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

Bo Li

University of Illinois at Urbana-Champaign

Course Logistics

Class information & resources

- Course website: https://aisecure.github.io/TEACHING/CS562/CS562.html
- Forum: Canvas (link)
- Office hours: after class each day.
- My office: Siebel 4310
- TA: Huichen Li: huichen3@illinois.edu
  - Office hour: Friday 3-4 pm CDT (Zoom link)

Prerequisites & Enrollment

- All enrolled students must have taken machine learning classes
- Projects will require training neural networks with standard automatic differentiation packages (TensorFlow, Pytorch)
- Goal: Every group (max 2) in the class should have one top-tier conference paper for your project!

Grading Policy

Criteria	Percent of Grade
Project	65%
(Initial Proposal, Due 9.23)	(5%)
(Status Report, Due 10.26)	(20%)
(Final Report & Presentation, Due 12.2)	(40%)
Paper reading and presentation	30%
(Paper reviews)	(10%)
(Presentation)	(15%)
(Peer rating)	(5%)
Class participation	5%

Possible Hacking days:

- Attack/defense competition
- Privacy/defense competition
- Other ideas? Vote on Canvas

What we will cover

- Syllabus on course website
- Different types of machine learning algorithms
- Different types of adversarial attacks (different perturbation bounds, different semantics)
- Different types of detection/defense methods
- Secure learning by leveraging open-world info.
- Privacy problems in machine learning
- Fairness of machine learning
- Robustness of ML
- Open problems, research talks, invited lectures

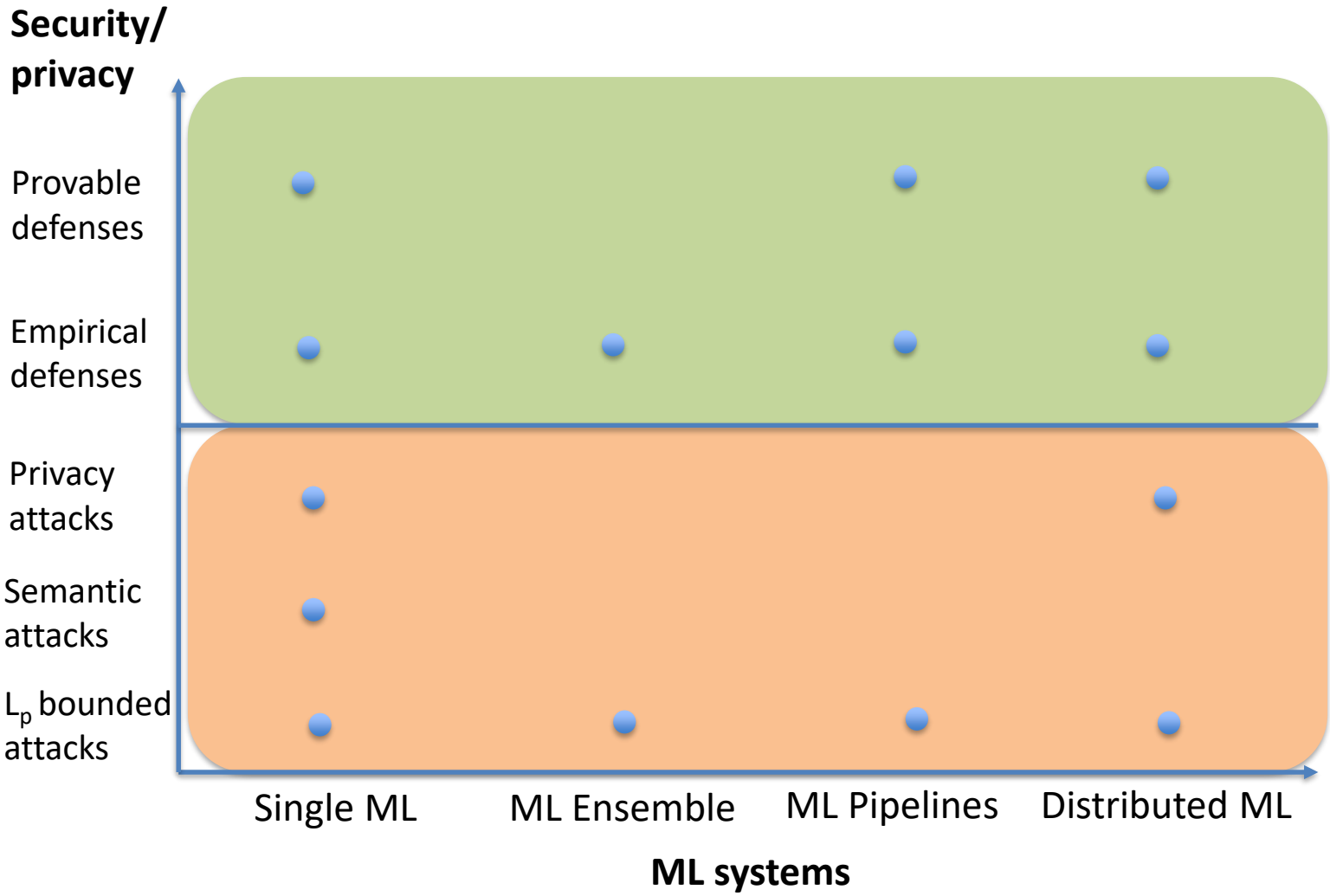
What we will not cover

- NO how to train GANs
- NO which network is more accurate on ImageNet
- NO playing RL games

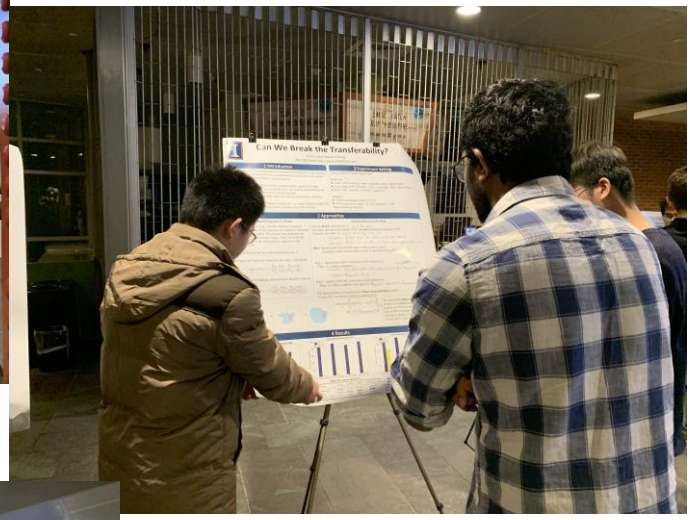
“Homework” today

- Start to form your final project group (maximum 2). If you prefer to work alone, it is also good
- Check out which topic you would like to present papers about and do project for (don’t need to be the same)
- Sign up for the presentation schedule
- Each class will have two presenters (from the same group or not)
- Please confirm with TA for your presentation topics by the end of next week
- Future: Please sign up for a time slot and we need to sync up to go through your slides before your presentation

Structure of the Course



Final Project



Final Project Topics

- Syllabus

What is adversarial learning, and why should we care?

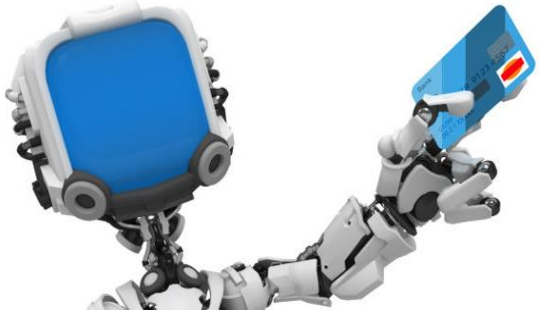
Machine Learning in Physical World

Autonomous Driving

Healthcare

Smart City

Malware Classification



Fraud Detection

Biometrics Recognition

Security & Privacy Problems

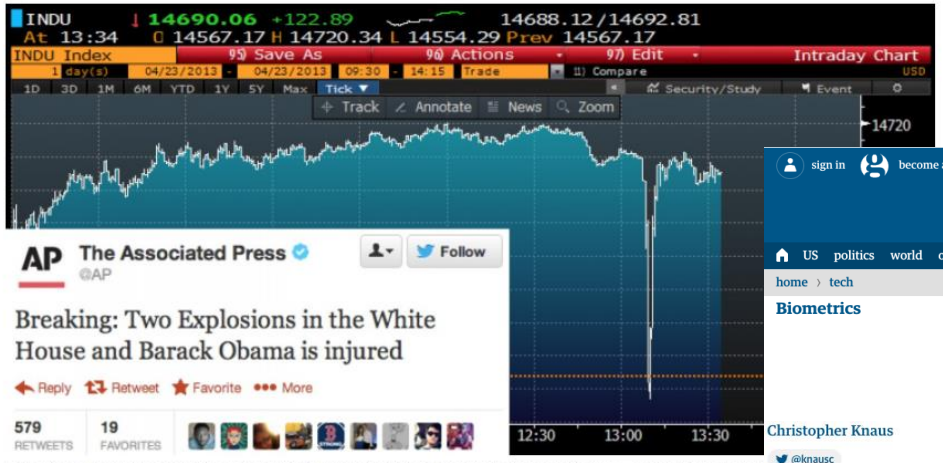
Sections

The Washington Post

WorldViews

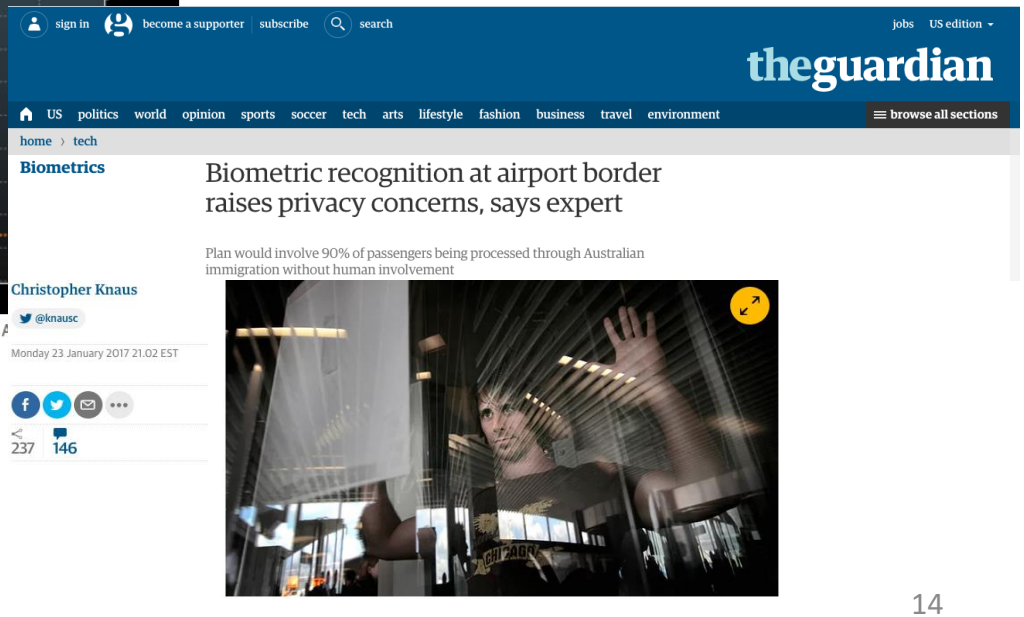
Syrian hackers claim AP hack that tipped stock market by $136 billion. Is it terrorism?

By Max Fisher April 23, 2013



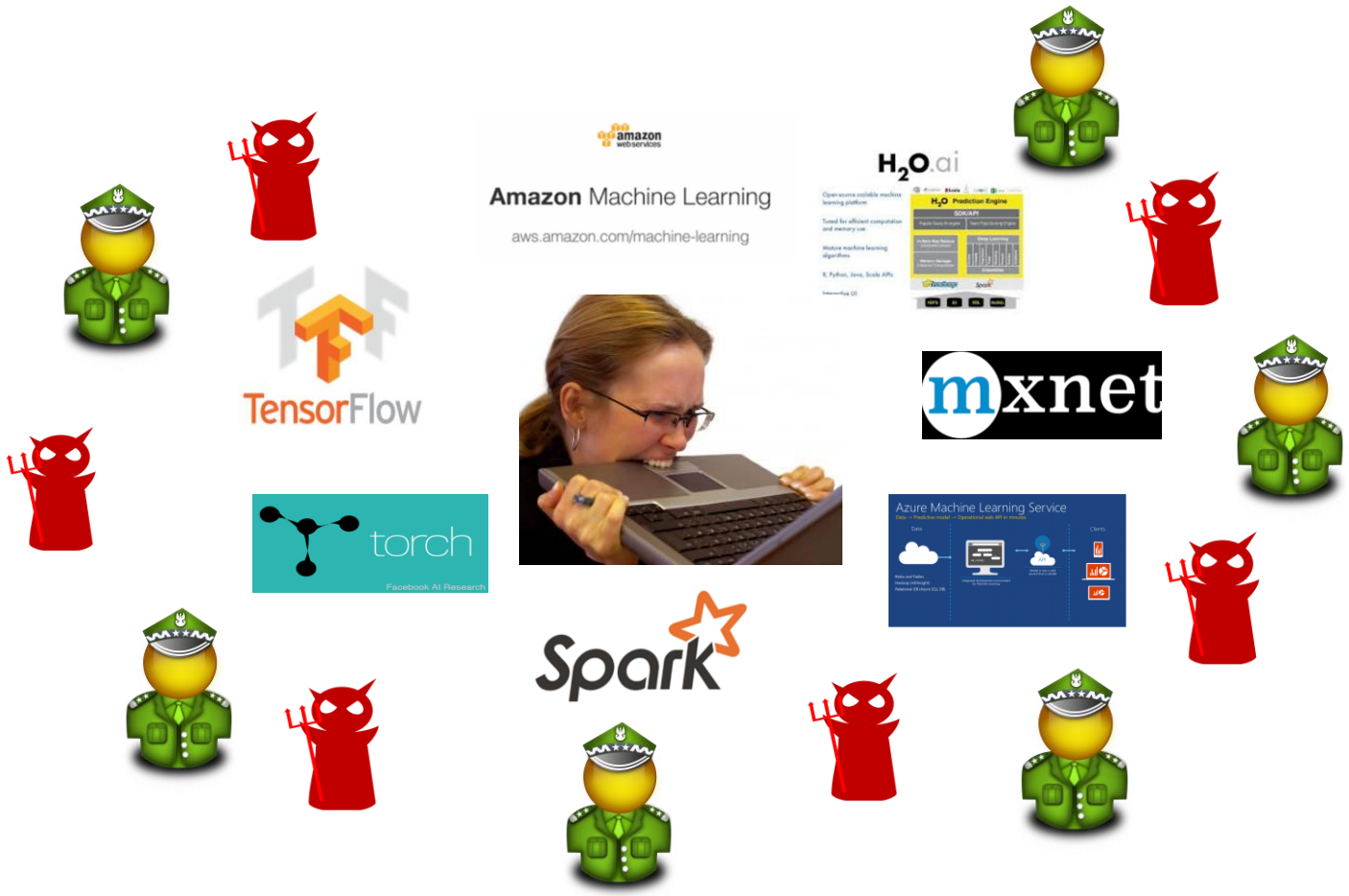
This chart shows the Dow Jones Industrial Average during Tuesday afternoon's drop, caused by a fake A

Security Problems



Privacy Concerns

We Are in Adversarial Environments

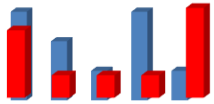


While cybersecurity R&D needs are addressed in greater detail in the NITRD Cybersecurity R&D Strategic Plan, some cybersecurity risks are specific to AI systems. ***One key research area is "adversarial machine learning"****, that explores the degree to which AI systems can be compromised by "contaminating" training data, by modifying algorithms, or by making subtle changes to an object that prevent it from being correctly identified….*

- National Science and Technology Council
2016

Perils of Stationary Assumption

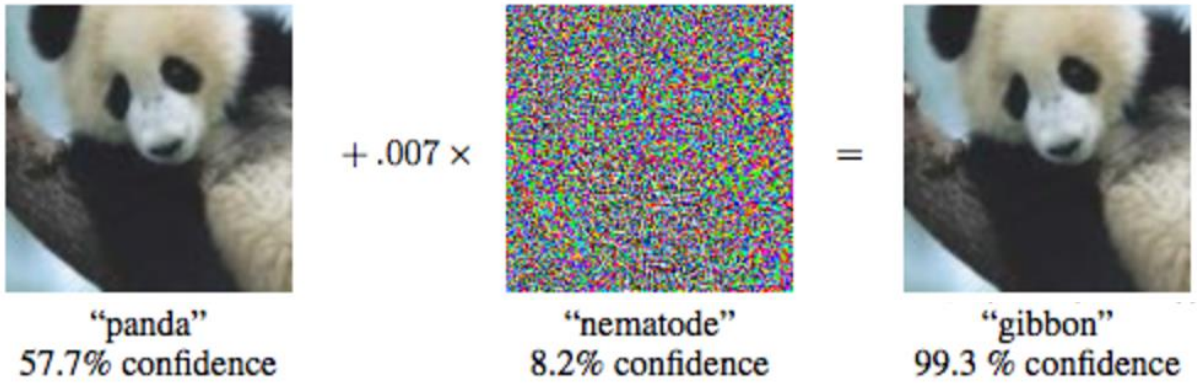
Traditional machine learning approaches assume

Training Data 

$\approx$

Testing Data 

Adversarial Examples



$$\boldsymbol{\eta} = \epsilon \text{sign}(\nabla_x J(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y}))$$

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *ICLR 2015.*

Li, Bo, Yevgeniy Vorobeychik, and Xinyun Chen. "A General Retraining Framework for Scalable Adversarial Classification." ICLR. (2016).

Lab Test Summary (Stationary)

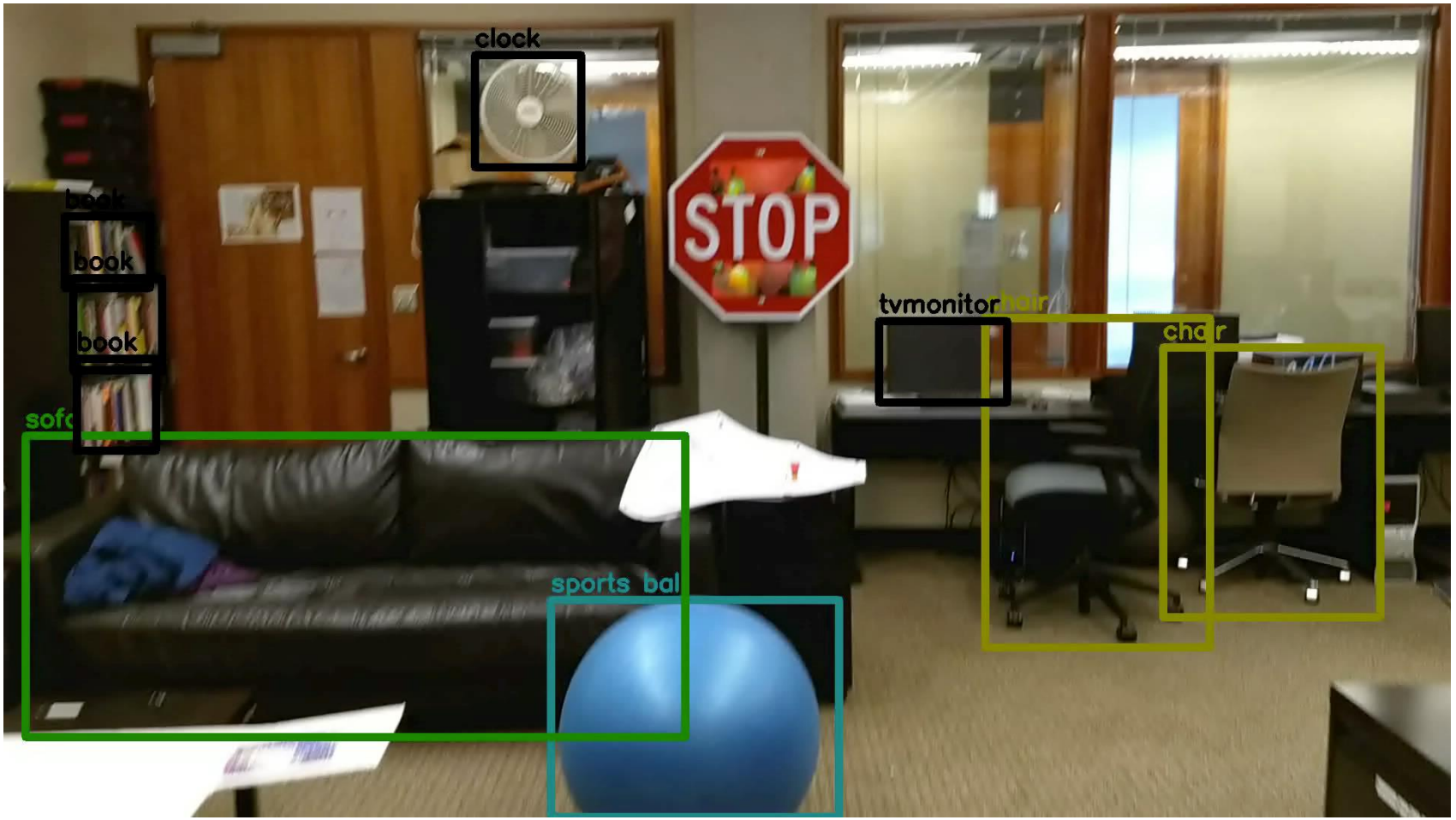
Subtle Poster | Subtle Poster | Camo Graffiti | Camo Art | Camo Art

Misclassify

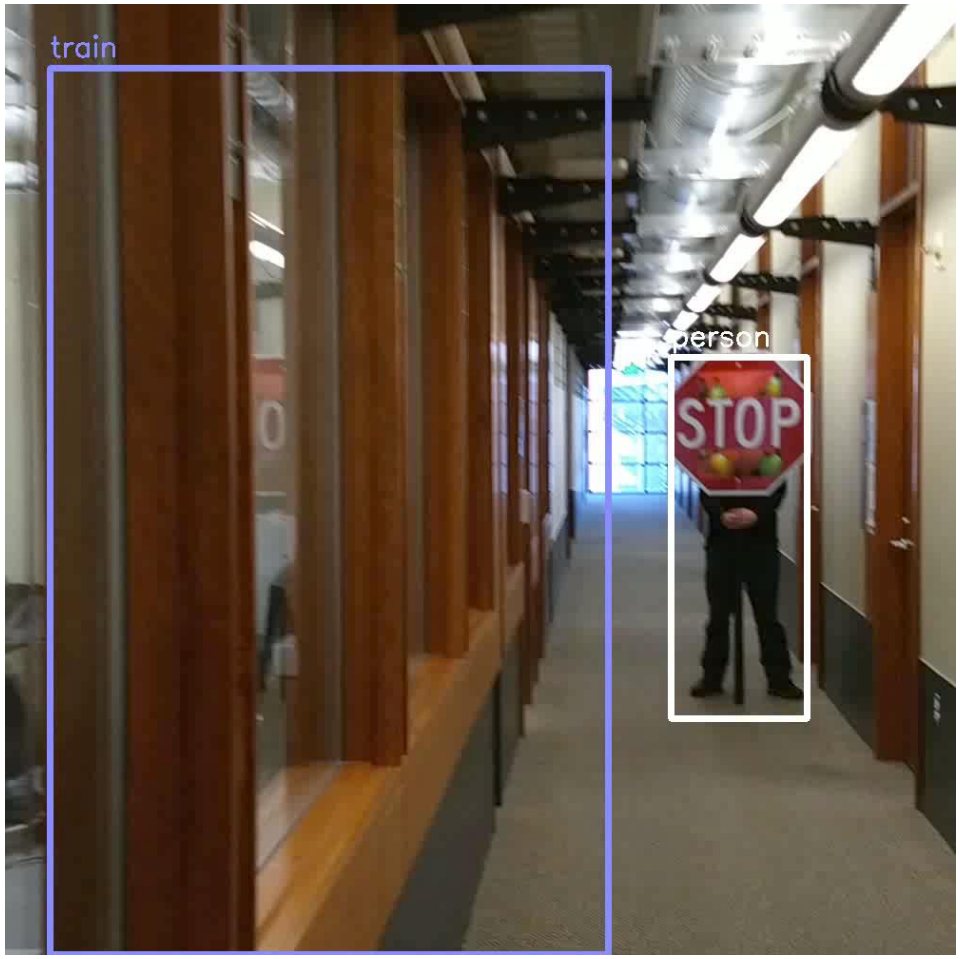
Adversarial Target

Art Perturbation

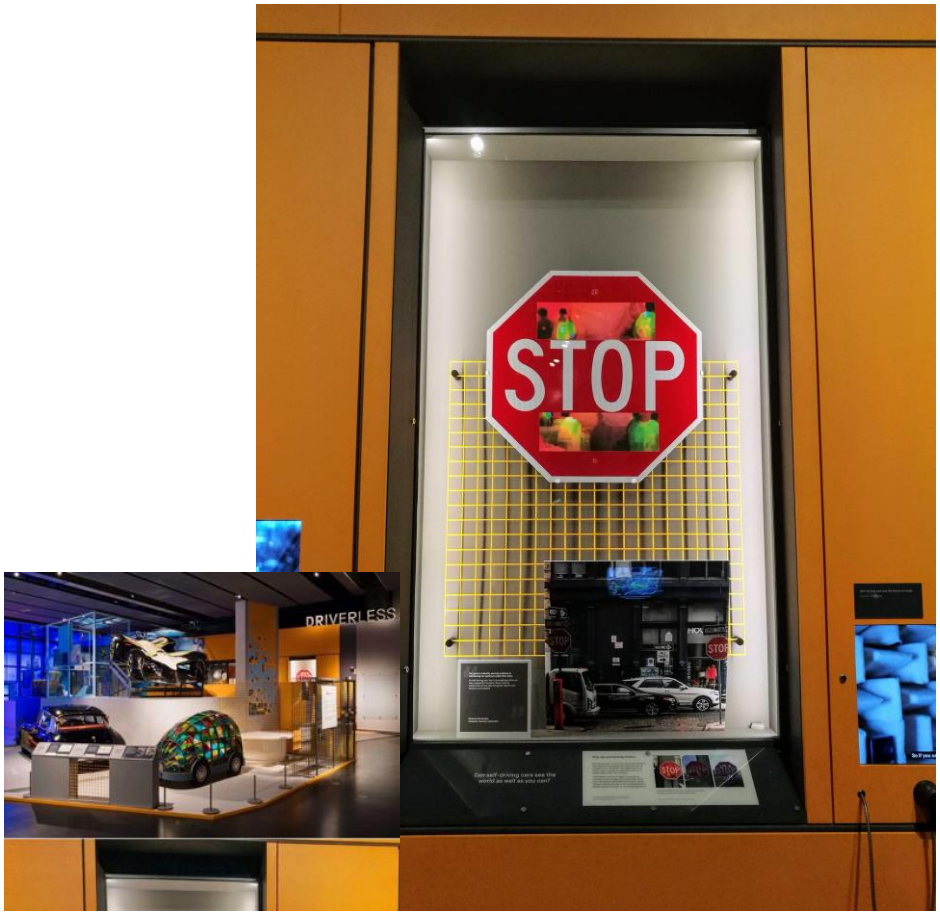
Physical Attacks Against Detectors



Physical Attacks Against Detectors



Physical Adversarial Stop Sign in the Science Museum of London



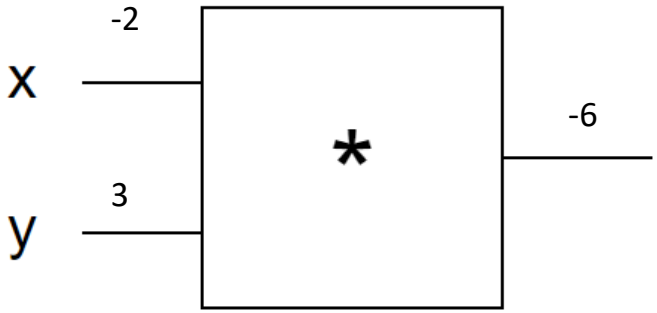
SCIENCE MUSEUM

DRIVELSS: WHO IS IN CONTROL?

Deep Learning Mini Crash Course

- Neural Networks Background
- Convolutional Neural Networks (CNNs)

Real-Valued Circuits



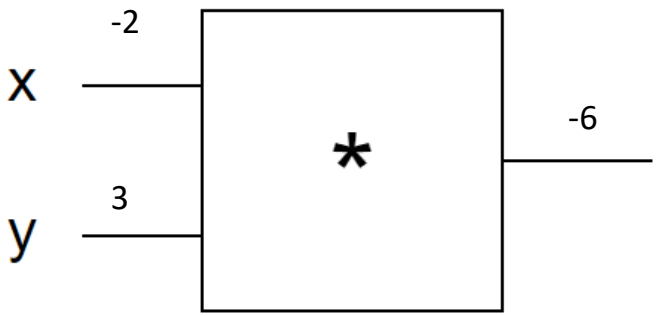
$$f(x, y) = xy$$

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

- Tweak the inputs. But how?
- Option 1. Random Search?

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

Real-Valued Circuits



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

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

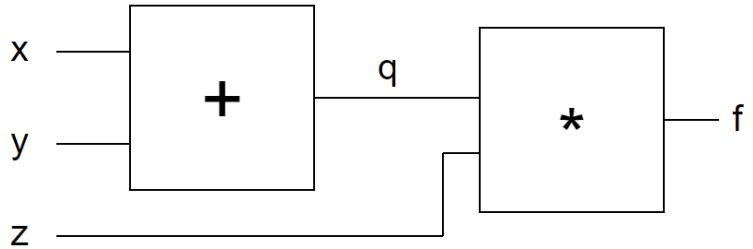
- Option 2. Analytic Gradient

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

Limit as h -> 0

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

Composable Real-Valued Circuits



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

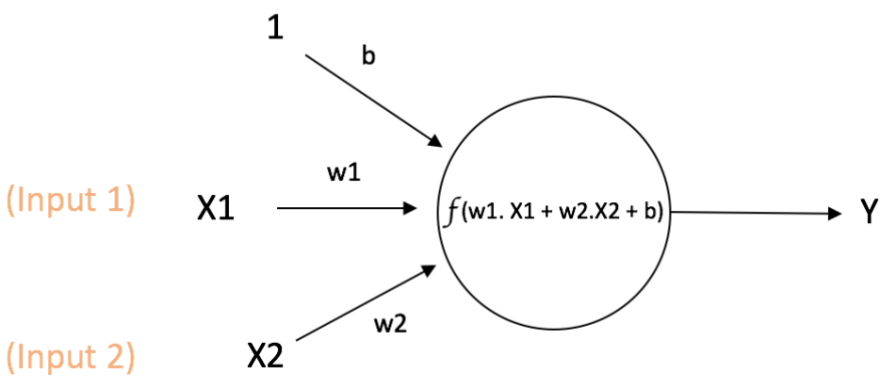
$$f(q, z) = qz \quad \Longrightarrow \quad \frac{\partial f(q,z)}{\partial q} = z, \qquad \frac{\partial f(q,z)}{\partial z} = q$$

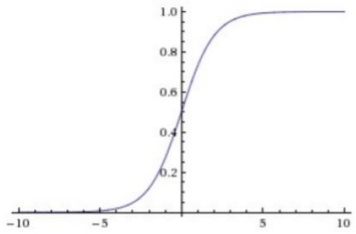
$$q(x, y) = x + y \quad \Longrightarrow \quad \frac{\partial q(x,y)}{\partial x} = 1, \qquad \frac{\partial q(x,y)}{\partial y} = 1$$

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

Backpropagation!

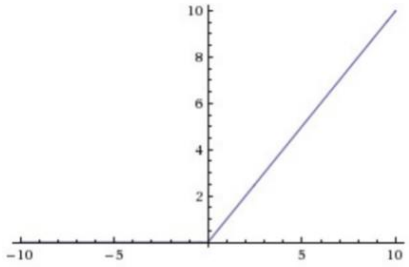
Single Neuron



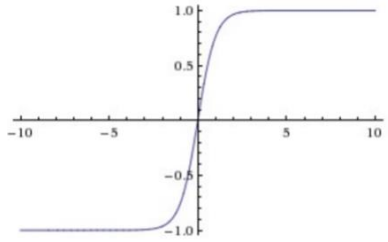


Sigmoid

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



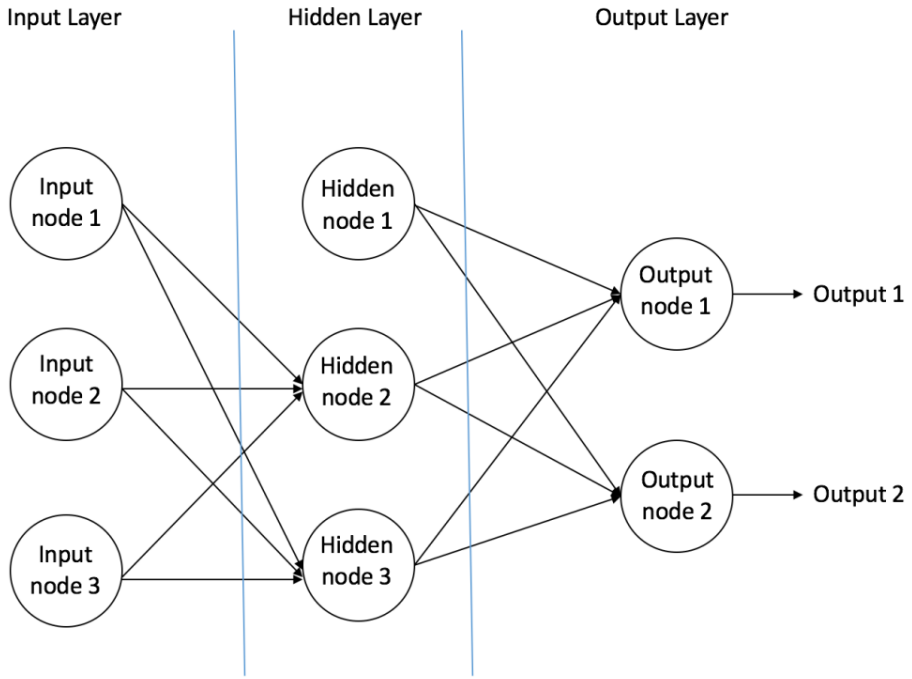
ReLU



tanh

Activation function

(Deep) Neural Networks!



Organize neurons into a structure

Train (Optimize) using backpropagation

Convolutional Neural Networks (CNNs)

Very widely used, and very useful

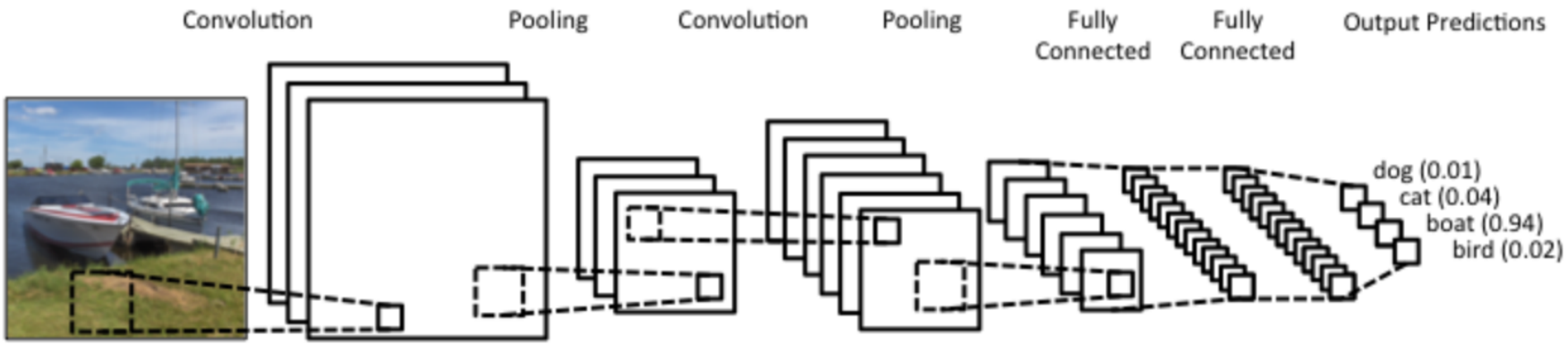
a plate with a sandwich and a salad

a group of motorcycles parked in front of a building

a man riding a wave on top of a surfboard

http://cs.stanford.edu/people/karpathy/neuraltalk2/demo.html

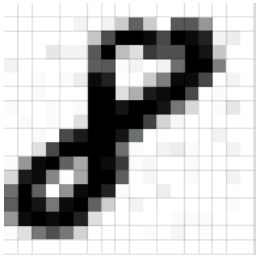
Convolutional Neural Networks (CNNs)



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

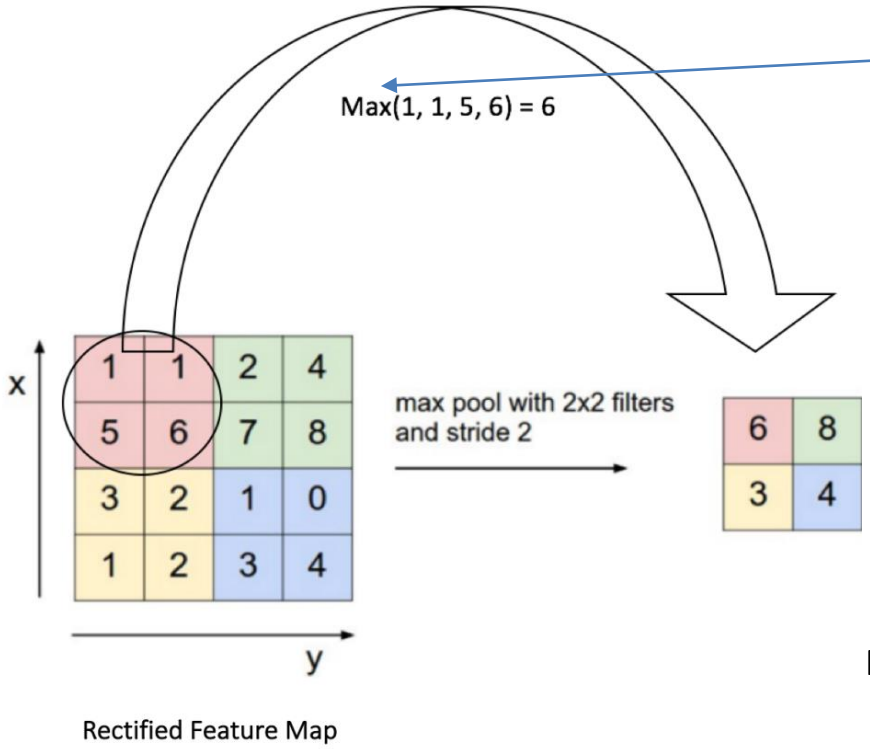
- Slide the kernel over the input matrix
- Compute element wise multiplication (Hadamard/schur product), add results to get a single value
- Output is a feature map

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)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

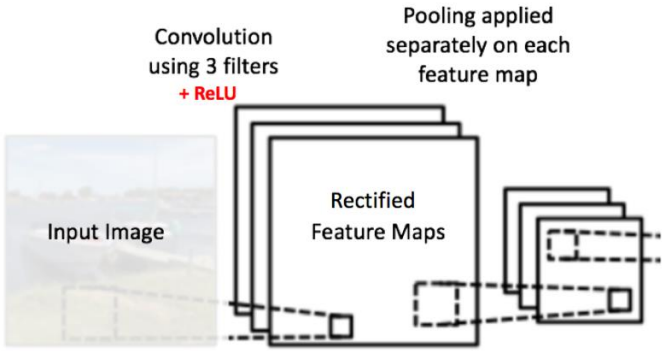
Many types of filters

A CNN learns these filters during training

Pooling

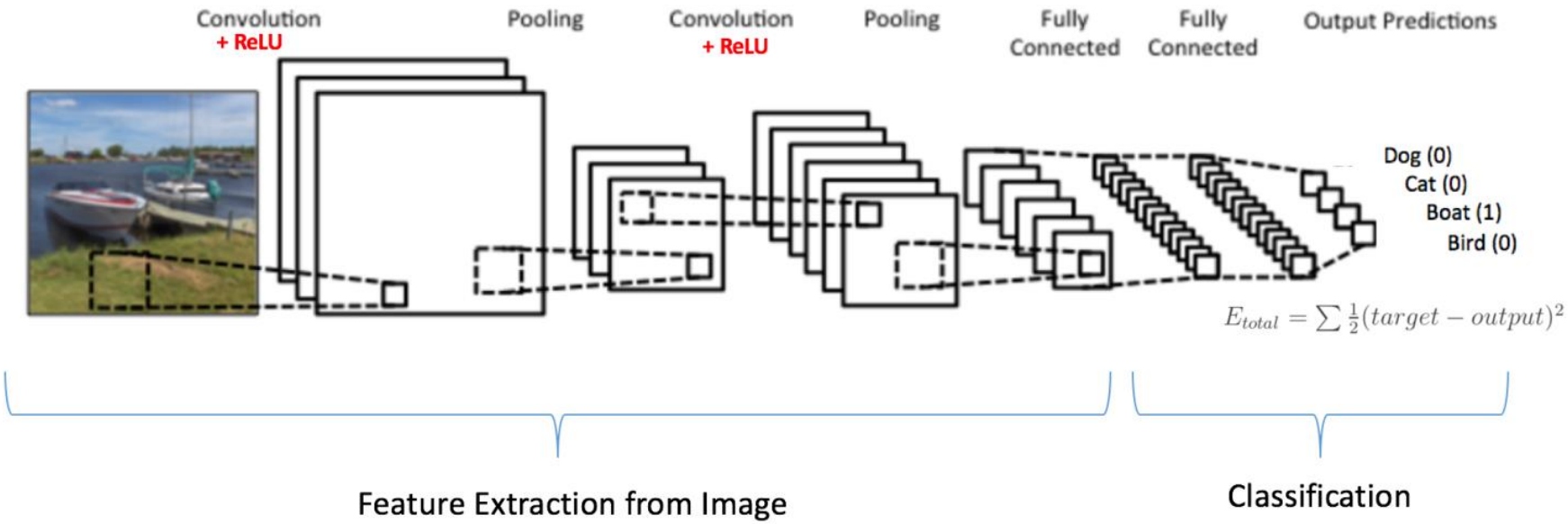


Can be Avg, sum, min, …



Reduce dimensionality, but retain important features

Putting Everything Together



Digital Adversarial Example

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 perturbation of non-linear models

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

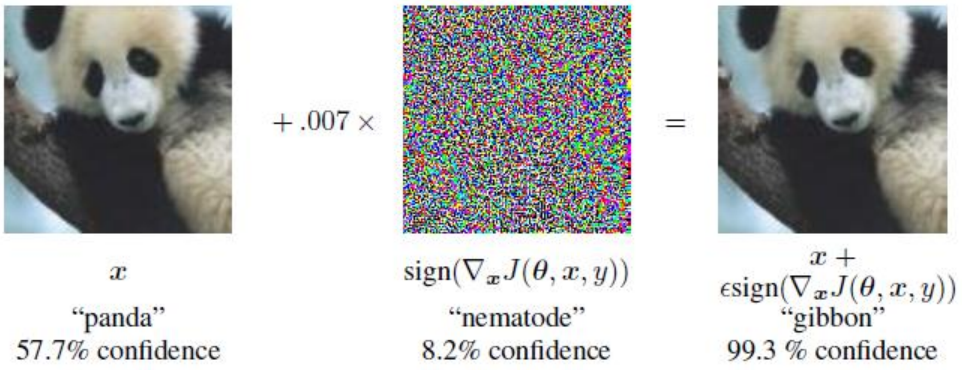
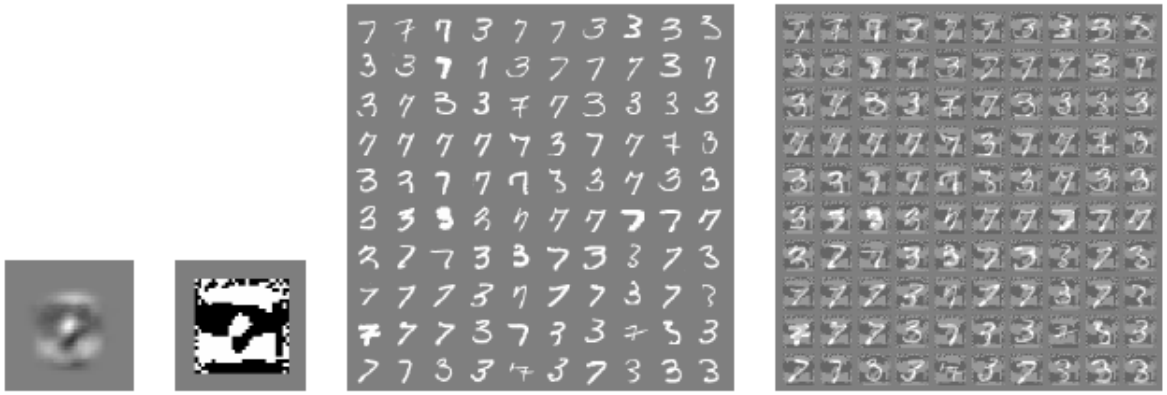


Image from reference paper

Fast gradient sign – logistic regression



1.6% error rate

99% error rate

Image from reference paper

Adversarial training of deep networks

- Deep networks are vulnerable to adversarial examples
- 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%

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

Alternate Hypothesis

- Generative training
  - MP-DBM: ϵ of 0.25, error rate of 97.5% on adversarial examples generated from the MNIST
  - Being generative alone is not sufficient
- Ensemble training
  - Ensemble of 12 maxout networks on MNIST: ϵ of 0.25, 91.1% error on adversarial examples on MNIST
  - One member of the ensemble: 87.9% error

Summary

- Some studies show that adversarial examples are a result of models being too linear
- 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

Numerous Defenses Proposed