

# Adversarial Examples for Deep Neural Networks

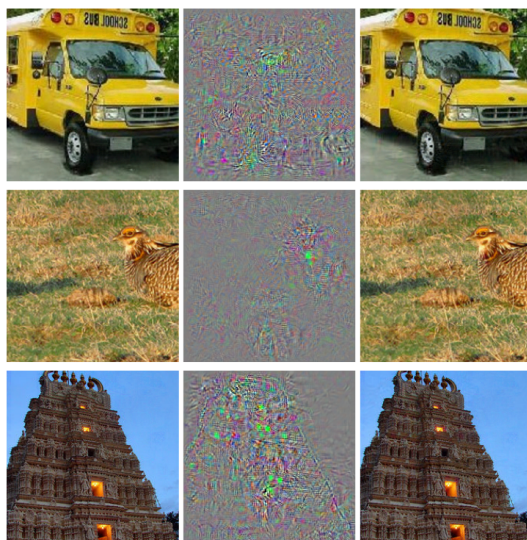
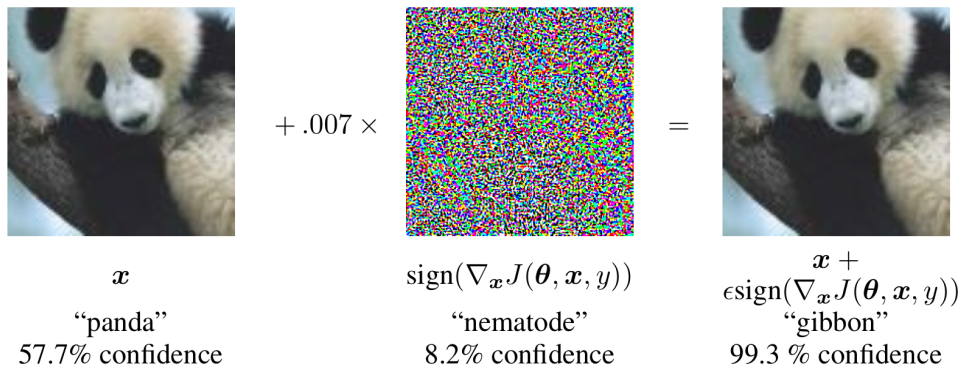
by Paul Hand  
Northeastern University

## Outline:

- Adversarial examples
- White box attacks
- Black box attacks
- Real-world attacks
- Adversarial Training

## Adversarial examples

(Goodfellow et al. 2015)



GoogleNet gets this image wrong, but a human gets it right

AlexNet classifies these as "ostrich".

(Szegedy et al. 2014)

## Formulation

There is a trained neural net classifier

$$f_{\theta}(x) = y$$

net      parameters      image      Probability distribution over classes

This net assigns to  $x$  the class  $C(x) = \underset{i}{\operatorname{argmax}} y_i$

For some image  $x$ , find perturbation  $\delta$

Such that  $C(x+\delta) \neq C(x)$       untargeted  
or  
 $C(x+\delta) = t$       targeted

Want:  $x+\delta$  to appear to a human as class  $C(x)$

## Adversarial Examples:

Targeted vs untargeted

White box vs black box vs no box

Imperceptible vs perceptible

Digital vs physical

Specific vs universal

Attack vs defense

# White Box Attacks

## Projected Gradient Descent

$$\text{Solve } X^* = \underset{X' \in P_x}{\text{argmax}} L(f(x'), y)$$

maximization      Set of images w/ small perturbations from  $x$       Cross entropy loss      true label of  $x$

$$\text{where } P_x = \{x' \mid \|x' - x\|_\infty < \epsilon\} \text{ for example}$$

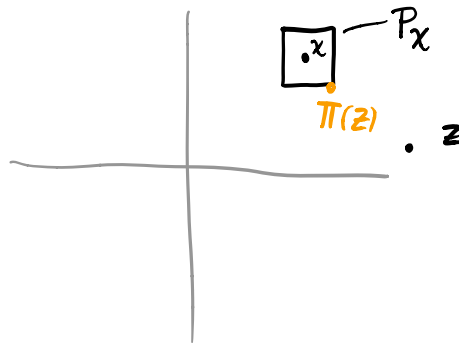
could be other p-norms      small constant

To solve:

$$x_{t+1} = \Pi(x_t + \eta_t \nabla_x L(f(x_t), y))$$

projector on to  $P_x$       step size      gives direction of maximal ascent of  $L$  wrt  $x$ , evaluated at  $x_t$

$$\text{w/ } \Pi(z) = \underset{x' \in P_x}{\text{argmin}} \|x' - z\|_2 \text{ — closest point in } P_x \text{ to } z$$



## Fast gradient sign method (FGSM) (Goodfellow et al. 2015)

$$X^* = X + \epsilon \operatorname{sgn} \nabla_x L(X, y)$$

Special case  
of PGD

roughly  
projecting on  
 $l_{\infty}$  ball

Noniterative  $\Rightarrow$  fast

## Carlini-Wagner attack (Carlini+Wagner 2017)

Want:  $\min_{\delta} \|\delta\|_p$  st.  $C(X+\delta) = t$

hard to work with  
this constraint

Suppose we have access to classifier  $f$

Notation:  $Z = f(X)$  is class logits

take  
positive  
part

Solve:  $\min_{\delta} \|\delta\|_p$  st.  $\left( \max_{i \neq t} Z(X+\delta)_i - Z(X+\delta)_t \right)^+ \leq 0$

largest logit  
other than  
class  $t$

Penalized form:  $\min_{\delta} \|\delta\|_p + \lambda \left( \max_{i \neq t} Z(X+\delta)_i - Z(X+\delta)_t \right)^+$

Commonalities of methods so far:

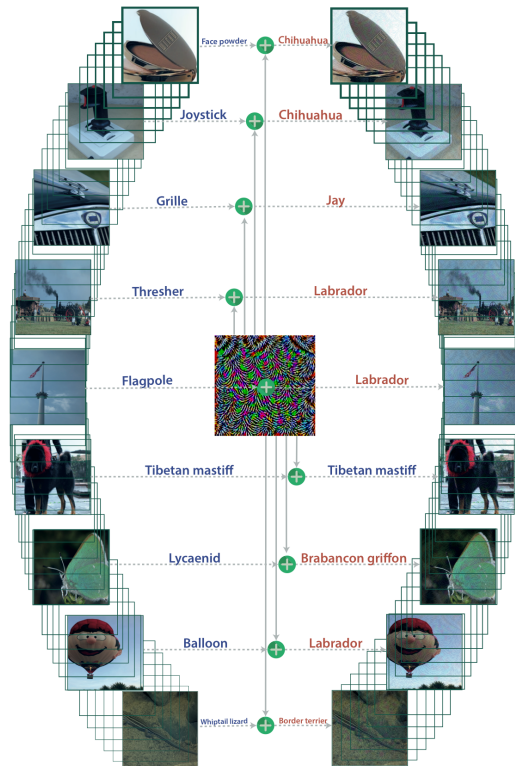
- Requires gradients of classifier (white box)
- Has variants for targeted and untargeted attacks
- Adversarial perturbation computed for a single image + classifier

# Universal Adversarial Perturbations

(Moosavi-Dezfooli et al. 2017)

Find  $\delta$  st  $X + \delta$   
is misclassified for  
most images  $X$

$\delta$  is a universal or  
image agnostic perturbation



## Algorithm 1 Computation of universal perturbations.

- 1: **input:** Data points  $X$ , classifier  $\hat{k}$ , desired  $\ell_p$  norm of the perturbation  $\xi$ , desired accuracy on perturbed samples  $\delta$ .
- 2: **output:** Universal perturbation vector  $v$ .
- 3: Initialize  $v \leftarrow 0$ .
- 4: **while**  $\text{Err}(X_v) \leq 1 - \delta$  **do**
- 5:     **for** each datapoint  $x_i \in X$  **do**
- 6:         **if**  $\hat{k}(x_i + v) = \hat{k}(x_i)$  **then**
- 7:             Compute the *minimal* perturbation that sends  $x_i + v$  to the decision boundary:  

$$\Delta v_i \leftarrow \arg \min_r \|r\|_2 \text{ s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i).$$
- 8:             Update the perturbation:  

$$v \leftarrow \mathcal{P}_{p,\xi}(v + \Delta v_i).$$
- 9:         **end if**
- 10:     **end for**
- 11: **end while**

$X_v$  - dataset perturbed  
by  $v$   
 $\{x_1 + v, x_2 + v, \dots\}$

$\text{Err}(X_v)$  - fraction of  
misclassified images  
in perturbed dataset

multiple ways  
to solve

project onto  $\ell_p$  ball  
of radius  $\xi$

## Examples of universal perturbations



(d) VGG-19

(e) GoogLeNet

(f) ResNet-152

## Universal Perturbations generalize across architectures

	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	<b>93.7%</b>	71.8%	48.4%	42.1%	42.1%	47.4%
CaffeNet	74.0%	<b>93.3%</b>	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	<b>78.9%</b>	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	<b>78.3%</b>	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	<b>77.8%</b>	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	<b>84.0%</b>

Table 2: Generalizability of the universal perturbations across different networks. The percentages indicate the fooling rates. The rows indicate the architecture for which the universal perturbations is computed, and the columns indicate the architecture for which the fooling rate is reported.

You can attack an unknown classifier by training your own (with a different architecture) and running a white box method

## Black box attacks

Cant backprop/differentiate the classifier you are attacking

You may have access to logit output or perhaps only to predictions or nothing

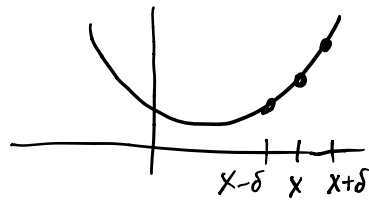
Approaches:

Zeroth order Optimization (ZOO)  
Transferability Attacks

ZOO in general:

To compute  $\min_{X \in \mathbb{R}^n} g(X)$  w/o derivatives:

Id case:



with the values of  $g$  at these 3 points, can estimate  $g'(x)$  &  $g''(x)$ .

$$g'(x) \approx \frac{g(x+\delta) - g(x-\delta)}{2\delta}$$

$$g''(x) \approx \frac{g(x+\delta) - 2g(x) + g(x-\delta)}{\delta^2}$$

Model as a parabola and find its minimum.

Higher dim case: Stochastic Coordinate Descent

Choose a random coordinate  $e_i$   
Compute  $\min_s g(x + s e_i)$  as in 1d

ZOO for adversarial examples: (w/ logits)

USE Stochastic coordinate descent on  
CW formulation (Chen et al. 2017)

ZOO for adversarial examples: (w/ only class labels)

Randomized gradient free (RGF) method  
(Cheng et al. 2018)

In adversarial examples, accurate gradients  
are not needed. (eg FGSM)



# Transferability Attacks (Liu et al. 2017)

## Ensemble approach

Train multiple classifiers w/ different architectures

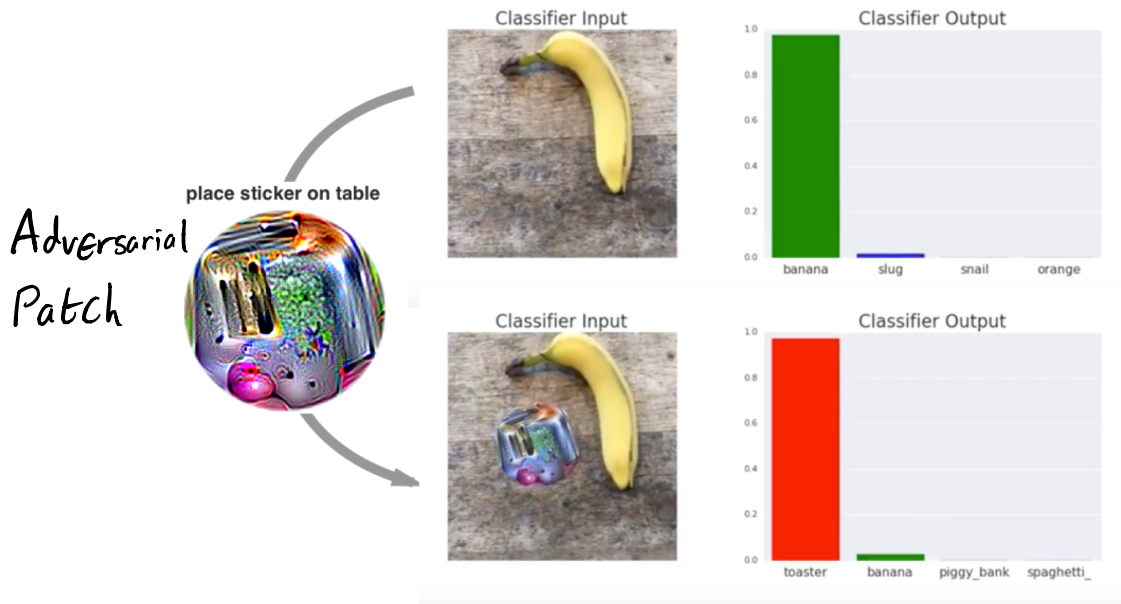
Try to find average (logit) output over the ensemble

	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
-ResNet-152	17.17	0%	0%	0%	0%	0%
-ResNet-101	17.25	0%	1%	0%	0%	0%
-ResNet-50	17.25	0%	0%	2%	0%	0%
-VGG-16	17.80	0%	0%	0%	6%	0%
-GoogLeNet	17.41	0%	0%	0%	0%	5%

Table 4: Accuracy of non-targeted adversarial images generated using the optimization-based approach. The first column indicates the average RMSD of the generated adversarial images. Cell  $(i, j)$  corresponds to the accuracy of the attack generated using four models except model  $i$  (row) when evaluated over model  $j$  (column). In each row, the minus sign “-” indicates that the model of the row is not used when generating the attacks. Results of top-5 accuracy can be found in the appendix (Table 14).

# Real World Attacks

(Brown et al. 2018)



Patch is so visually salient, a classifier ignores the rest of the image

Challenges:

Can't change all pixels

Needs to work for all backgrounds

Must be robust to physical transformation

Build a model for transformations:

$$A(\underbrace{\delta}_{\delta}, \underbrace{x}_{x}, \underbrace{\ell, T}_{\ell, T}, \text{location, rotation, scale, ...}) =$$

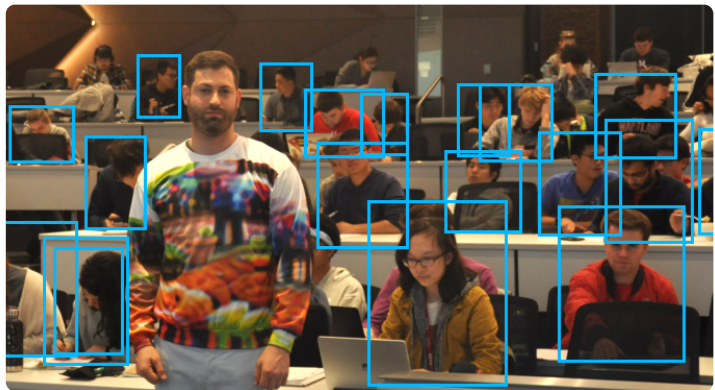


Find adversarial patch  $\delta$  by choosing target class  $t$ :

$$\operatorname{argmax}_{\delta} \mathbb{E}_{x, \ell, T} \log P(\text{img } A(\delta, x, \ell, T) \text{ is class } t)$$

Can make it further robust by using ensembles

Other examples:



(Wu et al. 2019)



(Sharif et al. 2016)



(Eykholt et al. 2018)

# Adversarial Training

(Goodfellow et al. 2015)

Train a classifier and try to ensure adversarial perturbations get correctly classified

Example for FGSM:

Instead of optimizing

$L(\theta, x, y)$  at train time,

optimize

$$\alpha L(\theta, x, y) + (1 - \alpha) L(\theta, x + \epsilon \operatorname{sgn} \nabla_x L(\theta, x, y), y)$$

weighted combination

FGSM adversarial example

Challenge:

won't be robust to other methods

Game of cat and mouse