Notice

- Due dates
  - Written paper critiques (on 01.12)
  - Homework 1 (discuss an extension to 01.17)
  - Term Project (team-up by 01.19)
- Sign-up (on Canvas)
  - Scribe Lecture Note
  - In-class Paper Presentation / Discussion
- Zoom link for the class
  - Please email me if you have (to be quarantined, illness, …)

CS 499/599: Machine Learning Security

01.10: Adversarial Examples (AE) 2

Mon/Wed 12:00 – 1:50 pm

Instructor: Sanghyun Hong

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL

Secure AI Systems Lab

Recap

- ML Matters
- Evasion attack
  - Motivation
  - Threat Model
  - Gradient Descent Attack
  - Attacks on MNIST and PDF Malware Classifiers
- Counter-intuitive Properties (CI Prop.)
  - CI Prop. 1 [Controversial]
  - CI Prop. 2

Topics for Today

- AE←ML
  - Motivation / CI Prop. 1
  - CI Prop. 2 – cont'd
  - Conclusions & Implications
- AE←ML
  - Motivation
  - FGSM Attack
  - Adversarial Training
  - More observations
  - Conclusions
- AE←Security
  - Practical considerations
  - Iterative Method
  - Real-world exploitation

Szegedy et al., Intriguing Properties of Neural Networks

: This work approaches the problem from a ML perspective

Empirical Observations on MNIST Models

- H1: Overfitting Matters?

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

  - Row 1-3:
    - Regularization (weight decay) can increase the min. required distortion (for 0% acc.)
    - Excessive regularization can increase the required distortion further, but, it also increases the classifier's training and testing errors at the same time

Empirical Observations on MNIST Models

- H2: Non-linearity Matters?

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

  - Row 4-6:
    - Non-linear models are also vulnerable to adv. examples
    - The vulnerability slightly decreases as:
      1) the # of hidden units increase, and
      2) we use the auto-encoder

Empirical Observations on MNIST Models

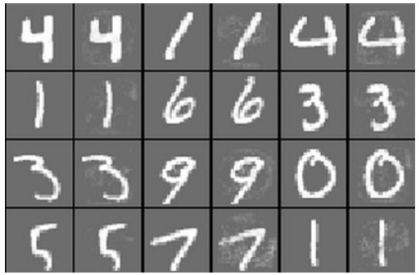
- H3: Are NNs resilient to input perturbations?

	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

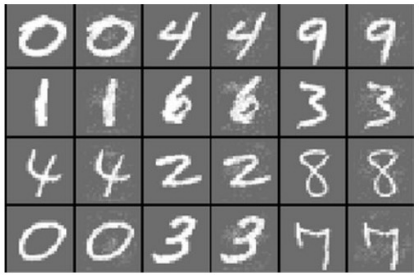
  - They are, against random Gaussian perturbations on the inputs (see the red box)
  - However, they are **NOT** against the worst-case perturbations (adv. examples)

Empirical Observations on MNIST Models

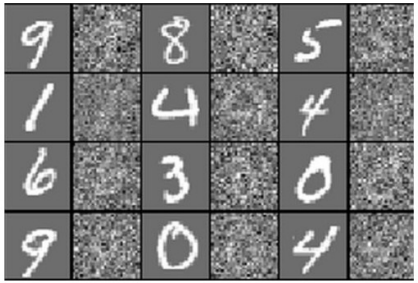
- Random perturbations (trivial ones),
  NOT the right way to measure the stability of neural networks



(a) Even columns: adversarial examples for a linear (FC) classifier (stddev=0.06)



(b) Even columns: adversarial examples for a 200-200-10 sigmoid network (stddev=0.063)



(c) Randomly distorted samples by Gaussian noise with stddev=1. Accuracy: 51%.

Empirical Observations on MNIST Models

- H4: Do 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 **transfer**!
  - The transferability varies depending on the choice of models, regularizations … used
  - (see Table 3 & 4) They transfer even btw the models trained on disjoint training sets

Spectral Analysis of Instability

- Lipschitz constant [?!]

Conclusions and Future Work

- [TL; DR] DNNs have counter intuitive properties
  - RQ1: Does a single neuron **represent** a high-level concept?
    - No distinction btw individual neurons and random linear combinations of neurons
  - RQ2: Are neural networks **resilient** to input perturbations?
    - No
      - They may have some resilience against random perturbations
      - However, it's not resilient to the worst-case test-time inputs (adversarial examples)
    - Even by adding human-imperceptible perturbations, adversarial examples are effective
    - This work suggests there maybe some ways to mitigate adv. examples
      - Reduce overfitting (e.g., using weight decay)
      - Use linear models (e.g., single layer feedforward networks)
    - This work also found that adversarial examples often **transfer**

Goodfellow et al., Explaining and Harnessing Adversarial Examples

Motivation

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

Motivation – cont'd

- Research Questions
  - RQ1: What is the **primary cause** of adversarial examples?
  - RQ2: How can we find the adversarial examples **efficiently**?
  - RQ3: How can an adversary **exploit adversarial examples in practice**?
  - RQ4: How can we **defend** models against adversarial examples?

RQ 1: Primary Cause of AEs

- Revisit H2: Non-linearity Matters?

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

  - 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

RQ 1: Primary Cause of AEs

- H in Prior work: Non-linearity Matters
- H in This work:
  - **H:** Perhaps, its linearity matters, too!
  - **Method:** 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

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

    - **(Let's show it)** We can find an AE if its input has sufficient dimensionality
  - **Implications**
    - Its linearity (and also the direction) matters
    - Perhaps, there is an easy way to find adversarial examples in NNs

RQ 1: Primary Cause of AEs

- NNs are non-linear, but are a stack of multiple linear models (link)
- Hypothesis:
  - We may exploit this property to find AEs efficiently!

RQ 2: 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 $||\eta||_\infty < \varepsilon$

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

- 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)

RQ 4: Defend ML Models against AEs

- Observation from the prior work
  - Regularizations (e.g., weight decay) reduce the error rate by AEs
  - Training a model with AEs somewhat reduces the error rate: **adversarial training (AT)**

- Challenges in AT
  - It is unclear on which adversarial examples a model should be trained
  - It is computationally expensive process with existing AE crafting methods (L-BFGS)

- **Adversarial Training** Proposed in This Work

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right).$$

RQ 4: Effectiveness of AT

- On MNIST
  - The error rate of AT models on the test-set is similar to the non-AT models
  - The AT models **become resilient** to adversarial examples
    - It reduces an error rate of a trained model from 89.4% to 17.9%
    - It also reduces the transferability: an error rate from 40.9% to 19.6%
  - But still, AT is not a perfect defense
    - If the AT models misclassify an AE, it's confidence is still high, *e.g.*, 80.4%
  - Extra observations
    - Training with random Gaussian perturbations is inefficient at preventing AEs
    - In DNNs, it is better to just perturb the original input than the activations
    - AT is useful when a model has the sufficient capacity to learn AEs

Conclusions

- Research Questions
  - RQ1: What is the primary cause of adversarial examples?
    - Empirical results may show that it's the **linearity** that matters
    - AEs are highly aligned with the **directions** of weight vectors (linear models)
    - Due to the two reasons, AEs transfer between models
  - RQ2: How can we find the adversarial examples efficiently?
    - **FGSM** (fast gradient sign method)
  - RQ4: How can we defend models against adversarial examples?
    - **Adversarial training:** we can make a model generalize on adversarial examples

Thank You!

Mon/Wed 12:00 – 1:50 pm

Instructor: Sanghyun Hong

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

Oregon State University

SAIL

Secure AI Systems Lab