Physical Attacks on Deep Learning Systems

Ivan Evtimov

Collaborators and slide content contributions:
Earlence Fernandes, Kevin Eykholt, Chaowei Xiao, Amir Rahmati, Florian Tramer, Bo Li, Atul Prakash, Tadayoshi Kohno, Dawn Song

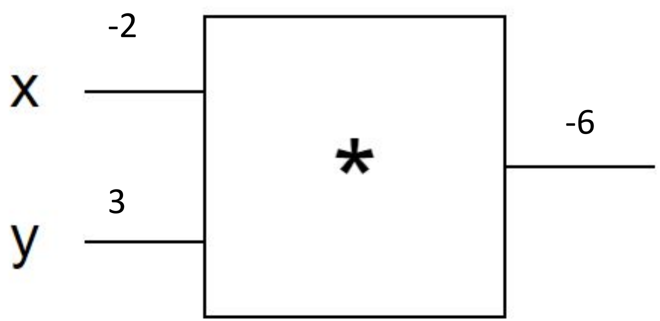
UNIVERSITY of WASHINGTON

Deep Learning Mini Crash Course

Neural Networks Background

Convolutional Neural Networks (CNNs)

Real-Valued Circuits



$$f(x, y) = xy$$

Goal: How do I increase the output of the circuit?

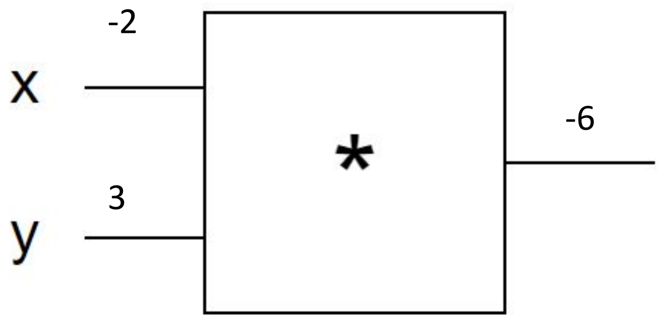
- Option 2. Analytic Gradient

$$\frac{\partial f(x,y)}{\partial x} = \frac{f(x+h,y) - f(x,y)}{h}$$

Limit as h -> 0

x = x + step_size * x_gradient
y = y + step_size * y_gradient

Real-Valued Circuits



$$f(x,y) = xy$$

Goal: How do I increase the output of the circuit?

- Tweak the inputs. But how?

- Option 1. Random Search?

x = x + step_size * random_value
y = y + step_size * random_value

Gradients and Gradient Descent

- Each component of the gradient tells you how quickly the function is changing (increasing) in the corresponding direction.
- The gradient vector together points in the direction of the steepest ascent.
- To minimize a function, move in the opposite direction.
- Easy update rule for minimizing a variable v controlling a function f:

  *v = v - step*gradient(f)*

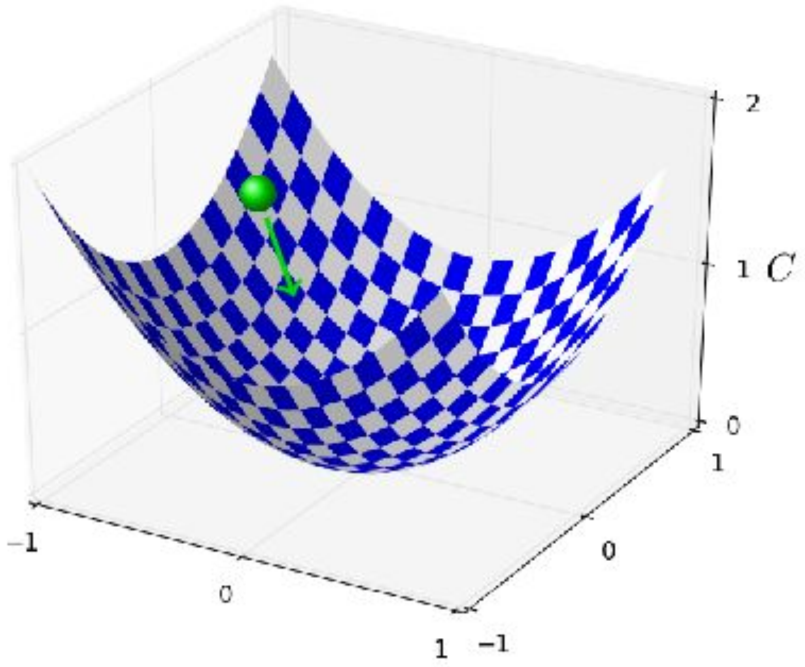
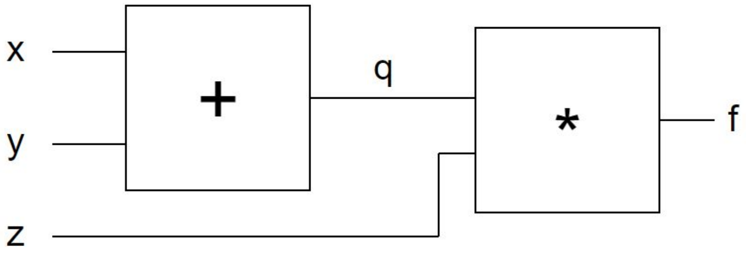


Image Credit:
http://neuralnetworksanddeeplearning.com/chap3.html

Composable Real-Valued Circuits



$$f(x, y, z) = (x + y)z.$$

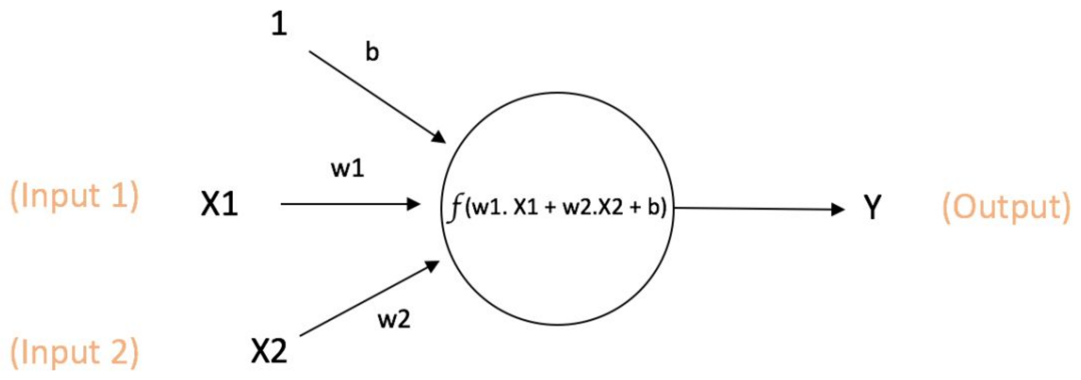
$$f(q, z) = qz \quad \Longrightarrow \quad \frac{\partial f(q, z)}{\partial q} = z, \quad \frac{\partial f(q, z)}{\partial z} = q$$

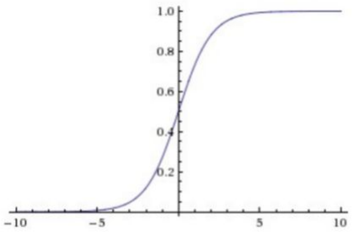
$$q(x, y) = x + y \quad \Longrightarrow \quad \frac{\partial q(x, y)}{\partial x} = 1, \quad \frac{\partial q(x, y)}{\partial y} = 1$$

Chain Rule $\quad \dfrac{\partial f(q, z)}{\partial x} = \dfrac{\partial q(x, y)}{\partial x} \dfrac{\partial f(q, z)}{\partial q}$

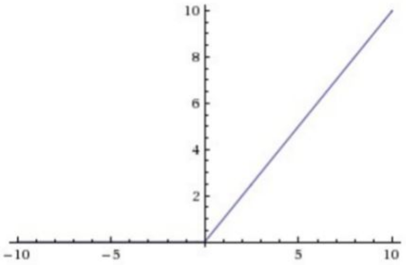
chain rule + some dynamic programming = backpropagation

Single Neuron

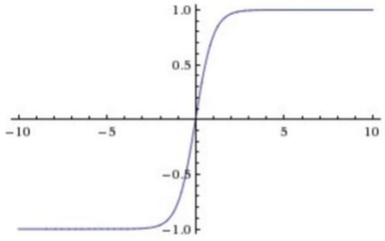




Sigmoid



ReLU

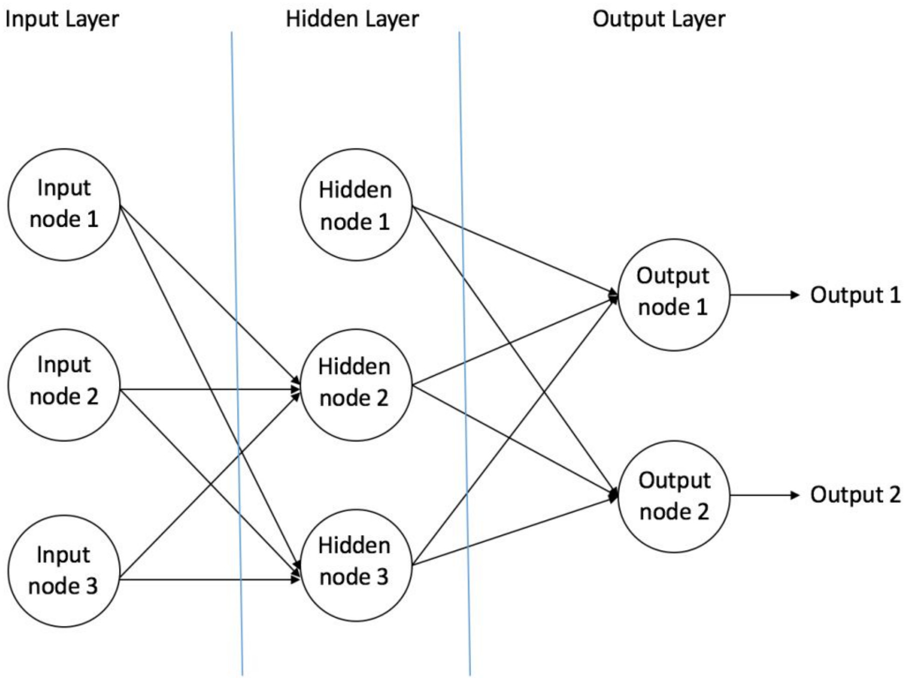


tanh

Output of neuron = Y= f(w1. X1 + w2.X2 + b)

Activation function

(Deep) Neural Networks!

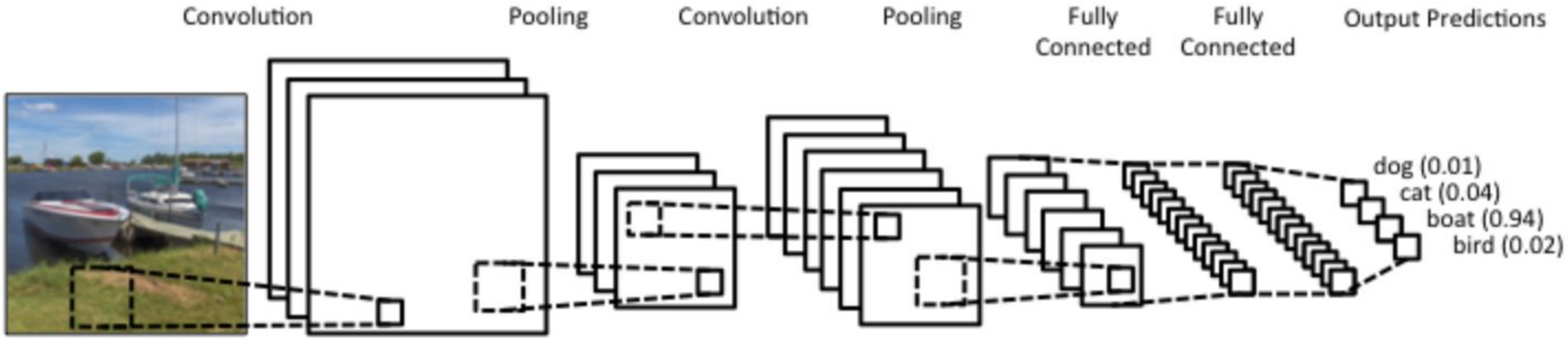


Organize neurons into a structure

Train (optimize) using backpropagation

Loss function: how far is the output of the network from the true label for the input?

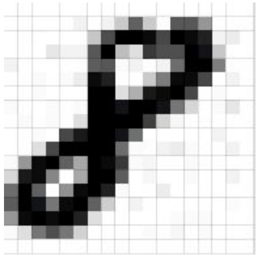
Convolutional Neural Networks (CNNs)



A CNN generally consists of 4 types of architectural units

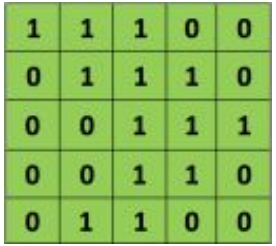
Convolution
Non Linearity (RELU)
Pooling or Subsampling
Classification (Fully Connected Layers)

How is an image represented for NNs?

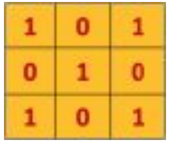


- Matrix of numbers, where each number represents pixel intensity
- If image is colored, then there are three channels per pixel, each channel representing (R, G, B) values

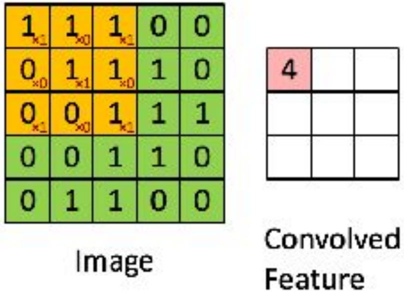
Convolution Operator



Grayscale Image



Kernel or Filter or
Feature Detector

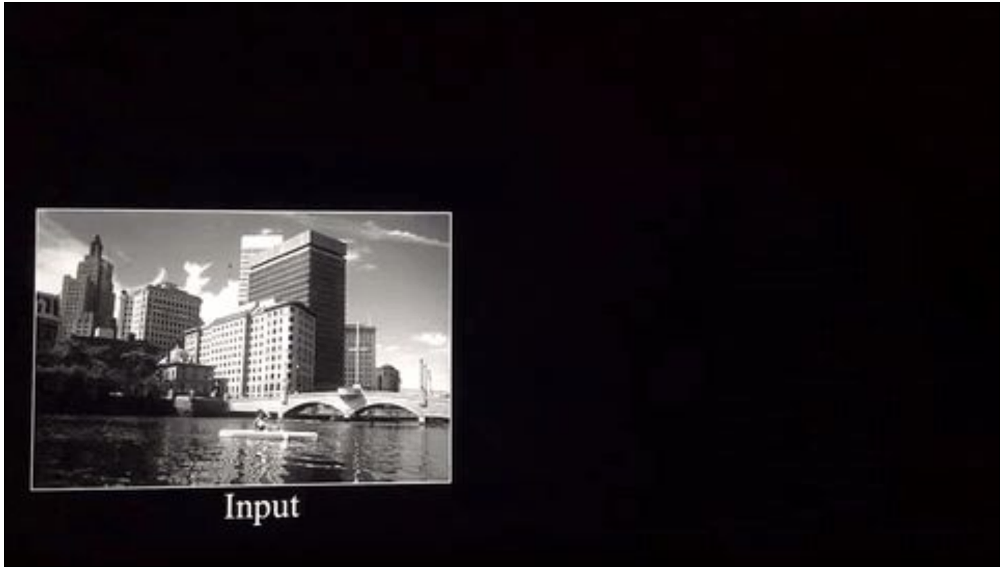


Feature map!

- Slide the kernel over the input matrix
- Compute element wise multiplication (Hadamard/schur product), add results to get a single value
- Output is a feature map

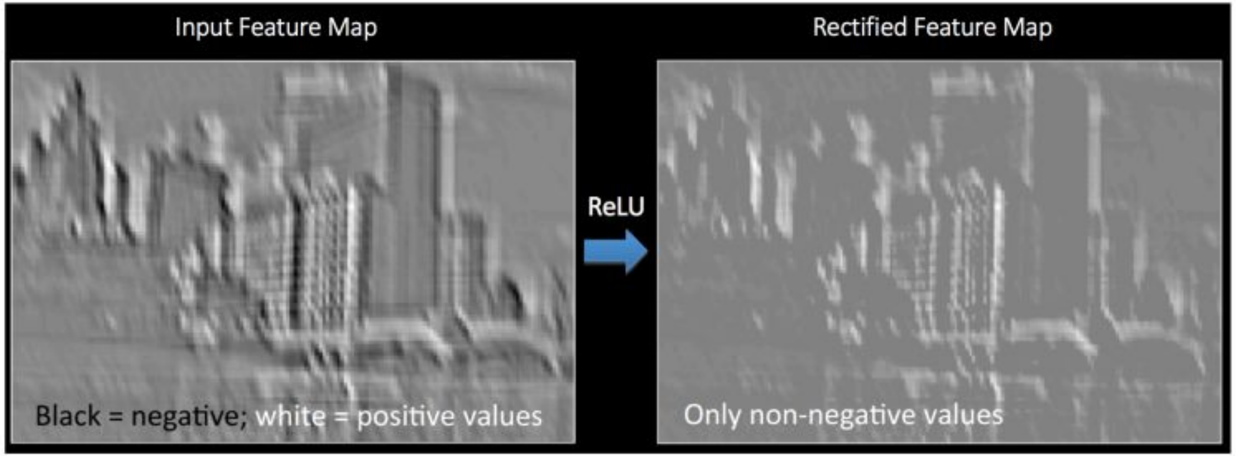
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Many types of filters

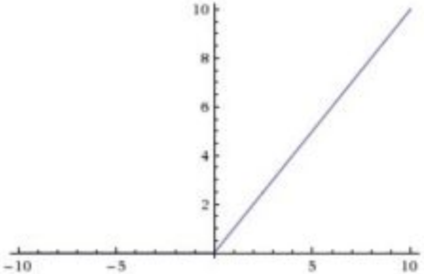


A CNN learns these filters during training

Rectified Linear Unit (Non-Linearity)

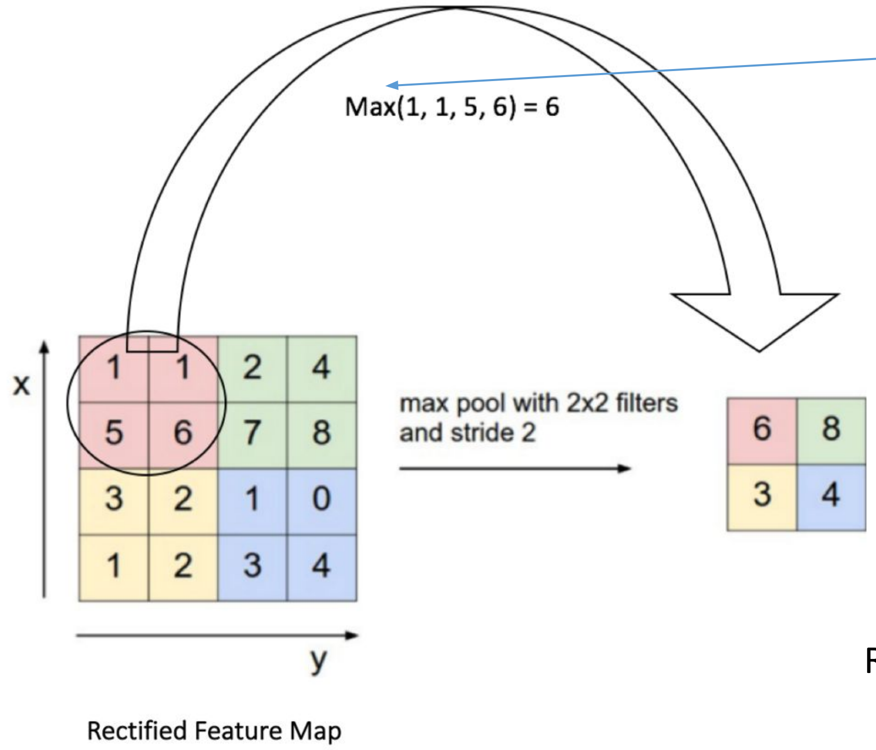


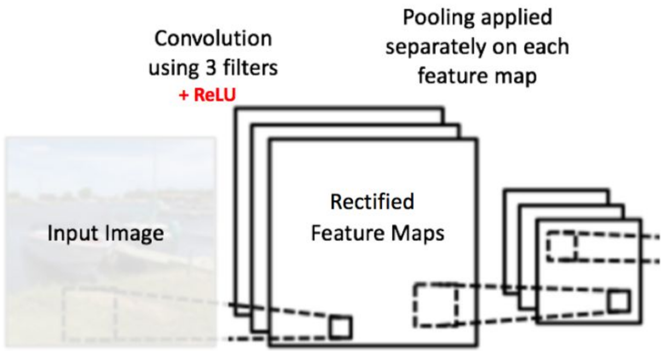
Output = Max(zero, Input)



Pooling

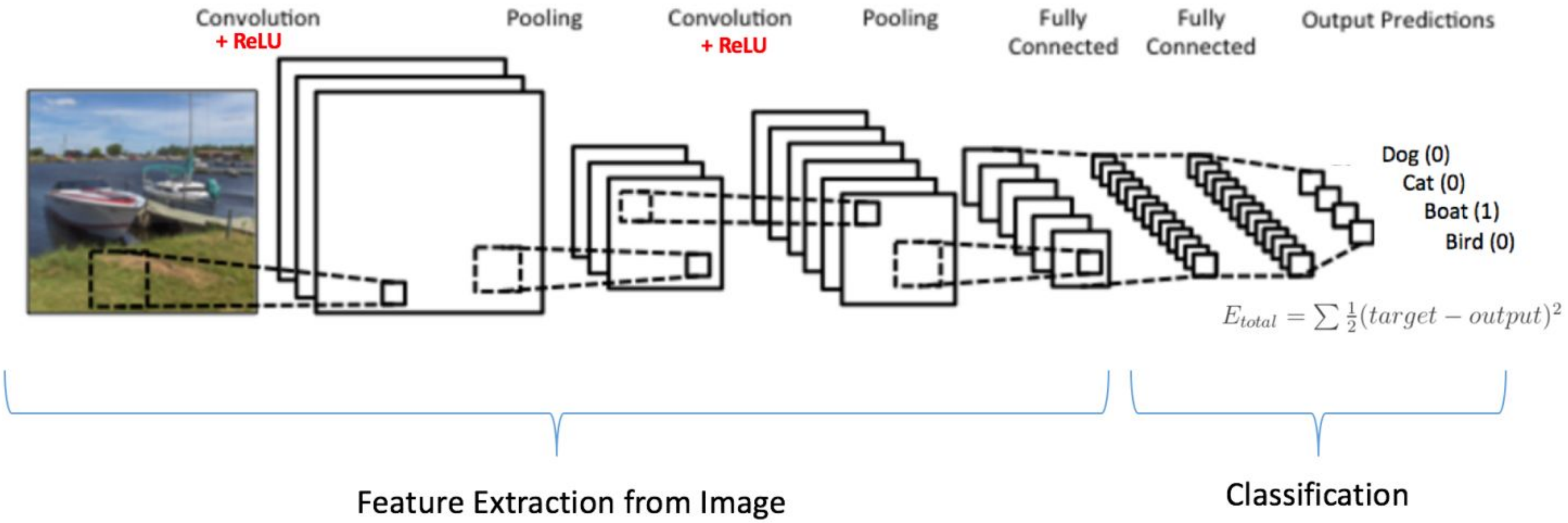
Can be Avg, sum, min, …





Reduce dimensionality, but retain important features

Putting Everything Together

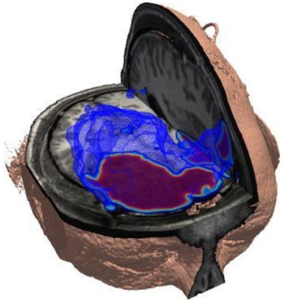


Deep Neural Networks are Useful

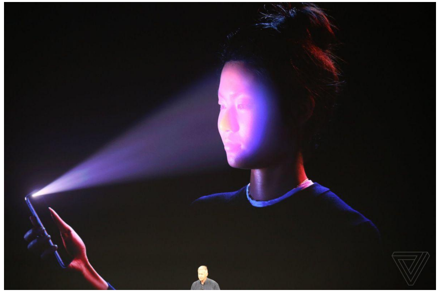
AlphaGo

Playing sophisticated games

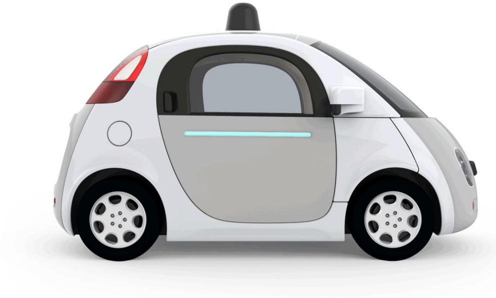
Understanding natural language



Processing medical images



Face recognition



Controlling cyber-physical systems?

Deep Neural Networks Can Fail

If you use a loss function that fulfills an adversary's goal, you can follow the gradient to find an image that misleads the neural network.

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

=

+ ε

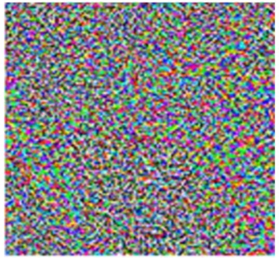


Image Courtesy: OpenAI

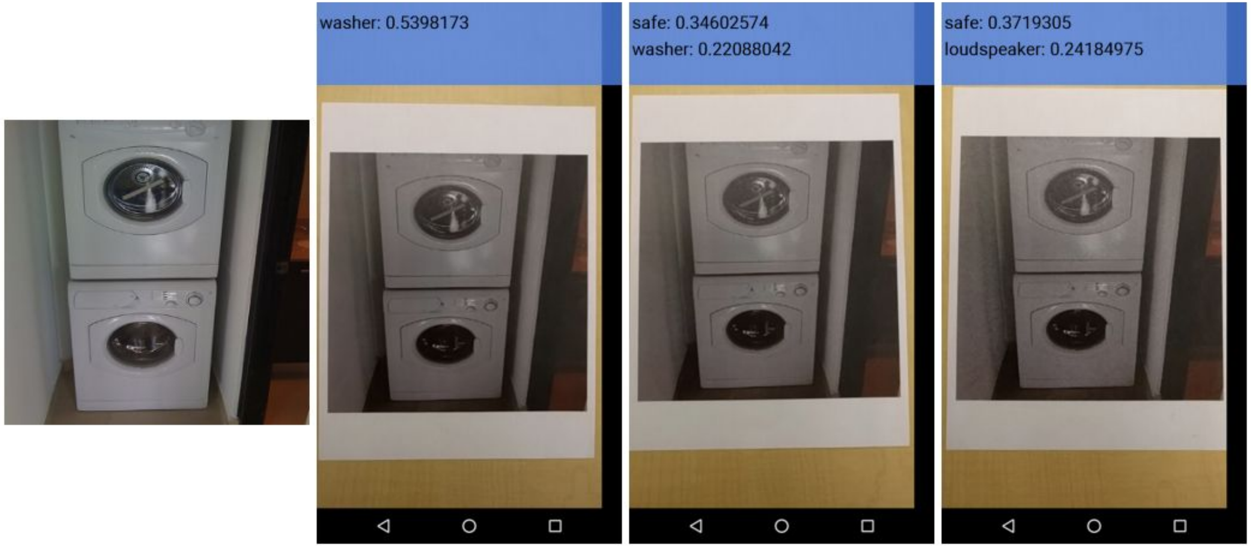
"gibbon"
99.3% confidence

"panda"
57.7% confidence

Explaining and Harnessing Adversarial Examples, Goodfellow et al., arXiv 1412.6572, 2015

Deep Neural Networks Can Fail...

...if adversarial images are printed out

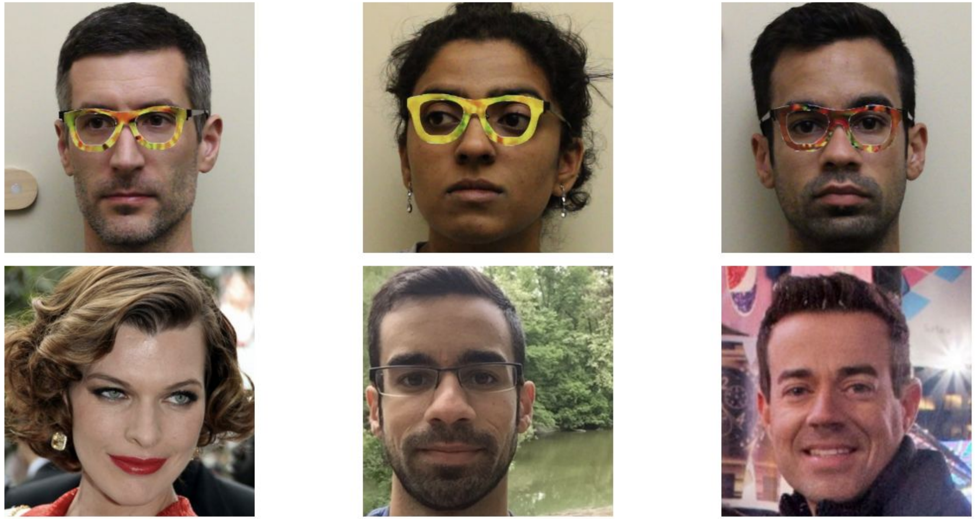


Kurakin et al. "Adversarial examples in the physical world." arXiv preprint arXiv:1607.02533 (2016).

Deep Neural Networks Can Fail...

...if an adversarially crafted physical object is introduced

This person wearing an “adversarial” glasses frame...



...is classified as this person by a state-of-the-art face recognition neural network.

Sharif et al. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016.

Deep neural network classifiers are vulnerable to adversarial examples in some physical world scenarios

However: In real-world applications, conditions vary more than in the lab.

Take autonomous driving as an example...

A road sign can be far away

or it could be at an angle

Can physical adversarial examples cause misclassification at large angles and distances?

An Optimization Approach To Creating Robust Physical Adversarial Examples

$$\underset{\delta}{\operatorname{argmin}}\ \lambda||\delta||_p + J(f_\theta(x+\delta), y^*)$$

Perturbation/Noise Matrix

Adversarial Target Label

Lp norm (L-0, L-1, L-2, …)

Loss Function

An Optimization Approach To Creating Robust Physical Adversarial Examples

$$\underset{\delta}{\operatorname{argmin}} \ \lambda||\delta||_p + J(f_\theta(x+\delta), y^*)$$

Perturbation/Noise Matrix

Adversarial Target Label

Lp norm (L-0, L-1, L-2, …)

Loss Function

Challenge: This formulation only generates perturbations valid for a single viewpoint. How can we make the perturbations viewpoint-invariant?

An Optimization Approach To Creating Robust Physical Adversarial Examples

$$\underset{\delta}{\operatorname{argmin}} \; \lambda ||\delta||_p + J(f_\theta(x+\delta), y^*)$$

Perturbation/Noise Matrix

Lp norm (L-0, L-1, L-2, …)

Loss Function

Adversarial Target Label

$$\underset{\delta}{\operatorname{argmin}} \; \lambda ||\delta||_p + \frac{1}{k}\sum_{i=1}^{k} J(f_\theta(x_i+\delta), y^*)$$

What about physical realizability?

Observation: Signs are often messy...

What about physical realizability?

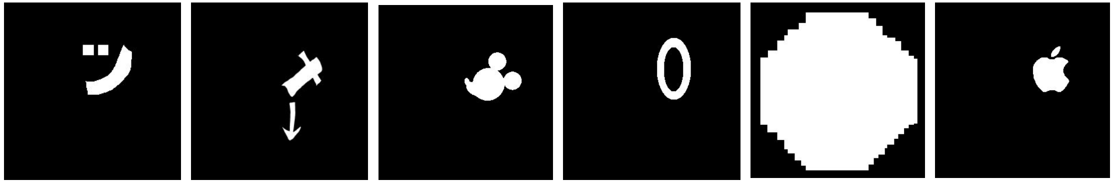
So: make the perturbation appear as vandalism

Camouflage
Sticker

Subtle
Poster

Optimizing Spatial Constraints

$$\underset{\delta}{\operatorname{argmin}}\ \lambda||M_x\cdot\delta||_p+\frac{1}{k}\sum_{i=1}^{k}J(f_\theta(x_i+M_x\cdot\delta),y^*)$$



Subtle Poster

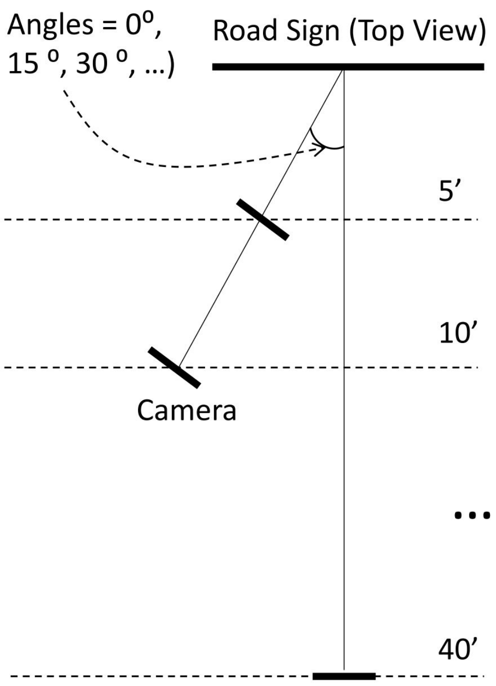
Camouflage Sticker

Mimic vandalism

“Hide in the human psyche”

How Can We Realistically Evaluate Attacks?

Lab Test (Stationary)



Field Test (Drive-By)

~ 250 feet, 0 to 20 mph

Record video

Sample frames every k frames

Run sampled frames through DNN

Subtle Poster
100%

Subtle Poster
73.33%

Camo Graffiti
66.67%

Camo Art
100%

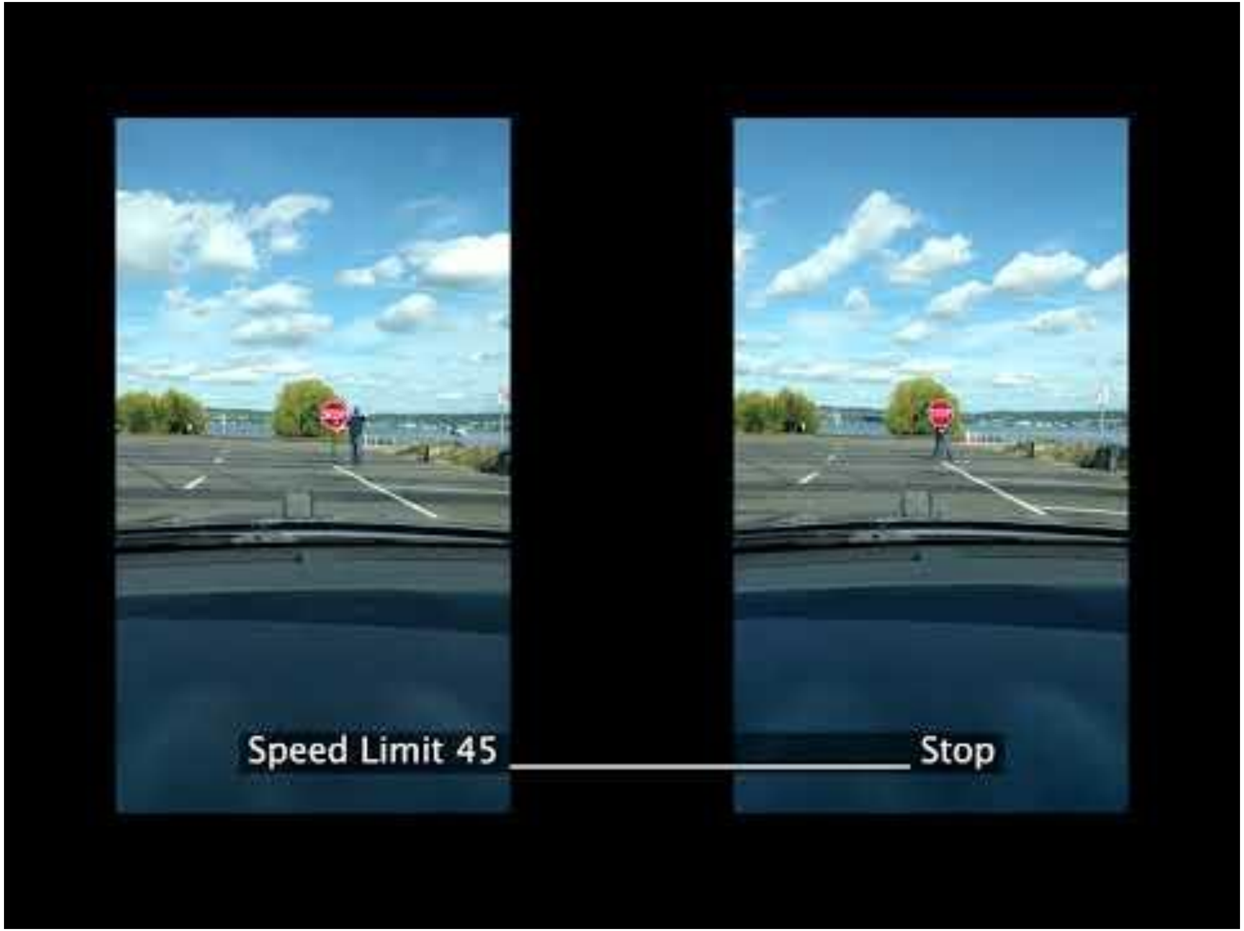
Camo Art
80%

Lab Test Summary (Stationary)

Target Classes:
Stop -> Speed Limit 45
Right Turn -> Stop

Numbers at the bottom of the images
are success rates

Video: camo graffiti
https://youtu.be/1mJMPqi2bSQ
Video: subtle poster
https://youtu.be/xwKpX-5Q98o



Field Test (Drive-by)

Target Classes:
Stop -> Speed Limit 45
Right Turn -> Stop

Classification top class is indicated at the bottom of the images.
Left: “Adversarial” stop sign
Right: Clean stop sign

Attacks on Inception-v3

Coffee Mug -> Cash Machine, 81% success rate

Open Questions and Future Work

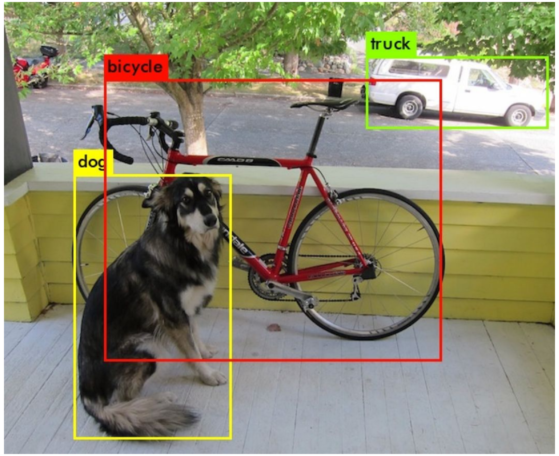
- Have we successfully hidden the perturbations from casual observers?
- Are systems deployed in practice truly vulnerable?
- How can we defend against these threats?

Classification

What’s the dominant object
in this image?

Object Detection

What are the objects in this scene, and where are they?



Semantic Segmentation

What are the precise shapes and locations of objects?

We know that physical adversarial examples exist for classifiers

Do they exist for richer classes of vision algorithms?

Challenges in Attacking Detectors

Detectors process entire scene, allowing them to use contextual information

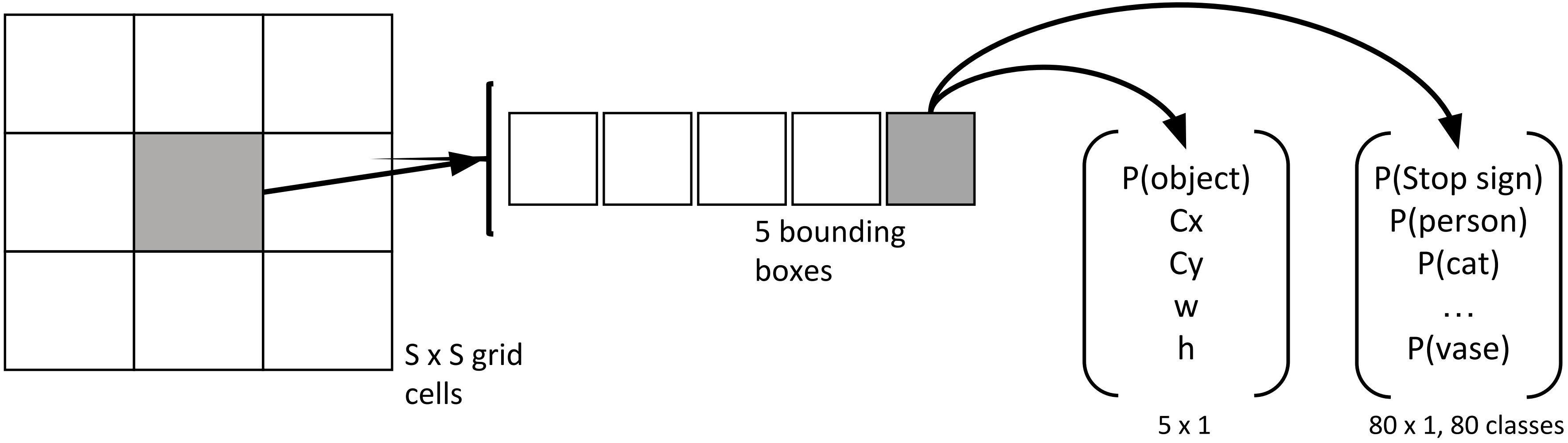
Not limited to producing a single labeling, instead labels all objects in the scene

The location of the target object within the scene can vary widely

Translational Invariance

$$\underset{\delta}{\operatorname{argmin}}\ \lambda||M_x \cdot \delta||_p + \mathbb{E}_{x_i \sim X^V} J(f_\theta(x_i + T_i(M_x \cdot \delta)), y^*)$$

Designing the Adversarial Loss Function



$$J_d(x,y) = \max_{s \in S^2, b \in B} P(s,b,y,f_\theta(x))$$

Prob. of object being class 'y'

Output of YOLO, 19 x 19 x 425 tensor

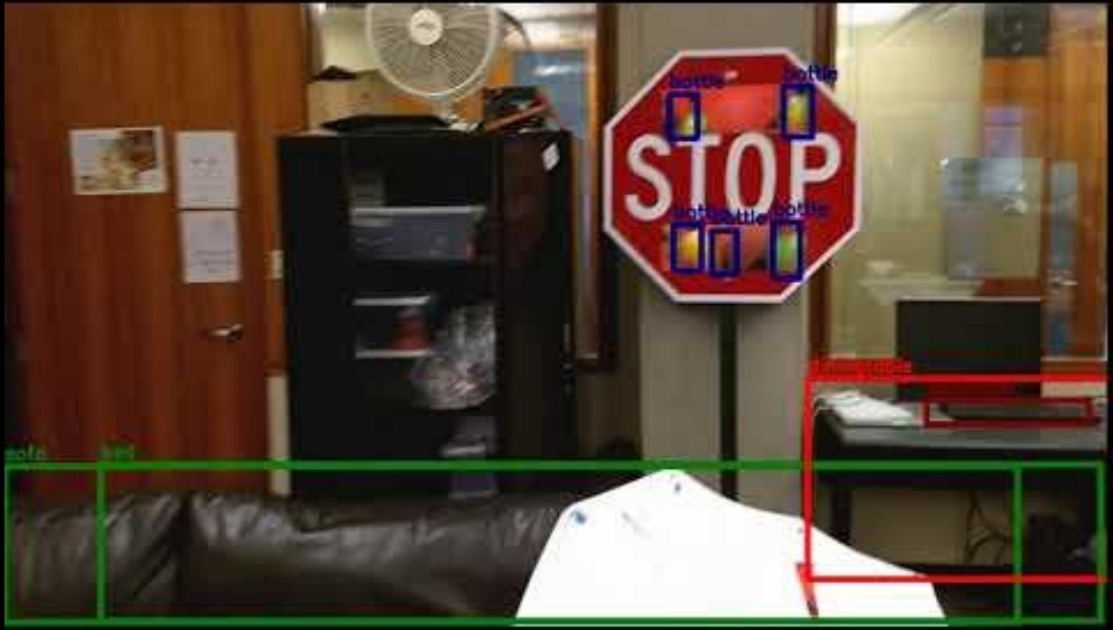
Input scene

Minimize the probability of "Stop" sign among all predictions

Poster and Sticker Attack

Poster Attack on YOLO v2

Sticker Attack on YOLO v2



Robust Physical-World Attacks on Deep Learning Models

Project website: https://iotsecurity.eecs.umich.edu/#roadsigns

Collaborators: Earlence Fernandes, Kevin Eykholt, Chaowei Xiao, Amir Rahmati, Florian Tramer, Bo Li, Atul Prakash, Tadayoshi Kohno, Dawn Song

Structure of Classifiers

Layer Type	Number of Channels	Filter Size	Stride	Activation
conv	3	1x1	1	ReLU
conv	32	5x5	1	ReLU
conv	32	5x5	1	ReLU
maxpool	32	2x2	2	-
conv	64	5x5	1	ReLU
conv	64	5x5	1	ReLU
maxpool	64	2x2	2	-
conv	128	5x5	1	ReLU
conv	128	5x5	1	ReLU
maxpool	128	2x2	2	-
FC	1024	-	-	ReLU
FC	1024	-	-	ReLU
FC	43	-	-	Softmax

GTSRB*-CNN

Accuracy: 95%

43 classes of German road signs* from the GTSRB classification dataset.

**The stop sign images were replaced with U.S. stop sign images both in training and in evaluation.*

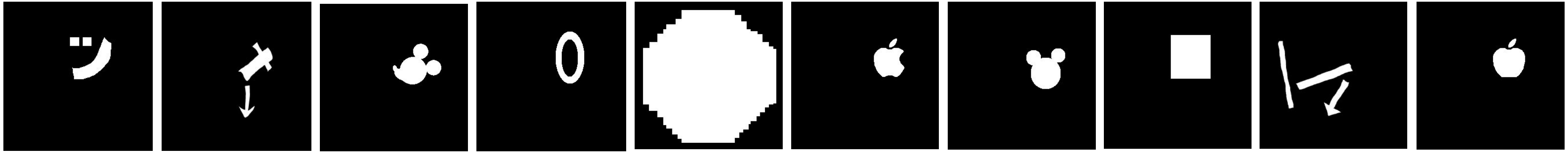
Layer Type	Number of Channels	Filter Size	Stride	Activation
conv	64	8x8	2	ReLU
conv	128	6x6	2	ReLU
conv	128	5x5	1	ReLU
FC	17	-	-	Softmax

LISA-CNN

Accuracy: 91%

17 classes of U.S. road signs from the LISA classification dataset

How Might One Choose A Mask?

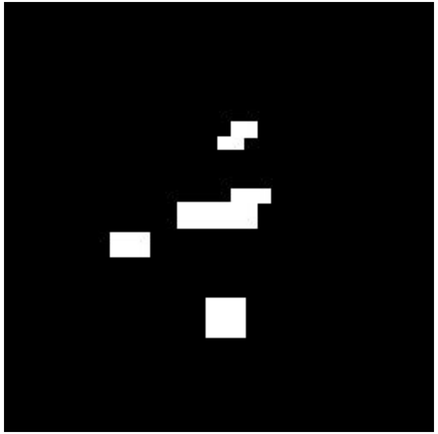


We had very good success with the octagonal mask

Hypothesis: Mask surface area should be large or should be focused on "sensitive" regions

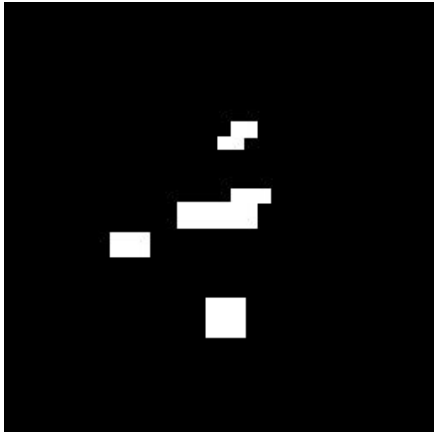
$$\underset{\delta}{\operatorname{argmin}}\ \lambda||M_x \cdot \delta||_p + \frac{1}{k}\sum_{i=1}^{k} J(f_\theta(x_i + M_x \cdot \delta), y^*)$$

Use L-1



Process of Creating a Useful Sticker Attack

L-1 Perturbation



Result Mask

Sticker Attack!

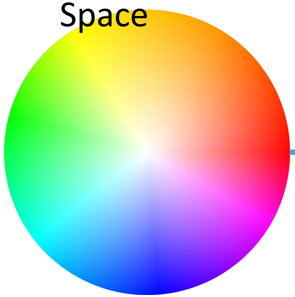
Handling Fabrication/Perception Errors

$$\underset{\delta}{\operatorname{argmin}}\ \lambda||M_x \cdot \delta||_p + \frac{1}{k}\sum_{i=1}^{k} J(f_\theta(x_i + M_x \cdot \delta), y^*) + NPS(M_x \cdot \delta)$$

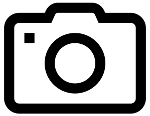
$$NPS(\delta) = \sum_{\hat{p} \in \delta} \prod_{p' \in P} |\hat{p} - p'|$$

P is a set of printable RGB triplets

Color Space



Sampled Set of RGB Triplets



NPS based on Sharif et al., “Accessorize to a crime,” CCS 2016