Towards Evaluating the Robustness of Neural Networks

Nicholas Carlini, David Wagner (IEEE S&P 2017)

Presenter: Zhixian Jin

Who is David Wagner?

David Wagner
Berkely

- **Research Interest**
  - wireless security
  - sensor network security.
  - applied cryptography.
  - software security.

- **Publication**
  - Towards Evaluating the Robustness of Neural Networks
    - N Carlini, D Wagner (IEEE S&P 2017)
  - Secure routing in wireless sensor networks: Attacks and countermeasures
    - C Karlof, D Wagner (Ad hoc network, 2003)
  - Practical techniques for searches on encrypted data (S&P, 2000)
    - DX Song, D Wagner, A Perrig
  - Android permissions demystified (CCS, 2011)
    - AP Felt, E Chin, S Hanna, D Song, D Wagner

Who is Nicholas Carlini?

Nicholas Carlini
Google Deep Mind

- **Research Interest**
  - Machine learning and computer security
  - Neural networks from an adversarial perspective

- **Publication**
  - Hidden Voice Commands
    - N Carlini, P Mishra, T Vaidya, Y Zhang, M Sherr, C Shields, D Wagner (USENIX Security, 2016)
  - Towards Evaluating the Robustness of Neural Networks
    - N Carlini, D Wagner (IEEE S&P 2017)
  - Adversarial examples are not easily detected: Bypassing ten detection methods
    - N Carlini, D Wagner (AISec, 2017)
  - Audio adversarial examples: Targeted attacks on speech-to-text
    - N Carlini, D Wagner (SPW 2018)
  - The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks
    - N Carlini, C Liu, J Kos, Ú Erlingsson, D Song(USENIX Security, 2017)

- **Personal Website:**
  - https://nicholas.carlini.com
  - I really enjoy reading his blog :)

Fun Fact

Top-100 Security Papers

This webpage is an attempt to assemble a ranking of top-cited papers from the area of computer security. The ranking has been created based on citations of papers published at top security conferences. More details are available here.

Top 100 papers from 1980 to 2023

1. Nicholas Carlini and David A. Wagner:
Towards Evaluating the Robustness of Neural Networks.
IEEE Symposium on Security and Privacy, 2017
7972 cites at Google Scholar | 6380% above average of year | Last visited: Oct-2023 | Paper: DOI

2. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters:
Attribute-based encryption for fine-grained access control of encrypted data.
ACM Conference on Computer and Communications Security (CCS), 2006
6771 cites at Google Scholar | 3269% above average of year | Last visited: Oct-2023 | Paper: DOI

3. John Bethencourt, Amit Sahai, and Brent Waters:
Ciphertext-Policy Attribute-Based Encryption.
IEEE Symposium on Security and Privacy, 2007
6509 cites at Google Scholar | 2947% above average of year | Last visited: Oct-2023 | Paper: DOI

WRITING — MAIN PAPERS TALKS CODE WRITING

Reflecting on "Towards Evaluating the Robustness of Neural Networks"
by **Nicholas Carlini** 2022-08-17

"It's somewhat strange knowing that my most cited work is behind me, and nothing I ever do match this paper no matter how hard I try"

Nicolas Carlini

What is adversarial?

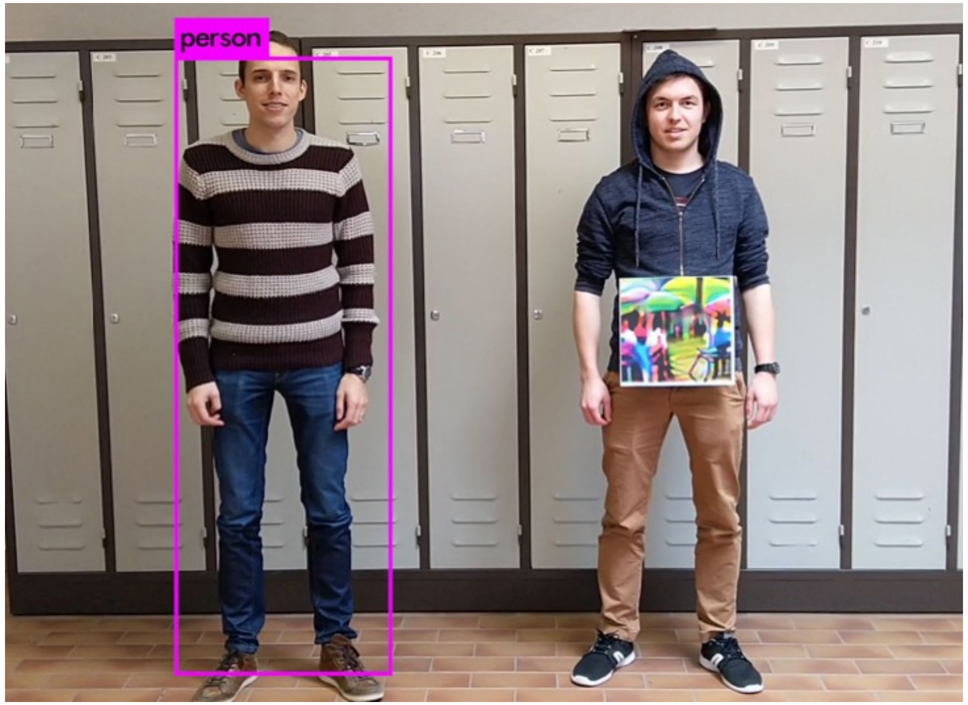
Adversarial

Perturbation

88 % Tabby Cat

99 % Guacamole

Adversarial Example



Person with patch does not consider as a person

Adversarial Example

Examples that already covered in the class

Introduction

Adversary has access to the model parameters

Goal: construct a adversarial example

Goal

Create an adversarial image

perturbation

True label = *Tabby Cat*

Target label T = *Guacamole*

(1) Make classifier recognize the fake cat image as Guacamole
(2) Minimize difference of true cat image and fake cat image

Goal

Create an adversarial image x' by adding small perturbation δ to the original image x (i.e., $x' = x + \delta$),

The classifier should assign the class label T to the adversarial image x', where T is different with the true label

x

True label = Tabby Cat

δ = perturbation

x'

Target label T = Guacamole

Find δ that

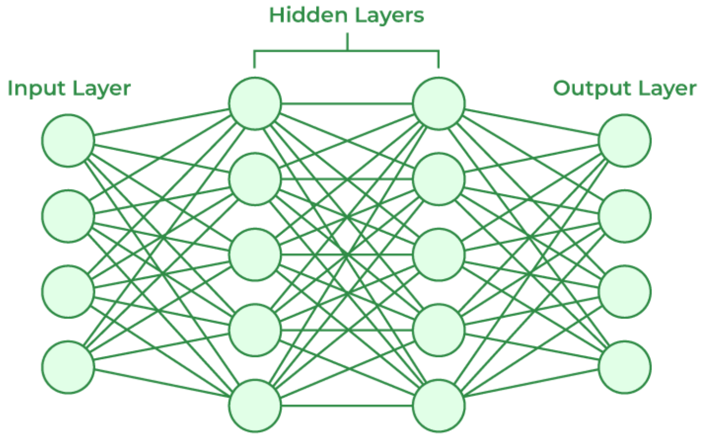
Minimize $\mathcal{D}(x, x')$

such that $C\,(x') = \mathrm{T}$

Challenge

Find δ that minimize $\mathcal{D}(x, x')$
such that $C\,(x') = \mathrm{T}$

But… $C(x')$



Highly non-linear

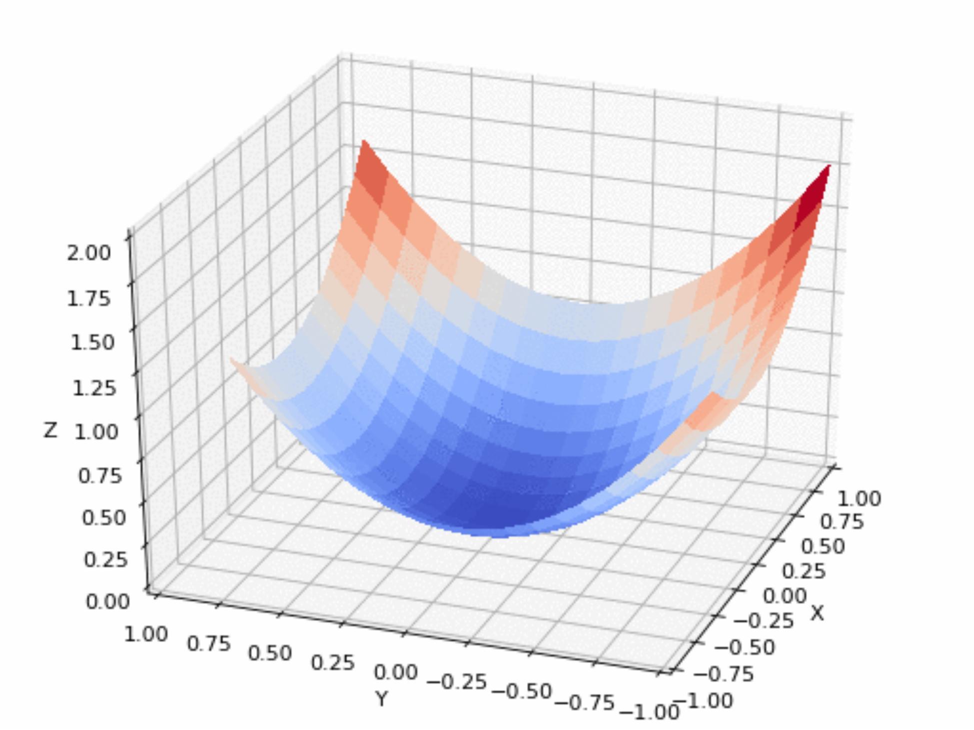
Key Insight : Gradient descent works very well for training neural networks.
Why not for breaking them too?

Challenge

- **Gradient Descent**

Loss Function

Measure how accurate the neural network is?



Problem Reformulation

Intial Problem

Find δ that minimize $\mathcal{D}(x, x')$

such that $C\,(x') = \mathrm{T}$

Change Problem

Find δ minimize $\mathcal{D}(x,x') + f(x')$

Where f(x′) is some kind of loss function on how close C(x′) is to target T

For loss function: $f(x') \leq 0$ if $C(x') = T$, $f(x') > 0$ if $C(x') \mathrel{!}= T$

New Problem:

Find δ such that $\mathcal{D}(x,x') + f(x') \leq 0$

Loss Function

New Optimization Problem: $\mathcal{D}(x,x') + f(x') \leq 0$

Question: How to choose f(x′) ?

Candidate loss fucntion:

$$f_1(x') = -\text{loss}_{F,t}(x') + 1$$
$$f_2(x') = (\max_{i \neq t}(F(x')_i) - F(x')_t)^+$$
$$f_3(x') = \text{softplus}(\max_{i \neq t}(F(x')_i) - F(x')_t) - \log(2)$$
$$f_4(x') = (0.5 - F(x')_t)^+$$
$$f_5(x') = -\log(2F(x')_t - 2)$$
$$f_6(x') = (\max_{i \neq t}(Z(x')_i) - Z(x')_t)^+$$
$$f_7(x') = \text{softplus}(\max_{i \neq t}(Z(x')_i) - Z(x')_t) - \log(2)$$

Loss function

- Emperically select the best one

	Best Case						Average Case						Worst Case					
	Change of Variable		Clipped Descent		Projected Descent		Change of Variable		Clipped Descent		Projected Descent		Change of Variable		Clipped Descent		Projected Descent	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
f_1	2.46	100%	2.93	100%	2.31	100%	4.35	100%	5.21	100%	4.11	100%	7.76	100%	9.48	100%	7.37	100%
f_2	4.55	80%	3.97	83%	3.49	83%	3.22	44%	8.99	63%	15.06	74%	2.93	18%	10.22	40%	18.90	53%
f_3	4.54	77%	4.07	81%	3.76	82%	3.47	44%	9.55	63%	15.84	74%	3.09	17%	11.91	41%	24.01	59%
f_4	5.01	86%	6.52	100%	7.53	100%	4.03	55%	7.49	71%	7.60	71%	3.55	24%	4.25	35%	4.10	35%
f_5	1.97	100%	2.20	100%	1.94	100%	3.58	100%	4.20	100%	3.47	100%	6.42	100%	7.86	100%	6.12	100%
f_6	1.94	100%	2.18	100%	1.95	100%	3.47	100%	4.11	100%	3.41	100%	6.03	100%	7.50	100%	5.89	100%
f_7	1.96	100%	2.21	100%	1.94	100%	3.53	100%	4.14	100%	3.43	100%	6.20	100%	7.57	100%	5.94	100%

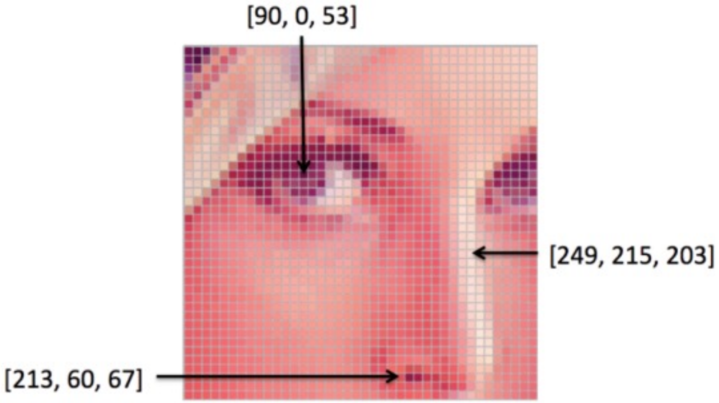
TABLE III

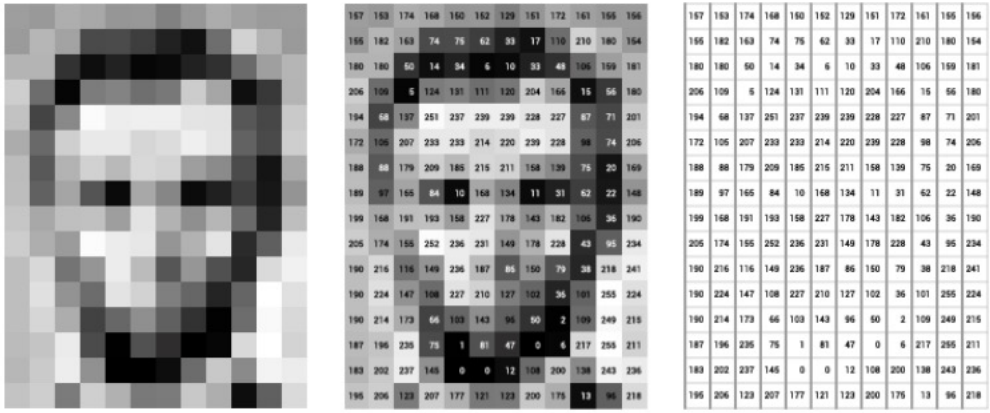
EVALUATION OF ALL COMBINATIONS OF ONE OF THE SEVEN POSSIBLE OBJECTIVE FUNCTIONS WITH ONE OF THE THREE BOX CONSTRAINT ENCODINGS. WE SHOW THE AVERAGE L_2 DISTORTION, THE STANDARD DEVIATION, AND THE SUCCESS PROBABILITY (FRACTION OF INSTANCES FOR WHICH AN ADVERSARIAL EXAMPLE CAN BE FOUND). EVALUATED ON 1000 RANDOM INSTANCES. WHEN THE SUCCESS IS NOT 100%, MEAN IS FOR SUCCESSFUL ATTACKS ONLY.

Box Constraint

Optimization Problem: $\mathcal{D}(x,x') + f(x') \leq 0$

Challenge: Find δ , $x' = x + \delta$ $(0 \leq x + \delta \leq 1)$





$x + \delta$ need to be in range [0, 255] or [0,1]

Box Constraint

Optimization Problem: $\mathcal{D}(x,x') + f(x') \le 0$

Challenge: Find δ , $x' = x + \delta$ $(0 \le x + \delta \le 1)$

Solution: Change the variable!

$$x + \delta \;\; \longrightarrow \;\; \frac{1}{2}(\tanh(w) + 1) \;\; \longrightarrow \;\; \delta = \frac{1}{2}(\tanh(w) + 1) - x$$

$$-1 \le \tanh(w) \le 1$$

$$0 \le \tanh(w) + 1 \le 2$$

$$0 \le \frac{1}{2}(\tanh(w) + 1) \le 1$$

$$0 \le x + \delta \le 1$$

Box Constraint

	Best Case						Average Case						Worst Case					
	Change of Variable		Clipped Descent		Projected Descent		Change of Variable		Clipped Descent		Projected Descent		Change of Variable		Clipped Descent		Projected Descent	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
f_1	2.46	100%	2.93	100%	2.31	100%	4.35	100%	5.21	100%	4.11	100%	7.76	100%	9.48	100%	7.37	100%
f_2	4.55	80%	3.97	83%	3.49	83%	3.22	44%	8.99	63%	15.06	74%	2.93	18%	10.22	40%	18.90	53%
f_3	4.54	77%	4.07	81%	3.76	82%	3.47	44%	9.55	63%	15.84	74%	3.09	17%	11.91	41%	24.01	59%
f_4	5.01	86%	6.52	100%	7.53	100%	4.03	55%	7.49	71%	7.60	71%	3.55	24%	4.25	35%	4.10	35%
f_5	1.97	100%	2.20	100%	1.94	100%	3.58	100%	4.20	100%	3.47	100%	6.42	100%	7.86	100%	6.12	100%
f_6	1.94	100%	2.18	100%	1.95	100%	3.47	100%	4.11	100%	3.41	100%	6.03	100%	7.50	100%	5.89	100%
f_7	1.96	100%	2.21	100%	1.94	100%	3.53	100%	4.14	100%	3.43	100%	6.20	100%	7.57	100%	5.94	100%

TABLE III

EVALUATION OF ALL COMBINATIONS OF ONE OF THE SEVEN POSSIBLE OBJECTIVE FUNCTIONS WITH ONE OF THE THREE BOX CONSTRAINT ENCODINGS. WE SHOW THE AVERAGE L_2 DISTORTION, THE STANDARD DEVIATION, AND THE SUCCESS PROBABILITY (FRACTION OF INSTANCES FOR WHICH AN ADVERSARIAL EXAMPLE CAN BE FOUND). EVALUATED ON 1000 RANDOM INSTANCES. WHEN THE SUCCESS IS NOT 100%, MEAN IS FOR SUCCESSFUL ATTACKS ONLY.

L2 Attack

Initial problem: Find δ, Minimize $\mathcal{D}$(x, x + δ), such that C (x + δ) = T

⬇ Use Gradient descent, Define Loss function

Problem Reformulation: Find δ, Minimize $\mathcal{D}$(x, x + δ) + f(x + δ)

Box Constraint:

$$\delta = \frac{1}{2}\,(\tanh(w) + 1) - x$$

⬇ Use L2 for $\mathcal{D}$(x, x + δ)

Fianl Optimization Problem: Find w, minimize $\left\| \frac{1}{2}\,(\tanh(w) + 1) - x \right\|_2^2 + \mathrm{f}(\frac{1}{2}\,(\tanh(w) + 1))$

L0 and L∞ Attack

L_0 attack

- L_0 is not differentiable
- Use iterative approach:
    - Goal: find pixels that are not important and don't have much effect on the classifier's output
    - Perform L_2 attack to find an adversarial example $x + \delta$
    - Identify the least important pixel and remove this pixel from the allowed set
    - Iterate until the L_2 attack fails to find an adversarial example

L_∞ attack

- L_∞ is not always differentiable
- gradient descent becomes stuck oscillating between two suboptimal point
- penalize δ that have large values by introducing τ
- iteratively decrease the size of τ

$$\text{minimize} \quad c \cdot f(x+\delta) + \cdot \sum_i \left[(\delta_i - \tau)^+\right]$$

Evaluation

Evaluation #1: Comparing to Other Attacks
Evaluation #2: Breaking Current Defenses

Dataset: MNIST, CIFAR-10, and ImageNet

Evaluation

- **Evaluation #1: Comparing to Other Attacks**

	Best Case				Average Case				Worst Case			
	MNIST		CIFAR		MNIST		CIFAR		MNIST		CIFAR	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
Our L_0	8.5	100%	5.9	100%	16	100%	13	100%	33	100%	24	100%
JSMA-Z	20	100%	20	100%	56	100%	58	100%	180	98%	150	100%
JSMA-F	17	100%	25	100%	45	100%	110	100%	100	100%	240	100%
Our L_2	1.36	100%	0.17	100%	1.76	100%	0.33	100%	2.60	100%	0.51	100%
Deepfool	2.11	100%	0.85	100%	—	-	—	-	—	-	—	-
Our L_∞	0.13	100%	0.0092	100%	0.16	100%	0.013	100%	0.23	100%	0.019	100%
Fast Gradient Sign	0.22	100%	0.015	99%	0.26	42%	0.029	51%	—	0%	0.34	1%
Iterative Gradient Sign	0.14	100%	0.0078	100%	0.19	100%	0.014	100%	0.26	100%	0.023	100%

	Best Case				Average Case				Worst Case			
	MNIST		CIFAR		MNIST		CIFAR		MNIST		CIFAR	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
Our L_0	10	100%	7.4	100%	19	100%	15	100%	36	100%	29	100%
Our L_2	1.7	100%	0.36	100%	2.2	100%	0.60	100%	2.9	100%	0.92	100%
Our L_∞	0.14	100%	0.002	100%	0.18	100%	0.023	100%	0.25	100%	0.038	100%

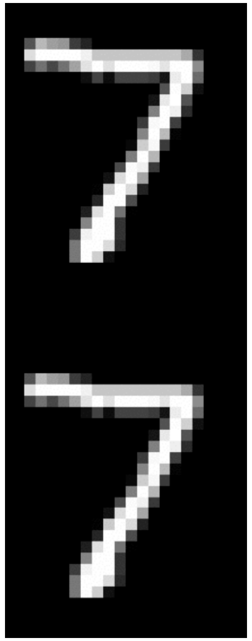
TABLE VI
COMPARISON OF OUR ATTACKS WHEN APPLIED TO DEFENSIVELY DISTILLED NETWORKS. COMPARE TO TABLE IV FOR UNDISTILLED NETWORKS.

	Untargeted		Average Case		Least Likely	
	mean	prob	mean	prob	mean	prob
Our L_0	48	100%	410	100%	5200	100%
JSMA-Z	-	0%	-	0%	-	0%
JSMA-F	-	0%	-	0%	-	0%
Our L_2	0.32	100%	0.96	100%	2.22	100%
Deepfool	0.91	100%	-	-	-	-
Our L_∞	0.004	100%	0.006	100%	0.01	100%
FGS	0.004	100%	0.064	2%	-	0%
IGS	0.004	100%	0.01	99%	0.03	98%

Evaluation

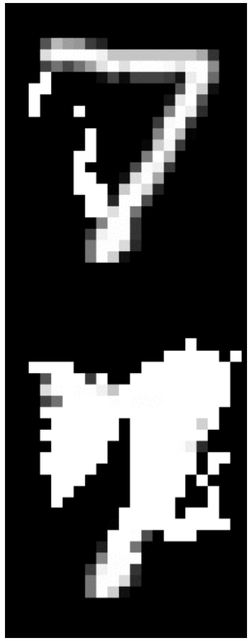
- **Evaluation #1: Comparing to Other Attacks**

Original



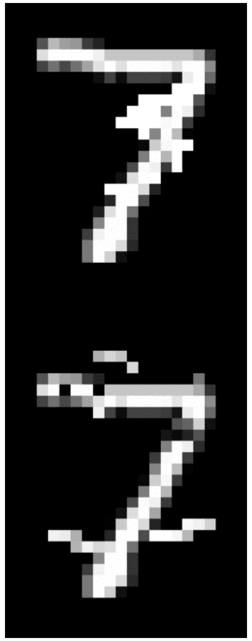
7

Previous Attack



9

CW attack



9

Evaluation

DOG

Hummingbird

Evaluation

DOG

Hummingbird

Evaluation

DOG (83%)

Hummingbird (98%)

Defending Against Adversarial Examples

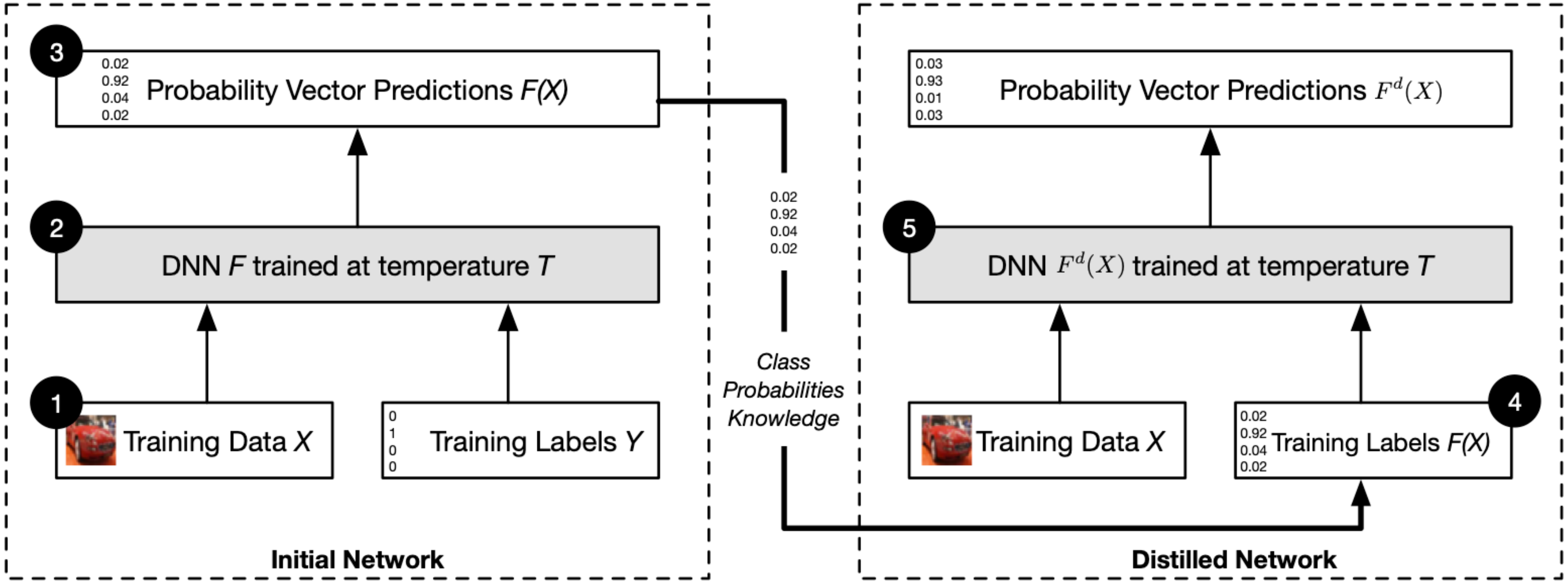
Evaluation #2: Breaking Current Defenses

Huang, R., Xu, B., Schuurmans, D., and Szepesvári, C. Learning with a strong adversary. CoRR, abs/1511.03034 (2015)
Jin, J., Dundar, A., and Culurciello, E. Robust convolutional neural networks under adversarial noise. arXiv preprint arXiv:1511.06306 (2015)
Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. IEEE Symposium on Security and Privacy (2016) Hendrycks, D., and Gimpel, K. Visible progress on adversarial images and a new saliency map. arXiv preprint arXiv:1608.00530 (2016)
Li, X., and Li, F. Adversarial examples detection in deep networks with convolutional filter statistics. arXiv preprint arXiv:1612.07767 (2016)
Wang, Q. et al. Using Non-invertible Data Transformations to Build Adversary-Resistant Deep Neural Networks. arXiv preprint arXiv:1610.01934 (2016).
Ororbia, I. I., et al. Unifying adversarial training algorithms with flexible deep data gradient regularization. arXiv preprint arXiv:1601.07213 (2016).
Wang, Q. et al. Learning Adversary-Resistant Deep Neural Networks. arXiv preprint arXiv:1612.01401 (2016).
Grosse, K., Manoharan, P., Papernot, N., Backes, M., and McDaniel, P. On the (statistical) detection of adversarial examples. arXiv preprint arXiv:1702.06280 (2017)
Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. On detecting adversarial perturbations. arXiv preprint arXiv:1702.04267 (2017)
Feinman, R., Curtin, R. R., Shintre, S., Gardner, A. B. Detecting Adversarial Samples from Artifacts. arXiv preprint arXiv:1703.00410 (2017)
Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. Adversarial and Clean Data Are Not Twins. arXiv preprint arXiv:1704.04960 (2017)
Dan Hendrycks and Kevin Gimpel. Early Methods for Detecting Adversarial Images. In International Conference on Learning Representations (Workshop Track) (2017)
Bhagoji, A. N., Cullina, D., and Mittal, P. Dimensionality Reduction as a Defense against Evasion Attacks on Machine Learning Classifiers. arXiv preprint arXiv:1704:02654 (2017)
Abbasi, M., and Christian G.. Robustness to Adversarial Examples through an Ensemble of Specialists. arXiv preprint arXiv:1702.06856 (2017).
Lu, J., Theerasit I., and David F. SafetyNet: Detecting and Rejecting Adversarial Examples Robustly. arXiv preprint arXiv:1704.00103 (2017)
Xu, W., Evans, D., and Qi, Y. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. arXiv preprint arXiv:1704.01155 (2017)
Hendrycks, D, and Gimpel, K. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. arXiv preprint arXiv:1610.02136 (2016)
Gondara, Lovedeep. Detecting Adversarial Samples Using Density Ratio Estimates. arXiv preprint arXiv:1705.02224 (2017)
Hosseini, Hossein, et al. Blocking transferability of adversarial examples in black-box learning systems. arXiv preprint arXiv:1703.04318 (2017)
Ji Gao, Beilun Wang, Zeming Lin, Weilin Xu, Yanjun Qi. DeepCloak: Masking Deep Neural Network Models for Robustness Against Adversarial Samples. In ICLR (Workshop Track) (2017)
Wang, Q. et al. Adversary Resistant Deep Neural Networks with an Application to Malware Detection. arXiv preprint arXiv:1610.01239 (2017)
Cisse, Moustapha, et al. Parseval Networks: Improving Robustness to Adversarial Examples. arXiv preprint arXiv:1704.08847 (2017).
Nayebi, Aran, and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks. arXiv preprint arXiv:1703.09202 (2017).

Evaluation

- **Defeat the strongest defense!**

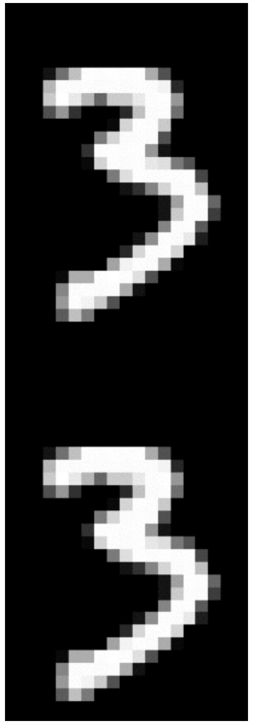
Defensive distillation : Let's not make our NN has high gradient



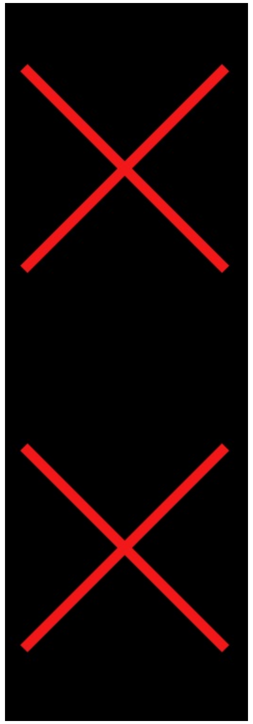
Evaluation

- **Defeat the strongest defense!**

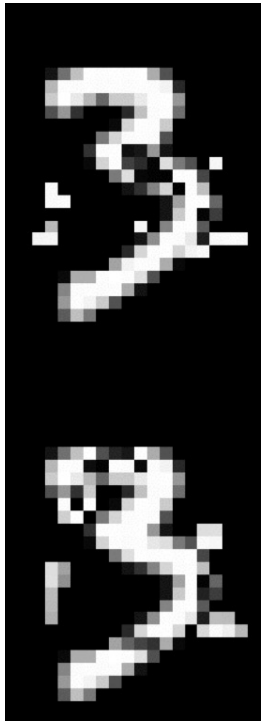
Original



Previous Attack
With Defensive distillation



CW attack
With Defensive distillation



Conclusion

+ Gradient descent work well with the adversarial example
+ Strongest attack ever at 2017

- White box model
- Attack is very slow

Related Work

- **Generative model vs Gradient optimization**
  - PGD: Towards Deep Learning Models Resistant to Adversarial Attacks
    - Gradient Descent work just fine!
  - AdvGan: Generating Adversarial Examples with Adversarial Networks
    - Use Generative Adversarial Network to generate the adversarial example

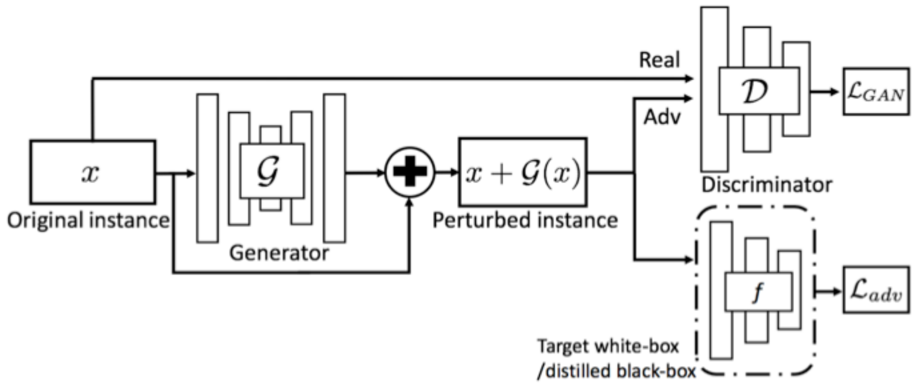


Figure 1: Overview of AdvGAN

Best Question

- **Jio Oh**
  - Can we use Lagrangian dual to make convex function? Why use change of variables?

- **Taeung Yoon**
  - Can we use CW attacks refine the defensive distillation approach?

- **Seunghyun Lee**
  - Can we define lower bound of the model roubutness?