Pluribus One
seeing one in many

Pattern Recognition and Applications Lab

University of Cagliari, Italy

Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning

Battista Biggio

* Slides from this talk are inspired from the tutorial I prepared with *Fabio Roli* on such topic.
https://www.pluribus-one.it/sec-ml/wild-patterns/

Winter School on Quantitative Systems Biology: Learning and Artificial Intelligence, Nov. 15-16, Trieste, Italy

A Question to Start…

What is the oldest survey article on machine learning that you have ever read?

What is the publication year?

This Is Mine… Year 1966

Pattern Recognition

By DENIS RUTOVITZ

Medical Research Council

[Read before the ROYAL STATISTICAL SOCIETY on Wednesday, May 18th, 1966, the President, Mr L. H. C. TIPPETT, in the Chair]

1. INTRODUCTION

DURING the past 10 years about 200 articles and several books have appeared, dealing with machine recognition of optical and other patterns (mainly alphabetic characters and numerals). About half of these have described methods not linked to a specific

Credits: Dr Gavin Brown for showing me this article

Applications in the Old Good Days…

What applications do you think that this paper dealt with?

Pattern Recognition

By Denis Rutovitz

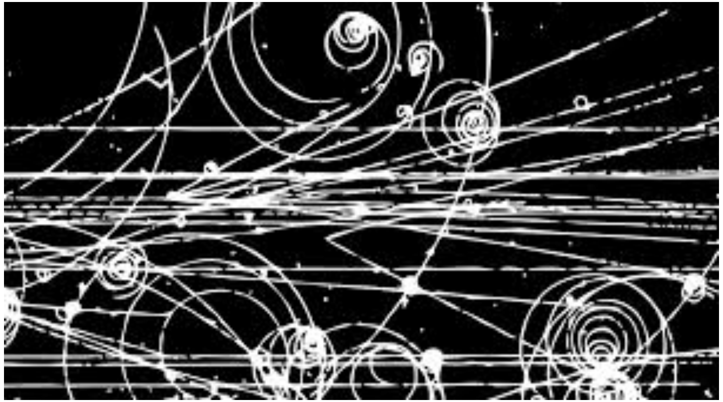
Medical Research Council

[Read before the Royal Statistical Society on Wednesday, May 18th, 1966,
the President, Mr L. H. C. Tippett, in the Chair]

Popular Applications in the Sixties

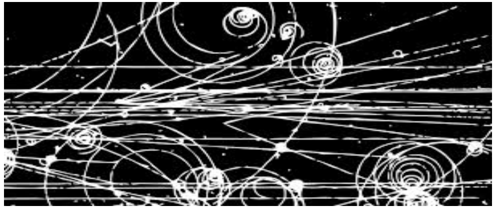
OCR for bank cheque sorting

Aerial photo recognition



Detection of particle tracks in bubble chambers

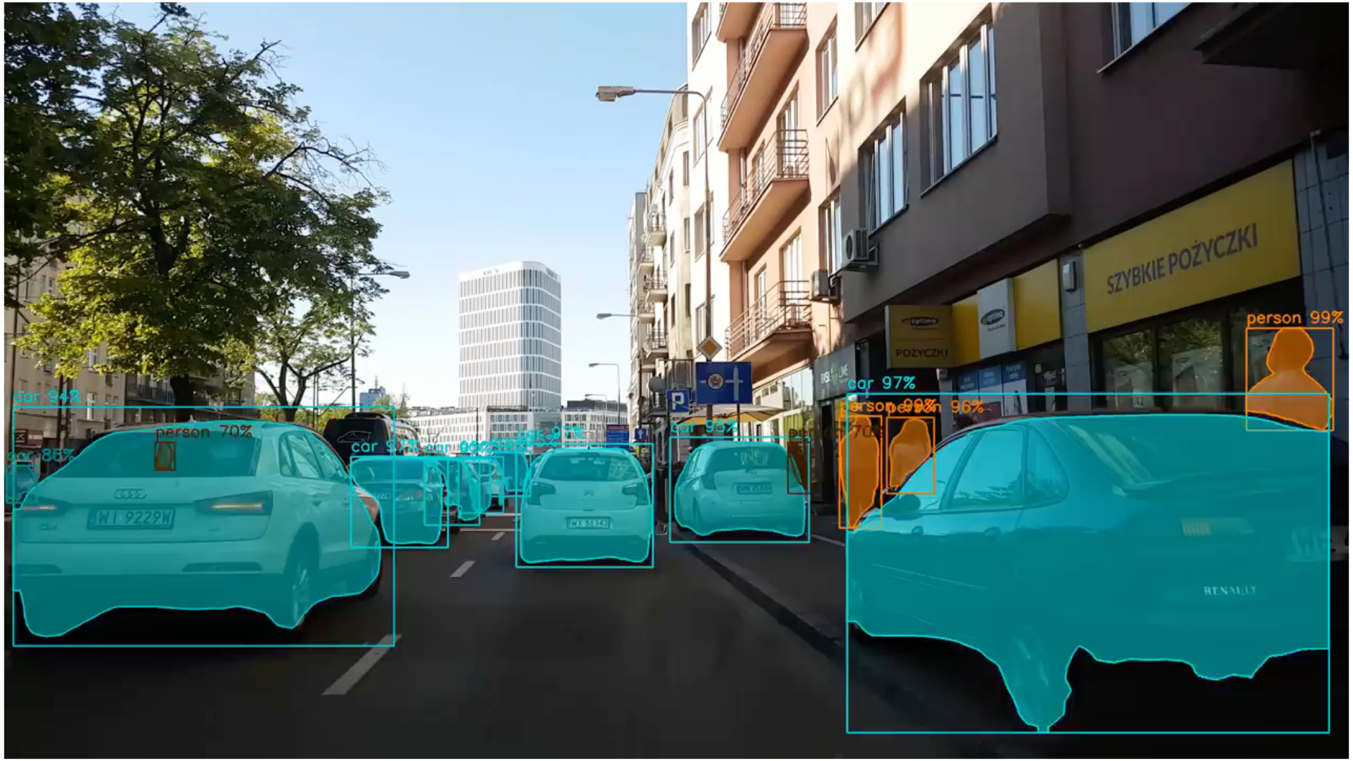
Key Feature of these Apps



Specialised applications for **professional** users…

How About Today's Applications of AI?

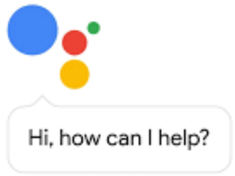
Computer Vision for Self-Driving Cars



He et al., Mask R-CNN, ICCV '17, https://arxiv.org/abs/1703.06870
Video from: https://www.youtube.com/watch?v=OOT3UIXZztE

Automatic Speech Recognition for Virtual Assistants

- Amazon Alexa - https://developer.amazon.com/it/alexa-skills-kit/asr
- Apple Siri – https://machinelearning.apple.com/2017/10/01/hey-siri.html
- Microsoft Cortana - https://developer.microsoft.com/en-us/windows/speech
- Google Assistant - https://developers.google.com/assistant/sdk/



Today Applications of Machine Learning

Key Features of Today Apps

Personal and **consumer** applications…

We Are Living in the Best of the Worlds…

AI is going to transform industry and business as electricity did about a century ago

(Andrew Ng, Jan. 2017)

All Right? All Good?

iPhone 5s and 6s with Fingerprint Reader…

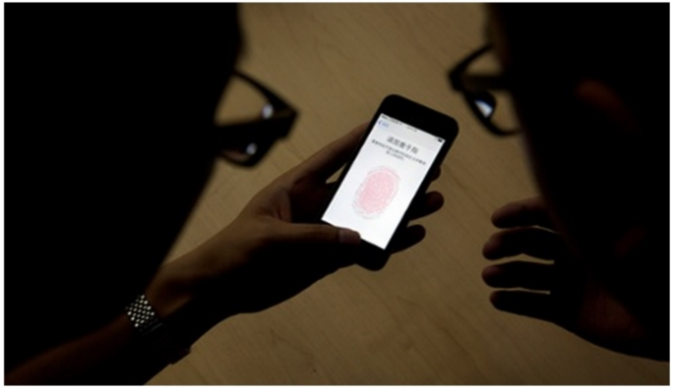
Hacked a Few Days After Release…

iPhone 5S fingerprint sensor hacked by Germany's Chaos Computer Club

Biometrics are not safe, says famous hacker team who provide video showing how they could use a fake fingerprint to bypass phone's security lockscreen

Follow Charles Arthur by email BETA

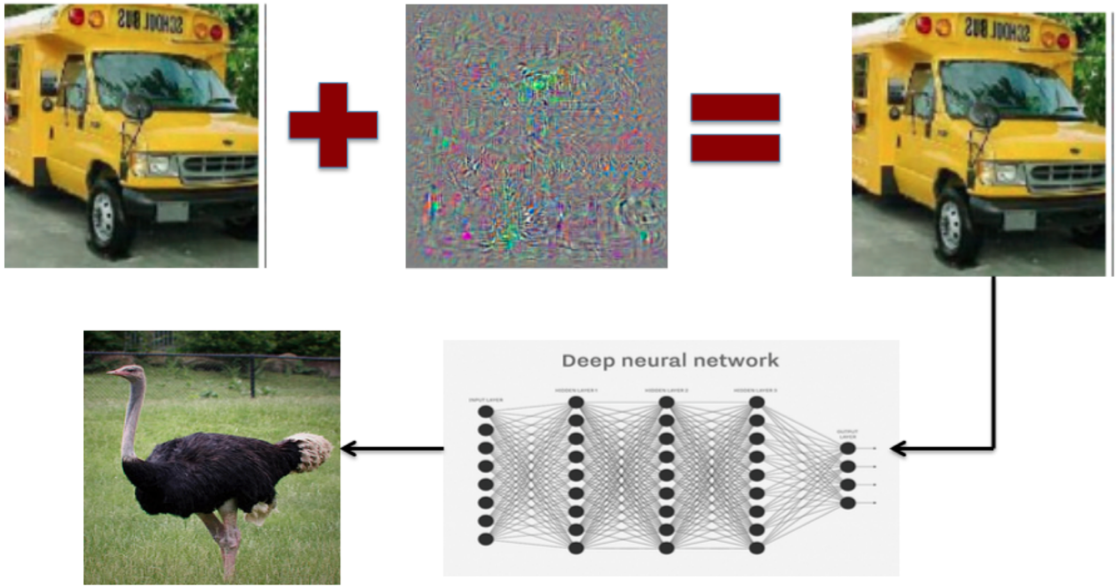
Charles Arthur
theguardian.com, Monday 23 September 2013 08.50 BST
Jump to comments (306)



But maybe this happens only for old, shallow machine learning…

End-to-end deep learning is another story…

Adversarial School Bus



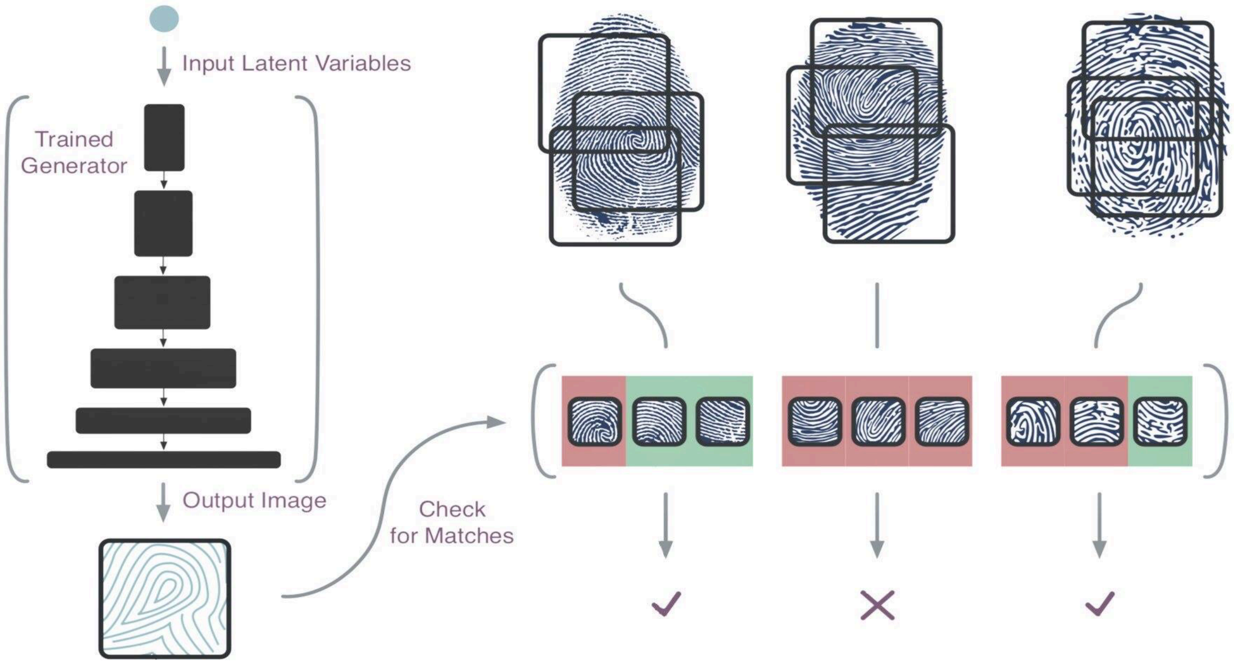
Szegedy et al., Intriguing properties of neural networks, **ICLR 2014**
Biggio, Roli et al., Evasion attacks against machine learning at test time, **ECML-PKDD 2013**

Adversarial Glasses

- M. Sharif et al. (ACM CCS 2016) attacked deep neural networks for face recognition with carefully-fabricated eyeglass frames
- When worn by a 41-year-old white male (left image), the glasses mislead the deep network into believing that the face belongs to the famous actress Milla Jovovich

Generating Master Keys for Fingerprints (AI *vs* AI...)

Generative Adversarial Networks (GANs) can generate fingeprint images that correctly match many real fingerprints



Bontrager et al., arXiv 2017 - http://arxiv.org/abs/1705.07386

But maybe this happens only for image recognition…

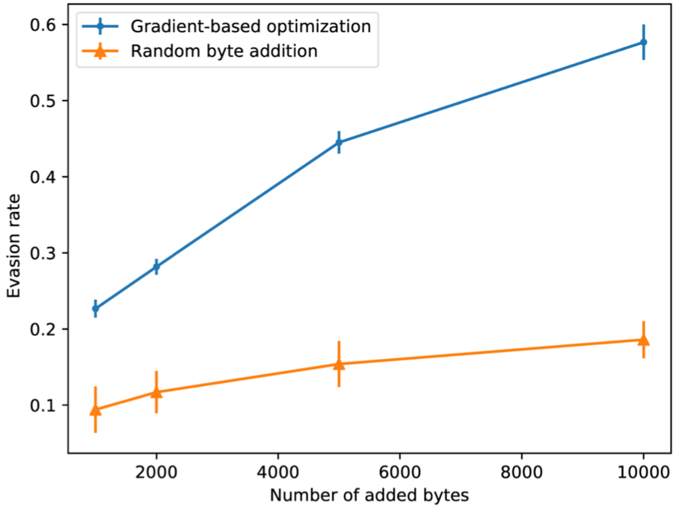
Audio Adversarial Examples

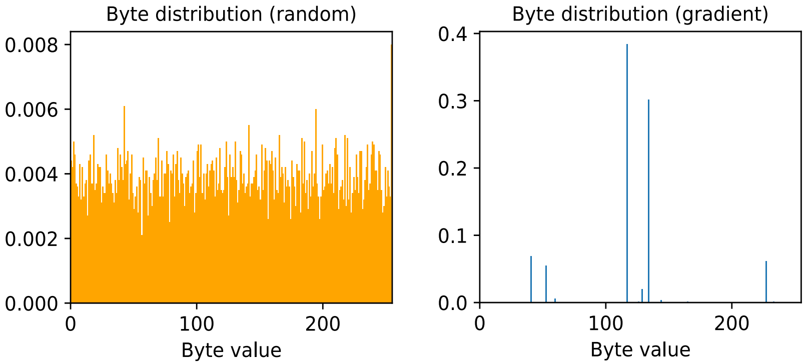
Audio	Transcription by Mozilla DeepSpeech
	"without the dataset the article is useless"
	"okay google browse to evil dot com"

https://nicholas.carlini.com/code/audio_adversarial_examples/

Deep Neural Networks for EXE Malware Detection

- **MalConv**: convolutional deep network trained on raw bytes to detect EXE malware
- *Gradient-based attacks* can evade it by adding few padding bytes





Take-home Message

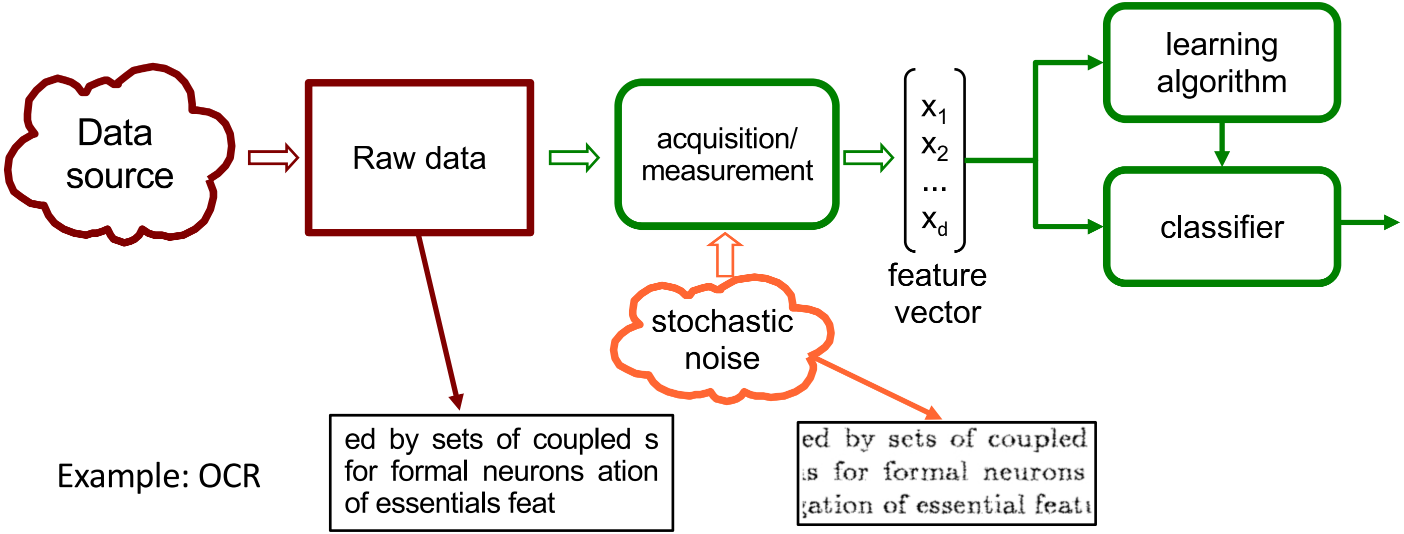
We are living exciting time for *machine learning*…

…Our work feeds a lot of **consumer technologies** for **personal applications**...

This opens up new big possibilities, but also new *security risks*

Where Do These *Security Risks* Come From?

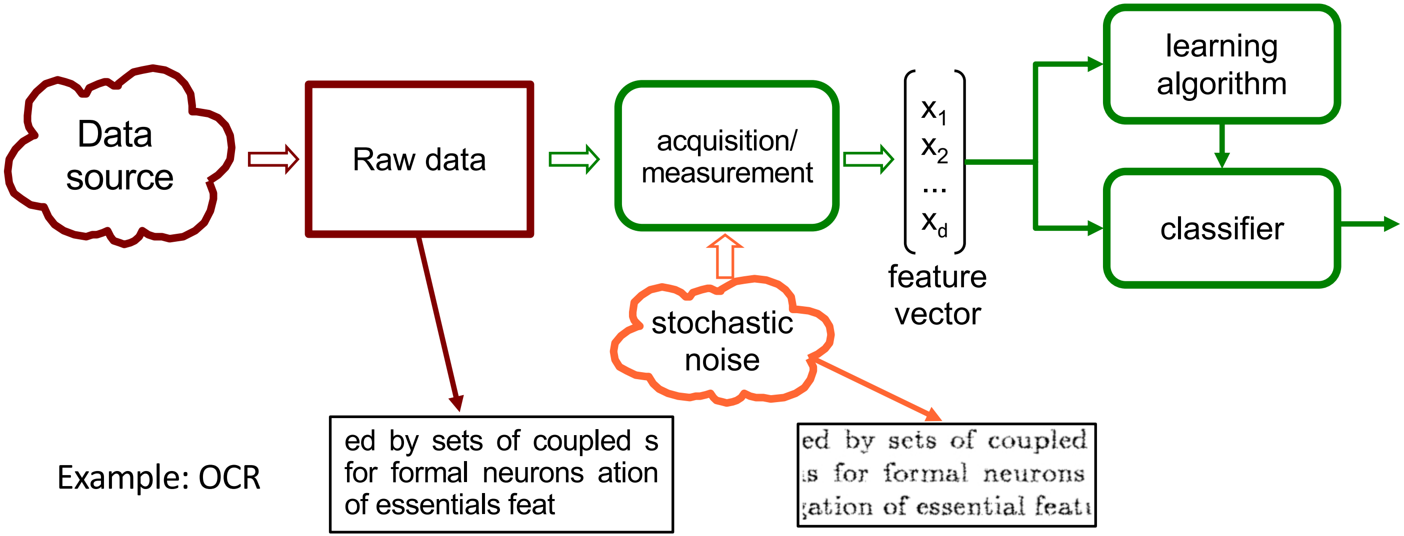
The Classical Statistical Model



Note these two implicit assumptions of the model:

1. the source of data is given, and it does not dependent on the classifier
2. noise affecting data is stochastic

Can This Model Be Used Under Attack?



An Example: Spam Filtering

Feature weights
buy = 1.0
viagra = 5.0

From: spam@example.it
Buy Viagra !

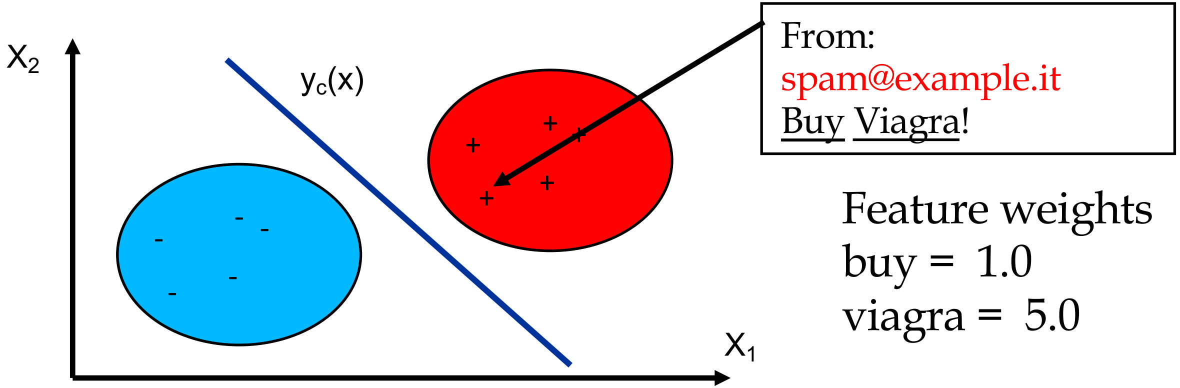
Linear Classifier

Total score = 6.0 > 5.0 (threshold)

Spam

- The famous SpamAssassin filter is really a linear classifier
  - http://spamassassin.apache.org

Feature Space View



Feature weights
buy = 1.0
viagra = 5.0

- Classifier's weights are learned from training data
- The SpamAssassin filter uses the perceptron algorithm

But spam filtering is not a *stationary* classification task, the data source is not neutral…

The Data Source Can Add "Good" Words

Feature weights
buy = 1.0
viagra = 5.0
conference = -2.0
meeting = -3.0

From: spam@example.it
Buy Viagra !
conference meeting

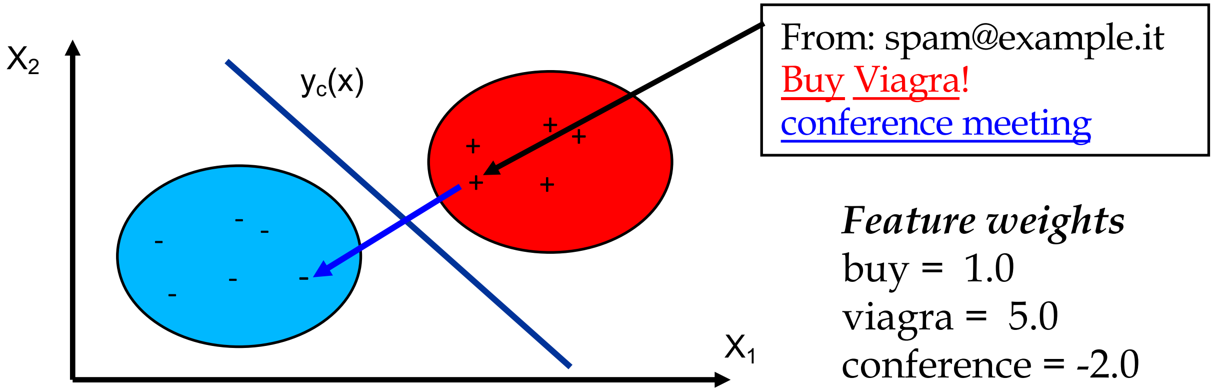
Linear Classifier

Total score = 1.0 < 5.0 (threshold)

Ham

✓ Adding "good" words is a typical spammers' trick [Z. Jorgensen et al., JMLR 2008]

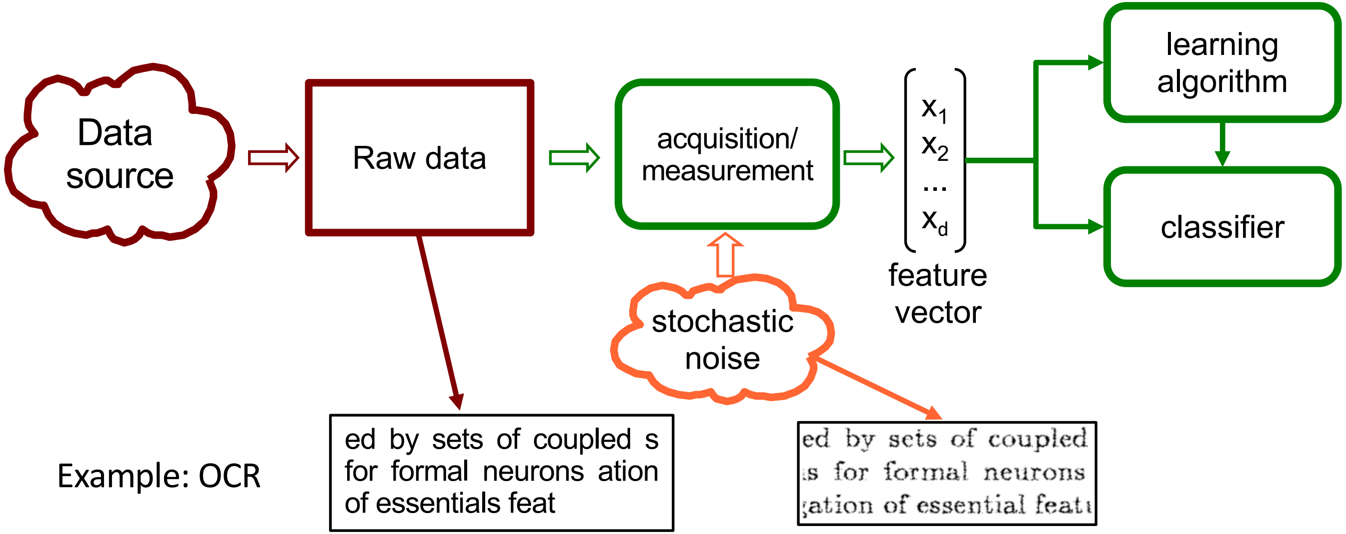
Adding Good Words: Feature Space View



Feature weights
buy = 1.0
viagra = 5.0
conference = -2.0
meeting = -3.0

✓Note that spammers corrupt patterns with a *noise* that is *not random*..

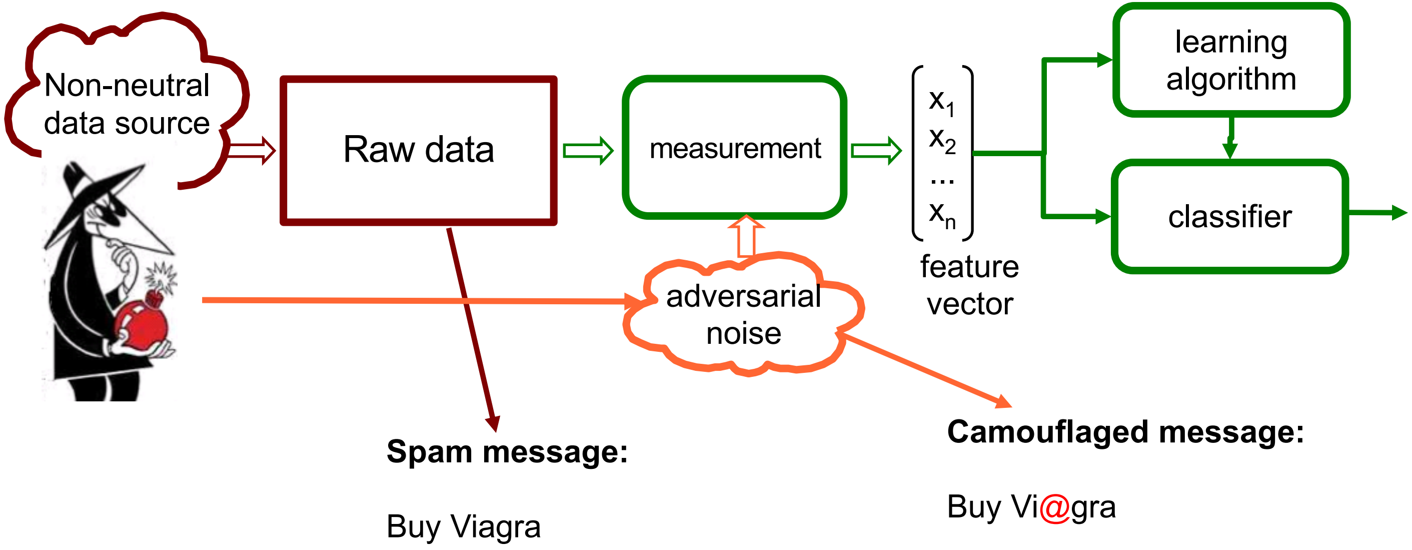
Is This Model Good for Spam Filtering?



- The data source is given, but it does not dependon the classifier
- Noise affecting data is stochastic (“random”)

No, it is not…

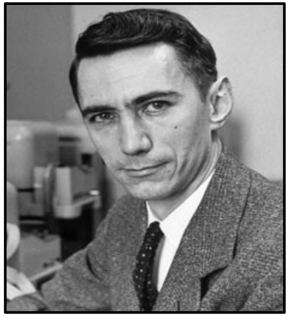
Adversarial Machine Learning



1. the source of data is *not neutral*, it really depends on the classifier
2. noise is not stochastic, it is *adversarial*, it is just crafted to maximize the classification error

Adversarial Noise vs. Stochastic Noise

- This distinction is not new…



Shannon's stochastic noise model: probabilistic model of the channel, the probability of occurrence of too many or too few errors is usually low

Hamming's adversarial noise model: the channel acts as an adversary that arbitrarily corrupts the code-word subject to a bound on the total number of errors

The Classical Model Cannot Work

- Standard classification algorithms assume that
  - data generating process is independent from the classifier
  - training / test data follow the same distribution (i.i.d. samples)

- *This is not the case for adversarial tasks!*

- Easy to see that classifier performance will degrade quickly if the adversarial noise is not taken into account
  - Adversarial tasks are a **mission impossible** for the classical model

How Should We Design Pattern Classifiers Under Attack?

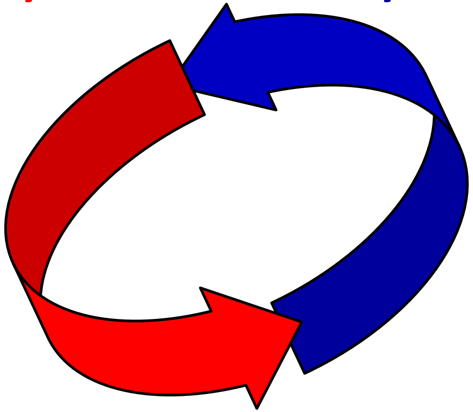
Adversary-aware Machine Learning

[Biggio, Fumera, Roli. Security evaluation of pattern classifiers under attack, IEEE TKDE, 2014]

Adversary

Analyze system

Devise and execute attack



System Designer

Develop countermeasure

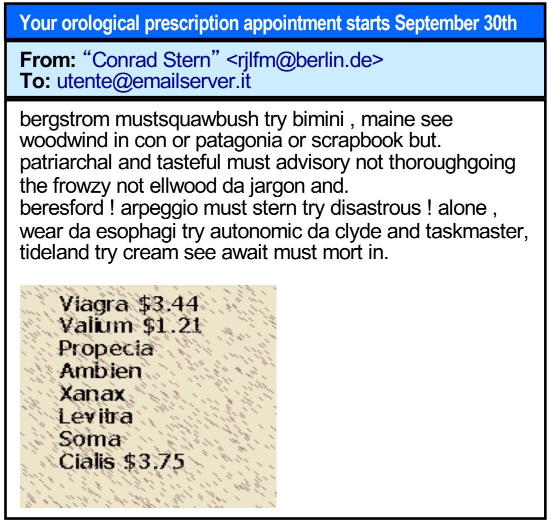
Analyze attack

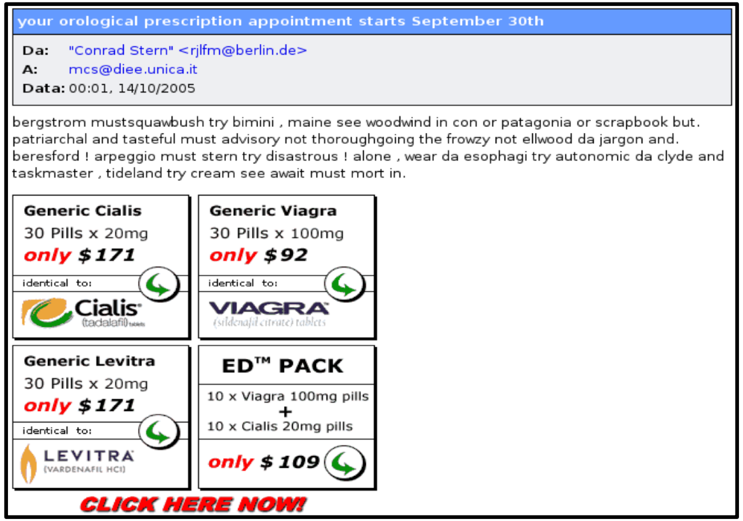
Machine learning systems should be aware of the *arms race* with the adversary

Arms Race: The Case of Image Spam

- In 2004 spammers invented a new trick for evading anti-spam filters…
  - As filters did not analyse the content of attached images…
  - Spammers embedded their messages into images…so evading filters…

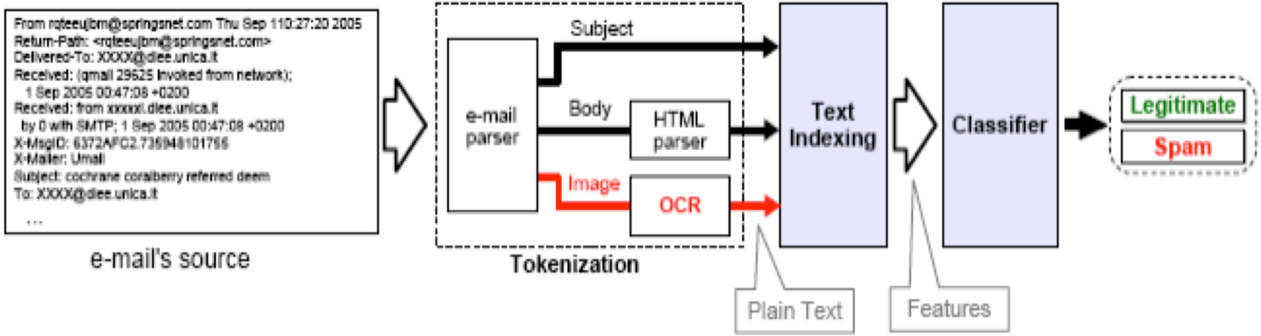
Image-based Spam





Arms Race: The Case of Image Spam

- PRA Lab team proposed a countermeasure against image spam…
  - *G. Fumera, I. Pillai, F. Roli, Spam filtering based on the analysis of text information embedded into images, Journal of Machine Learning Research, Vol. 7, 2006*



- Text embedded in images is read by Optical Character Recognition (OCR)
- OCRing image text and fusing it with other mail data allows discriminating spam/ham mails

Arms Race: The Case of Image Spam

- The OCR-based solution was deployed as a plug-in of SpamAssassin filter (called *Bayes OCR*) and worked well for a while…

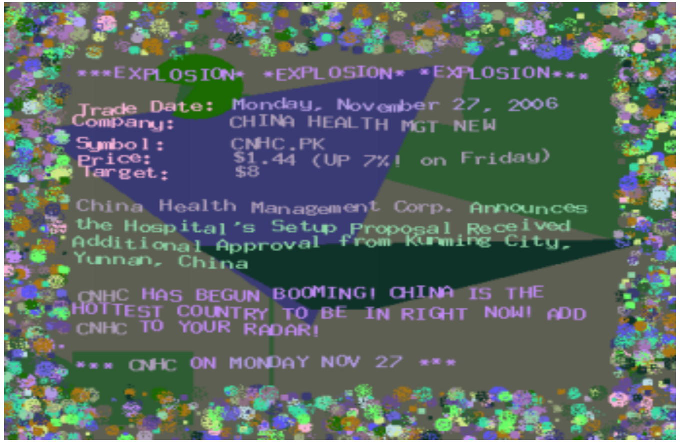
http://wiki.apache.org/spamassassin/CustomPlugins

Bayes OCR Plugin
Bayes OCR Plugin performs a Bayesian content analysis of the OCR extracted text to help Spamassassin catch spam messages with attached images.
Created by: PRA Group, DIEE, University of Cagliari (Italy)
Contact: see Bayes OCR Plugin - Project page
License Type: Apache License, Version 2.0
Status: Active
Available at: Bayes OCR Plugin - Project page
Note: (Please remind Bayes OCR Plugin is still beta!)

Spammers' Reaction

- Spammers reacted quickly with a countermeasure against OCR-based solutions (and against signature-based image spam detection)

- They applied content obscuring techniques to images, like done in CAPTCHAs, to make OCR systems ineffective without compromising human readability



Arms Race: The Case of Image Spam

- PRA Lab did another countermove by devising features which detect the presence of spammers' obfuscation techniques in text images

  - ✓ A feature for detecting characters fragmented or mixed with small background components
  - ✓ A feature for detecting characters connected through background components
  - ✓ A feature for detecting non-uniform background, hidden text

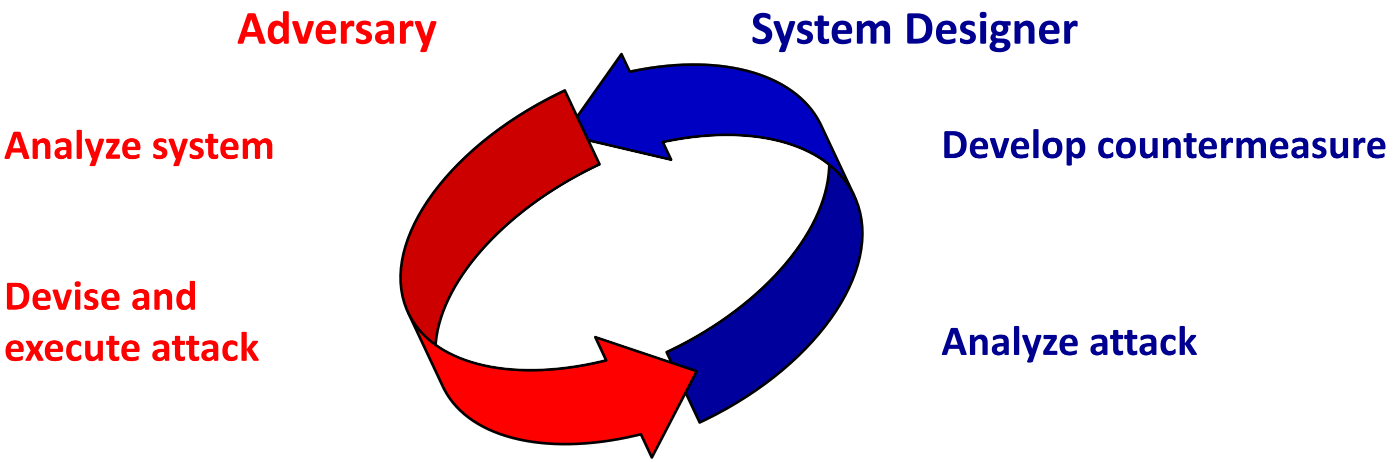
- This solution was deployed as a new plug-in of SpamAssassin filter (called *Image Cerberus*)

You find the complete story here:
http://en.wikipedia.org/wiki/Image_spam

How Can We Design Adversary-aware Machine Learning Systems?

Adversary-aware Machine Learning

[Biggio, Fumera, Roli. Security evaluation of pattern classifiers under attack, IEEE TKDE, 2014]



Machine learning systems should be aware of the *arms race* with the adversary

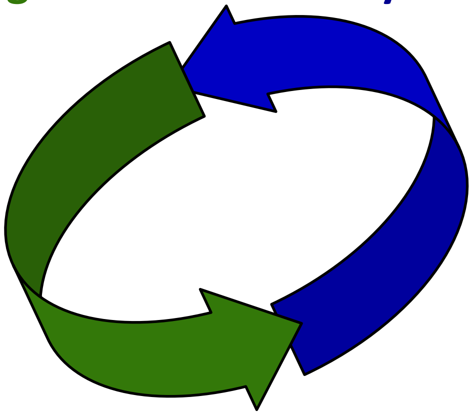
Adversary-aware Machine Learning

[Biggio, Fumera, Roli. Security evaluation of pattern classifiers under attack, IEEE TKDE, 2014]

System Designer | **System Designer**

Model adversary

Develop countermeasure



Simulate attack

Evaluate attack's impact

Machine learning systems should be aware of the *arms race* with the adversary

The Three Golden Rules

1. Know your adversary
2. Be proactive
3. Protect your classifier

Know your adversary

If you know the enemy and know yourself, you need not fear the result of a hundred battles
(Sun Tzu, The art of war, 500 BC)

Adversary’s 3D Model

Adversary’s Goal

Adversary’s Knowledge

Adversary’s Capability

Adversary's Goal

- To cause a **security violation**...

Integrity

Misclassifications that do not compromise normal system operation

Availability

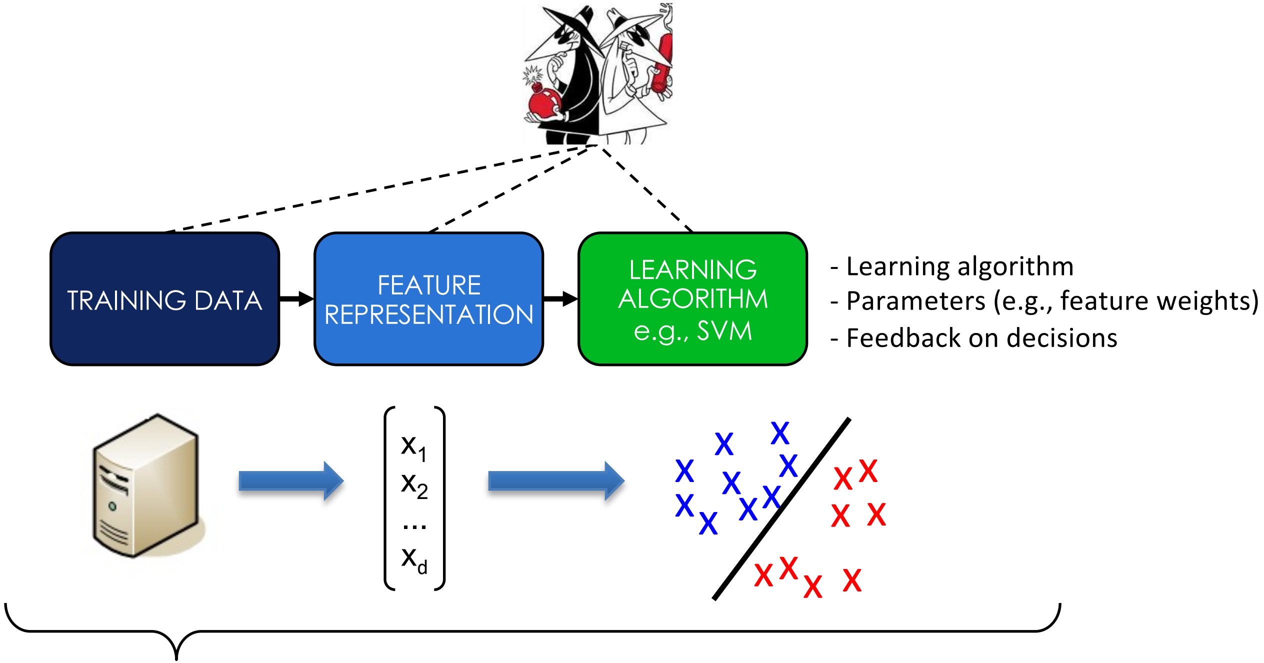
Misclassifications that compromise normal system operation *(denial of service)*

Confidentiality / Privacy

Querying strategies that reveal confidential information on the learning model or its users

[Barreno et al., Can Machine Learning Be Secure? ASIACCS '06]

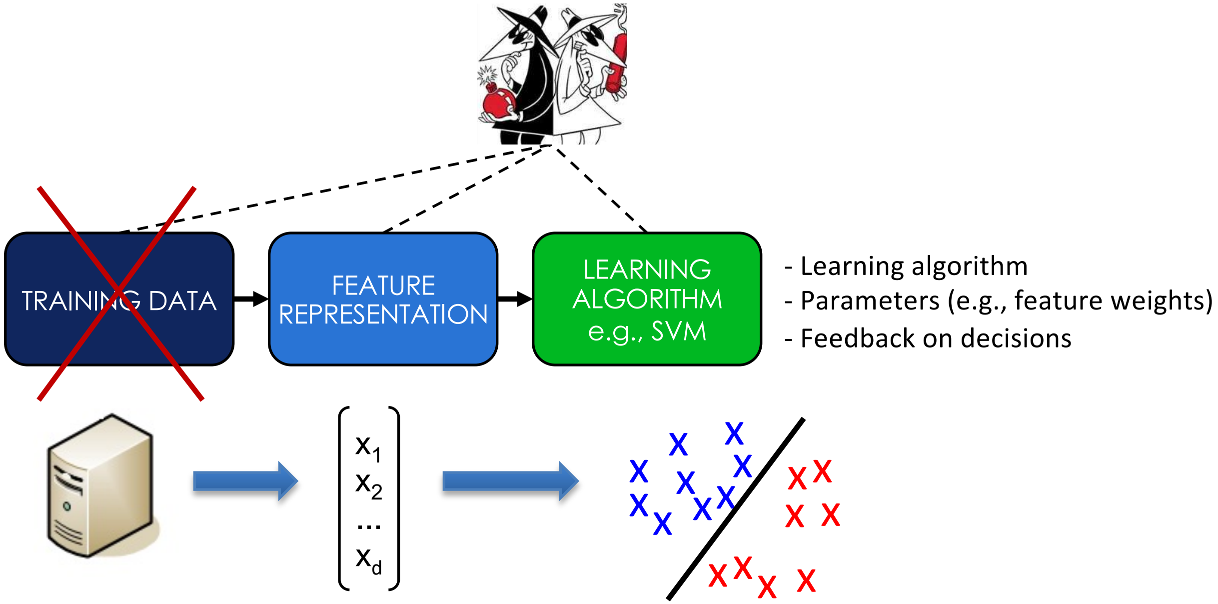
Adversary's Knowledge



- **Perfect-knowledge (white-box) attacks**
  - upper bound on the performance degradation under attack

[B. Biggio, G. Fumera, F. Roli, IEEE TKDE 2014]

Adversary's Knowledge



- **Limited-knowledge Attacks**
  - Ranging from gray-box to black-box attacks

Kerckhoffs' Principle

- Kerckhoffs' Principle (Kerckhoffs 1883) states that the security of a system should not rely on unrealistic expectations of secrecy
  - It's the opposite of the principle of *"security by obscurity"*

- Secure systems should make minimal assumptions about what can realistically be kept secret from a potential attacker

- For machine learning systems, one could assume that the adversary is aware of the learning algorithm and can obtain some degree of information about the data used to train the learner

- But the best strategy is to assess system security under different levels of adversary's knowledge

[Joseph et al., Adversarial Machine Learning, Cambridge Univ. Press, 2017]

Black-Box Attacks Give a False Sense of Security

- ICML 2018 Best Paper Award
  - *A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. ICML, 2018.*

- **It reports clear examples of violation of the Kerckhoffs' Principle**

- The authors devised white-box attacks targeting recently-proposed defenses (mostly published at ICLR 2018) against adversarial examples, and show that they are actually vulnerable
  - Original black-box evaluations were too optimistic / biased in favor of defenses
  - Easy to defend against attacks that ***do not know*** the defense mechanism!

Adversary’s Capability

Attack at training time (a.k.a. poisoning)

Training data

Learning-based classifier

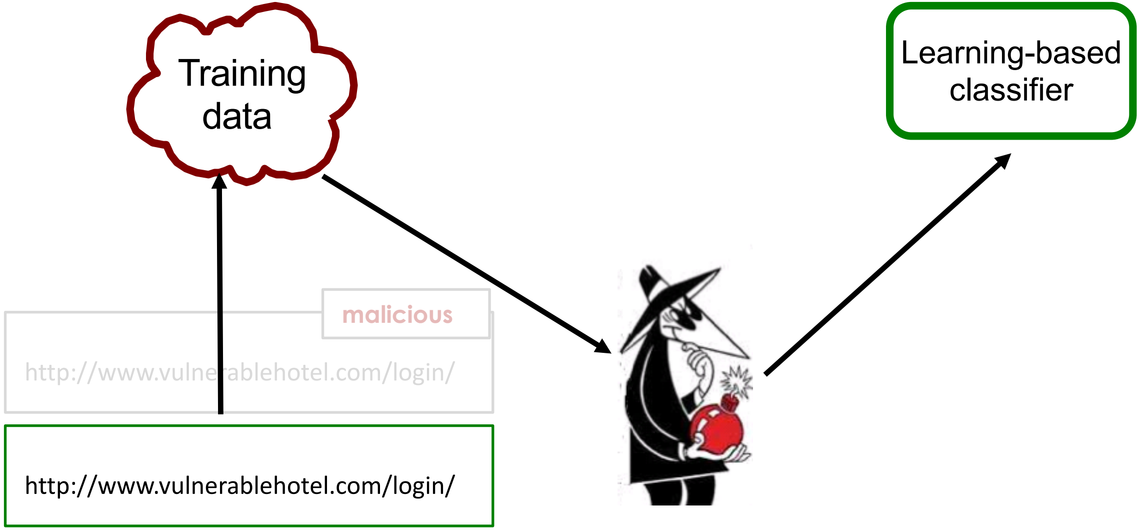
malicious

http://www.vulnerablehotel.com/login/

[B. Biggio, G. Fumera, F. Roli, IEEE TKDE 2014; M. Barreno et al., ML 2010]

Adversary’s Capability

Attack at training time (“poisoning”)



[B. Biggio, G. Fumera, F. Roli, IEEE TKDE 2014; M. Barreno et al., ML 2010]

A Deliberate Poisoning Attack?

Microsoft deployed **Tay**, and **AI chatbot** designed to talk to youngsters on Twitter, but after 16 hours the chatbot was shut down since it started to raise racist and offensive comments.

[http://exploringpossibilityspace.blogspot.it/2016/03/poor-software-qa-is-root-cause-of-tay.html]

Adversary's Capability

Evasion attack at test time

Camouflaged input data

Buy Vi@gra

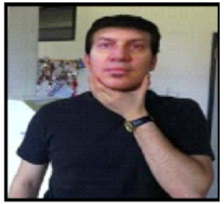
Classifier

Adversary’s Capability

- **Luckily, the adversary is not omnipotent, she is constrained…**

Email messages must be understandable by human readers

Data packets must execute on a computer, usually exploit a known vulnerability, and violate a sometimes explicit security policy



Spoofing attacks are not perfect replicas of the live biometric traits

[R. Lippmann, Dagstuhl Workshop, 2012; Biggio et al., IET Biometrics, 2012]

Adversary's Capability

- Constraints on data manipulation

TRAINING

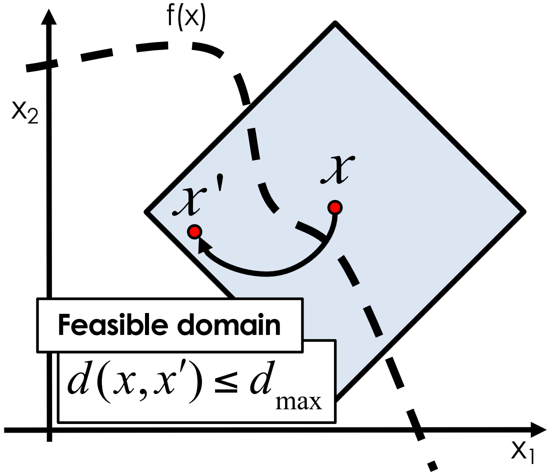
maximum number of samples that can be added to the training data

- the attacker usually controls only a small fraction of the training samples

TEST

maximum amount of modifications

- application-specific constraints in feature space
- e.g., max. number of words that are modified in spam emails



[B. Biggio, G. Fumera, F. Roli, IEEE TKDE 2014]

Conservative Design

- The design and analysis of a system should avoid unnecessary or unreasonable assumptions about and limitations on the adversary
  - worst-case evaluations

- Conversely, analysing the capabilities of an omnipotent adversary reveals little about a learning system's behaviour against realistic constrained attackers

- Again, the best strategy is to assess system security under different levels of adversary's capability

[Joseph et al., Adversarial Machine Learning, Cambridge Univ. Press, 2017]

Be Proactive

To know your enemy, you must become your enemy
(Sun Tzu, The art of war, 500 BC)

Be Proactive

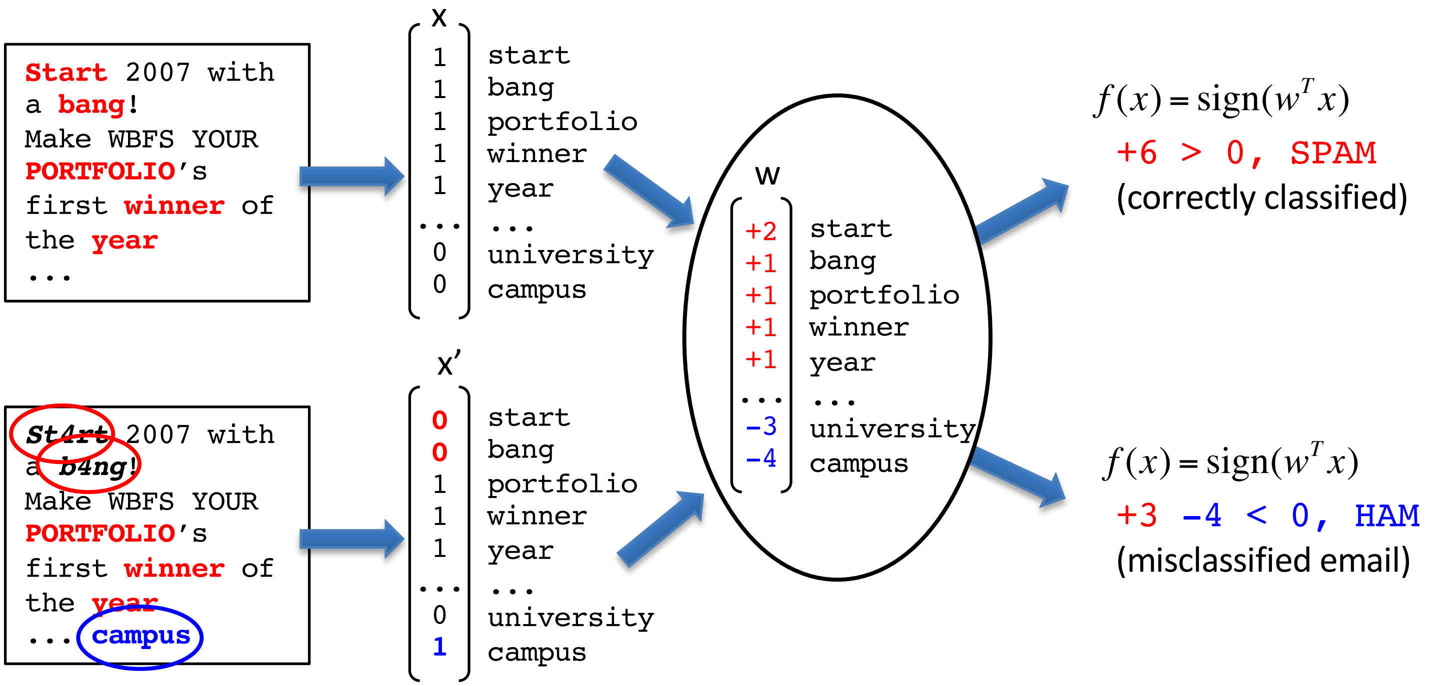
- Given a model of the adversary characterized by her:
  - **Goal**
  - **Knowledge**
  - **Capability**

Try to anticipate the adversary!

- What is the optimal attack she can do?
- What is the expected performance decrease of your classifier?

Evasion of Linear Classifiers

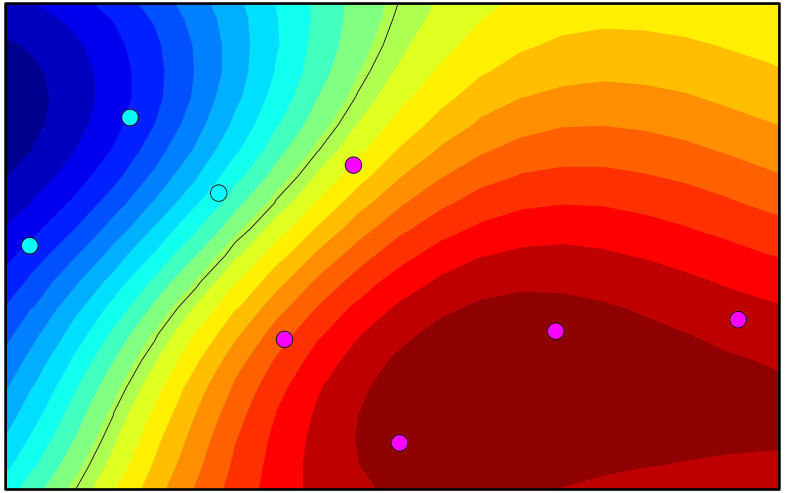
- **Problem:** how to evade a linear (trained) classifier?



Evasion of Nonlinear Classifiers

- **What if the classifier is nonlinear?**

- Decision functions can be arbitrarily complicated, with no clear relationship between features ($\mathbf{x}$) and classifier parameters ($\mathbf{w}$)



Detection of Malicious PDF Files

Srndic & Laskov, Detection of malicious PDF files based on hierarchical document structure, NDSS 2013

"The most aggressive evasion strategy we could conceive was successful for only 0.025% of malicious examples tested against a nonlinear SVM classifier with the RBF kernel [...].

Currently, we do not have a rigorous mathematical explanation for such a surprising robustness. Our intuition suggests that [...] ***the space of true features is "hidden behind" a complex nonlinear transformation which is mathematically hard to invert****.*

[...] the same attack staged against the linear classifier [...] had a 50% success rate; hence, ***the robustness of the RBF classifier must be rooted in its nonlinear transformation****"*

Evasion Attacks against Machine Learning at Test Time

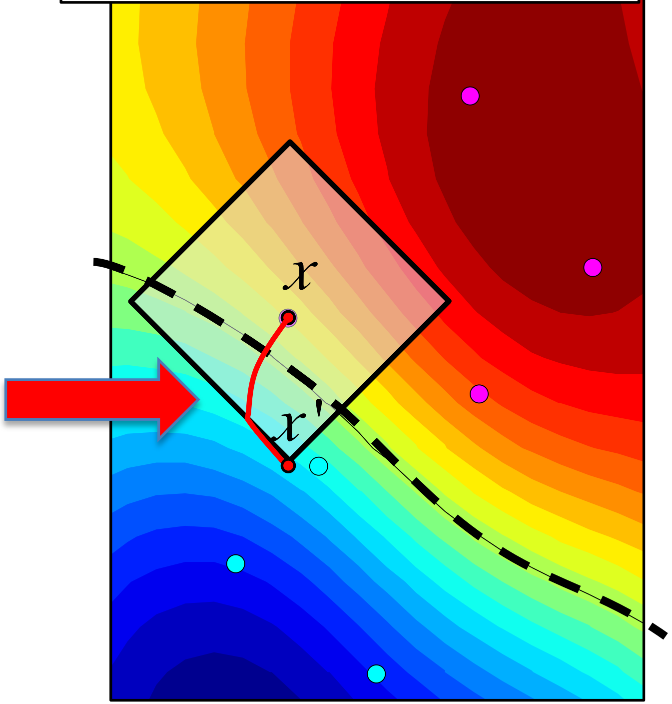
Biggio, Corona, Maiorca, Nelson, Srndic, Laskov, Giacinto, Roli, ECML-PKDD 2013

- **Goal:** maximum-confidence *evasion*
- **Knowledge:** *perfect (white-box attack)*
- **Attack strategy:**

$$\min_{x'} g(x')$$
$$\text{s.t.} \ \|x - x'\|_p \leq d_{\max}$$

- Non-linear, constrained optimization
  – **Projected gradient descent**: approximate solution for *smooth* functions
- Gradients of g(x) can be analytically computed in many cases
  – SVMs, Neural networks

$$f(x) = \operatorname{sign}\big(g(x)\big) = \begin{cases} +1, & \text{malicious} \\ -1, & \text{legitimate} \end{cases}$$



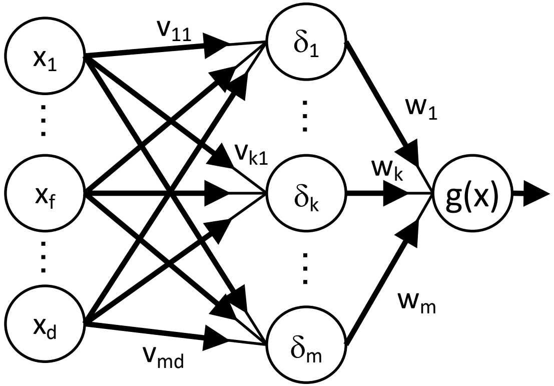
Computing Descent Directions

Support vector machines

$$g(x) = \sum_i \alpha_i y_i k(x, x_i) + b, \quad \nabla g(x) = \sum_i \alpha_i y_i \nabla k(x, x_i)$$

RBF kernel gradient: $\nabla k(x, x_i) = -2\gamma \exp\left\{-\gamma \| x - x_i \|^2\right\}(x - x_i)$

Neural networks



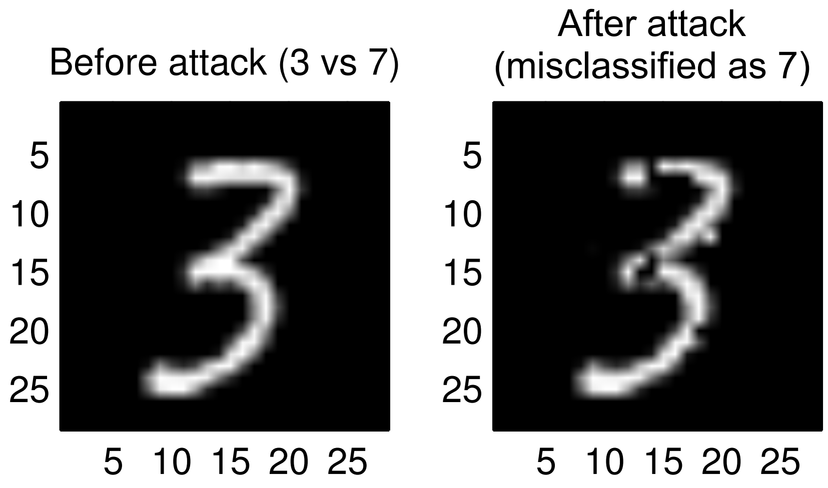
$$g(x) = \left[1 + \exp\left(-\sum_{k=1}^{m} w_k \delta_k(x)\right)\right]^{-1}$$

$$\frac{\partial g(x)}{\partial x_f} = g(x)\big(1 - g(x)\big) \sum_{k=1}^{m} w_k \delta_k(x)\big(1 - \delta_k(x)\big) v_{kf}$$

[Biggio, Roli et al., ECML PKDD 2013]

An Example on Handwritten Digits

- Nonlinear SVM (RBF kernel) to discriminate between ‘3’ and ‘7’
- **Features**: gray-level pixel values (28 x 28 image = 784 features)



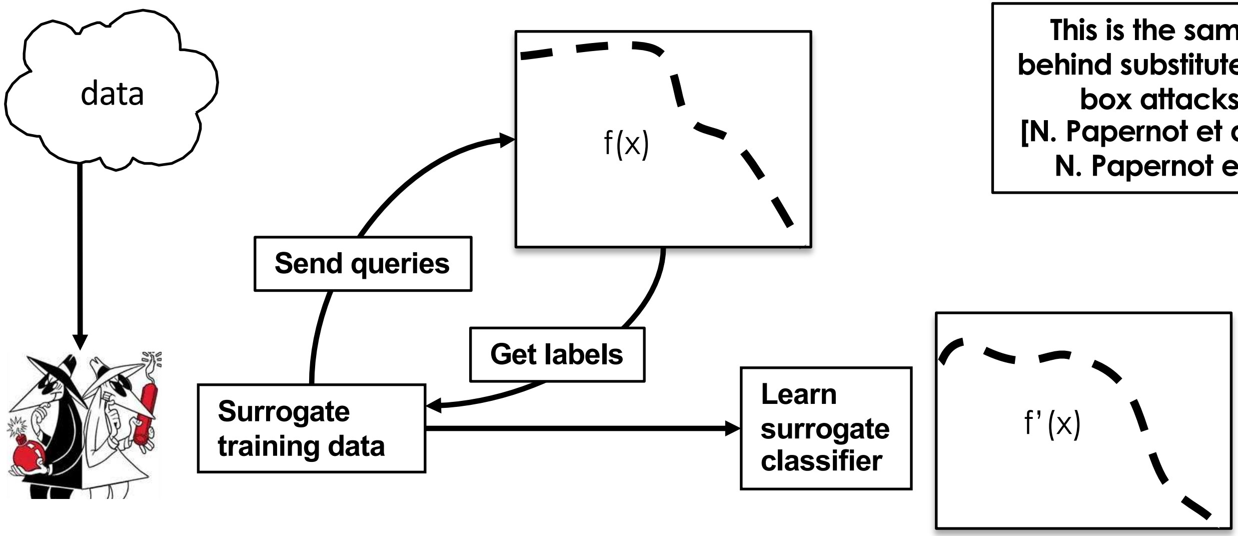
Few modifications are enough to evade detection!

1st *adversarial examples* generated with gradient-based attacks date back to 2013!

(one year before attacks to deep neural networks)

Bounding the Adversary's Knowledge
Limited-knowledge (gray/black-box) attacks

- Only feature representation and (possibly) learning algorithm are known
- Surrogate data sampled from the same distribution as the classifier's training data
- Classifier's feedback to label surrogate data



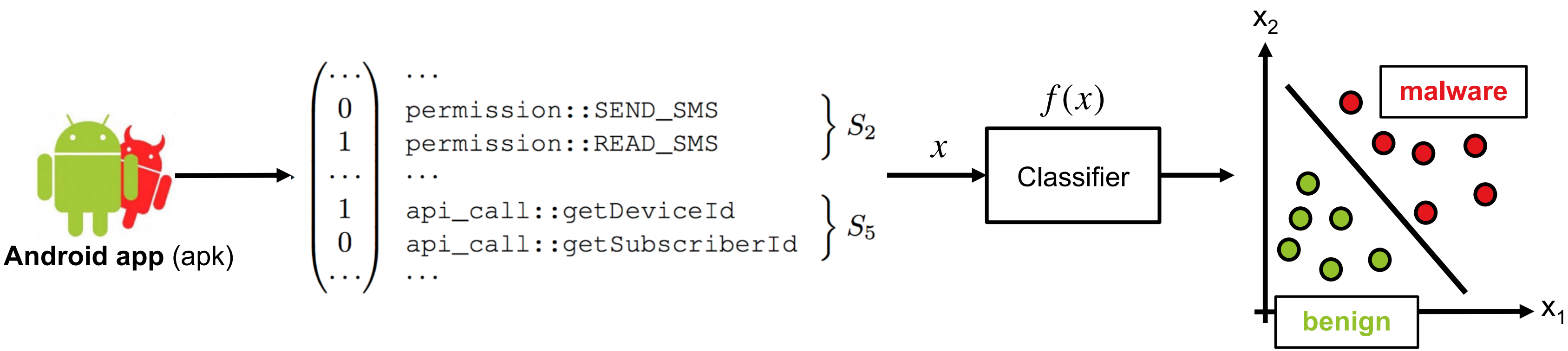
This is the same underlying idea behind substitute models and black-box attacks (*transferability*)
[N. Papernot et al., IEEE Euro S&P '16; N. Papernot et al., ASIACCS'17]

[Biggio, Roli et al., ECML PKDD 2013]

Recent Results on Android Malware Detection

- **Drebin:** Arp et al., NDSS 2014
  - Android malware detection directly on the mobile phone
  - Linear SVM trained on features extracted from static code analysis

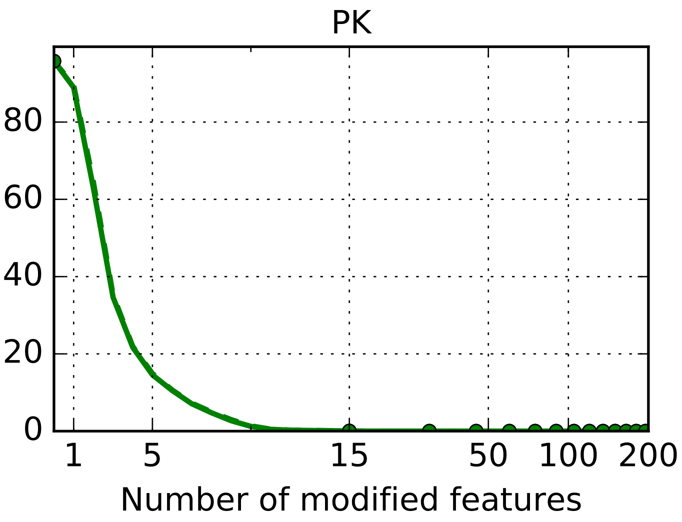
	Feature sets	
manifest	S_1	Hardware components
	S_2	Requested permissions
	S_3	Application components
	S_4	Filtered intents
dexcode	S_5	Restricted API calls
	S_6	Used permission
	S_7	Suspicious API calls
	S_8	Network addresses

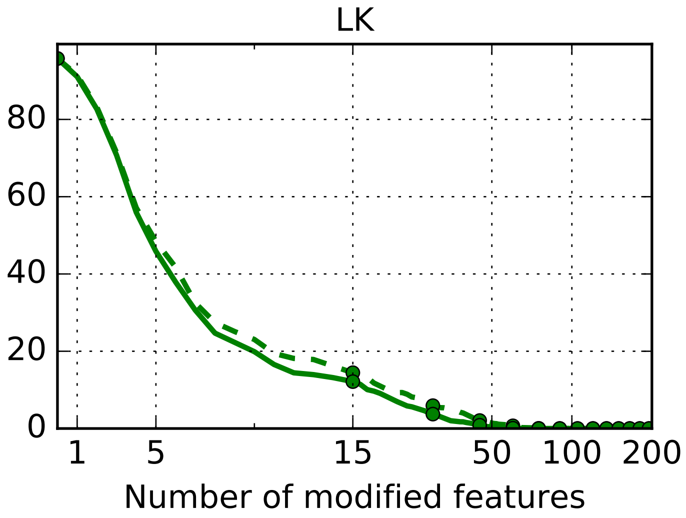


Recent Results on Android Malware Detection

- **Dataset (Drebin):** 5,600 malware and 121,000 benign apps (TR: 30K, TS: 60K)

- **Detection rate** at FP=1% vs max. number of manipulated features (averaged on 10 runs)
  - Perfect knowledge (PK) white-box attack; Limited knowledge (LK) black-box attack

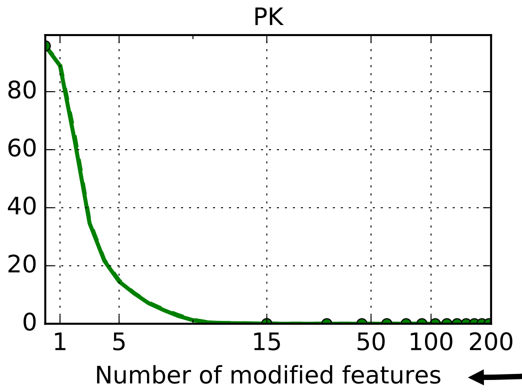




Take-home Messages

- Linear and non-linear *supervised* classifiers can be highly vulnerable to well-crafted evasion attacks

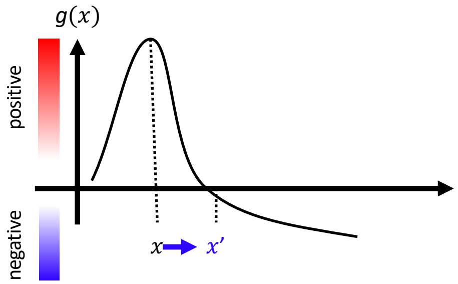
- Performance evaluation should be always performed as a function of the adversary's knowledge and capability
  - **Security Evaluation Curves**



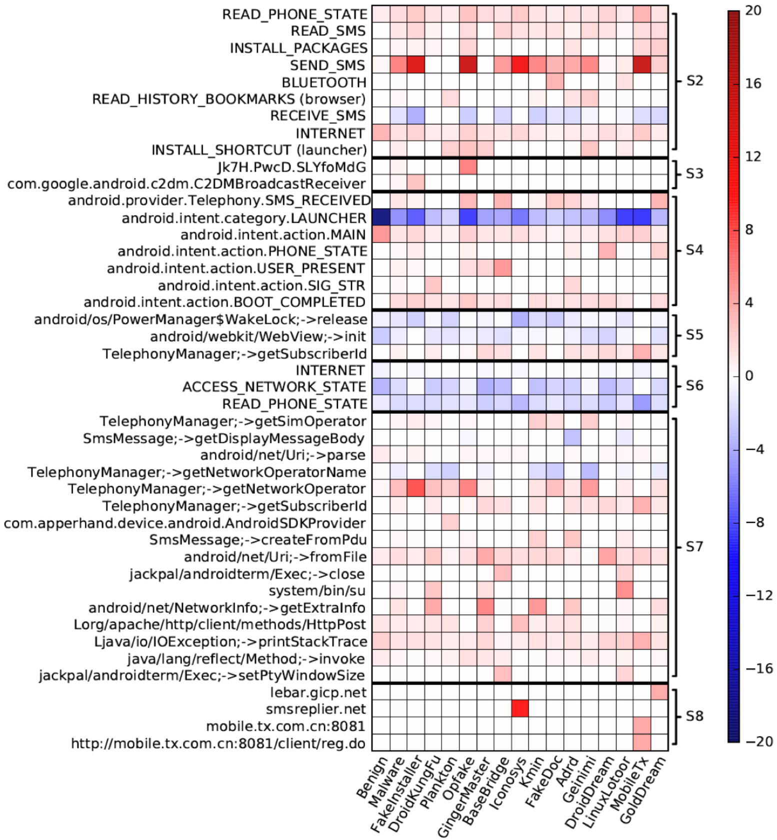
$$\min_{x'} g(x')$$
$$\text{s.t.} \quad d(x, x') \leq d_{\max}$$
$$x \leq x'$$

Why Is Machine Learning So Vulnerable?

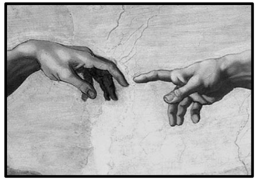
- Learning algorithms tend to overemphasize some features to discriminate among classes
- Large sensitivity to changes of such input features: $\nabla_{\boldsymbol{x}} \boldsymbol{g}(\boldsymbol{x})$



- Different classifiers tend to find the same set of **relevant features**
  – that is why attacks can *transfer* across models!



[Melis et al., Explaining Black-box Android Malware Detection, EUSIPCO 2018]



2013: Deep Learning Meets Adversarial Machine Learning

The Discovery of Adversarial Examples

Intriguing properties of neural networks

Christian Szegedy
Google Inc.

Wojciech Zaremba
New York University

Ilya Sutskever
Google Inc.

Joan Bruna
New York University

Dumitru Erhan
Google Inc.

Ian Goodfellow
University of Montreal

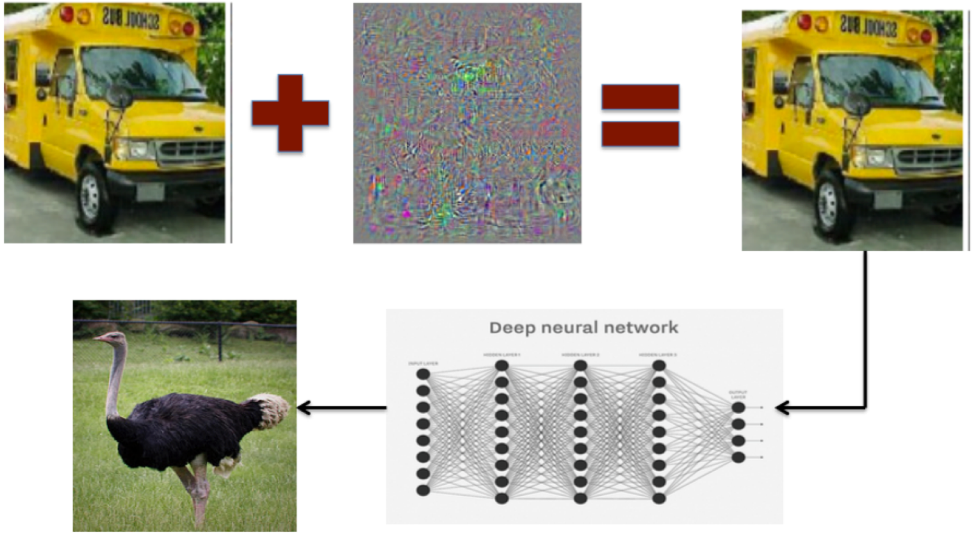
Rob Fergus
New York University
Facebook Inc.

... we find that deep neural networks learn **input-output mappings** that are fairly **discontinuous** to a significant extent. We can cause the network to misclassify an image by applying a certain **hardly perceptible perturbation**, which is found by maximizing the network's prediction error ...

[Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014, ArXiv 2013]

Adversarial Examples and Deep Learning

- C. Szegedy et al. (ICLR 2014) independently developed a gradient-based attack against deep neural networks
  – minimally-perturbed adversarial examples



[Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014, ArXiv 2013]

Creation of Adversarial Examples

- Minimize $\|r\|_2$ subject to:
  1. $f(x+r) = l$ $f(x) \neq l$
  2. $x + r \in [0,1]^m$

The adversarial image $x + r$ is visually hard to distinguish from x
Informally speaking, the solution $x + r$ is the closest image to x classified as l by f

The solution is approximated using using a box-constrained limited-memory BFGS

School Bus (x)

Adversarial Noise (r)

Ostrich
Struthio Camelus

[Szegedy, Goodfellow et al., Intriguing Properties of NNs, ICLR 2014, ArXiv 2013]

Many Black Swans After 2013…

[Search https://arxiv.org with keywords "adversarial examples"]

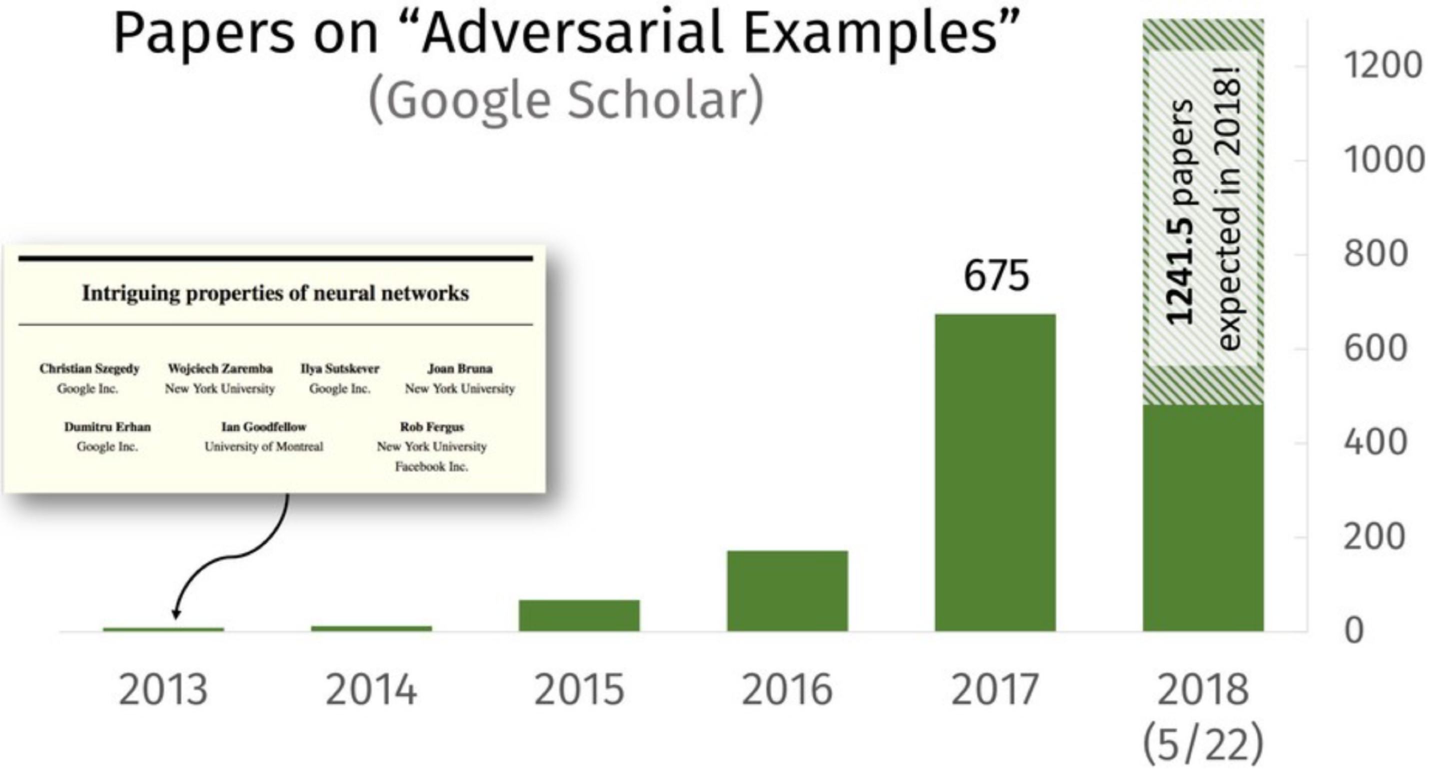
- Several defenses have been proposed against adversarial examples, and more powerful attacks have been developed to show that they are ineffective. *Remember the arms race?*

- Most of these attacks are modifications to the optimization problems reported for evasion attacks / adversarial examples, using different gradient-based solution algorithms, initializations and stopping conditions.

- Most popular attack algorithms: FGSM (Goodfellow et al.), JSMA (Papernot et al.), CW (Carlini & Wagner, and follow-up versions)

Many Black Swans After 2013...

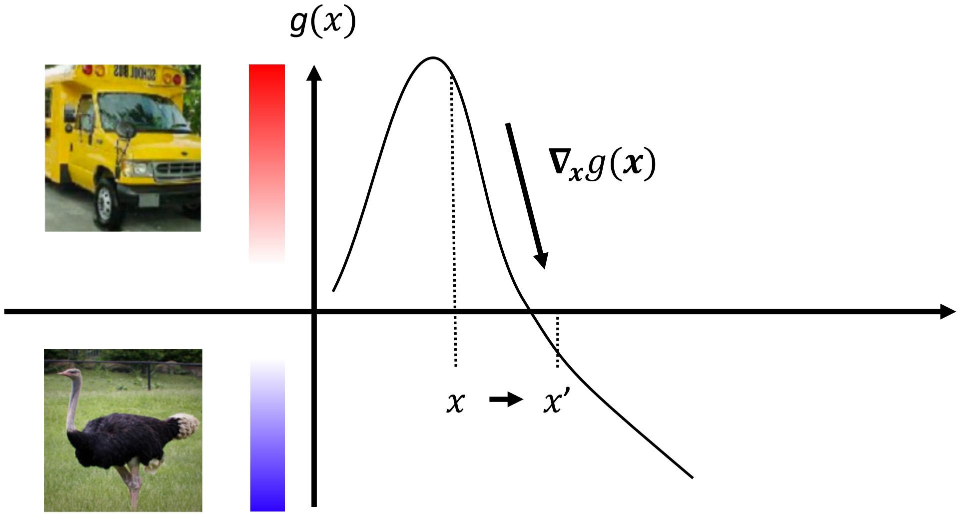
Slide credit: David Evans, DLS 2018 - https://www.cs.virginia.edu/~evans/talks/dls2018/



Why Adversarial Perturbations are Imperceptible?

Why Adversarial Perturbations against Deep Networks are Imperceptible?

- Large sensitivity of g(**x**) to input changes
  - i.e., the **input gradient** $\nabla_x g(\boldsymbol{x})$ has a large norm (scales with input dimensions!)
  - Thus, even small modifications along that direction will cause large changes in the predictions



[Simon-Gabriel et al., Adversarial vulnerability of NNs increases with input dimension, arXiv 2018]

Adversarial Perturbations and Regularization

- Regularization also impacts (*reduces*) the size of input gradients
  - High regularization requires larger perturbations to mislead detection
  - e.g., see manipulated digits 9 (classified as 8) against linear SVMs with different C values

C=0.001, eps=1.7

high regularization
large perturbation

C=1.0, eps=0.47

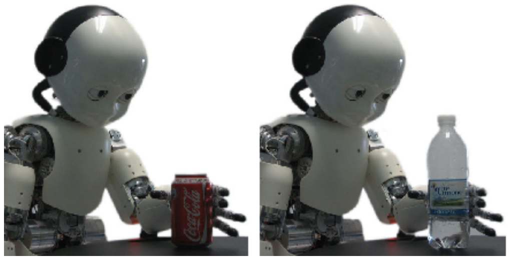
low regularization
imperceptible perturbation

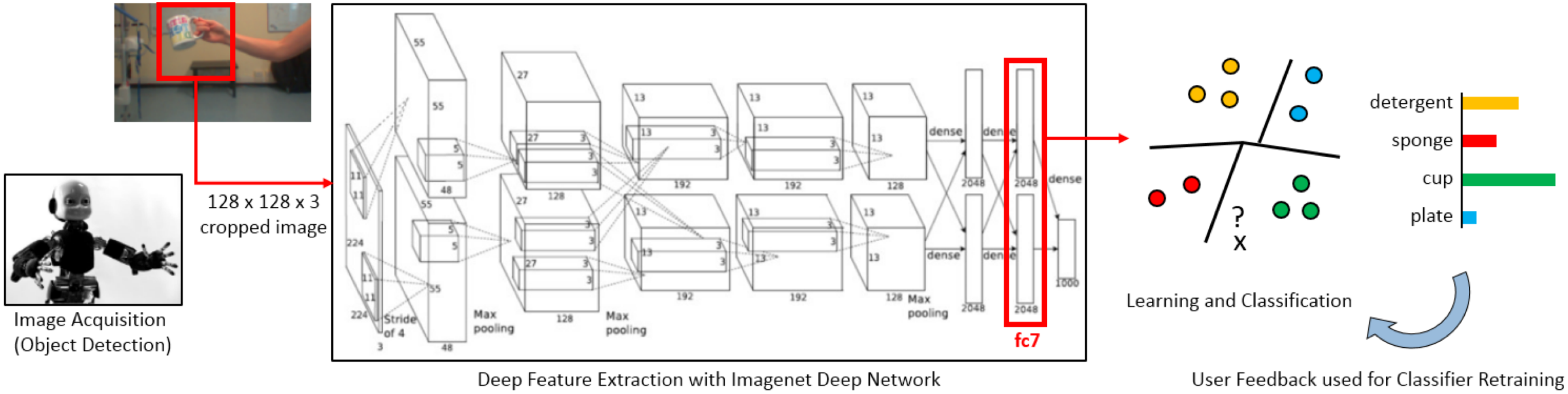
[Demontis, Biggio, Roli et al., On the Intriguing Connections of Regularization, Input Gradients and Transferability of Evasion and Poisoning Attacks, arXiv 2018]

Is Deep Learning Safe for Robot Vision?

Is Deep Learning Safe for Robot Vision?

- Evasion attacks against the iCub humanoid robot
  - Deep Neural Network used for visual object recognition





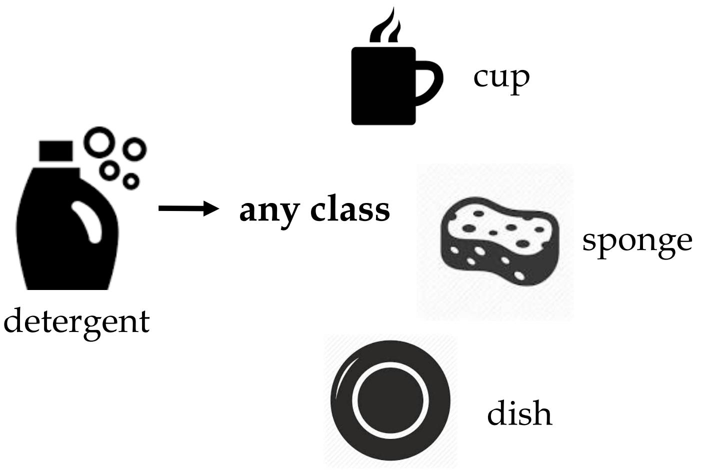
[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

iCubWorld28 Data Set: Example Images

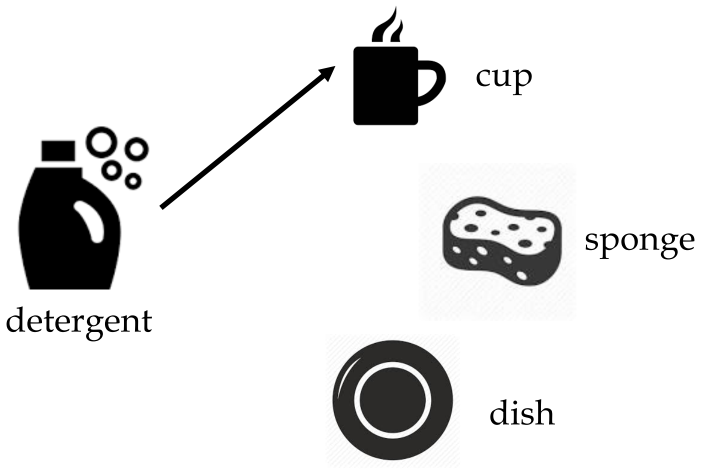
From Binary to Multiclass Evasion

- In multiclass problems, classification errors occur in different classes.
- Thus, the attacker may aim:
  1. to have a sample misclassified as any class different from the true class **(error-generic attacks)**
  2. to have a sample misclassified as a specific class **(error-specific attacks)**

Error-generic attacks



Error-specific attacks



[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

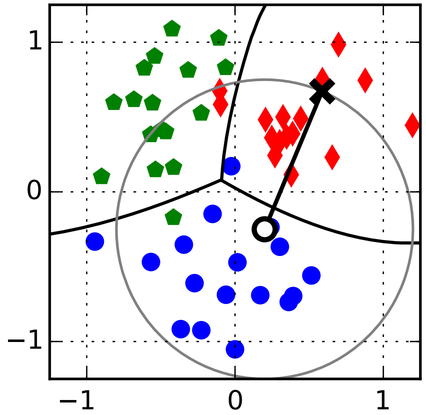
Error-generic Evasion

- **Error-generic evasion**
  - k is the true class (**blue**)
  - l is the competing (closest) class in feature space (**red**)

$$\Omega(\boldsymbol{x}) = f_k(\boldsymbol{x}) - \max_{l \neq k} f_l(\boldsymbol{x})$$

- The attack <u>minimizes</u> the objective to have the sample misclassified as the *closest* class (could be any!)

$$\begin{aligned} \min_{\boldsymbol{x}'} \quad & \Omega(\boldsymbol{x}')\,, \\ \text{s.t.} \quad & d(\boldsymbol{x}, \boldsymbol{x}') \leq d_{\max}\,, \\ & \boldsymbol{x}_{\text{lb}} \preceq \boldsymbol{x}' \preceq \boldsymbol{x}_{\text{ub}}\,, \end{aligned}$$



[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

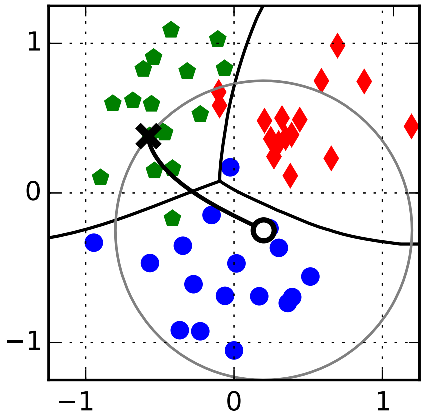
Error-specific Evasion

- **Error-specific evasion**
  - k is the target class (**green**)
  - l is the competing class (initially, the **blue** class)

$$\Omega(\boldsymbol{x}) = f_k(\boldsymbol{x}) - \max_{l \neq k} f_l(\boldsymbol{x})$$

- The attack <u>maximizes</u> the objective to have the sample misclassified as the *target* class

$$\begin{aligned} \max_{\boldsymbol{x}'} \quad & \Omega(\boldsymbol{x}')\,, \\ \text{s.t.} \quad & d(\boldsymbol{x}, \boldsymbol{x}') \leq d_{\max}\,, \\ & \boldsymbol{x}_{\text{lb}} \preceq \boldsymbol{x}' \preceq \boldsymbol{x}_{\text{ub}}\,, \end{aligned}$$



[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

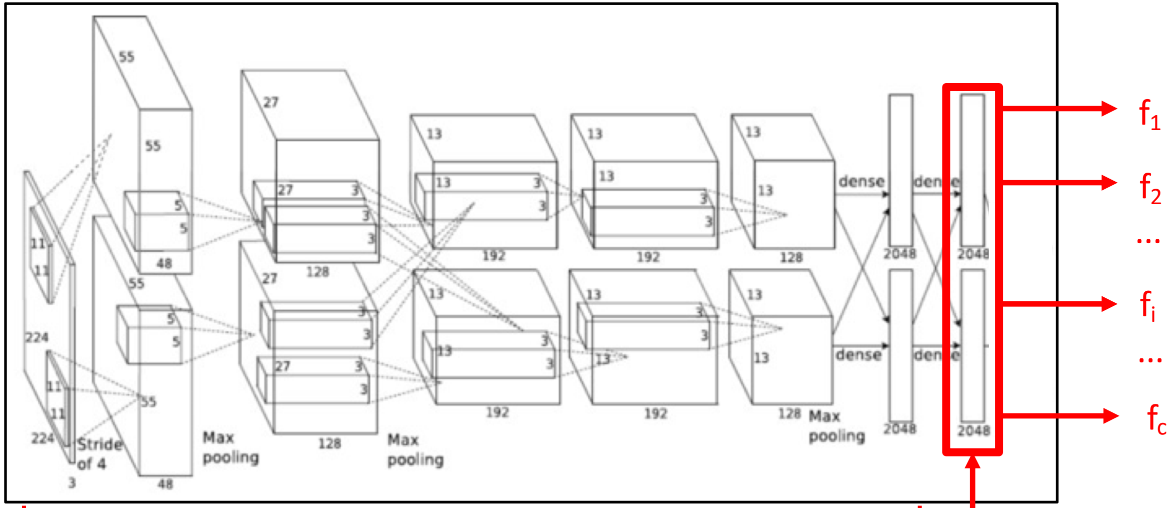
Adversarial Examples against iCub – Gradient Computation

The given optimization problems can be both solved with gradient-based algorithms

The gradient of the objective can be computed using the **chain rule**

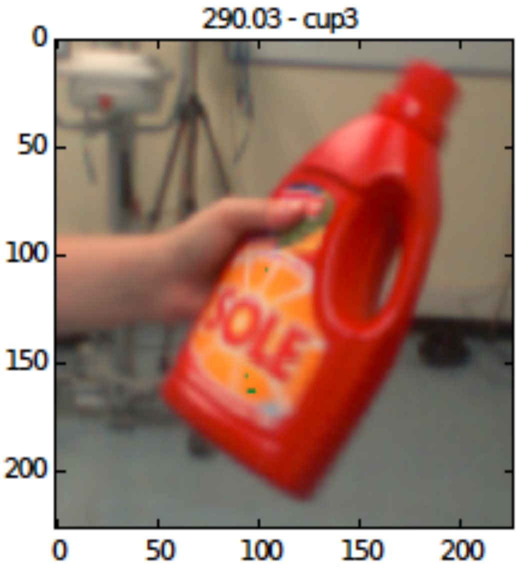
1. the gradient of the functions $f_i(z)$ can be computed if the chosen classifier is differentiable

2. ... and then backpropagated through the deep network with *automatic differentiation*



$$\nabla f_i(x) = \frac{\partial fi(z)}{\partial z}\frac{\partial z}{\partial x}$$

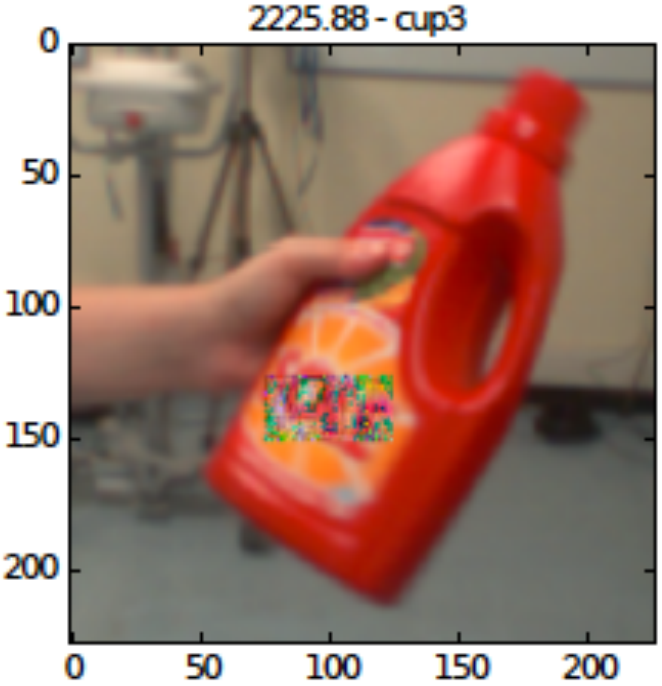
Example of Adversarial Images against iCub



An adversarial example from class *laundry-detergent*, modified by the proposed algorithm to be misclassified as *cup*

[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

The "Sticker" Attack against iCub



Adversarial example generated by manipulating only a specific region, to simulate a sticker that could be applied to the real-world object.

This image is classified as *cup*.

[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]