Adversarial Defense

May 20, 2020

Dr. Wei Wei, Prof. James Landay

CS 335: Fair, Accountable, and Transparent (FAccT) Deep Learning

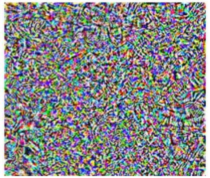
Stanford University

Recap

Evasion Attacks

Natural Sample

+

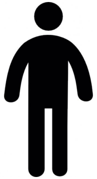


Perturbations

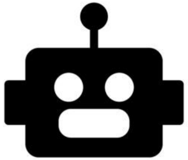
=

Adversarial Sample

Adversarial Sample



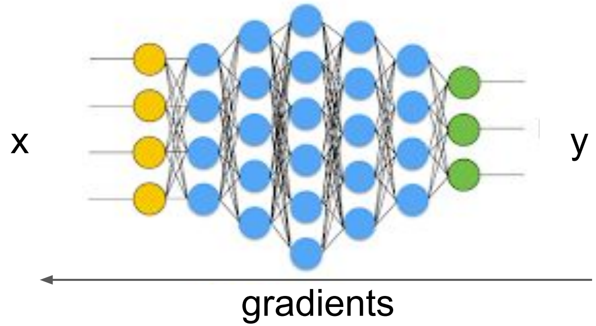
pig



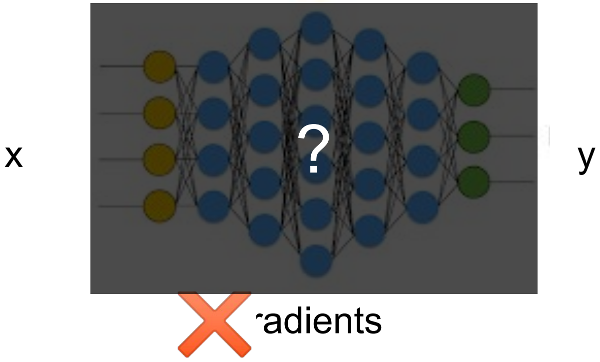
airliner

Recap

White-box Setting



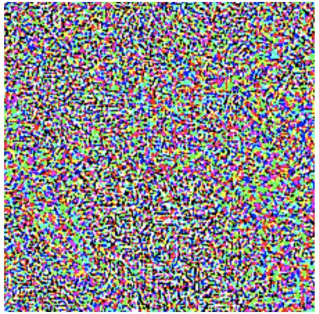
Black-box Setting



Recap

Untargeted Attack

$+ .007 \times$



$=$

$\boldsymbol{x}$

"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

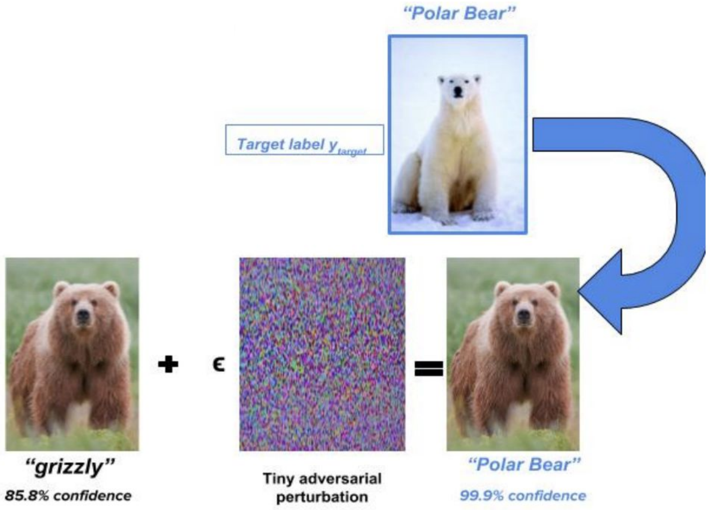
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

[Goodfellow et al, 2015]

Recap

Targeted Attack

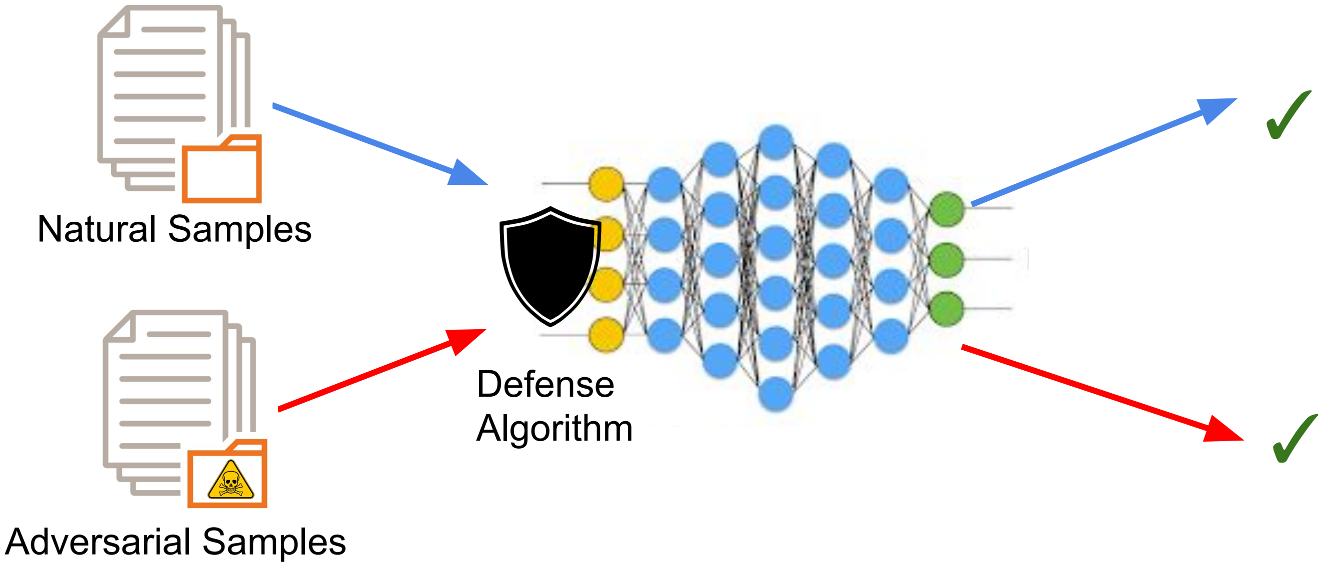


Younis et al, 2019

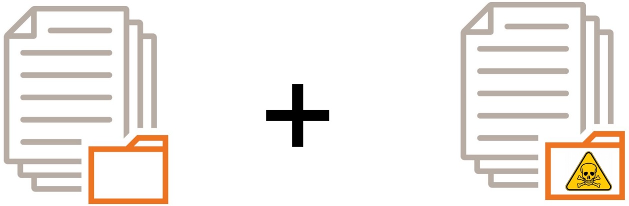
Outline

- Adversarial Defense
- Defense Strategies
    - Adversarial Training
    - Input Transformations
    - Stochastic Gradients
- Obfuscated Gradients and BPDA
- Robust Optimization
- Certified Defense

Adversarial Defense



Adversarial Training



Natural Samples + Adversarial Samples

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x^{adv}, y)$$

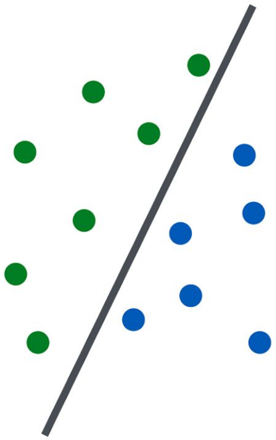
Loss Function

Natural Samples

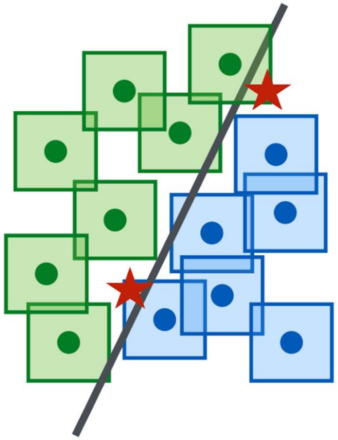
Adversarial Samples

[Goodfellow et al, 2014](https://arxiv.org/abs/1412.6572)

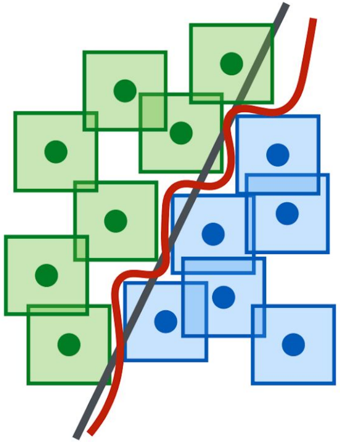
Adversarial Training



Natural Samples



Natural Samples with L_∞ Perturbation Space



Adversarial Training

[Madry et al, 2017](#)

Results on FGSM

- Accuracy on Adversarial Examples

FGSM

$$\boldsymbol{X}^{adv} = \boldsymbol{X} + \epsilon\,\mathrm{sign}\big(\nabla_X J(\boldsymbol{X}, y_{true})\big)$$

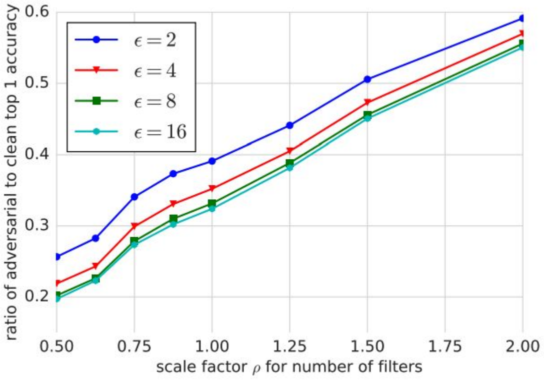
		Clean	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$
Baseline (standard training)	top 1	78.4%	30.8%	27.2%	27.2%	29.5%
	top 5	94.0%	60.0%	55.6%	55.1%	57.2%
Adv. training	top 1	77.6%	73.5%	74.0%	74.5%	73.9%
	top 5	93.8%	91.7%	91.9%	92.0%	91.4%
Deeper model (standard training)	top 1	78.7%	33.5%	30.0%	30.0%	31.6%
	top 5	94.4%	63.3%	58.9%	58.1%	59.5%
Deeper model (Adv. training)	top 1	78.1%	75.4%	75.7%	75.6%	74.4%
	top 5	94.1%	92.6%	92.7%	92.5%	91.6%

Kurakin et al, 2017

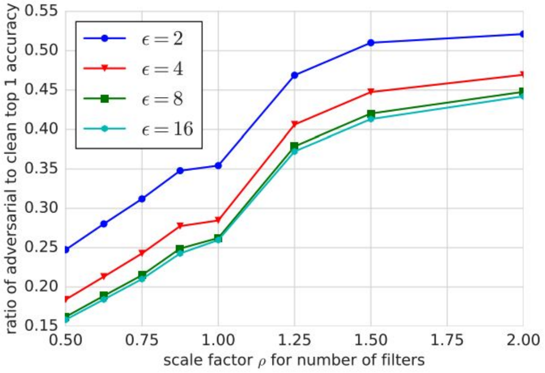
Dataset: ImageNet

Results on FGSM

- Adversarial Accuracy / Clean Image Accuracy
  - Ratio -> 1 successful adversarial attack
  - Ratio -> 0 successful adversarial defense



No adversarial training, “basic iter.” adv. examples



With adversarial training, “basic iter.” adv. examples

fast - FGSM
basic iter. - iterative untargeted FGSM

Kurakin et al, 2017

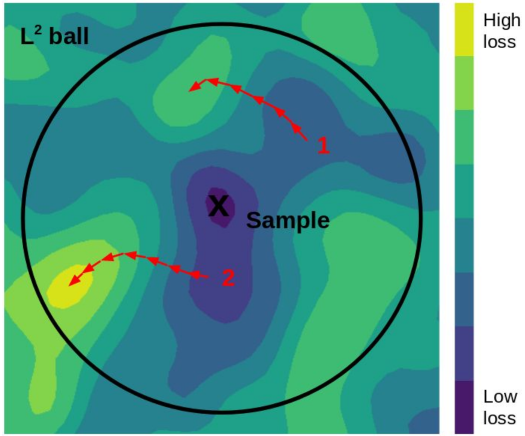
Flexibility

- Plug-in any attack techniques

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1-\alpha) J(\theta, x^{adv}, y)$$

- Examples
  - FGSM
  - Projected Gradient Descent (PGD) ([Madry et al, 2017](#))

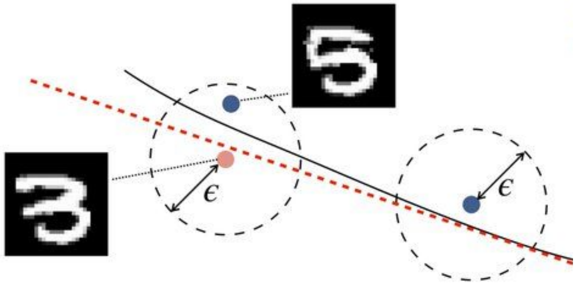
$$\max_{\mathbf{x}' : ||\mathbf{x}' - \mathbf{x}||_\infty < \alpha} \mathcal{L}(\mathbf{x}', y; \boldsymbol{\theta})$$



Computational Costs

- Costs Associated with Generating Adversarial Samples

$$\boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_N^{adv} + \alpha\,\mathrm{sign}\big(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{true})\big)\right\}$$

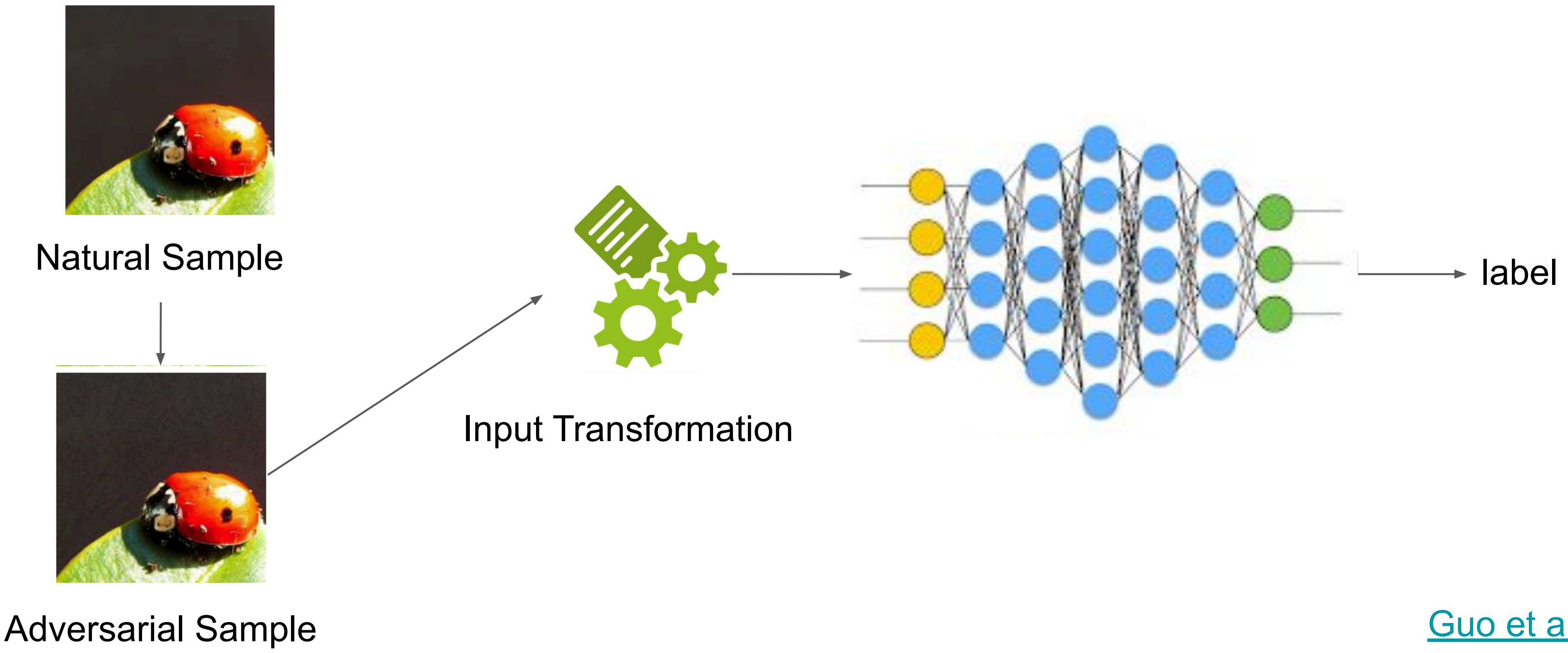


$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1-\alpha) J(\theta, x^{adv}, y)$$

Outline

- Adversarial Defense
- Defense Strategies
  - Adversarial Training
  - Input Transformations
  - Stochastic Gradients
- Obfuscated Gradients and BPDA
- Robust Optimization
- Certified Defense

Input Transformations

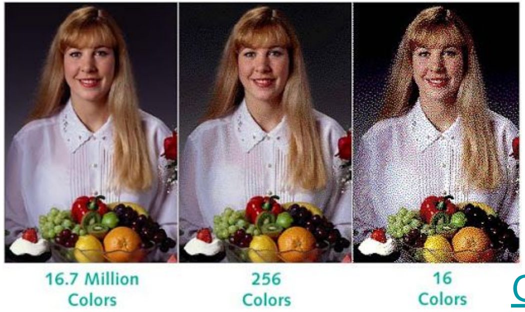


Guo et al, 2018

Input Transformations

- Goal: Disrupt Adversarial Perturbations

- Image cropping/re-scaling
- Bit-depth reduction

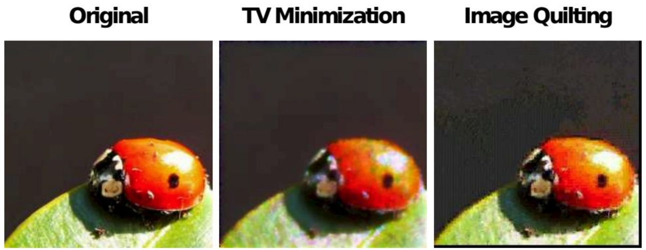


Guo et al, 2018

Input Transformations

- Goal: Disrupt Adversarial Perturbations

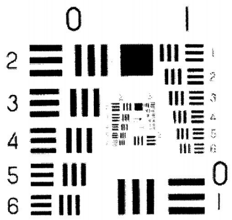
- Image cropping/re-scaling
- Bit-depth reduction
- JPEG compression
- Total variation minimization
- Image quilting



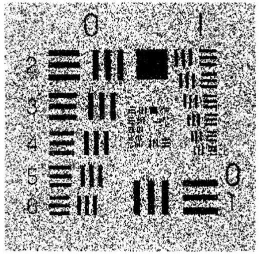
[Guo et al, 2018]

Total Variation Minimization

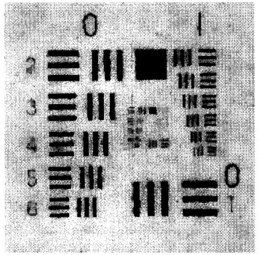
- Generate a denoised image z by minimizing TV



Original Image



Noisy Image



Denoised Image minimizing TV

$$\mathrm{TV}_p(\mathbf{z}) = \sum_{k=1}^{K} \left[\sum_{i=2}^{N} \underbrace{\|\mathbf{z}(i,:,k) - \mathbf{z}(i-1,:,k)\|_p}_{\text{row variance}} + \sum_{j=2}^{N} \underbrace{\|\mathbf{z}(:,j,k) - \mathbf{z}(:,j-1,k)\|_p}_{\text{column variance}} \right]$$

Transformed Image

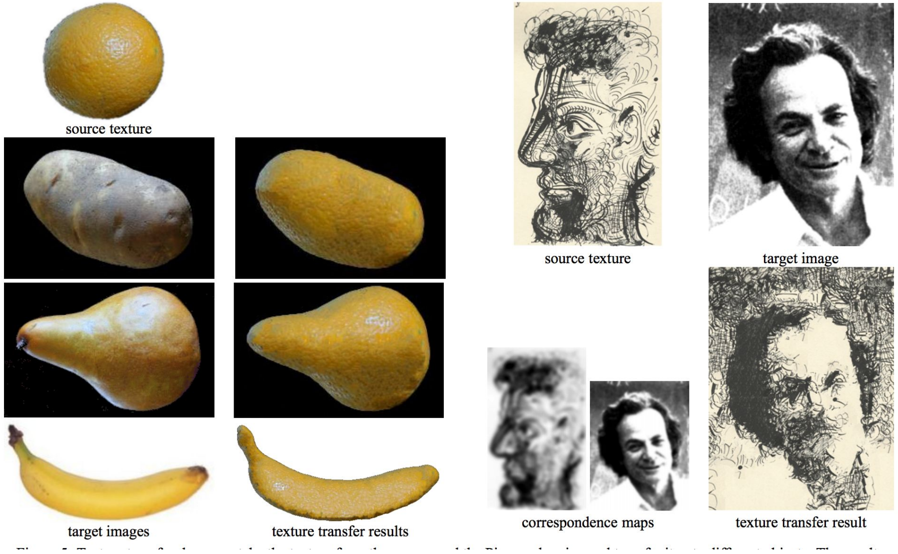
row variance

column variance

Rudin et al, 1992

Image Quilting

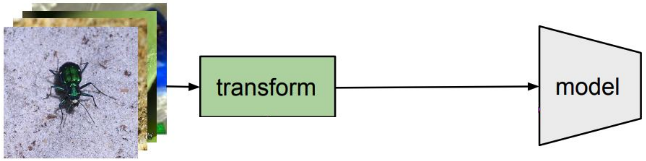
- Synthesizes images by piecing together small patches taken from a database of image patches
- Database contains only clean images



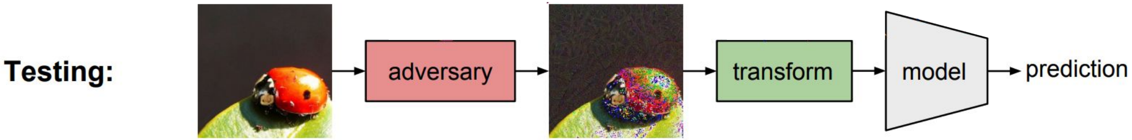
Efros et al, 2001

Input Transformation Defense

Setting a, clean training



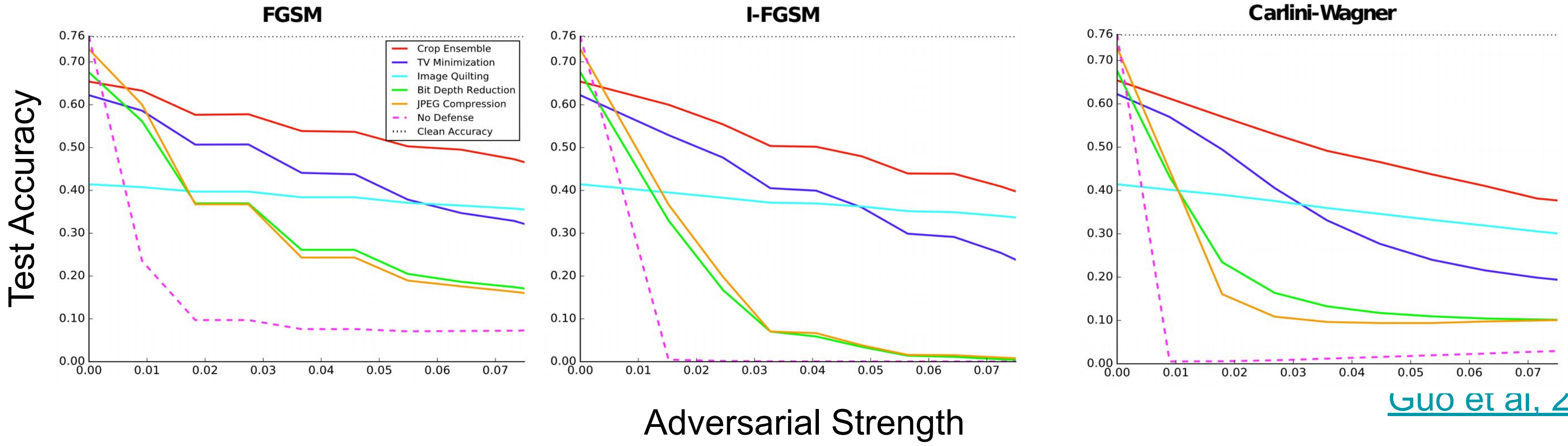
Setting b, training with transformations



[Guo et al, 2018]

Results with Clean Image Training

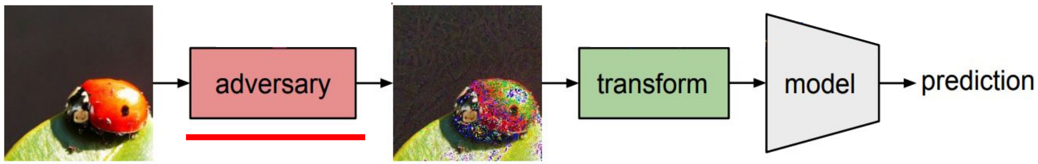
ResNet on ImageNet



Guo et al, 2018

Gradient Shattering

- Can we design specialized attacks that target input transformations?
  - We show previously the results using FGSM and C&W
- Input Transformations belongs to a family of defense methods that causes Gradient Shattering

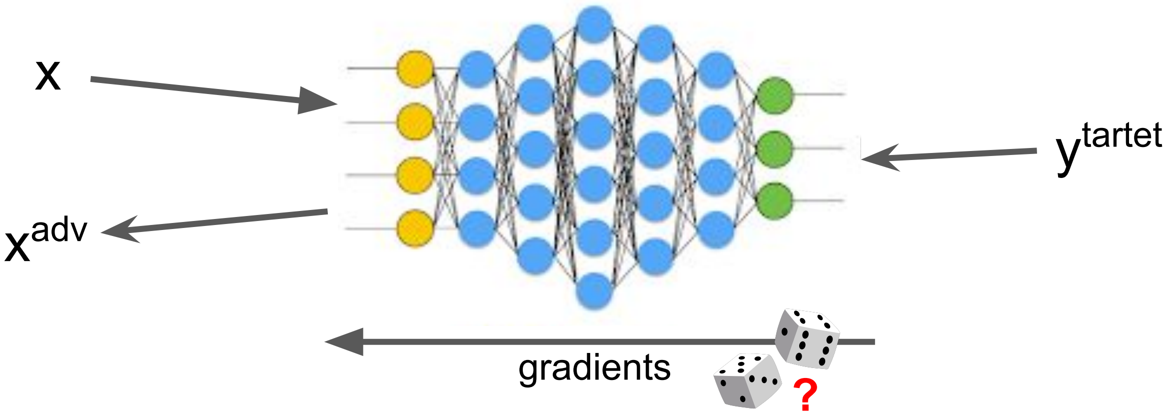


Train our own adversary that targets input transformations?

Outline

- Adversarial Defense
- Defense Strategies
  - Adversarial Training
  - Input Transformations
  - Stochastic Gradients
- Obfuscated Gradients and BPDA
- Robust Optimization
- Certified Defense

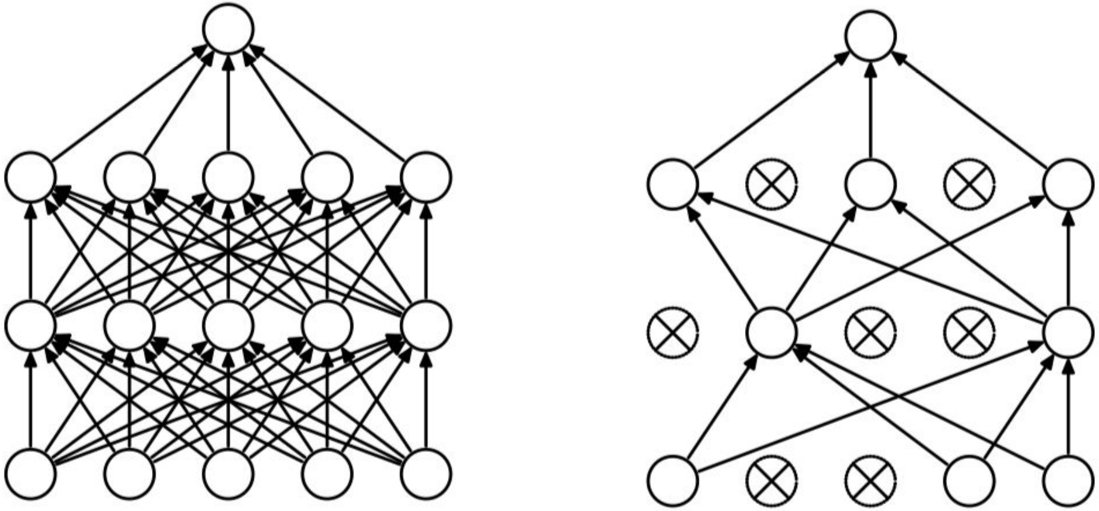
Stochastic Gradients



$$\boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_{N}^{adv} + \alpha\, \mathrm{sign}\big(\underline{\nabla_X J(\boldsymbol{X}_{N}^{adv}, y_{true})}\big)\right\}$$

Dropout

- Dropout randomly turns off activations by a fixed probability r
- Originally introduced to prevent overfitting



(a) Standard Neural Net (b) After applying dropout.

Stochastic Activation Pruning (SAP)

- Stochastic Activation Pruning turns off activations based on a learned probability
- Draw with replacement for each activation

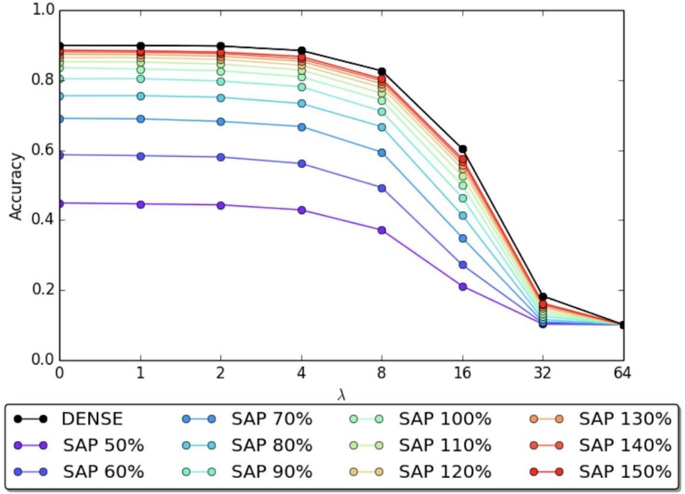
$$\underline{p_j^i} = \frac{|(h^i)_j|}{\sum_{k=1}^{a^i} \underline{|(h^i)_k|}}$$

probability of turning on
the j^{th} activation on the i^{th} layer

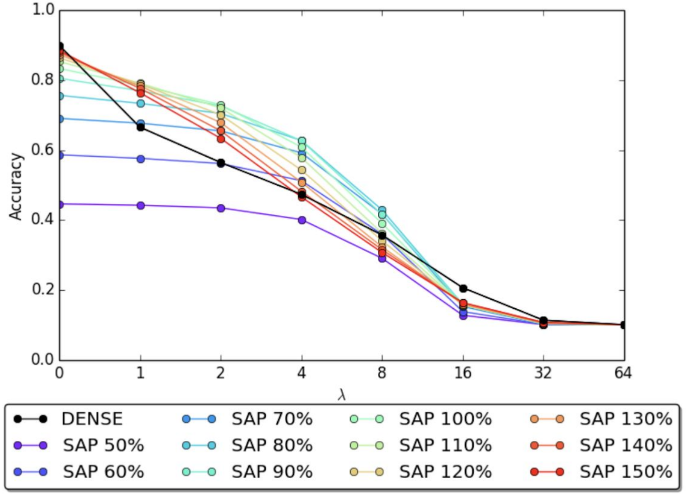
embeddings of
the j^{th} activation on the k^{th} layer

Dhillon et al, 2018

Defense Results



Random Attack



FGSM Attack

SAP % - the percentages of samples drawn for each layer
λ - perturbation strength

Dhillon et al, 2018

Summary of Defense Strategies

Defense Methods	General Idea
Adversarial Training	Mixing adversarial samples with natural samples during training
Input Transformation	Adding transformation to make defense non-differentiable
Stochastic Gradients	Causing gradients to be randomized

Outline

- Adversarial Defense
- Defense Strategies
  - Adversarial Training
  - Input Transformations
  - Stochastic Gradients
- Obfuscated Gradients and BPDA
- Robust Optimization
- Certified Defense

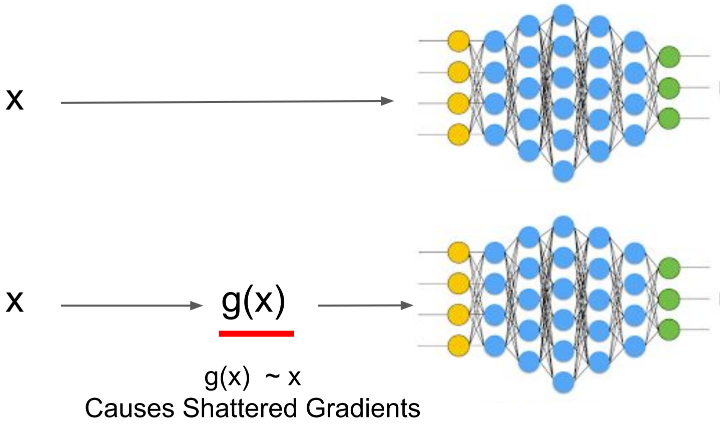
Obfuscated Gradients

- A defense method is said to achieve Obfuscated Gradients if
  - It prevents the attack methods from utilizing useful gradient information
- Shattered Gradients
  - Present a defense method that is non-differentiable or numerically unstable
  - e.g., Input Transformations
- Stochastic Gradients
  - Present a defense method that is randomized, causing single samples to incorrectly estimate the true gradients.
  - e.g., Stochastic Activation Pruning

Athalye et al, 2018

Backward Pass Differentiable Approximation (BPDA)

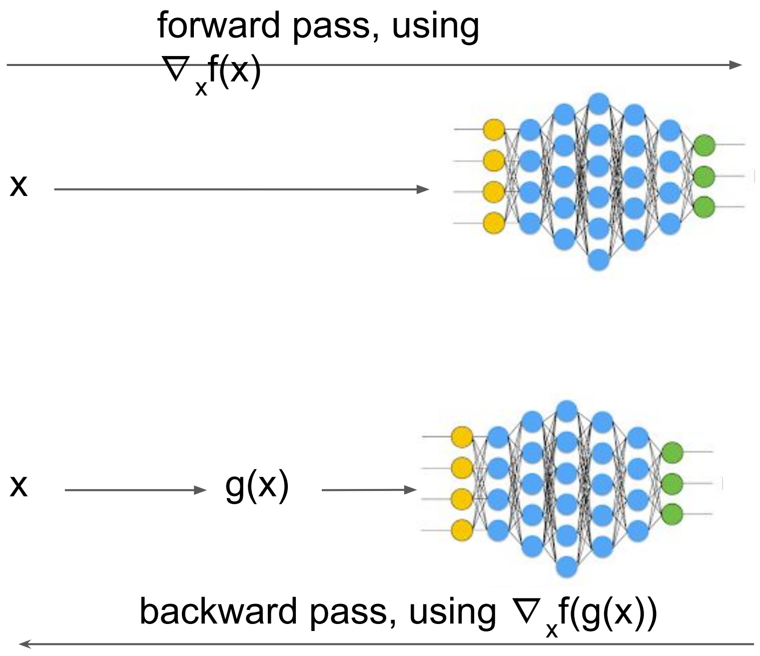
- Bypass Shattered Gradients by its differnetable approximations.



$$\nabla_x f(g(x))|_{x=\hat{x}} \approx \nabla_x f(x)|_{x=g(\hat{x})}$$

[Athalye et al, 2018](#)

BPDA In Neural Networks



[Athalye et al, 2018]

Handling Stochastic Gradients

- Applying the expectations of multiple Stochastic Gradients

$$\nabla \mathbb{E}_{t \sim T} f(t(x)) = \mathbb{E}_{t \sim T} \nabla f(t(x))$$

Results

Defense	Dataset	Distance	Accuracy on Adversarial Samples
Adversarial Training (Madry et al, 2018)	CIFAR	$0.031(l_\infty)$	47%
Input Transformations (Guo et al, 2018)	ImageNet	$0.005(l_2)$	0%
Stochastic Gradients (Dhillon et al, 2018)	CIFAR	$0.031\ (l_\infty)$	0%

Athalye et al, 2018

But Why is Adversarial Training More Robust?

Outline

- Adversarial Defense
- Defense Strategies
  - Adversarial Training
  - Input Transformations
  - Stochastic Gradients
- Obfuscated Gradients and BPDA
- Robust Optimization
- Certified Defense

Robust Optimization

- Train a robust model
  - In the neighborhood of x
  - Under the worst case scenario in terms of the loss function

$$\min_{\theta} \tilde{J}(\theta, x, y) = \min_{\theta} \sum_{i=1}^{m} \max_{\tilde{x}_i \in \mathcal{U}_i} J(\theta, \tilde{x}_i, y_i)$$

uncertainty sets

loss function

[Shaham et al, 2016](#)

Linear Regression As A Robust Optimization

- We can write Linear Regression in the form of Robust Optimization

$$\min_x \|Ax - b\| + \lambda \|x\|_1$$

$$\Downarrow$$

$$\min_x \max_{\|\Delta A\|_{\infty,2} \leq \rho} \|(A + \Delta A)x - b\|$$

Robust Optimization

Shaham et al, 2016

Adversarial Training As A Robust Optimization

- We can also write Adversarial Training in the form of Robust Optimization

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x^{adv}, y)$$

$$\min_{\theta} \tilde{J}(\theta, x, y) = \min_{\theta} \sum_{i=1}^{m} \max_{\tilde{x}_i \in \mathcal{U}_i} J(\theta, \tilde{x}_i, y_i)$$

$$\Delta_{x_i} = \arg \max_{\Delta : x_i + \Delta \in \mathcal{U}_i} J_{\theta, y_i}(x_i + \Delta)$$

Shaham et al, 2016

Outline

- Adversarial Defense
- Defense Strategies
  - Adversarial Training
  - Input Transformations
  - Stochastic Gradients
- Obfuscated Gradients and BPDA
- Robust Optimization
- Certified Defense

Certified Defense

- Guarantee the performance against Adversarial Attack
- Guaranteed for a family of networks

$$f^i(x) = V_i^\top \sigma(Wx)$$

Two-layer Neural Network

Raghunathan et al, 2018

Bounded Performance

Error Margin $$f(x) = \underbrace{f^1(x)}_{\text{incorrect class}} - \underbrace{f^2(x)}_{\text{correct class}}$$

$$\underbrace{f(A(x))}_{\text{Error of any attack}} \leq \underbrace{f(A_{\text{opt}}(x))}_{\text{Error of optimal attack}} \leq \underbrace{f(x) + \epsilon \max_{\tilde{x} \in B_\epsilon(x)} \|\nabla f(\tilde{x})\|_1}_{\text{Bounds}} \leq f_{\text{QP}}(x) \leq \underbrace{f_{\text{SDP}}(x)}_{\text{Computationally Feasible Bounds}}$$

Bounded Performance

Error Margin $f(x) = \underbrace{f^1(x)}_{\text{incorrect class}} - \underbrace{f^2(x)}_{\text{correct class}}$

$$f(A(x)) \leq f(A_{\text{opt}}(x)) \leq f(x) + \epsilon \max_{\tilde{x} \in B_\epsilon(x)} \|\nabla f(\tilde{x})\|_1 \leq f_{\text{QP}}(x) \leq f_{\text{SDP}}(x)$$

Upper Bound
(SDP)

$$\underbrace{f_{\text{SDP}}(x)}_{\text{solution to semidefinite program}} \stackrel{\text{def}}{=} f(x) + \frac{\epsilon}{4} \max_{P \succeq 0, \text{diag}(P) \leq 1} \langle M(v, W), P \rangle$$

$$M(v, W) \stackrel{\text{def}}{=} \begin{bmatrix} 0 & 0 & \mathbf{1}^\top W^\top \text{diag}(v) \\ 0 & 0 & W^\top \text{diag}(v) \\ \text{diag}(v)^\top W \mathbf{1} & \text{diag}(v)^\top W & 0 \end{bmatrix} \quad v \stackrel{\text{def}}{=} V_1 - V_2$$

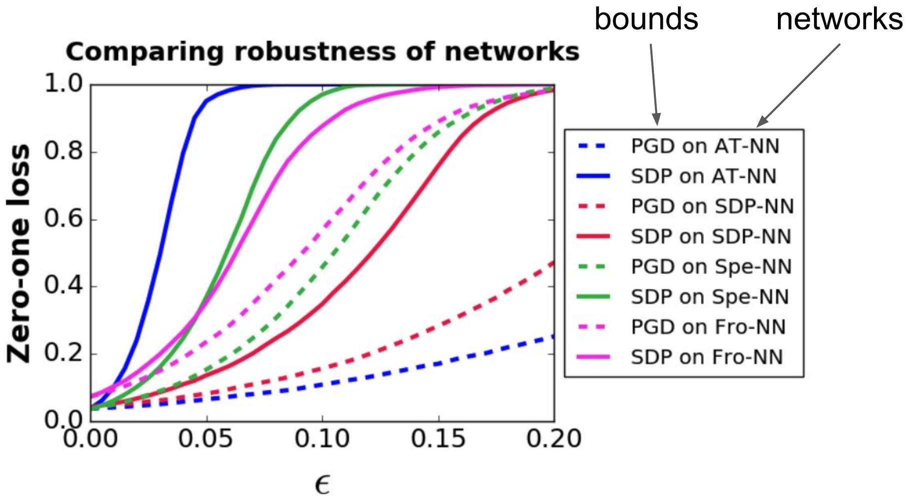
Training Certified Defense

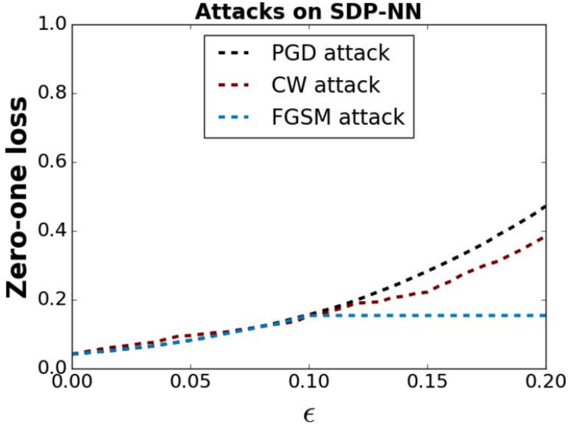
$$f(A(x)) \leq f(A_{\text{opt}}(x)) \;\leq\; f(x) + \epsilon \max_{\tilde{x} \in B_\epsilon(x)} \|\nabla f(\tilde{x})\|_1 \;\leq\; f_{\text{QP}}(x) \;\leq\; \underline{f_{\text{SDP}}(x)}$$

$$f_{\text{SDP}}(x) \stackrel{\text{def}}{=} f(x) + \frac{\epsilon}{4} \max_{P \succeq 0, \text{diag}(P) \leq 1} \langle M(v, W), P \rangle$$

$$(W^\star, V^\star) = \underbrace{\arg\min_{W,V}}_{\text{parameters to the two-layer neural network}} \sum_n \underbrace{\ell_{\text{cls}}(V, W; x_n, y_n)}_{\text{loss function}} + \sum_{i \neq j} \underbrace{\lambda^{ij}}_{\text{hyper-parameter}} \underbrace{\max_{P \succeq 0, \text{diag}(P) \leq 1} \left\langle M^{ij}(V, W), P \right\rangle}_{\text{Defense Certification}}$$

Results





AT-NN - Adversarial training using PGD (Madry et al, 2018)
SDP-NN - Proposed training objective
Spe-NN - Spectral norm regularization i.e., $\lambda(||W||_2 + ||v||_2)$
Fro-NN - Frobenius norm regularization i.e., $\lambda(||W||_F + ||v||_2)$ $\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$

PGD - lower bound
SDP - upper bound

$$\underbrace{f(A(x))}_{\text{PGD lower bound}} \leq f(A_{\text{opt}}(x)) \leq f(x) + \epsilon \max_{\tilde{x} \in B_\epsilon(x)} \|\nabla f(\tilde{x})\|_1 \leq f_{\text{QP}}(x) \leq \underbrace{f_{\text{SDP}}(x)}_{\text{SDP lower bound}}$$

Raghunathan et al, 2018

Results

- No attack that perturbs each pixel by at most = 0.1 can cause more than 35% test error.

Network	PGD error	SDP bound
SDP-NN	15%	35%

SDP-NN - Proposed training objective
PGD - upper bound
SDP - lower bound
$\varepsilon = 0.1$

Summary

- Robustness of ML Models
  - Preventing models from being abused by malicious attack
- Adversarial Attack
  - Confuses models by manipulating input data
  - Evasion attack
  - Poisoning attack
  - Exploratory attack
- Attack Strategies
  - FGSM - white-box
  - C&W -white-box
  - Jacobian-based Data Augmentation - black-box

Summary

- Adversarial Defense
    - Equip models with the ability to defend adversarial attacks
- Defense Strategies
    - Adversarial Training
        - Robust Optimization
    - Gradient Shattering
    - Stochastic Gradients
- BPDA
    - Attack all defense models utilizing Obfuscated Gradients
- Certified Defense
    - Provable performance for certain types of networks

Reading Assignments

- Metzen, Jan Hendrik, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations, ICLR 2017
- Raghunathan, Aditi, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples, ICLR 2018
- Cohen, Jeremy M., Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, ICML 2019
- Samangouei, Pouya, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models, ICLR 2018
- Tramèr, Florian, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses, ICLR 2018