Robustness and Evasion Attacks

May 15, 2020

Dr. Wei Wei, Prof. James Landay

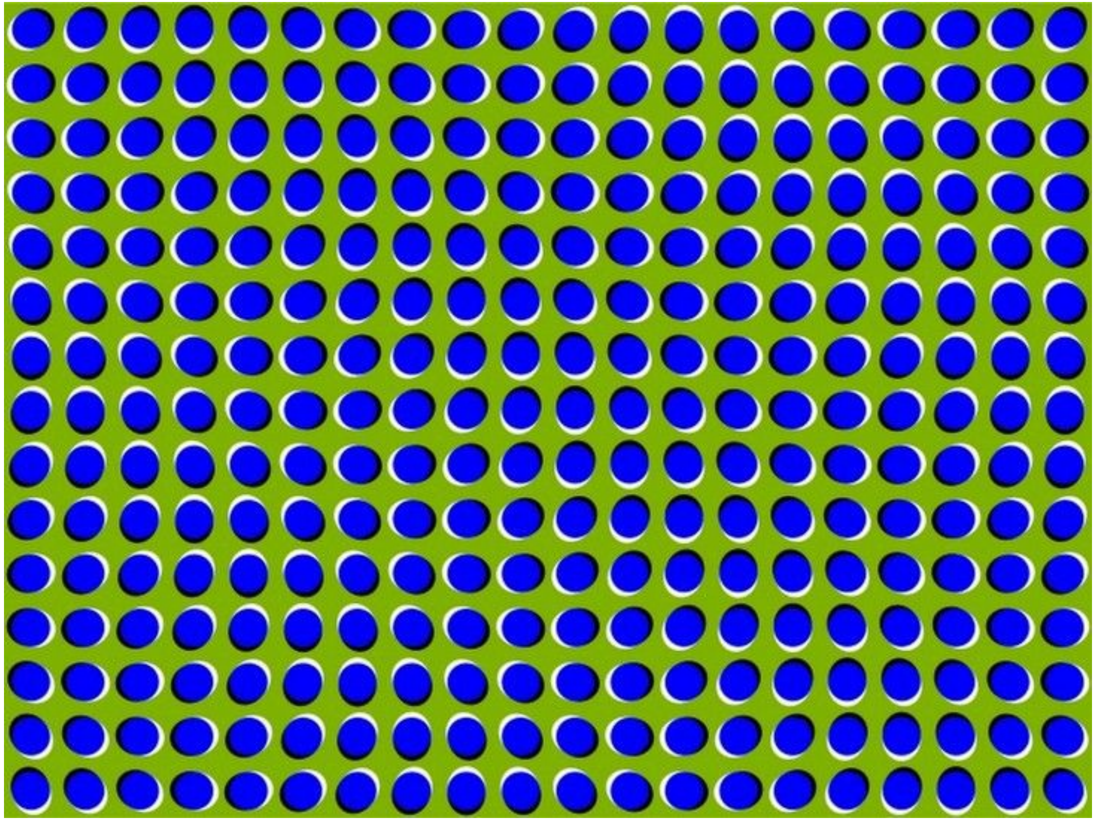
CS 335: Fair, Accountable, and Transparent (FAccT) Deep Learning

Stanford University

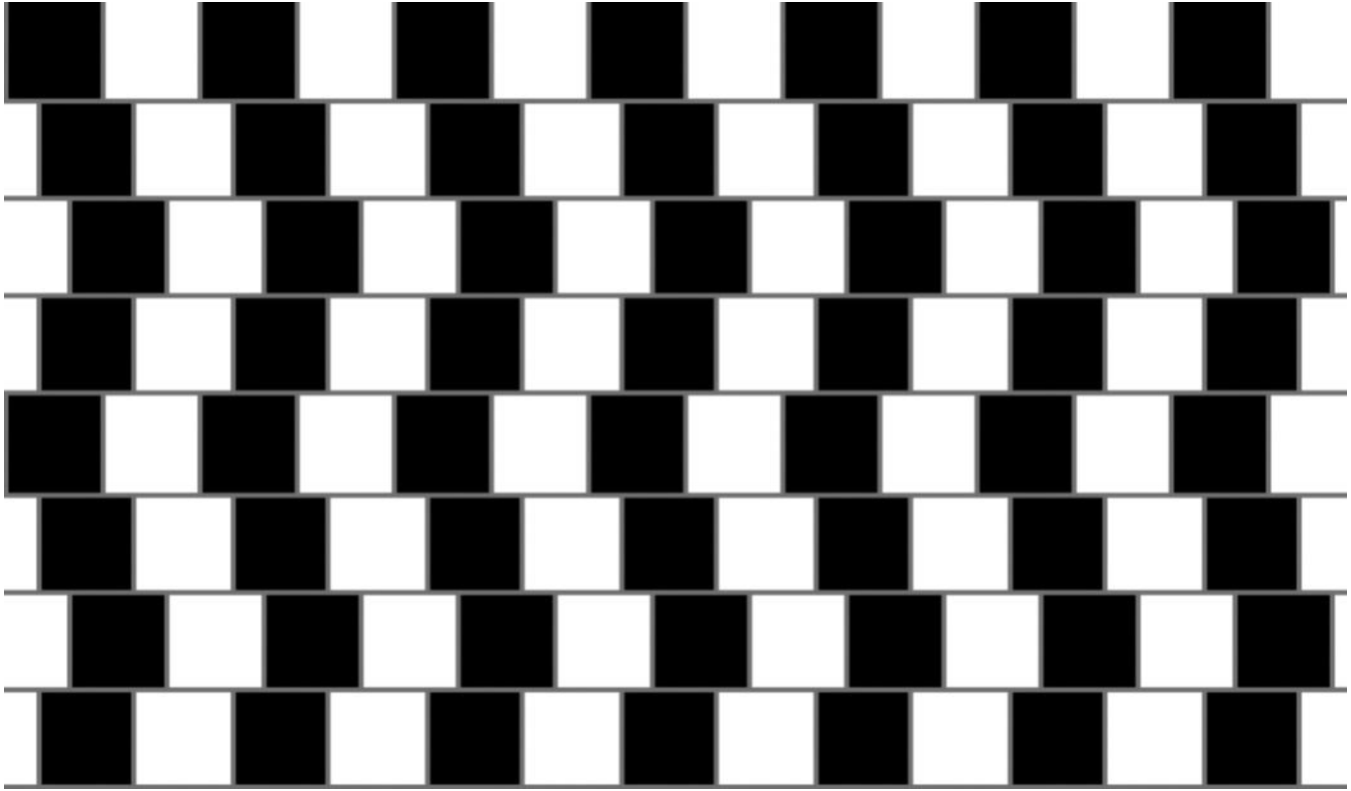
Outline

- Optical Illusions
- Adversarial Attack
- White-box Evasion Attack
  - FGSM
  - C&W
  - Physical Attack
- Transferability of Attack
- Black-box Evasion Attack
  - Jacobian-based Data Augmentation

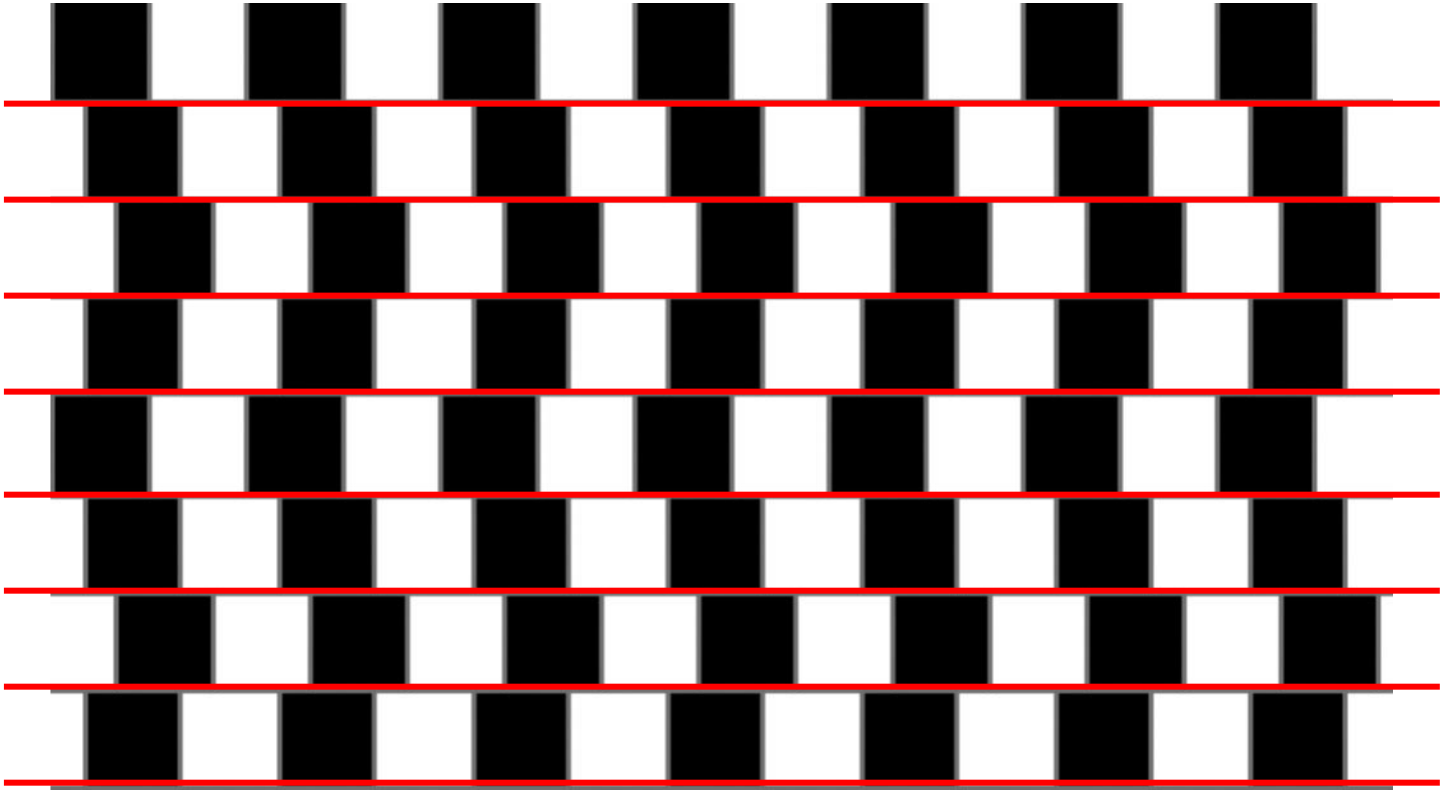
Optical Illusions



Optical Illusions

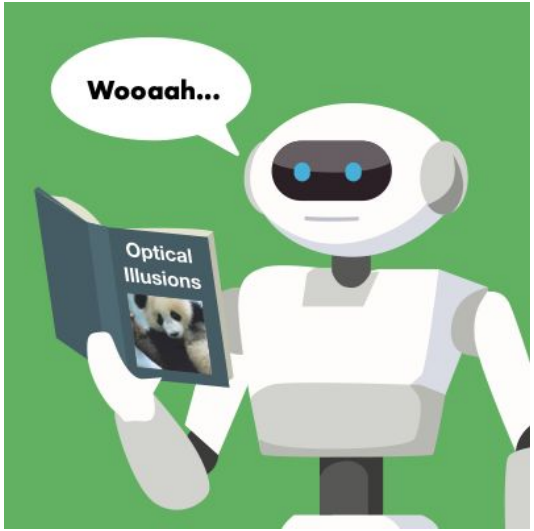


Optical Illusions



Robustness of ML Models

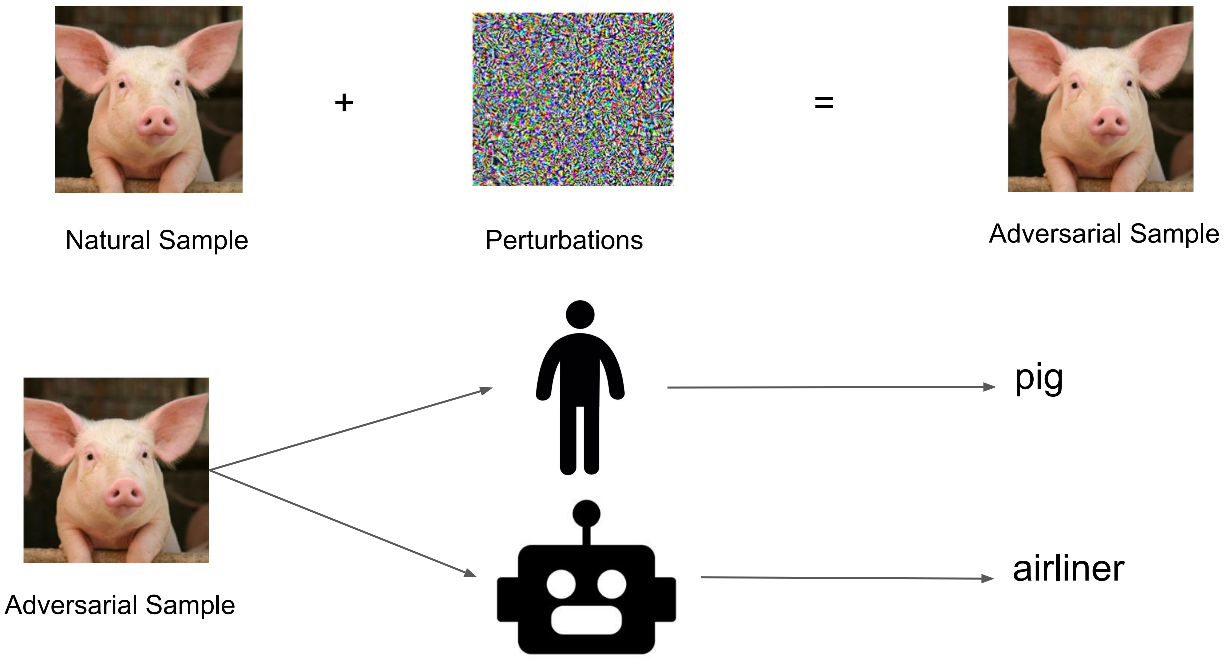
- Optical illusions trick human brains
- Can ML models be tricked?



Outline

- Optical Illusions
- Adversarial Attack
- White-box Evasion Attack
  - FGSM
  - C&W
  - Physical Attack
- Transferability of Attack
- Black-box Evasion Attack
  - Jacobian-based Data Augmentation

Adversarial Samples



Driverless Car

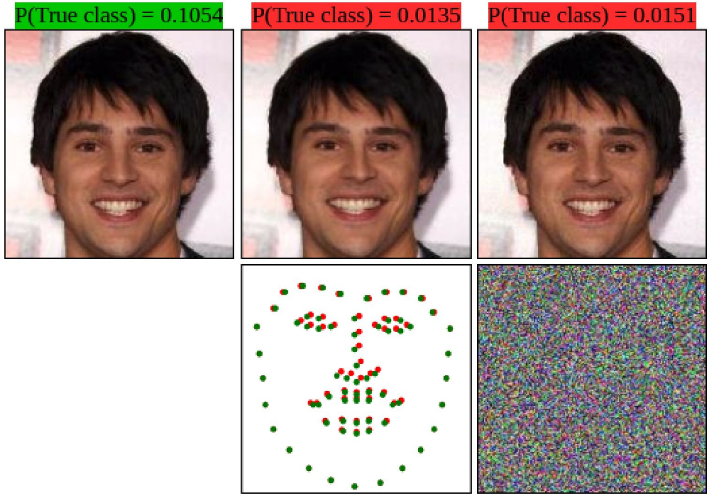
Sitawarin et al, 2018

classified as : Stop Speed Limit (30 mph)

Eykholt et al, 2018

classified as : Speed Limit (45 mph)

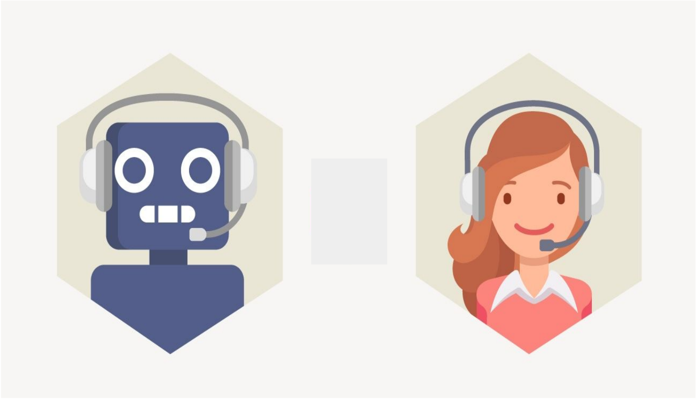
Facial Recognition



Dabouei et al, 2018

AI Chatbots

Input		
Adv agent	1x**book** value 0 1x**hat** value 1 3x**ball** value 3	
RL agent	1x**book** value 1 1x**hat** value 0 3x**ball** value 3	
Adv agent	i would like the balls and the hat	
RL agent	i need the balls and the book	
Adv agent	i need the balls and fine book	
RL agent	$\langle selection \rangle$	
Output		**Reward**
Adv agent	1x**hat** 1x**book** 3x**ball**	10/10
RL agent		0/10

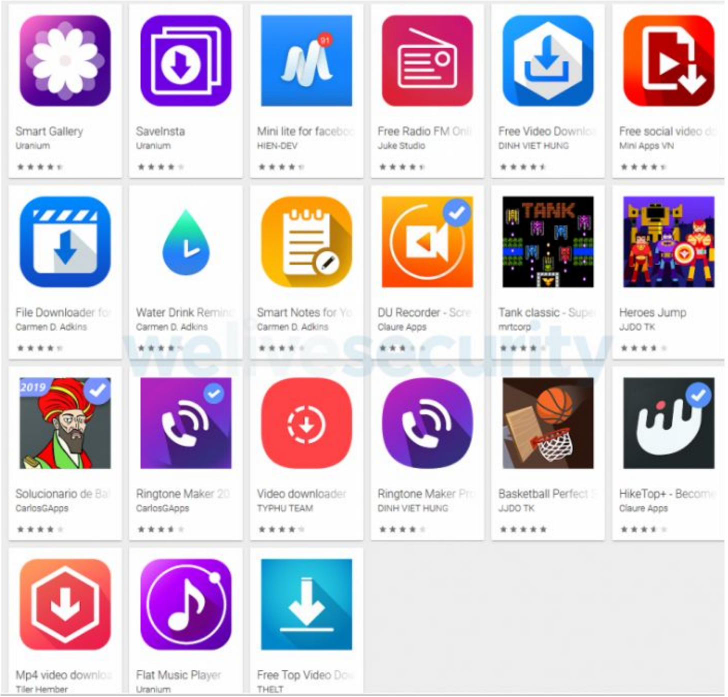


[Cheng et al, 2019]

Spam Detections

Malware Detection

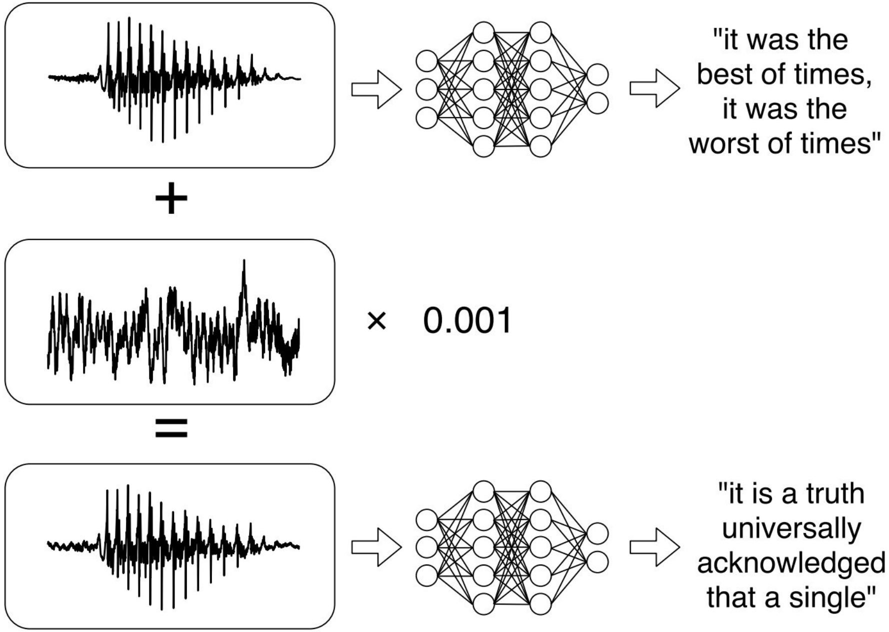
- Mislead 60% to 80% of the malicious application samples



[Grosse et al, 2016](#)

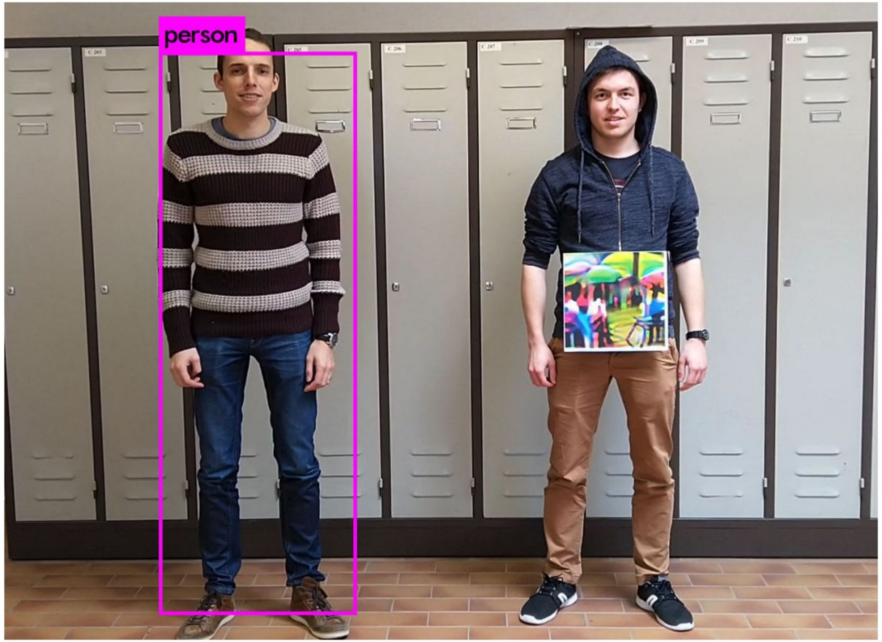
Newly discovered 42 malicious apps on Google Play store [Rohit KVN, 2019](#)

Speech Recognition



Carlini et al, 2018

Universal Adversarial Patch

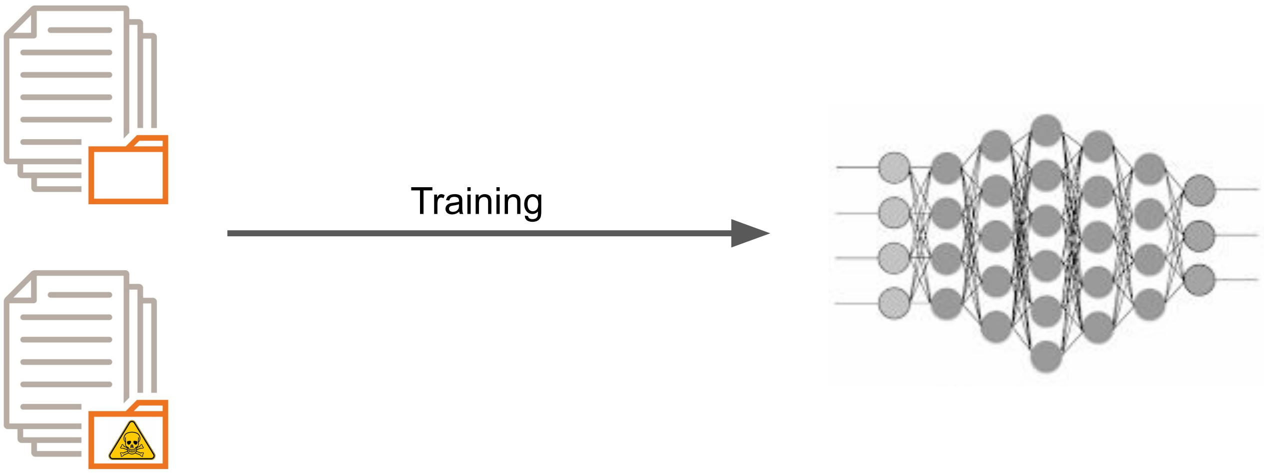


Thys et al, 2019

https://www.youtube.com/watch?v=MIbFvK2S9g8

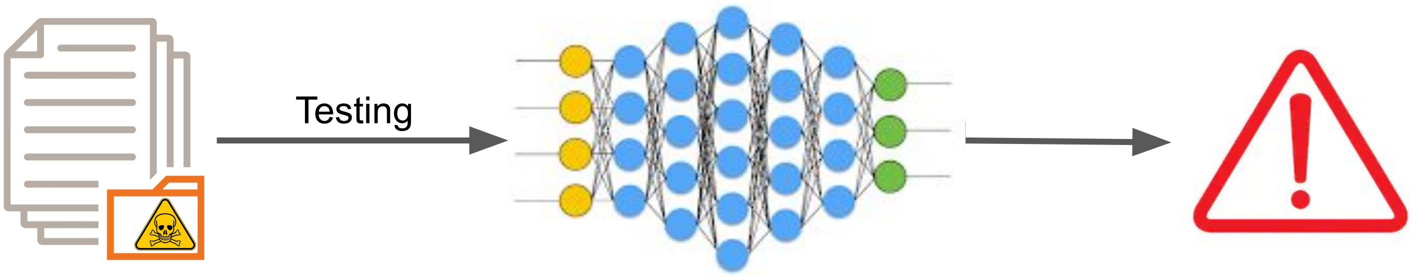
Types of Adversarial Attack

- Data Poisoning Attack
    - Insert poisonous samples during training



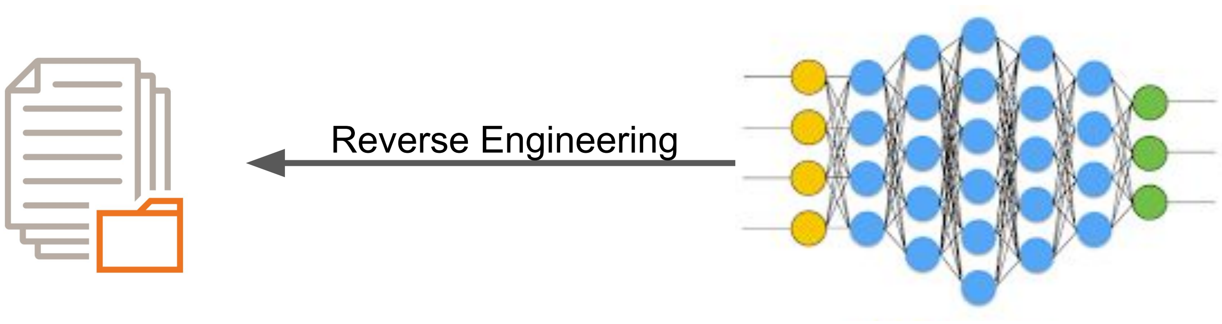
Types of Adversarial Attack

- Evasion Attack
  - Generate malicious samples to fool ML models



Types of Adversarial Attack

- Exploratory Attack
  - Reverse engineer user data from a trained model



Types of Adversarial Attack

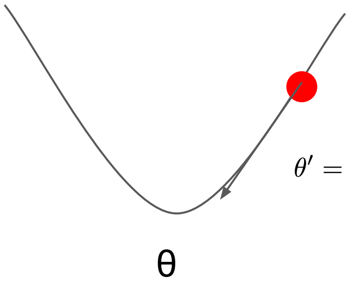
	Attack Phase	Goal
Evasion	Testing	Compromise Model Performance
Data Poisoning	Training	Compromise Model Performance
Exploratory	Testing	Explore Model Characteristics Reconstruct User Data

Outline

- Optical Illusions
- Adversarial Attack
- White-box Evasion Attack
  - FGSM
  - C&W
  - Physical Attack
- Transferability of Attack
- Black-box Evasion Attack
  - Jacobian-based Data Augmentation

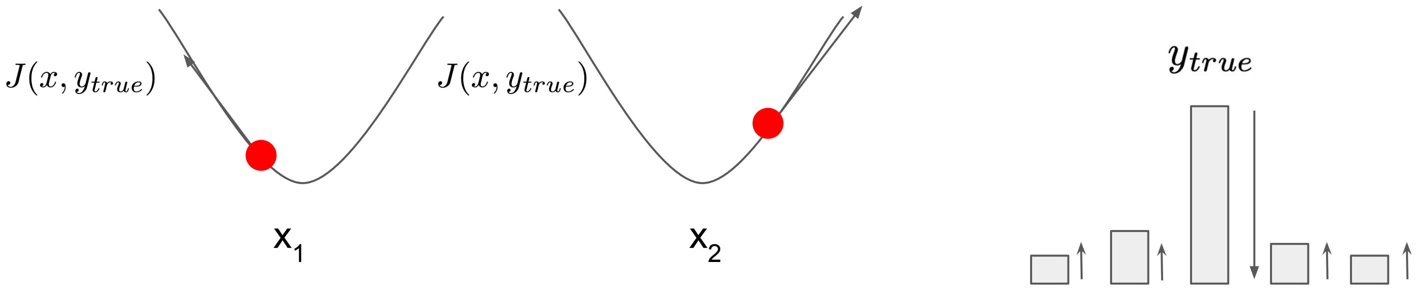
Training ML Models

$$\sum_{x,y} J(x, y_{true})$$



$$\theta' = \theta - \nabla_\theta \sum_{x,y} J(x, y_{true})$$

Fast Gradient Sign Method (FGSM)



FGSM

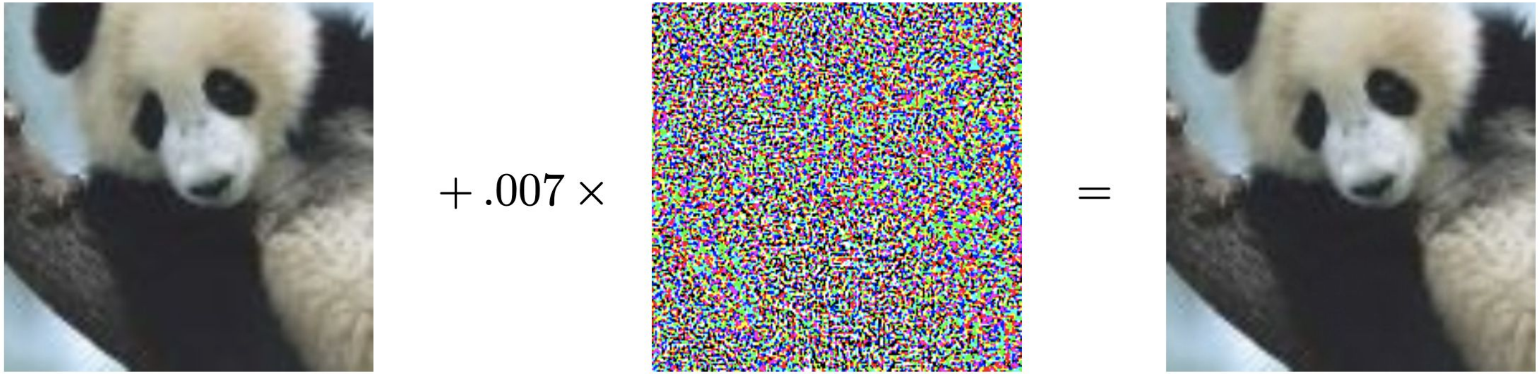
$$\boldsymbol{X}^{adv} = \boldsymbol{X} + \epsilon\, \mathrm{sign}\big(\nabla_X J(\boldsymbol{X}, y_{true})\big)$$

SGD

$$\theta' = \theta - \nabla_\theta \sum_{x,y} J(x, y_{true})$$

[Goodfellow et al, 2015]

Untargeted Adversarial Examples

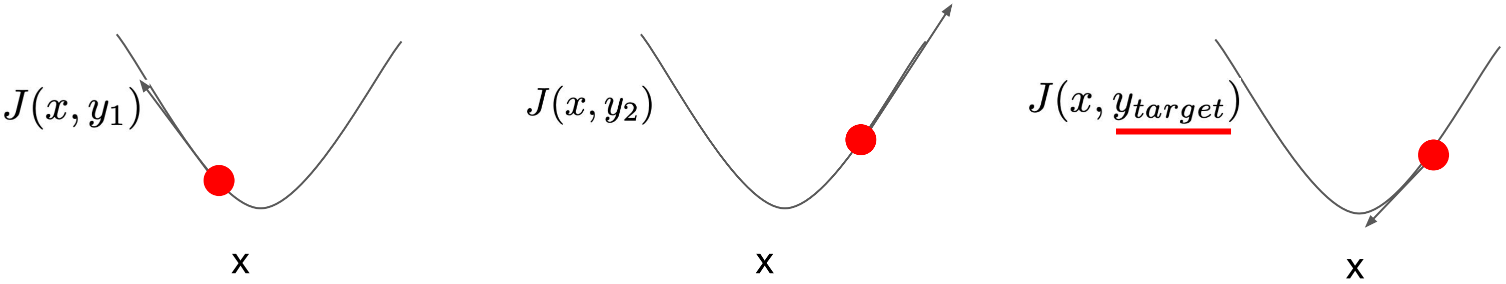


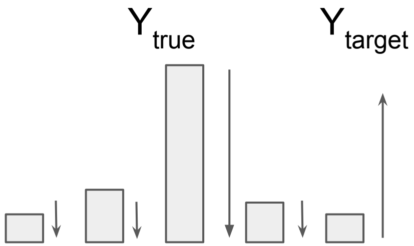
$\boldsymbol{x}$ $+ .007 \times$ $\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ $=$ $\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"panda"	"nematode"	"gibbon"
57.7% confidence	8.2% confidence	99.3 % confidence

Goodfellow et al, 2015

Targeted FGSM





Targeted FGSM

$$\boldsymbol{X}^{adv} = \boldsymbol{X} \underline{-} \epsilon \, \mathrm{sign}\big(\nabla_X J(\boldsymbol{X}, \underline{y_{target}})\big)$$

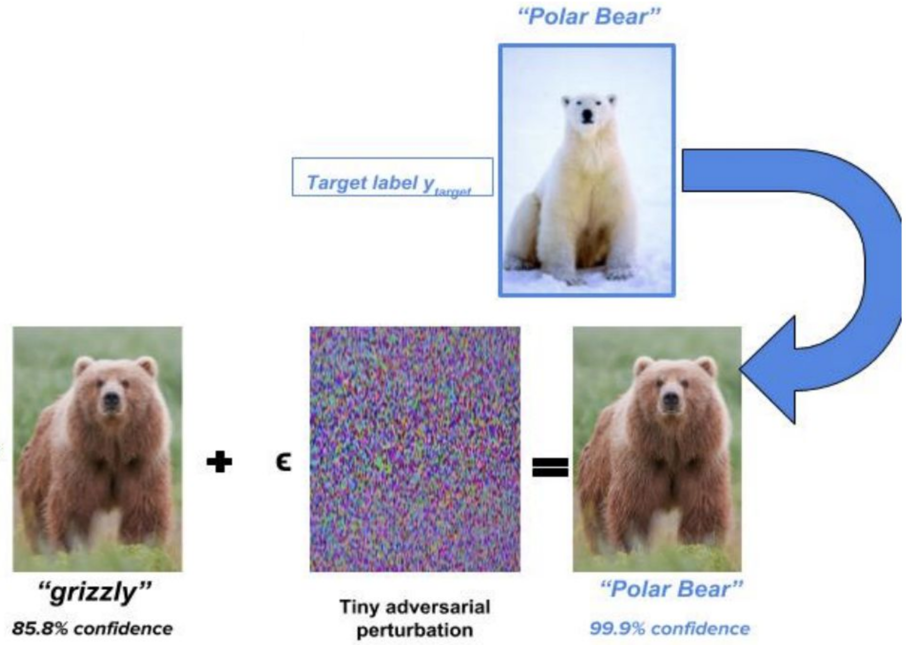
Untargeted FGSM

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

SGD: $\theta' = \theta - \nabla_\theta \sum_{x,y} J(x, y_{true})$

Kurakin et al, 2016

Targeted Adversarial Examples



Younis et al, 2019

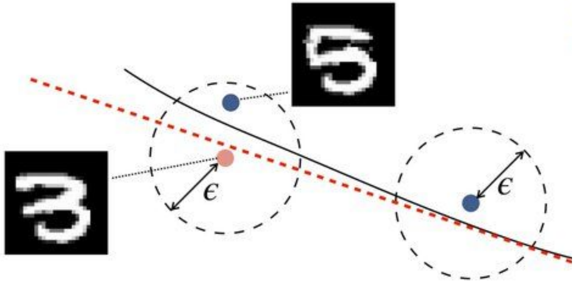
Basic Iterative Methods

- Untargeted Attack

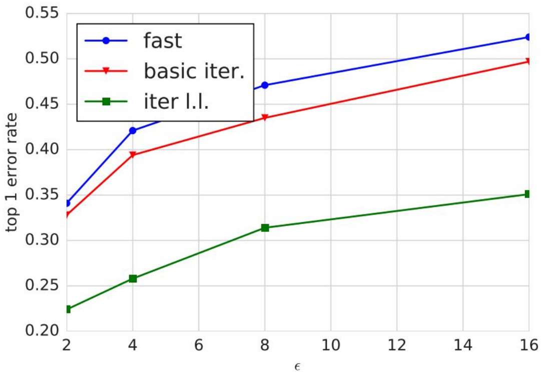
$$\boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_{N}^{adv} + \alpha\,\mathrm{sign}\big(\nabla_X J(\boldsymbol{X}_{N}^{adv}, y_{true})\big)\right\}$$

- Targeted Attack

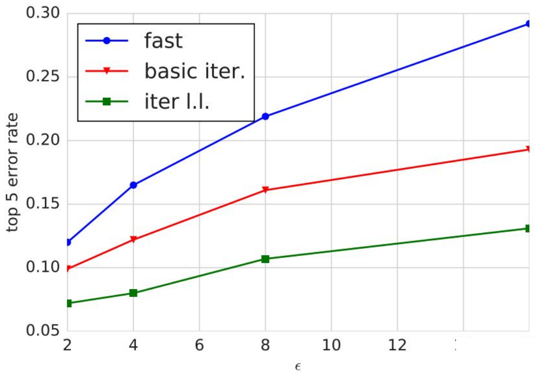
$$\boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_{N}^{adv} - \alpha\,\mathrm{sign}\big(\nabla_X J(\boldsymbol{X}_{N}^{adv}, y_{target})\big)\right\}$$



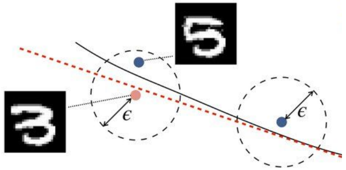
Error Rate and Perturbation Tolerance



Top 1 error rate.



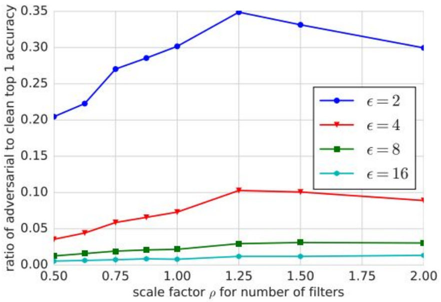
Top 5 error rate.



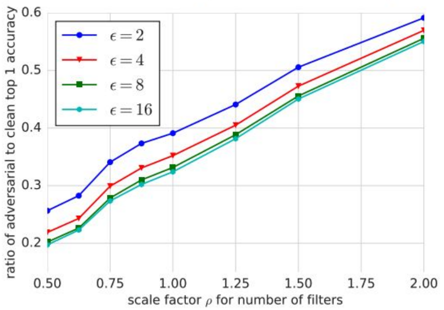
fast - FGSM
basic iter. - iterative untargeted FGSM
iter 1.1 - iteration using least likely target $y_{LL} = \arg\min_y \{p(y \mid \boldsymbol{X})\}$

Kurakin et al, 2016

Model Capacity and Attacks



One-step Targeted



Iterative Method

ρ - the factor in the number for InceptionNet
1 - unchanged
0.5 - keep half of the filters

Kurakin et al, 2016

Outline

- Optical Illusions
- Adversarial Attack
- White-box Evasion Attack
  - FGSM
  - C&W
  - Physical Attack
- Transferability of Attack
- Black-box Evasion Attack
  - Jacobian-based Data Augmentation

C&W Attack

- C&W attack
  - perturb the sample in the direction of the target class
  - minimizes the distance from the original sample x

$$
\begin{aligned}
\text{minimize} \quad & \mathcal{D}(x, x+\delta) \\
\text{such that} \quad & C(x+\delta) = t \\
& x + \delta \in [0,1]^n
\end{aligned}
$$

D - distance function
C - classifier
x - original natural sample
δ - perturbations
t - target class

Targeted FGSM

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

Carlini et al, 2017

C&W Attack

$$\begin{aligned} \text{minimize} \quad & \mathcal{D}(x, x+\delta) \\ \text{such that} \quad & C(x+\delta) = t \\ & x+\delta \in [0,1]^n \end{aligned}$$

$$\begin{aligned} \text{minimize} \quad & \mathcal{D}(x, x+\delta) + c \cdot f(x+\delta) \\ \text{such that} \quad & x+\delta \in [0,1]^n \end{aligned}$$

$$C(x+\delta) = t \iff f(x+\delta) \leq 0$$

[Carlini et al, 2016]

C&W Attack

$$
\begin{aligned}
\text{minimize} \quad & \mathcal{D}(x, x+\delta) + c \cdot f(x+\delta) \\
\text{such that} \quad & x + \delta \in [0,1]^n
\end{aligned}
$$

$$C(x+\delta) = t \quad \Longleftrightarrow \quad f(x+\delta) \leq 0$$

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

Carlini et al, 2016

Comparisons of F

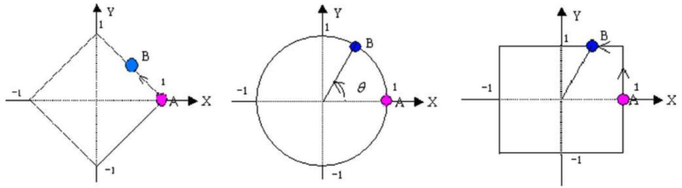
$$f_1(x') = -\text{loss}_{F,t}(x') + 1$$
$$f_2(x') = (\max_{i \neq t}(F(x')_i) - F(x')_t)^+$$
$$f_3(x') = \text{softplus}(\max_{i \neq t}(F(x')_i) - F(x')_t) - \log(2)$$
$$f_4(x') = (0.5 - F(x')_t)^+$$
$$f_5(x') = -\log(2F(x')_t - 2)$$
$$f_6(x') = (\max_{i \neq t}(Z(x')_i) - Z(x')_t)^+$$
$$f_7(x') =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

	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%

Carlini et al, 2016

C&W L_∞ Attack

$$\text{minimize} \quad c \cdot f(x+\delta) + \|\delta\|_\infty$$



(a) (b) (c)

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \qquad \|x\|_2 = \sqrt{\textstyle\sum_{i=1}^{n} x_i^2} \qquad \|x\|_\infty = \max_{1\le i\le n} |x_i|$$

FGSM

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

[Carlini et al, 2016](#)

Results

	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_∞	0.13	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 - select the least difficult class to attack among the incorrect ones
Average Case- select the target class randomly among the incorrect ones
Worst Case - select the most difficult class to attack among the incorrect ones

Carlini et al, 2016

Outline

- Optical Illusions
- Adversarial Attack
- White-box Evasion Attack
  - FGSM
  - C&W
  - Physical Attack
- Transferability of Attack
- Black-box Evasion Attack
  - Jacobian-based Data Augmentation

Physical Objects

- https://youtu.be/zQ_uMenoBCk

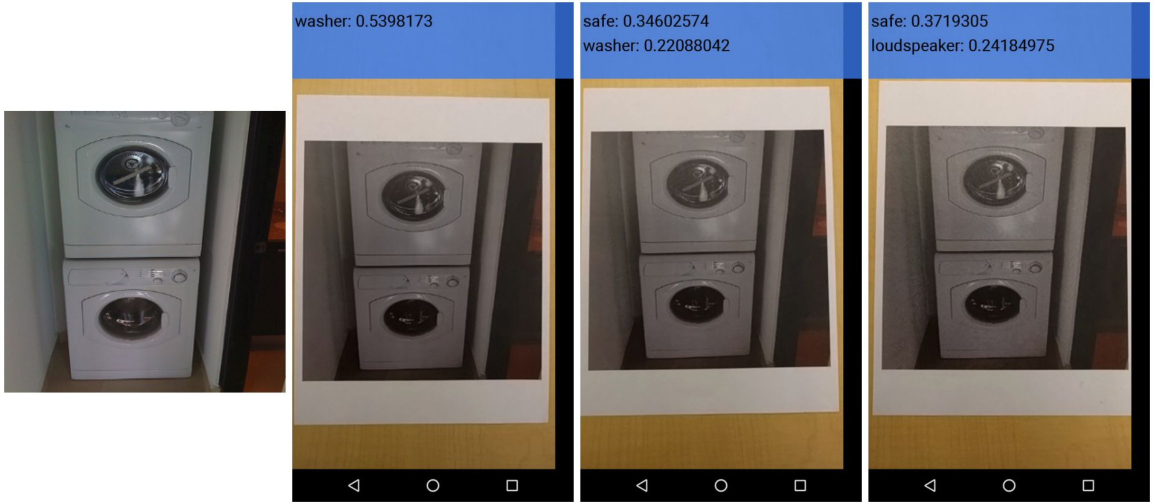
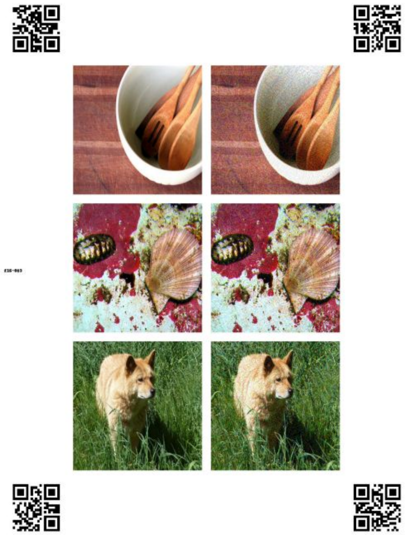


Image from dataset | Clean image | Adv. image, $\epsilon = 4$ | Adv. image, $\epsilon = 8$

[Kurakin et al, 2017]()

Evasion Attacks on Physical



Printout

Photo of printout

Cropped image

[Kurakin et al, 2017](#)

Comparisons

Adversarial method	Photos				Source images			
	Clean images		Adv. images		Clean images		Adv. images	
	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
fast $\epsilon = 16$	79.8%	91.9%	36.4%	67.7%	85.3%	94.1%	36.3%	58.8%
fast $\epsilon = 8$	70.6%	93.1%	49.0%	73.5%	77.5%	97.1%	30.4%	57.8%
fast $\epsilon = 4$	72.5%	90.2%	52.9%	79.4%	77.5%	94.1%	33.3%	51.0%
fast $\epsilon = 2$	65.7%	85.9%	54.5%	78.8%	71.6%	93.1%	35.3%	53.9%
iter. basic $\epsilon = 16$	72.9%	89.6%	49.0%	75.0%	81.4%	95.1%	28.4%	31.4%
iter. basic $\epsilon = 8$	72.5%	93.1%	51.0%	87.3%	73.5%	93.1%	26.5%	31.4%
iter. basic $\epsilon = 4$	63.7%	87.3%	48.0%	80.4%	74.5%	92.2%	12.7%	24.5%
iter. basic $\epsilon = 2$	70.7%	87.9%	62.6%	86.9%	74.5%	96.1%	28.4%	41.2%
l.l. class $\epsilon = 16$	71.1%	90.0%	60.0%	83.3%	79.4%	96.1%	1.0%	1.0%
l.l. class $\epsilon = 8$	76.5%	94.1%	69.6%	92.2%	78.4%	98.0%	0.0%	6.9%
l.l. class $\epsilon = 4$	76.8%	86.9%	75.8%	85.9%	80.4%	90.2%	9.8%	24.5%
l.l. class $\epsilon = 2$	71.6%	87.3%	68.6%	89.2%	75.5%	92.2%	20.6%	44.1%

fast - FGSM
iter. basic - iterative FGSM
l.l. - iterative FGSM with least likely target $y_{LL} = \arg\min_y \{p(y \mid \boldsymbol{X})\}$

Kurakin et al, 2017

Comparisons (Filtered)

Adversarial method	Photos				Source images			
	Clean images		Adv. images		Clean images		Adv. images	
	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
fast $\epsilon = 16$	81.8%	97.0%	5.1%	39.4%	100.0%	100.0%	0.0%	0.0%
fast $\epsilon = 8$	77.1%	95.8%	14.6%	70.8%	100.0%	100.0%	0.0%	0.0%
fast $\epsilon = 4$	81.4%	100.0%	32.4%	91.2%	100.0%	100.0%	0.0%	0.0%
fast $\epsilon = 2$	88.9%	99.0%	49.5%	91.9%	100.0%	100.0%	0.0%	0.0%
iter. basic $\epsilon = 16$	93.3%	97.8%	60.0%	87.8%	100.0%	100.0%	0.0%	0.0%
iter. basic $\epsilon = 8$	89.2%	98.0%	64.7%	91.2%	100.0%	100.0%	0.0%	0.0%
iter. basic $\epsilon = 4$	92.2%	97.1%	77.5%	94.1%	100.0%	100.0%	0.0%	0.0%
iter. basic $\epsilon = 2$	93.9%	97.0%	80.8%	97.0%	100.0%	100.0%	0.0%	1.0%
l.l. class $\epsilon = 16$	95.8%	100.0%	87.5%	97.9%	100.0%	100.0%	0.0%	0.0%
l.l. class $\epsilon = 8$	96.0%	100.0%	88.9%	97.0%	100.0%	100.0%	0.0%	0.0%
l.l. class $\epsilon = 4$	93.9%	100.0%	91.9%	98.0%	100.0%	100.0%	0.0%	0.0%
l.l. class $\epsilon = 2$	92.2%	99.0%	93.1%	98.0%	100.0%	100.0%	0.0%	0.0%

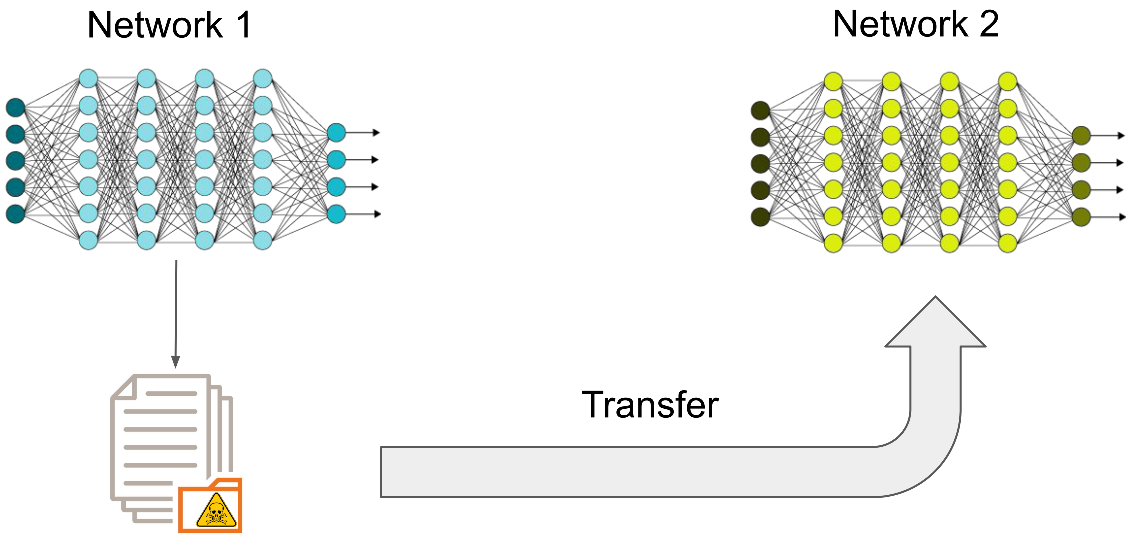
fast - FGSM
iter. basic - iterative FGSM
l.l. - iterative FGSM with least likely target $y_{LL} = \arg\min_y \{p(y \mid \boldsymbol{X})\}$

Kurakin et al, 2017

Outline

- Optical Illusions
- Adversarial Attack
- White-box Evasion Attack
  - FGSM
  - C&W
  - Physical Attack
- Transferability of Attack
- Black-box Evasion Attack
  - Jacobian-based Data Augmentation

Transferability of Attack



Transferability of Attack

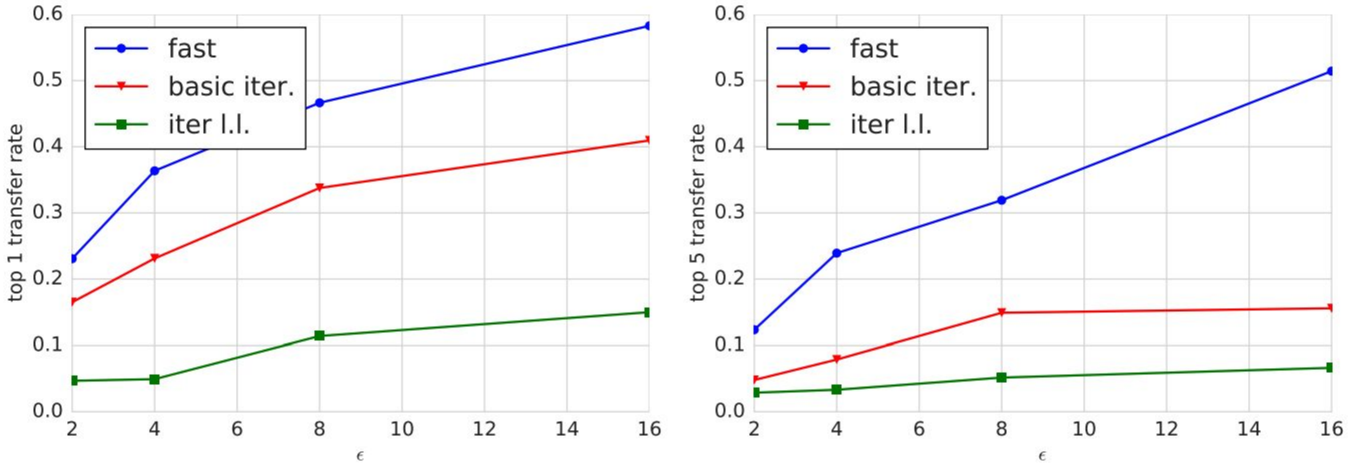
	source model	FGSM target model A	B	C	D	basic iter. target model A	B	C	D	iter l.l. target model A	B	C	D
top 1	A (v3)	100	56	58	47	100	46	45	33	100	13	13	9
	B (v3)	58	100	59	51	41	100	40	30	15	100	13	10
	C (v3 ELU)	56	58	100	52	44	44	100	32	12	11	100	9
	D (v4)	50	54	52	100	35	39	37	100	12	13	13	100
top 5	A (v3)	100	50	50	36	100	15	17	11	100	8	7	5
	B (v3)	51	100	50	37	16	100	14	10	7	100	5	4
	C (v3 ELU)	44	45	100	37	16	18	100	13	6	6	100	4
	D (v4)	42	38	46	100	11	15	15	100	6	6	6	100

A - Inception v3
B - Inception v3 with different initialization
C - Inception v3 with ELU activation
D - Inception v4

iter. basic - iterative FGSM
itera l.l. - iterative FGSM with least likely target

Kurakin et al, 2017

Transferability of Attack



Top 1 transfurability. Top 5 transferability.

Targeted FGSM

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

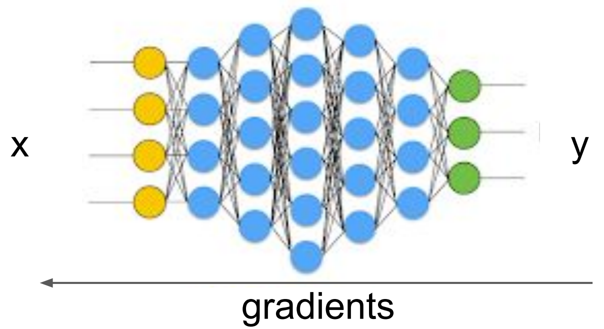
Kurakin et al, 2017

Outline

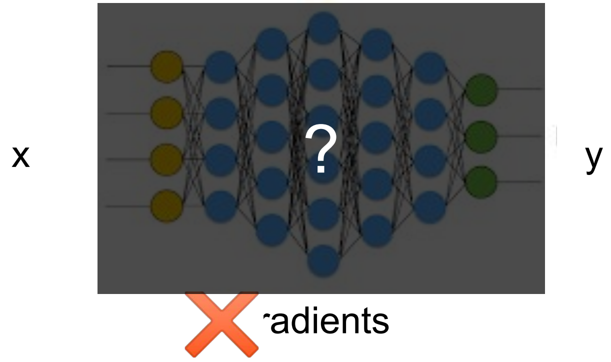
- Optical Illusions
- Adversarial Attack
- White-box Evasion Attack
    - FGSM
    - C&W
    - Physical Attack
- Transferability of Attack
- Black-box Evasion Attack
    - Jacobian-based Data Augmentation

White-box and Black-box Attack

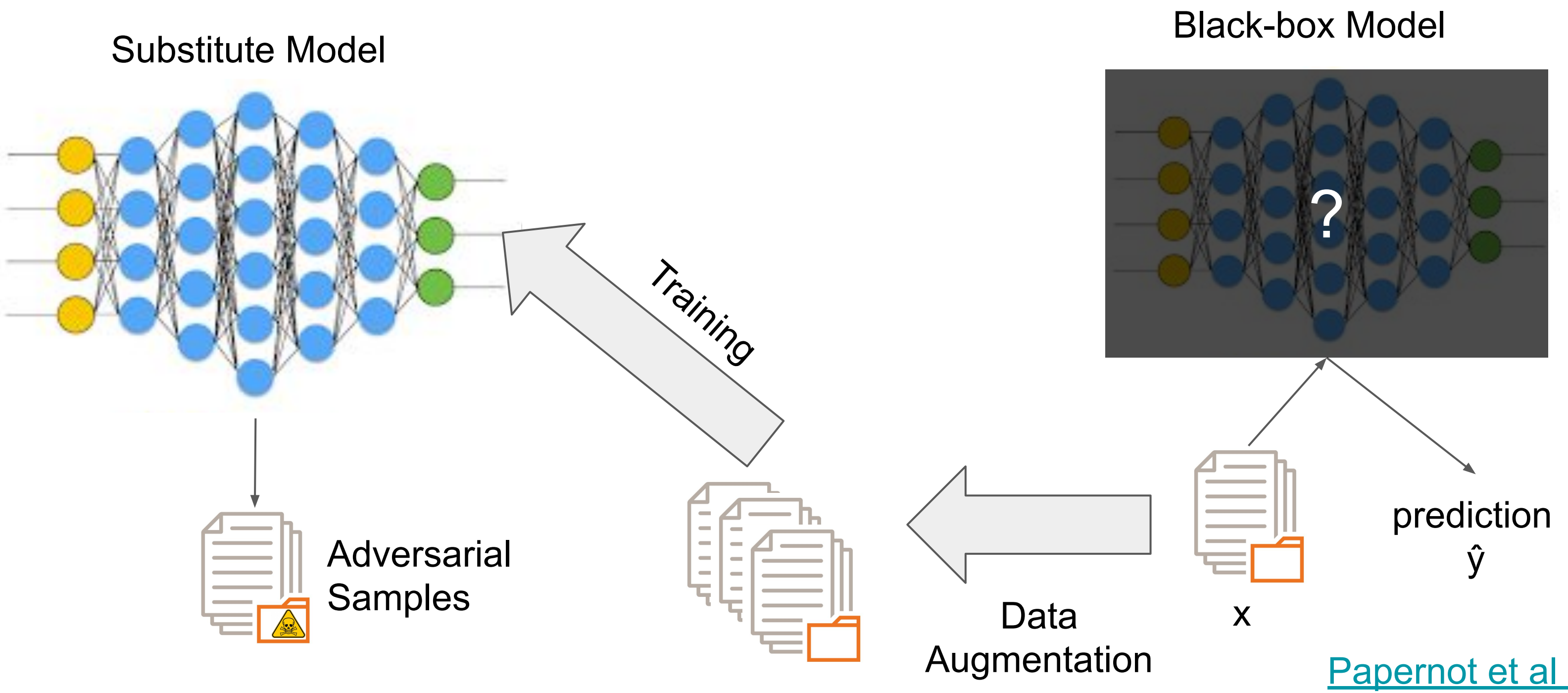
White-box Setting



Black-box Setting



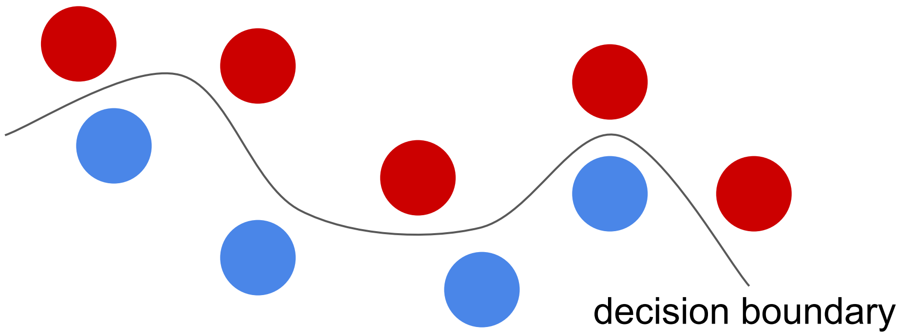
Substitute Model for Black-box Adversarial Attack

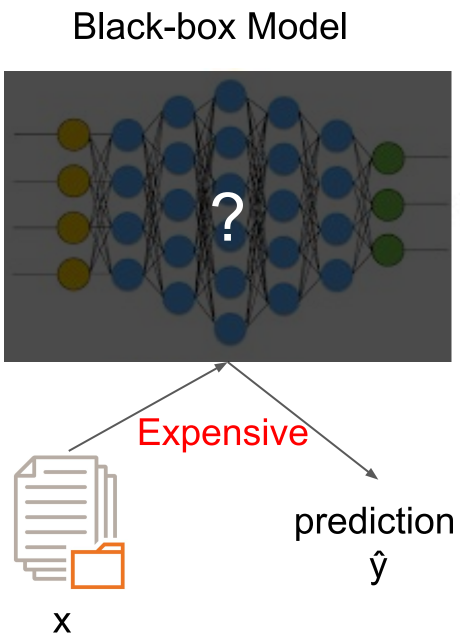


[Papernot et al 2016]

Data Augmentation for the Substitute Model

- Data annotation using the black-box model is expensive
- It's difficult to find a good dataset x to probe the performance of the black-box model





Jacobian-based Data Augmentation

- Start with an initial dataset $S_0 = \{x_i\}$
- Expand it in the direction of the model prediction $\hat{y}_i$ for each x_i

$$S_{\rho+1} = \{\vec{x} + \lambda \cdot \mathrm{sgn}(J_F[\underbrace{\tilde{O}(\vec{x})}_{\text{prediction of the black-box model}}]) : \vec{x} \in S_\rho\} \cup S_\rho$$

f: $\mathbb{R}^n \to \mathbb{R}$

$$\mathrm{grad}_x(f) := \left[\frac{\partial f}{\partial x_1} \; \frac{\partial f}{\partial x_2} \; \cdots \; \frac{\partial f}{\partial x_n}\right]\Bigg|_x$$

f: $\mathbb{R}^n \to \mathbb{R}^m$

$$\mathrm{Jac}_x(f) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}\Bigg|_x$$

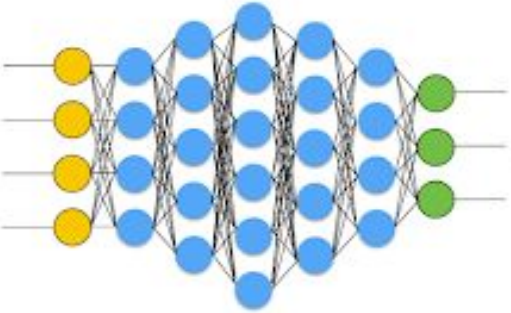
Jacobian-based Data Augmentation

- Start with an initial dataset $S_0 = \{x_i\}$
- Expand it in the direction of the model prediction $\hat{y}_i$ for each x_i

$$S_{\rho+1} = \{\vec{x} + \lambda \cdot \mathrm{sgn}(J_F[\underbrace{\tilde{O}(\vec{x})}_{\text{prediction of the black-box model}}]) : \vec{x} \in S_\rho\} \cup S_\rho$$

$f: \mathbb{R}^n \to \mathbb{R}^m$

$x_{i,\ 1..n}$



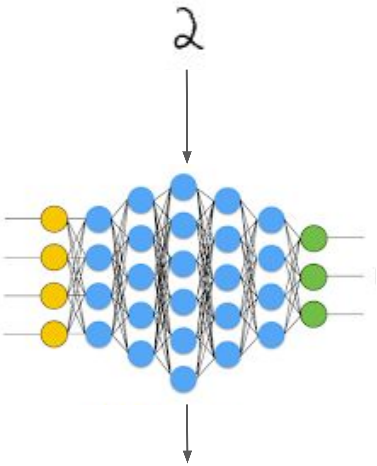
$\hat{y}_{i,1..m}$

$$\mathrm{Jac}_x(f) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}_x \quad \tilde{O}(\vec{x})$$

Jacobian-based Data Augmentation

$\vec{x}$

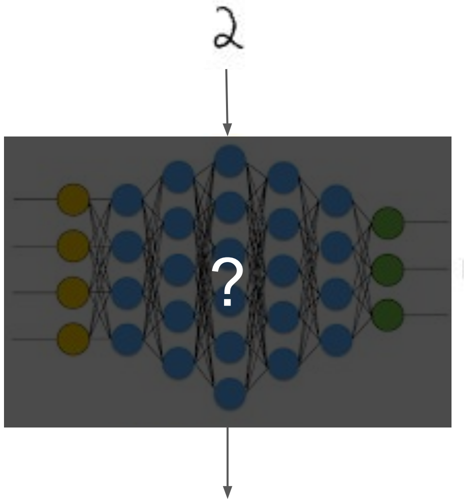
$$\vec{x'} = \vec{x} + \lambda \cdot \mathrm{sgn}(J_F[\tilde{O}(\vec{x})])$$



y= "2"

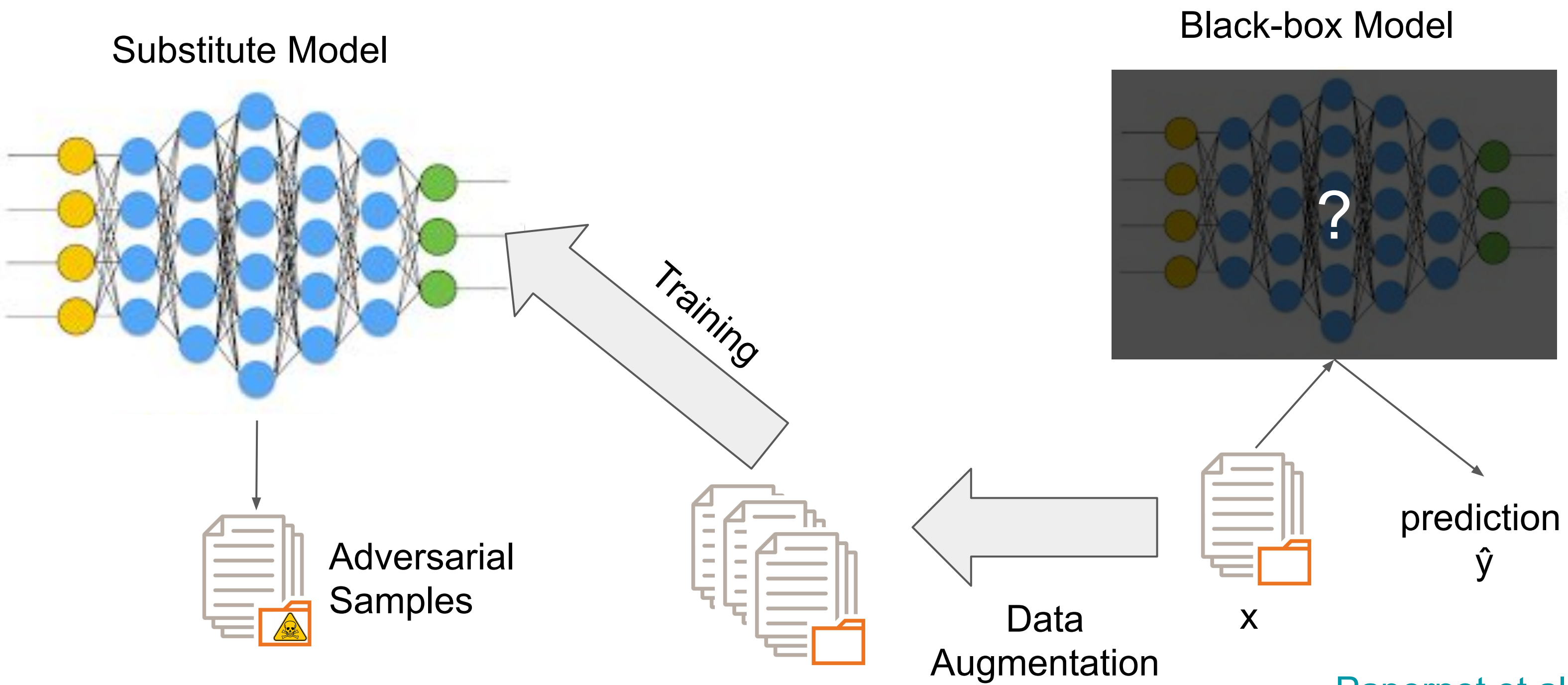
$$\mathrm{Jac}_x(f) = \begin{array}{cc} \left[\begin{matrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{matrix}\right]_x & \begin{matrix} 1 \\ 2 \\ \\ n \end{matrix} \end{array}$$

y= $\tilde{O}(\vec{x'})$



$\tilde{O}(\vec{x})]$ ="1"

Substitute Model for Black-box Adversarial Attack



Papernot et al 2016

Results on Attacking Amazon and Google Services

		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
$\rho = 6^*$	2,000	95.68	95.83	91.57	**97.72**

DNN - Deep Neural Networks
LG - Logistic Regression
* - reservoir sampling

$$\vec{x} + \lambda \cdot \mathrm{sgn}(J_F[\tilde{O}(\vec{x})])$$

$$\lambda_\rho = \lambda \cdot (-1)^{\left\lfloor \frac{\rho}{\tau} \right\rfloor}$$

Reading Assignments

- Sitawarin, Chawin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. Darts: Deceiving autonomous cars with toxic signs. arXiv 2018
- Ilyas, Andrew, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features, NeurIPS 2019
- Athalye, Anish, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples, ICML 2018
- Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations, CVPR 2017
- Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, CVPR 2016