

Tutorial on Adversarial Machine Learning with CleverHans

Nicholas Carlini

University of California, Berkeley

Nicolas Papernot

Pennsylvania State University

Did you git clone `https://github.com/carlini/odsc_adversarial_nn?`



Getting setup

If you have not already:

```
git clone https://github.com/carlini/odsc_adversarial_nn
```

```
cd odsc_adversarial_nn
```

```
python test_install.py
```

Why neural networks?

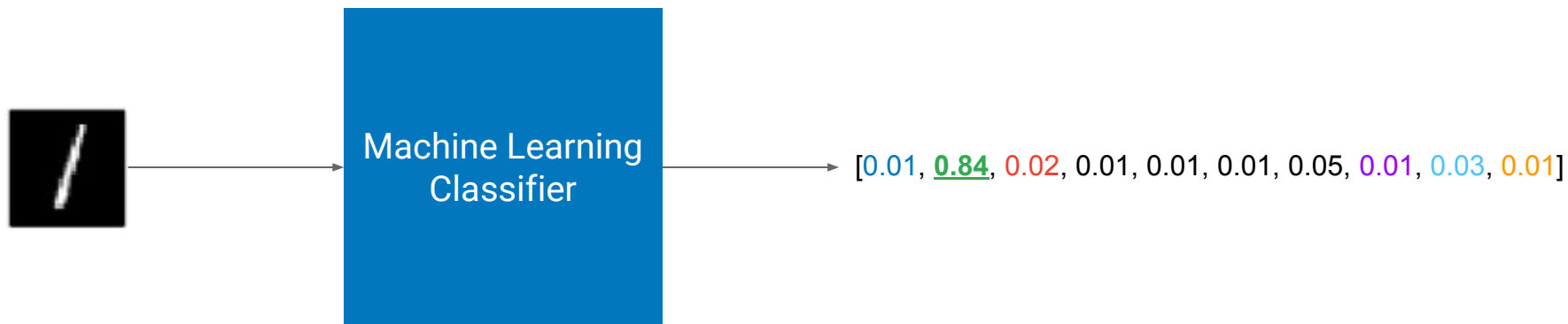
IM  GENET



AlphaGo

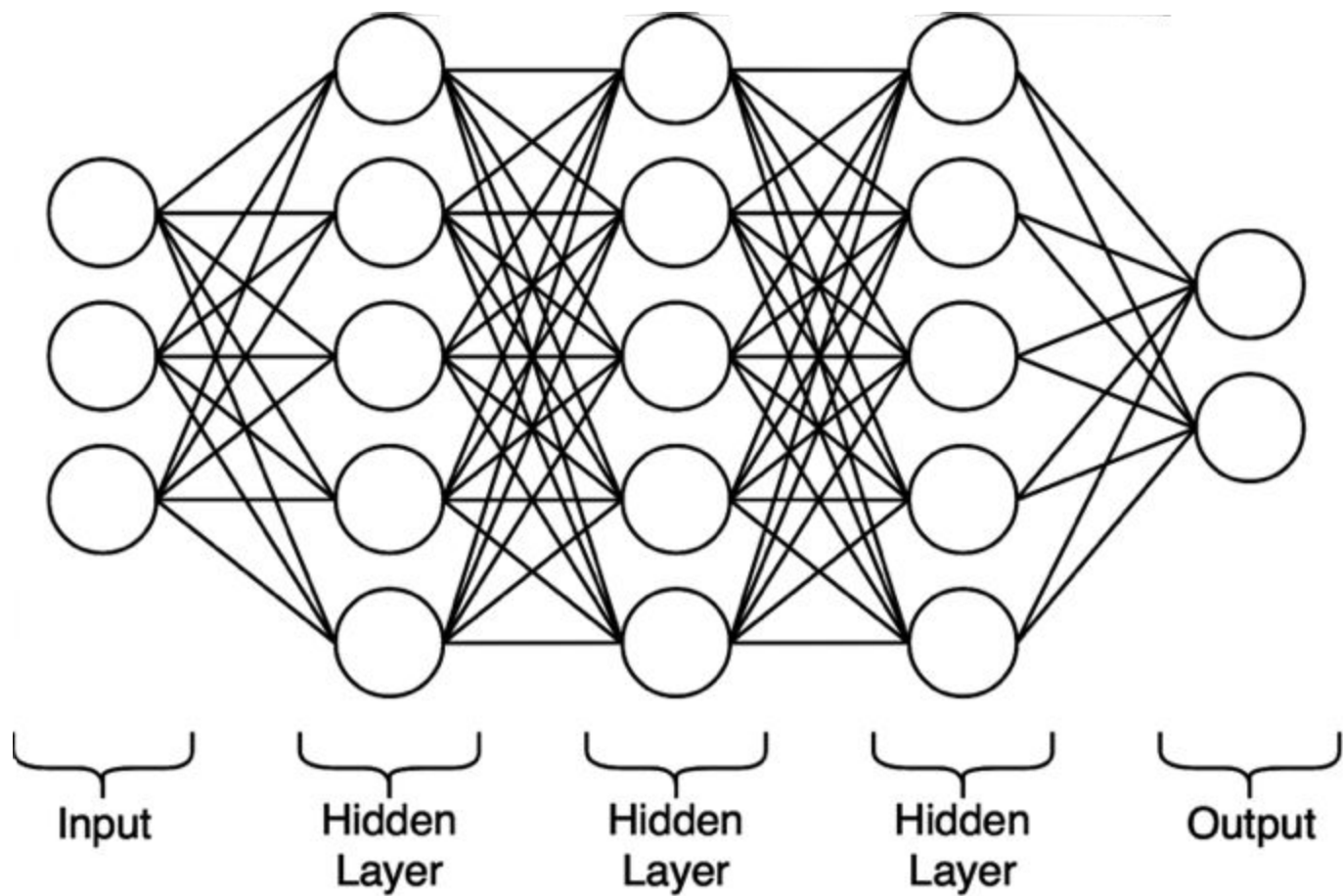


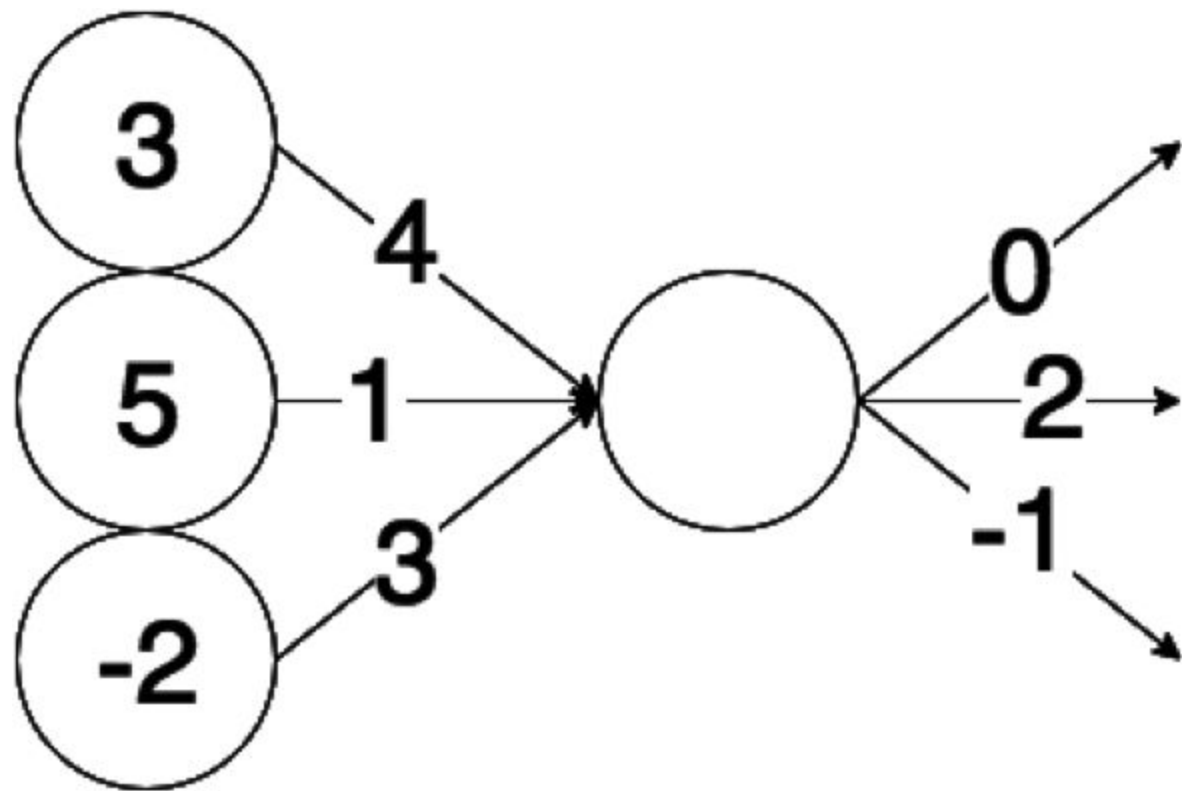
Classification with neural networks

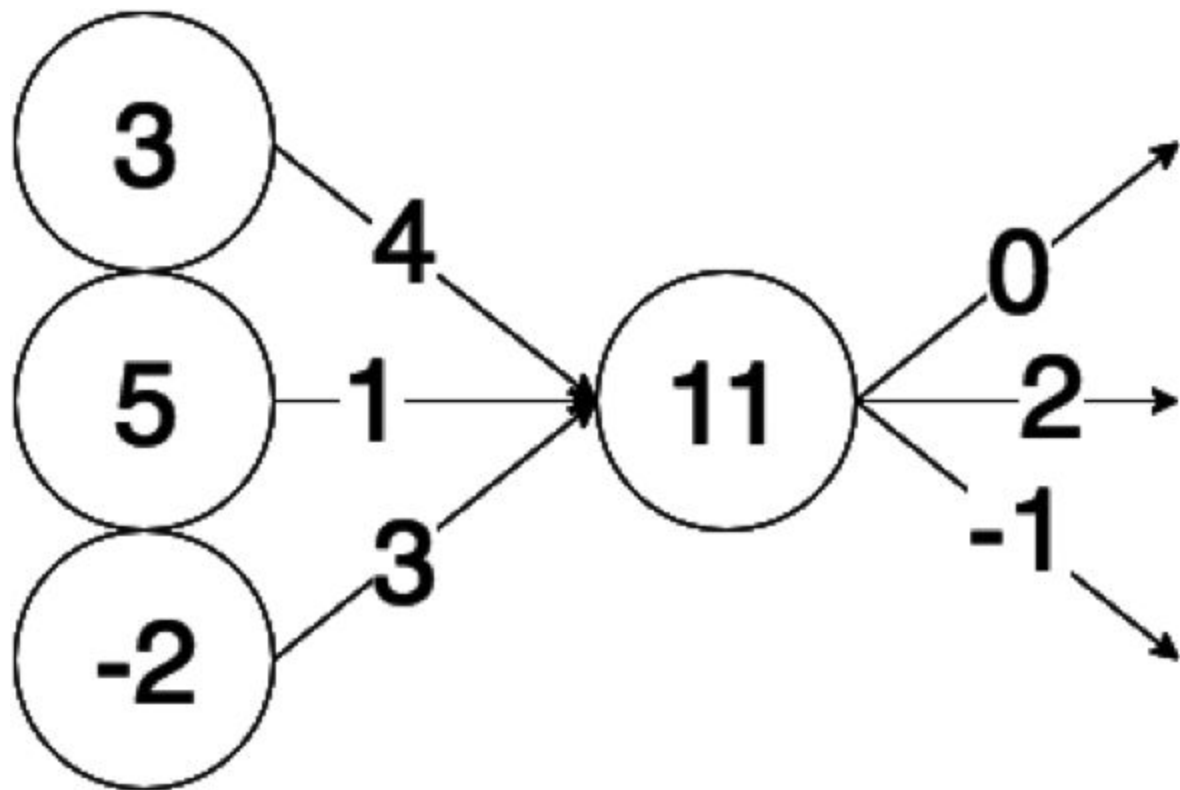


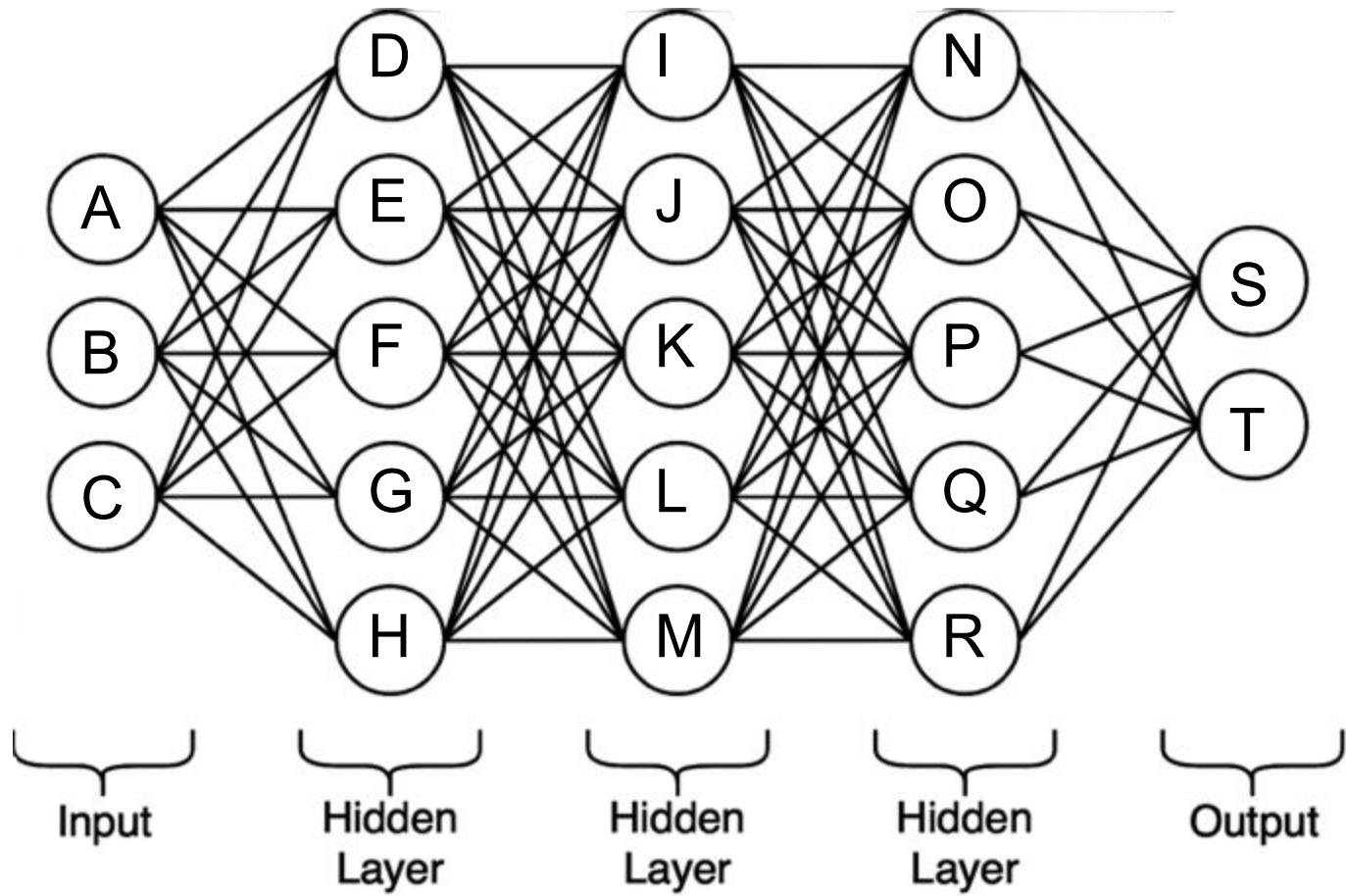
$$x \longrightarrow f(x, \theta) \longrightarrow [p(0|x, \theta), p(1|x, \theta), p(2|x, \theta), \dots, p(7|x, \theta), p(8|x, \theta), p(9|x, \theta)]$$

Classifier: map inputs to one class among a predefined set

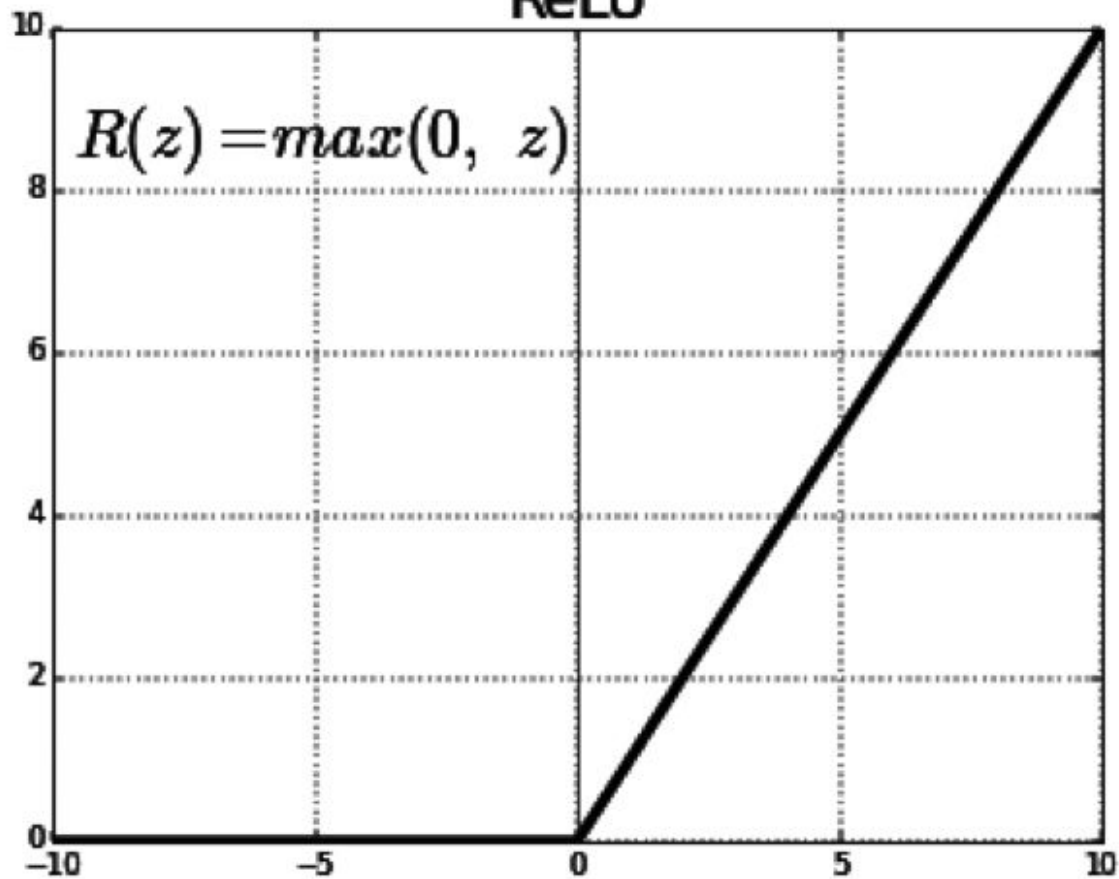


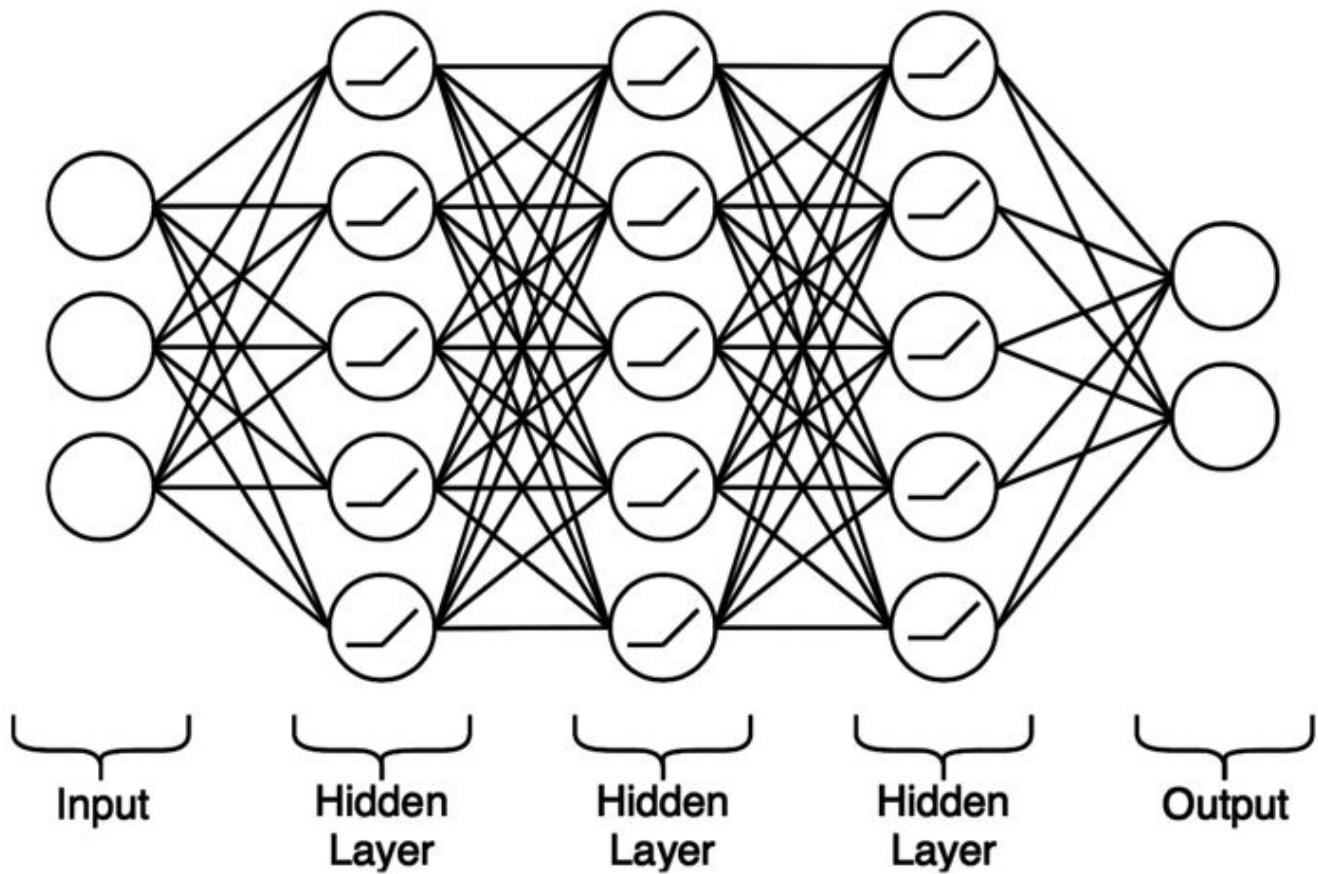


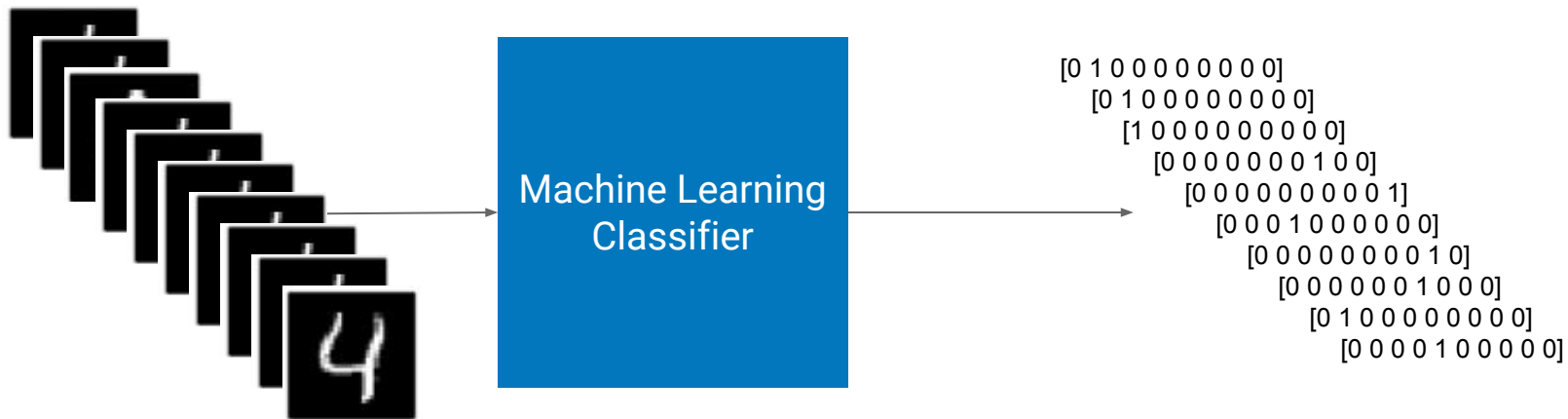




ReLU







Learning: find internal classifier parameters θ that minimize a cost/loss function (\sim model error)

NNs give better results than any other approach

But there's a catch ...

Adversarial examples



“panda”
57.7% confidence

+ .007 ×

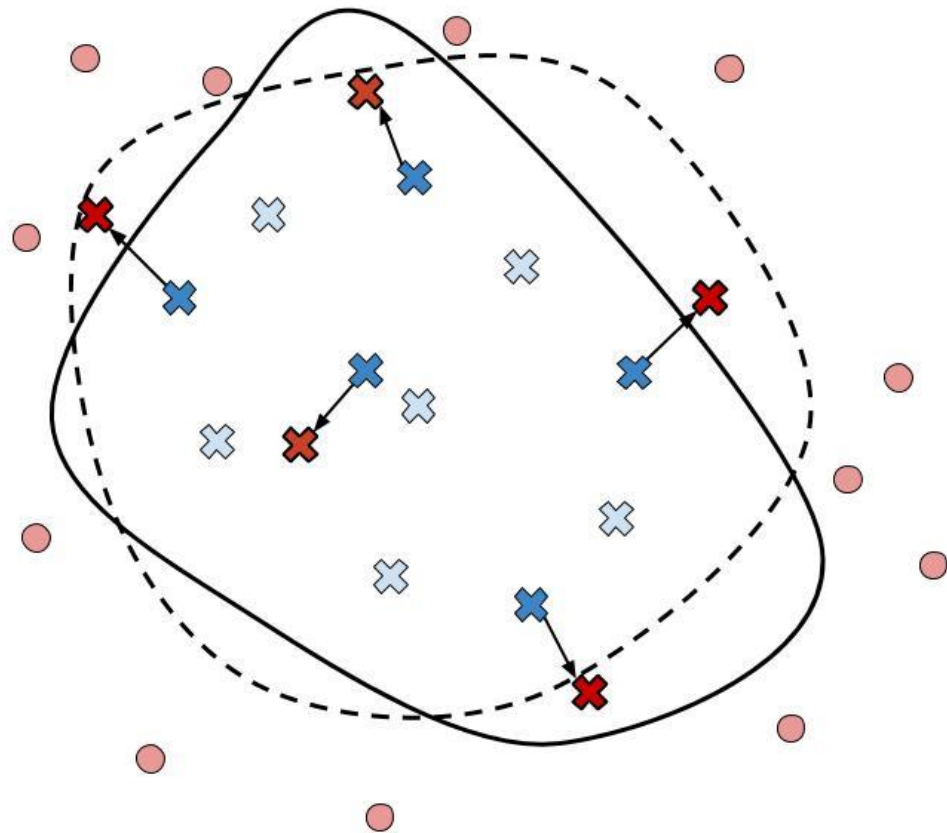


“nematode”
8.2% confidence

=



“gibbon”
99.3 % confidence



- - - Task decision boundary
- Model decision boundary
- ⊗ Testing points for class 1
- ⊗ Training points for class 1
- Training points for class 2
- ⊗ Adversarial examples for class 1

Crafting adversarial examples: *fast gradient sign method*

During training, the classifier uses a loss function to **minimize** model prediction errors

After training, **attacker** uses loss function to **maximize** model prediction error

1. Compute its gradient with respect to the input of the model

$$\nabla_x J(\theta, x, y)$$

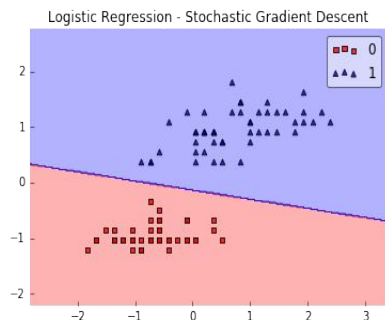
2. Take the sign of the gradient and multiply it by a threshold

$$x + \varepsilon \cdot \text{sgn}(\nabla_x J(\theta, x, y))$$

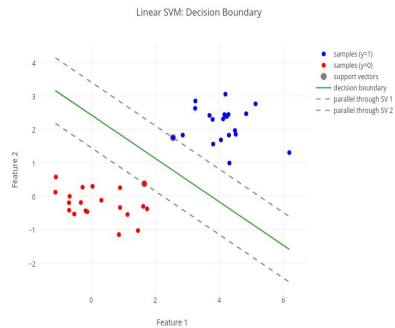
Transferability

Source Machine Learning Technique	DNN	LR	SVM	DT	kNN
DNN	38.27	23.02	64.32	79.31	8.36
LR	6.31	91.64	91.43	87.42	11.29
SVM	2.51	36.56	100.0	80.03	5.19
DT	0.82	12.22	8.85	89.29	3.31
kNN	11.75	42.89	82.16	82.95	41.65

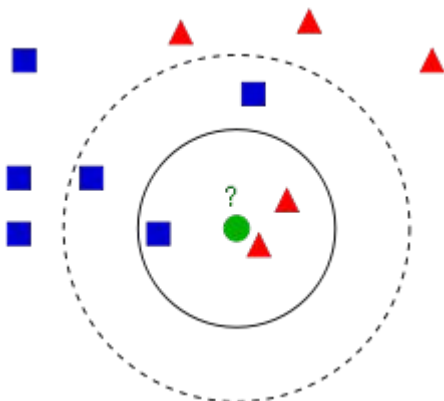
Not specific to neural networks



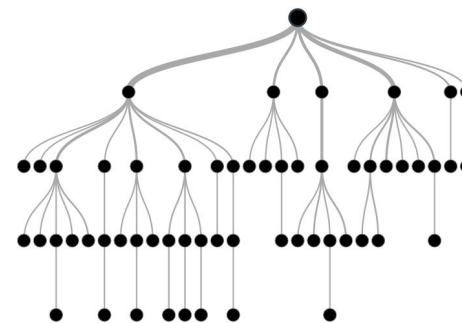
Logistic regression



SVM



Nearest Neighbors



Decision Trees

Machine Learning with TensorFlow

```
import tensorflow as tf

sess = tf.Session()

five = tf.constant(5)

six = tf.constant(6)

sess.run(five+six) # 11
```

Machine Learning with TensorFlow

```
import tensorflow as tf

sess = tf.Session()

five = tf.constant(5)

number = tf.placeholder(tf.float32, [])

added = five+number

sess.run(added, {number: 6}) # 11

sess.run(added, {number: 8}) # 13
```

Machine Learning with TensorFlow

```
import tensorflow as tf

number = tf.placeholder(tf.float32, [])

squared = number * number

derivative = tf.gradients(squared, [number])[0]

sess.run(derivative, {number: 5}) # 10
```

Classifying ImageNet with the Inception Model [Hands On]



Attacking ImageNet



[Pull requests](#)
[Issues](#)
[Marketplace](#)
[Explore](#)

[tensorflow / cleverhans](#)
Unwatch 75
Star 1,154
Fork 272

[Code](#)
[Issues 8](#)
[Pull requests 2](#)
[Projects 0](#)
[Wiki](#)
[Settings](#)
[Insights](#)

A library for benchmarking vulnerability to adversarial examples
 [Edit](#)

[machine-learning](#)
[security](#)
[benchmarking](#)
[Manage topics](#)


[997 commits](#)
[1 branch](#)
[4 releases](#)
[34 contributors](#)
[MIT](#)

Branch: [master](#)
[New pull request](#)
[Create new file](#)
[Upload files](#)
[Find file](#)
[Clone or download](#)

AlexeyKurakin committed on GitHub	Merge pull request #251 from AlexeyKurakin/master	Latest commit 2fbd28e a day ago
assets	add PSD file for logo	2 months ago
cleverhans	Merge pull request #244 from goodfeli/stop_gradient	6 days ago
cleverhans_tutorials	Add the bias to Conv2D computation in tutorial_mnist_tf. The bias was...	6 days ago
examples	Fixing comments	a day ago
tests_tf	move test_attacks_tf into tests_tf folder	6 days ago
tests_th	Update deprecated usage	3 months ago
.gitignore	make accuracy tests run on travis	6 months ago
.travis.yml	tensorflow apparently ignores log level flag	2 months ago
CODE_OF_CONDUCT.rst	update capitalization when using CleverHans as a library name	3 months ago
CONTRIBUTING.md	complete CONTRIBUTING.md file	2 months ago
LICENSE	Add Google Inc. to LICENSE	3 months ago
README.md	avoid implying that contents of master are v2.0.0	7 days ago
requirements.txt	remove theano from requirements and install in Travis only	3 months ago
setup.py	fix typo	2 months ago

[README.md](#)

CleverHans (latest release: v2.0.0)



cleverhans

build passing

Growing community

1.3K+ stars

300+ forks

40+ contributors

Attacking the Inception Model for ImageNet [Hands On]

```
python attack.py
```

Replace panda.png with adversarial_panda.png

```
python classify.py
```

Things to try:

1. Replace the given image of a panda with your own image
2. Change the target label which the adversarial example should be classified as

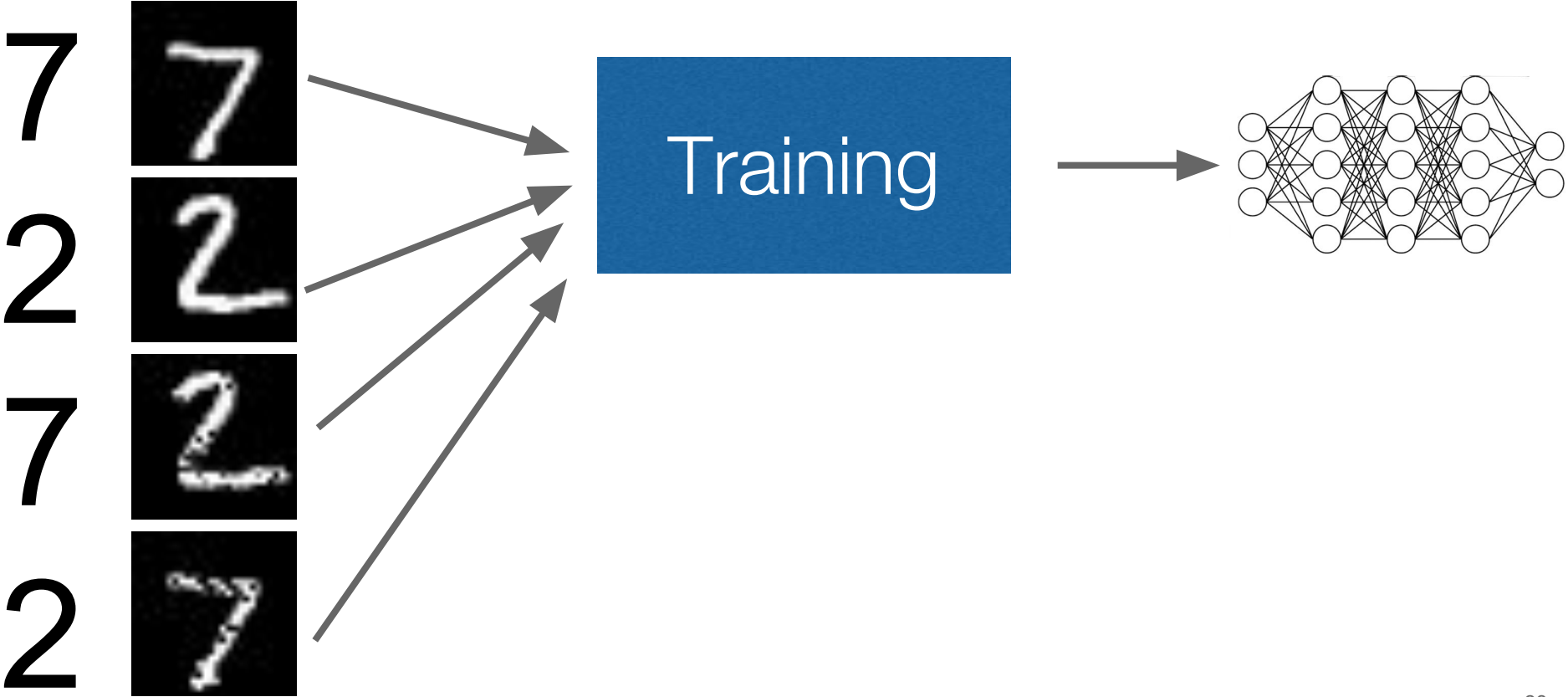
Adversarial Training



Adversarial Training



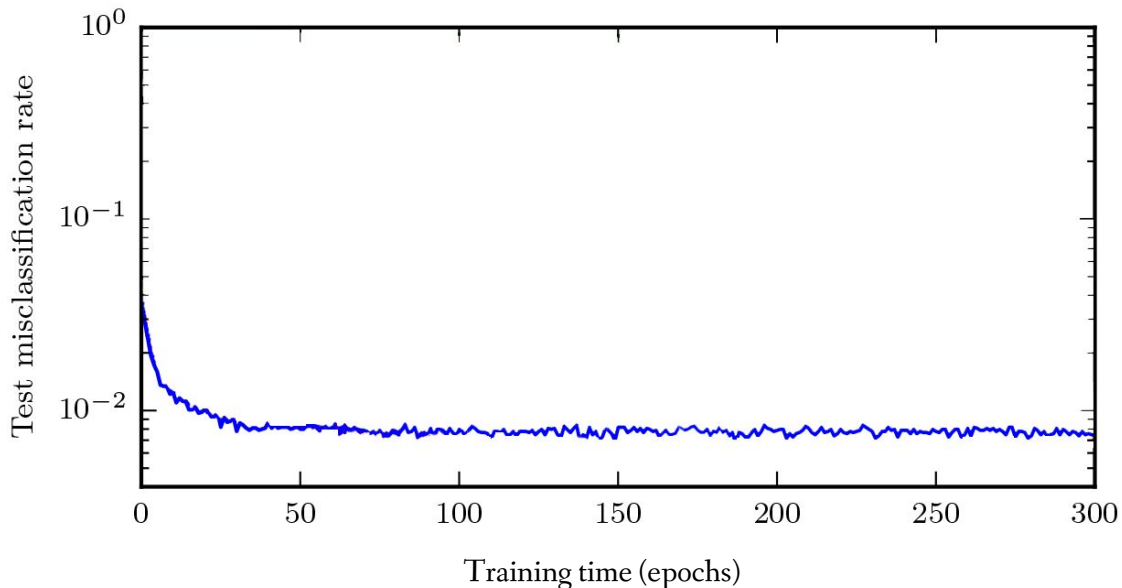
Adversarial Training



Adversarial training

Intuition: **injecting** adversarial example during training with **correct** labels

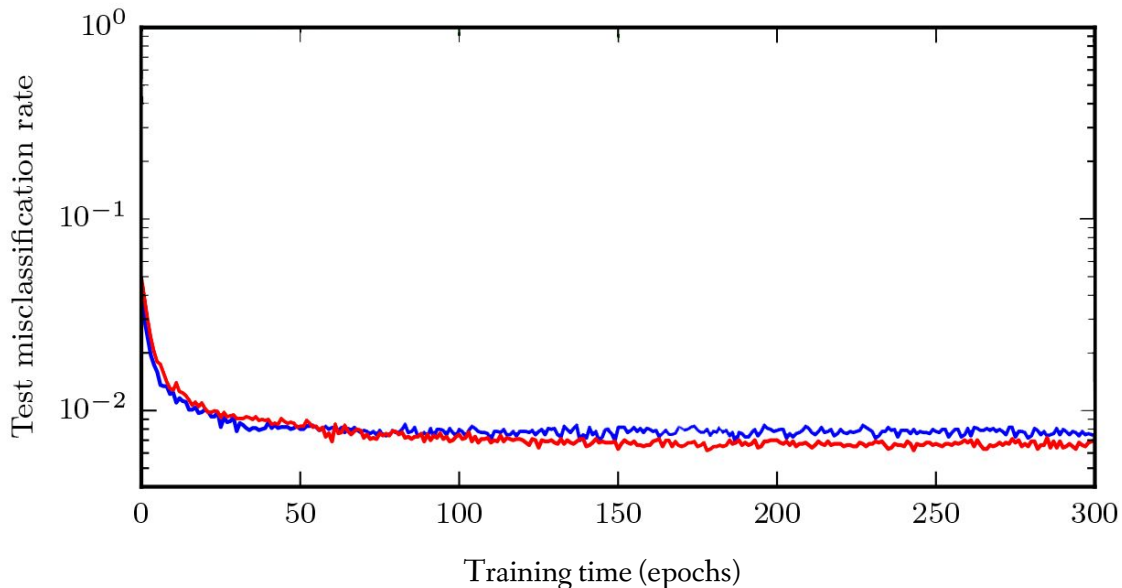
Goal: improve model **generalization** outside of training manifold



Adversarial training

Intuition: **injecting** adversarial example during training with **correct** labels

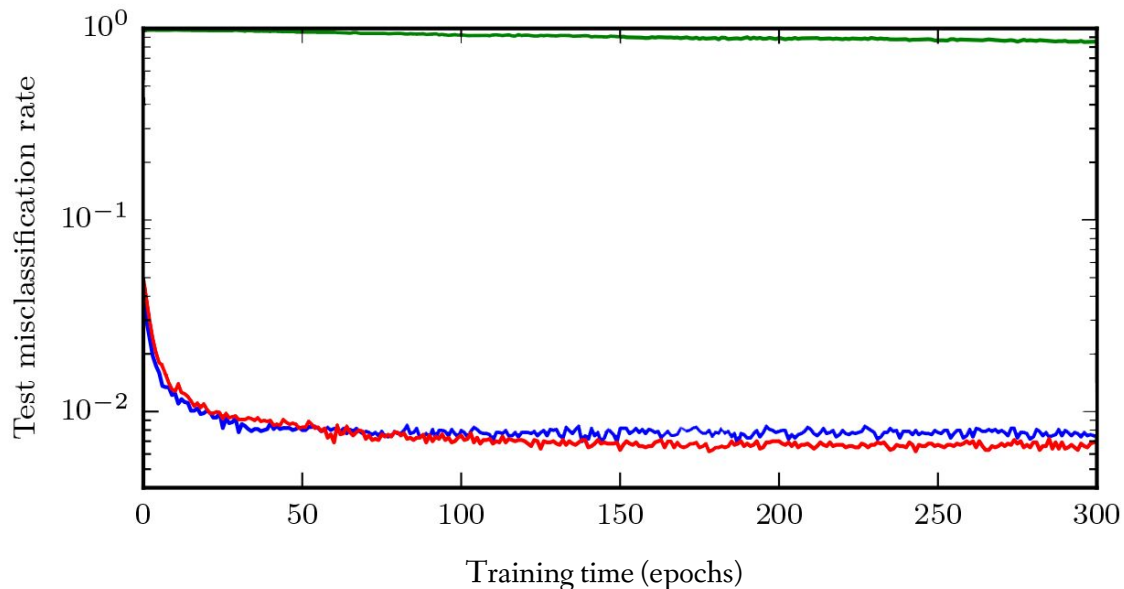
Goal: improve model **generalization** outside of training manifold



Adversarial training

Intuition: **injecting** adversarial example during training with **correct** labels

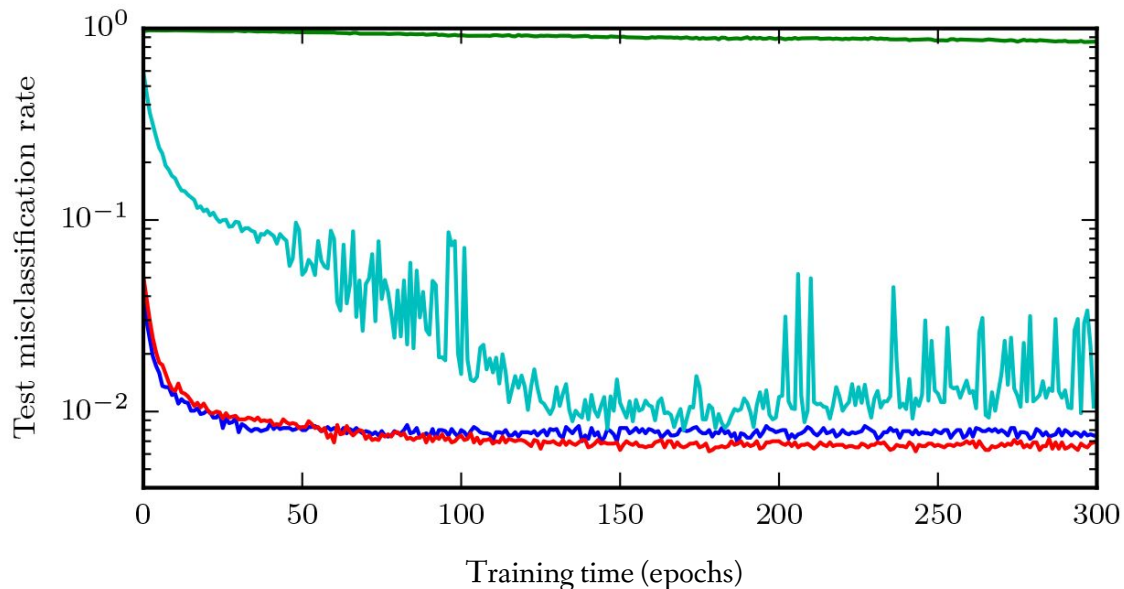
Goal: improve model **generalization** outside of training manifold



Adversarial training

Intuition: **injecting** adversarial example during training with **correct** labels

Goal: improve model **generalization** outside of training manifold



Efficient Adversarial Training through Loss Modification

$$\text{loss}(x, y)$$



Small when prediction is
correct on legitimate input

Efficient Adversarial Training through Loss Modification

$$\text{loss}(x, y) + \text{loss}(x + \epsilon \cdot \mathbf{sign}(\text{grad}), y)$$



Small when prediction is
correct on legitimate input

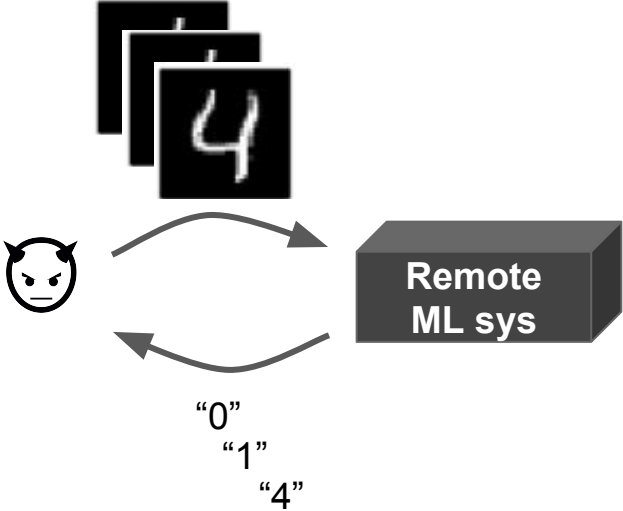


Small when prediction is
correct on adversarial input

Adversarial Training Demo

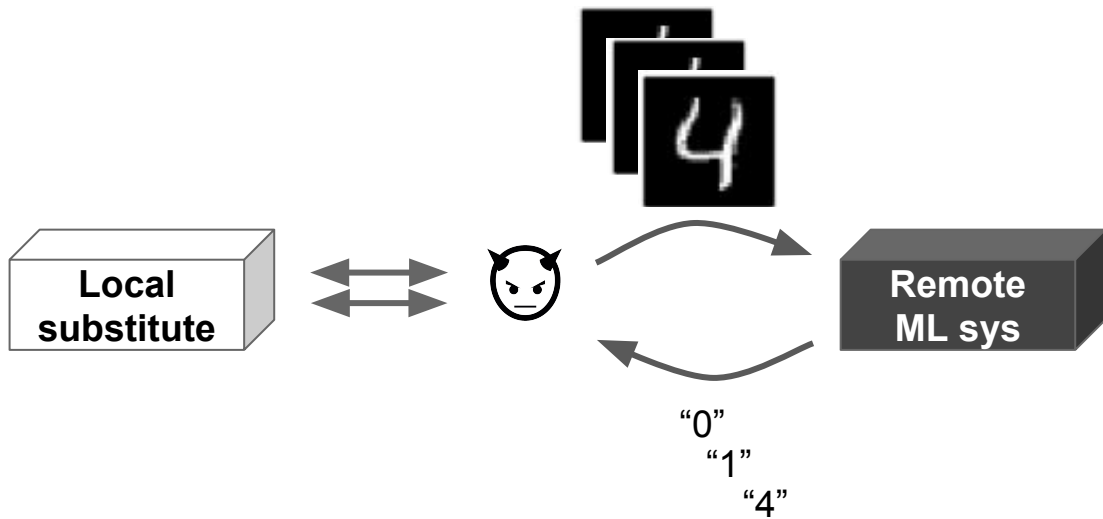


Attacking remotely hosted black-box models



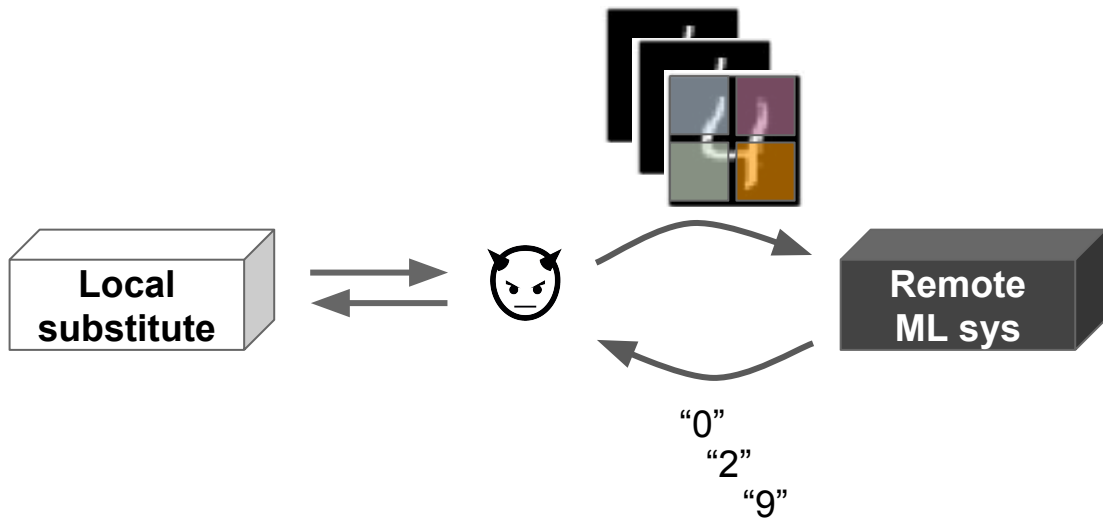
(1) The adversary queries remote ML system for labels on inputs of its choice.

Attacking remotely hosted black-box models



(2) The adversary uses this labeled data to train a local substitute for the remote system.

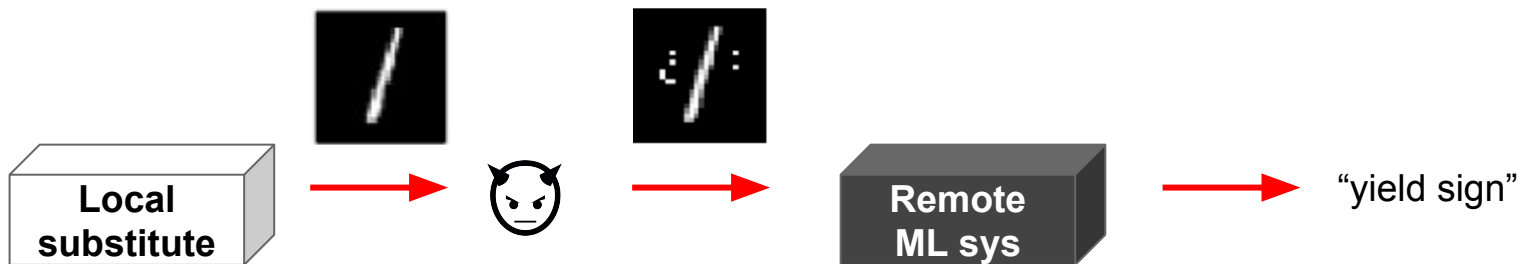
Attacking remotely hosted black-box models



$$S_{\rho+1} = \{\vec{x} + \lambda_{\rho+1} \cdot \text{sgn}(J_F[\tilde{O}(\vec{x})]) : \vec{x} \in S_{\rho}\} \cup S_{\rho}$$

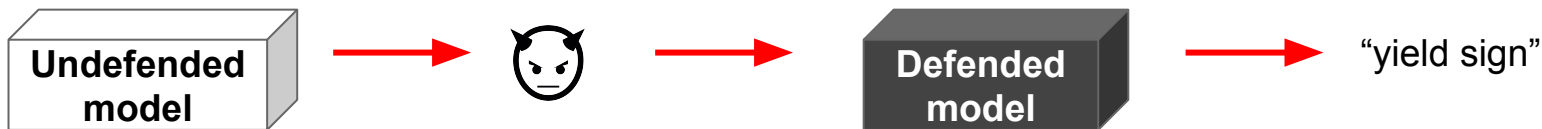
- (3) The adversary selects new synthetic inputs for queries to the remote ML system based on the local substitute's output surface sensitivity to input variations.

Attacking remotely hosted black-box models



(4) The adversary then uses the local substitute to craft adversarial examples, which are misclassified by the remote ML system because of transferability.

Attacking with transferability



(4) The adversary then uses the local substitute to craft adversarial examples, which are misclassified by the remote ML system because of transferability.

Attacking Adversarial Training with Transferability Demo



How to test your model for adversarial examples?

White-box attacks

- One shot

`FastGradientMethod`

- Iterative/Optimization-based

`BasicIterativeMethod, CarliniWagnerL2`

Transferability attacks

- Transfer from undefended
- Transfer from defended

Defenses

Adversarial training:

- Original variant
- Ensemble adversarial training
- Madry et al.

Reduce dimensionality of input space:

- Binarization of the inputs
- Thermometer-encoding

Adversarial examples represent
worst-case distribution drifts





Adversarial examples are a *tangible* instance of hypothetical AI safety problems

How to reach out to us?

Nicholas Carlini

nicholas@carlini.com

Nicolas Papernot

nicolas@papernot.fr