Adversarial Examples

AI602: Recent Advances in Deep Learning

Lecture 6

Slide made by

Jongheon Jeong

KAIST EE

Previous Course Information

- Assignment: 2 paper summary + 1 presentation
  - Each student should choose two deep learning papers published at NIPS, ICML or ICLR, CVPR, ICCV, ECCV in last 3 years, where the authors do not release their codes.
  - I will help for deciding which papers to study (e.g., use the office hours to ask or send emails to me, including your generic interests and backgrounds)
  - Once you choose papers, try implementing the algorithms on your own using TensorFlow or PyTorch, reproducing the authors' results (reported in their papers) and applying to other datasets
  - Send the report on the first paper by Oct. 25th and the report on the second paper by Dec. 20th to TA. You also have to send your source-code files with the reports.
  - You have to present one of two papers at the end of this class.

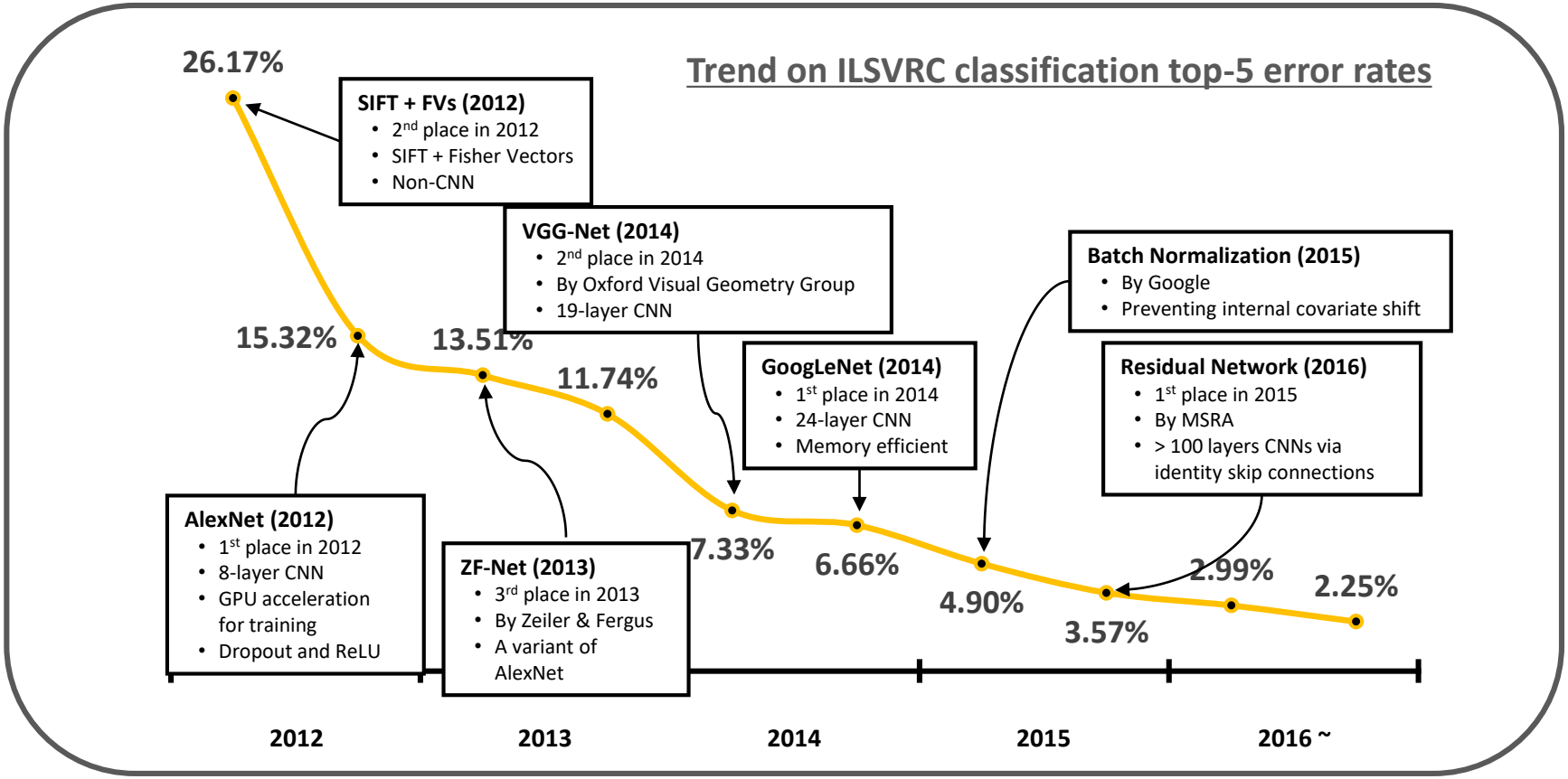
New Course Information

- Assignment: 2 paper summary + 1 presentation
  - Each student should choose two deep learning papers published at NIPS, ICML or ICLR, CVPR, ICCV, ECCV in last 3 years.
  - You can use the authors' codes, but you will receive better grades if (a) the authors do not release their codes or (b) you modify the authors' code for better performance.
  - I will help for deciding which papers to study (e.g., send emails to me or ask after the class)
  - Try reproducing the authors' results (reported in their papers) and applying to other datasets. Or, modify the authors' code or algorithm for better performance.
  - Send the report on the first paper by Oct. 25th and the report on the second paper by Dec. 20th to TA. You also have to send your source-code files with the reports.
  - You have to present one of two papers at the end of this class.

Table of Contents

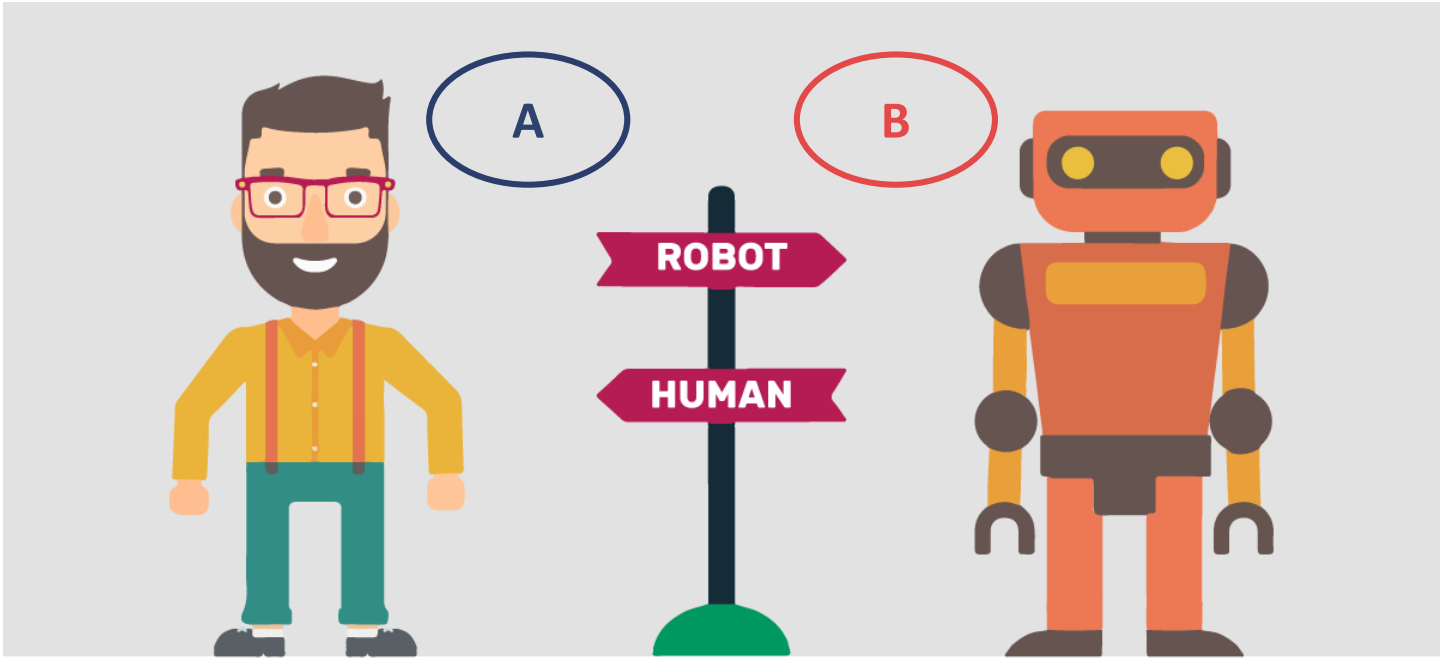
Table of Contents

- **Deep learning system** have achieved state-of-art on various AI-related tasks
  - Super-human performance on image recognition problems



Back Side of The Neural Network Success

- **Deep learning system** have achieved state-of-art on various AI-related tasks
  - Super-human performance on image recognition problems
- **Problem**: ML systems are highly vulnerable to a small noise on input that are specifically designed by an adversary
  - In other words, **answer of machine ≠ answer of human**

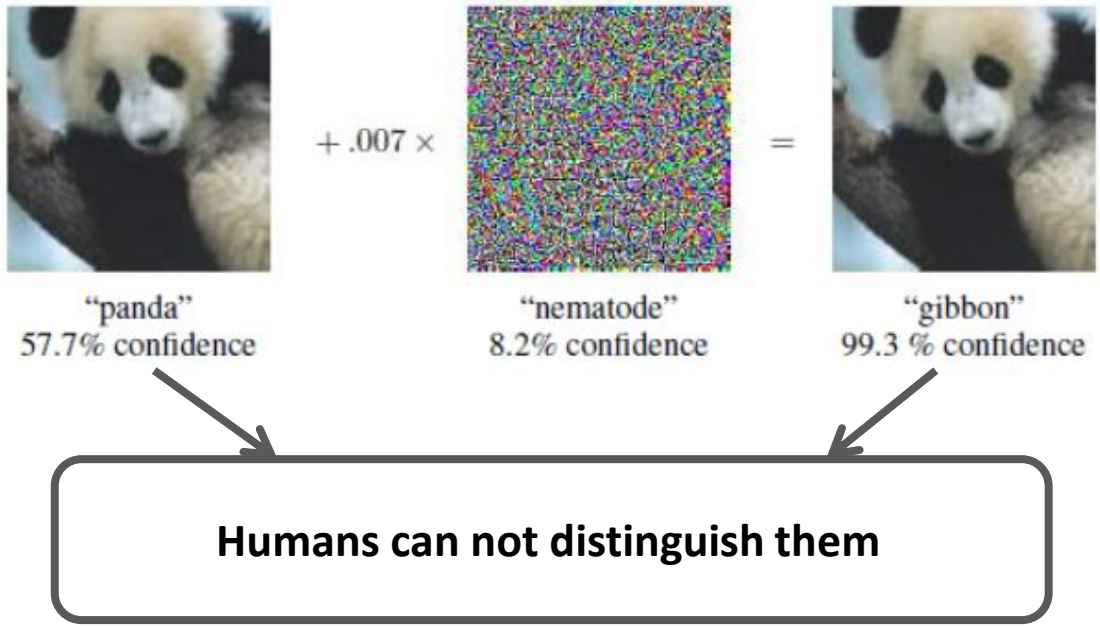


WORDBERRY
SOFTWARE TRANSLATION & LOCALIZATION

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

What is The Adversarial Example?

- **Deep learning system** have achieved state-of-art on various AI-related tasks
  - Super-human performance on image recognition problems
- **Problem**: ML systems are highly vulnerable to a small noise on input that are specifically designed by an adversary
  - In other words, **answer of machine ≠ answer of human**
- Even state-of-the-art-level neural networks make erroneous outputs
  - **Example**: GoogleNet trained on ImageNet dataset



*source: Goodfellow et al., Explaining and Harnessing Adversarial Examples, ICLR 2015

What is The Adversarial Example?

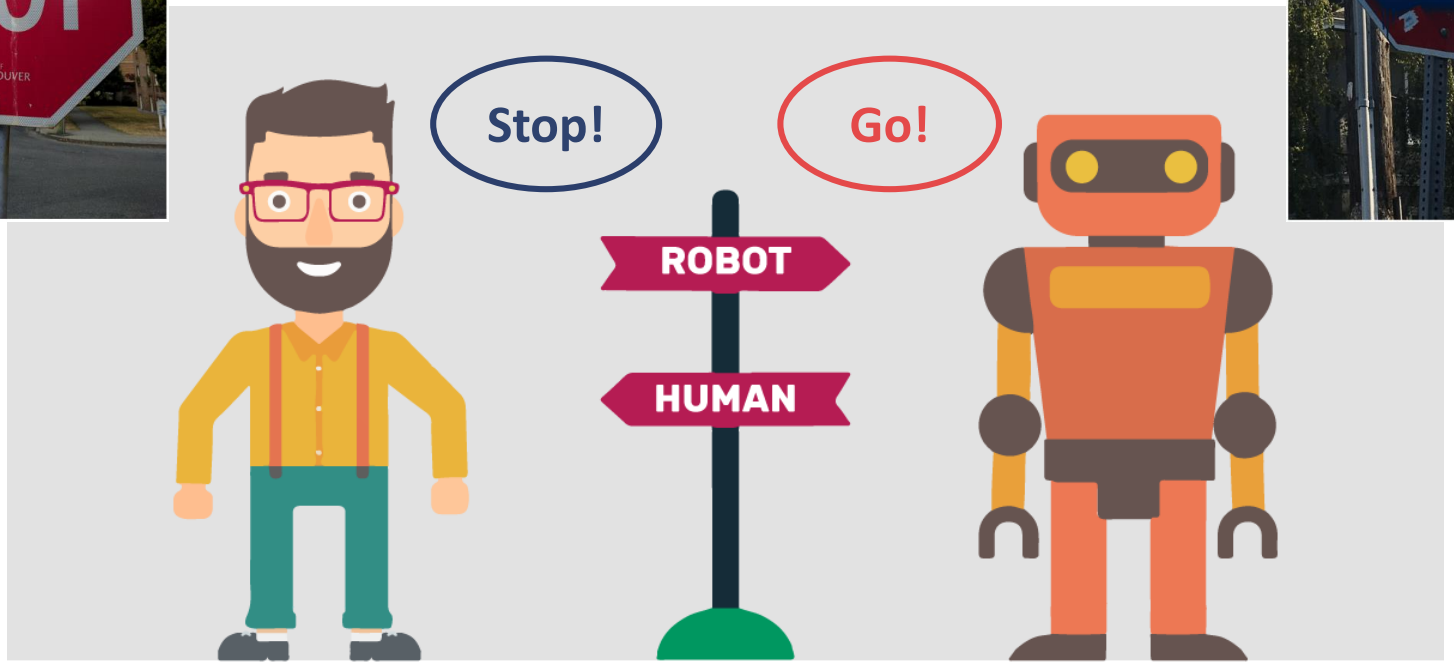
- **Deep learning system** have achieved state-of-art on various AI-related tasks
  - Super-human performance on image recognition problems
- **Problem**: ML systems are highly vulnerable to a small noise on input that are specifically designed by an adversary
  - In other words, **answer of machine ≠ answer of human**
- Even state-of-the-art-level neural networks make erroneous outputs
  - **Example**: GoogleNet trained on ImageNet dataset

It is called an adversarial example!

*source: Goodfellow et al., Explaining and Harnessing Adversarial Examples, ICLR 2015

Threat of Adversarial Examples

- Adversarial examples raise issues critical to the "**AI safety**" in the real world
  - e.g. Autonomous vehicles may misclassify graffiti stop signs



WORDBERRY
SOFTWARE TRANSLATION & LOCALIZATION

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

- Furthermore, adversarial examples exist across various tasks or modalities
  - Adversarial examples for segmentation task [Xie et al., 2017]

  - Adversarial examples for automatic speech recognition [Qin et al., 2019]

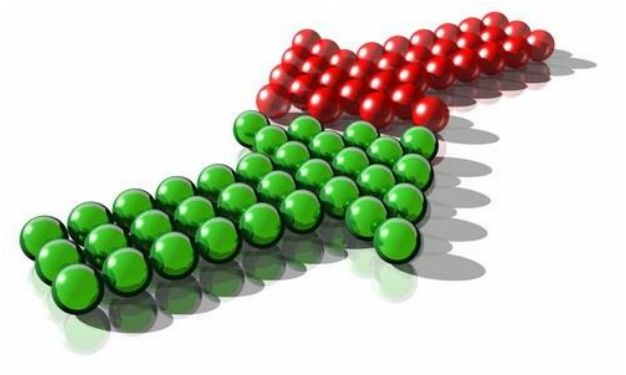
Clean: "The sight of you bartley to see you living and happy and successful can I never make you understand what that means to me"

Adversarial: "Hers happened to be in the same frame too but she evidently didn't care about that"

*source:
Xie et al., Adversarial Examples for Semantic Segmentation and Object Detection, ICCV 2017
Qin et al., Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition, ICML 2019

The Adversarial Game: Attacks and Defenses

- The literature of adversarial example commonly stated in security perspective
  - **Attacks:** Design inputs for a ML system to produce erroneous outputs
  - **Defenses:** Prevent the misclassification by adversarial examples



- In this perspective, specifying a **threat model** of the game is important

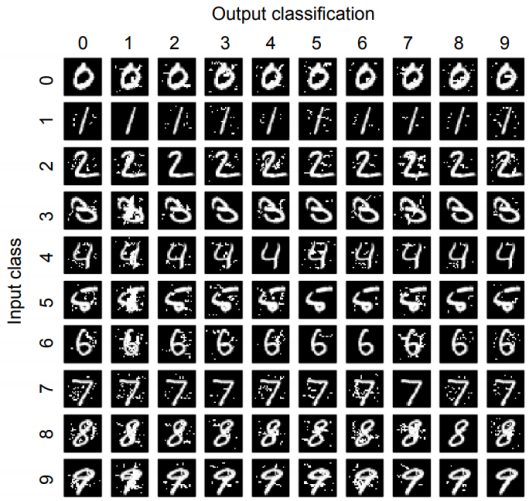
1. **Adversary goals**
2. **Adversarial capabilities**
3. **Adversary knowledge**

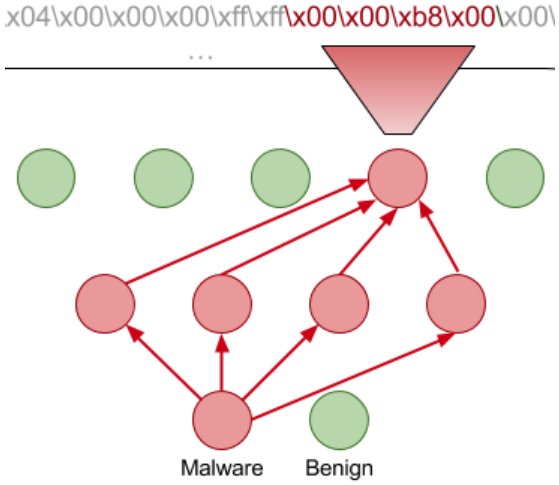
*source: https://gwynteatro.wordpress.com/2011/10/30/ambiguity-and-contradiction-leadership-certainties

The Adversarial Game: Threat model

- The literature of adversarial example commonly stated in security perspective
- In this perspective, specifying a **threat model** of the game is important

1. **Adversary goals**: Simply to cause misclassification, or else?
   - Some adversary may be interested in to attack into a target class of their choice
     - "**Source-target**" [Papernot et al., 2016], or "**targeted**" [Carlini & Wagner, 2017] attack
   - In other setting, only a specific type of misclassification may be interesting
     - e.g. Malware detection: "Benign → malware" is usually out-of-interest





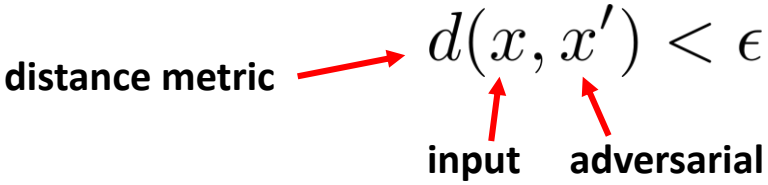
*source:
Carlini & Wagner, Towards Evaluating the Robustness of Neural Networks, IEEE SSP 2017
https://devblogs.nvidia.com/malware-detection-neural-networks/

- The literature of adversarial example commonly stated in security perspective
- In this perspective, specifying a **threat model** of the game is important

2. Adversarial capabilities

- Reasonable constraints to adversary allow us to build more meaningful defenses
  - Too large perturbations to an image may break even the human's decision

- To date, most defenses restrict the adversary to make "small" changes to inputs

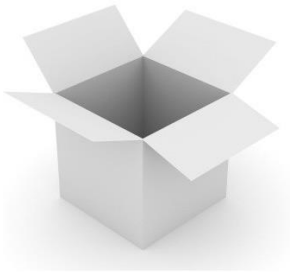


- A common choice for $d(\cdot,\cdot)$ is ℓ_p**-distance** (especially for image classification)
  - ℓ_∞**-norm ball**: the adversary cannot modify each pixel by more than ϵ
  - ℓ_0**-norm ball**: the adversary can arbitrary change at most ϵ pixels

- The literature of adversarial example commonly stated in security perspective
- In this perspective, specifying a **threat model** of the game is important

3. Adversary knowledge

- A threat model must describe what knowledge the adversary is assumed to have
  - **White-box model:** Complete knowledge of the model and its parameter
  - **Black-box model:** No knowledge of the model
- **Gray-box**: Some threat models specify the various degree of access
  - A limited number of queries to the model
  - Access to the predicted probabilities, or just class
  - Access to the training data
- The guiding principle: **Kerckhoffs' principle** [Kerckhoffs, 1883]
  - The adversary is assumed to completely know the inner workings of the defense



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/

- A precise threat model → **well-defined measures** of adversarial robustness
  1. "**Adversarial risk**": The worst-case loss L for a given perturbation budget

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{x':d(x,x')<\epsilon} L(f(x'),y)\right]$$

Data distribution **model**

  2. The average minimum-distance of the adversarial perturbation

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\min_{x'\in A_{x,y}} d(x,x')\right]$$

set of adv. examples

- For misclassification, $A_{x,y} = \{x' : f(x') \neq y\}$
- For targeted attack, $A_{x,y} = \{x' : f(x') = t\}$ for some target class t

- **Key challenge:** Computing adversarial risk is usually intractable
  - We have to approximate these quantities
    - Much harder problem than approximating "average-case" robustness
  - The heart reason of why evaluating adversarial robustness is difficult

Table of Contents

Table of Contents

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

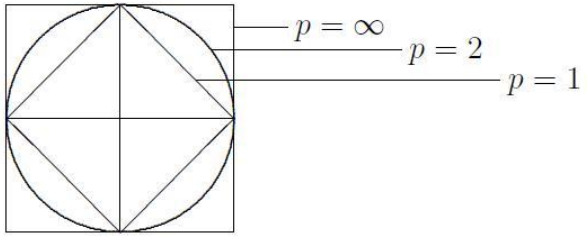
- In **vision ML system**, the following threat model is common:
  1. **Goal -** Untargeted attack: Find $\text{argmax}_{x':d(x,x')<\epsilon} L(f(x'), y)$
  2. **Capabilities -** Pixel-wise restriction: $d(x, x') = ||x - x'||_\infty := \max_i |x_i - x'_i| \leq \epsilon$
  3. **Knowledge -** White-box: Full access to the target network

- **Fast Gradient Sign Method (FGSM)**: A fast approximation of this threat model
  - **Idea:** In white-box setting, one can get the gradients w.r.t input of the network
- FGSM solves the maximization via **linearizing** the loss:

$$\max_{x':||x-x'||_\infty \leq \epsilon} L(f(x'), y) \approx L(f(x), y) + \delta \cdot \nabla_x L(f(x), y)$$

- To meet the max-norm constraint, FGSM takes $\text{sign}(\cdot)$ on the gradient
  - **Quiz.** Why the use of $\text{sign}(\cdot)$ maximizes the loss?

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x L(f(x), y))$$



*source: Goodfellow et al., Explaining and Harnessing Adversarial Examples, ICLR 2015

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

- The idea of FGSM can be directly applied to **targeted** attack model:
  1. **Goal -** Targeted attack
  2. **Capabilities -** Pixel-wise restriction: $d(x, x') = ||x - x'||_\infty := \max_i |x_i - x'_i| \leq \epsilon$
  3. **Knowledge -** White-box: Full access to the target network

- Unlike FGSM, **Least-likely Class Method** minimizes the loss for the target class
- Nevertheless, one could also linearize the loss L

$$\min_{x': ||x - x'||_\infty \leq \epsilon} L(f(x'), y_{\text{target}})$$

- This formulation leads to an attack method similar to FGSM:

$$x' = x - \epsilon \cdot \text{sign}(\nabla_x L(f(x), y_{\text{target}}))$$

Now, we perform "gradient descent"

*source: Kurakin et al., Adversarial Machine Learning at Scale, ICLR 2017

White-Box: Projected Gradient Descent [Madry et al., 2018]

- FGSM can be **generalized** toward a stronger method
  1. Single-step update → multi-step optimization
  2. $\text{sign}(\cdot)$ → Generalized projection operation

- Essentially, our goal is to solve the following optimization:

$$\max_{x' \in x+\mathcal{B}} L(f(x'), y)$$

set of neighbors

- **Projected Gradient Descent (PGD)** is a direct way to solve this:

$$x^{t+1} = \Pi_{x+\mathcal{B}}(x^t + \alpha \cdot \text{sign}(\nabla_x L(f(x^t), y)))$$

projection

  - **Basic Iterative Method (BIM):** $x^0 := x$
  - Usually, **PGD** refers the case when x^0 is randomly-chosen inside $x + \mathcal{B}$

- In some sense, PGD is regarded as the strongest first-order adversary
  - It is the best way we could try using only gradient information

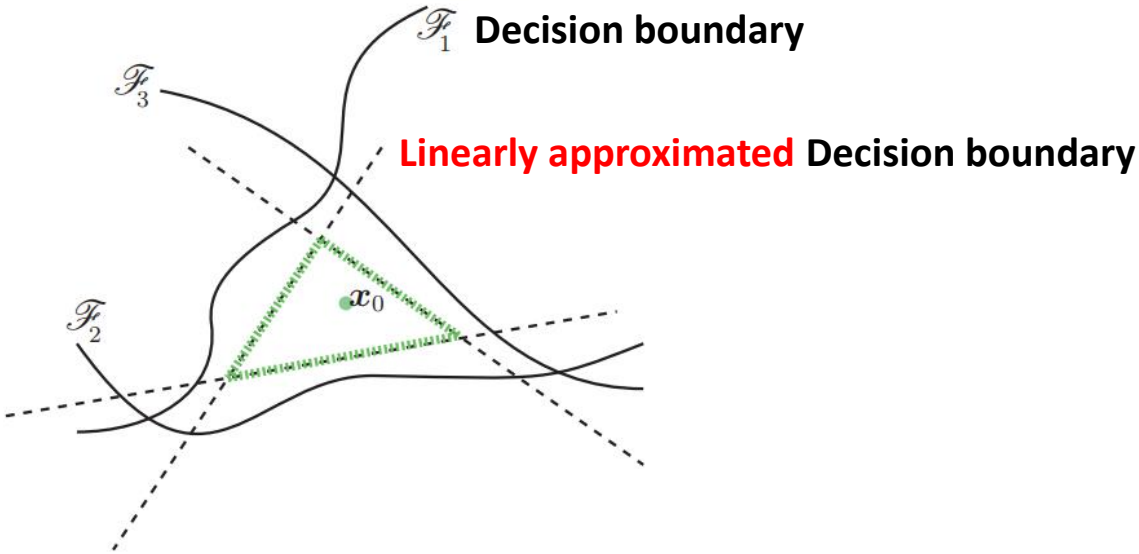
*source: Madry et al., Towards Deep Learning Models Resistant to Adversarial Attacks, ICLR 2018

- **Recall:** One may interest to measure the average minimum-distance

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\min_{x'\in A_{x,y}} d(x,x')\right]$$

  - For misclassification, $A_{x,y} = \{x' : f(x') \neq y\}$
  - For targeted attack, $A_{x,y} = \{x' : f(x') = t\}$ for some target class t

- However, FGSM and PGD do not give explicit information about this
- **DeepFool** approximates this by computing the closest decision boundary
  - By using linear approximation to decision boundaries



*source: Moosavi-Dezfooli et al., DeepFool: a simple and accurate method to fool deep neural networks, CVPR 2016

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

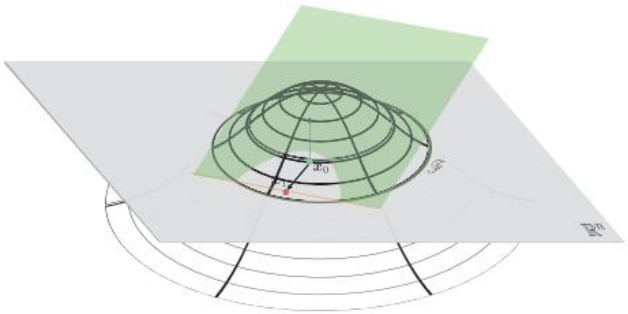
- **DeepFool** approximates this by computing the closest decision boundary
  - By using linear approximation to decision boundaries
- Suppose a multi-class classifier $f(x)$ is defined by:

$$\hat{k}(x) = \underset{k}{\operatorname{argmax}}\, f_k(x)$$

classifier for k-th class

- Under linearity, the distance from x to the boundary of f_l is computable:

$$d_l = \frac{|f_l(x) - f_{\hat{k}(x)}(x)|}{||\nabla_x f_l(x) - \nabla_x f_{\hat{k}(x)}(x)||_2}$$



- Like FGSM → PGD, This process is done **iteratively**:
  - More accurate approximation of d is possible
  - Also, a good adversarial example could also obtained

$$\hat{l} = \operatorname{argmin}_{l \neq \hat{k}(x_0)}\, d_l$$

$$x^{t+1} = x^t + d_{\hat{l}} \cdot (\nabla_x f_{\hat{l}}(x_t) - \nabla_x f_{\hat{k}(x_0)}(x_t))$$

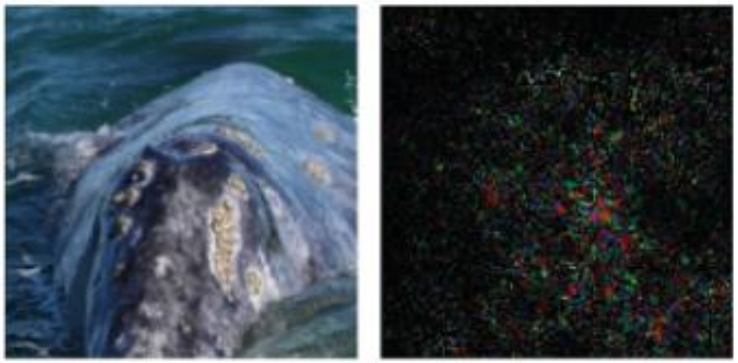
*source: Moosavi-Dezfooli et al., DeepFool: a simple and accurate method to fool deep neural networks, CVPR 2016

- **Experimental Results**
  - **Avg. minimum-distance** among four different networks
  - DeepFool finds more accurate approximation of avg. minimum-distance

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

  - Adversarial examples made by DeepFool have a smaller perturbation

FGSM



DeepFool

*source: Moosavi-Dezfooli et al., DeepFool: a simple and accurate method to fool deep neural networks, CVPR 2016

- **Carlini & Wagner (CW)**: Even tighter approximation is possible:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\min_{x'\in A_{x,y}} d(x,x')\right]$$

- **CW** attempts to directly minimize the distance $||\delta||$ in targeted attack

$$\min_{\delta: k(x+\delta)=y_{\text{target}}} \|\delta\|_2$$

  - **Key challenge**: How to incorporate the constraint during optimization
  - CW takes the Lagrangian relaxation to allow the gradient-based optimization:

$$\min_{\delta} \|\delta\|_2 + \alpha \cdot g(x+\delta)$$

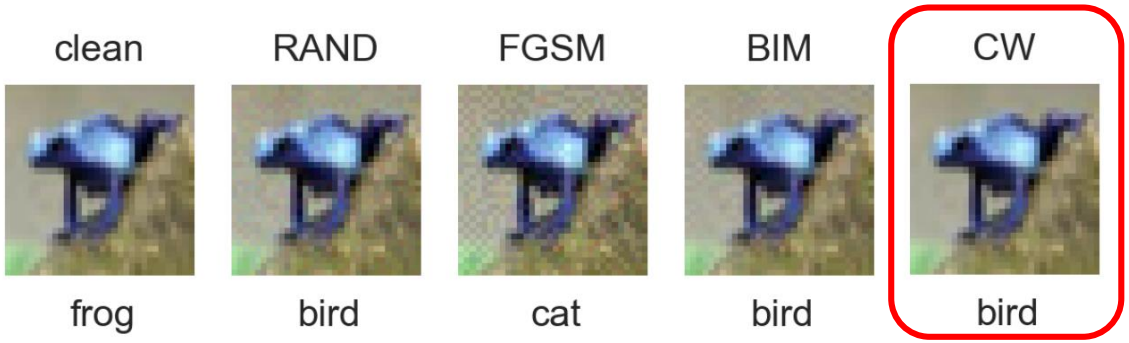
  - $g(x) = \left(\max_{i\neq\text{target}} f_i(x) - f_{y_{\text{target}}}(x)\right)^+$ $\max(0,x)$
  - $g(x)$ attains the minimum when x is an adversarial example

*source: Carlini & Wagner, Towards Evaluating the Robustness of Neural Networks, IEEE S&P 2017

- **Experimental Results**
  - CW finds **much smaller avg. minimum-distance** than DeepFool

		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%

  - Comparison of images generated from several attacks [Y. Song et al., 2018]



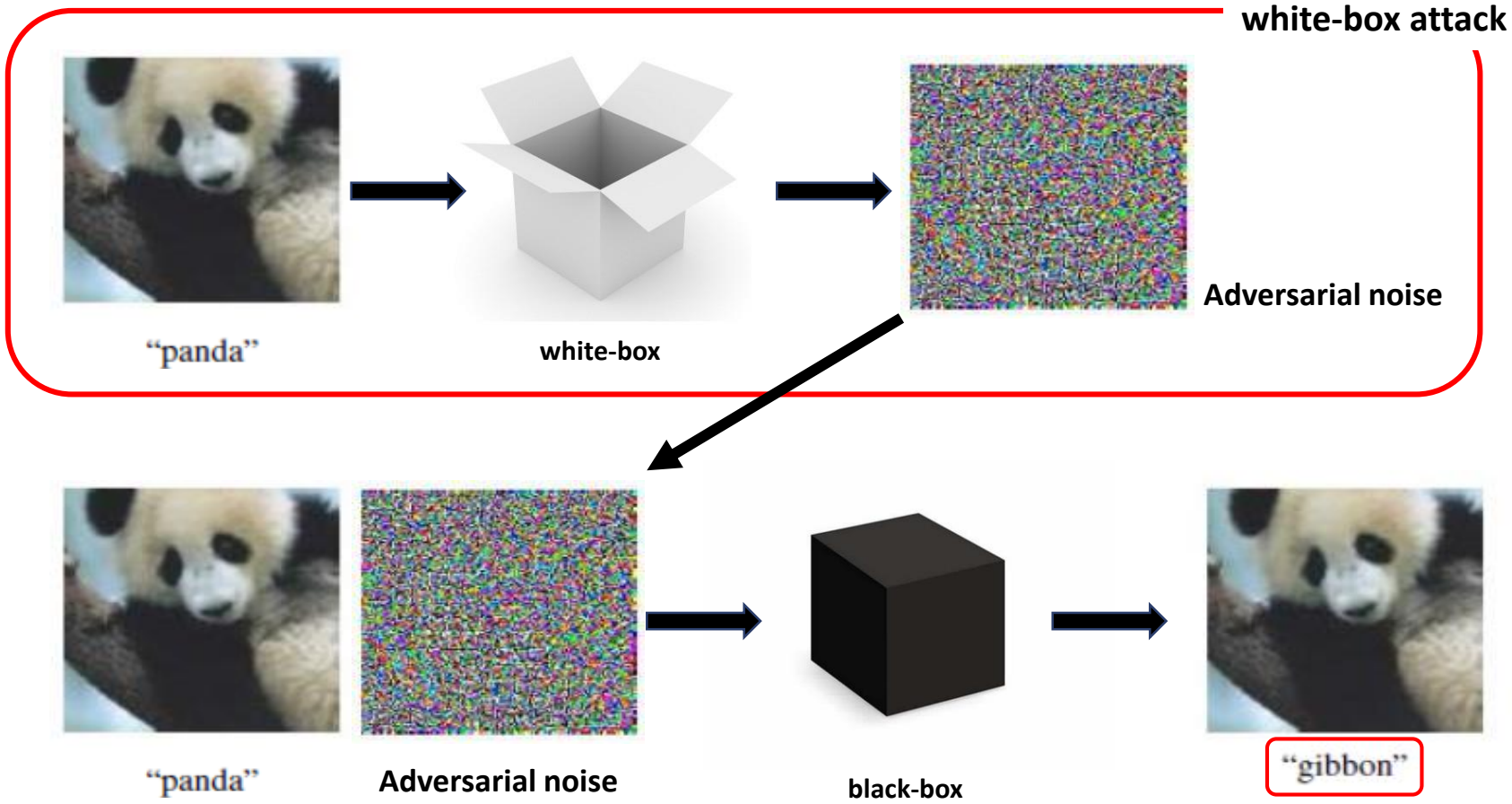
It is the most similar to clean image

*source:
Carlini & Wagner, Towards Evaluating the Robustness of Neural Networks, IEEE S&P 2017
Y. Song et al., PixelDefend, ICLR 2018

Table of Contents

Black-Box: Transferability of Adversarial Example

- Some adversarial examples strongly transfer across **different networks**



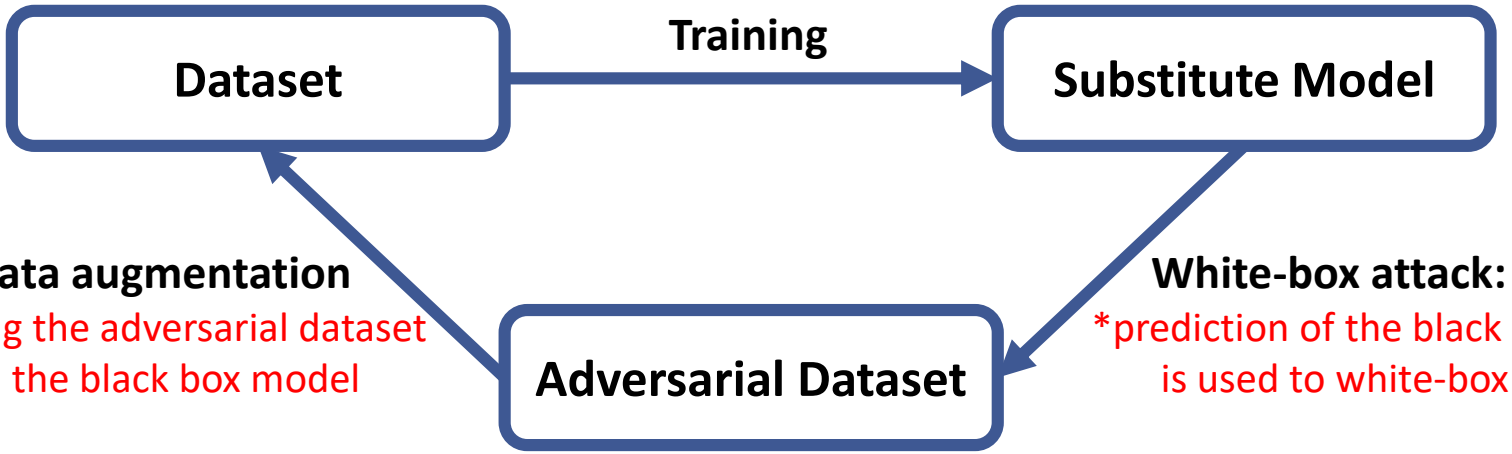
*source: Goodfellow et al., Explaining and Harnessing Adversarial Examples, ICLR 2015

- **Motivation:** The transferability enables us 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
  - **Method:**

$$x' = x + \epsilon \cdot \mathrm{sign}(\nabla_x L(f(x), y_{\mathrm{pred}}))$$

substitute model

black-box prediction



- **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

*source: Papernot et al., Practical Black-Box Attacks against Machine Learning, ACM CCS 2017

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

- **Motivation:** Stronger substitute model using ensemble model?
  - **Idea:** White-box attack to an ensemble of the substitute models

- Consider k substitute models and let $J_1, ..., J_k$ be their softmax outputs.
- For given (x, y), **ensemble black-box attack** objective is the follow:

$$\min_{\delta} \left(-\log \left(1 - \left(\sum_{i=1}^{k} \alpha_i J_i(x+\delta) \right)_y \right) + \lambda d(x, x+\delta) \right)$$

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

  - $d(x, x')$: **Root Mean Square Deviation (RMSD)**

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

*source: Liu et al., Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

- **Experimental Results**
  - Ensemble of modern architecture DNNs
    - "**-X**": an ensemble without the model **X**
  - RMSD: **Root Mean Square Deviation** of adversarial perturbations

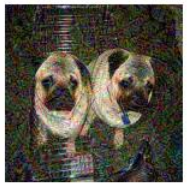
Black-box models

	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

*source: Liu et al., Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

- **Experimental Results**
  - The first successful black-box attack against Clarifai.com, a commercial image classification system

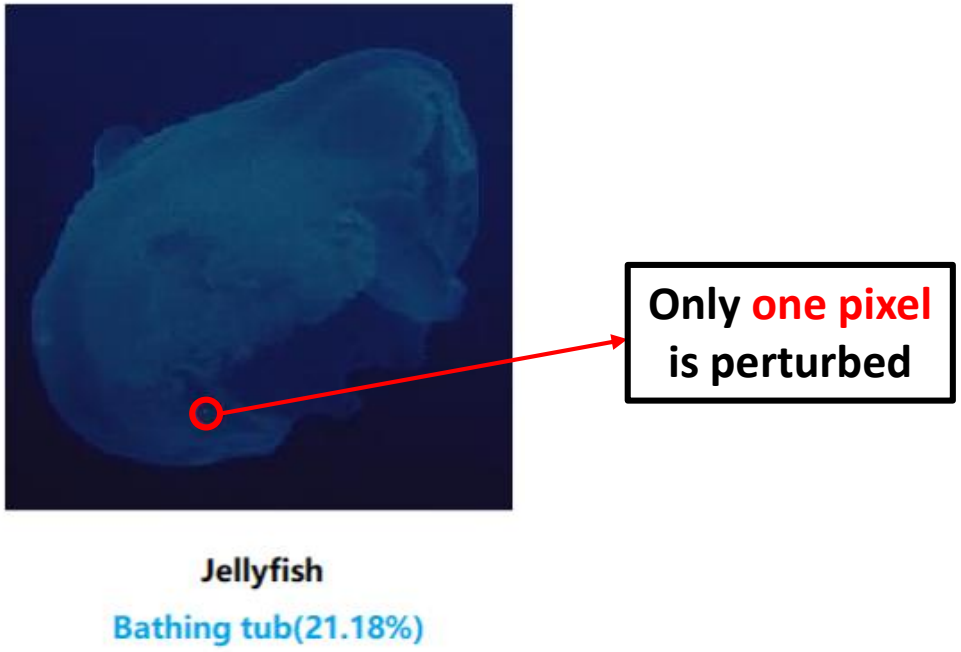
original image	true label	Clarifai.com results of original image	target label	targeted adversarial example	Clarifai.com results of targeted adversarial example
	viaduct	bridge, sight, arch, river, sky	window screen		window, wall, old, decoration, design
	hip, rose hip, rosehip	fruit, fall, food, little, 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

*source: Liu et al., Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

Table of Contents

Unrestricted and Physical Attacks

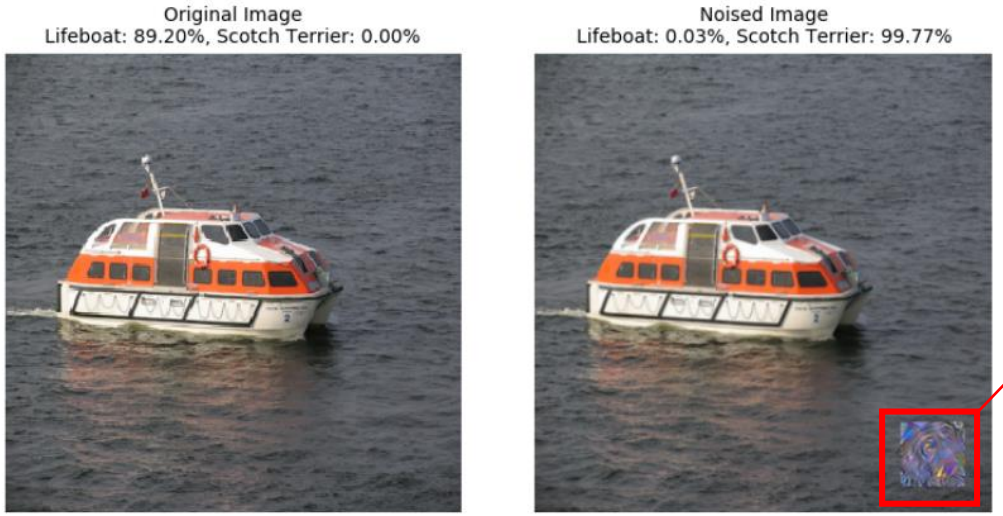
- So far, all we have considered is about restricted attacks
  - An adversary is restricted to **bounded perturbations** (e.g. ℓ_2, ℓ_∞, …)
- However, this threat model is highly limited in real world threats
- There are much more noise types that humans don't aware
  - **Example**: Single-pixel attack [Su et al., 2017]



*source: Su et al., One pixel attack for fooling deep neural networks, arXiv, 2017

Unrestricted and Physical Attacks

- So far, all we have considered is about restricted attacks
  - An adversary is restricted to **bounded perturbations** (e.g. ℓ_2, ℓ_∞, …)
- However, this threat model is highly limited in real world threats
- There are much more noise types that humans don't aware
  - **Example**: Localized & visible noise [Karmon et al., 2018]

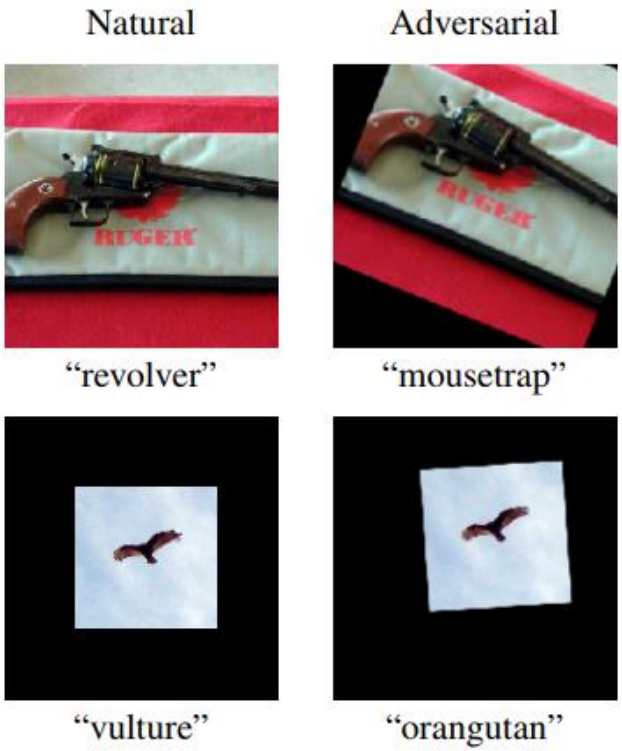


Lifeboat (89.2%) → Scotch Terrier (99.8%)

*source: Karmon et al., LaVAN: Localized and Visible Adversarial Noise, ICML 2018

Unrestricted and Physical Attacks

- So far, all we have considered is about restricted attacks
  - An adversary is restricted to **bounded perturbations** (e.g. ℓ_2, ℓ_∞, …)
- However, this threat model is highly limited in real world threats
- There are much more noise types that humans don't aware
  - **Example**: Rotation & translation [Engstrom et al., 2018]



*source: Engstrom et al., A Rotation and a Translation Suffice: Fooling CNNs with Simple Transformations, arXiv 2017

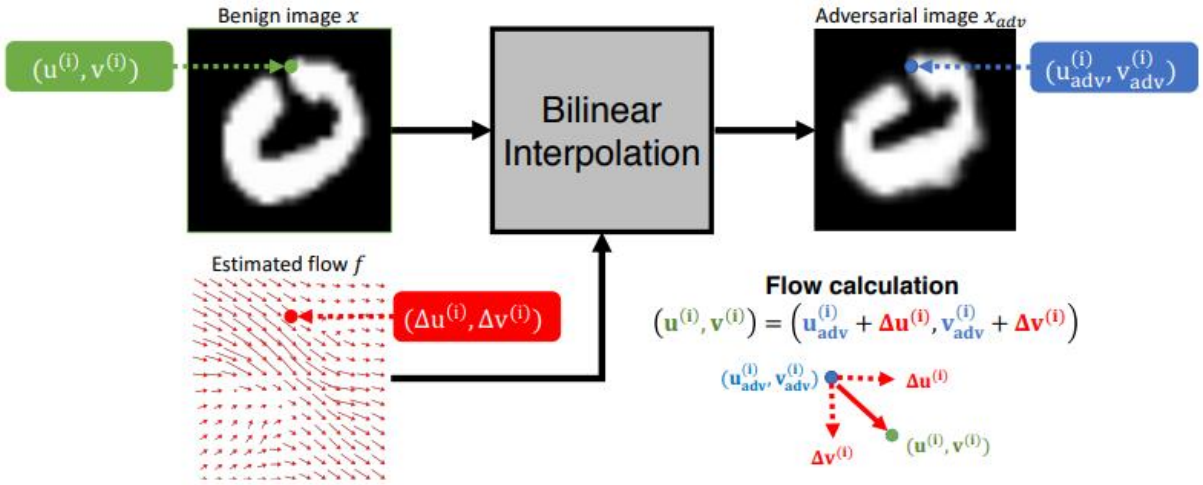
Unrestricted and Physical Attacks

- So far, all we have considered is about restricted attacks
  - An adversary is restricted to **bounded perturbations** (e.g. ℓ_2, ℓ_∞, …)
- However, this threat model is highly limited in real world threats
- There are much more noise types that humans don't aware
- In particular, adversarial attack is possible even using physical perturbation
  - **Example**: Physically designed perturbation [Eykholt et al., 2018]

Real graffiti Simulated perturbation

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

- **Xiao et al. (2018)**: Adversarial example via spatial transformation
  - It has large distance in ℓ_p-measure, but much realistic



- Each pixels is transformed by an optimized flow $f = (\Delta u^{(i)}, \Delta v^{(i)})$
  - f is optimized with L-BFGS solver [Liu & Nocedal, 1989]

$$f^* = \underset{f}{\operatorname{argmin}} \underbrace{L_{\text{adv}}(x, f)}_{\textbf{CW objective}} + \tau \cdot L_{\text{flow}}(f)$$

$$L_{\text{flow}}(f) := \sum_{p:\text{pixels}} \sum_{q:\mathcal{N}(p)} \sqrt{||\Delta u^{(p)} - \Delta u^{(q)}||_2^2 + ||\Delta v^{(p)} - \Delta v^{(q)}||_2^2}$$

"Flow should be smooth over neighbors"

Neighbors of p

*source: Xiao et al., Spatially Transformed Adversarial Examples, ICLR 2018

- **Xiao et al. (2018)**: Adversarial example via spatial transformation

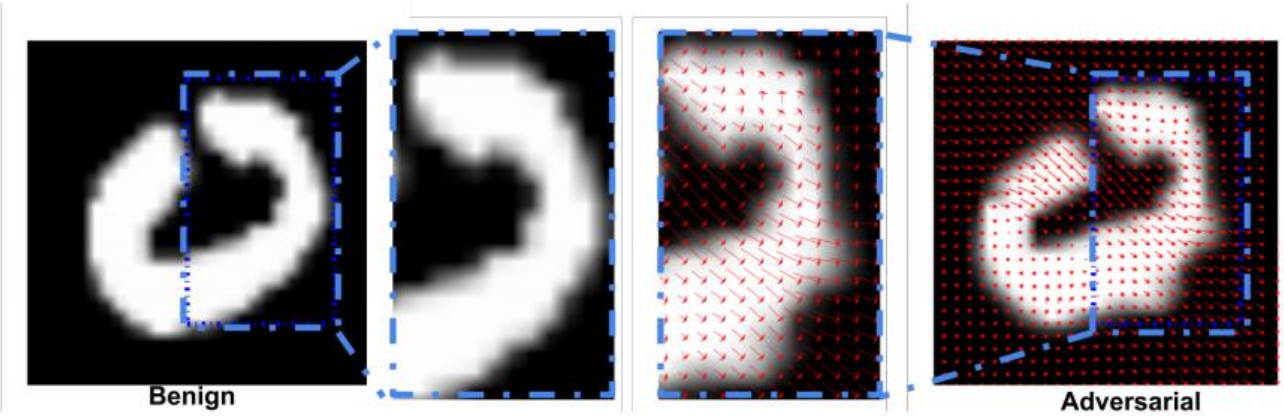


Figure 5: Flow visualization on MNIST. The digit “0" is misclassified as “2".

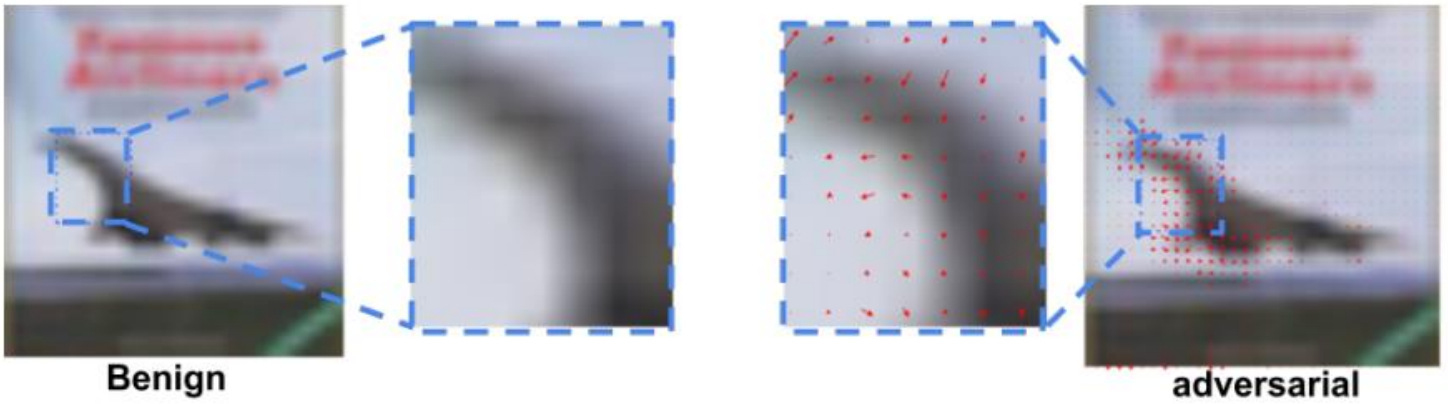
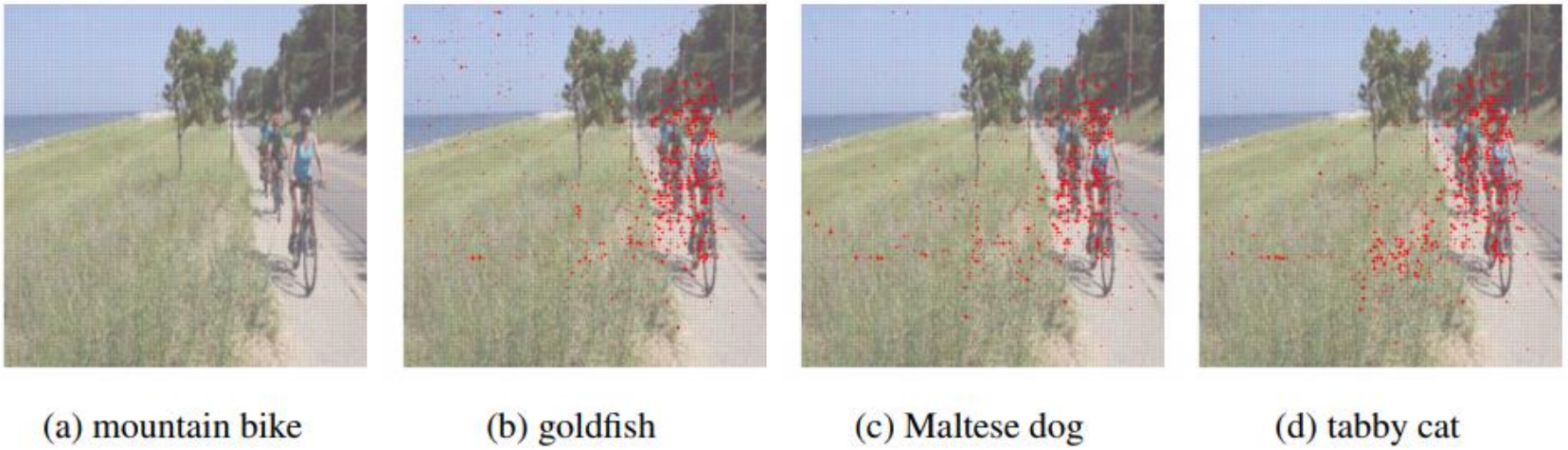


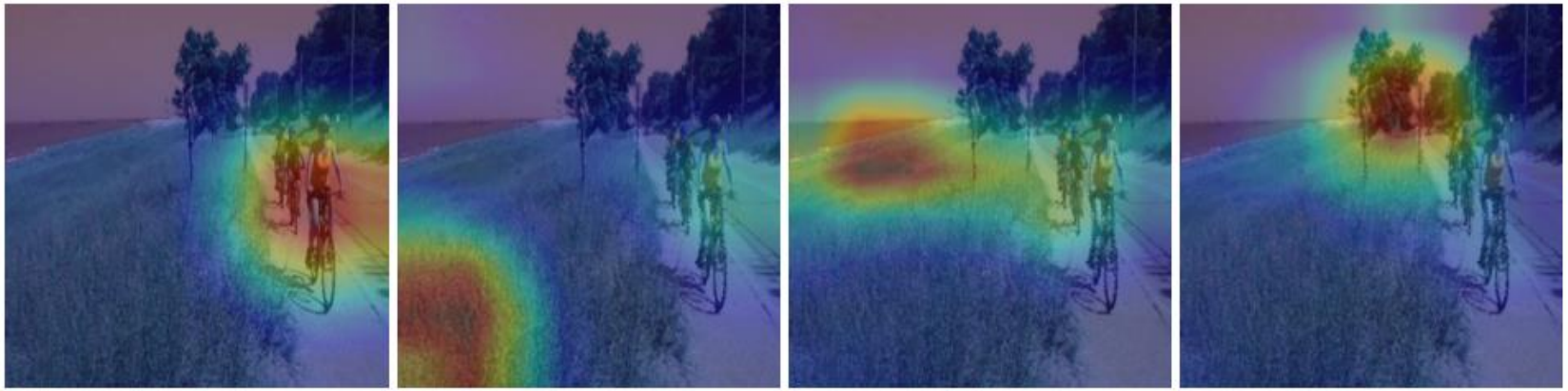
Figure 6: Flow visualization on CIFAR-10. The example is misclassified as bird.

*source: Xiao et al., Spatially Transformed Adversarial Examples, ICLR 2018

- **Xiao et al. (2018)**: Adversarial example via spatial transformation



(a) mountain bike (b) goldfish (c) Maltese dog (d) tabby cat



CAM interpretation for ImageNet Inception-v3 model

*source: Xiao et al., Spatially Transformed Adversarial Examples, ICLR 2018

Table of Contents

Table of Contents

Adversarial Training [Madry et al., 2018]

- **Motivation:** An optimization view on attacks and defenses
  - **Recall:** Adversarial attacks aim to find inputs so that:

$$\max_{x':d(x,x')<\epsilon} L(f(x'), y)$$

  - In the viewpoint of defense, our goal is to minimize the **adversarial risk:**

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{x':d(x,x')<\epsilon} L(f(x'), y)\right]$$

- **Adversarial training** framework aims to minimize adversarial risk in **training**

$$\min_{\theta}\left(\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{x':d(x,x')<\epsilon} L(f(x'), y; \theta)\right]\right)$$

Training parameters

FGSM, PGD, …

  - **Challenge**: Computing the inner-maximization is difficult
  - **Idea**: Use strong attack methods to approximate the inner-maximization
    - e.g. FGSM, PGD, DeepFool, …

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

Adversarial Training [Madry et al., 2018]

- Up to now, **adversarial training** is the only framework that has passed the test-of-time to show its effectiveness against adversarial attack
  - Nowadays, most of "real" defense methods are based on this framework
- **MNIST results**

Method	Steps	Restarts	Source	Accuracy
Natural	-	-	-	98.8%
FGSM	-	-	A	95.6%
PGD	40	1	A	93.2%
PGD	100	1	A	91.8%
PGD	40	20	A	90.4%
PGD	100	20	A	**89.3%**
Targeted	40	1	A	92.7%
CW	40	1	A	94.0%
CW+	40	1	A	93.9%

White-box

Method	Steps	Restarts	Source	Accuracy
FGSM	-	-	A'	96.8%
PGD	40	1	A'	96.0%
PGD	100	20	A'	**95.7%**
CW	40	1	A'	97.0%
CW+	40	1	A'	96.4%
FGSM	-	-	B	**95.4%**
PGD	40	1	B	96.4%
CW+	-	-	B	95.7%

Black-box

- **CIFAR10 results**

Method	Steps	Source	Accuracy
Natural	-	-	87.3%
FGSM	-	A	56.1%
PGD	7	A	50.0%
PGD	20	A	**45.8%**
CW	30	A	46.8%

White-box

Method	Steps	Source	Accuracy
FGSM	-	A'	67.0%
PGD	7	A'	**64.2%**
CW	30	A'	78.7%
FGSM	-	A_{nat}	85.6%
PGD	7	A_{nat}	86.0%

Black-box

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

Adversarial Training [Madry et al., 2018]

- Up to now, **adversarial training** is the only framework that has passed the test-of-time to show its effectiveness against adversarial attack
  - Nowadays, most of "real" defense methods are based on this framework
- Madry et al. also released the "attack challenges" against their trained models
  - **MNIST:** https://github.com/MadryLab/mnist_challenge
  - **CIFAR10:** https://github.com/MadryLab/cifar10_challenge

MNIST white-box leaderboard

Attack	Submitted by	Accuracy	Submission Date
Interval Attacks	Shiqi Wang	**88.42%**	Feb 28, 2019
Distributionally Adversarial Attack merging multiple hyperparameters	Tianhang Zheng	88.56%	Jan 13, 2019
Interval Attacks	Shiqi Wang	88.59%	Jan 6, 2019
Distributionally Adversarial Attack	Tianhang Zheng	88.79%	Aug 13, 2018
First-order attack on logit difference for optimally chosen target label	Samarth Gupta	88.85%	May 23, 2018
100-step PGD on the cross-entropy loss with 50 random restarts	(initial entry)	89.62%	Nov 6, 2017
100-step PGD on the CW loss with 50 random restarts	(initial entry)	89.71%	Nov 6, 2017
100-step PGD on the cross-entropy loss	(initial entry)	92.52%	Nov 6, 2017
100-step PGD on the CW loss	(initial entry)	93.04%	Nov 6, 2017
FGSM on the cross-entropy loss	(initial entry)	96.36%	Nov 6, 2017
FGSM on the CW loss	(initial entry)	96.40%	Nov 6, 2017

CIFAR-10 white-box leaderboard

Attack	Submitted by	Accuracy	Submission Date
FAB: Fast Adaptive Boundary Attack	Francesco Croce	**44.51%**	Jun 7, 2019
Distributionally Adversarial Attack	Tianhang Zheng	44.71%	Aug 21, 2018
20-step PGD on the cross-entropy loss with 10 random restarts	Tianhang Zheng	45.21%	Aug 24, 2018
20-step PGD on the cross-entropy loss	(initial entry)	47.04%	Dec 10, 2017
20-step PGD on the CW loss	(initial entry)	47.76%	Dec 10, 2017
FGSM on the CW loss	(initial entry)	54.92%	Dec 10, 2017
FGSM on the cross-entropy loss	(initial entry)	55.55%	Dec 10, 2017

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

Table of Contents

Large Margin Training [Elsayed et al., 2018]

- **Adversarial training** attempts to minimize the adversarial risk

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{x':d(x,x')<\epsilon} L(f(x'),y)\right]$$

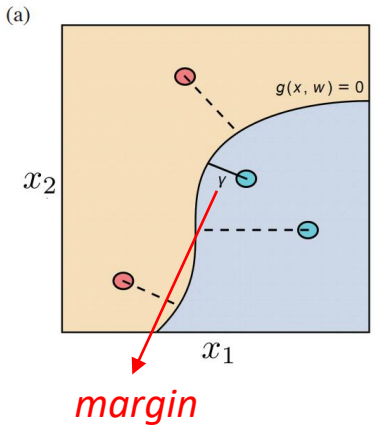
- Similarly, one may want to optimize the another measure of robustness
  - … the average minimum-distance!

$$\max_{\theta}\left(\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\min_{x'\in A_{x,y}} d(x,x')\right]\right)$$

$$\boxed{A_{x,y}=\{x' : f(x')\neq y\}}$$

- **Large margin training** attempts to maximize the margin:
  - the smallest distance from a sample to the decision boundary

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

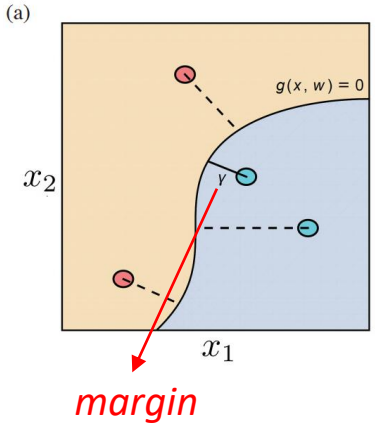


*source: Elsayed et al., Large Margin Deep Networks for Classification, NeurIPS 2018

Large Margin Training [Elsayed et al., 2018]

- **Large margin training** attempts to maximize the margin:
  - the smallest distance from a sample to the decision boundary

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



- Similar to DeepFool [S. Moosavi-Dezfooli et al., 2016], the margin is linearly approximated:

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

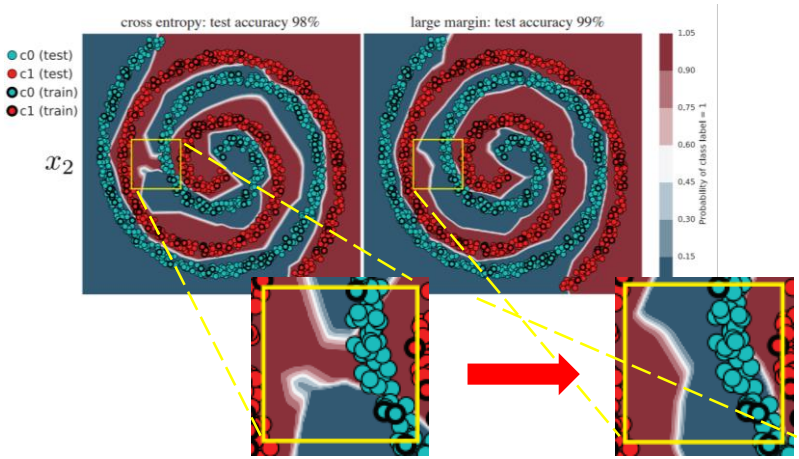
where $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$, $q = \frac{p}{p-1}$

- Based on this, a new loss is proposed:

$$\min_{\theta} \sum_{(x,y)\sim\mathcal{D}} \mathcal{A}_{i\neq y} \left(\gamma + \frac{f_i(x) - f_y(x)}{\|\nabla f_i(x) - \nabla f_y(x)\|} \right)^+$$

margin

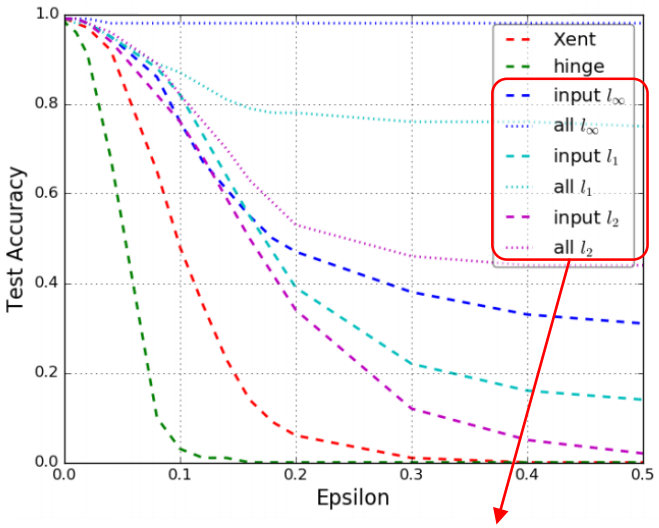
$\mathcal{A}$: aggregate operator; max or sum



*source: Elsayed et al., Large Margin Deep Networks for Classification, NeurIPS 2018

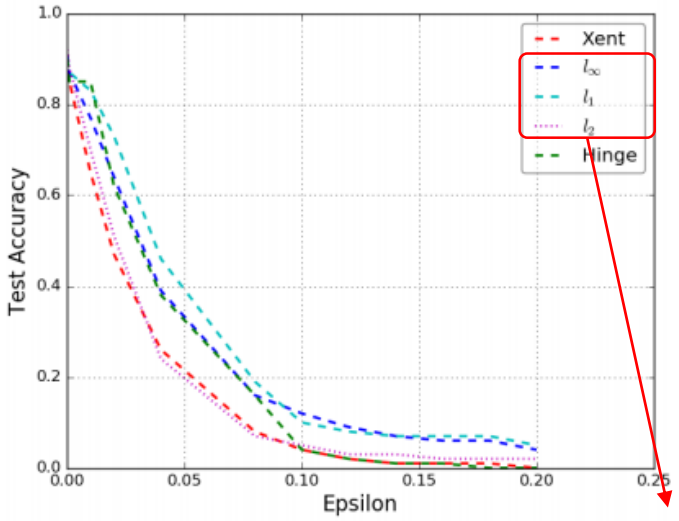
- **Experimental Result**
  - Test accuracy of standard model: 99.5%
  - Test accuracy of the margin classifier models: 99.3~99.5%
  - White-box: BIM attack
    - Xent: Cross-entropy loss

MNIST



Large margin classifiers

CIFAR 10



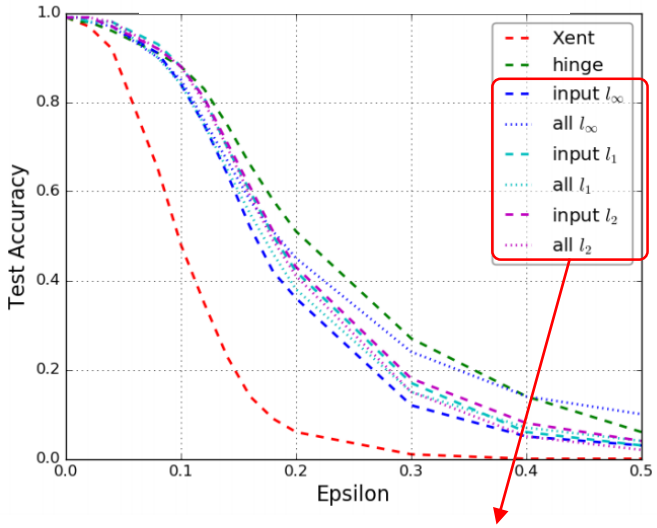
applying on multiple layer
(input, output, some conv layers)

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)

*source: Elsayed et al., Large Margin Deep Networks for Classification, NeurIPS 2018

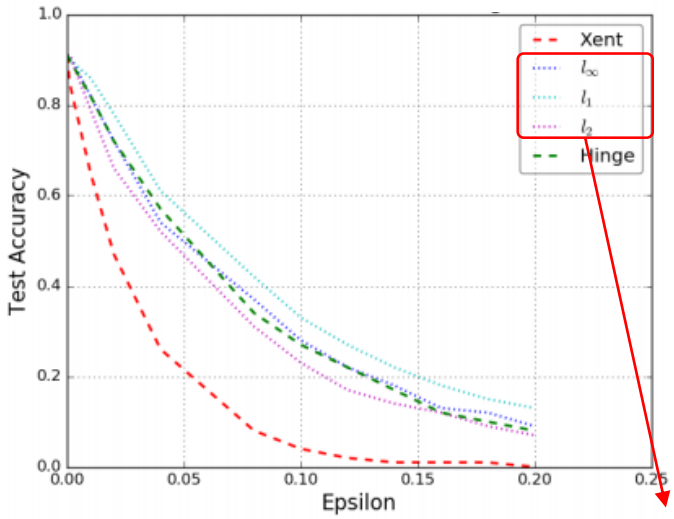
- **Experimental Result**
  - Test accuracy of standard model: 99.5%
  - Test accuracy of the margin classifier models: 99.3~99.5%
  - Black-box: BIM attack to Xent model
    - Xent: Cross-entropy loss

MNIST



Large margin classifiers

CIFAR 10



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

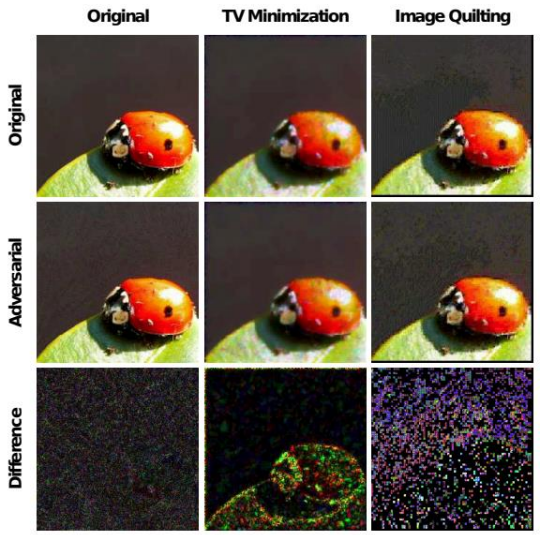
Input l_p: applying the *margin loss* on first layer (input) only
All l_p: applying the *margin loss* on all hidden layer (hidden features)

*source: Elsayed et al., Large Margin Deep Networks for Classification, NeurIPS 2018

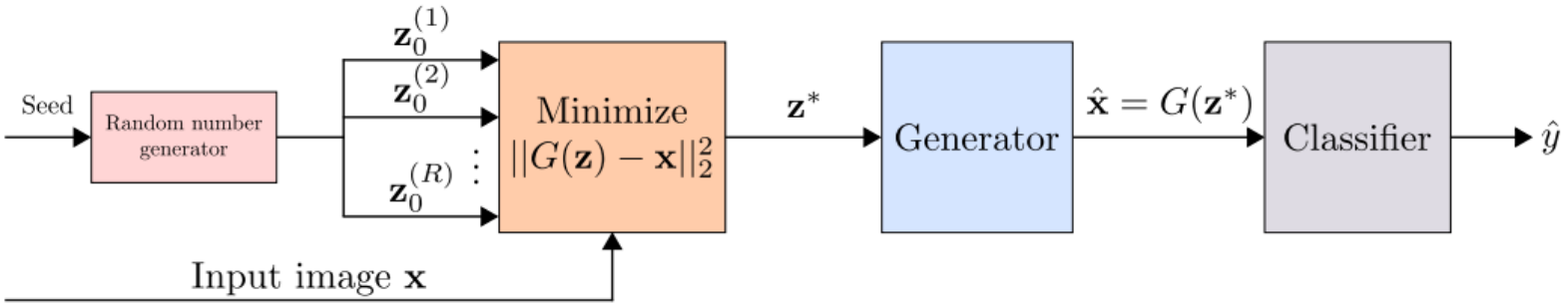
Table of Contents

Obfuscated gradients: False sense of security [Athalye et al., 2018]

- In ICLR 2018, 9 defense papers were published including adversarial training:
  - Adversarial training [Madry et al., 2018]
  - Thermometer Encoding [Buckman et al., 2018]
  - Input Transformations [Guo et al., 2018]
  - Local Intrinsic Dimensionality [Ma et al., 2018]
  - Stochastic Activation Pruning [Dhillon et al., 2018]
  - Defense-GAN [Samangouei et al., 2018]
  - PixelDefend [Song et al., 2018]
  - …



Input transformation [Guo et al., 2018]

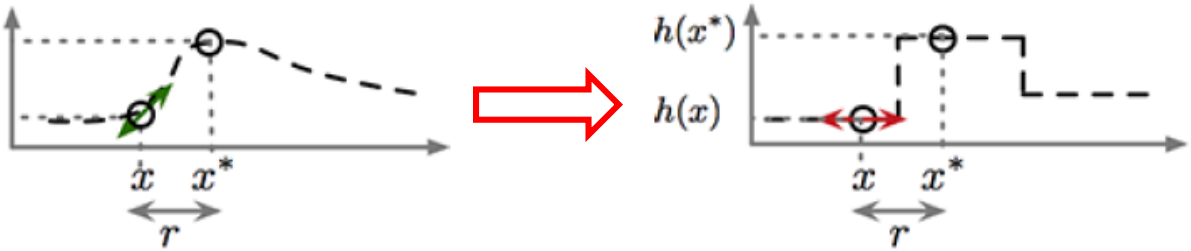


Defense-GAN [Samangouei et al., 2018]

*source: Athalye et al., Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, ICML 2018

Obfuscated gradients: False sense of security [Athalye et al., 2018]

- In ICLR 2018, 9 defense papers were published including adversarial training:
  - Adversarial training [Madry et al., 2018]
  - Thermometer Encoding [Buckman et al., 2018]
  - Input Transformations [Guo et al., 2018]
  - Local Intrinsic Dimensionality [Ma et al., 2018]
  - Stochastic Activation Pruning [Dhillon et al., 2018]
  - Defense-GAN [Samangouei et al., 2018]
  - PixelDefend [Song et al., 2018]
  - …
- **Athalye et al. (ICML 2019)**: In fact, most of them are "fake" defenses
  - **"Fake" defense?**: They don't aim the non-existence of adversarial example
    - Rather, they aim to obfuscate the gradient information
    - Obfuscated gradient makes gradient-based attacks (FGSM, PGD, …) harder



*source: Athalye et al., Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, ICML 2018

Obfuscated gradients: False sense of security [Athalye et al., 2018]

- **Athalye et al. (ICML 2019)**: In fact, most of them are "fake" defenses
  - **"Fake" defense?**: They don't aim the non-existence of adversarial example
    - Rather, they aim to obfuscate the gradient information
    - Obfuscated gradient makes gradient-based attacks (FGSM, PGD, …) harder

  - They identified **three obfuscation techniques** used in the defenses

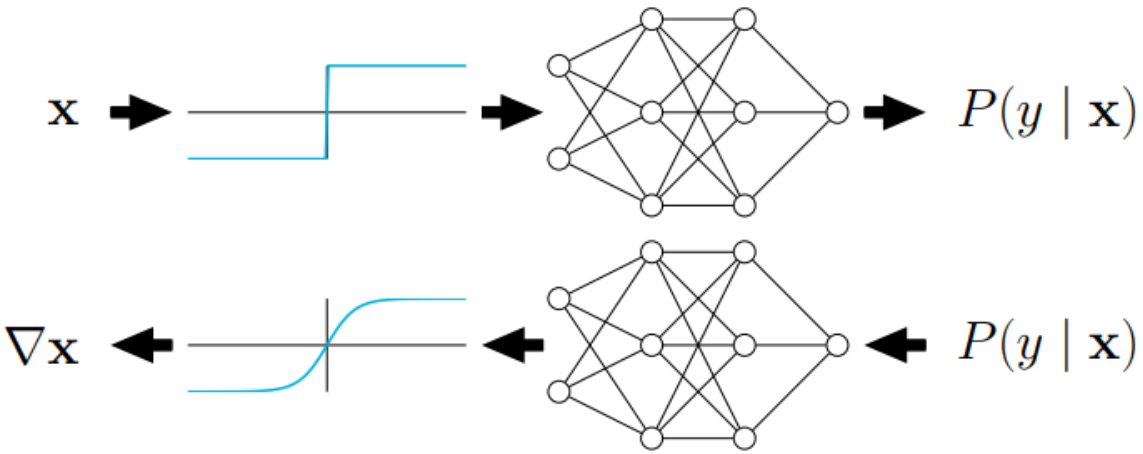
Obfuscation	Defenses
Shattered Gradients	**Existence of a non-differentiable layer**
	• Thermometer Encoding [Buckman et al., 2018] • Input Transformation [Guo et al., 2018] • Local Intrinsic Dimensionality (LID) [Ma et al., 2018]
Stochastic Gradients	**Artificial randomness on computing gradient**
	• Stochastic Activation Pruning (SAP) [Dhillon et al., 2018] • Mitigating Through Randomization [Xie et al., 2018]
Exploding & Vanishing Gradients	**Multiple iterations, or extremely deep DNN**
	• Pixel Defend [Song et al., 2018] • Defense-GAN [Samangouei et al., 2018]

*source: Athalye et al., Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, ICML 2018

Obfuscated gradients: False sense of security [Athalye et al., 2018]

- **Athalye et al. (ICML 2019)**: In fact, most of them are "fake" defenses
  - **"Fake" defense?**: They don't aim the non-existence of adversarial example
    - Rather, they aim to obfuscate the gradient information
    - Obfuscated gradient makes gradient-based attacks (FGSM, PGD, …) harder

- Those kinds of defenses can be easily bypassed by **3 simple tricks**
  1. **Backward Pass Differentiable Approximation (BPDA)**
     - Replace the non-differentiable parts only at backward pass
     - Use some differentiable approximative function



*source: Athalye et al., Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, ICML 2018

Obfuscated gradients: False sense of security [Athalye et al., 2018]

- **Athalye et al. (ICML 2019)**: In fact, most of them are "fake" defenses
  - **"Fake" defense?**: They don't aim the non-existence of adversarial example
    - Rather, they aim to obfuscate the gradient information
    - Obfuscated gradient makes gradient-based attacks (FGSM, PGD, …) harder

- Those kinds of defenses can be easily bypassed by **3 simple tricks**

2. Expectation Over Transformation (EOT)

- Take the expectation of attacks to mitigate stochastic defenses

$$\max_{x':d(x,x')<\epsilon} \mathbb{E}_{t\sim T}[L(f(t(x')),y)]$$

Random transformation

3. Reparameterization

- Replace deep or recurrent parts by simpler differentiable function

*source: Athalye et al., Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, ICML 2018

Obfuscated gradients: False sense of security [Athalye et al., 2018]

- **Athalye et al. (ICML 2019)**: In fact, most of them are "fake" defenses
  - **"Fake" defense?**: They don't aim the non-existence of adversarial example
    - Rather, they aim to obfuscate the gradient information
    - Obfuscated gradient makes gradient-based attacks (FGSM, PGD, …) harder

- Those kinds of defenses can be easily bypassed by **3 simple tricks**
  - **6 of the 9** defense papers were completely broken using those tricks
  - 1 of the 9 was partially broken (Defense-GAN)
  - Adversarial training [Madry et al. 2018; Na et al., 2018] were the only survivals

Defense	Type	Behavior	Attack technique
Thermometer Encoding	Shattered	Black-box is better	BPDA
Local Intrinsic Dimensionality (LID)	Shattered	Unbounded attack do not reach 100%	BPDA
Input Transformation	Shattered	Black-box is better	BPDA, EOT
Stochastic Activation Pruning (SAP)	Stochastic, Exploding	·	modified EOT
Mitigating Through Randomization	Stochastic	·	EOT
Pixel Defend	Vanishing	·	BPDA
Defense-GAN	Vanishing	Unbounded attack do not reach 100%	BPDA

*source: Athalye et al., Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, ICML 2018

Obfuscated gradients: False sense of security [Athalye et al., 2018]

- **Athalye et al. (ICML 2019)**: In fact, most of them are "fake" defenses
  - **"Fake" defense?**: They don't aim the non-existence of adversarial example
    - Rather, they aim to obfuscate the gradient information
    - Obfuscated gradient makes gradient-based attacks (FGSM, PGD, …) harder

- Then… what should we do?
  - At least, we have to do sanity checks on evaluating defenses
  - Do your best to show that the proposed defense is a "real" defense

  - **Some "red-flags" indicating obfuscated gradients**

> ① One-step attacks perform better than iterative attacks
>
> ② Black-box attacks are better than white-box attacks
>
> ③ Unbounded attacks do not reach 100% success
>
> ④ Random sampling finds adversarial examples better

*source: Athalye et al., Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, ICML 2018

Table of Contents

Certified Robustness via Wasserstein Adversarial Training [Sinha et al., 2018]

- In **adversarial training**, the inner-maximization is solved via existing attacks

$$\min_{\theta} \left(\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{x':d(x,x')<\epsilon} L(f(x'), y; \theta) \right] \right)$$

FGSM, PGD, …

- **Challenge**: Attack methods do not fully solve the inner-maximization
  - Still, there will be a practical gap between the optimal worst-case loss
  - More stronger adversary? → Much more expensive to compute

- **Motivation:** Adversarial training with a rigorous guarantee?
  - To this end, **Wasserstein adversarial training** considers distributional robustness

$$\min_{\theta} \left(\sup_{P:W_c(P_0,P)\leq\rho} \mathbb{E}_{(X,Y)\sim P} \left[L(f(X), Y; \theta) \right] \right)$$

"Wasserstein ball"

Original data distribution

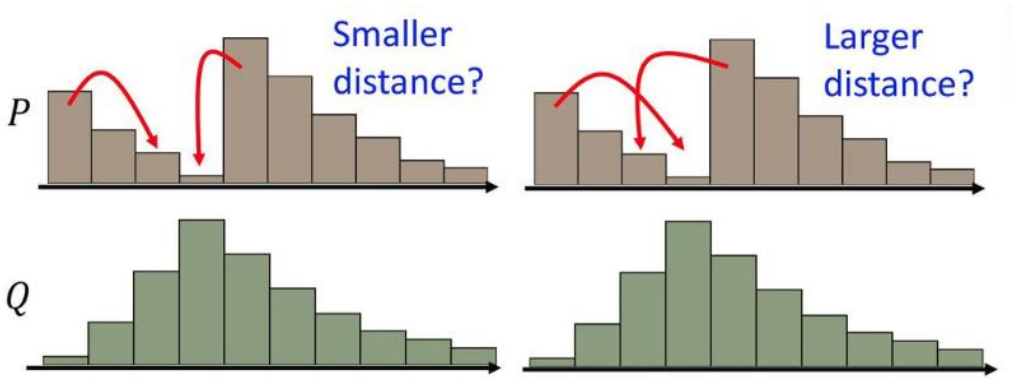
*source: Shina et al., Certifying Some Distributional Robustness with Principled Adversarial Training, ICLR 2018

- **Wasserstein metric** W_c: The avg. cost to move a distribution P to Q

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

(X,Y)

  - W_c specifies a cost function: $c(z,z') := \|x - x'\|_p^2 + \infty \cdot 1_{y \neq y'}$



- Next, we take the Lagrangian dual form of the original objective

$$\min_\theta \left(\sup_{P: W_c(P_0,P) \leq \rho} \mathbb{E}_{(X,Y) \sim P} [L(f(X),Y;\theta)] \right)$$

$$\Rightarrow \quad \min_\theta \left(\sup_P \mathbb{E}_P [L(f(X),Y;\theta) - \gamma W_c(P,P_0)] \right)$$

*source:
https://slideplayer.com/slide/12699282/
Shina et al., Certifying Some Distributional Robustness with Principled Adversarial Training, ICLR 2018

- Next, we take the Lagrangian dual form of the original objective

$$\min_{\theta}\left(\sup_{P:W_c(P_0,P)\leq\rho}\mathbb{E}_{(X,Y)\sim P}\left[L(f(X),Y;\theta)\right]\right)$$

$$\Rightarrow \quad \min_{\theta}\left(\sup_{P}\mathbb{E}_{P}\left[L(f(X),Y;\theta)-\gamma W_c(P,P_0)\right]\right)$$

- Then, [J. Blanchet et al., 2016] induces the form to the relaxed objective to

$$\Rightarrow \quad \min_{\theta}\left(\mathbb{E}_{P_0}\left[\sup_{z\in\mathcal{Z}}\{L(z;\theta)-\gamma c(z,z_0)\}\right]\right)$$

(x,y)

  - This is the final objective of Wasserstein adversarial training

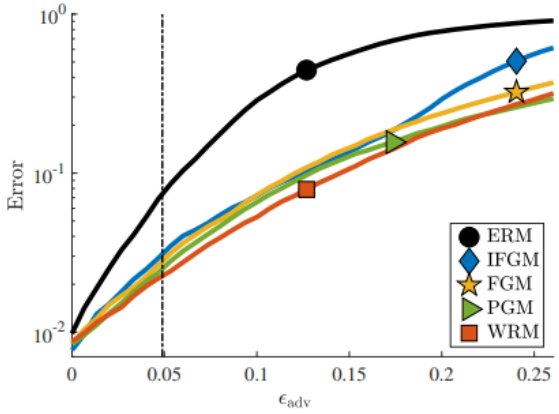
Algorithm 1 Distributionally robust optimization with adversarial training

INPUT: Sampling distribution P_0, constraint sets Θ 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 ϵ-approximate maximizer $\widehat{z}^t$ of $\ell(\theta^t; z) - \gamma c(z, z^t)$
  $\theta^{t+1} \leftarrow \mathrm{Proj}_{\Theta}(\theta^t - \alpha_t \nabla_\theta \ell(\theta^t; \widehat{z}^t))$

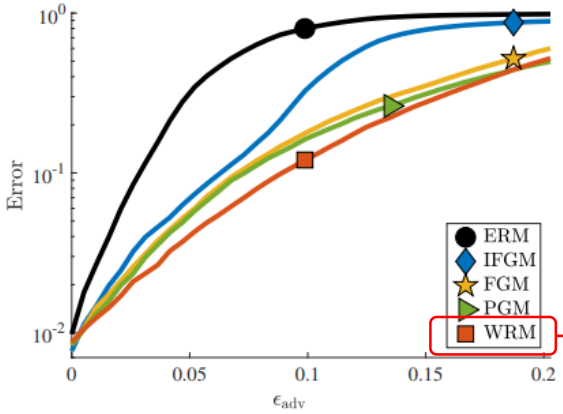
*source: Shina et al., Certifying Some Distributional Robustness with Principled Adversarial Training, ICLR 2018

Certified Robustness via Wasserstein Adversarial Training [Sinha et al., 2018]

- **Experimental Results:** White-box attack with l_2 and l_∞ metric
  - Wasserstein adversarial training (**WRM**) outperform the baselines



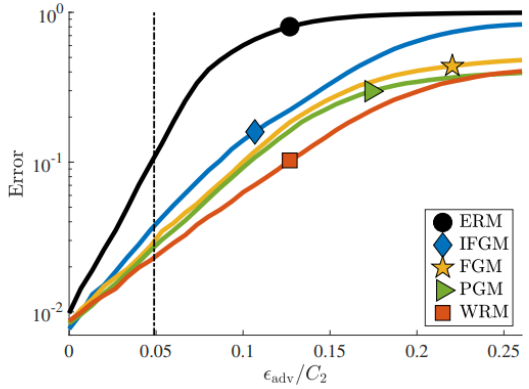
(a) Test error vs. ϵ_{adv} for $\|\cdot\|_2$-FGM attack



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

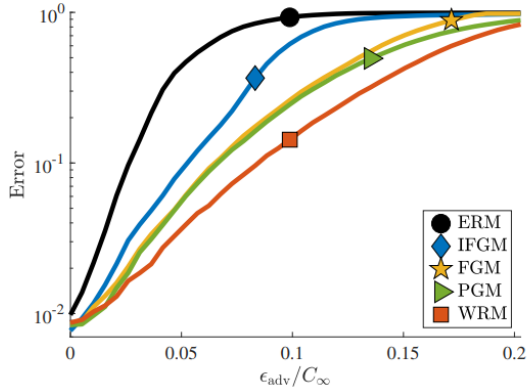
Best performance among the baselines

*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

* PGD attack



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

* PGD attack

*source: Shina et al., Certifying Some Distributional Robustness with Principled Adversarial Training, ICLR 2018

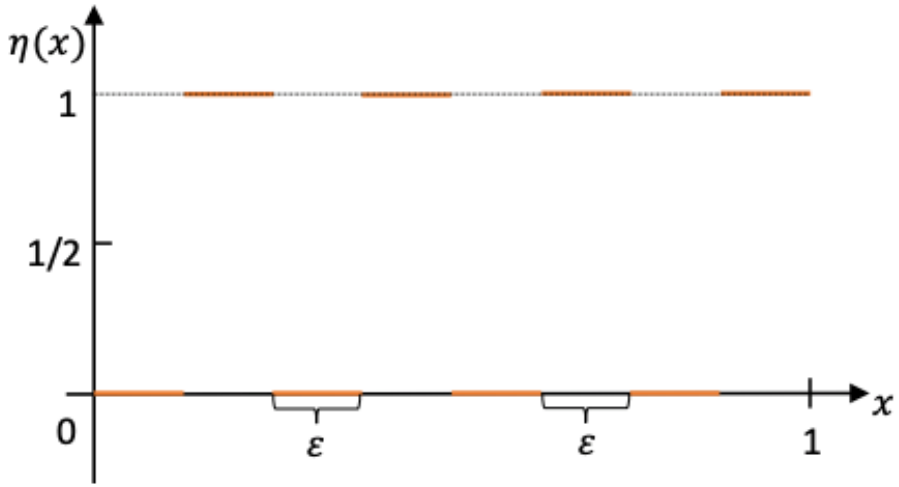
Table of Contents

Tradeoff between accuracy and robustness [Zhang et al., 2019]

- **Motivation**: Robust model → accuracy reduction? [Tsipras et al. ,2019]

- Consider (X, Y) modeled by $\eta(x)$
  - Bayes optimal classifier: $\text{sign}(2\eta(x) - 1)$
- We are using an "accuracy-biased" loss function

- Can we exploit this **trade-off** for better robustness?



$$\eta(x) := \Pr(Y = 1 | X = x)$$
$$= \begin{cases} 0, & x \in [2k\epsilon, (2k+1)\epsilon), \\ 1, & x \in ((2k+1)\epsilon, (2k+2)\epsilon]. \end{cases}$$

	Bayes Optimal Classifier	All-One Classifier
$\mathcal{R}_{\text{nat}}$	0 (optimal)	1/2
$\mathcal{R}_{\text{bdy}}$	1	0
$\mathcal{R}_{\text{rob}}$	1	1/2 (optimal)

*source: Zhang et al., Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2018

Tradeoff between accuracy and robustness [Zhang et al., 2019]

- We re-write the relationship between **robust error** and **natural error**
- Consider a binary classification with $Y \in \{-1, 1\}$
  - **Natural error**: $\mathcal{R}_{\text{nat}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}[f(\boldsymbol{X})Y \leq 0]$
  - **Robust error** under ϵ-perturbation:
    - [Schmidt et al., 2018; Cullina et al., 2018; Bubeck et al., 2018]

$$\mathcal{R}_{\text{rob}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}[\exists \boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon) \text{ s.t. } f(\boldsymbol{X}')Y \leq 0]$$

*source: Zhang et al., Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2018

Tradeoff between accuracy and robustness [Zhang et al., 2019]

- We re-write the relationship between **robust error** and **natural error**
- Consider a binary classification with $Y \in \{-1, 1\}$
  - **Natural error**: $\mathcal{R}_{\text{nat}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}[f(\boldsymbol{X})Y \leq 0]$
  - **Robust error** under ϵ-perturbation:
    - [Schmidt et al., 2018; Cullina et al., 2018; Bubeck et al., 2018]

$$\mathcal{R}_{\text{rob}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}[\exists \boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon) \text{ s.t. } f(\boldsymbol{X}')Y \leq 0]$$

- Zhang et al. (2019) also defines the **boundary error:**

$$\mathcal{R}_{\text{bdy}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}[\boldsymbol{X} \in \mathbb{B}(\text{DB}(f), \epsilon), f(\boldsymbol{X})Y > 0]$$

Decision boundary

- $\mathbb{B}(\text{DB}(f), \epsilon) := \{\boldsymbol{x} : \exists \boldsymbol{x}' \in \mathbb{B}(\boldsymbol{x}, \epsilon) \text{ s.t. } f(\boldsymbol{x})f(\boldsymbol{x}') \leq 0\}$

- **Boundary error** identifies the gap between $\mathcal{R}_{\text{rob}}(f)$ and $\mathcal{R}_{\text{nat}}(f)$

$$\mathcal{R}_{\text{rob}}(f) = \mathcal{R}_{\text{nat}}(f) + \mathcal{R}_{\text{bdy}}(f)$$

*source: Zhang et al., Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2018

Tradeoff between accuracy and robustness [Zhang et al., 2019]

- **Goal**: Find $\hat{f}$ such that $\mathcal{R}_{\text{rob}}(\hat{f}) - \mathcal{R}^*_{\text{nat}}$ is small

$$\mathcal{R}_{\text{rob}}(\hat{f}) - \mathcal{R}^*_{\text{nat}} = \underbrace{(\mathcal{R}_{\text{nat}}(\hat{f}) - \mathcal{R}^*_{\text{nat}})}_{\text{Natural error gap}} + \mathcal{R}_{\text{bdy}}(\hat{f}) \leq \delta$$

- **Theorem 1 (upper bound, informal).** Let ϕ be a usual surrogate loss. We have:

$$\begin{aligned}
\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}} &= (\mathcal{R}_{\text{nat}}(f) - \mathcal{R}^*_{\text{nat}}) + \mathcal{R}_{\text{bdy}}(f) \\
&\leq (\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathcal{R}_{\text{bdy}(f)} \quad \textbf{(Bartlett et al., 2006)} \\
&\leq (\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathbb{E}\left[\max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\underline{\lambda})\right]
\end{aligned}$$

- **Theorem 2 (lower bound, informal).** for any $\xi > 0$, there exist $\mathcal{D}$, f, and $\lambda > 0$ such that:

$$\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}} \geq (\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathbb{E}\left[\max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)\right] - \xi$$

  - The upper bound is tight if there is no assumption on $\mathcal{D}$

*source: Zhang et al., Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2018

Tradeoff between accuracy and robustness [Zhang et al., 2019]

- **Goal**: Find $\hat{f}$ such that $\mathcal{R}_{\text{rob}}(\hat{f}) - \mathcal{R}^*_{\text{nat}}$ is small

$$\mathcal{R}_{\text{rob}}(\hat{f}) - \mathcal{R}^*_{\text{nat}} = (\mathcal{R}_{\text{nat}}(\hat{f}) - \mathcal{R}^*_{\text{nat}}) + \mathcal{R}_{\text{bdy}}(\hat{f}) \leq \delta$$

- The theorems naturally suggests **a new surrogate loss**:

$$\min_f \mathbb{E}\left[\underbrace{\mathcal{L}(f(\boldsymbol{X}), \boldsymbol{Y})}_{\text{accuracy}} + \underbrace{\max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \mathcal{L}(f(\boldsymbol{X}), f(\boldsymbol{X}'))/\lambda}_{\text{robustness}}\right]$$

- **TRADES**: **TR**adeoff-inspired **A**dv. **DE**fense via **S**urrogate-loss minimization
  - λ: The **balancing** hyper-parameter
    - We can boost the robust accuracy with little loss of natural accuracy
- **Key difference**: TRADES finds $\boldsymbol{X}'$ by solving $\max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} L(f(\boldsymbol{X}), f(\boldsymbol{X}'))/\lambda$
  - Adversarial training [Madry et al., 2018]: $\max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} L(f(\boldsymbol{X}'), Y)$

- Up to now, **TRADES** is regarded as the **state-of-the-art** defense method

*source: Zhang et al., Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2018

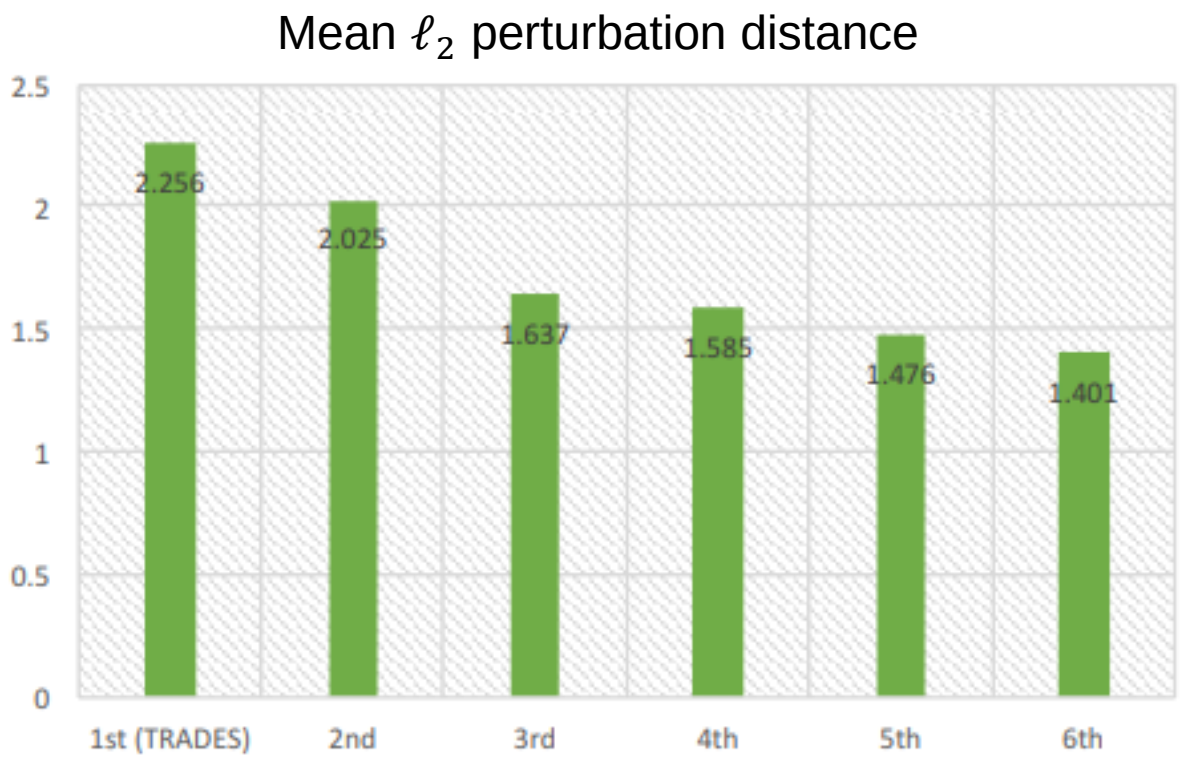
- **Experimental results**
  - White-box attack results (CIFAR-10 & MNIST)

Table 5: Comparisons of TRADES with prior defense models under white-box attacks.

Defense	Defense type	Under which attack	Dataset	Distance	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$
[BRRG18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	0%
[MLW+18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	5%
[DAL+18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	0%
[SKN+18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	9%
[NKM17]	gradient mask	[ACW18]	CIFAR10	0.015 (ℓ_∞)	-	15%
[WSMK18]	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	27.07%	23.54%
[MMS+18]	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	87.30%	**47.04%**
[ZSLG16]	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	94.64%	0.15%
[KGB17]	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	85.25%	45.89%
[RDV17]	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	95.34%	0%
TRADES ($1/\lambda = 1$)	regularization	FGSM^{1,000} (PGD)	CIFAR10	0.031 (ℓ_∞)	88.64%	48.90%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{1,000} (PGD)	CIFAR10	0.031 (ℓ_∞)	84.92%	**56.43%**
TRADES ($1/\lambda = 1$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	88.64%	49.14%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	84.92%	**56.61%**
TRADES ($1/\lambda = 1$)	regularization	DeepFool (ℓ_∞)	CIFAR10	0.031 (ℓ_∞)	88.64%	59.10%
TRADES ($1/\lambda = 6$)	regularization	DeepFool (ℓ_∞)	CIFAR10	0.031 (ℓ_∞)	84.92%	61.38%
TRADES ($1/\lambda = 1$)	regularization	LBFGSAttack	CIFAR10	0.031 (ℓ_∞)	88.64%	84.41%
TRADES ($1/\lambda = 6$)	regularization	LBFGSAttack	CIFAR10	0.031 (ℓ_∞)	84.92%	81.58%
TRADES ($1/\lambda = 1$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	88.64%	51.26%
TRADES ($1/\lambda = 6$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	84.92%	57.95%
TRADES ($1/\lambda = 1$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	88.64%	84.03%
TRADES ($1/\lambda = 6$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	84.92%	81.24%
[SKC18]	gradient mask	[ACW18]	MNIST	0.005 (ℓ_2)	-	55%
[MMS+18]	robust opt.	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.36%	96.01%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{1,000} (PGD)	MNIST	0.3 (ℓ_∞)	99.48%	95.60%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.48%	96.07%
TRADES ($1/\lambda = 6$)	regularization	C&W	MNIST	0.005 (ℓ_2)	99.48%	99.46%

*source: Zhang et al., Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2018

- **Experimental results**
  - NeurIPS 2018 Adversarial Vision Challenge
    - Black-box setting on Tiny-ImageNet dataset
    - Attacks are generated from the top-5 entries in the attack track
    - TRADES surpassed the runner-up by 11.41%



*source: Zhang et al., Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2018

Summary

- **Adversarial examples** are one of the biggest problems that makes harder to deploy deep learning models into real-world
  - Especially on error-sensitive applications: Autonomous driving

- The literature of adversarial example commonly stated in **security perspective**
  - Defining a feasible & realistic threat model is important

- Attack methods are evolving across various threat models
  - White-box attacks are mainly based on the gradient of model
  - Transferability of adversarial examples allow black-box attack
  - Unrestricted and physical attacks are gaining attention

- Up to now, **adversarial training** is the only framework that has passed the test-of-time to show its effectiveness against adversarial attack

References

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

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

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

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

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

[Carlini et al., 2019] On Evaluating Adversarial Robustness, arXiv 2019
https://arxiv.org/abs/1902.06705

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

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

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

[Goodfellow et al., 2015] Explaining and Harnessing Adversarial Examples, ICLR 2015
https://arxiv.org/abs/1412.6572

References

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

[Kurakin et al., 2017a] Adversarial Examples in the Physical World, ICLR Workshop 2017
https://arxiv.org/abs/1607.02533

[Kurakin et al., 2017b] Adversarial Machine Learning at Scale, ICLR 2017
https://arxiv.org/abs/1611.01236

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

[Liu et al., 2017] Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017
https://arxiv.org/abs/1611.02770

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

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

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

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

[Qin et al., 2019] Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition, ICML 2019
http://proceedings.mlr.press/v97/qin19a.html

References

[Samangouei et al., 2018] Defense-GAN: Protecting Classifiers Against Adversarial Attacks using Generative Models, ICLR 2018
https://arxiv.org/abs/1805.06605

[Sinha et al., 2018] Certifying Some Distributional Robustness with Principled Adversarial Training, ICLR 2018
https://arxiv.org/abs/1710.10571

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

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

[Xiao et al., 2018] Spatially Transformed Adversarial Examples, ICLR 2018
https://arxiv.org/abs/1801.02612

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

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

[Zhang et al., 2019] Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2019
https://arxiv.org/abs/1901.08573