CS 562. Advanced Topics in Security, Privacy and Machine Learning

Bo Li

University of Illinois at Urbana-Champaign

Today's Goal

- Adversarial attacks: FGSM, CW, PGD, BIM, Deepfool, JSMA, Physical attack
- Potential defense principles
  - Leverage intrinsic information from data/ML tasks
  - *Leverage extrinsic information from the open world, such as commonsense knowledge*
- Review format
- Discuss the potential topics for final projects

Intriguing Properties of Neural Networks

Background

- A neural network is a function with trainable parameters that learns a given mapping
- Given an image, classify it as different classes
- Give a review, classify it as good or bad
- Given a file, classify it as malware or benign

Background: accuracy

- ImageNet 2011 best result: 75% accuracy
  - No neural nets used
- ImageNet 2012 best result: 85% accuracy
  - Only top submission uses Neural nets
- ImageNet 2013 best result: 89% accuracy
  - All top submissions use Neural nets

Intriguing Findings

- No distinction between individual high level units and their random linear combinations.
- Network can misclassify an image if we apply certain specific hardly perceptible perturbations to the image.
- These distorted images or *adversarial examples* generalize fairly well even with different hyper-parameters as well as datasets.

Interpretation of Higher Units

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

(a) Unit sensitive to lower round stroke.

(b) Unit sensitive to upper round stroke, or lower straight stroke.

(c) Unit senstive to left, upper round stroke.

(d) Unit senstive to diagonal straight stroke.

Figure 1: An MNIST experiment. The figure shows images that maximize the activation of various units (maximum stimulation in the natural basis direction). Images within each row share semantic properties.

(a) Direction sensitive to upper straight stroke, or lower round stroke.

(b) Direction sensitive to lower left loop.

(c) Direction senstive to round top stroke.

(d) Direction sensitive to right, upper round stroke.

Figure 2: An MNIST experiment. The figure shows images that maximize the activations in a random direction (maximum stimulation in a random basis). Images within each row share semantic properties.

(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.

Figure 3: Experiment performed on ImageNet. Images stimulating single unit most (maximum stimulation in natural basis direction). Images within each row share many semantic properties.

(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.

Figure 4: Experiment performed on ImageNet. Images giving rise to maximum activations in a random direction (maximum stimulation in a random basis). Images within each row share many semantic properties.

Generating Adversarial Examples

Input image: $x \in \mathbb{R}^m$

Classifier: $f : \mathbb{R}^m \longrightarrow \{1 \dots k\}$

Target label: $l \in \{1 \dots k\}$

Minimize $\|r\|_2$ **subject to:**

1. $f(x+r) = l$
2. $x + r \in [0,1]^m$

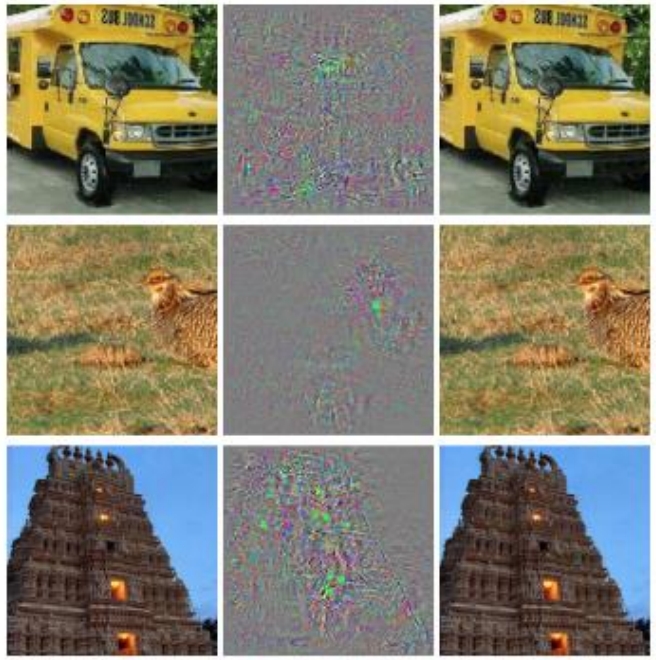
x+r is the closest image to ***x*** classified as ***l*** by ***f***.

When $f(x) \neq l$:

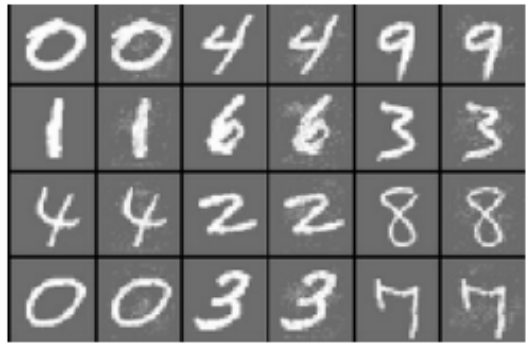
Minimize $c|r| + \text{loss}_f(x+r, l)$ **subject to** $x + r \in [0,1]^m$

Generating Adversarial Examples

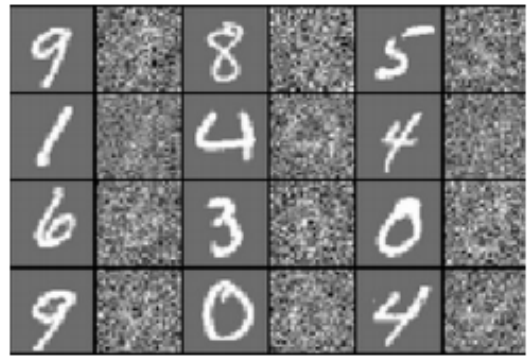
Changing an image, originally correctly classified in a way imperceptible to human eyes, can cause a net to label the image as something else entirely.



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



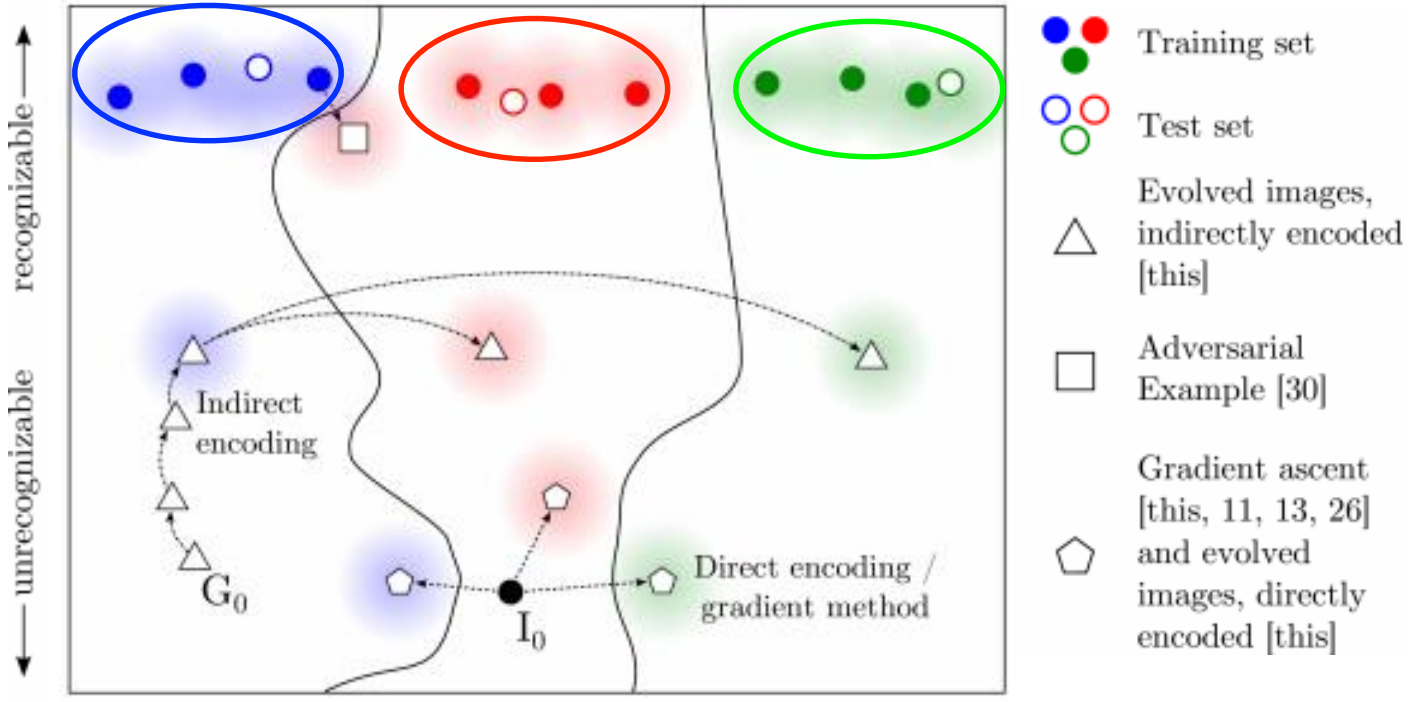
(b) Even columns: adversarial examples for a 200-200-10 sigmoid network (stddev=0.063)



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

Figure 7: Adversarial examples for a randomly chosen subset of MNIST compared with randomly distorted examples. Odd columns correspond to original images, and even columns correspond to distorted counterparts. The adversarial examples generated for the specific model have accuracy 0% for the respective model. Note that while the randomly distorted examples are hardly readable, still they are classified correctly in half of the cases, while the adversarial examples are never classified correctly.

Why adversarial examples exist?



In discriminative models, decision boundary is loose. Data points occupy much less space than what is assigned to them. Generative models would not be easily fooled.

Why adversarial examples exist?

- Adversarial examples can be explained as a property of high-dimensional dot products.
- The generalization of adversarial examples across different models can be explained as a result of adversarial perturbations being highly aligned with the weight vectors of a model and different models learning similar functions when trained to perform the same task.

Some meat

- Transferability

Definition 1. *For a classifier* $f : \mathcal{X} \to \mathcal{Y}$*, probability* $\delta \in (0, 1)$ *and random* $x \in \mathcal{X}$*, an attack strategy* $\mathcal{A}(\cdot)$ *is called* (δ, f)-effective *if* $\Pr(f(x) \neq f(\mathcal{A}(x))) \geq 1 - \delta$.

Definition 2. ***Total variation distance [1].*** *For two probability distributions* P_X *and* $P_{\mathcal{A}(X)}$ *on* $\mathcal{X}$*, the total variation distance between them is defined by*

$$||P_X - P_{\mathcal{A}(X)}||_{TV} = \max_{C \subset \mathcal{X}} |P_X(C) - P_{\mathcal{A}(X)}(C)|.$$

Definition 3. *Given* $\rho \in (0, 1)$*, an attack strategy* $\mathcal{A}(\cdot)$ *is called* ρ-covert [1]*, if for* $X \sim P_X$*,* $||P_X - P_{\mathcal{A}(X)}||_{TV} \leq \rho$.

Interesting topic: how to prove transferability?

Lemma 1. *Let $f, g : \mathcal{X} \to \mathcal{Y}$ be classifiers, $\delta, \rho, \epsilon \in (0, 1)$ be constants, and $\mathcal{A}(\cdot)$ be an attack strategy. Suppose that $\mathcal{A}(\cdot)$ is ρ-covert and f, g have risk at most ϵ. Then $\Pr(f(\mathcal{A}(x)) \neq g(\mathcal{A}(x))) \leq 2\epsilon + \rho$ for a random instance $x \sim P_X$.*

Theorem 3. *Let $f, g : \mathcal{X} \to \mathcal{Y}$ be classifiers ($\mathcal{Y} \in \{-1, 1\}$), $\delta, \rho, \epsilon \in (0, 1)$ be constants, and $\mathcal{A}(\cdot)$ an attack strategy. Suppose that $\mathcal{A}(\cdot)$ is ρ-covert and f, g have risk at most ϵ. Given random instance $x \in \mathcal{X}$, if $\mathcal{A}(\cdot)$ is (δ, g)-effective, then it is also $(\delta + 4\epsilon + \rho, f)$-effective.*

Explaining and Harnessing Adversarial Examples

Highlights

- Adversarial examples: speculative explanations
- Flaws in the linear nature of models
- Fast gradient sign method
- Adversarial training of deep networks
- Why adversarial examples generalize?
- Alternate Hypothesis

Introduction

- Szegedy et al. (2014b) : Vulnerability of machine learning models to adversarial examples
- A wide variety of models with different architectures trained on different subsets of the training data misclassify the same adversarial example – fundamental blind spots in training algorithms?
- Speculative explanations:
  - Extreme non linearity
  - Insufficient model averaging and insufficient regularization

Linear explanation of adversarial examples

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$$

$$||\boldsymbol{\eta}||_\infty < \epsilon$$

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

$$\eta = \text{sign}(\boldsymbol{w})$$

Linear explanation of adversarial examples

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$$

$$||\boldsymbol{\eta}||_\infty < \epsilon$$

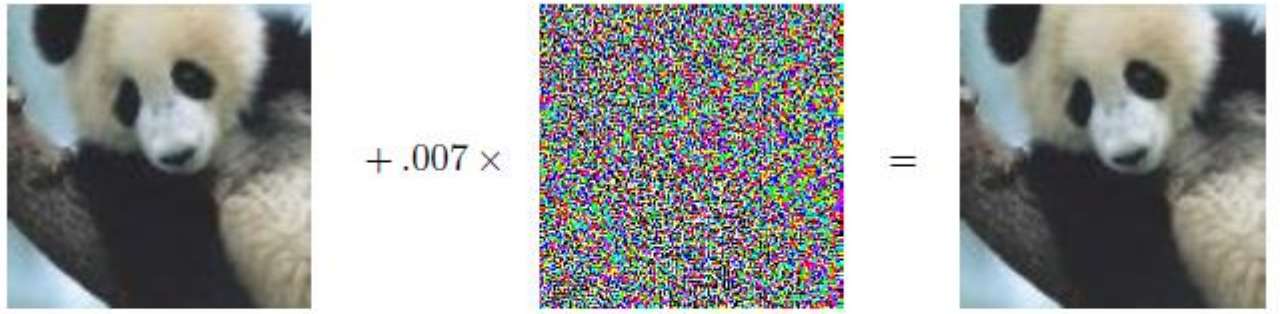
$$\boldsymbol{w}^\top \tilde{\boldsymbol{x}} = \boldsymbol{w}^\top \boldsymbol{x} + \boldsymbol{w}^\top \boldsymbol{\eta}$$

Activations grow linearly!

$$\eta = \text{sign}(\boldsymbol{w})$$

Linear perturbation of non-linear models

- ReLUs, maxout networks etc. - easier to optimize linear networks
- “Fast gradient sign method”



Fast gradient sign – logistic regression

1.6% error rate

99% error rate

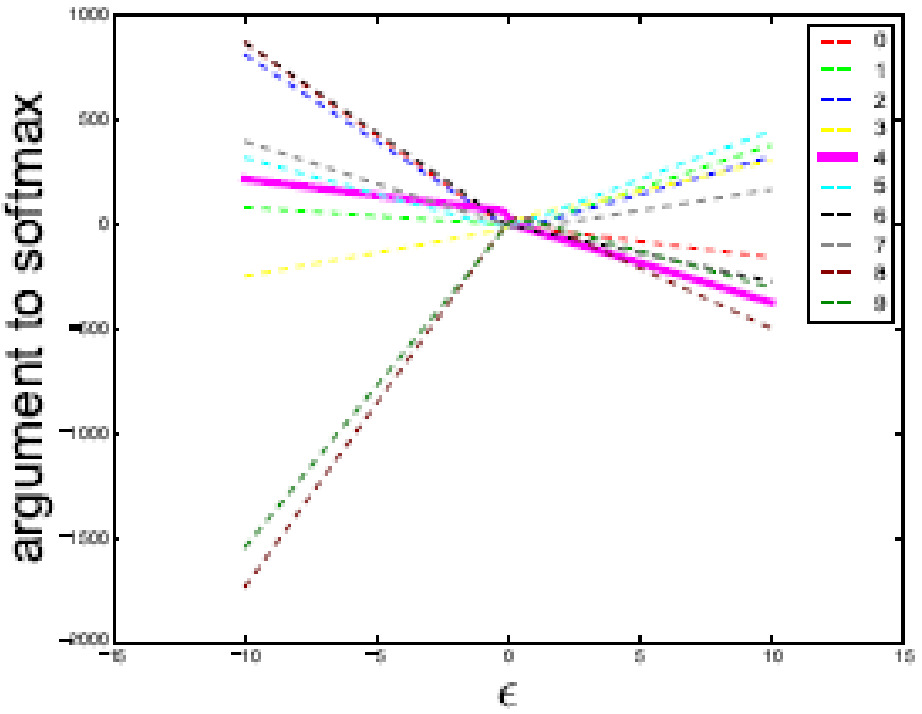
Adversarial training of deep networks

- Deep networks are vulnerable to adversarial examples - Misguided assumption

- How to overcome this?
  - Training with an adversarial objective function based on the fast gradient sign method
  - Error rate reduced from 0.94% to 0.84%

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

Why do adversarial examples generalize?



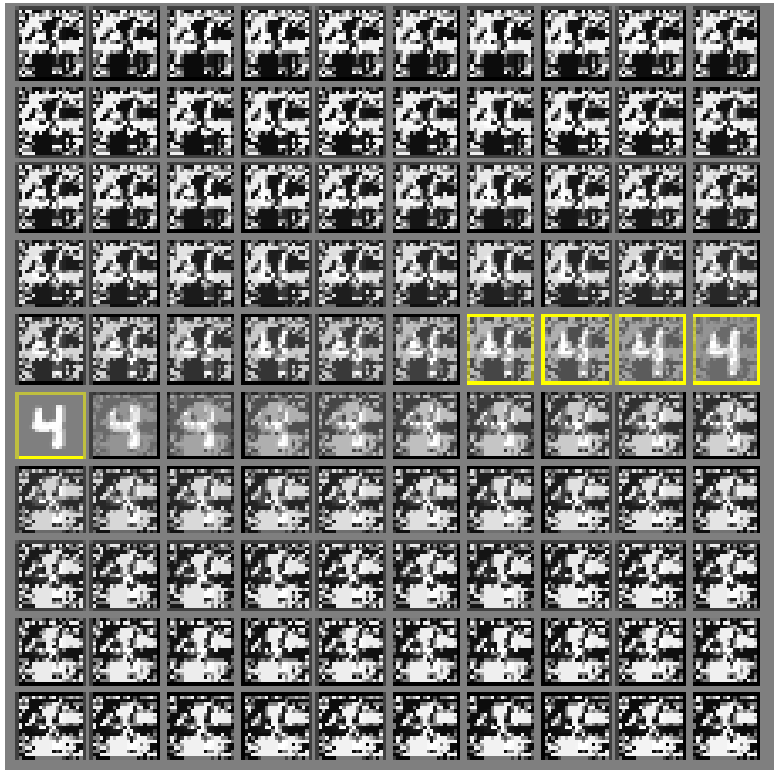


Image from reference paper

Summary

- Adversarial examples are a result of certain linearity
- Generalization of adversarial examples across different models occurs as a result of adversarial perturbations being highly aligned with the weight vector
- The direction of perturbation rather than space matters the most
- Introduces fast methods of generating adversarial examples
- Adversarial training can result in regularization
- Models easy to optimize are easy to perturb

Summary

- Linear models lack the capacity to resist adversarial perturbation; only structures with a hidden layer can
- RBF networks are more resistant to adversarial examples
- Models trained to model the input distribution are not resistant to adversarial examples.
- Ensembles are not resistant to adversarial examples

General attack strategies

- FGSM
  - Op $x' = x - \epsilon \cdot \text{sign}(\nabla \text{loss}_{F,t}(x))$
  - It is designed to be L_∞ fast instead of optimal examples
- BIM
  - Tar $x'_i = x'_{i-1} - \text{clip}_\epsilon(\alpha \cdot \text{sign}(\nabla \text{loss}_{F,t}(x'_{i-1})))$
- JSMA
  - Jacobian-based saliency map attack $\frac{\partial Z(t)}{\partial x_i}$
  - Greedily search for vulnerable pixels
- Deepfool
  - Untargeted attack optimized for L_2
  - Greedy algorithm searching against “linear” hyper plans
- CW
  - Combinatorial optimization $\min_\delta ||\delta||_p + c \cdot f(x + \delta)$

Towards Evaluating the Robustness of Neural Networks

Threat Model

- Adversary has access to model parameters
- Goal: construct adversarial examples

Two ways to evaluate robustness of DNN

- Construct a proof of robustness
- Demonstrate constructive attack
  - Break gradient descent?

Optimization based attack

$$\begin{aligned} &\min d(x, x') \\ s.t.\ &F(x') = y* \\ &x' is \text{“}valid\text{”} \end{aligned}$$

Reformulation

$$\min d(x, x') + g(x')$$
$$s.t.\ x' is \text{“}valid\text{”}$$

g(x') <= 0, if F(x') = y*
g(x') > 0, if F(x') != y*

$$g(x') = 1 - F(x')_{y*}$$

	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%

Summary

- A strong attack that can defeat defensive distillation
- Evaluation of different loss functions
- Transferability needs to be taken into account when proposing defenses

Review format

- Summary
  - Goal
  - Contributions
  - Specific technique details/analysis
- Advantages
- Disadvantages
- Potential improvement and other thoughts

Potential Final Project Topics

- Attacks against general machine learning models such as BERT and RL systems.
- Detection against attacks such as Deepfake
- Robustness against poisoning attacks
- GWAS for AI -- explainability
- Theoretically understanding of generative models from the game theoretic perspective
- Applications of GANs (GAN Zoo)
- Provable robustness for classifiers against different types of perturbation
- Differential private graphs, and robust graph neural networks
- Privacy analysis for generative models
- Improve model robustness with unlabeled data via semi-supervised learning
- Robust RL
- Robust autoML
- Semantic Forensics: sensing-reasoning models
- Design an ensemble model which guarantees the diversity of the individual classifiers and therefore improve robustness
- Red-blue team on autonomous driving platform Carla