CS 499/579: Trustworthy ML

04.06: Adversarial Examples (Prelim.)

Tu/Th 10:00 – 11:50 am

Instructor: Sanghyun Hong

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL
Secure AI Systems Lab

Heads-up!

- Due dates
  - 4/11: Written paper critique
  - 4/11: Team-up for the term project (teaming with your awesome colleagues!)
  - 4/13: HW 1 due
- Announcement
  - Zoom link for the lectures is available on Canvas (left navigation pane)
  - Made the request for your GPU cluster access
- Call for actions
  - Term project team-up
  - In-class presentation sign-ups
- Any questions?

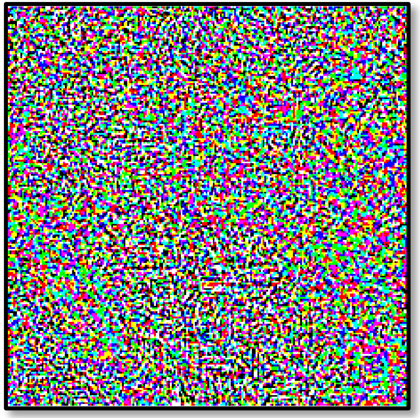
Topics for Today

- Motivation
  - What is it?
  - Why do we care about adversarial examples?

Motivation: what is the adversarial example?

- An input to a neural network that contains human-imperceptible perturbations carefully crafted with the objective of fooling the network

$+ \, 0.007 \times$



$=$

Prediction: **Panda**

Human-imperceptible Noise

Prediction: **Gibbon**

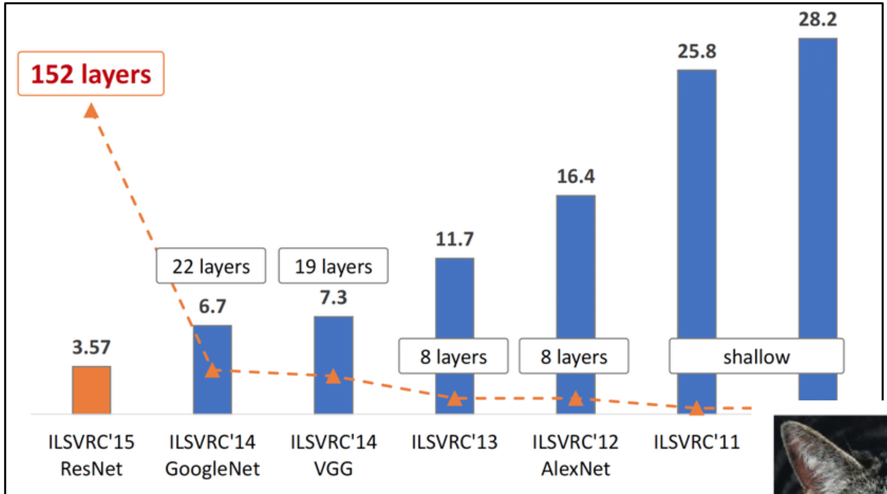
Goodfellow et al., *Explaining and Harnessing Adversarial Examples, International Conference on Learning Representations (ICLR)*, 2015.

Motivation: why do we care about the adversarial example?

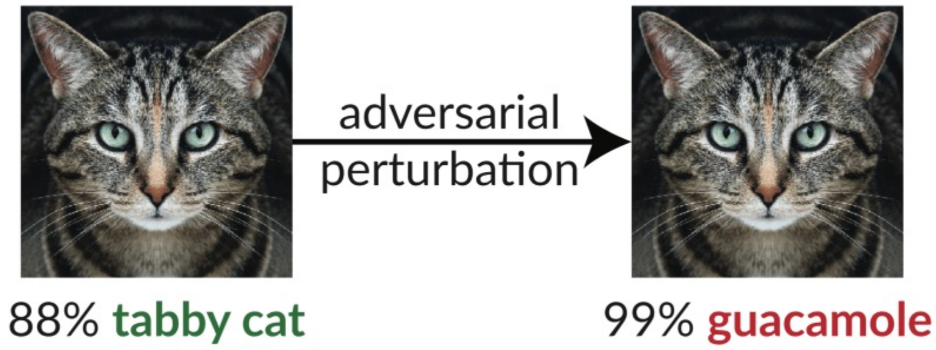
- from the security perspective: it makes ML-enabled systems unavailable

Motivation: why do we care about the adversarial example?

- from the ML perspective: it is counter-intuitive



ImageNet Classification Top-5 Error Rate (%)



Topics for Today

- Motivation
  - What is it?
  - Why do we care about adversarial examples?
- Research questions
  - How can we find adversarial examples?
  - How can a real-world attacker exploit them in practice?
  - How can we remove adversarial examples?

RQ: How can we find adversarial examples?

- Sub research questions
  - SRQ1: What is the attack scenario (threat model)?
  - SRQ2: What are the goals for the attacker (under the threat model)?
  - SRQ3: What is the right method for finding adversarial examples?
  - SRQ4: What properties do an adversarial examples exploit?

SRQ 1: What is the attack scenario (threat model)?

- Evasion attack
  - **Goal:**
    - Craft human-imperceptible perturbations
      that can make a test-time sample misclassified by a model
  - **Knowledge:**
    - (Trivial) Test-time samples to attack
    - Training data
    - Model architecture and parameters
    - Two cases:
      - White-box: knows training data and model internals
      - Black-box: does not know both
  - **Capability:**
    - Sufficient computational power to craft adversarial examples

SRQ 2: What are the goals for the attacker?

- Evasion attack (cont'd)
  - Formulation:

$$\mathbf{x}^* = \arg \max_{\mathrm{x}} \quad \hat{g}(\mathbf{x}\ , \mathrm{y})$$

$$\text{s.t.} \quad d(\mathbf{x}, \mathbf{x}^0) \leq d_{\max}.$$

    - x: test-time sample
    - x^0, x': adversarial examples
    - g(x, y): error (loss) computed on a test-time sample w.r.t the true label y
    - $d(x, x^0)$: pixel-wise distance between x and x^0 (typically L1, L2, L-inf)
  - Specific goals:
    - Untargeted attack: any misclassification
    - Targeted attack: misclassification toward a specific class

SRQ 3: How can we find adversarial examples?

- Potential approaches
  - **Handcraft:** manipulate pixels that are likely to lead to adversarial examples
  - **Game-theoretic approach:** minimax or Nash-equilibrium
  - (or easily) **Gradient-based approach:**
    - Thanks to automatic differentiation deep learning frameworks (PyTorch) offer, we can compute the input that will increase the error g(x, y) – loss function

SRQ 3: Gradient-based attack for finding adversarial example

- Attack formulated by Biggio *et al.*[1]

Algorithm 1 Gradient-descent evasion attack

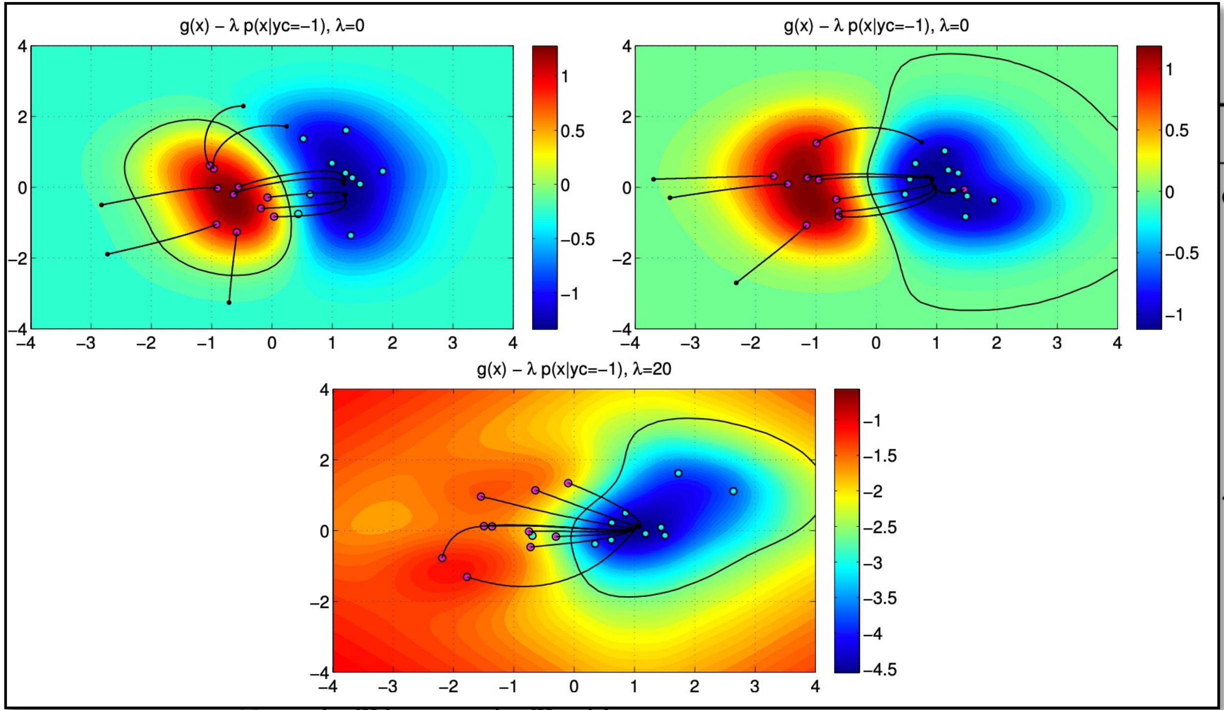
Input: $\mathbf{x}^0$, the initial attack point; t, the step size; λ, the trade-off parameter; $\epsilon > 0$ a small constant.

Output: $\mathbf{x}^*$, the final attack point.

$m \leftarrow 0$.
repeat
  $m \leftarrow m + 1$
  Set $\nabla F(\mathbf{x}^{m-1})$ to a unit vector aligned with $\nabla g(\mathbf{x}^{m-1}) - \lambda \nabla p(\mathbf{x}^{m-1} | y^c = -1)$.
  $\mathbf{x}^m \leftarrow \mathbf{x}^{m-1} - t \nabla F(\mathbf{x}^{m-1})$
  if $d(\mathbf{x}^m, \mathbf{x}^0) > d_{\max}$ **then**
    Project $\mathbf{x}^m$ onto the boundary of the feasible region.
  end if
until $F(\mathbf{x}^m) - F(\mathbf{x}^{m-1}) < \epsilon$
return: $\mathbf{x}^* = \mathbf{x}^m$

[1] Biggio et al., Evasion Attacks against Machine Learning Models at Test Time

SRQ 3: Gradient-based attack for finding adversarial example



off parameter; $\epsilon > 0$ a

$\lambda \nabla p(\mathbf{x}^{m-1}|y^c = -1)$.

Mimicry Component

9: **until** $F(\mathbf{x}^m) - F(\mathbf{x}^{m-1}) < \epsilon$

10: **return:** $\mathbf{x}^* = \mathbf{x}^m$

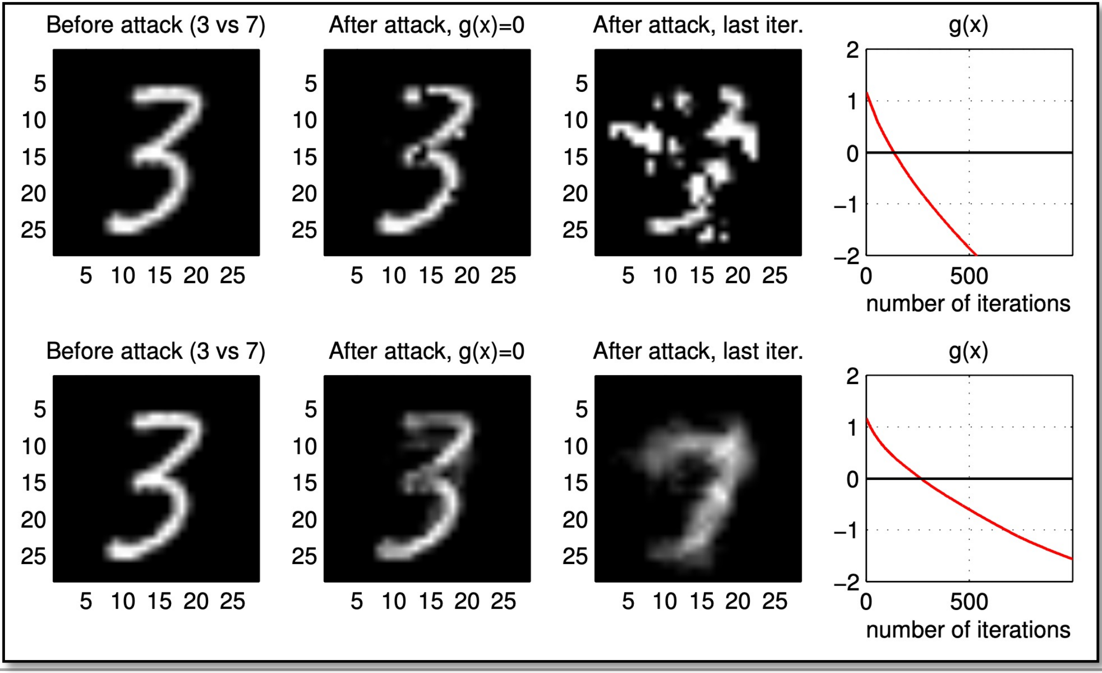
[1] Biggio et al., Evasion Attacks against Machine Learning Models at Test Time

SRQ 3: Gradient-based attack results

- 2 Tasks (MNIST-3/7 and PFD Malware Detection)
  - **(Toy example)** MNIST-3/7
    - **Task:** Binary classification problem 3 vs. 7
    - **Attacker:**
      - PK (white-box)
      - Limited: bound the perturbations to $||x' - x||_{l_1} \leq 5000/255$
    - **Model:** SVM (w. $C = 1$)
    - **Targets:** 100 randomly chosen training samples [?!]

SRQ 3: Gradient-based attack results – cont'd

- MNIST-3/7 Results
  - **(I assume)** The crafted samples cause misclassifications on the SVM classifier
  - Mimicry component results in a visually-nice samples **(why it's important?)**

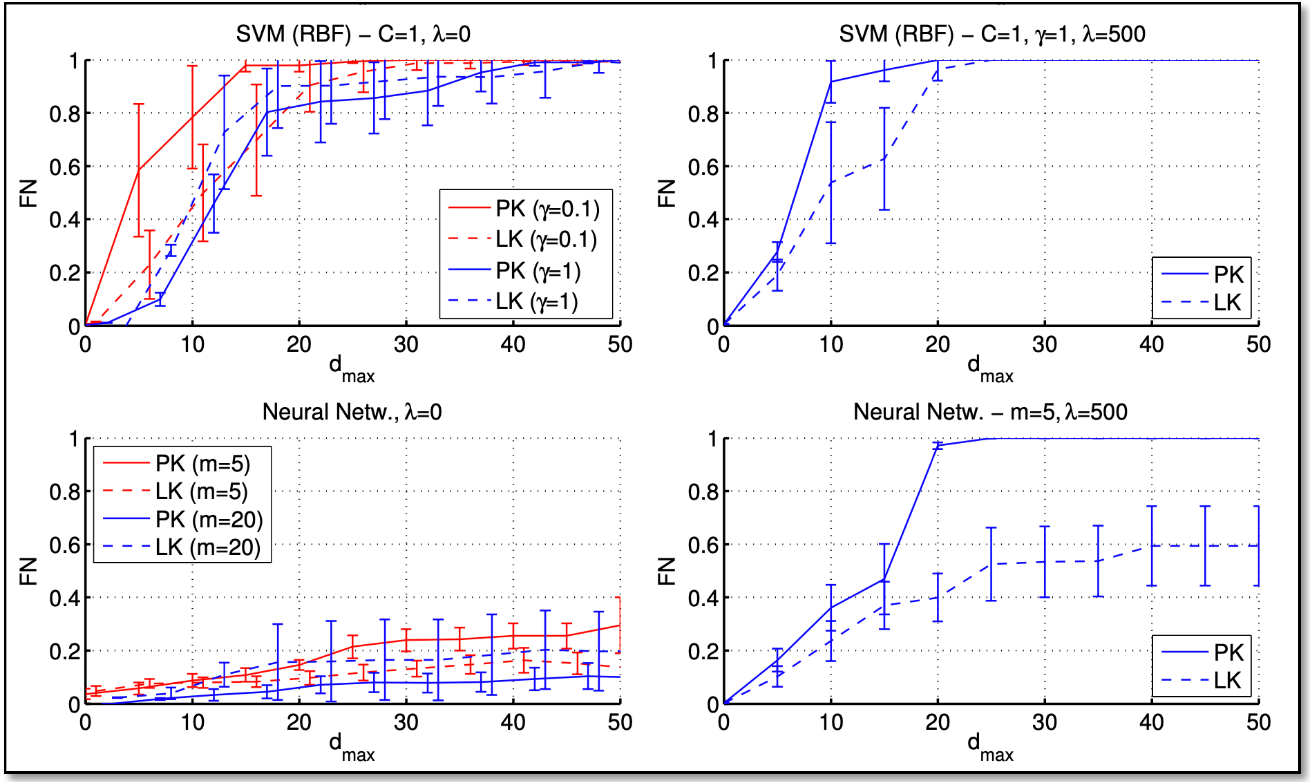


SRQ 3: Gradient-based attack results – cont'd

- 2 Tasks (MNIST-3/7 and PDF Malware Detection)
  - PDF Malware Detection
    - **Task:** Binary classification problem
    - **Attacker:**
      - PK (white-box) and LK (Black-box)
      - Limited: bound the perturbations to $||x' - x||_{l_1} \leq 5000/255$
    - **Model:** SVM (w. $C = 100, \gamma = 1$ RBF) or neural network
    - **Targets:** 100 randomly chosen training samples

SRQ 3: Gradient-based attack results – cont'd

- PDF Malware Detection Results



SRQ 4: What properties do adversarial examples exploit?

- Revisit'ed: common belief at that time (on neural networks)
  - B1: Prior work in ML empirically show that neurons represent certain features
    - People use this intuition to find *semantically-similar* inputs
    - Neural networks may have the ability to *disentangle* features at neuron-level
  - B2: Neural Networks are stable when there is small perturbations to their inputs
    - *Random perturbations* to inputs are difficult to change networks' predictions

SRQ 4: What properties do adversarial examples exploit?

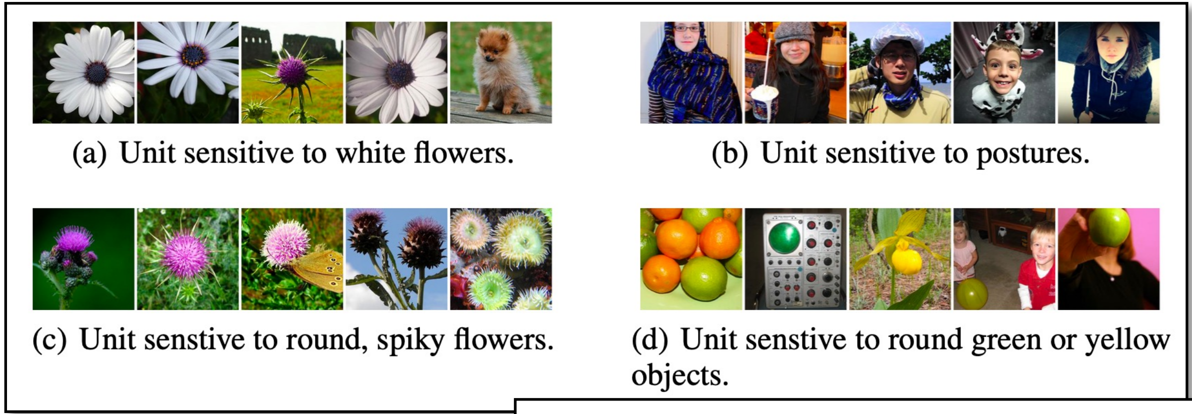
- B1: Neurons represent certain features
- Re-evaluate this hypothesis[1]:
  - Find a set of inputs that maximally increases
    - The activation of i-th hidden neuron
    - The activation of random vector
  - Compare those two sets of inputs
  - More formally:

$$\boxed{x' = \arg\max_{x \in \mathcal{I}} \langle \phi(x), e_i \rangle}$$

$$\boxed{x' = \arg\max_{x \in \mathcal{I}} \langle \phi(x), v \rangle}$$

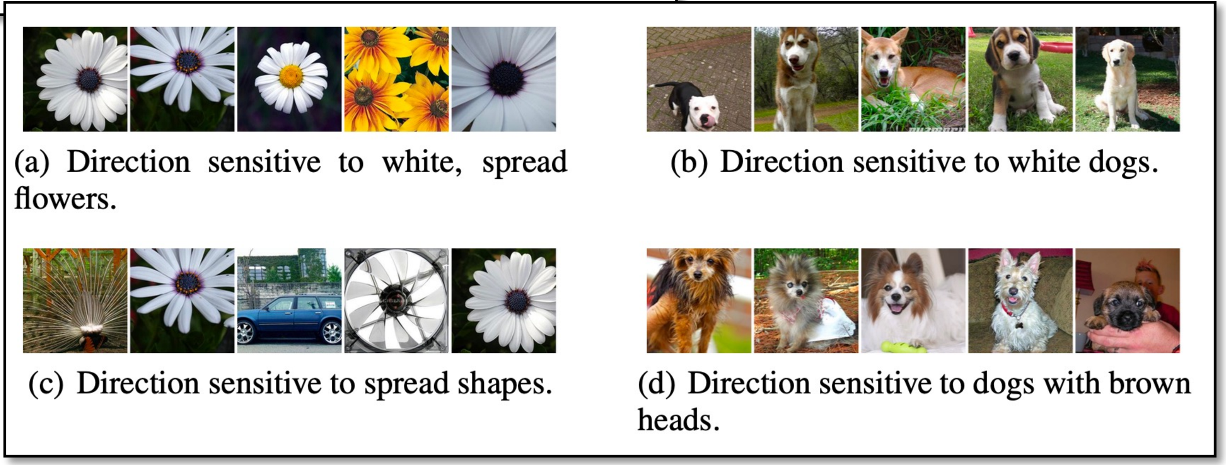
[1] Szegedy et al., Intriguing Properties of Neural Networks, ICLR

SRQ 4: What properties do adversarial examples exploit?



(a) Unit sensitive to white flowers.
(b) Unit sensitive to postures.
(c) Unit senstive to round, spiky flowers.
(d) Unit senstive to round green or yellow objects.

Images that activates a ce-rtain neuron the most



(a) Direction sensitive to white, spread flowers.
(b) Direction sensitive to white dogs.
(c) Direction sensitive to spread shapes.
(d) Direction sensitive to dogs with brown heads.

Images that activates a random dir. the most

SRQ 4: What properties do adversarial examples exploit?

- B2: Neural networks are resilient to small input perturbations
- Re-evaluate this hypothesis[1]:
  - Let's find the worst-case inputs: **adversarial examples**
  - Solve a constrained-optimization problem

> - Minimize $\|r\|_2$ subject to:
>   1. $f(x+r) = l$
>   2. $x + r \in [0,1]^m$

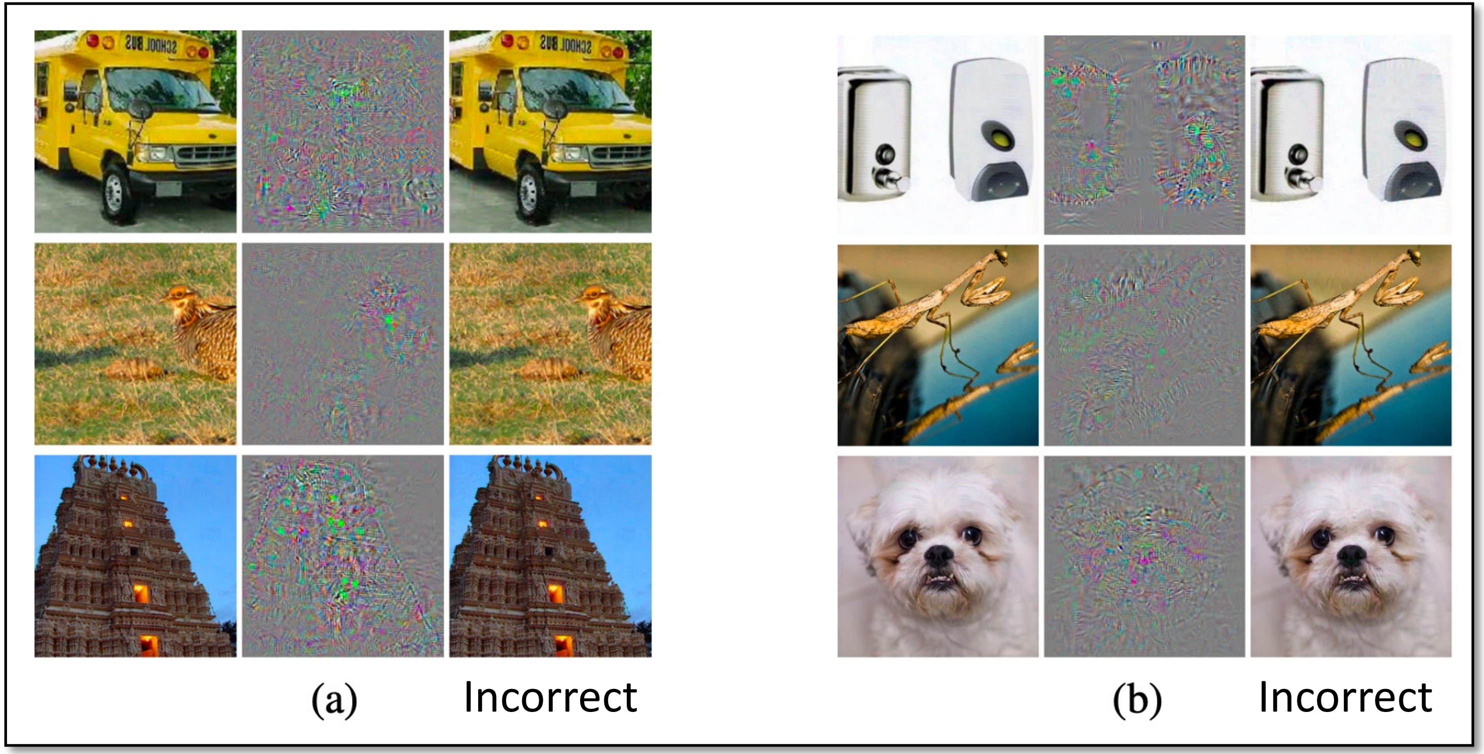
  - Formally:

> - Minimize $c|r| + \text{loss}_f(x+r, l)$ subject to $x + r \in [0,1]^m$

[1] Szegedy et al., Intriguing Properties of Neural Networks, ICLR

SRQ 4: What properties do adversarial examples exploit?

- Results from attacking AlexNet models trained on ImageNet



SRQ 4: What properties do adversarial examples exploit?

- Important lessons:
  - Random perturbations are **NOT** the right way to measure the stability of neural networks
  - Adversarial examples **transfer**

	FC10(10^{-4})	FC10(10^{-2})	FC10(1)	FC100-100-10	FC200-200-10	AE400-10	Av. distortion
FC10(10^{-4})	100%	11.7%	22.7%	2%	3.9%	2.7%	0.062
FC10(10^{-2})	87.1%	100%	35.2%	35.9%	27.3%	9.8%	0.1
FC10(1)	71.9%	76.2%	100%	48.1%	47%	34.4%	0.14
FC100-100-10	28.9%	13.7%	21.1%	100%	6.6%	2%	0.058
FC200-200-10	38.2%	14%	23.8%	20.3%	100%	2.7%	0.065
AE400-10	23.4%	16%	24.8%	9.4%	6.6%	100%	0.086
Gaussian noise, stddev=0.1	5.0%	10.1%	18.3%	0%	0%	0.8%	0.1
Gaussian noise, stddev=0.3	15.6%	11.3%	22.7%	5%	4.3%	3.1%	0.3

- Adversarial examples crafted on a model often work against others
- AEs crafted on a model (trained with a disjoint training set) also works against the others

SRQ 4: What properties do adversarial examples exploit?

- Observations from the work by Szegedy *et al.*
  - NNs are vulnerable to adv. examples
    - False sense of security
      - They are resilient to trivial, random (Gaussian) perturbations
      - However, it does **NOT** mean NNs are resilient to the worst-case perturbations
    - The vulnerability reduces when
      - We use regularization in training
      - We use linear models
    - Adv. examples **transfer**!

RQ: How can we find adversarial examples?

- Sub research questions
  - SRQ1: What is the attack scenario (threat model)?
  - SRQ2: What are the goals for the attacker (under the threat model)?
  - SRQ3: What is the right method for finding adversarial examples?
  - SRQ4: What properties do an adversarial examples exploit?

SRQ 3: How can we find adversarial examples, efficiently?

- Results from the prior work

Model Name	Description	Training error	Test error	Av. min. distortion
FC10(10^{-4})	Softmax with $\lambda = 10^{-4}$	6.7%	7.4%	0.062
FC10(10^{-2})	Softmax with $\lambda = 10^{-2}$	10%	9.4%	0.1
FC10(1)	Softmax with $\lambda = 1$	21.2%	20%	0.14
FC100-100-10	Sigmoid network $\lambda = 10^{-5}, 10^{-5}, 10^{-6}$	0%	1.64%	0.058
FC200-200-10	Sigmoid network $\lambda = 10^{-5}, 10^{-5}, 10^{-6}$	0%	1.54%	0.065
AE400-10	Autoencoder with Softmax $\lambda = 10^{-6}$	0.57%	1.9%	0.086

- Linear vs. non-linear models
  - Observations:
    - The min. distortion required to make a model's acc. to 0%
      is larger in the non-linear models (Row 4-6) than the linear models (Row 1-3)
    - **Non-linearity** may be the primary cause of adversarial examples

SRQ 3: Finding adversarial examples on non-linear models

- Intuitions in the work by Goodfellow *et al.*[1]:
  - Finding adv. examples in non-linear models are computationally demanding
  - **(Hypothesis)** Let's only consider linearity in non-linear models!
  - **(Evaluation)** Show the existence of adversarial examples in linear models
    - Suppose an input x and its adv. input $x + \eta$, where $||\eta||_\infty < \varepsilon$, and a linear model

$$\boxed{\boldsymbol{w}^\top \tilde{\boldsymbol{x}} = \boldsymbol{w}^\top \boldsymbol{x} + \boldsymbol{w}^\top \boldsymbol{\eta}.}$$

  - **(Potential implications)**
    - Its linearity (and also the direction) matters
    - Introduce an easy way to find adversarial examples

SRQ 3: Fast Gradient Sign Method (FGSM)

- Given
  - A test-time input (x, y)
  - A NN model f and its parameters θ
  - A loss (or a cost) function $J(\theta, x, y)$
- Find
  - An adversarial perturbation η such that $f(x + \eta) \neq y$ and $||\eta||_\infty < \varepsilon$

$$\boxed{\boldsymbol{\eta} = \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right).}$$

- Results on the test-sets
  - On MNIST: 99.9% error rate with an avg. confidence of 79.3% (eps = 0.25)
  - On CIFAR10: 87.2% error rate with an avg. confidence of 96.6% (eps = 0.1)

SRQ 3: Basic Iterative Method (BIM)

- Objectives
  - To craft **powerful** AEs

- BIM Method
  - Run FGSM over multiple iterations

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\Big\{\boldsymbol{X}_N^{adv} \boxed{+} \alpha\,\mathrm{sign}\big(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{true})\big)\Big\}$$

- Iterative Least-Likely (ILL) Class Method
  - Choose a desired class as the class with the lowest logit value (y_{LL})

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_N^{adv} \boxed{-} \alpha\,\mathrm{sign}\left(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{LL})\right)\right\}$$

Topics for Today

- Motivation
  - What is it?
  - Why do we care about adversarial examples?
- Research questions
  - How can we find adversarial examples?
  - How can a real-world attacker exploit them in practice?
  - How can we remove adversarial examples?

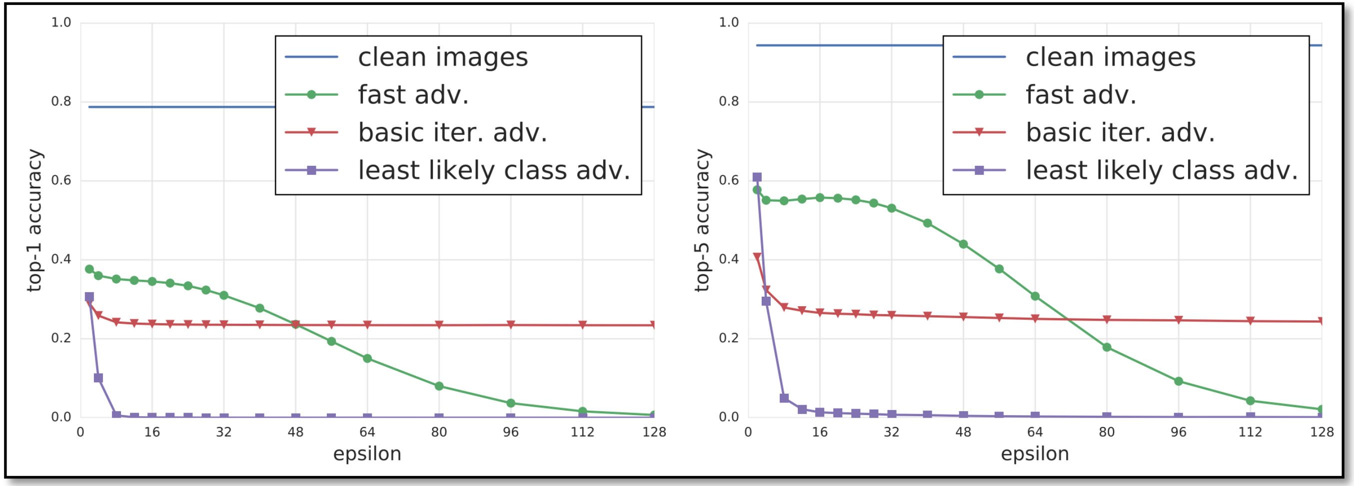
RQ 2: How does the attacker exploit AEs in practice?

- **C1:** AE in the numerical world $\neq$ AE in the physical world
  - Numerical perturbations by FGSM lead to the input values like 34.487
  - In the pixel space, such perturbations do not exist (*i.e.*, quantized pixel values)
  - One may take only classification results with a high probability (*e.g.*, > 0.8)
  - …

- Evaluation on CIFAR-10
  - Craft AEs on a DNN model (~an error rate of 99.9%)
  - Store these AEs into PNG files
  - Upload them to object recognition services (~an error rate of 10%)

RQ 2: How does the attacker exploit AEs in practice?

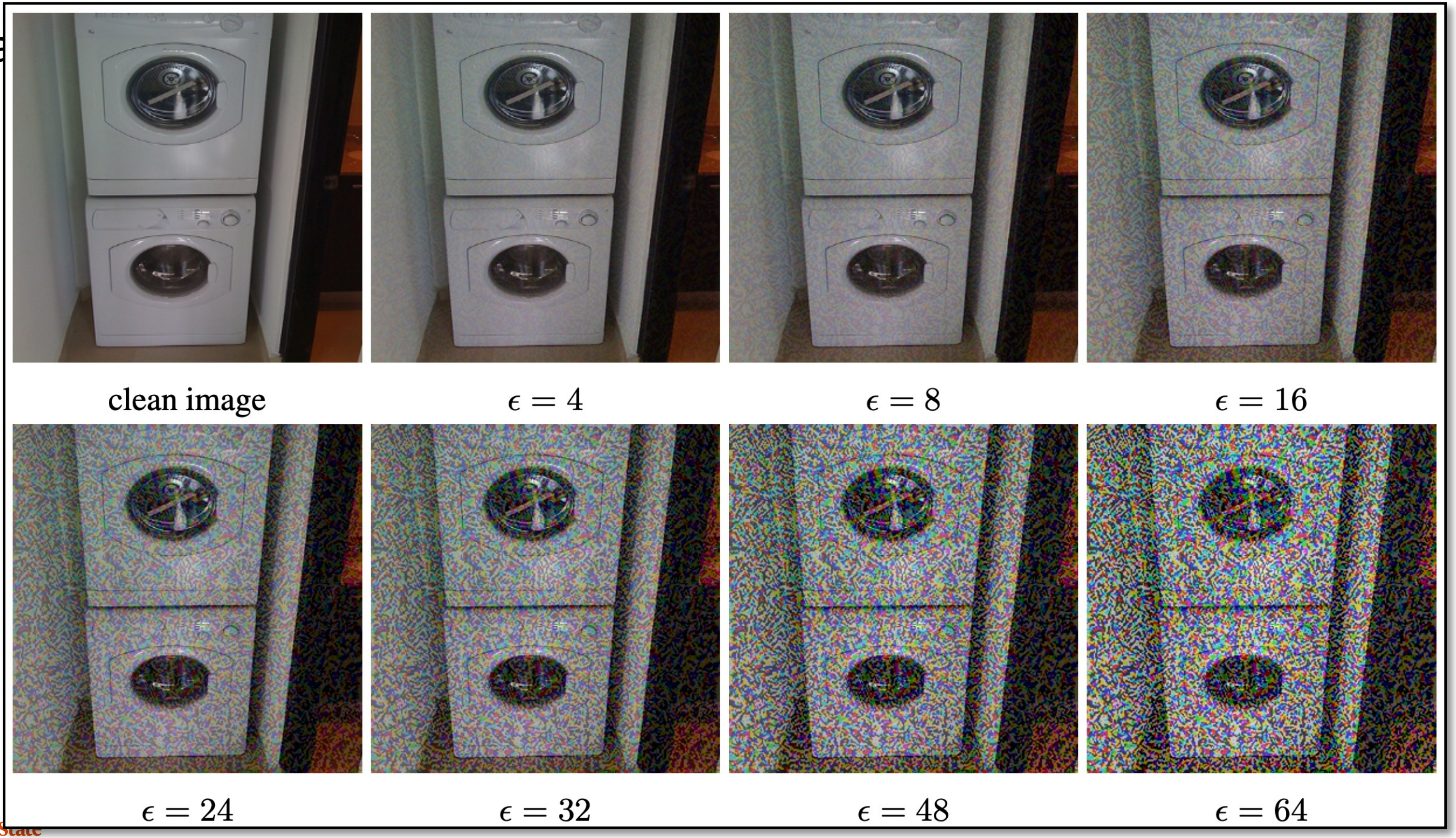
- Evaluation results of attacks on the ImageNet Inception-v3



  - In FGSM, the error rate increases as we increase epsilon
  - In the large eps, the error rate is ILL > FGSM > BIM
  - In the smaller eps, the error rate is ILL > BIM > FGSM
  - ILL achieves the highest error rate in both Top1 and Top5

RQ 2: How does the attacker exploit AEs in practice?

- Eva



RQ 2: How does the attacker exploit AEs in practice?

- Evaluation of attacks in realistic setup
  1. Craft AEs, store them in PNG, and print them
  2. Take photos of printed AEs with a cell phone
  3. Resize and center-crop the images from 2
  4. Run classification on the images from 3

- Measure
  - Classification accuracy
  - Destruction rate (error)

(a) Printout (b) Photo of printout (c) Cropped image

RQ 2: How does the attacker exploit AEs in practice?

- Observations
  - AEs work in the physical world
    - Misclassification rate is higher in AEs than what we observe with clean examples
    - Chances increase when we increase the perturbations (*i.e.*, eps from 2 to 16)
  - Prefiltering can reduce the misclassification significantly
    - **Prefilter:** only accept the classification with a high probability > 0.8
    - It reduces an error rate by 40 – 90%

RQ 2: Still, I can't believe if it works

- Link, Link, Link

Sato et al., Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack

[Let’s watch the demo together: https://www.youtube.com/watch?v=qNCXAojeEV4]

Thank You!

Tu/Th 10:00 – 11:50 am

Instructor: **Sanghyun Hong**

https://secure-ai.systems/courses/MLSec/W22

Oregon State University

SAIL
Secure AI Systems Lab