@NicolasPapernot

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`**?

November 2017 - ODSC

#ODSC

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?

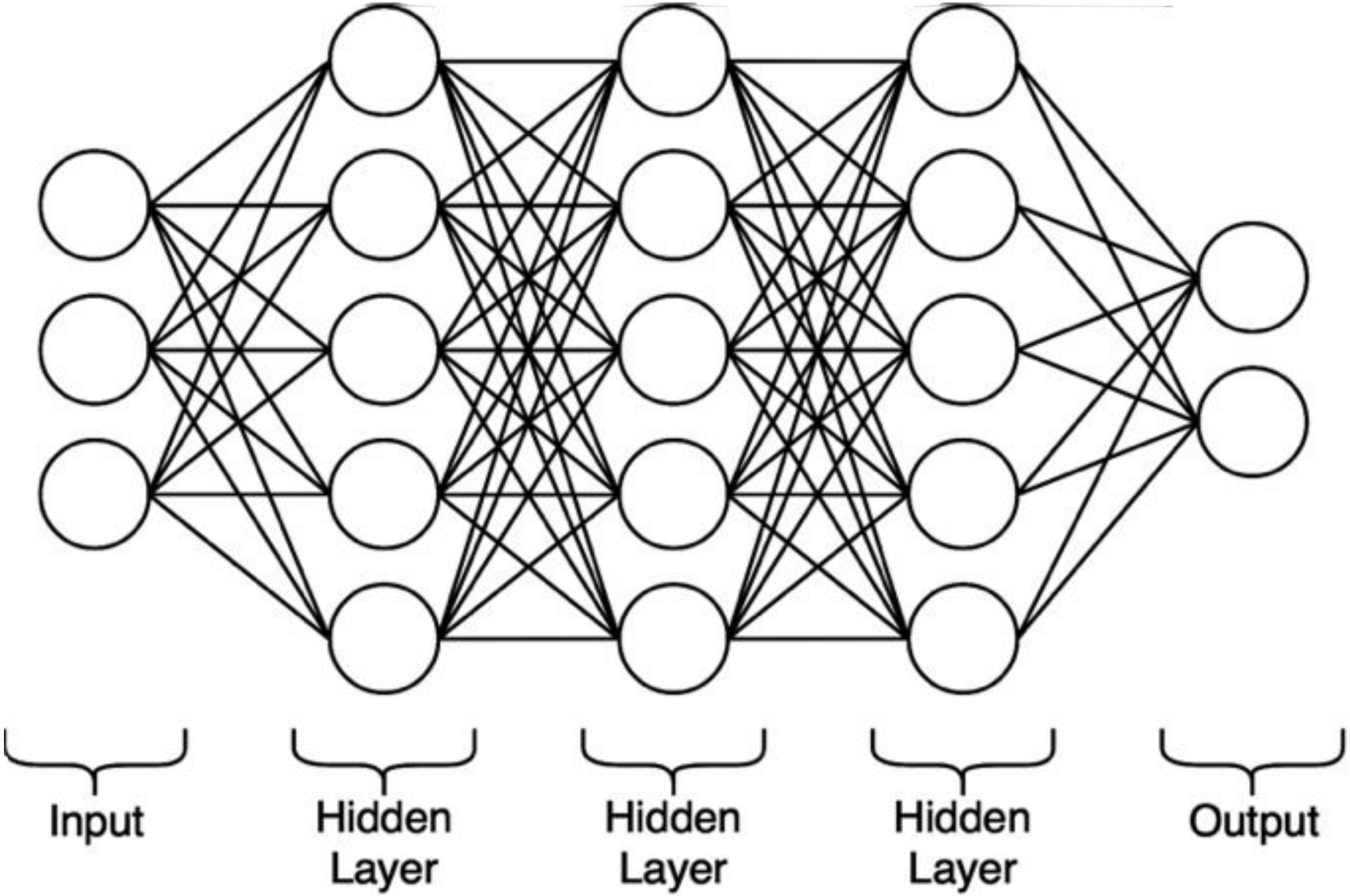
Classification with neural networks

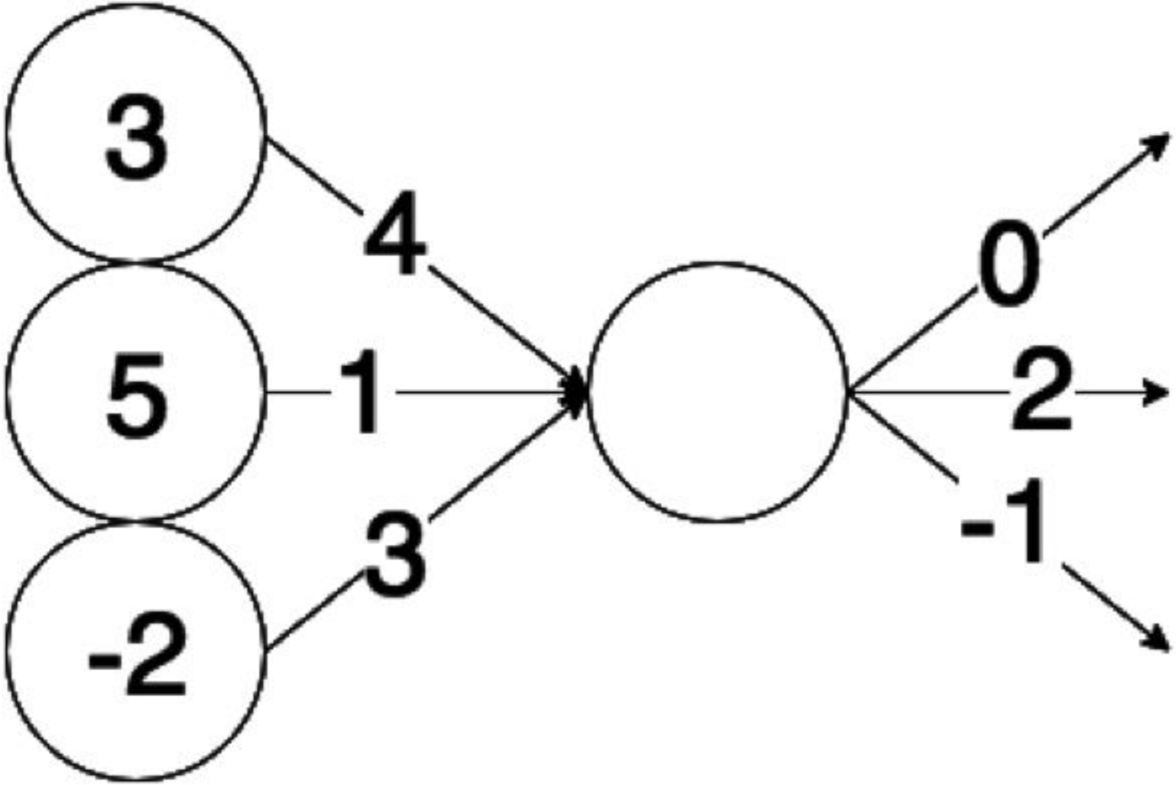
Machine Learning Classifier

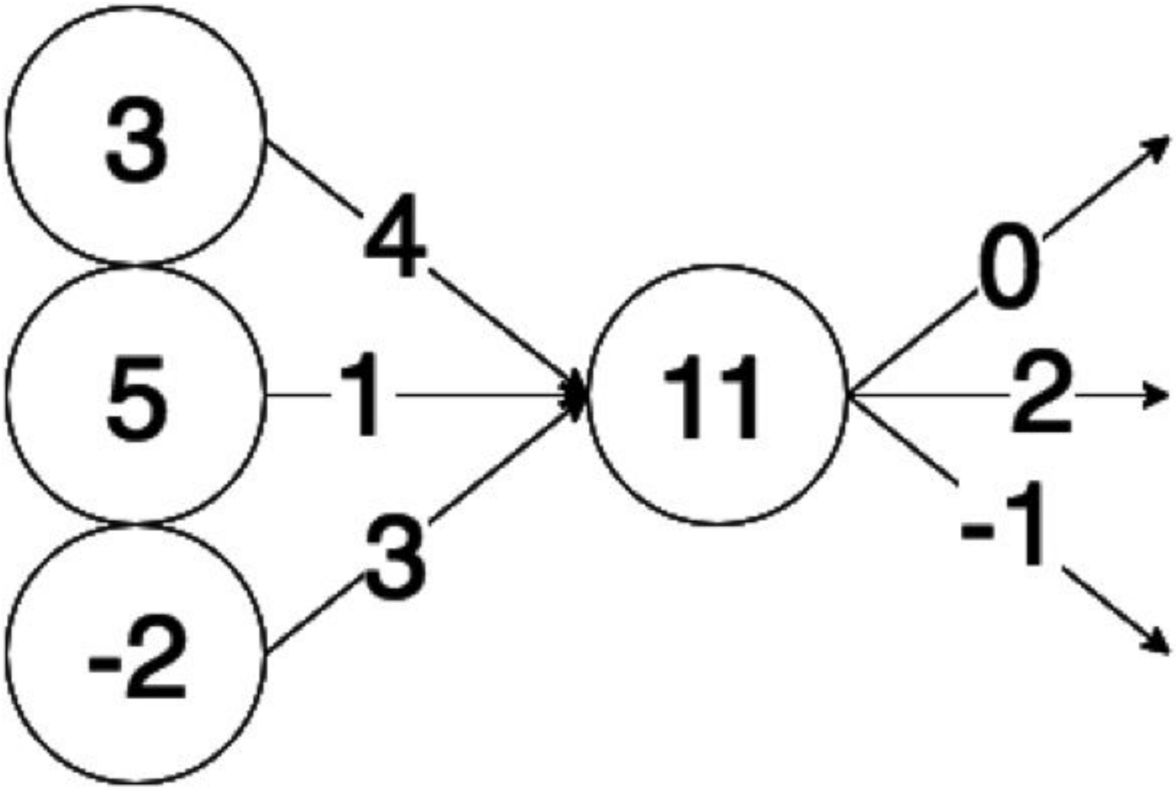
[0.01, **<u>0.84</u>**, 0.02, 0.01, 0.01, 0.01, 0.05, 0.01, 0.03, 0.01]

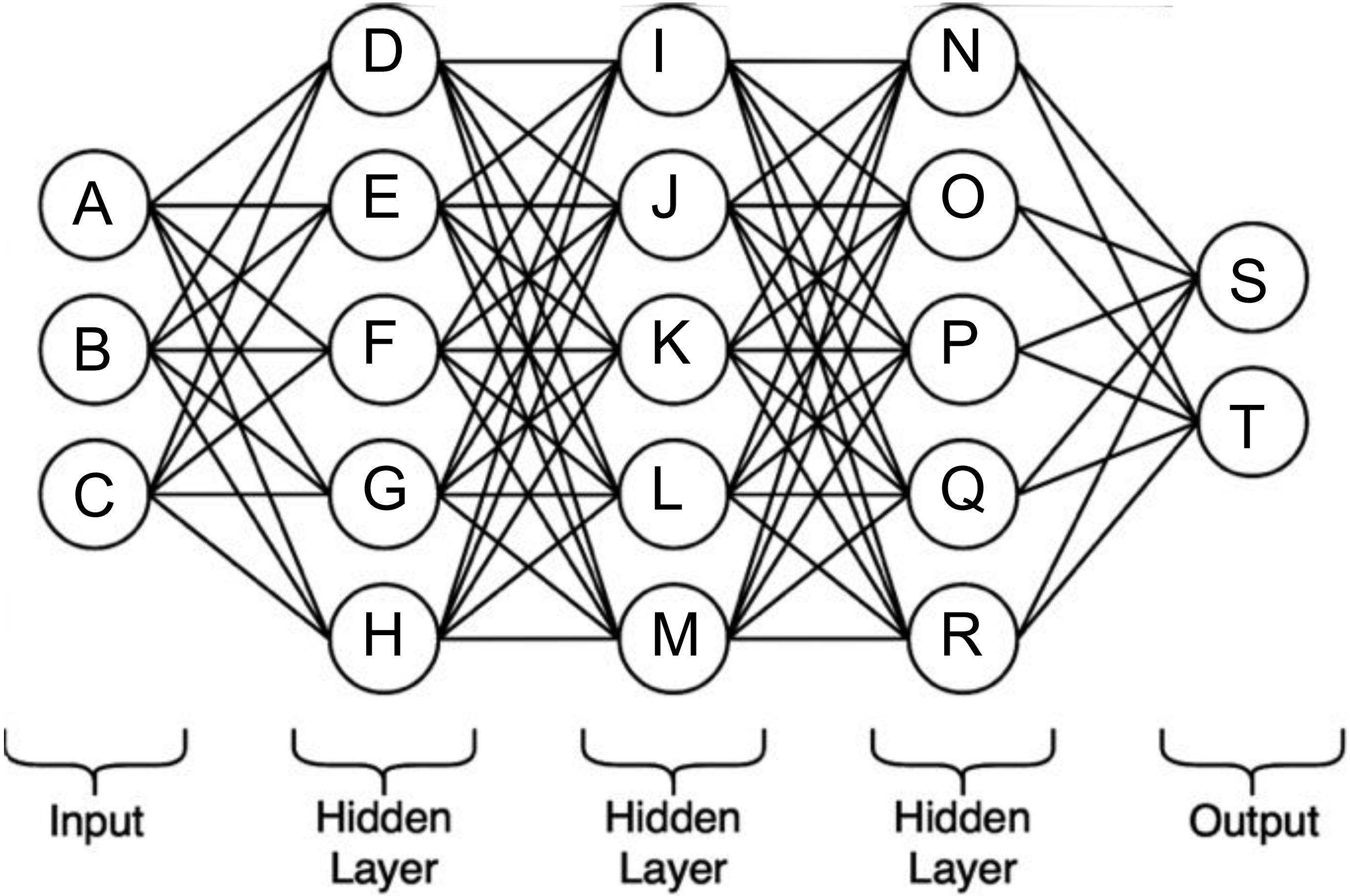
x → $f(x,\theta)$ → $[p(0|x,\theta), p(1|x,\theta), p(2|x,\theta), \ldots, p(7|x,\theta), p(8|x,\theta), p(9|x,\theta)]$

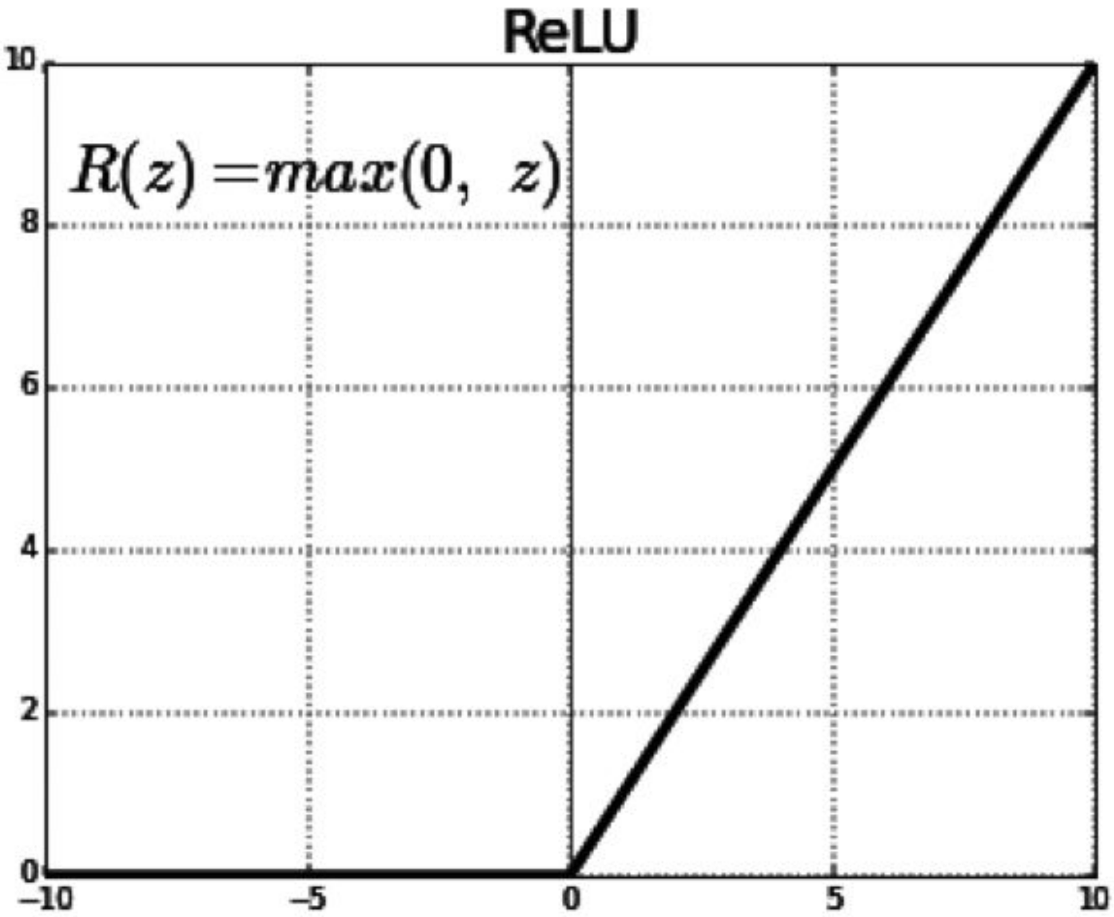
Classifier: map inputs to one class among a predefined set

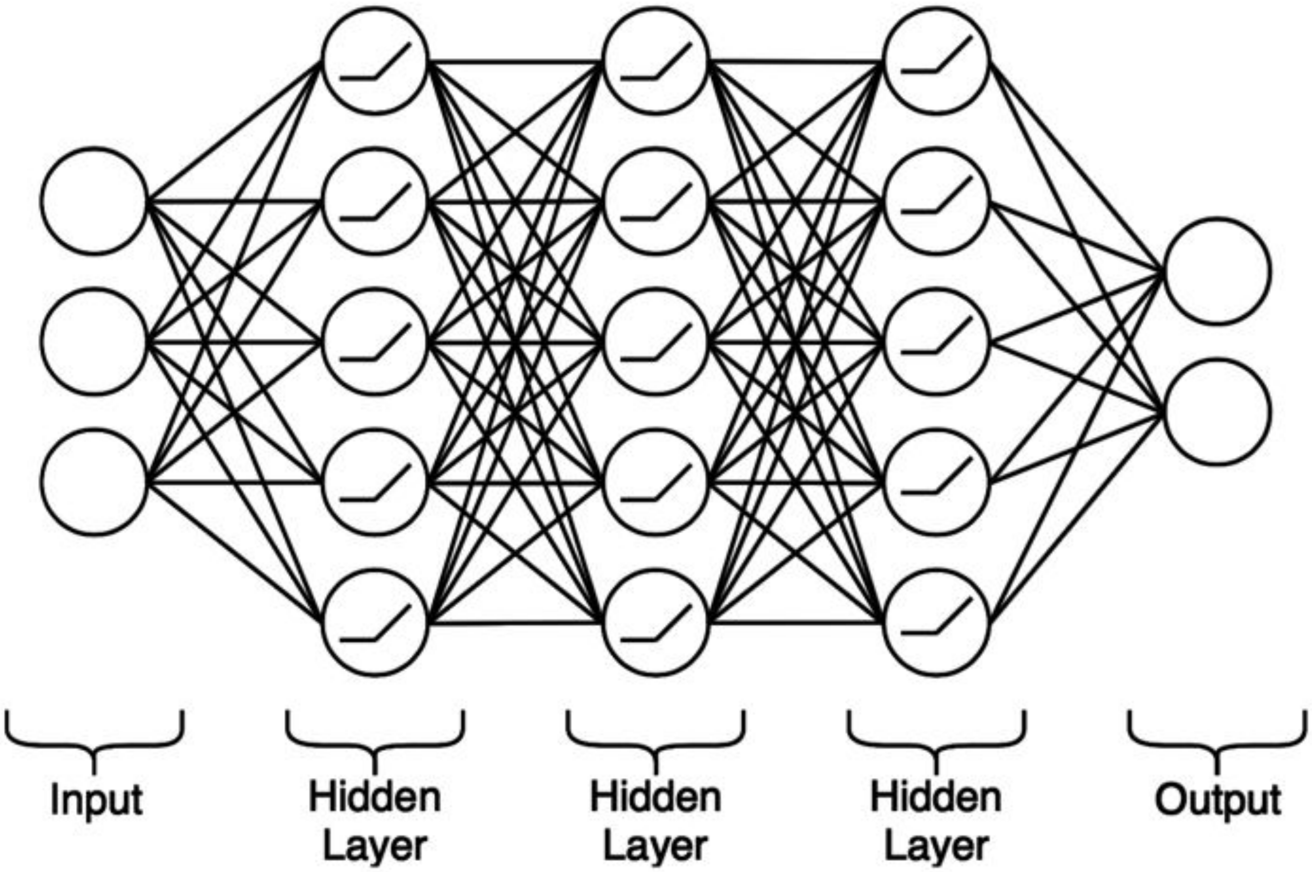

Input
Hidden Layer
Hidden Layer
Hidden Layer
Output

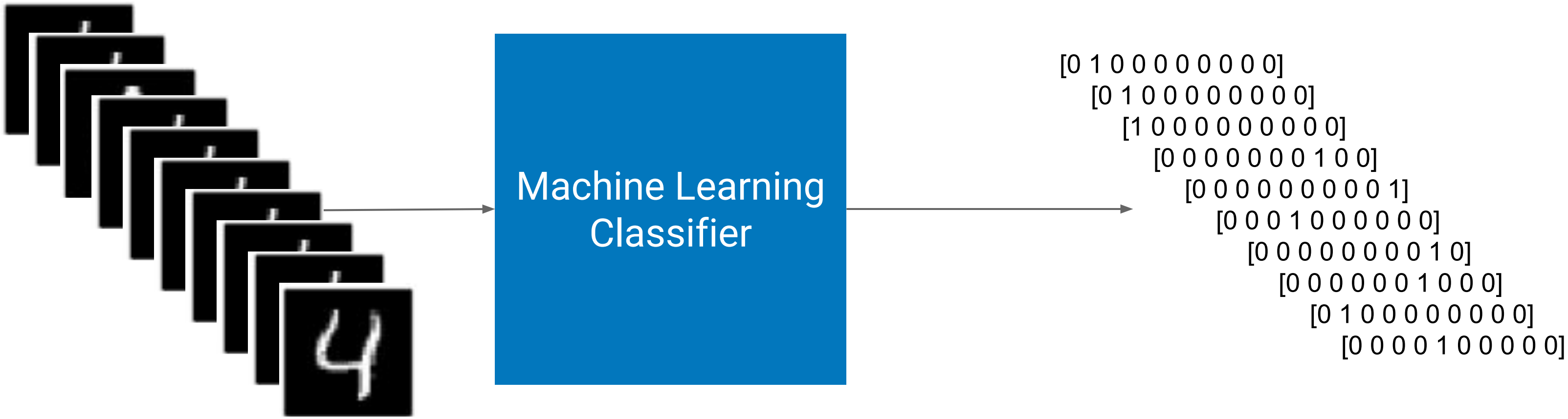

3
5
-2
4
1
3
0
2
-1


3
5
-2
4
1
3
11
0
2
-1


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
Input
Hidden Layer
Hidden Layer
Hidden Layer
Output


ReLU
R(z) = max(0, z)
10
8
6
4
2
0
−10
−5
0
5
10


Input
Hidden Layer
Hidden Layer
Hidden Layer
Output

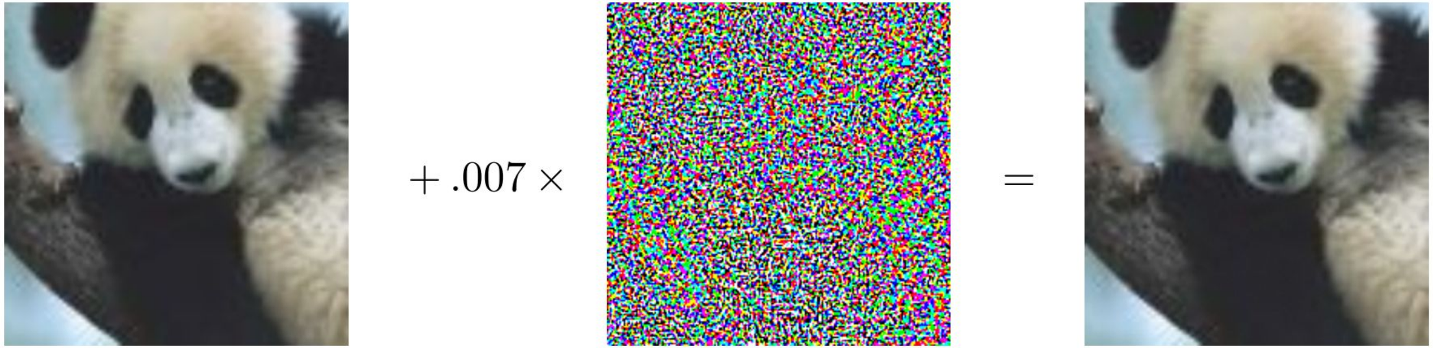


Learning: find internal classifier parameters θ that minimize a cost/loss function (~model error)

NNs give better results than any other approach

But there’s a catch ...

Adversarial examples

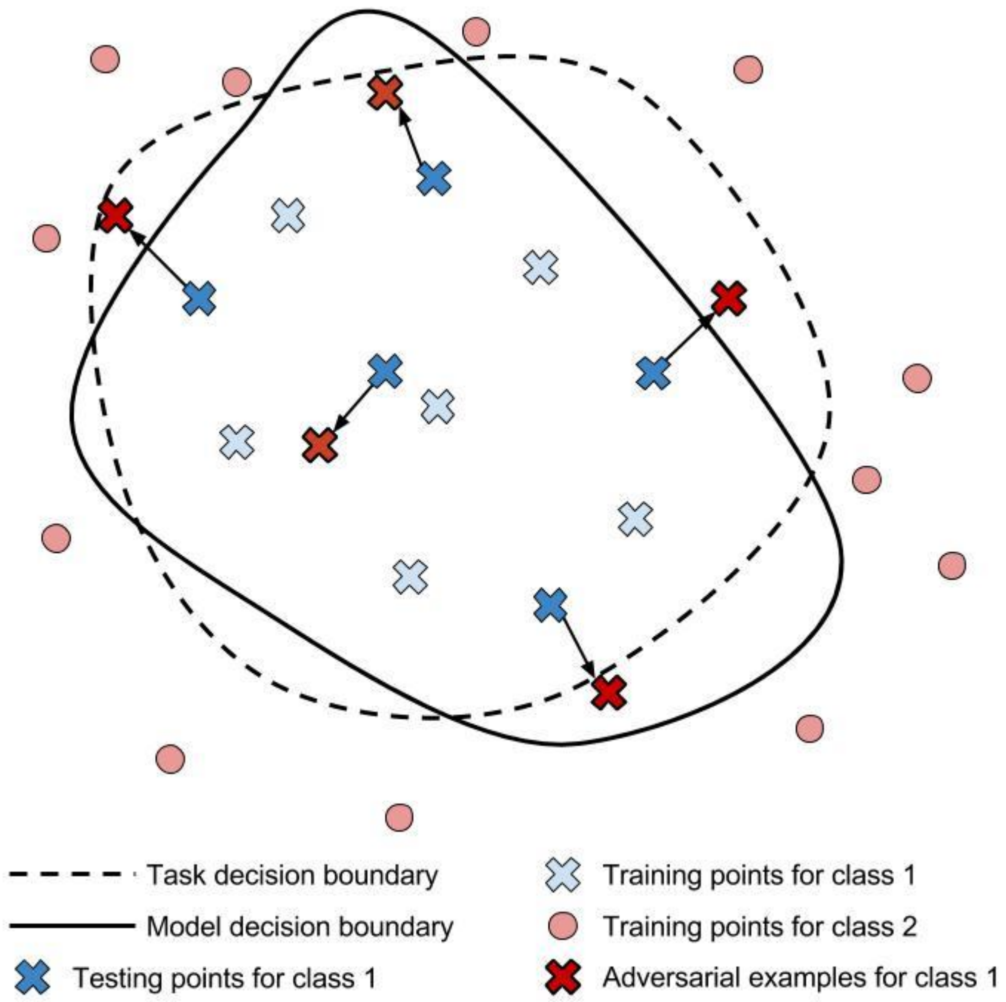


"panda"
57.7% confidence

"nematode"
8.2% confidence

"gibbon"
99.3 % confidence

[GSS15] Goodfellow et al. Explaining and Harnessing Adversarial Examples


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

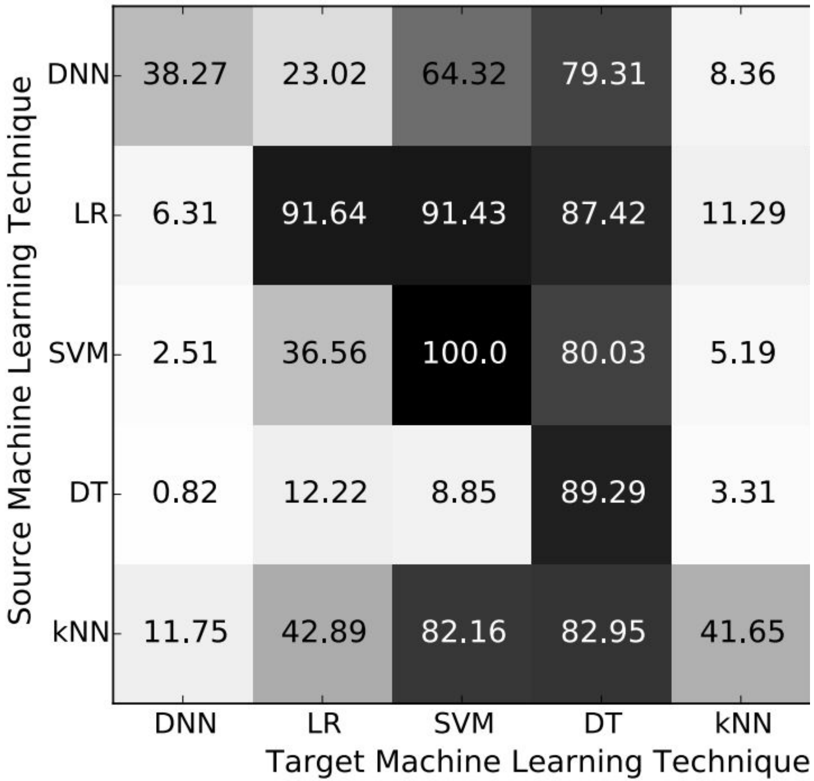
$$\nabla_x J(\theta, x, y)$$

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

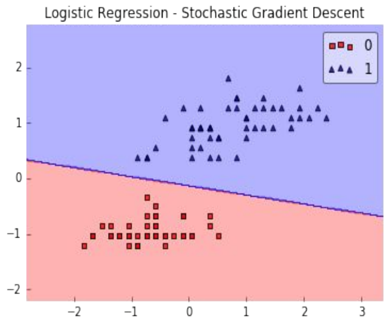
$$x + \varepsilon \cdot sgn(\nabla_x J(\theta, x, y))$$

[GSS15] Goodfellow et al. Explaining and Harnessing Adversarial Examples

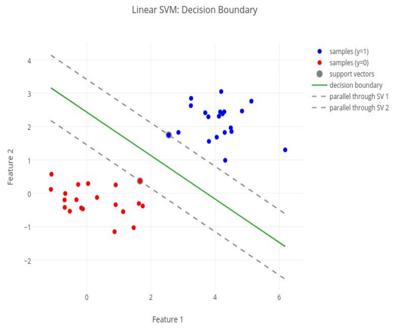
Transferability



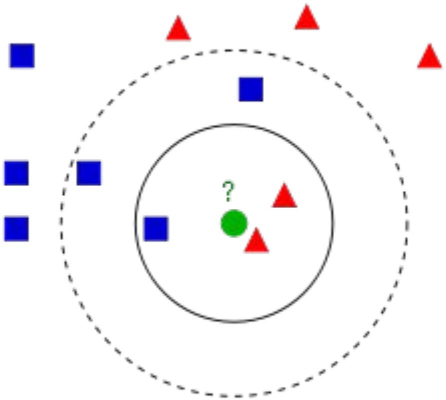
Not specific to neural networks



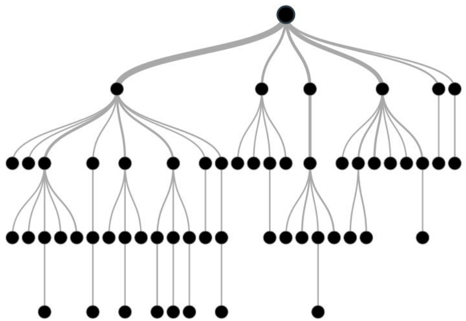
Logistic regression



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

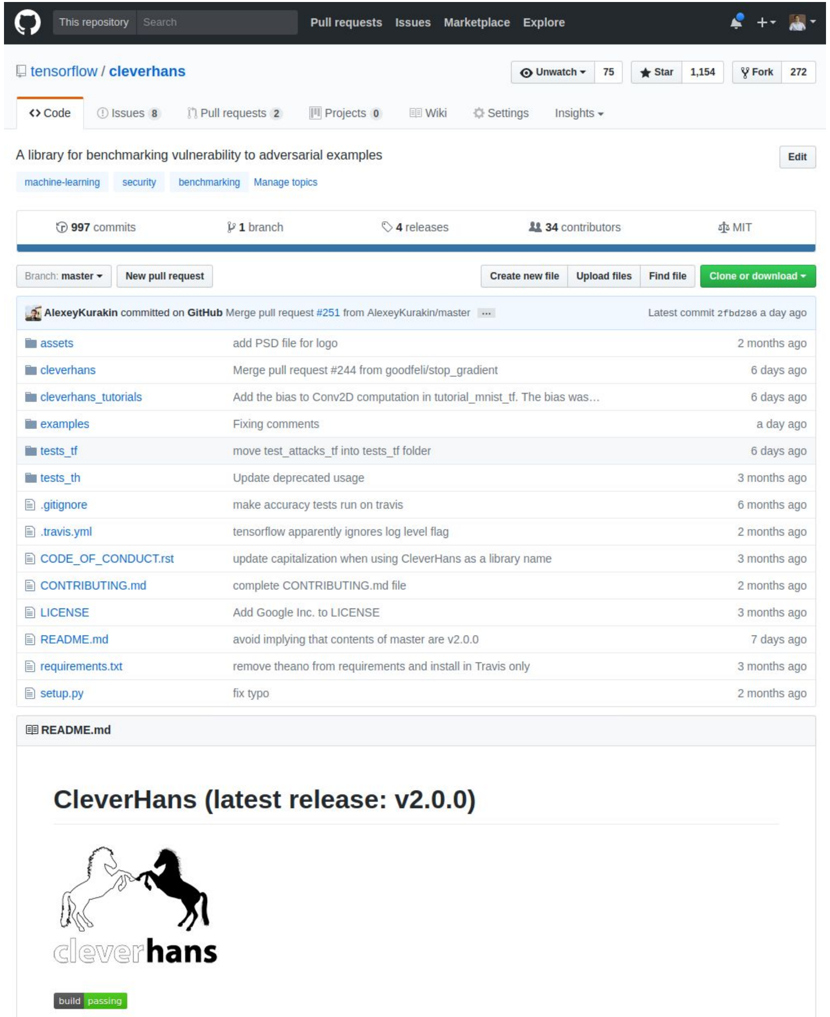
```
import tensorflow as tf

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



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
```

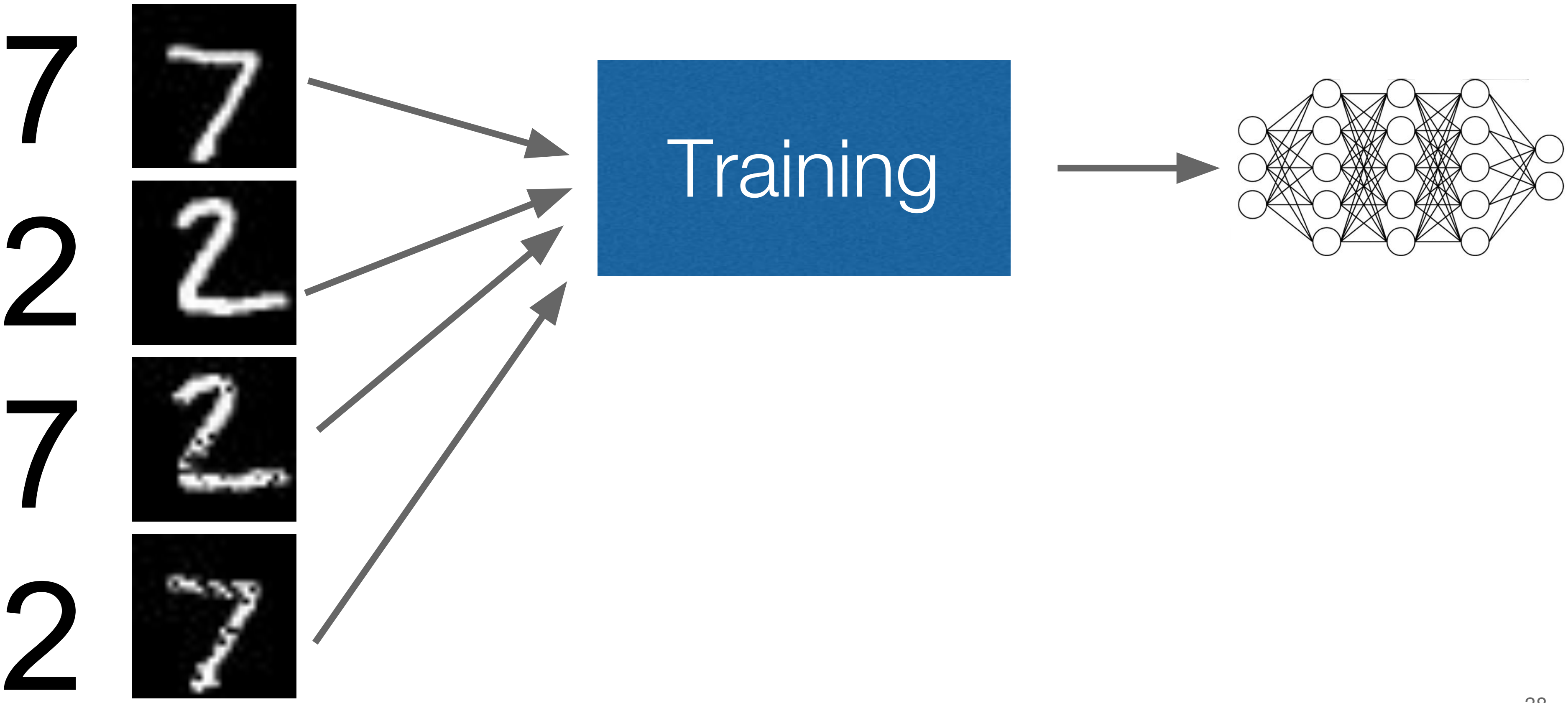
Things to try:

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

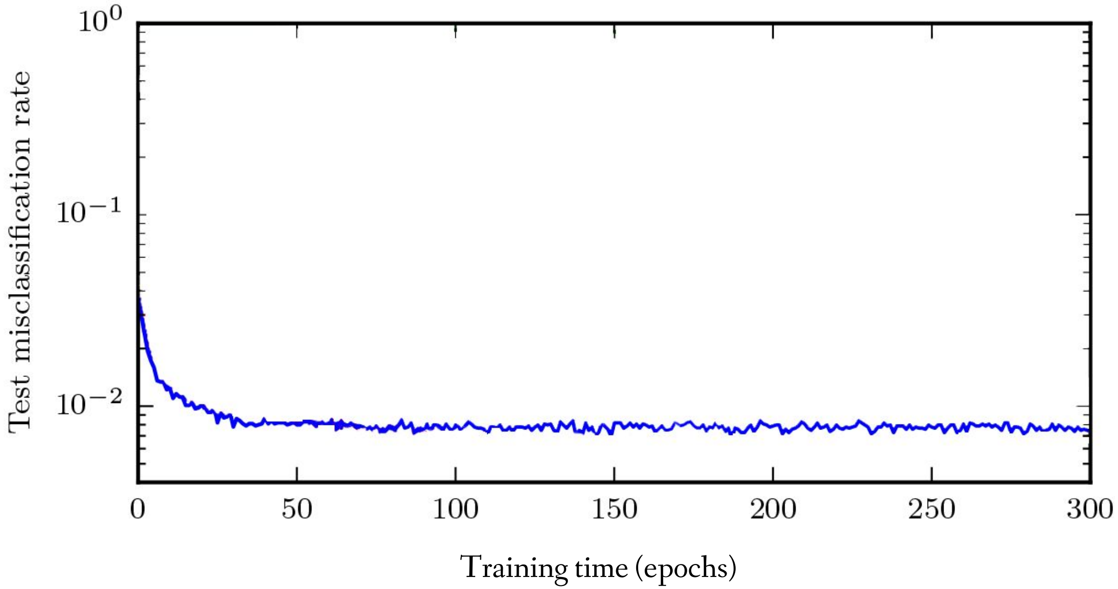


Figure by Ian Goodfellow

Adversarial training

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

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

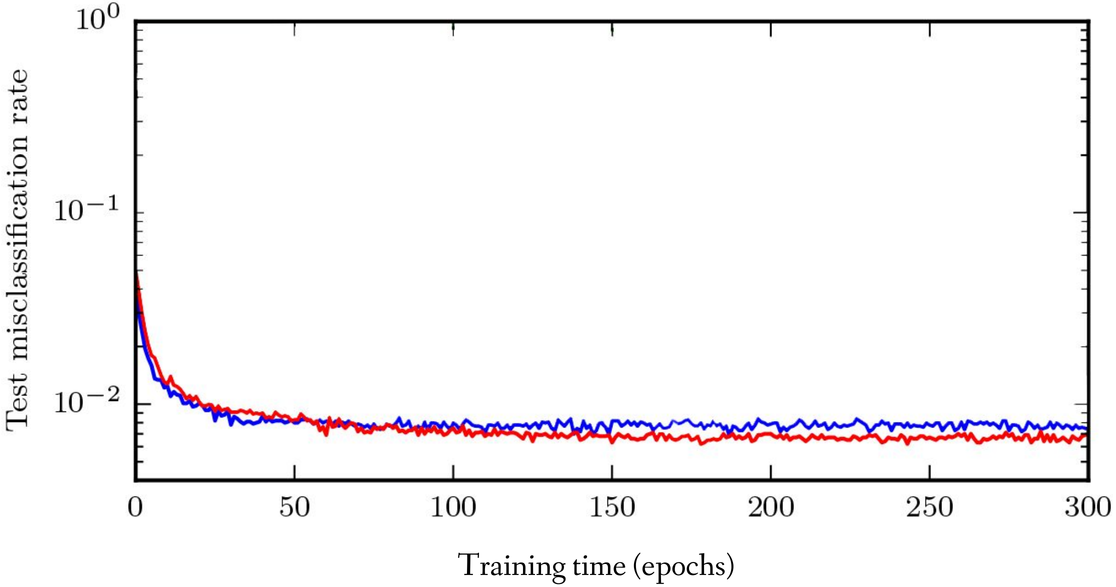


Figure by Ian Goodfellow

Adversarial training

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

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

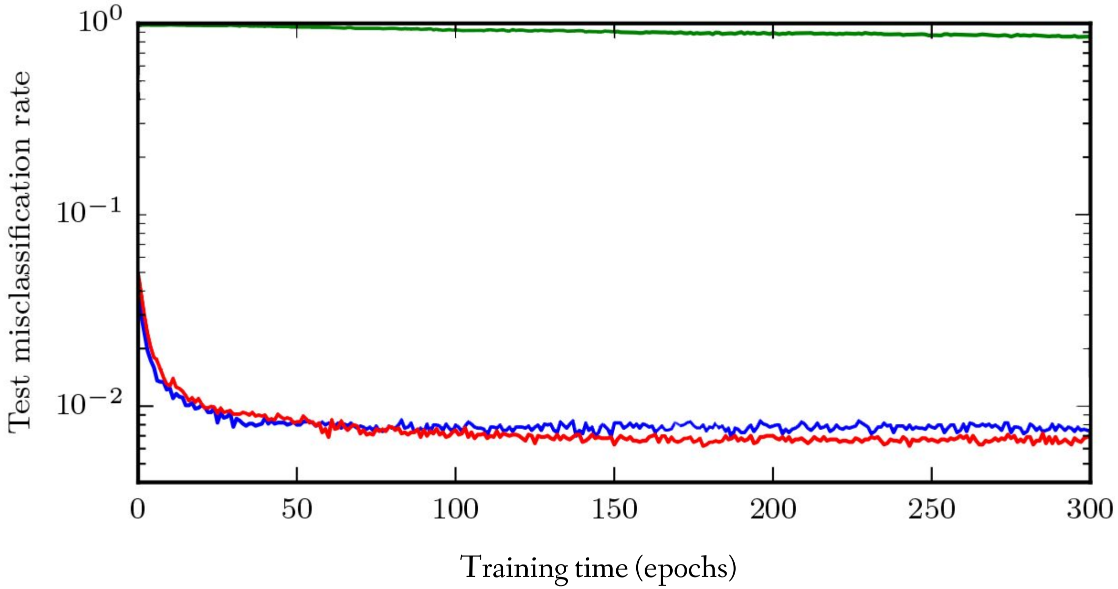


Figure by Ian Goodfellow

Adversarial training

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

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

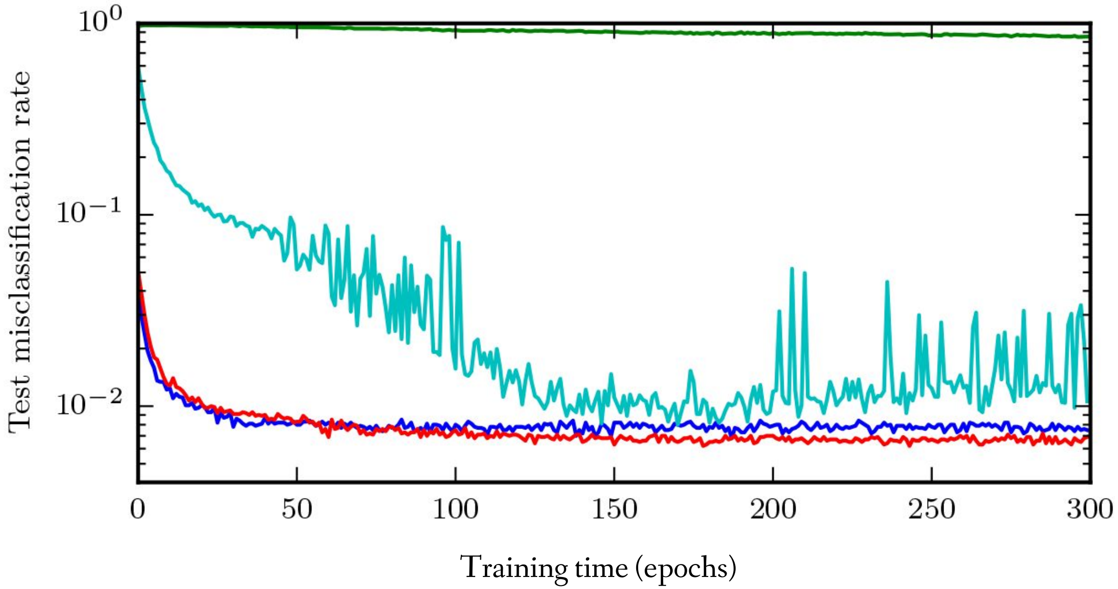


Figure by Ian Goodfellow

Efficient Adversarial Training through Loss Modification

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

Small when prediction is
correct on legitimate input

Efficient Adversarial Training through Loss Modification

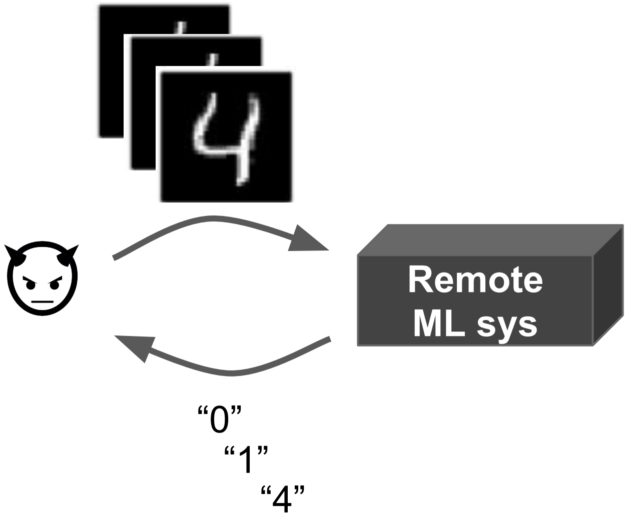
$$\mathrm{loss}(x, y) + \mathrm{loss}(x + \epsilon \cdot \mathtt{sign}(\mathrm{grad}), y)$$

Small when prediction is correct on legitimate input

Small when prediction is correct on adversarial input

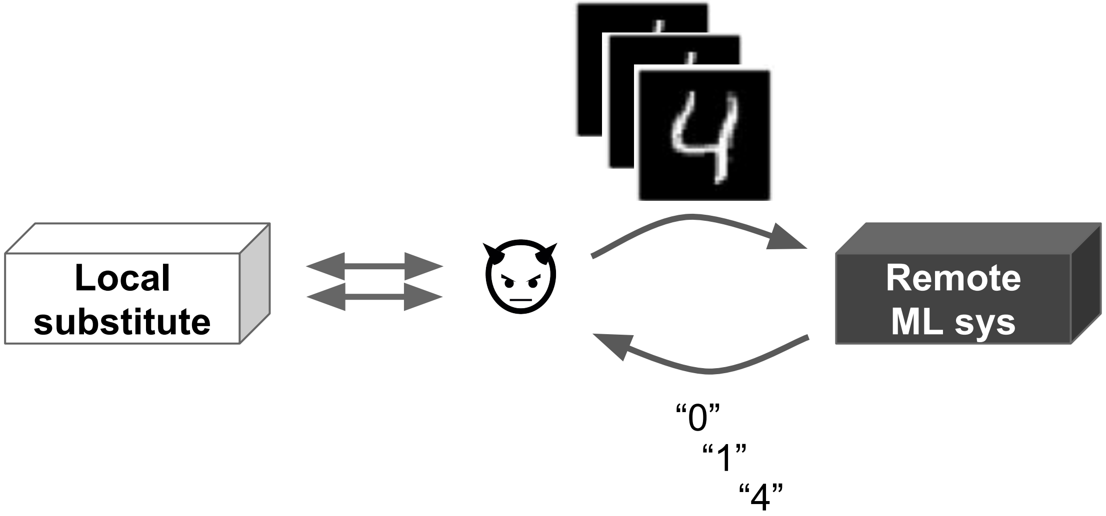
Adversarial Training Demo

Attacking remotely hosted black-box models



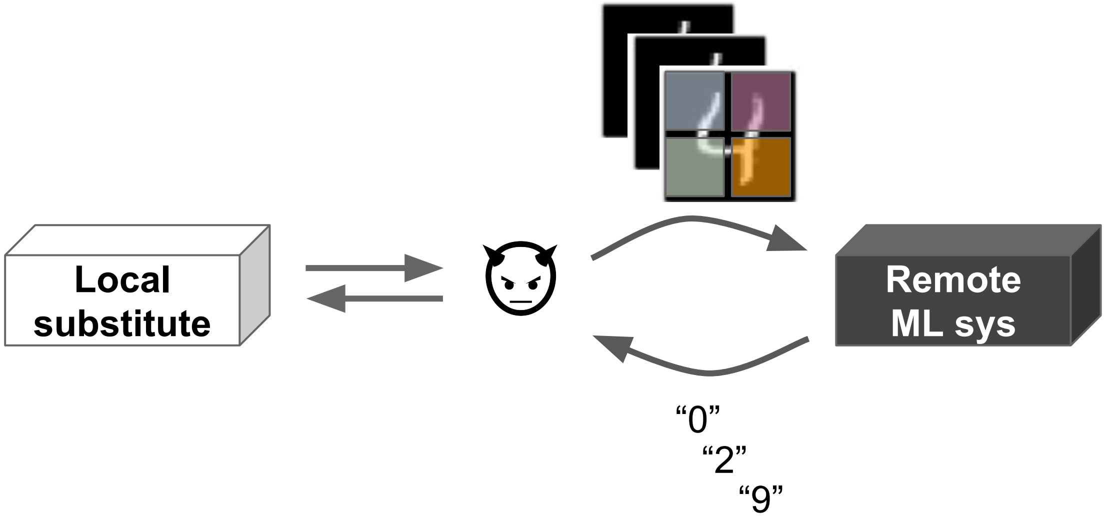
(1) The adversary queries remote ML system for labels on inputs of its choice.

Attacking remotely hosted black-box models



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

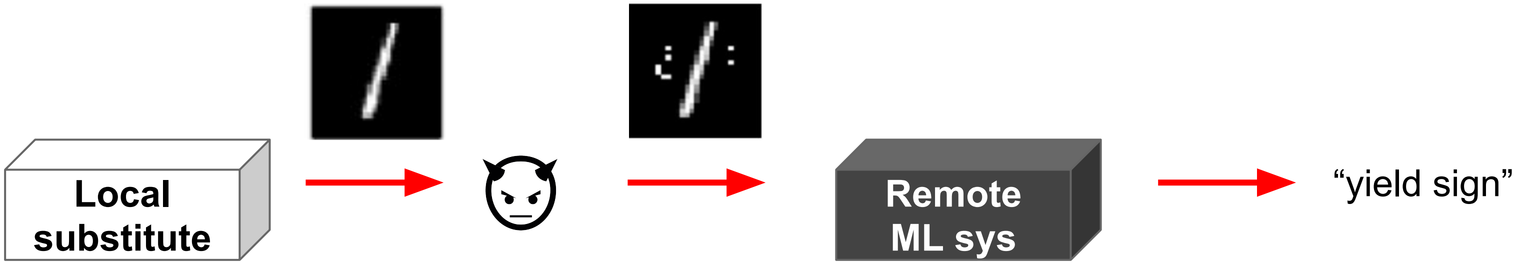
Attacking remotely hosted black-box models



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

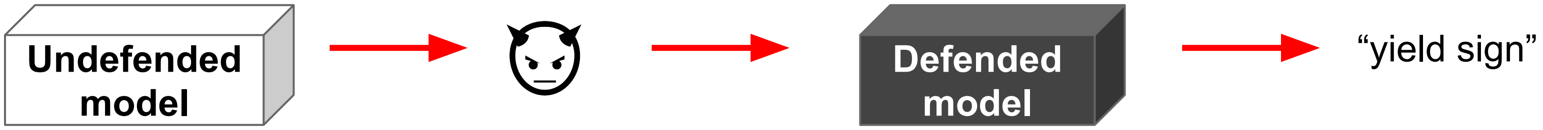
(3) The adversary selects new synthetic inputs for queries to the remote ML system based on the local substitute’s output surface sensitivity to input variations.

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

[DDS04] Dalvi et al. *Adversarial Classification* (KDD)

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

Image source: http://www.nerdist.com/wp-content/uploads/2013/07/Space-Odyssey-4.jpg

How to reach out to us?

Nicholas Carlini nicholas@carlini.com

Nicolas Papernot nicolas@papernot.fr