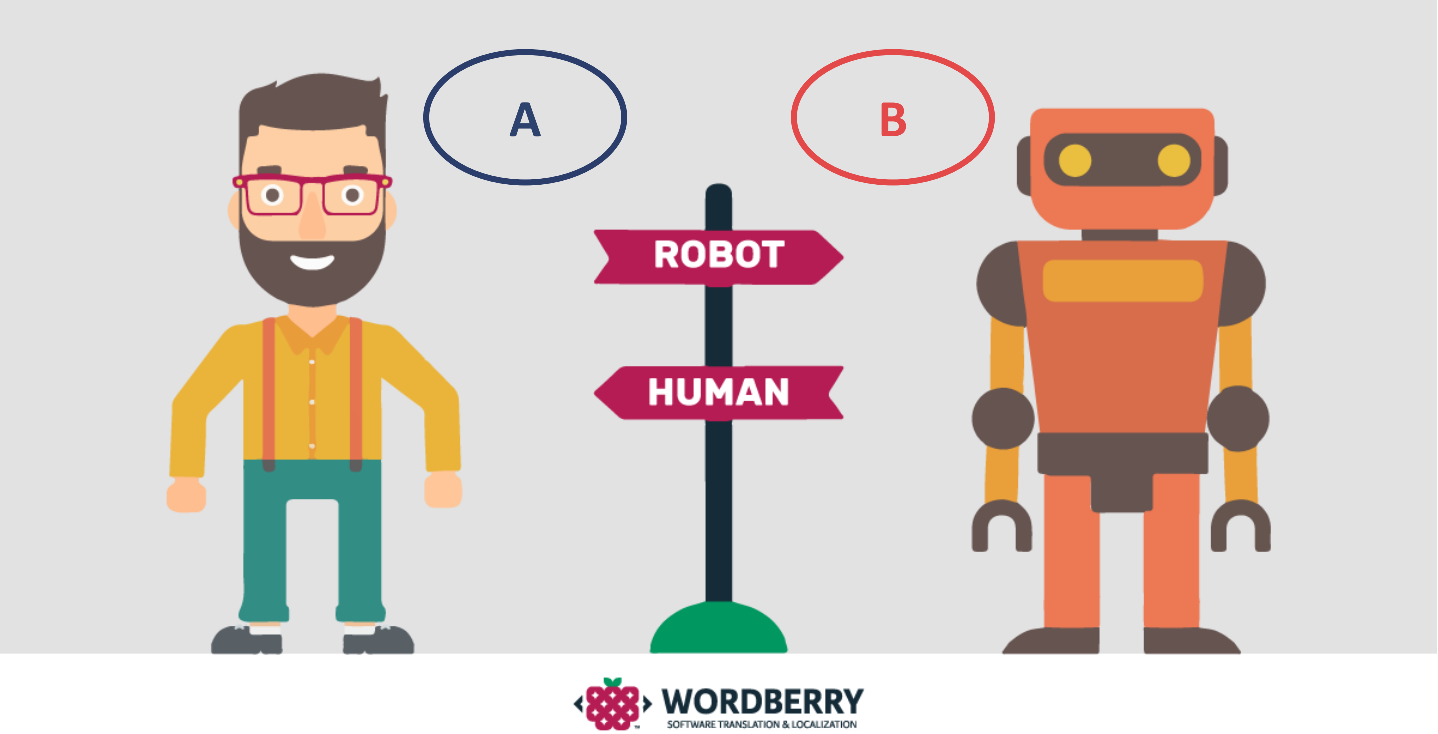
Residual Network (2016)
- 1st place in 2015
- By MSRA
- > 100 layers CNNs via identity skip connections

AlexNet (2012)
- 1st place in 2012
- 8-layer CNN
- GPU acceleration for training
- Dropout and ReLU

ZF-Net (2013)
- 3rd place in 2013
- By Zeiler & Fergus
- A variant of AlexNet

Back Side of The Neural Network Success

- Nowadays, **Convolutional Neural Network** shows impressive performance
  - The problem is that a neural network is highly vulnerable to a small perturbation of an input
  - In other words, the **answer of machine** is different from the **answer of human**

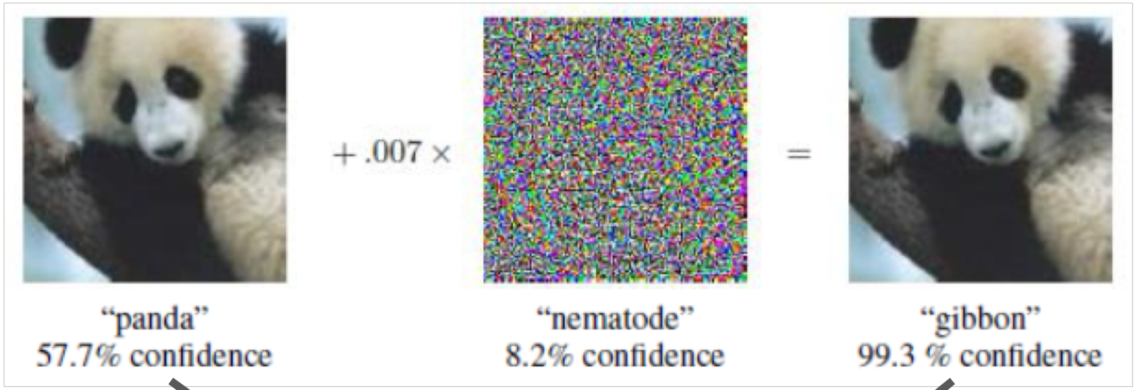


*source: https://wordberry.com/choosing-human-vs-machine-website-translation/

What is The Adversarial Example?

- Nowadays, **Convolutional Neural Network** shows impressive performance
  - The problem is that a neural network is highly vulnerable to a small perturbation of an input
  - In other words, the **answer of machine** is different from the **answer of human**

- Several machine learning models, including state-of-the-art neural networks, **misclassify** examples that are modified from clean data by imperceptible perturbations



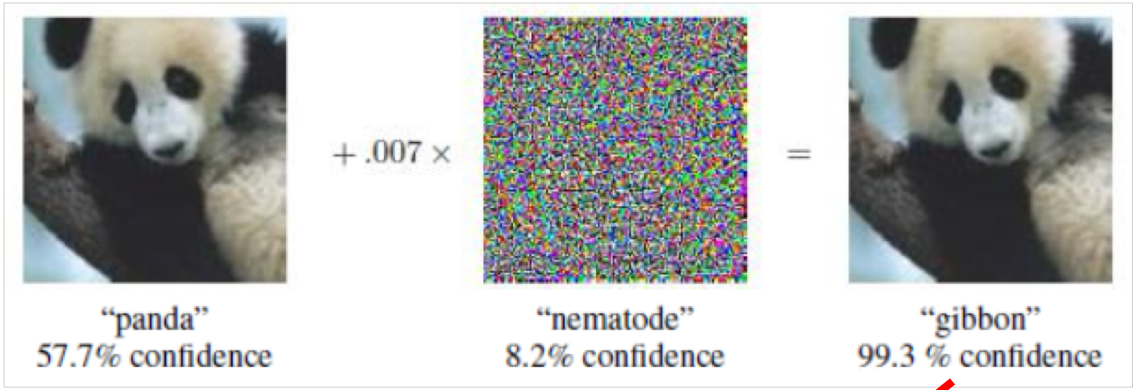
A Human eye can not distinguish them

*source: I. Goodfellow et al., EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES, In ICLR, 2015

What is The Adversarial Example?

- Nowadays, **Convolutional Neural Network** shows impressive performance
  - The problem is that a neural network is highly vulnerable to a small perturbation of an input
  - In other words, the **answer of machine** is different from the **answer of human**

- Several machine learning models, including state-of-the-art neural networks, **misclassify** examples that are modified from clean data by imperceptible perturbations

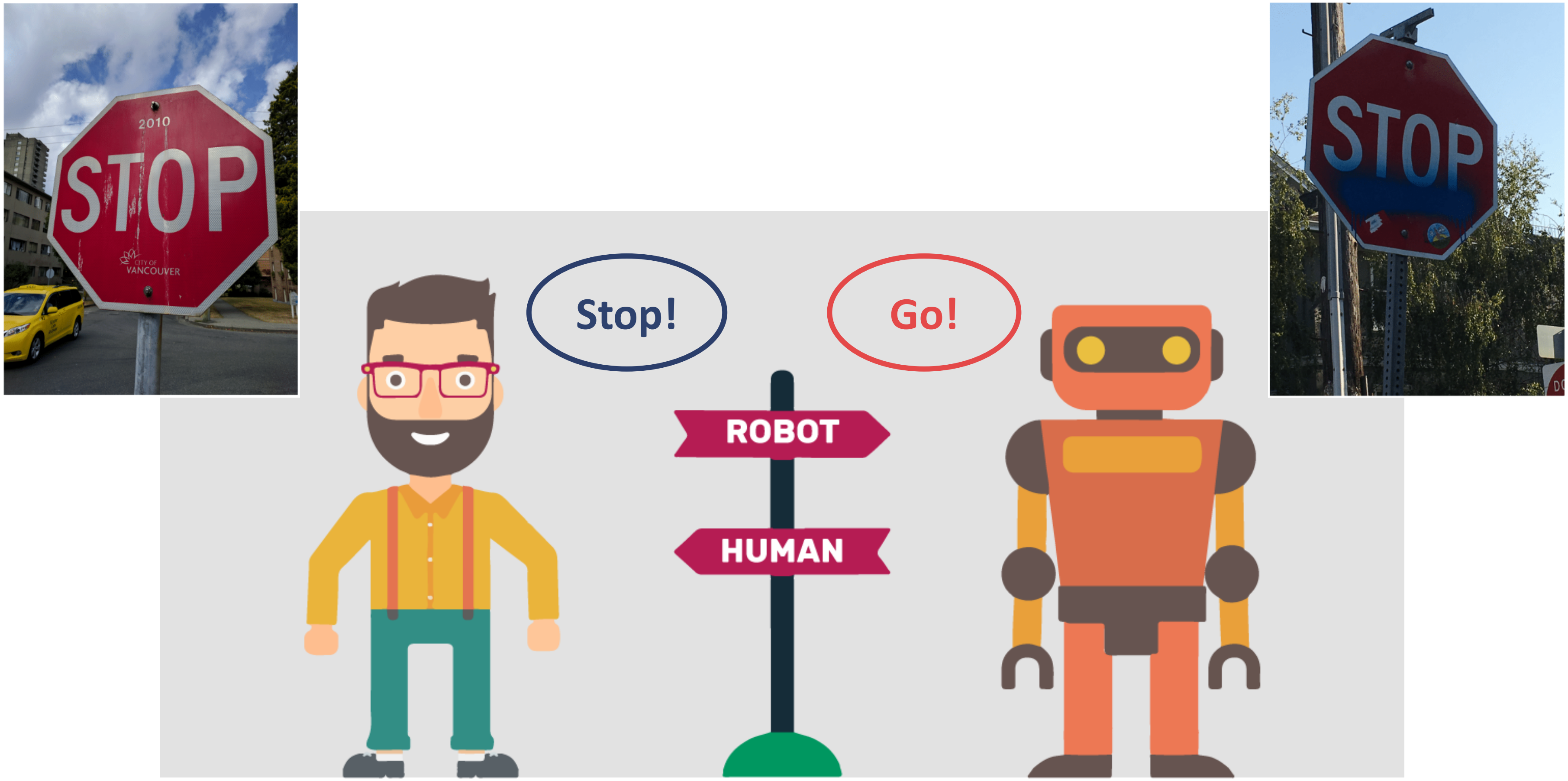


It is called an adversarial example!

*source: I. Goodfellow et al., EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES, In ICLR, 2015

Threat of The Adversarial Example

- Adversarial examples raise issues that are critical to the safety of AI in the real world
  - e.g. An autonomous vehicle may misclassify graffiti stop signs

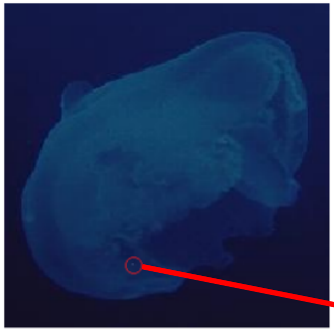


WORDBERRY
SOFTWARE TRANSLATION & LOCALIZATION

*source: K. Eykholt et al., Robust Physical-World Attacks on Deep Learning Visual Classification, In CVPR, 2018

Threat of The Adversarial Example

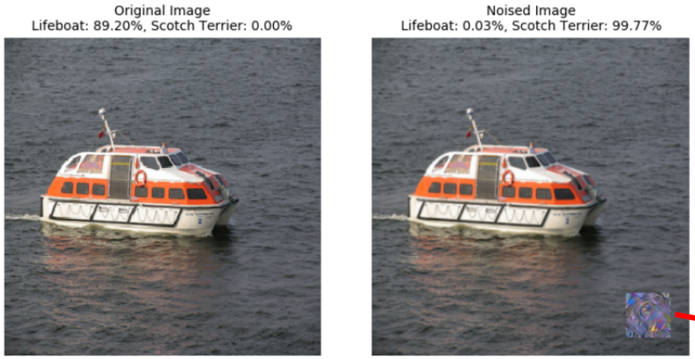
- There are various types of adversarial perturbations
  - Adversarial perturbations can be constructed in *local regions*



Jellyfish

Bathing tub(21.18%)

Only *one pixel* perturbed



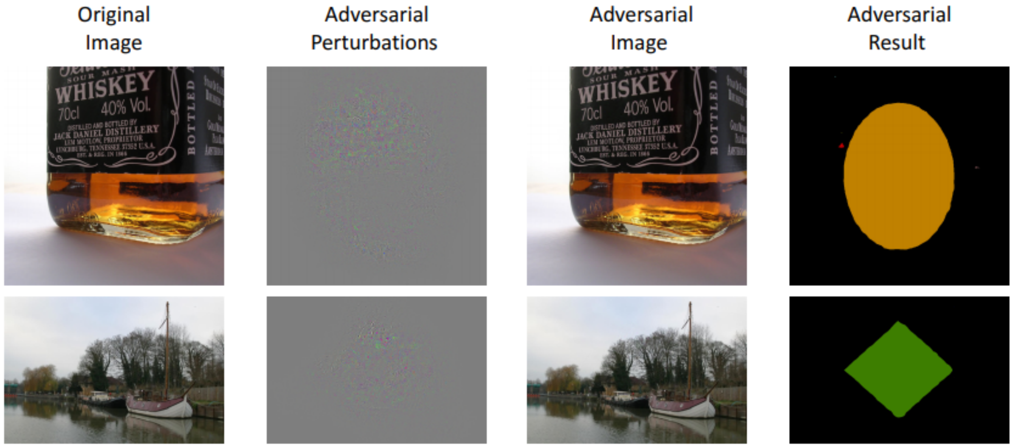
Original Image
Lifeboat: 89.20%, Scotch Terrier: 0.00%

Noised Image
Lifeboat: 0.03%, Scotch Terrier: 99.77%

Lifeboat (89.2%) → Scotch Terrier (99.8%)

Away from *the main object*

- For segmentation task, adversarial perturbation could control to generate geometric patterns



Original Image

Adversarial Perturbations

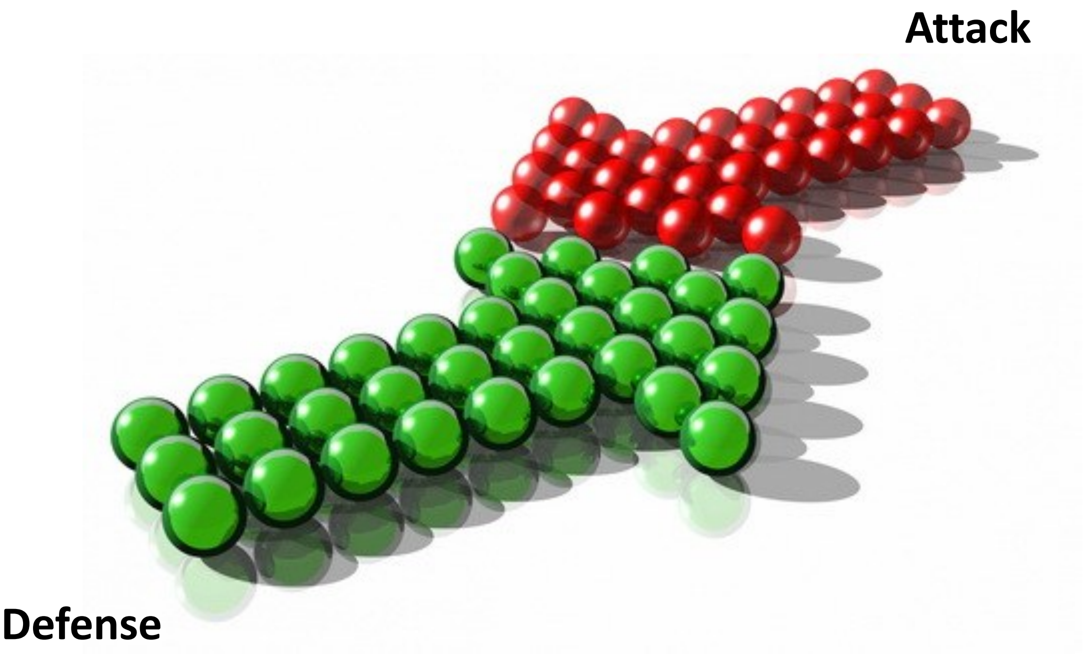
Adversarial Image

Adversarial Result

*source: J. Su et al., One pixel attack for fooling deep neural networks, arXiv, 2017
D. Karmon et al., LaVAN: Localized and Visible Adversarial Noise, In ICML, 2018
C. Xie et al., Adversarial Examples for Semantic Segmentation and Object Detection, In ICCV, 2017

Overview: Studies on The Adversarial Example

- Studies on the adversarial example are divided into
  - Adversarial **Attack**
    - How to find a perturbation that generate adversarial example
  - Adversarial **Defense**
    - How to prevent a perturbation that generate adversarial example



*source: https://gwynteatro.wordpress.com/2011/10/30/ambiguity-and-contradiction-leadership-certainties

Table of Contents

1. Introduction
   - What is the adversarial example?
   - Threat of the adversarial example

2. **Adversarial attack**
   - White-box attack
   - Black-box attack

3. Adversarial Defense
   - Adversarial training
   - Input pre-processing
   - Robust network construction

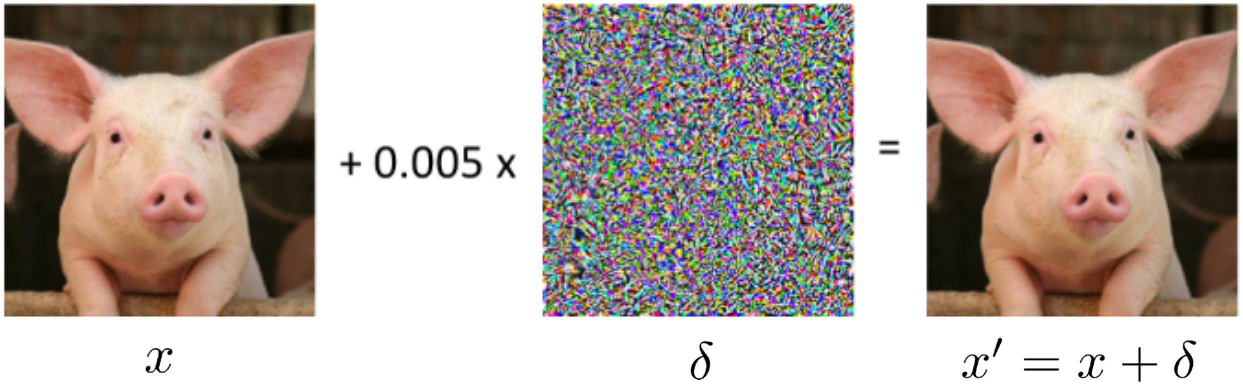
Formal Definitions: The Adversarial Example and Attack

- **Definition:** An input x' is called an **adversarial example** of an input x if x' satisfies

  (1) $\mathcal{D}(x, x')$ is small for some distance metric $\mathcal{D}$

  (2) $c(x') \neq c^*(x)$ where $c(\cdot)$ and $c^*(\cdot)$ denote the prediction and true label

- **Definition: Adversarial attack** is a method of finding adversarial perturbations δ that satisfies $c(x+\delta) \neq c^*(x)$
  - The smaller the size of δ, the better the adversarial attack method
    - Finding the smallest perturbation δ is a major challenge

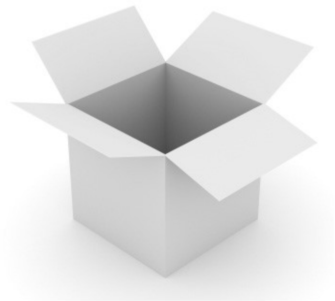


Adversarial Attack: Threat Model

- How to find the adversarial perturbations?
  - **Random perturbation** is the weakest attack method
  - What information is available?

- **White-box Model**
  - Adversary, who creates an adversarial example, has full knowledge of the neural network classifiers
  - e.g. model parameters, network architecture, training procedure, …

- **Black-box Model**
  - Adversary has no knowledge of the neural network classifier



white-box

black-box

*source: https://emperorsgrave.wordpress.com/2016/10/18/black-box/
https://reqtest.com/testing-blog/test-design-techniques-explained-1-black-box-vs-white-box-testing/

Table of Contents

1. **Introduction**
   - What is the adversarial example?
   - Threat of the adversarial example

2. **Adversarial attack**
   - White-box attack
   - Black-box attack

3. **Adversarial Defense**
   - Adversarial training
   - Input pre-processing
   - Robust network construction

White-Box: Fast Gradient Sign Method [I. Goodfellow et al., 2015]

- **Motivation:** How to change the network's prediction of an input?
  - In white-box setting, the gradients of a network is available
  - **Idea:** A perturbation maximizes the loss function $\ell(x, y_{\text{true}})$ would change the prediction
  - **Goal:** Solving the objective optimization below by using the linear approximation to ℓ

$$\underset{\delta : ||\delta||_\infty \leq \varepsilon}{\text{maximize}} \; \ell(x + \delta, y_{\text{true}})$$

  - **Method: Fast Gradient Sign Method**
    - The adversarial examples are computed by

$$x' = x + \varepsilon \cdot \text{sign}(\nabla_x \ell(x, y_{\text{true}}))$$

    - It can generate adversarial examples fast
    - Generated adversarial examples could have any label (untargeted)

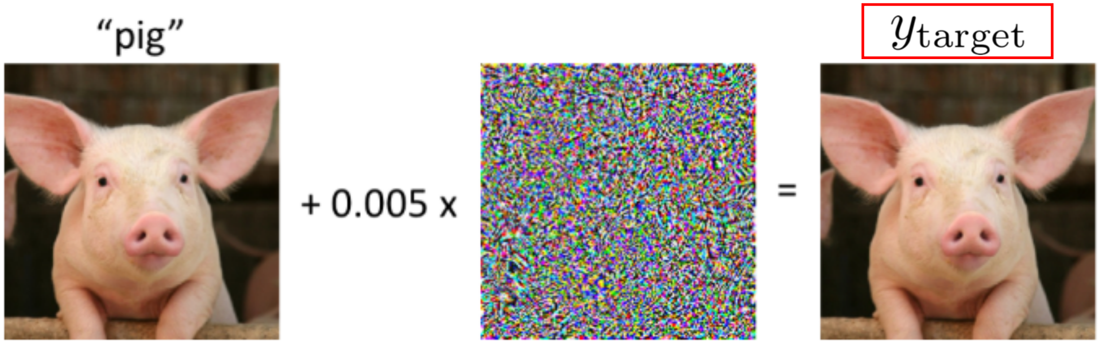
White-Box: Least-likely Class Method [A. Kurakin et al., 2017b]

- It is a variant of **Fast Gradient Sign Method**
- **Motivation:** How to control the prediction of the adversarial example?
  - **Idea:** A perturbation minimizes the loss function $\ell(x, y_{\text{target}})$ would change the prediction to the target label
  - **Goal:** Solving the objective optimization below by using the linear approximation to ℓ

$$\underset{\delta:\|\delta\|_{\infty}\leq\varepsilon}{\text{minimize}}\ \ell(x+\delta, y_{\text{target}})$$

  - **Method:** The adversarial examples are computed by

$$x' = x - \varepsilon \cdot \text{sign}(\nabla_x \ell(x, y_{\text{target}}))$$



White-Box: Basic Iterative Method [A. Kurakin et al., 2017b]

- It is a extension of **Fast Gradient Sign Method**
- **Motivation:** How to find an adversarial perturbation that is stronger than the perturbation from **Fast Gradient Sign Method**?
  - **Idea:** Extending “single-step” to “multi-step”
  - **Goal:** Solving the objective optimization of **Fast Gradient Sign Method** with the number of iterations

$$\underset{\delta:\|\delta\|_\infty\leq\varepsilon}{\text{maximize}}\ \ell(x+\delta, y_{\text{true}})$$

  - **Method:** The adversarial examples are computed by

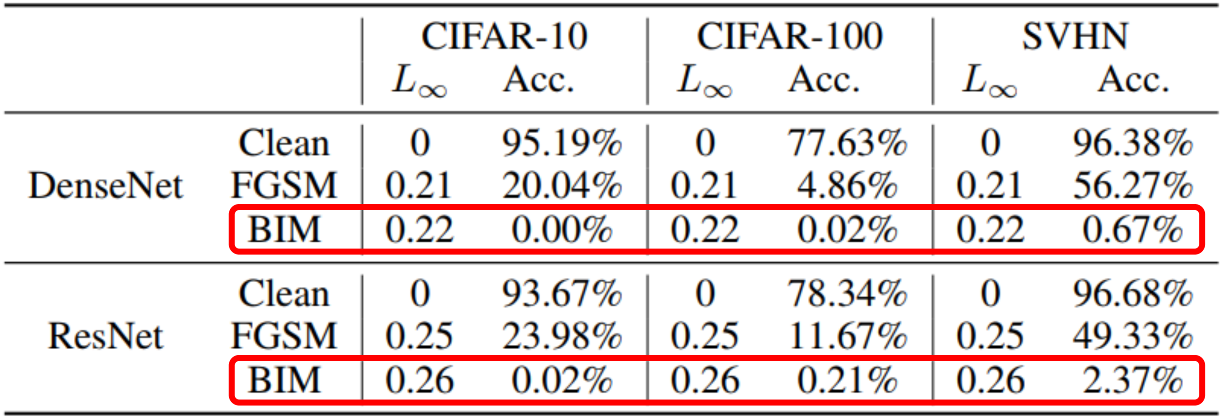
$$x_{t+1} = x_t + \alpha \cdot \text{sign}(\nabla_{x_t}\ell(x_t, y_{\text{true}}))$$

$$\text{where } \|x_0 - x_t\|_\infty \leq \varepsilon \text{ for all } t$$

  - **Least-likely Class Method** also can be extended to **Iterative least-likely class method**

White-Box: Basic Iterative Method [A. Kurakin et al., 2017b]

- **Experimental Results**
  - Comparison between **Fast Gradient Sign**, **Basic Iterative Method**
    - Mean of perturbation and classification accuracy [K. Lee et al., 2018]

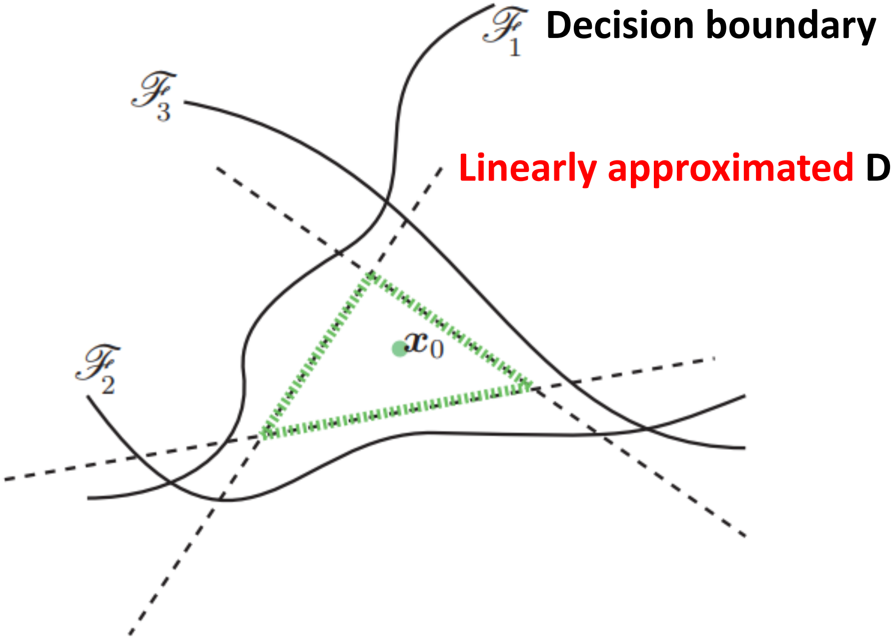


		CIFAR-10		CIFAR-100		SVHN	
		L_∞	Acc.	L_∞	Acc.	L_∞	Acc.
DenseNet	Clean	0	95.19%	0	77.63%	0	96.38%
	FGSM	0.21	20.04%	0.21	4.86%	0.21	56.27%
	BIM	0.22	0.00%	0.22	0.02%	0.22	0.67%
ResNet	Clean	0	93.67%	0	78.34%	0	96.68%
	FGSM	0.25	23.98%	0.25	11.67%	0.25	49.33%
	BIM	0.26	0.02%	0.26	0.21%	0.26	2.37%

- **Projected Gradient Method** [A. Madry et al., 2018]
  - It is a variant of **Basic Iterative Method**
  - **Motivation:** Sometimes, **Basic Iterative Method** falls into local maxima and it does not generate proper adversarial examples
  - **Idea:** Adding random initialization
  - **Method:** Generating a lot of randomly initialized input by adding the random noises before to compute **Basic Iterative Method**

White-Box: DeepFool Method [S. Moosavi-Dezfooli et al., 2016]

- **Motivation:** Perturbing an input to a decision boundary direction would change the prediction of input
  - **Idea:** The smallest adversarial perturbation has a direction to the closest decision boundary
  - **Goal:** Finding a direction to the closest decision boundary by using linear approximation to decision boundaries



Decision boundary

Linearly approximated Decision boundary

White-Box: DeepFool Method [S. Moosavi-Dezfooli et al., 2016]

- **Motivation:** Perturbing an input to a decision boundary direction would change the prediction of input
  - **Idea:** The smallest adversarial perturbation has a direction to the closest decision boundary
  - **Goal:** Finding a direction to the closest decision boundary by using linear approximation to decision boundaries
  - **Method:** Distance d from an input x to a decision boundary between y and y_{true} is computed by

$$d = \frac{|f_y(x) - f_{y_{\text{true}}}(x)|}{\left\| \nabla_x f_y(x) - \nabla_x f_{y_{\text{true}}}(x) \right\|_2}$$

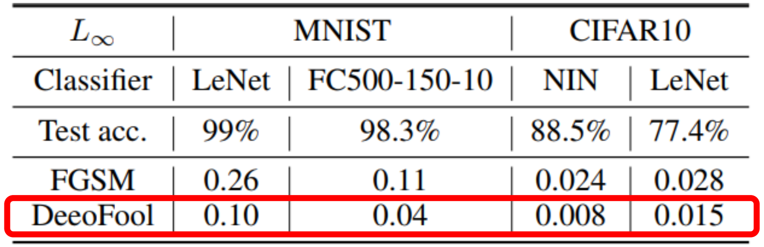
$$\text{where } f_i(\cdot) \text{ is a } i\text{-th logit output of the classifier } f(\cdot)$$

- Let $\widehat{d}$ be the smallest distance to decision boundary and $\widehat{y}$, y_{true} are the corresponding labels. Then the adversarial examples are computed by

$$x_{t+1} = x_t + \widehat{d}\,(f_{\widehat{y}}(x_t) - f_{y_{\text{true}}}(x_t))$$

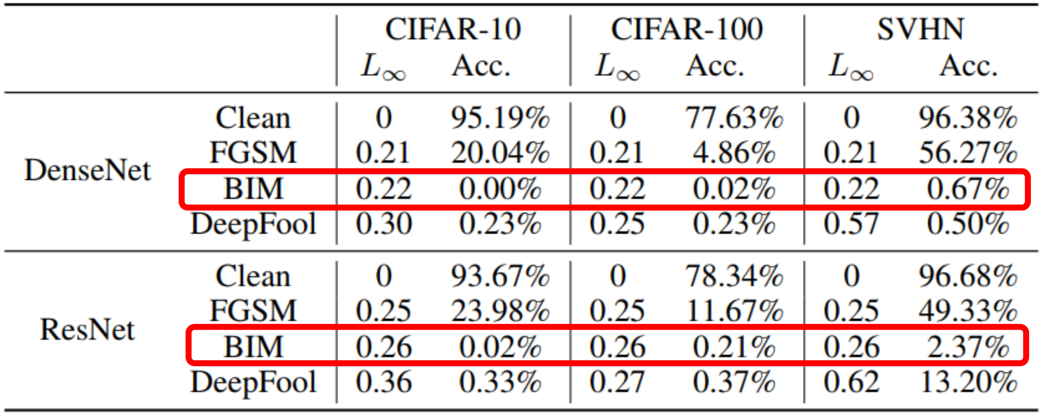
White-Box: DeepFool [S. Moosavi-Dezfooli et al., 2016]

- **Experimental Results**
  - Comparison between **Fast Gradient Sign**, **DeepFool Method**
    - Mean of perturbation among four different network



L_∞	MNIST		CIFAR10	
Classifier	LeNet	FC500-150-10	NIN	LeNet
Test acc.	99%	98.3%	88.5%	77.4%
FGSM	0.26	0.11	0.024	0.028
DeeoFool	0.10	0.04	0.008	0.015

    - **DeepFool Method** is a stronger method than **Fast Gradient Sign Method**
  - Comparison among **Fast Gradient Sign**, **Basic Iterative, DeepFool Method**
    - Mean of perturbation and classification accuracy [K. Lee et al., 2018]



		CIFAR-10		CIFAR-100		SVHN	
		L_∞	Acc.	L_∞	Acc.	L_∞	Acc.
DenseNet	Clean	0	95.19%	0	77.63%	0	96.38%
	FGSM	0.21	20.04%	0.21	4.86%	0.21	56.27%
	BIM	0.22	0.00%	0.22	0.02%	0.22	0.67%
	DeepFool	0.30	0.23%	0.25	0.23%	0.57	0.50%
ResNet	Clean	0	93.67%	0	78.34%	0	96.68%
	FGSM	0.25	23.98%	0.25	11.67%	0.25	49.33%
	BIM	0.26	0.02%	0.26	0.21%	0.26	2.37%
	DeepFool	0.36	0.33%	0.27	0.37%	0.62	13.20%

    - **DeepFool Method** is not a stronger method than **Basic Iterative Method**

White-Box: Carlini-Wagner Method [N. Carlini et al., 2017a]

- **Motivation:** Large-scale perturbation should change the prediction, but above attacks are sometimes not successful with large-scale perturbation
  - **Idea:** Finding the smallest perturbation subject to the perturbation makes adversarial example
  - **Goal:** Minimizing the scale of adversarial perturbation $||\delta||$ subject to the perturbation δ makes the input to be an adversarial example

$$\underset{\delta : c(x+\delta)=y_{\text{target}}}{\text{minimize}} \quad ||\delta||_2$$

  - **Method:** Applying the Lagrangian relaxation to the objective with a function g that satisfying

$$g(x) = \left(\max_{i \neq \text{target}} f_i(x) - f_{y_{\text{target}}}(x) \right)^+$$

$$\text{where } f_i(\cdot) \text{ is a } i\text{-th logit output of the given classifier } f(\cdot)$$

$$\text{and } (e)^+ = \max(e, 0)$$

    - $g(x)$ has the minimum value 0 when x is the adversarial example

White-Box: Carlini-Wagner Method [N. Carlini et al., 2017a]

- **Motivation:** Large-scale perturbation should change the prediction, but above attacks are sometimes not successful with large-scale perturbation
  - **Idea:** Finding the smallest perturbation subject to the perturbation makes adversarial example
  - **Goal:** Minimizing the scale of adversarial perturbation $\|\delta\|$ subject to the perturbation δ makes the input to be an adversarial example

$$\underset{\delta : c(x+\delta)=y_{\text{target}}}{\text{minimize}} \quad \|\delta\|_2$$

  - **Method:** Solving the relaxed objective

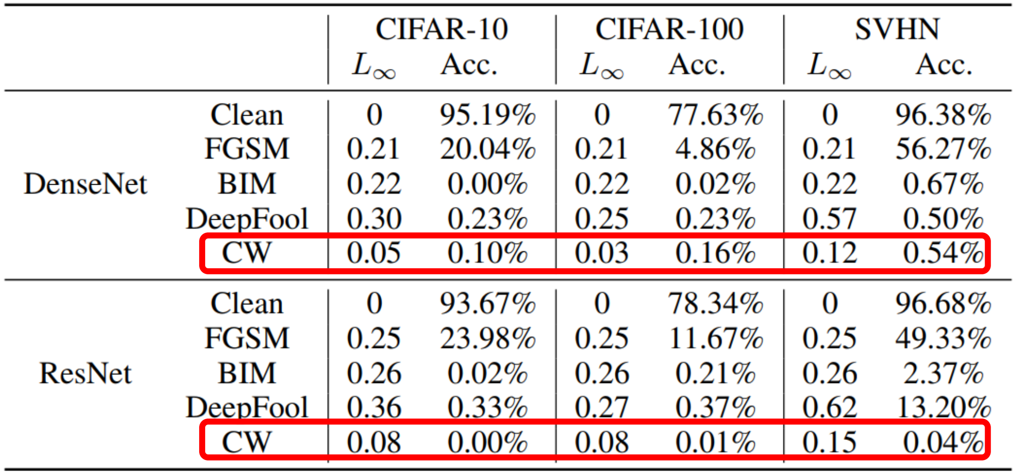
$$\underset{\delta}{\text{minimize}}\, \|\delta\|_2 + \alpha \cdot g(x+\delta)$$

α: hyper-parameter

    - Use Gradient Descent to solve the optimization

White-Box: Carlini-Wagner Method [N. Carlini et al., 2017a]

- **Experimental Results**
  - Comparison between among **Fast Gradient Sign**, **Basic Iterative, DeepFool, Carlini-Wagner Method**
    - Mean of perturbation and classification accuracy [K. Lee et al., 2018]



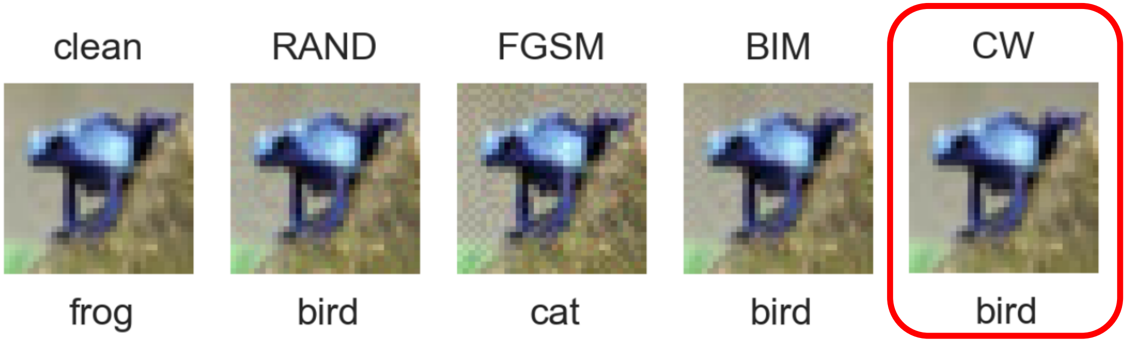
		CIFAR-10		CIFAR-100		SVHN	
		L_∞	Acc.	L_∞	Acc.	L_∞	Acc.
DenseNet	Clean	0	95.19%	0	77.63%	0	96.38%
	FGSM	0.21	20.04%	0.21	4.86%	0.21	56.27%
	BIM	0.22	0.00%	0.22	0.02%	0.22	0.67%
	DeepFool	0.30	0.23%	0.25	0.23%	0.57	0.50%
	CW	0.05	0.10%	0.03	0.16%	0.12	0.54%
ResNet	Clean	0	93.67%	0	78.34%	0	96.68%
	FGSM	0.25	23.98%	0.25	11.67%	0.25	49.33%
	BIM	0.26	0.02%	0.26	0.21%	0.26	2.37%
	DeepFool	0.36	0.33%	0.27	0.37%	0.62	13.20%
	CW	0.08	0.00%	0.08	0.01%	0.15	0.04%

- **Carlini-Wagner Method** find the smallest adversarial perturbations among the several attacks
- 10× slower than **Basic Iterative Method**

White-Box: Carlini-Wagner Method [N. Carlini et al., 2017a]

- **Experimental Results**
  - Comparison between among **Fast Gradient Sign**, **Basic Iterative, DeepFool, Carlini-Wagner Method**
    - Mean of perturbation and classification accuracy [K. Lee et al., 2018]

  - Images of adversarial examples among the several attacks [Y. Song et al., 2018]



It is the most similar to clean image

*source: Y. Song et al., PIXELDEFEND: LEVERAGING GENERATIVE MODELS TO UNDERSTAND AND DEFEND AGAINST ADVERSARIAL EXAMPLES, In ICLR, 2018

Table of Contents

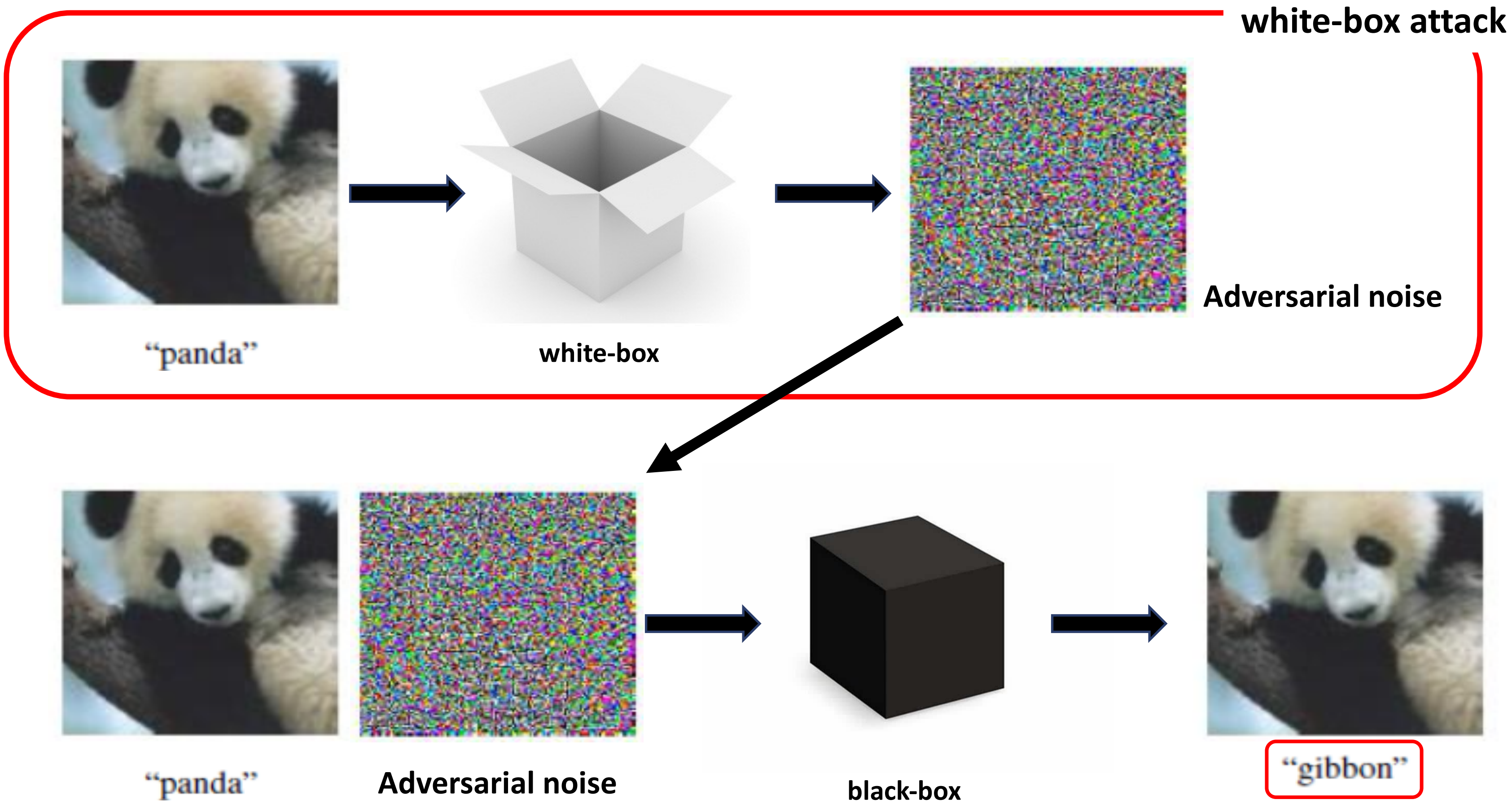
1. Introduction
   - What is the adversarial example?
   - Threat of the adversarial example

2. **Adversarial attack**
   - White-box attack
   - Black-box attack

3. Adversarial Defense
   - Adversarial training
   - Input pre-processing
   - Robust network construction

Black-Box: Transferability of Adversarial Example

- Some of perturbations make adversarial example among **different network architecture**



white-box attack

"panda"

white-box

Adversarial noise

"panda"

Adversarial noise

black-box

"gibbon"

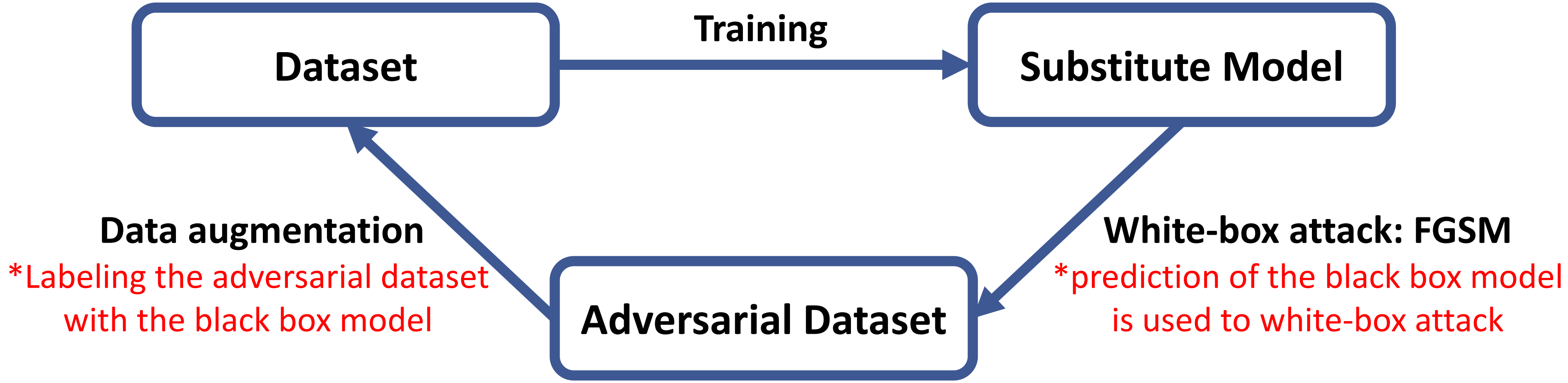
*source: I. Goodfellow et al., EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES, In ICLR, 2015

Black-Box: The Local Substitute Model [N. Papernot et al., 2017]

- **Motivation:** The transferability of adversarial example allow to attack a black-box model
  - **Idea:** Finding an adversarial example via white-box attack on the local substitute model
  - **Goal:** Training a local substitute model via FGSM-based adversarial dataset augmentation
    - FGSM-based adversarial examples are computed to change the prediction of the black-box model

$$x' = x + \varepsilon \cdot \mathrm{sign}(\nabla_x \ell(x, y_{\mathrm{pred}}))$$

  - **Method:**



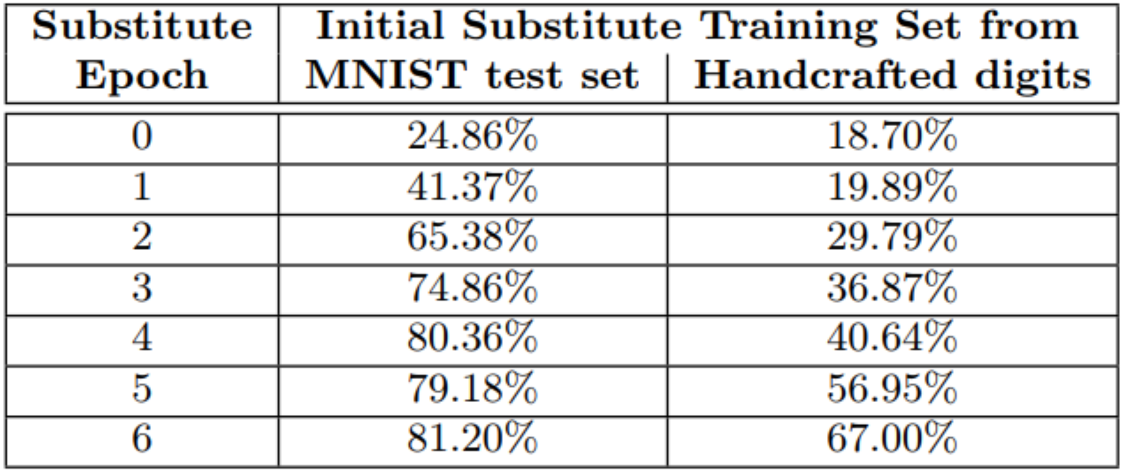
*Labeling the adversarial dataset with the black box model

*prediction of the black box model is used to white-box attack

Black-Box: The Local Substitute Model [N. Papernot et al., 2017]

- **Experimental Results**
  - The local substitute model training
    - Initial training dataset: subset of **MNIST**, **Handcrafted** set
    - **Handcrafted** set is used to ensure the results do not stem from the similarities between the MNIST test and training sets

  - Accuracies of the local substitute models

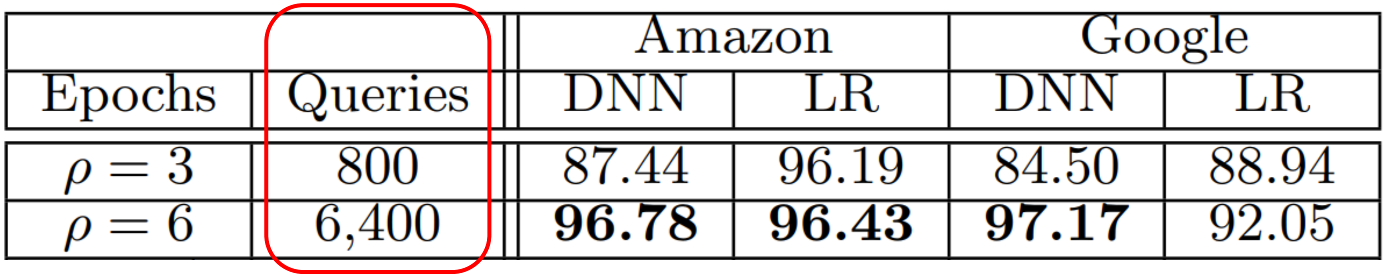


Substitute Epoch	Initial Substitute Training Set from	
	MNIST test set	Handcrafted digits
0	24.86%	18.70%
1	41.37%	19.89%
2	65.38%	29.79%
3	74.86%	36.87%
4	80.36%	40.64%
5	79.18%	56.95%
6	81.20%	67.00%

Black-Box: The Local Substitute Model [N. Papernot et al., 2017]

- **Experimental Results**
  - Black-box attack to the Amazon and Google Oracle
  - Two types of architecture:
    - **DNN**: Deep Neural Network
    - **LR**: Logistic Regression

Misclassification rates (%)



		Amazon		Google	
Epochs	Queries	DNN	LR	DNN	LR
$\rho = 3$	800	87.44	96.19	84.50	88.94
$\rho = 6$	6,400	**96.78**	**96.43**	**97.17**	92.05

Number of queries to train the local substitute model

Black-Box: Ensemble Based Method [Y. Liu et al., 2017]

- **Motivation:** Adversarial examples from the substitute model are sometimes weak
  - **Idea:** White-box attack to an ensemble of the substitute models could generate the strong adversarial example
  - **Goal:** Finding the adversarial examples from the ensemble model
  - **Method:** Consider k number of substitute models and let $J_1, \ldots, J_k$ be their softmax outputs. Then for given (x, y_{true}), the objective is follow:

$$\underset{\delta}{\text{minimize}} \; -\log\left(1 - \left(\sum_{i=1}^{k} \alpha_i J_i(x+\delta)\right)_{\boxed{y_{\text{true}}}}\right) + \lambda d(x, x+\delta)$$

y_{true}-th softmax output of the ensemble model

where α_i is a ensemble weight with $\sum_{i=1}^{k} \alpha_i = 1$,

$$d(x, x') = \sqrt{(\textstyle\sum_i (x'_i - x_i)^2 / N)}, \qquad x, x' \in \mathbb{R}^N$$

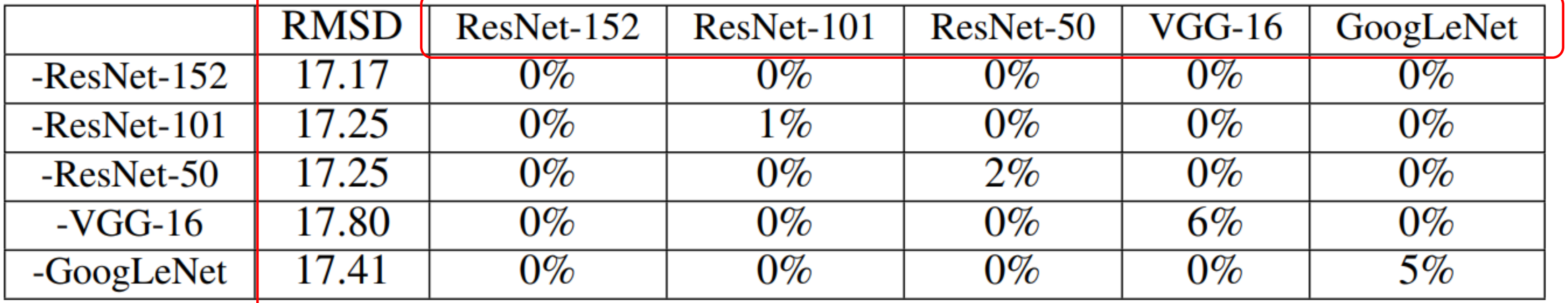
λ: hyper-parameter

  - The metric $d(x, x')$ is called the Root Mean Square Deviation (RMSD)

Black-Box: Ensemble Based Method [Y. Liu et al., 2017]

- **Experimental Results**
  - Ensemble of modern architecture DNNs
    - “-model A” means an ensemble without “model A”
  - RMSD: Root Mean Square Deviation of adversarial perturbations

Black-box models

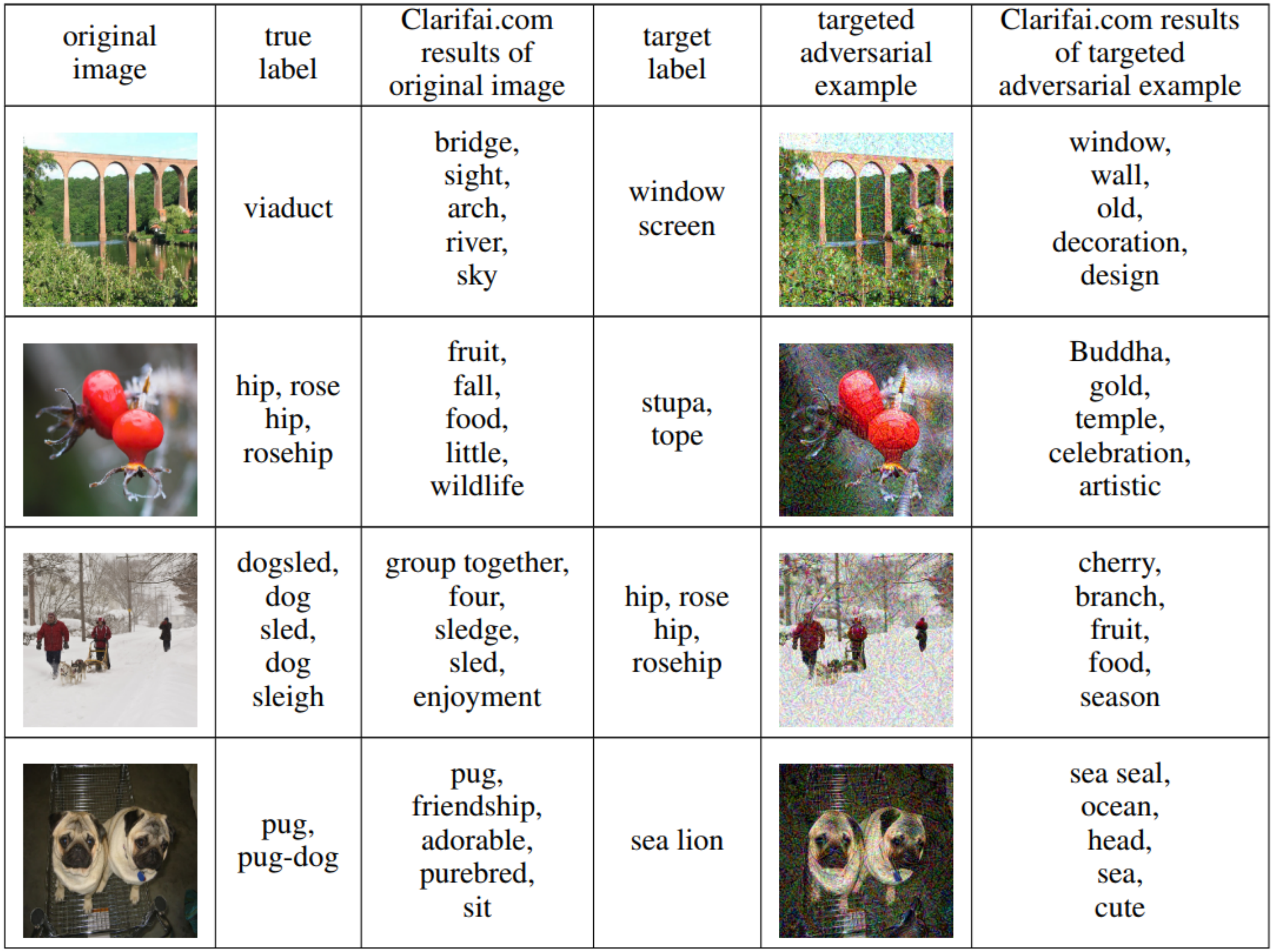


	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
-ResNet-152	17.17	0%	0%	0%	0%	0%
-ResNet-101	17.25	0%	1%	0%	0%	0%
-ResNet-50	17.25	0%	0%	2%	0%	0%
-VGG-16	17.80	0%	0%	0%	6%	0%
-GoogLeNet	17.41	0%	0%	0%	0%	5%

Adversarial examples from the ensemble models via white-box attack

Black-Box: Ensemble Based Method [Y. Liu et al., 2017]

- **Experimental Results**
  - **Black-box** attack on Clarifai.com (commercial black-box image classification system)



original image	true label	Clarifai.com results of original image	target label	targeted adversarial example	Clarifai.com results of targeted adversarial example
	viaduct	bridge, sight, arch, river, sky	window screen		window, wall, old, decoration, design
	hip, rose hip, rosehip	fruit, fall, food, little, wildlife	stupa, tope		Buddha, gold, temple, celebration, artistic
	dogsled, dog sled, dog sleigh	group together, four, sledge, sled, enjoyment	hip, rose hip, rosehip		cherry, branch, fruit, food, season
	pug, pug-dog	pug, friendship, adorable, purebred, sit	sea lion		sea seal, ocean, head, sea, cute

Table of Contents

1. **Introduction**
   - What is adversarial example?
   - Overview

2. **Adversarial attack**
   - White-box attack
   - Black-box attack

3. **Adversarial Defense**
   - Adversarial training
   - Input pre-processing
   - Robust network construction

Adversarial Defense

- How to defend such adversarial attack?
  - 3 types of **Defense scenarios**
    - **Adversarial training**: Re-train with adversarial examples
    - **Input pre-processing**: Pre-process an input to make a clean example
    - **Robust network construction**: Construct a new network to be robust

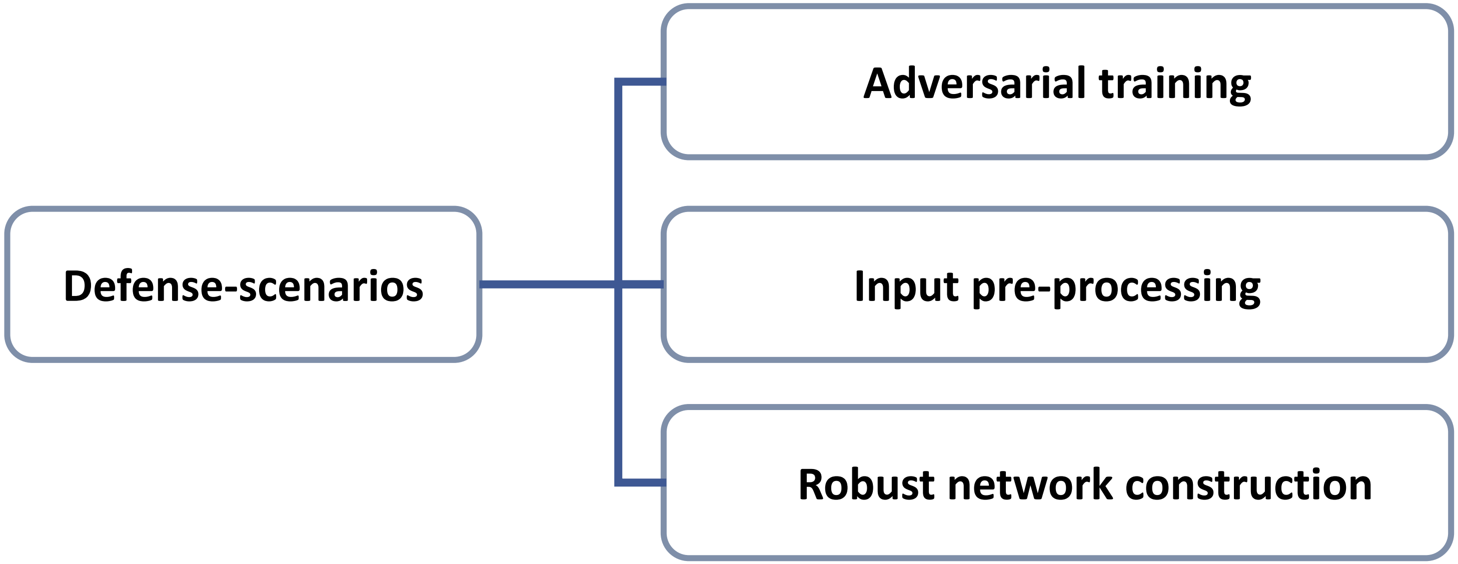


Table of Contents

1. **Introduction**
    - What is adversarial example?
    - Overview

2. **Adversarial attack**
    - White-box attack
    - Black-box attack

3. **Adversarial Defense**
    - Adversarial training
    - Input pre-processing
    - Robust network construction

Defense-Scenarios: FGSM Adversarial Training [I. Goodfellow et al., 2015]

- **Motivation:** Training a model with adversarial examples would make the model to be robust on adversarial examples
  - **Idea:** Re-training a model with adversarial examples which are combined to true labels
  - **Goal:** Solving the min-max objective optimization below

$$\underset{\theta \in \Theta}{\text{minimize}}\, \mathbb{E}_{p(x,y)} \left[\max_{\delta : \|\delta\|_\infty \le \varepsilon} \ell(\theta; x + \delta, y) \right]$$

    - Adversarial examples could substitutes the inner-maximization of the objective
  - **Question:** How to generate adversarial examples?
    - **FGSM:** Fast and simple (single-step) method
    - **BIM:** Slow, but strong (multi-step) method
    - **Carlini-Wagner Method:** Extremely slow to use adversarial training
      - Not appropriate to adversarial training

Defense-Scenarios: FGSM Adversarial Training [I. Goodfellow et al., 2015]

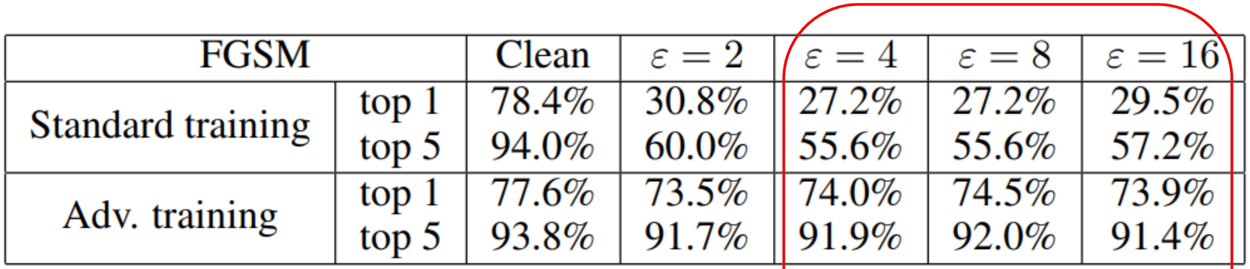
- **Method:** Generate adversarial examples via **FGSM** during the training
  - Then total loss function is

$$\widehat{\ell}(\theta; x, y) = \alpha\ell\left((\theta; x, y)\right) + (1-\alpha)\ell\left(\theta; \boxed{x + \varepsilon\text{sign}\left(\nabla_x \ell(\theta; x, y)\right)}, y\right)$$

$$\alpha\text{: hyper-parameter}$$

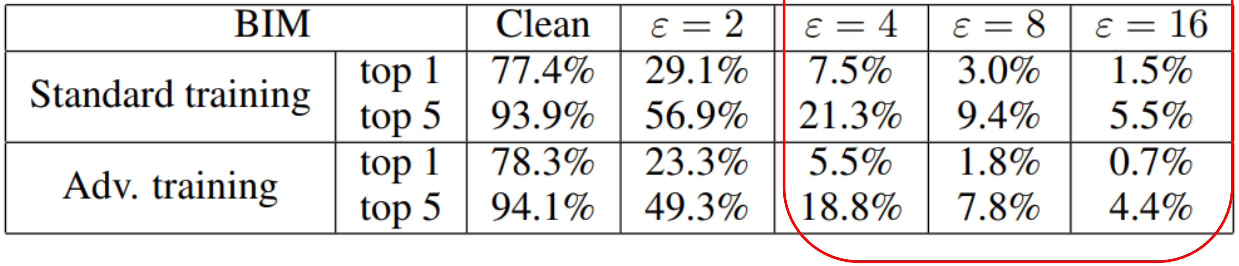
FGSM example

- **Experimental Results**
  - Robust in FGSM-attack, but not in BIM-attack [A. Kurakin et al., 2017b]



FGSM		Clean	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = 16$
Standard training	top 1	78.4%	30.8%	27.2%	27.2%	29.5%
	top 5	94.0%	60.0%	55.6%	55.6%	57.2%
Adv. training	top 1	77.6%	73.5%	74.0%	74.5%	73.9%
	top 5	93.8%	91.7%	91.9%	92.0%	91.4%

noise scale $\varepsilon \in [0, 255]$



BIM		Clean	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = 16$
Standard training	top 1	77.4%	29.1%	7.5%	3.0%	1.5%
	top 5	93.9%	56.9%	21.3%	9.4%	5.5%
Adv. training	top 1	78.3%	23.3%	5.5%	1.8%	0.7%
	top 5	94.1%	49.3%	18.8%	7.8%	4.4%

*source: I. Goodfellow et al., EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES, In ICLR, 2015
A. Kurakin et al., ADVERSARIAL MACHINE LEARNING AT SCALE, In ICLR, 2017

Defense-Scenarios: PGD Adversarial Training [A. Madry et al., 2018]

- **Motivation:** Adversarial training against a strong adversarial attacks (e.g. **BIM**) makes the network be a constant
  - It sacrifices the classification performance
  - **Idea:** Large capacity network would not sacrifice the classification performance
  - **Goal:** Solving the min-max objective optimization below

$$\underset{\theta \in \Theta}{\text{minimize}}\ \mathbb{E}_{p(x,y)} \left[\max_{\delta : \|\delta\|_\infty \leq \varepsilon} \ell(\theta; x + \delta, y) \right]$$

    - Substituting the inner-maximization of the objective via strong adversarial attack (**PGD**)
  - **Method:** Generate adversarial examples via **PGD** during the training
    - Then total loss function is

$$\widehat{\ell}(\theta; x, y) = \alpha \ell\left((\theta; x, y)\right) + (1 - \alpha)\ell\left(\theta; \boxed{x'}, y\right)$$

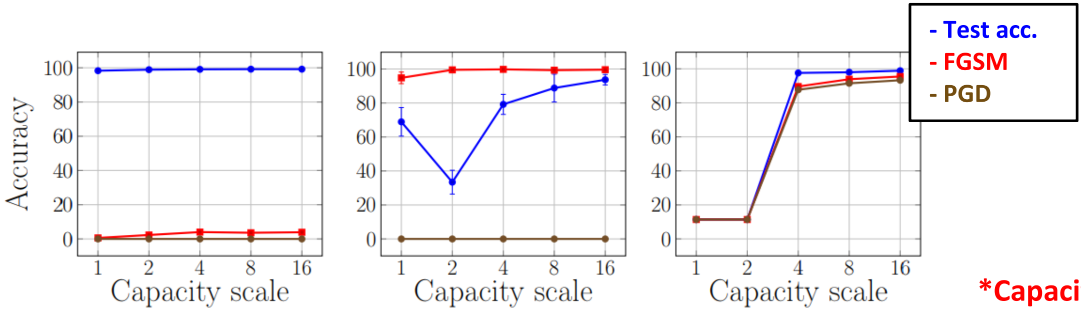
$$\alpha:\ \text{hyper-parameter}$$

PGD example

*source: A. Madry et al., Towards Deep Learning Models Resistant to Adversarial Attacks, In ICLR, 2018

Defense-Scenarios: PGD Adversarial Training [A. Madry et al., 2018]

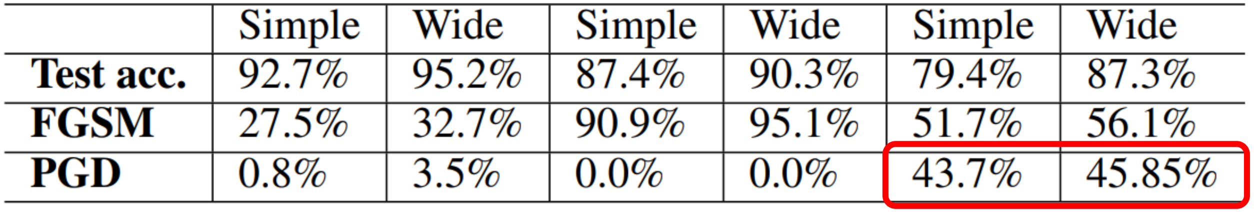
- **Experimental Results**
  - PGD adversarial training with small and large capacity
  - MNIST



(a) Standard Training **(b) FGSM Adv. Training** **(c) PGD Adv. Training**

***Capacity scale 8 means 8× wider network**

- CIFAR10



	(a) Standard Training		(b) FGSM Adv. Training		(c) PGD Adv. Training	
	Simple	Wide	Simple	Wide	Simple	Wide
Test acc.	92.7%	95.2%	87.4%	90.3%	79.4%	87.3%
FGSM	27.5%	32.7%	90.9%	95.1%	51.7%	56.1%
PGD	0.8%	3.5%	0.0%	0.0%	43.7%	45.85%

***Simple: ResNet**
Wide: ResNet with 10× wider

Defense-Scenarios: Wasserstein Adversarial Training [A. Sinha et al., 2018]

- **Motivation:** Adversarial examples substitute the inner-maximization objective of adversarial training,

$$\underset{\theta\in\Theta}{\text{minimize}}\ \mathbb{E}_{p(x,y)}\left[\boxed{\max_{\delta:\|\delta\|_{\infty}\leq\varepsilon}\ \ell(\theta;x+\delta,y)}\right]$$

  - But, the **gap** between the worst case (inner-maximization objective) and the adversarial example depends on adversarial attack methods
  - It can not be ensured that white-box attacks converges to the worst case
    - Inner-maximization objective is generally non-concave
  - **Idea:** Making a concave objective by applying the Lagrangian relaxation to the inner-maximization objective
  - The adversarial-training objective is newly defined as

$$\underset{\theta\in\Theta}{\text{minimize}}\ \sup_{P\in\mathcal{P}}\ \mathbb{E}_P\left[\ell(\theta;Z)\right]$$

where $Z=(X,Y)\sim P_0$ with the training data X and the label Y

and $\mathcal{P}$ is a class of distribution around the data-generating distribution P_0

Defense-Scenarios: Wasserstein Adversarial Training [A. Sinha et al., 2018]

- $\mathcal{P}$ can be written as the Wasserstein metric W_c

$$\mathcal{P} = \{P : W_c(P, P_0) \leq \rho\} \quad \text{where a hyper-parameter } \rho \geq 0$$

  - Wasserstein metric W_c between distributions P and Q is defined as

$$W_c(P, Q) := \inf_{M \in \prod(P,Q)} \mathbb{E}_M \left[c(Z, Z')\right]$$

$$\text{where } c(z, z') := \|x - x'\|_p^2 + \infty \cdot 1_{y \neq y'},\ 1_y \text{ is a indicator function}$$

  - The objective cover the worst-case with some $P' \in W_c(P', P_0) \leq \rho$

$$\underset{\theta \in \Theta}{\text{minimize}} \sup_{P \in W_c(P, P_0) \leq \rho} \mathbb{E}_P[\ell(\theta; Z)]$$

  - Lagrangian relaxation for a fixed parameter γ induces the objective to

➡ $$\underset{\theta \in \Theta}{\text{minimize}} \sup_{P} \mathbb{E}_P \left[\ell(\theta; Z) - \gamma W_c(P, P_0)\right]$$

Defense-Scenarios: Wasserstein Adversarial Training [A. Sinha et al., 2018]

- Furthermore, theorem [J. Blanchet et al., 2016] induces the relaxed objective to

➡ $$\underset{\theta \in \Theta}{\text{minimize}}\, \mathbb{E}_{P_0}\left[\sup_{z \in \mathcal{Z}} \{\ell(\theta; z) - \gamma c(z, z_0)\}\right]$$

  - It is the final objective of Wasserstein adversarial training [A. Shinha et al., 2018]

- **Goal:** Solving the final objective optimization below

$$\underset{\theta \in \Theta}{\text{minimize}}\, \mathbb{E}_{P_0}\left[\sup_{z \in \mathcal{Z}} \{\ell(\theta; z) - \gamma c(z, z_0)\}\right]$$

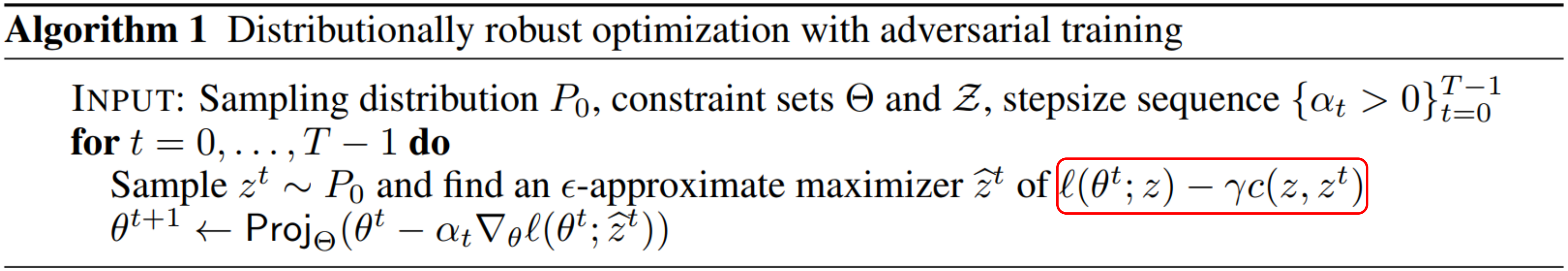
$$\text{where } c(z, z') := \|x - x'\|_p^2 + \infty \cdot 1_{y \neq y'},\ 1_y \text{ is a indicator function}$$

- The objective of previous adversarial training is follow:

$$\underset{\theta \in \Theta}{\text{minimize}}\, \mathbb{E}_{p(x,y)}\left[\max_{\delta : \|\delta\|_\infty \leq \varepsilon} \ell(\theta; x + \delta, y)\right]$$

Defense-Scenarios: Wasserstein Adversarial Training [A. Sinha et al., 2018]

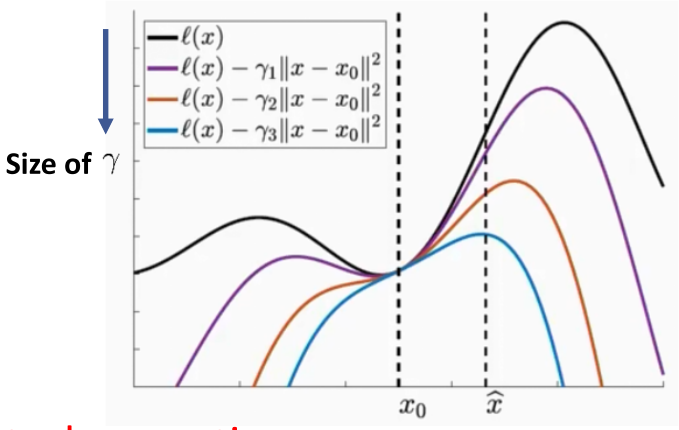
- **Method:** Wasserstein adversarial training algorithm



Algorithm 1 Distributionally robust optimization with adversarial training

```
INPUT: Sampling distribution $P_0$, constraint sets $\Theta$ and $\mathcal{Z}$, stepsize sequence $\{\alpha_t > 0\}_{t=0}^{T-1}$
for $t = 0, \ldots, T-1$ do
    Sample $z^t \sim P_0$ and find an $\epsilon$-approximate maximizer $\widehat{z}^t$ of $\boxed{\ell(\theta^t; z) - \gamma c(z, z^t)}$
    $\theta^{t+1} \leftarrow \mathrm{Proj}_\Theta(\theta^t - \alpha_t \nabla_\theta \ell(\theta^t; \widehat{z}^t))$
```

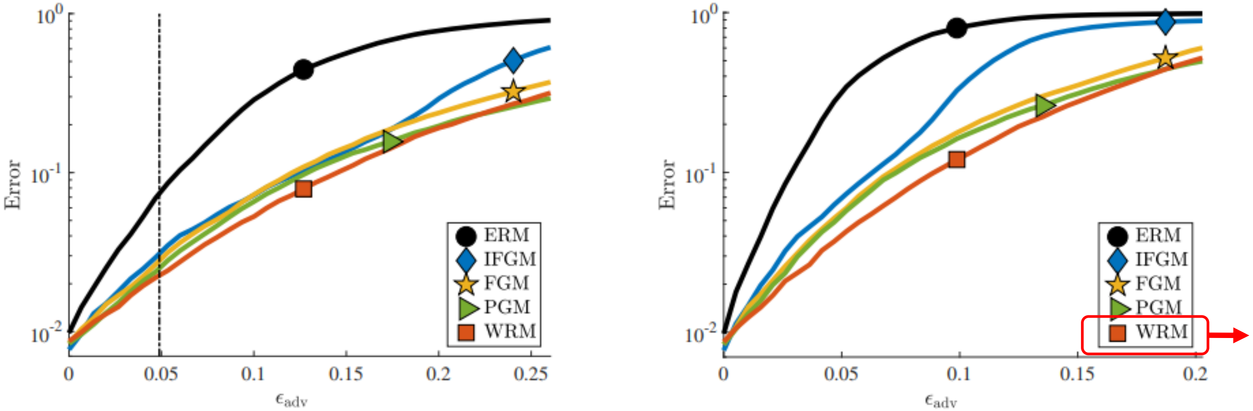
- Large enough γ makes $\ell(\theta^t; z) - \gamma c(z, z^t)$ concave, and it helps the optimization be easy



- The algorithm is attack-agnostic
  - It does not need any adversarial attack method

Defense-Scenarios: Wasserstein Adversarial Training [A. Sinha et al., 2018]

- **Experimental Results:** white-box attack with l_2 and l_∞ metric
  - It shows Wasserstein adversarial training (**WRM**) outperform the baselines (other adversarial trainings)

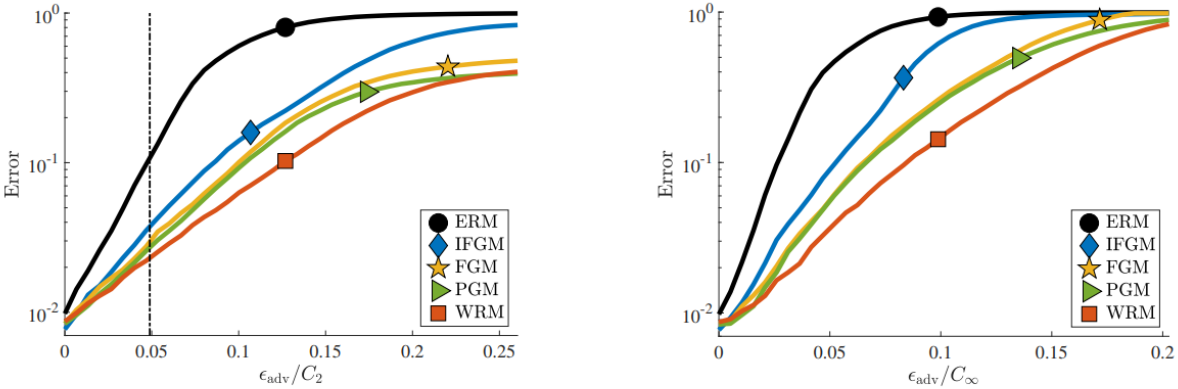


(a) Test error vs. ϵ_{adv} for $\|\cdot\|_2$-FGM attack

(b) Test error vs. ϵ_{adv} for $\|\cdot\|_\infty$-FGM attack

Best performance among the baselines

*ERM: Standard Training
*IFGM: BIM Adv. Training
*FGM: FGSM Adv. Training
*PGM: PGD Adv. Training
*WRM: Wasserstein Adv. Training



(a) Test error vs. ϵ_{adv} for $\|\cdot\|_2$ attack

(b) Test error vs. ϵ_{adv} for $\|\cdot\|_\infty$ attack

* PGD attack

* PGD attack

Table of Contents

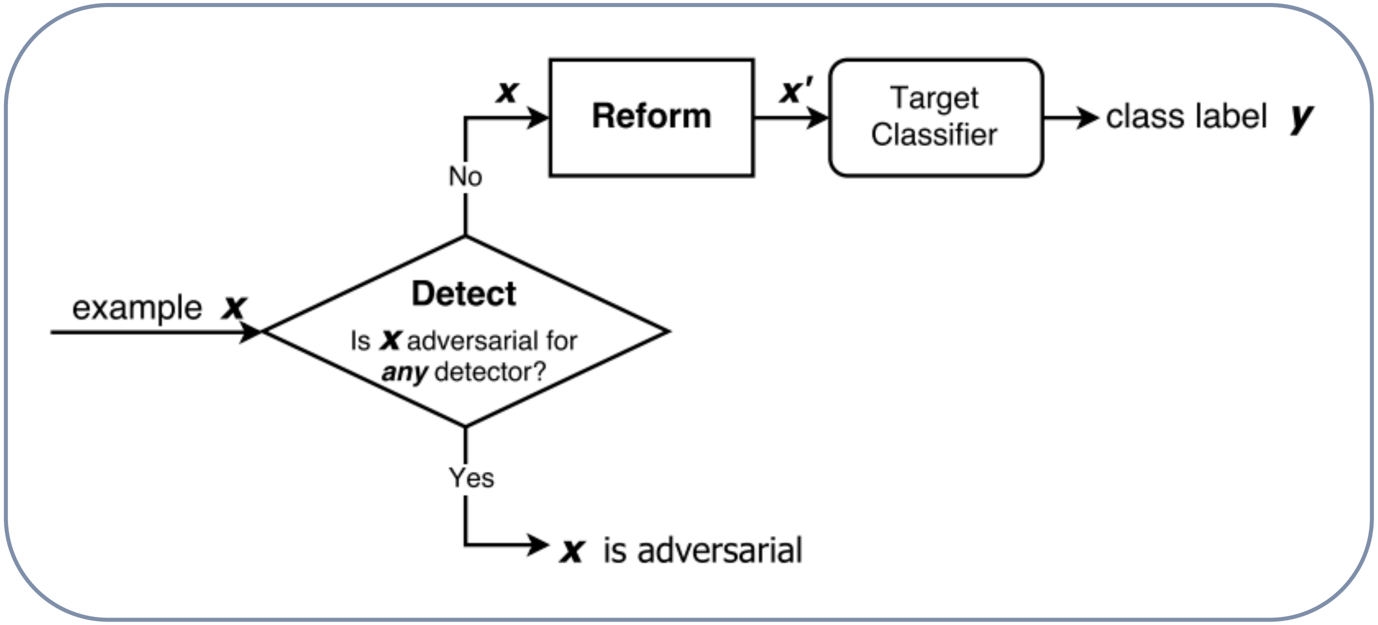
1. **Introduction**
   - What is adversarial example?
   - Overview

2. **Adversarial attack**
   - White-box attack
   - Black-box attack

3. **Adversarial Defense**
   - Adversarial training
   - Input pre-processing
   - Robust network construction

Defense-Scenarios: MagNet [D. Meng et al., 2017]

- **Motivation:** Pre-process a input to make a clean data
  - Denoise adversarial perturbation
  - **Idea:** Double-check an input with **Detector** and **Reformer**
    - Detecting an input whether it is an adversarial example or not
    - Reforming the input which is detected as clean one to prepare in case of the **Detector**'s failure



*source: D. Meng et al., MagNet: a Two-Pronged Defense against Adversarial Examples, In CCS, 2017

Defense-Scenarios: MagNet [D. Meng et al., 2017]

- **Goal:** Training **Detector** and **Reformer** which are based on auto-encoders
- **Method:**
  - **Detector**: Detect an input x is an adversarial example or not
    - Training an auto-encoder f_{AE} to minimize the loss over the training data X

$$\ell(X) = \frac{1}{|X|} \sum_{x \in X} \|x - f_{\mathrm{AE}}(x)\|_2$$

    - **Detector** decides abnormality via the reconstruction error E or Jensen-Shannon Divergence JSD with a threshold t_i with a given classifier f

$$E(x) = \|x - f_{\mathrm{AE}}(x)\|_p > t_1 \text{ where } t_1 = \sup_{x \in X_{\mathrm{val}}} \|x - f_{\mathrm{AE}}(x)\|_p$$

$$\text{where } X_{\mathrm{val}} \text{ is the validation data}$$

$$JSD\left(f_{\mathrm{soft}}(x) || f_{\mathrm{soft}}\left(f_{\mathrm{AE}}(x)\right)\right) > t_2 \text{ where } t_2 \text{ is a hyper-parameter}$$

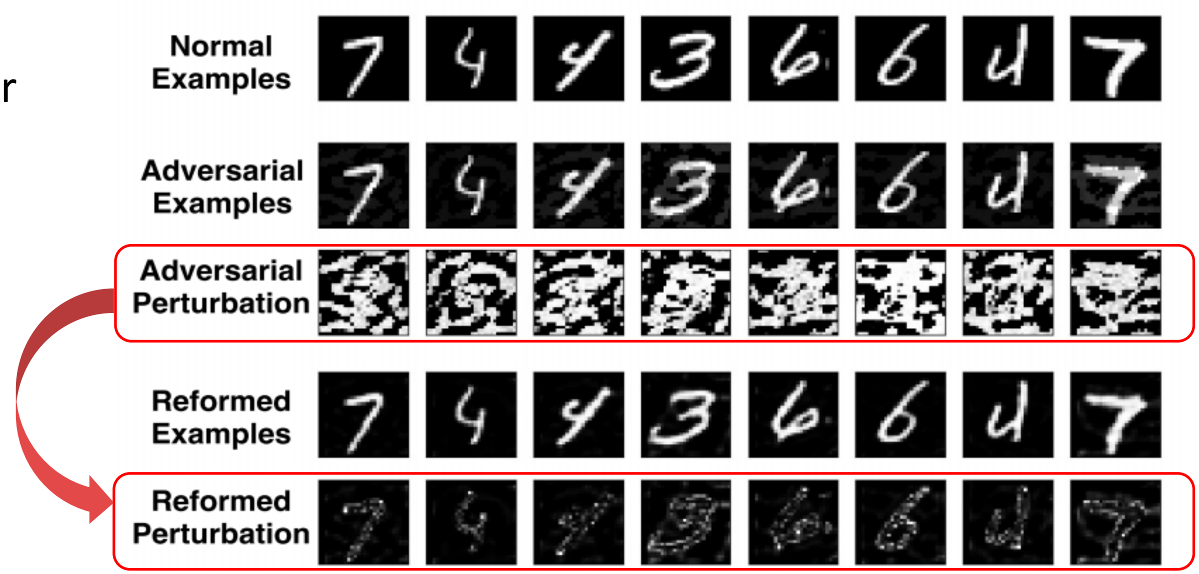
$$\text{where } f_{\mathrm{soft}}(x) \text{ is the softmax ouput of the classifier } f \text{ on an input } x$$

Defense-Scenarios: MagNet [D. Meng et al., 2017]

- **Goal:** Training **Detector** and **Reformer** which are based on auto-encoders
- **Method:**
  - **Reformer**: Reform an input x to lie on the data manifold
    - Training an auto-encoder f_{AE} to minimize the loss over the training data X

$$\ell(X) = \frac{1}{|X|} \sum_{x \in X} \|x - f_{\mathrm{AE}}(x)\|_2$$

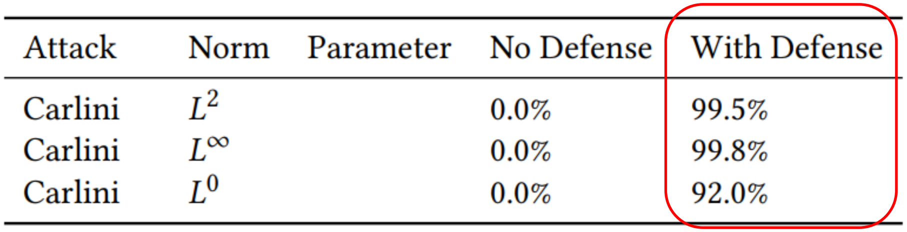
  - To strengthen the network on adversarial attacks, **Detector** and **Reformer** are randomly selected from large number of trained **Detectors** and **Reformers** with different architectures
- **Experimental Results**
  - The Effect of Reformer



Defense-Scenarios: MagNet [D. Meng et al., 2017]

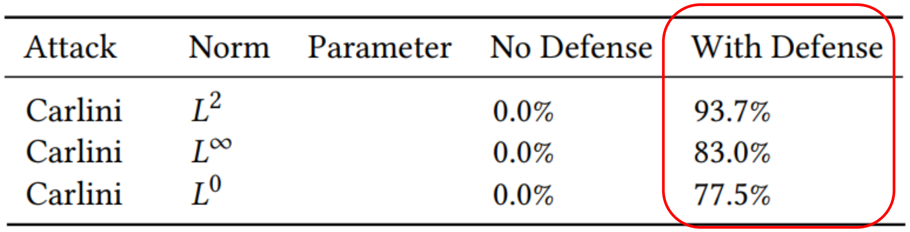
- **Experimental Results**
  - MagNet [D. Meng et al., 2017] reports that it defends CW method

(a) MNIST



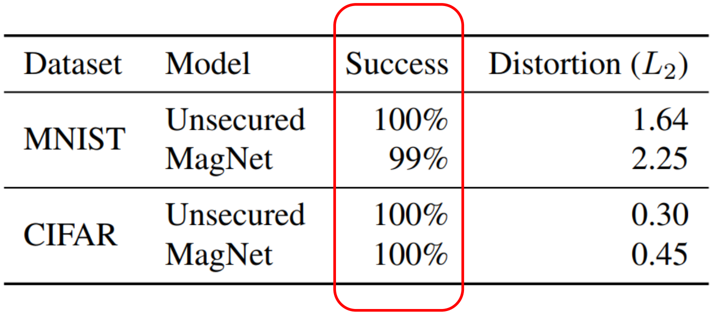
Attack	Norm	Parameter	No Defense	With Defense
Carlini	L^2		0.0%	99.5%
Carlini	L^∞		0.0%	99.8%
Carlini	L^0		0.0%	92.0%

(b) CIFAR



Attack	Norm	Parameter	No Defense	With Defense
Carlini	L^2		0.0%	93.7%
Carlini	L^∞		0.0%	83.0%
Carlini	L^0		0.0%	77.5%

  - But, the results is broken [N. Carlini et al., 2017c]
    - CW method by performing 10,000 iterations of gradient descent



Dataset	Model	Success	Distortion (L_2)
MNIST	Unsecured	100%	1.64
	MagNet	99%	2.25
CIFAR	Unsecured	100%	0.30
	MagNet	100%	0.45

*source: D. Meng et al., MagNet: a Two-Pronged Defense against Adversarial Examples, In CCS, 2017

Defense-Scenarios: Defense GAN [P. Samangouei et al., 2018]

- **Motivation:** Pre-process an input to make a clean data via **WGAN** [M. Arjovsky et al., 2017]
  - MagNet [D. Meng et al., 2017] is based on auto-encoders
  - **WGAN:** For input data x and random vectors w, generator G and discriminator D minimize the following min-max loss V

$$\min_G \max_D V(D,G) = \mathbb{E}_{p(x)}\left[D(x)\right] - \mathbb{E}_{p(w)}\left[D\left(G(w)\right)\right]$$

  - **Idea:** The minimizer $\widehat{w}$ of reconstruction error would generate the clean data

$$\text{Reconstruction error} = \|x - G(z)\|_2^2$$

  - **Method:** Finding a minimizer $\widehat{w}$ of the reconstruction error for given generator G and input x

$$\widehat{w} = \operatorname*{argmin}_w \|x - G(w)\|_2^2$$

    - Choose initial w randomly
    - Use Gradient Descent (GD) with some fixed step (e.g. 200-step)

***min-max loss of original GAN:** $\min_G \max_D V(D,G) = \mathbb{E}_{p(x)}[\log D(x)] - \mathbb{E}_{p(z)}[\log(1 - D(G(z)))]$

Defense-Scenarios: Defense GAN [P. Samangouei et al., 2018]

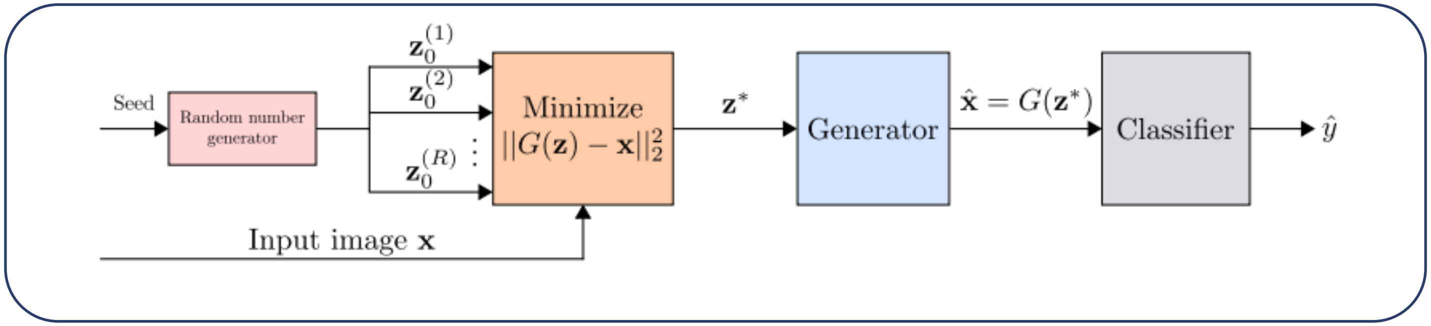
- **Motivation:** Pre-process an input to make a clean data via **WGAN** [M. Arjovsky et al., 2017]
  - MagNet [D. Meng et al., 2017] is based on auto-encoders
  - **WGAN:** For input data x and random vectors w, generator G and discriminator D minimize the following min-max loss V

$$\min_G \max_D V(D,G) = \mathbb{E}_{p(x)}\left[D(x)\right] - \mathbb{E}_{p(w)}\left[D\left(G(w)\right)\right]$$

  - **Idea:** The minimizer $\widehat{w}$ of reconstruction error would generate the clean data

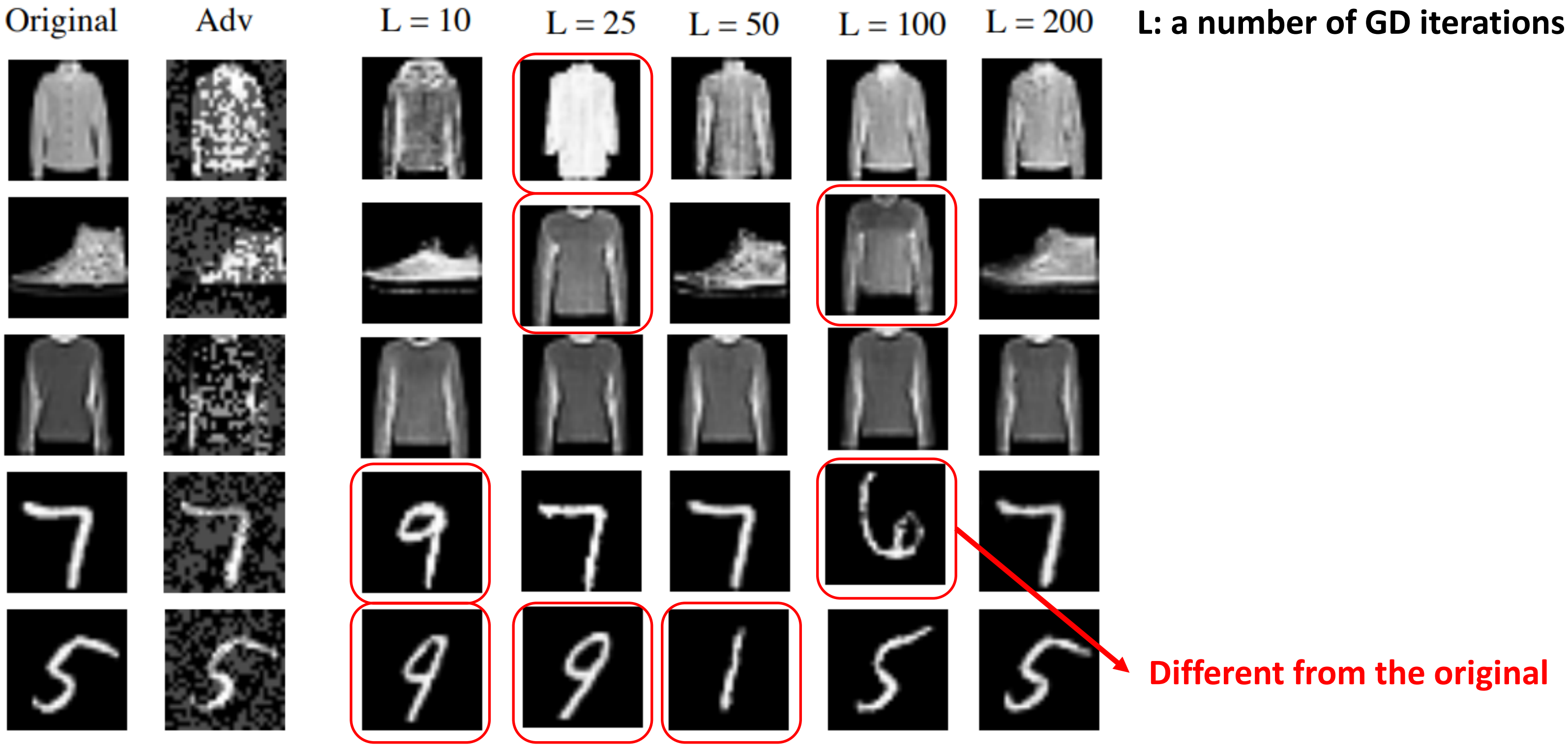
$$\text{Reconstruction error} = \|x - G(z)\|_2^2$$

  - **Method:**
    - **Classifier** can be trained by using either original data or reconstructed data
    - The flow of **Defense-GAN** is



Defense-Scenarios: Defense GAN [P. Samangouei et al., 2018]

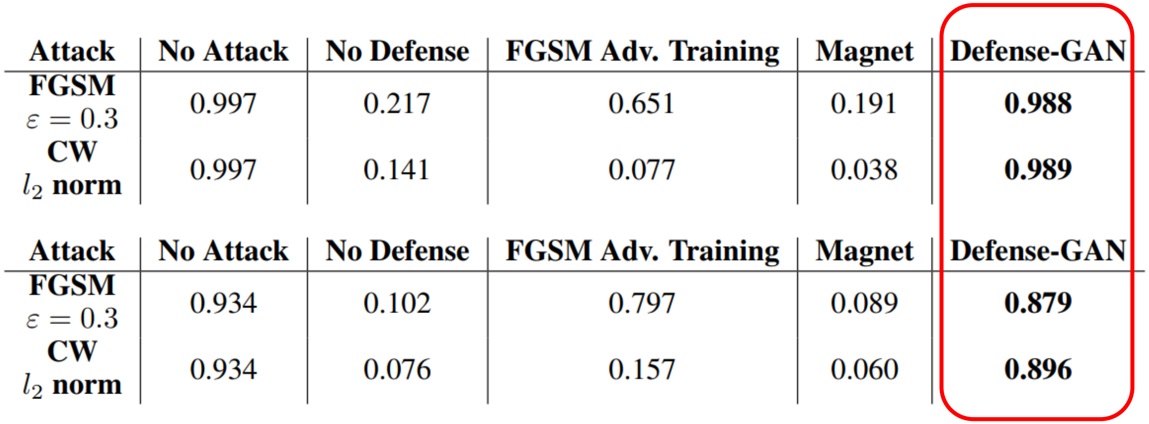
- **Experimental Result**
  - The effect of **Generator**
    - The reconstructed data are clean, but it often different from the original



Reconstructed data with various L

Defense-Scenarios: Defense GAN [P. Samangouei et al., 2018]

- **Experimental Result**
  - White-box attack with FGSM and CW method



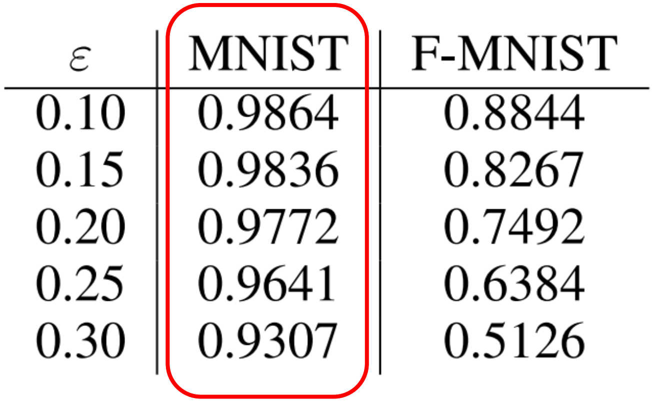
MNIST

Attack	No Attack	No Defense	FGSM Adv. Training	Magnet	Defense-GAN
FGSM $\varepsilon = 0.3$	0.997	0.217	0.651	0.191	**0.988**
CW l_2 **norm**	0.997	0.141	0.077	0.038	**0.989**

F-MNIST

Attack	No Attack	No Defense	FGSM Adv. Training	Magnet	Defense-GAN
FGSM $\varepsilon = 0.3$	0.934	0.102	0.797	0.089	**0.879**
CW l_2 **norm**	0.934	0.076	0.157	0.060	**0.896**

  - Black-box attack with FGSM



ε	MNIST	F-MNIST
0.10	0.9864	0.8844
0.15	0.9836	0.8267
0.20	0.9772	0.7492
0.25	0.9641	0.6384
0.30	0.9307	0.5126

Table of Contents

1. **Introduction**
   - What is adversarial example?
   - Overview

2. **Adversarial attack**
   - White-box attack
   - Black-box attack

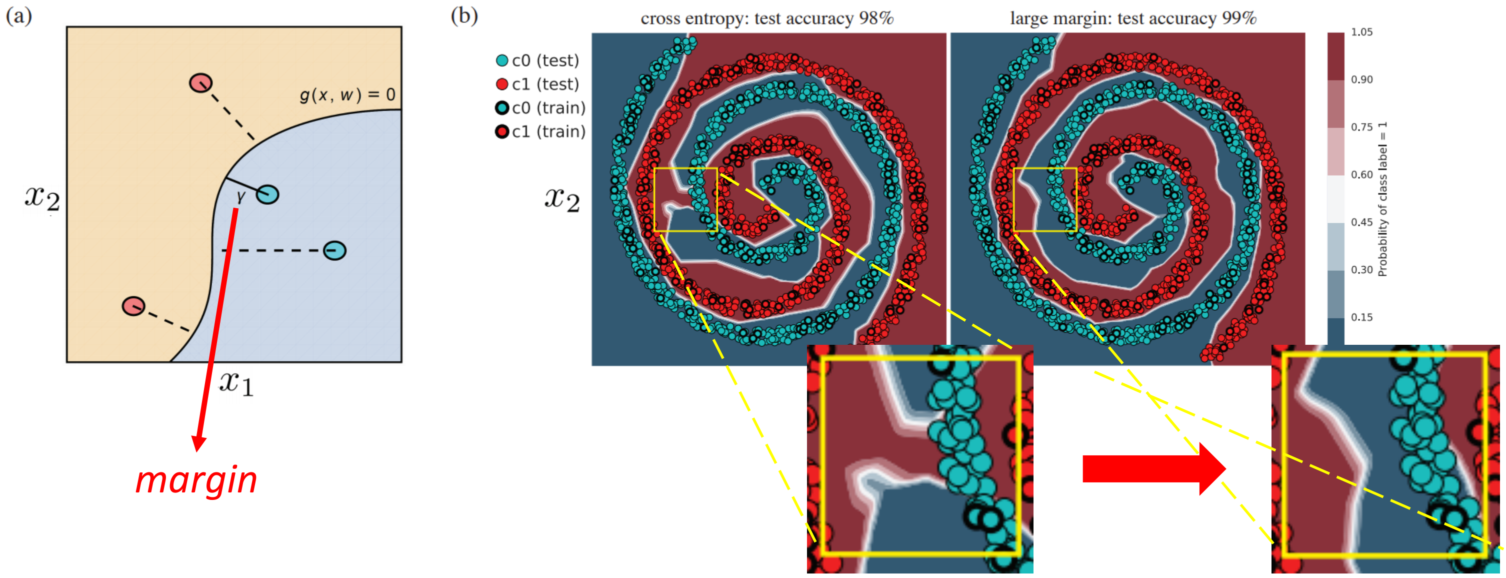
3. **Adversarial Defense**
   - Adversarial training
   - Input pre-processing
   - Robust network construction

Defense-Scenarios: Large Margin Classifier [G. Elsayed et al., 2018]

- **Motivation:** Model robustness would be related to the margin of model
  - **Idea:** Construct a model to have a large margin
    - Margin is the smallest distance from the training data to the decision boundary
    - Define a margin d as follow:

$$d = \min_{\delta} \|\delta\|_p \text{ s.t. } f_i(x+\delta) = f_j(x+\delta)$$

where $f_i(\cdot)$ is i-th logit value of a classifier $f(\cdot)$



margin

Defense-Scenarios: Large Margin Classifier [G. Elsayed et al., 2018]

- **Motivation:** Model robustness would be related to the margin of model
  - **Idea:** Construct a model to have a large margin
    - Margin is the smallest distance from the training data to the decision boundary
    - Define a margin d as follow:

$$d = \min_{\delta} \|\delta\|_p \text{ s.t. } f_i(x+\delta) = f_j(x+\delta)$$

$$\text{where } f_i(\cdot) \text{ is } i\text{-th logit value of a classifier } f(\cdot)$$

    - Appling linear approximation to f_i
    - The approximated margin $\widehat{d}$ can be written in closed form as follow:

$$\widehat{d} = \frac{|f_i(x) - f_j(x)|}{\|\nabla_x f_i(x) - \nabla_x f_j(x)\|_q}$$

$$\text{where } \|\cdot\|_q \text{ is the dual norm of } \|\cdot\|_p, \ q = \frac{p}{p-1}$$

Defense-Scenarios: Large Margin Classifier [G. Elsayed et al., 2018]

- **Goal:** Solving a new loss function to maximize the margin $\widehat{d}\,(\geq \gamma)$

$$\underset{\theta}{\text{minimize}} \sum_{(x,y)\sim\mathcal{D}} \mathcal{A}_{i\neq y} \max\{0, \gamma + \boxed{\frac{f_i(x)-f_y(x)}{\|f_i(x)-f_y(x)\|}}\}$$

margin

$\mathcal{A}:$ aggregate operator; max or sum

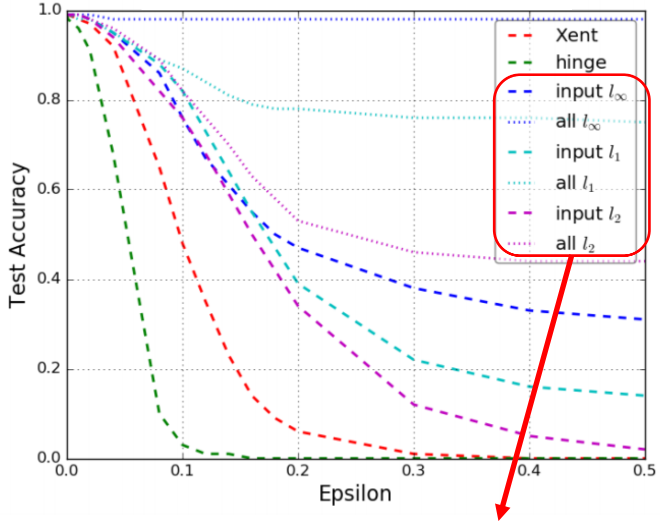
$\gamma:$ hyper-parameter

- **Method:** Adding cross-entropy loss with a small coefficient (less than 0.1%) to keep the classification performance
  - Various metrics can be used
    - l_1, l_2, l_∞
  - The above margin is about input x, but it can be extended to hidden features

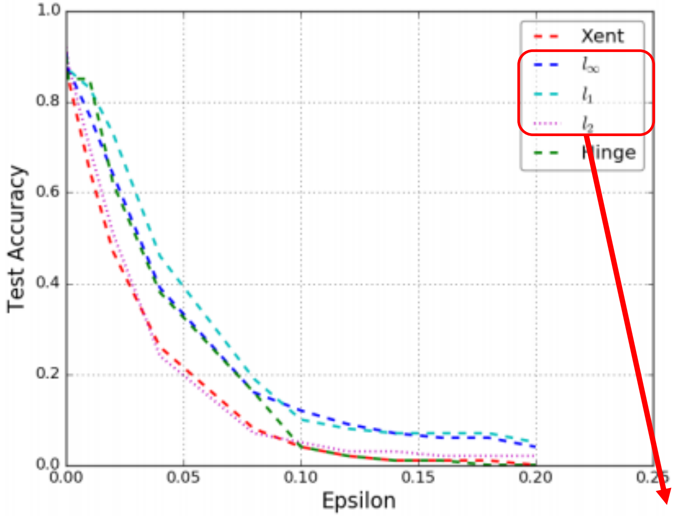
Defense-Scenarios: Large Margin Classifier [G. Elsayed et al., 2018]

- **Experimental Result**
  - Test accuracy of standard model: 99.5%
  - Test accuracy of the margin classifier models: 99.3~99.5%
  - White-box: BIM attack
    - Xent: Cross-entropy loss

MNIST



CIFAR 10



Large margin classifiers

applying on multiple layer
(input, output, some conv layers)

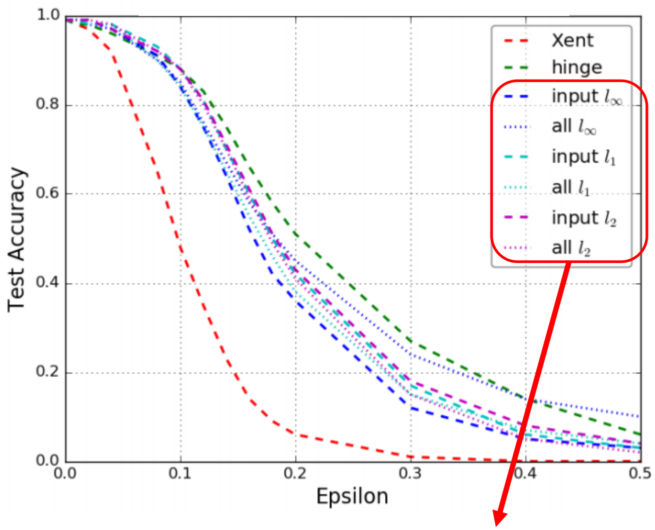
Input l_p: applying the *margin loss* on first layer (input) only

All l_p: applying the *margin loss* on all hidden layer (hidden features)

Defense-Scenarios: Large Margin Classifier [G. Elsayed et al., 2018]

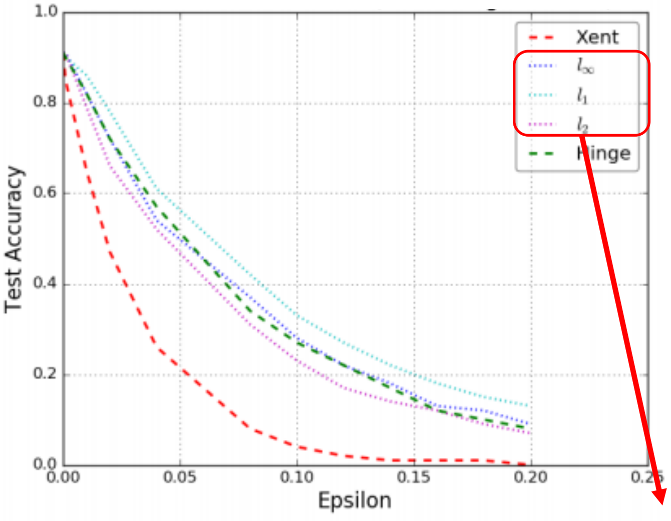
- **Experimental Result**
  - Test accuracy of standard model: 99.5%
  - Test accuracy of the margin classifier models: 99.3∼99.5%
  - Black-box: BIM attack to Xent model
    - Xent: Cross-entropy loss

MNIST



Large margin classifiers

CIFAR 10



applying on multiple layer
(input, output, some conv layers)

Input l_p: applying the *margin loss* on first layer (input) only

All l_p: applying the *margin loss* on all hidden layer (hidden features)

Defense-Scenarios: Parseval Network [M. Cisse et al., 2017]

- **Motivation:** Model robustness would be related to the Lipchitz constant of model
  - **Idea:** The global Lipchitz constant of model is bounded by the function of Lipchitz constants of all layers
  - **Goal:** Controlling the Lipchitz constants (the spectral norm) of all linear and convolutional layers to be smaller than 1
  - **Method:** Perseval regularization with $R_k(\cdot)$
    - A weight matrix $\mathcal{W}$ is called approximately a Parseval tight frame if $\mathcal{W}$ satisfies

$$\mathcal{W}^T\mathcal{W} \approx I \text{ where } I \text{ is the identity matrix}$$

    - Consider layer-wise regularizer $R_k(\cdot)$ of a weight matrix as follow:

$$R_k(\mathcal{W}) = \frac{k}{2}\left\|\mathcal{W}^T\mathcal{W} - I\right\|_2^2$$

    - Optimizing the regularizer $R_k(\cdot)$ is expensive
      - The gradient of the regularizer $R_k(\cdot)$ is follow:

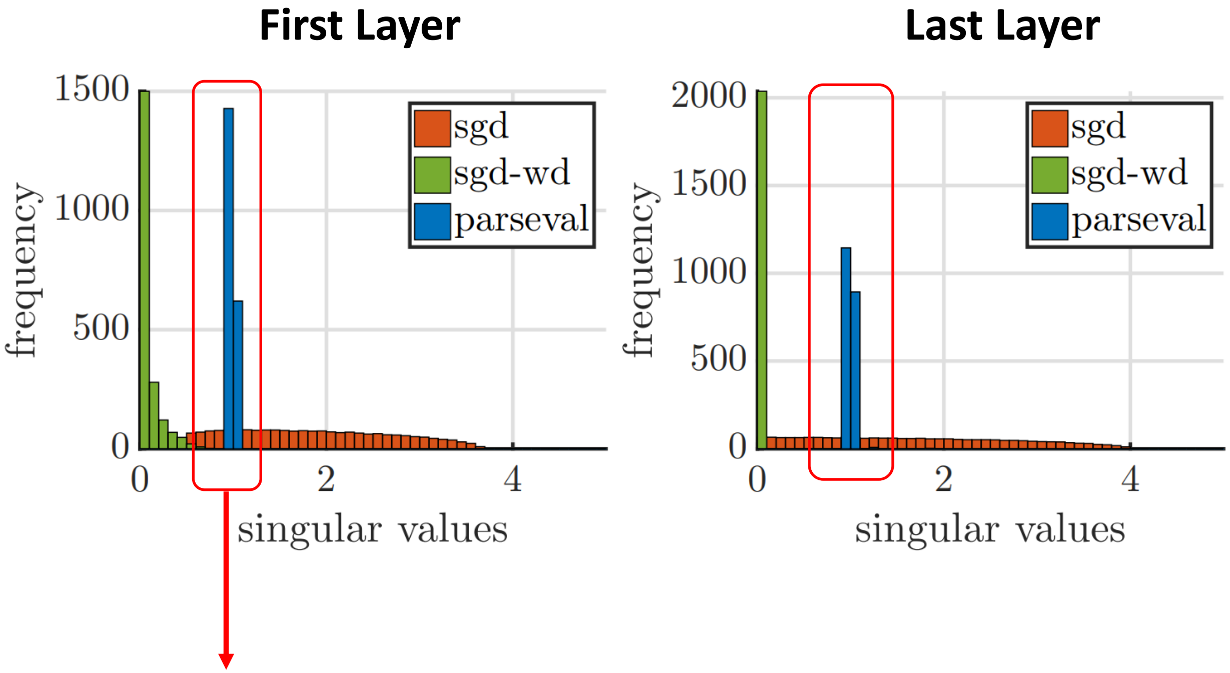
$$\nabla_{\mathcal{W}} R_k(\mathcal{W}) = k(\mathcal{W}\mathcal{W}^T - I)\mathcal{W}$$

      - Performing the following update for regularizer to reduce the cost

$$\mathcal{W} \leftarrow (1+k)\mathcal{W} - k\mathcal{W}\mathcal{W}^T\mathcal{W}$$

Defense-Scenarios: Parseval Network [M. Cisse et al., 2017]

- **Experimental Result**
  - The effect of the perseval regularization
    - Singular values of the weight matrices at the first and last layers of fully connected network in CIFAR10

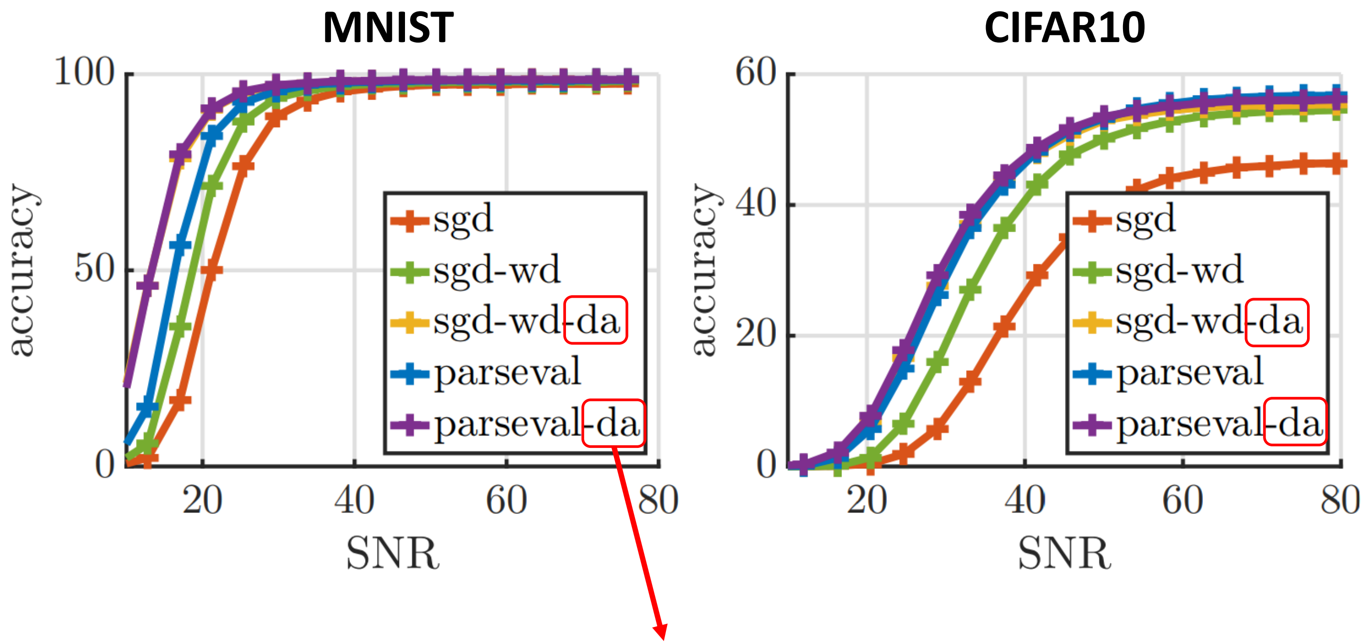


The largest singular value (the spectral norm) is almost 1

Defense-Scenarios: Parseval Network [M. Cisse et al., 2017]

- **Experimental Result**
  - White-box: FGSM attack
    - Signal to Noise Ratio (SNR)

$$\mathrm{SNR}(x, \delta) = 20 \log \frac{\|x\|_2}{\|\delta\|_2}$$



Adversarial Training (Data Augmentation)

Summary

- In this lecture, we cover threat of adversarial examples and various methods of adversarial attack and adversarial defense
    - Adversary could control the prediction of neural network via adversarial examples
- White-box attacks are mainly based on the gradient of model
    - FGSM, BIM, CW method
- Transferability of adversarial examples allow black-box attack
    - [Y. Liu et al., 2017]
    - [N. Papernot et al., 2017]
- There are many adversarial defense methods, but there is no perfect one
    - The most of defenses are heuristic

References

[I. Goodfellow et al., 2015] EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES, In ICLR, 2015
https://arxiv.org/abs/1412.6572

[A. Kurakin et al., 2017a] ADVERSARIAL EXAMPLES IN THE PHYSICAL WORLD, In ICLR Workshop, 2017
https://arxiv.org/abs/1607.02533

[K. Eykholt et al., 2017] Robust Physical-World Attacks on Deep Learning Visual Classification, In CVPR, 2018
https://arxiv.org/abs/1707.08945

[J. Su et al., 2017] One pixel attack for fooling deep neural networks, arXiv, 2017
https://arxiv.org/abs/1710.08864

[D. Karmon et al., 2018] LaVAN: Localized and Visible Adversarial Noise, In ICML, 2018
https://arxiv.org/abs/1801.02608

[C. Xie et al., 2017] Adversarial Examples for Semantic Segmentation and Object Detection, In ICCV, 2017
https://arxiv.org/abs/1703.08603

[A. Kurakin et al., 2017b] ADVERSARIAL MACHINE LEARNING AT SCALE, In ICLR, 2017
https://arxiv.org/abs/1611.01236

[N. Carlini et al., 2017] Towards Evaluating the Robustness of Neural Networks, In IEEE S&P, 2017
https://arxiv.org/abs/1608.04644

[C. Xie et al., 2018] Improving Transferability of Adversarial Examples with Input Diversity, arXiv, 2018
https://arxiv.org/abs/1803.06978

[Y. Liu et al., 2017] DELVING INTO TRANSFERABLE ADVERSARIAL EXAMPLES AND BLACK-BOX ATTACKS, In ICLR, 2017
https://arxiv.org/abs/1611.02770

[A. Madry et al., 2018] Towards Deep Learning Models Resistant to Adversarial Attacks, In ICLR, 2018
https://arxiv.org/abs/1706.06083

References

[D. Meng et al., 2017] MagNet: a Two-Pronged Defense against Adversarial Examples, In CCS, 2017
https://arxiv.org/abs/1705.09064

[N. Carlini et al., 2017b] MagNet and "Efficient Defenses Against Adversarial Attacks" are Not Robust to Adversarial Examples, arXiv, 2017
https://arxiv.org/abs/1711.08478

[A. Athalye et al., 2018a] Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples, In ICML, 2018
https://arxiv.org/abs/1802.00420

[G. Elsayed et al., 2018] Large Margin Deep Networks for Classification, In NIPS, 2018
https://arxiv.org/abs/1803.05598

[N. Papernot et al., 2017] Practical Black-Box Attacks against Machine Learning, In ACM CCS,2017
https://arxiv.org/abs/1602.02697

[A. Athalye et al., 2018b] Synthesizing robust adversarial examples, In ICML, 2018
https://arxiv.org/abs/1707.07397

[A. Sinha et al., 2018] CERTIFYING SOME DISTRIBUTIONAL ROBUSTNESS WITH PRINCIPLED ADVERSARIAL TRAINING, In ICLR, 2018
https://arxiv.org/abs/1710.10571

[E. Wong et al., 2018] Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope, In ICML, 2018
https://arxiv.org/abs/1711.00851

[K. Lee et al., 2018] A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, In NIPS, 2018
https://arxiv.org/abs/1807.03888

References

[S. Moosavi-Dezfooli et al., 2016] Universal adversarial perturbations, In CVPR, 2017
https://arxiv.org/abs/1610.08401

[N. Carlini et al., 2017c] MagNet and “Efficient Defenses Against Adversarial Attacks” are Not Robust to Adversarial Examples, In arXiv, 2017
https://arxiv.org/abs/1711.08478

[J. Buckman et al., 2018] Thermometer Encoding: One Hot Way To Resist Adversarial Examples, In ICLR, 2018
https://openreview.net/forum?id=S18Su--CW

[G. Dhillon et al., 2018] Stochastic Activation Pruning for Robust Adversarial Defense, In ICLR, 2018
https://arxiv.org/abs/1803.01442

[J. Blanchet et al., 2016] Quantifying Distributional Model Risk via Optimal Transport, In arXiv, 2016
https://arxiv.org/abs/1604.01446

[P. Samangouei et al., 2018] DEFENSE-GAN: PROTECTING CLASSIFIERS AGAINST ADVERSARIAL ATTACKS USING GENERATIVE MODELS, In ICLR, 2018
https://arxiv.org/abs/1805.06605

[M. Arjovsky et al., 2017] Wasserstein GAN, In ICML, 2017
https://arxiv.org/abs/1701.07875

[M. Cisse et al., 2017] Parseval Networks: Improving Robustness to Adversarial Examples, In ICML, 2017
https://arxiv.org/abs/1704.08847

[S. Moosavi-Dezfooli et al., 2016] DeepFool: a simple and accurate method to fool deep neural networks, In CVPR, 2016
https://arxiv.org/abs/1511.04599