Adversarial AI & Adversarial Robustness Toolbox

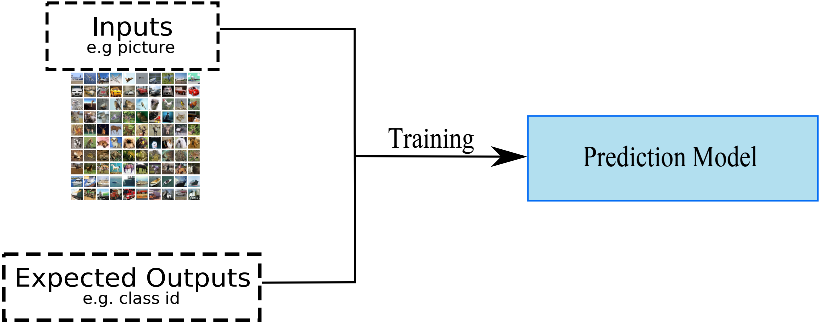
Irina Nicolae
AI & Machine Learning
IBM Research Ireland

May 31, 2018

Evasion Attacks Against Machine Learning

Machine Learning

Training



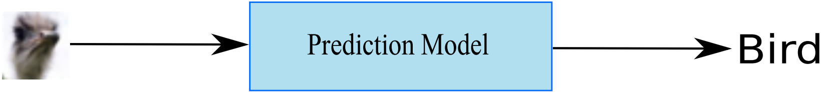
Inputs
e.g picture

Training

Prediction Model

Expected Outputs
e.g. class id

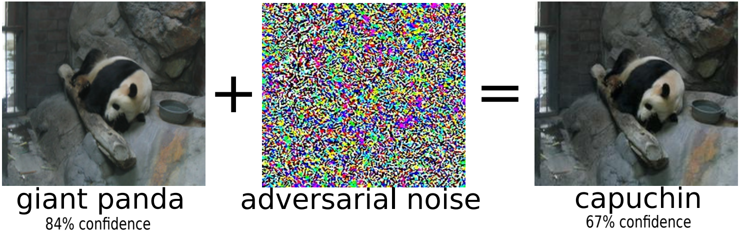
Prediction



Prediction Model

Bird

Adversarial Examples

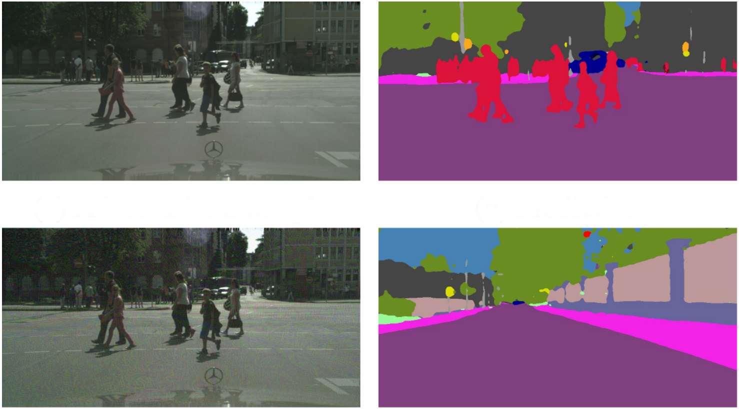


- Perturb model inputs with crafted noise
- Model fails to recognize input correctly
- Attack undetectable by humans
- Random noise does not work.

Self-Driving Cars

Image segmentation[1]

Attack noise hides pedestrians from the detection system.



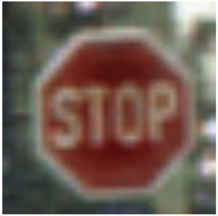
[1] Metzen et al., *Universal Adversarial Perturbations Against Semantic Image Segmentation*. `https://arxiv.org/abs/1704.05712`.

Self-Driving Cars

Road signs[2]

Car ends up ignoring the stop sign.

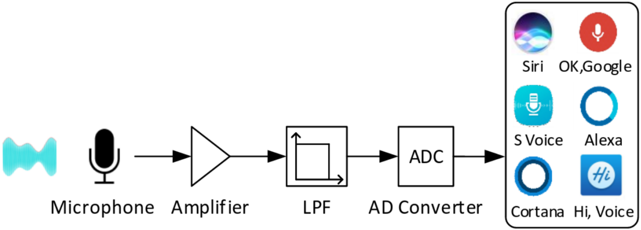
True image



Adversarial image

[2] McDaniel et al., *Machine Learning in Adversarial Settings*. IEEE Security and Privacy, vol. 14, pp. 68-72, 2016.

Executing Voice Commands

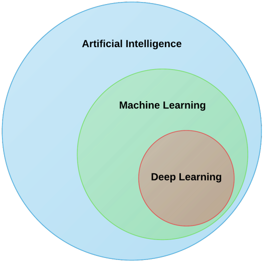


Okay Google, text John![3]

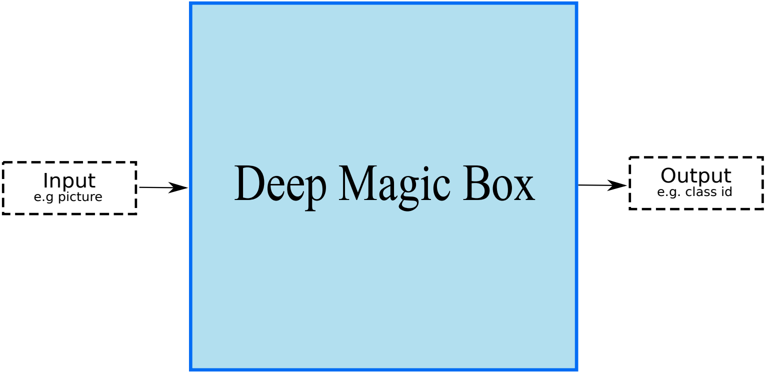
- Stealthy voice commands recognized by devices
- Humans cannot detect it.

[3] Zhang et al., *DolphinAttack: Inaudible Voice Commands*, ACM CCS 2017.

Deep Learning and Adversarial Samples



Deep Neural Networks

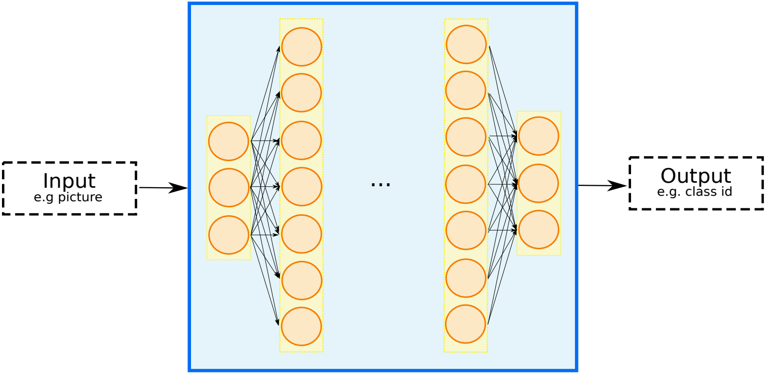


Input
e.g picture

Deep Magic Box

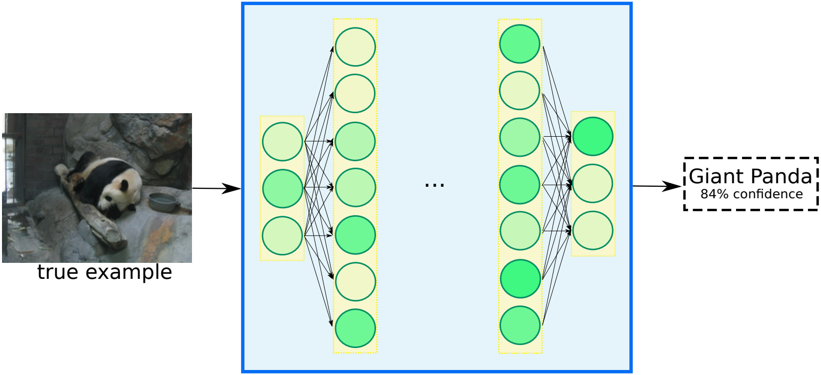
Output
e.g. class id

Deep Neural Networks



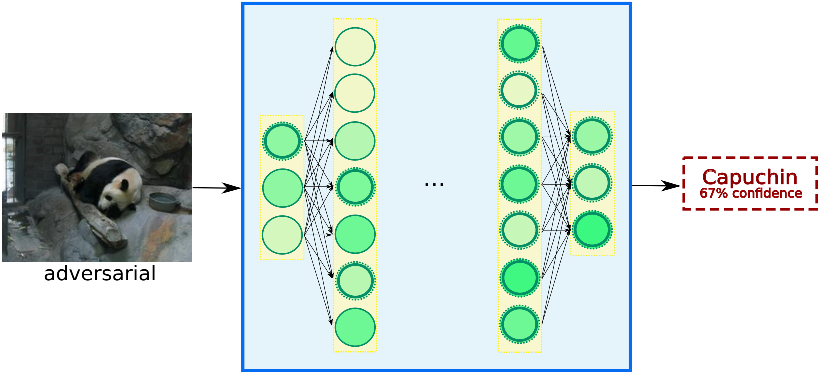
- Interconnected layers propagate the information forward.
- Model learns weights for each neuron.

Deep Neural Networks



- Specific neurons light-up depending on the input.
- Cumulative effect of activation moves forward in the layers.

Deep Neural Networks

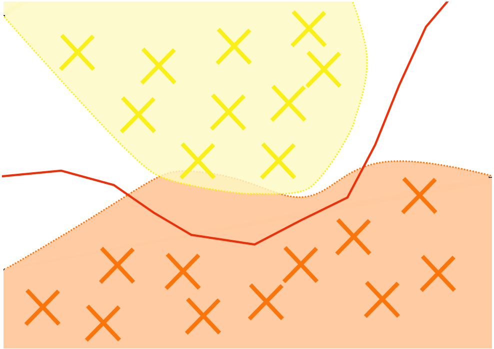


Small variations in the input → important changes in the output.

+ Enhanced discriminative capacities

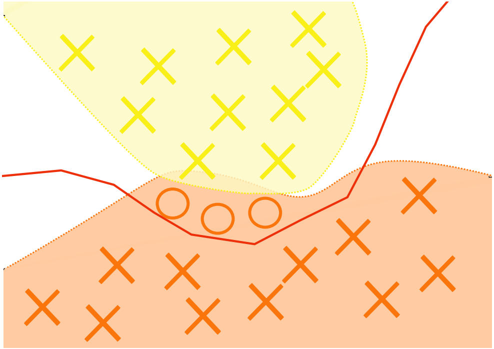
– Opens the door to adversarial examples

Decision Boundary of the Model



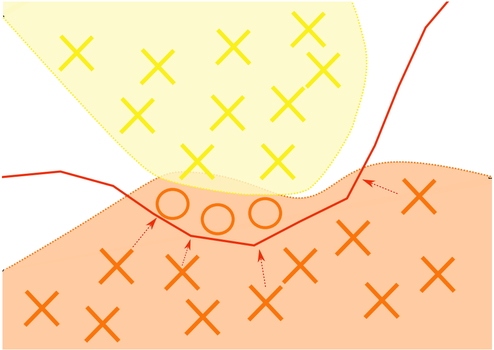
The **learned model** slightly differs from the **true** data distribution...

The Space of Adversarial Examples



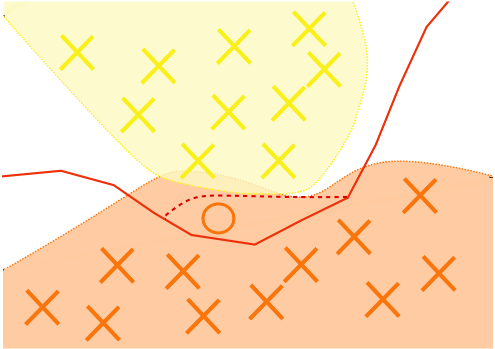
... which makes room for **adversarial examples**.

Attack: Use the Adversarial Directions



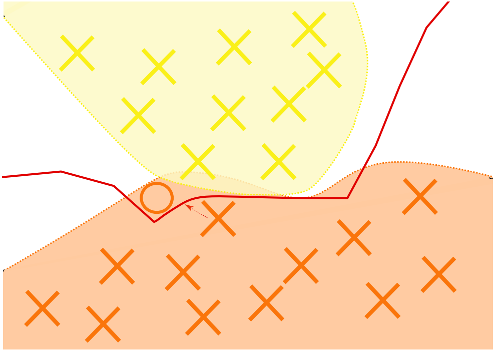
- Most attacks try to move inputs across the boundary.
- Attacking with a random distortion doesn't work well in practice.

Defense: Adversarial Training



- Adapt the classifier to attack directions by including adversarial data at training.

Defense: Adversarial Training



- Adapt the classifier to attack directions by including adversarial data at training.
- But there are always new adversarial samples to be crafted.

The Adversarial Robustness Toolbox

Adversarial Robustness Toolbox (ART)

`https://github.com/IBM/adversarial-robustness-toolbox`

- Python library
- Evasion attacks, defenses, detection, robustness metrics
- Framework-agnostic
- Focus on image data
- Target users
  - Researchers → rapid prototyping
  - Developers → adversarial robustness services
- Open-source release at RSA 2018

Supported Methods

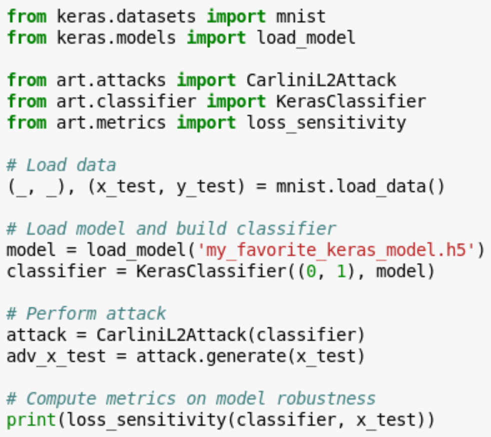
Attacks	Defenses
DeepFool	Feature Squeezing
Fast Gradient Method	Spatial Smoothing
Jacobian Saliency Map	Label Smoothing
NewtonFool	Adversarial Training
Universal Perturbation	Virtual Adversarial Training
C&W Attack	Gaussian Augmentation
Virtual Adversarial Method	

Frameworks	Metrics
TensorFlow	Loss sensitivity
Keras	Empirical robustness
PyTorch (soon)	CLEVER
MXNet (soon)	

Competitor Analysis

	CleverHans	FoolBox	Nemesis
Release date	Sept 16, 2016	June 4, 2017	March 25, 2018
Affiliation	Open AI, Google	Tubingen U.	IBM Research
GitHub org	tensorflow	bethgelab	IBM
GitHub metrics	1927 stars, 503 forks	492 stars, 83 forks	229 stars, 59 forks
Features			
Attacks	✓	✓	✓
Defenses	✗	✗	✓
Detection	✗	✗	in progress
Robustness metrics	✗	✗	✓
Fwk-agnostic	✗	✓	✓
Other data types	✗	✗	planned

Example



```
from keras.datasets import mnist
from keras.models import load_model

from art.attacks import CarliniL2Attack
from art.classifier import KerasClassifier
from art.metrics import loss_sensitivity

# Load data
(_, _), (x_test, y_test) = mnist.load_data()

# Load model and build classifier
model = load_model('my_favorite_keras_model.h5')
classifier = KerasClassifier((0, 1), model)

# Perform attack
attack = CarliniL2Attack(classifier)
adv_x_test = attack.generate(x_test)

# Compute metrics on model robustness
print(loss_sensitivity(classifier, x_test))
```

Conclusion

- The problem of adversarial examples needs to be solved before applying machine learning.
- The arms race for attacks and defenses continues.

Getting started with ART

- Code `https://github.com/IBM/adversarial-robustness-toolbox`
- Documentation `https://adversarial-robustness-toolbox.readthedocs.io`