Pattern Recognition
and Applications Lab

Machine Learning Security

Introduction to the Course

Battista Biggio
battista.biggio@unica.it

What Number Is This?

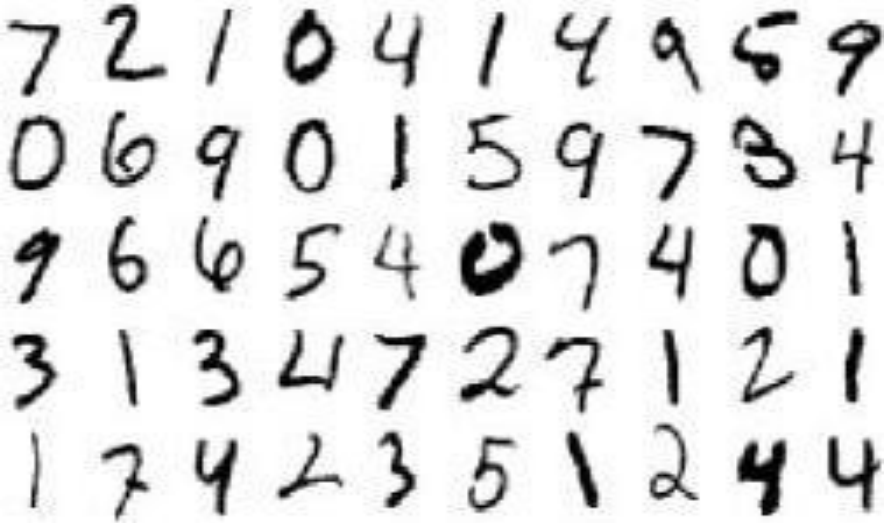
What Number Is This?

What Number Is This?

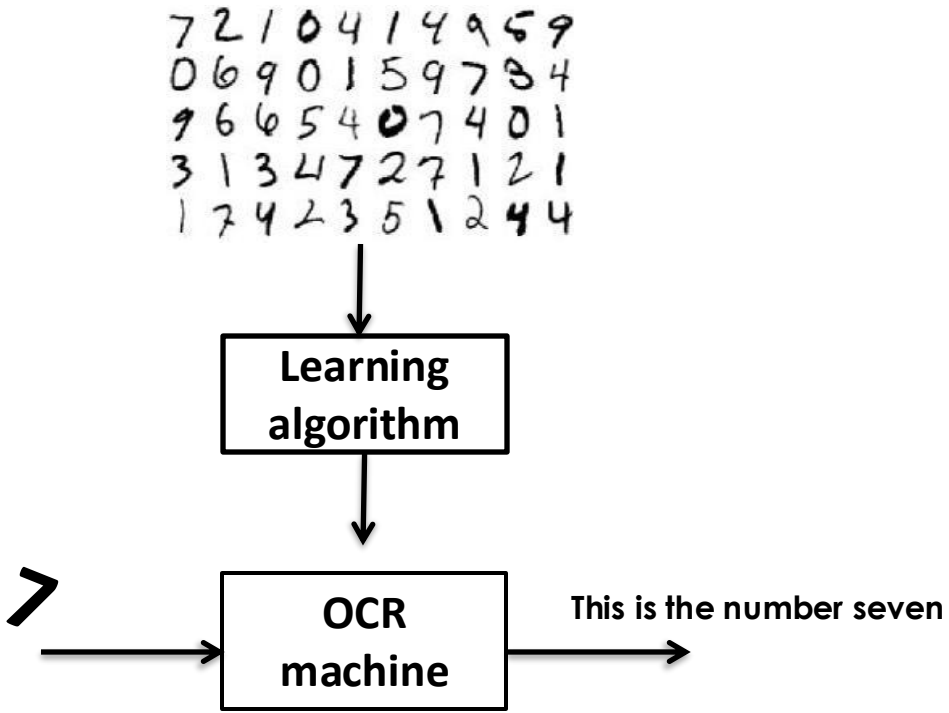
- Are you able to write in Python (or any other language) the **exact algorithm** (step after step) that you use to **recognize** the above numbers?

Writing a **deterministic** algorithm to recognize numbers from images is very difficult…

But we can collect easily many example images…



If We Could Design a Machine that Learns from Examples…



So, What Is Machine Learning?

*Machine learning is the technology that we use to solve a problem by **learning** the solution **by examples***

"The goal of machine learning is to build computer systems that automatically improve with experience"

Tom M. Mitchell, The discipline of Machine Learning, 2006

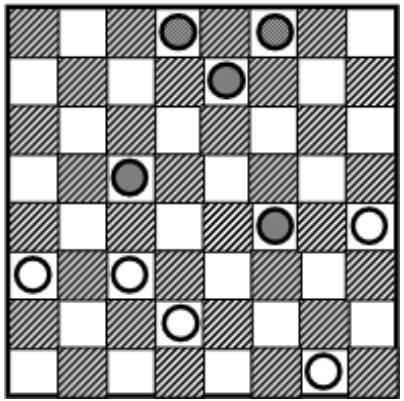
Take-Home Message

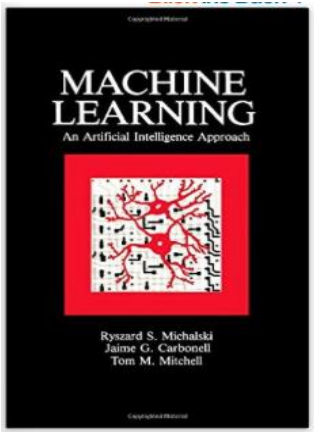
- Machine learning is very useful when **no algorithmic solution** is known. It also avoids a detailed algorithm to overfit known cases, reducing errors

- When you are able to devise algorithmic solutions (*step after step through every possible corner case*) that work 100% of the time, **you should not use machine learning**!

Machine Learning at the Beginning…

- Arthur Samuel (1959) wrote a program that **learned** to play *checkers*
  - ("draughts" if you're British)





R.S. Michalski,J.G. Carbonell,T.M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, 1985

A Question …

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: Prof. 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

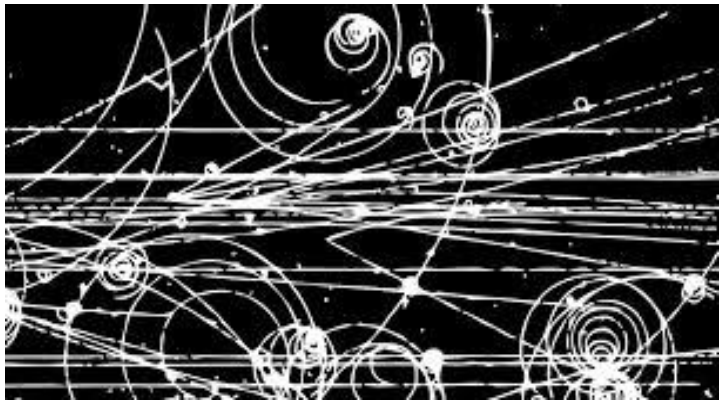
By DENIS RUTOVITZ

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

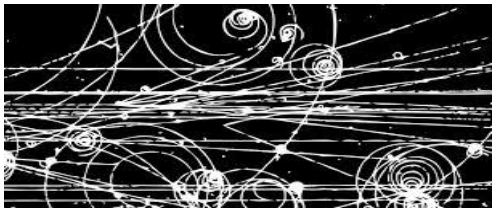
OCR for bank cheque sorting



Detection of particle tracks in bubble chambers

Aerial photo recognition

Key Feature of these Apps

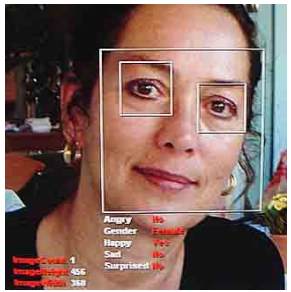


Specialised applications for **professional** users…

What is machine learning today?

It is mostly learning from (big) data for recognizing patterns

amazon.com



Google

Google Privacy & Terms

Overview | Privacy Policy | Terms of Service | Technologies and Principles | FAQ | My Account

Technologies

Advertising

How Google uses cookies

How Google uses pattern recognition

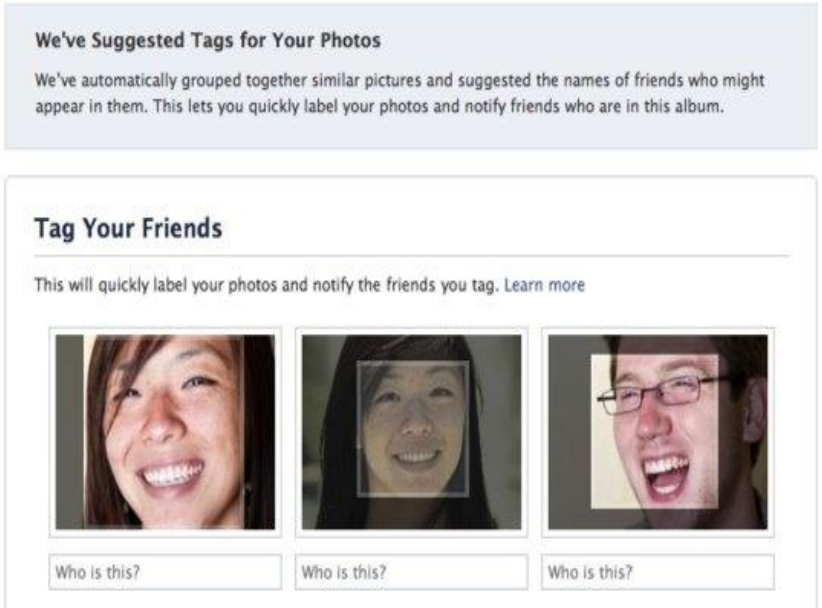
How Google uses pattern recognition

How Google uses pattern recognition to make sense of images

Computers don't "see" photos and videos in the same way that people do. When you look at a photo, you might see your best friend standing in front of her house. From a computer's perspective, that same image is simply a load of data that it may interpret as shapes and information about colour values. While a computer won't react like you do when you see that photo, a computer can be trained to recognise certain patterns of colour and shapes. For

Baidu 百度

What is ML today? It is Pattern Recognition!

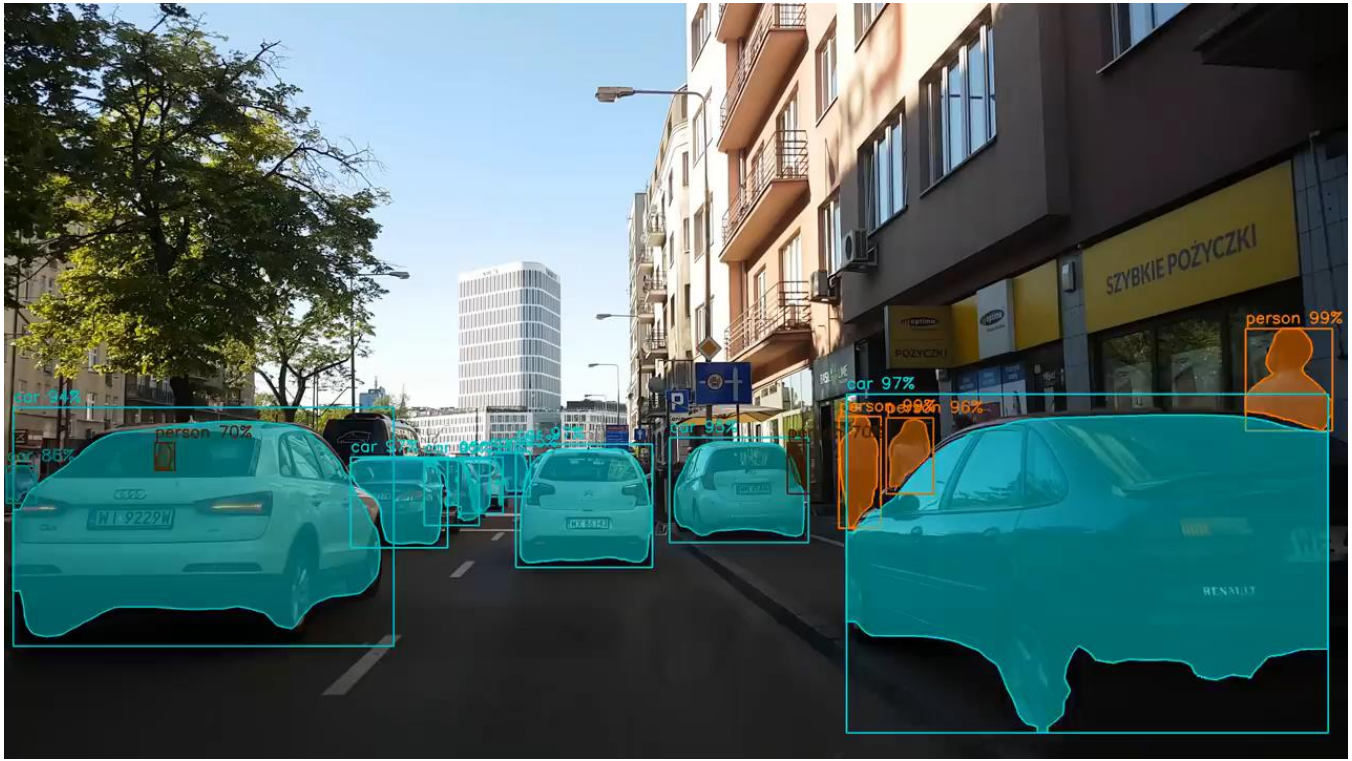


Today machine learning is pattern recognition

- Therefore, this course is focused on **machine learning** for *pattern recognition* (often called *pattern classification*)
  - i.e., on the design of **learning-based machines** for **pattern recognition**

What about Today Applications of ML?

Computer Vision for Self-Driving Cars



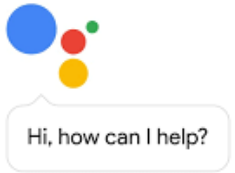
He et al., Mask R-CNN, ICCV '17, https://arxiv.org/abs/1703.06870
Video from: https://www.youtube.com/watch?v=OOT3UIXZztE

Automatic Speech Recognition for Virtual Assistants

Amazon Alexa

Apple Siri

Microsoft Cortana



Google Assistant

Today Applications of Machine Learning

Key Features of Today Apps

Personal and **consumer** applications…

Artificial Intelligence Today

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

(*Andrew Ng, Jan. 2017*)

Applications:

- Cybersecurity
- Robotics
- Healthcare
- Speech recognition
- Virtual assistants
- ...

But... What's the Difference between AI/ML?

Mat Velloso
@matvelloso

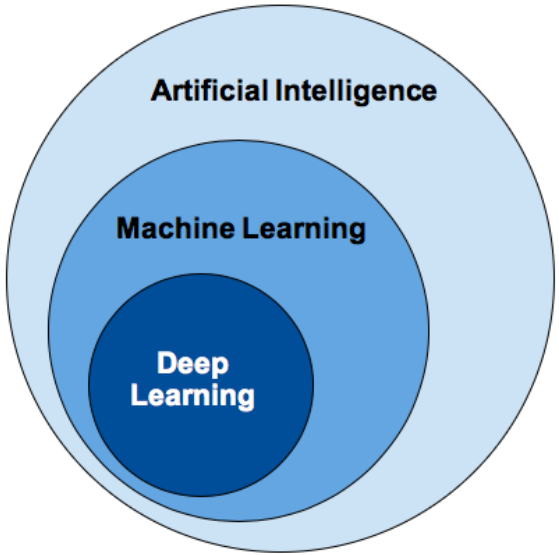
Difference between machine learning and AI:

If it is written in Python, it's probably machine learning

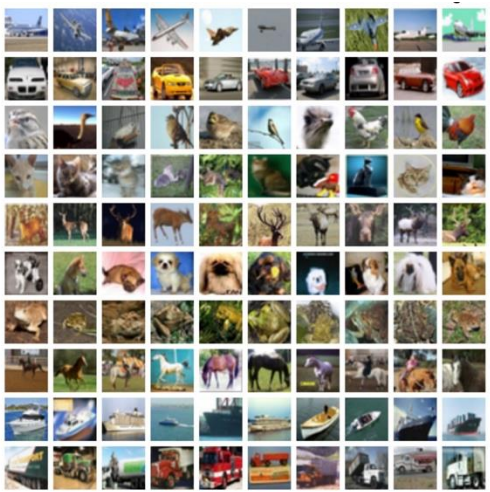
If it is written in PowerPoint, it's probably AI

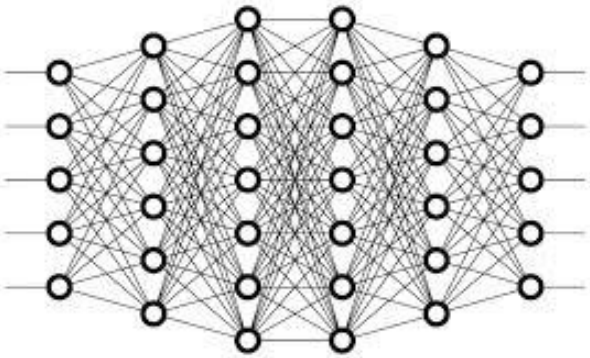
2:25 AM · Nov 23, 2018 · Twitter Web Client

8.6K Retweets **24.1K** Likes



Modern AI is Numerical Optimization + Big Data





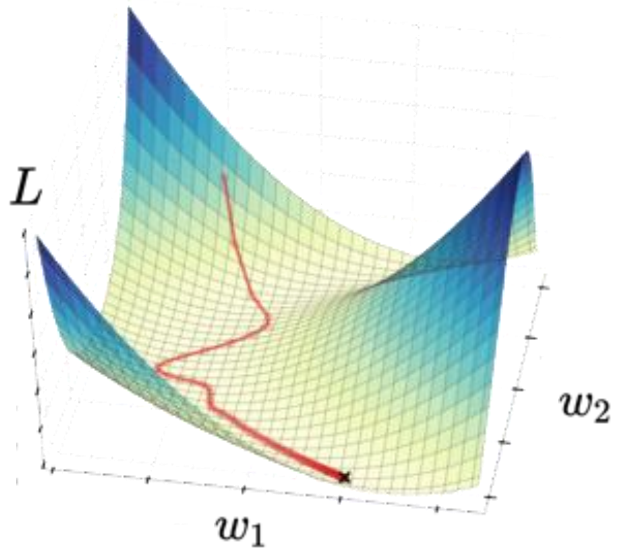
$$\min_{\boldsymbol{w}} L(D;\boldsymbol{w})$$

The goal is to minimize the fraction of *classification errors*

... by iteratively updating the classifier parameters $\boldsymbol{w}$ along the gradient direction $\nabla_{\boldsymbol{w}} L(D;\boldsymbol{w})$

The Workhorse of Machine Learning: *Gradient Descent*

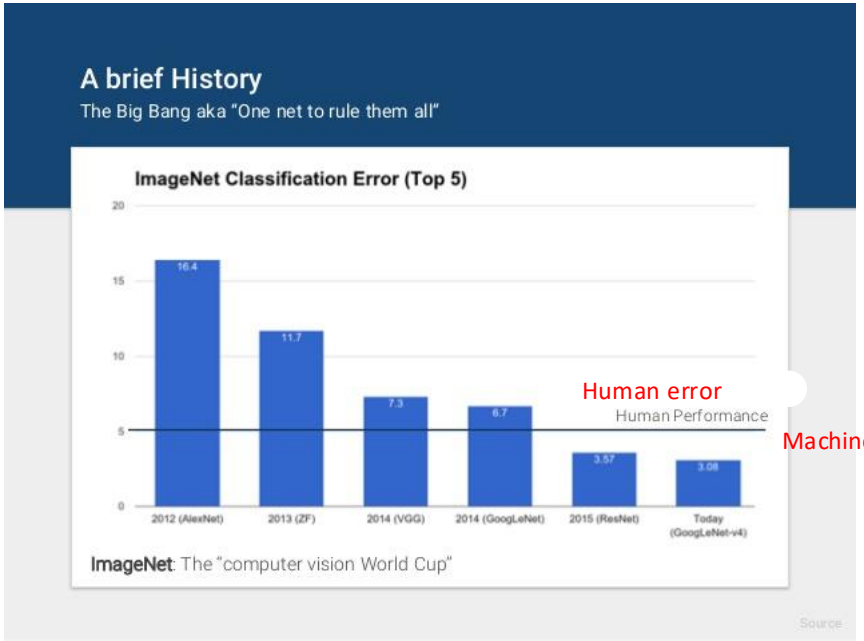
$\mathbf{w} \leftarrow \mathbf{w}_0$
$i \leftarrow 0$
while $i < maxiter$ **do**
    $\mathbf{w} \leftarrow \mathbf{w} - \eta \, \nabla_{\mathbf{w}} L(\mathbf{X}, \mathbf{y})$
    $i \leftarrow i + 1$
end while
return $\mathbf{w}$



The Bright Side of AI: Super-Human Performance

ImageNet Challenge

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.



All Right? All Good?

iPhone 5s and 6s with Fingerprint Reader… (2013-2015)

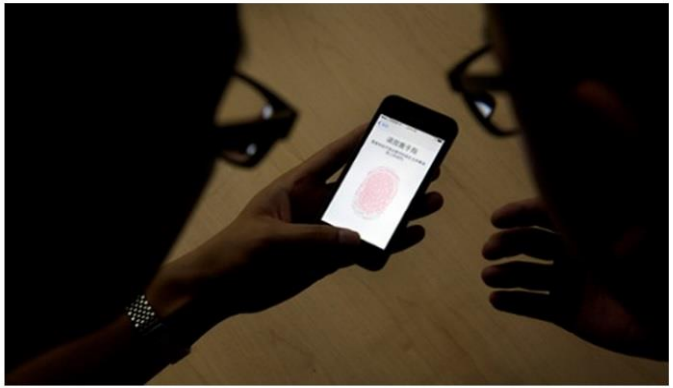
Hacked a Few Days After Release…

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

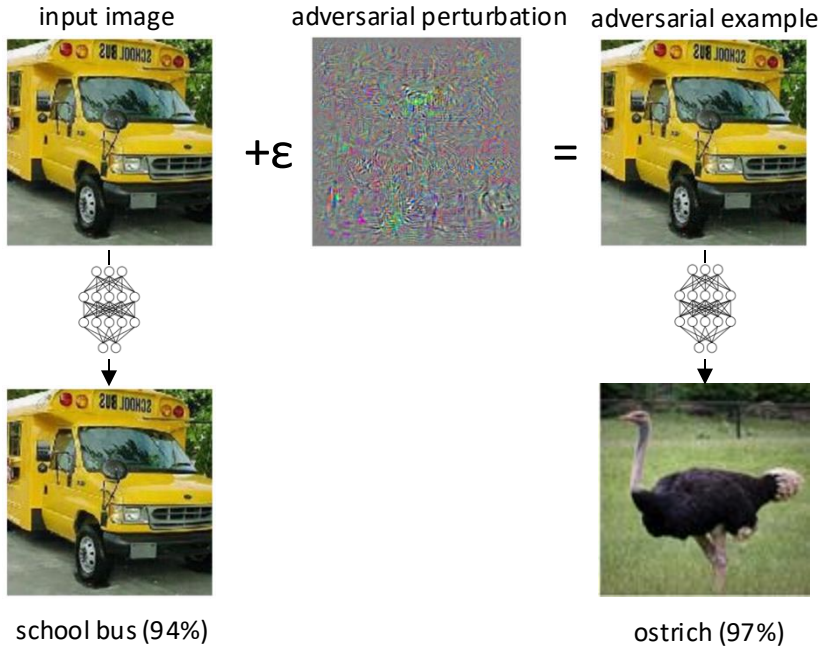
Charles Arthur
theguardian.com, Monday 23 September 2013 08.50 BST
Jump to comments (306)



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

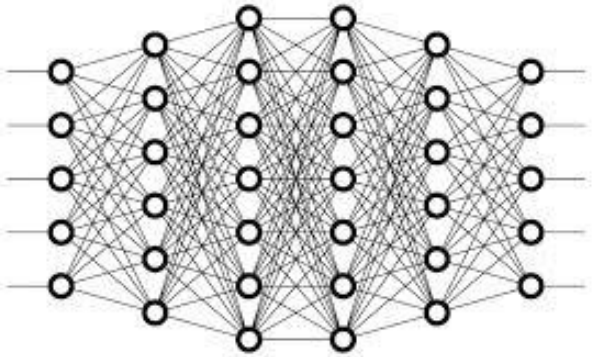
End-to-end deep learning is another story…

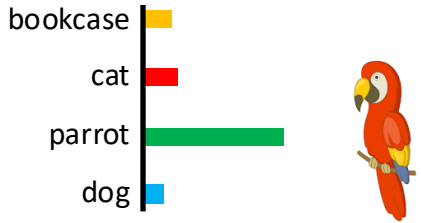
Adversarial Examples (Gradient-based Evasion Attacks)



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

Adversarial Attacks

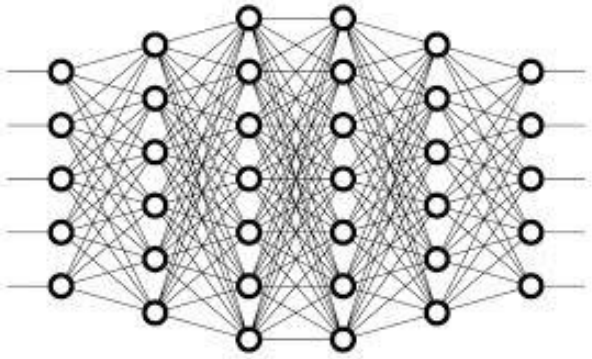


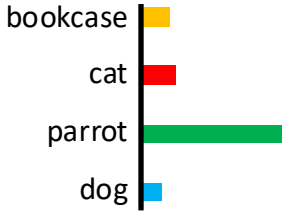


Adversarial attacks exploit the same underlying mechanism of learning, but aim to maximize the probability of error on the input data: $\max_{D} L(D; \boldsymbol{w})$

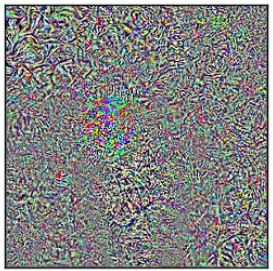
This problem can also be solved with gradient-based optimizers
(Biggio et al., ICML 2012; Biggio et al., ECML 2013; Szegedy et al., ICLR 2014)

How Do These Attacks Work?



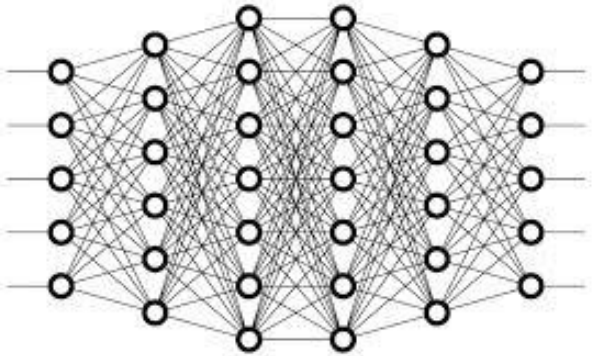


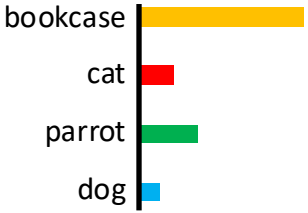
$$\max_{D} L(D; \boldsymbol{w})$$

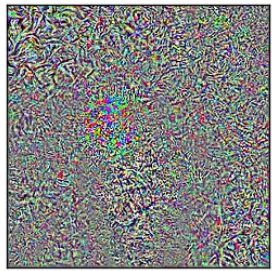


The gradient of the objective allows us to compute an *adversarial perturbation...*

How Do These Attacks Work?







... which is then added to the input image to cause misclassification

… not only in the digital domain!

Adversarial Road Signs

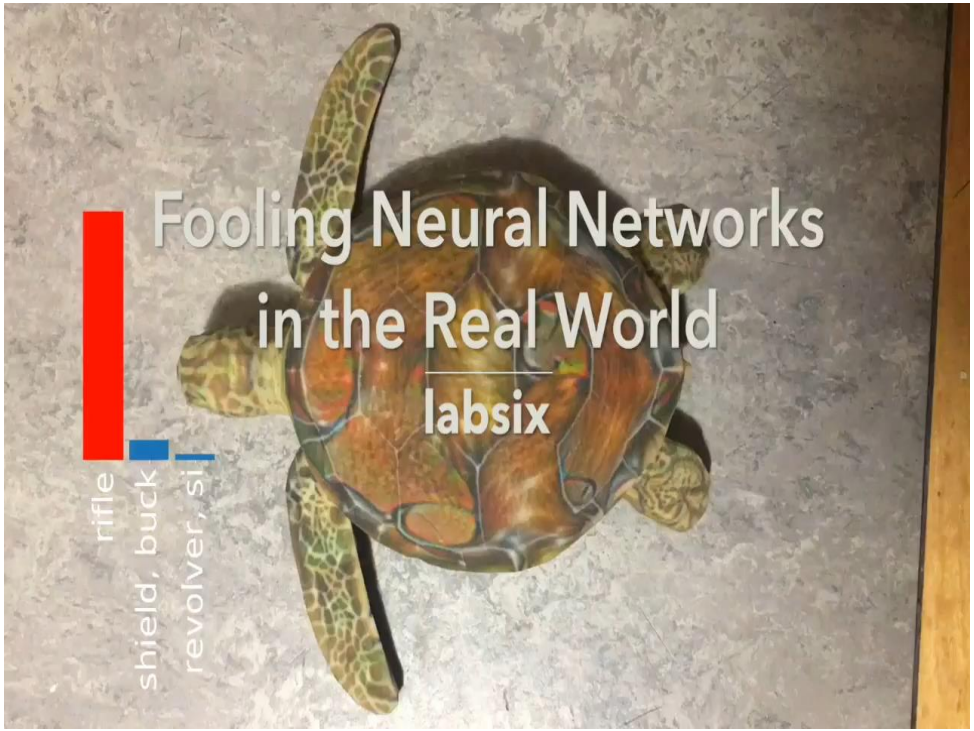
Eykholt et al., *Robust physical-world attacks on deep learning visual classification*, CVPR 2018

Adversarial Glasses

- Attacks against DNNs 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**

Sharif et al., *Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition*, ACM CCS 2016

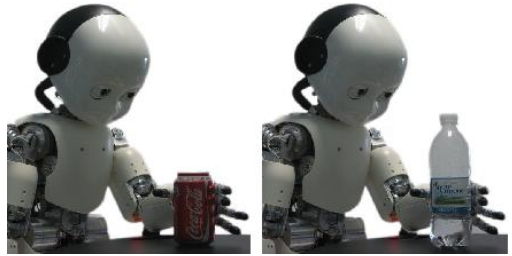
Adversarial Turtles

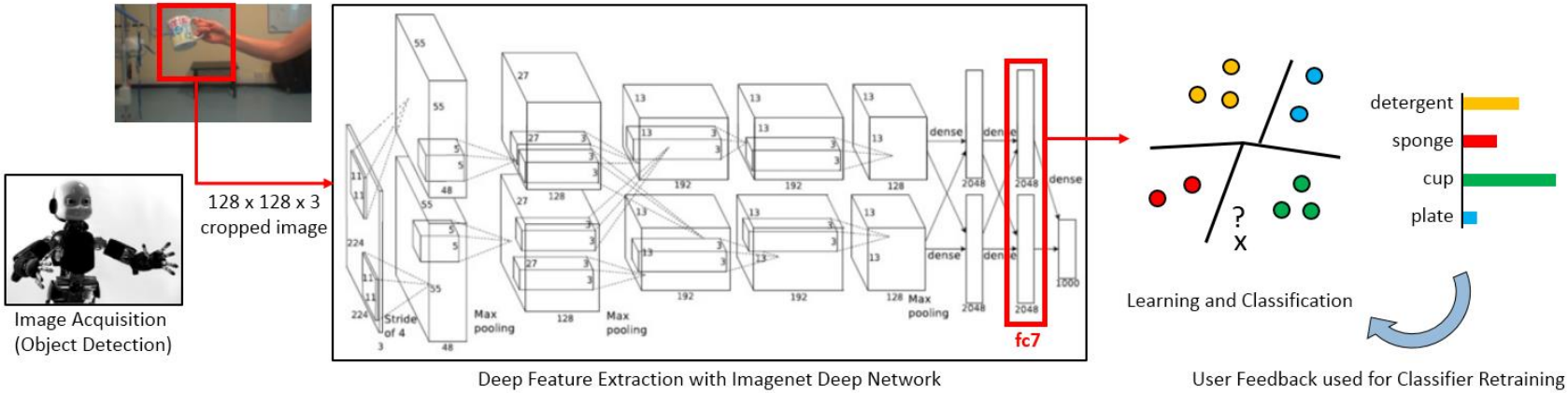


Athalye et al., *Synthesizing robust adversarial examples*. ICLR, 2018

Fooling the iCub Humanoid Robot

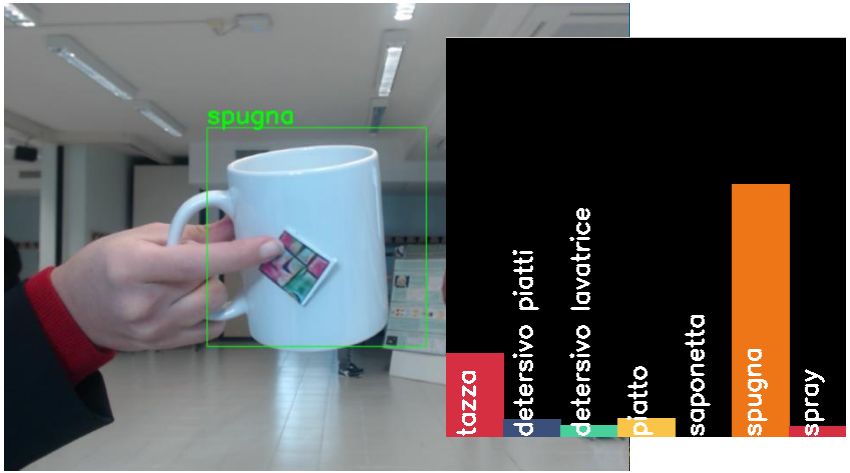
- Attacks against the iCub humanoid robot
  - Deep Neural Network used for visual object recognition

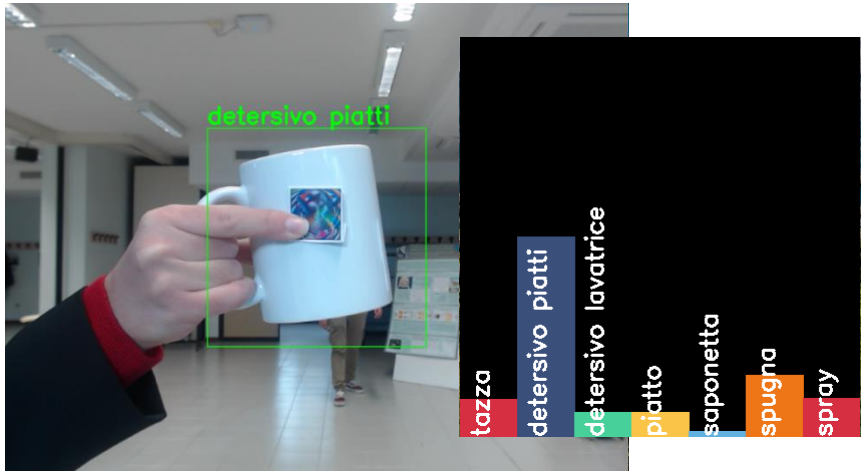




Melis, Biggio et al., ViPAR 2017

Adversarial Stickers against the iCub Humanoid Robot





ongoing work by: Sotgiu, Demontis, Biggio, Roli

Adversarial Stickers against the iCub Humanoid Robot

ongoing work by: Sotgiu, Demontis, Biggio, Roli

But maybe this happens only for image recognition…

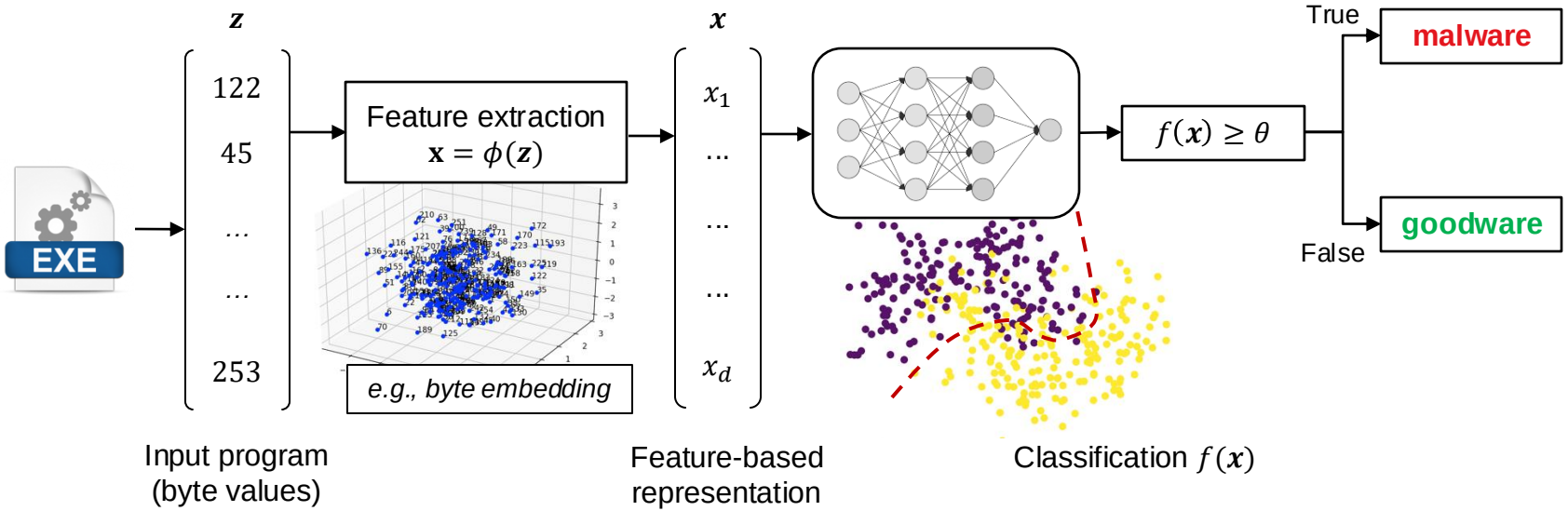
Audio Adversarial Examples

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

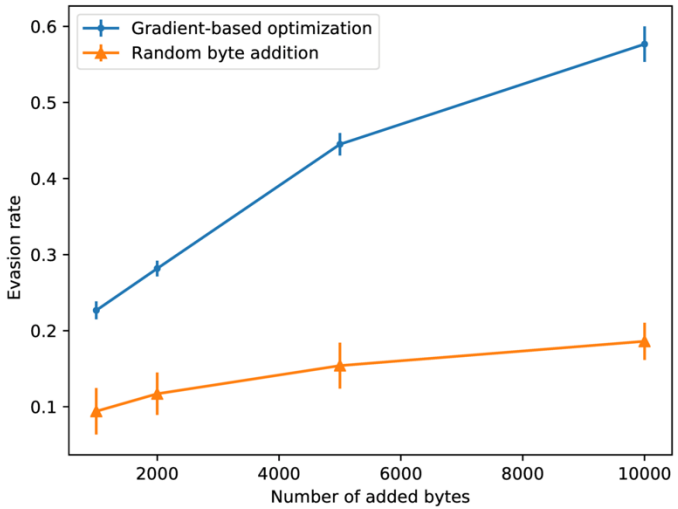
- **MalConv**: convolutional deep network trained on raw bytes to detect EXE malware

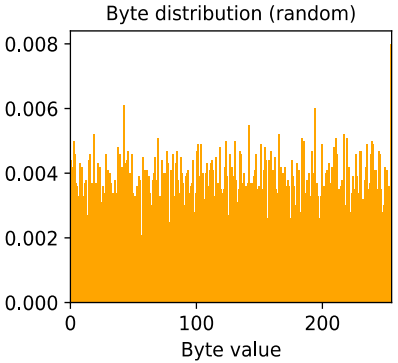


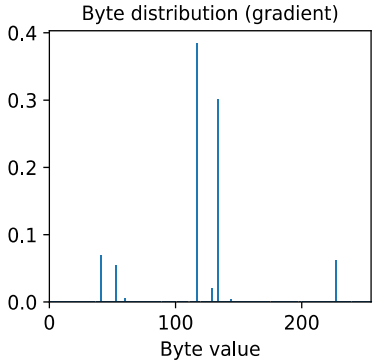
Raff et al., Malware Detection by Eating a Whole EXE, arXiv 2017

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







[Kolosniaji, Biggio, Roli et al., Adversarial Malware Binaries, EUSIPCO2018]

Adversarial Malware Examples

- PDF Malware
  - Biggio et al., *Evasion attacks against ML at test time*, ECML PKDD 2013.
  - Srndic, Laskov, *Practical Evasion of a Learning-based Classifier* ... IEEE SP 2014
  - Maiorca et al., *Towards adversarial malware detection: Lessons learned from PDF-based attacks*. ACM Comput. Surv., 2019.

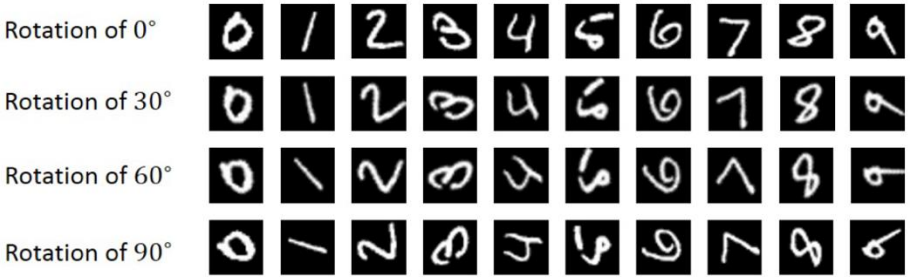
- Android Malware
  - Grosse et al., Adversarial Examples for Malware Detection, ESORICS 2017
  - Demontis et al., *Yes, Machine Learning Can Be More Secure!* ... IEEE TDSC 2019
  - Pierazzi et al., *Intriguing Properties of Adversarial ML Attacks in the Problem Space, IEEE SP 2020*

- Windows Malware
  - Demetrio et al., Functionality-preserving Black-Box Optimization of Adversarial Windows Malware, IEEE TIFS 2021 https://arxiv.org/abs/2003.13526
  - Demetrio et al., Adversarial EXEmples, ACM TOPS 2021 https://arxiv.org/abs/2008.07125
  - Demetrio, Biggio, secml-malware, https://arxiv.org/abs/2104.12848

Not Only Security Risks in AI…

Not only security risks in AI…

Lack of Robustness…



Bias, Discrimination, Fairness….

The EU AI Act for Trustworthy AI

AI is good …

- For citizens
- For business
- For the public interest

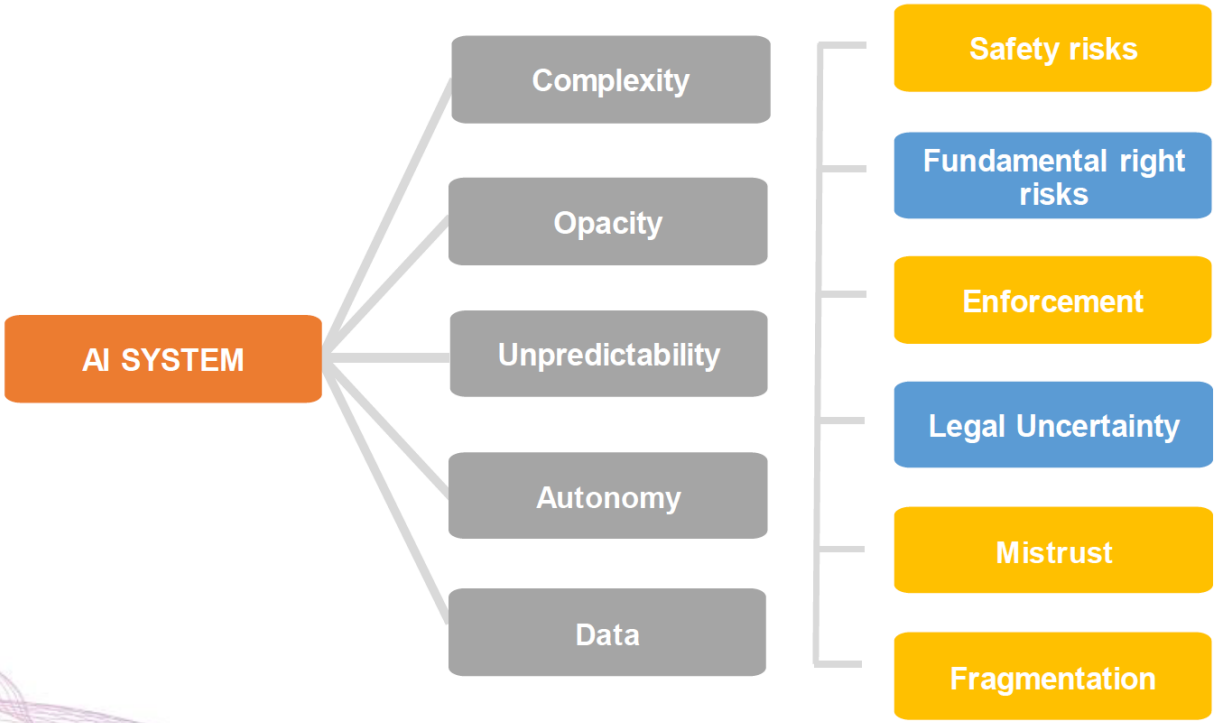
… but creates some risks

- For the safety of consumers and users
- For fundamental rights

European Commission

Lucilla Sioli, A European Strategy for Artificial Intelligence, 2021

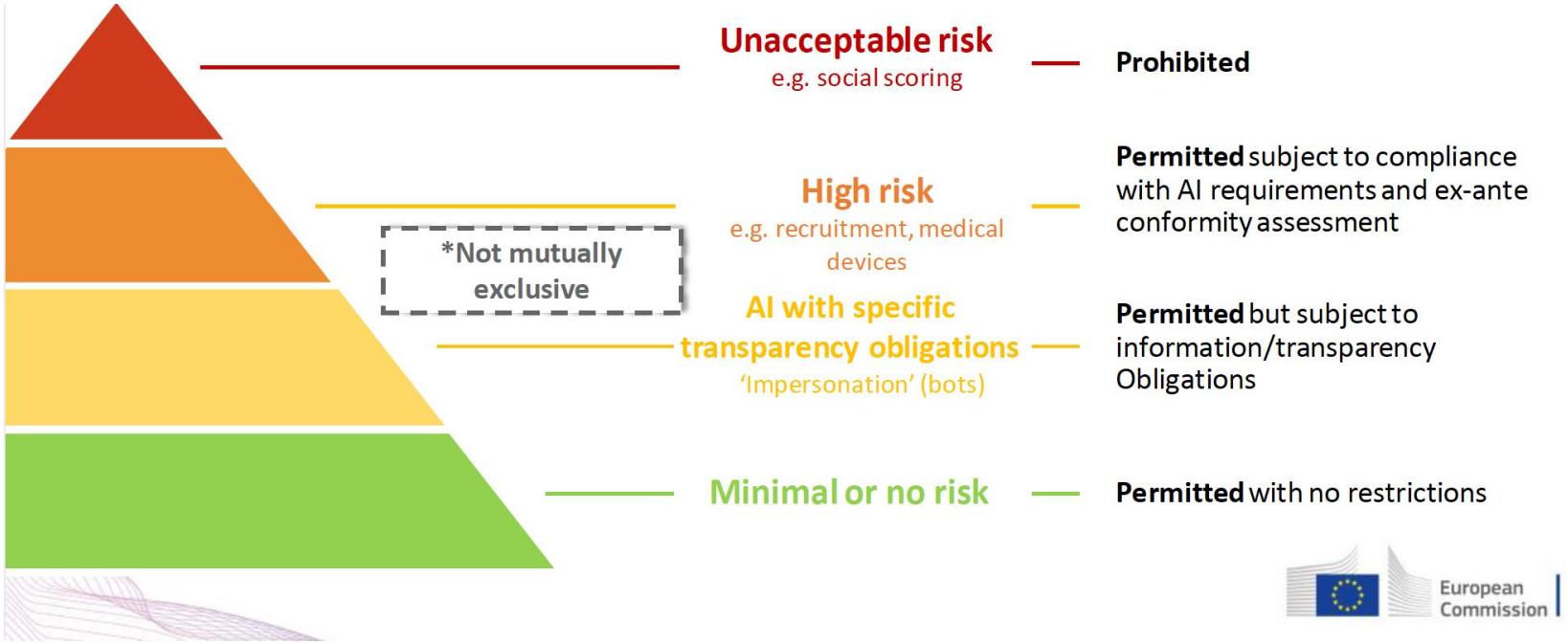
Why Should We Regulate AI?



European Commission

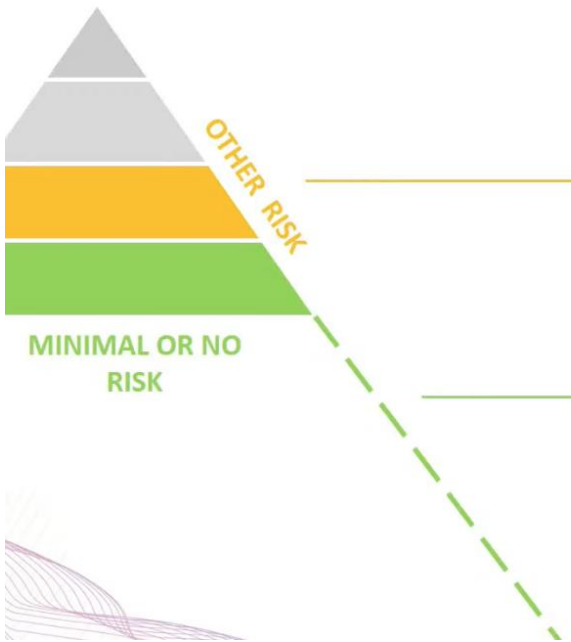
Lucilla Sioli, A European Strategy for Artificial Intelligence, 2021

The EU risk-based Approach to AI Regulation



Lucilla Sioli, A European Strategy for Artificial Intelligence, 2021

The EU risk-based Approach to AI Regulation



New transparency obligations for certain AI systems

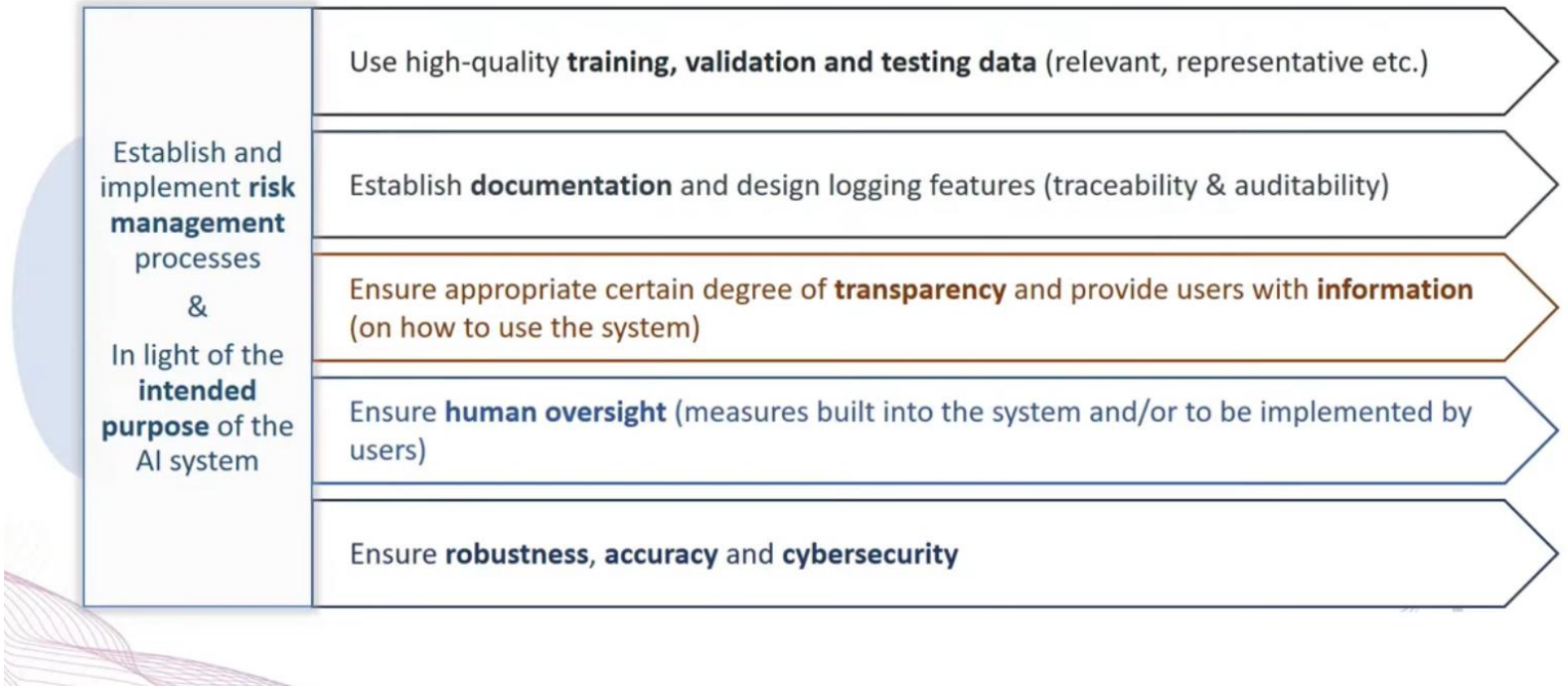
- **Notify humans** that they are **interacting with an AI system** unless this is evident
- Notify humans that emotional recognition or biometric categorisation systems are applied to them
- Apply **label to deep fakes** (unless necessary for the exercise of a fundamental right or freedom or for reasons of public interests)

Possible voluntary codes of conduct for AI with specific transparency requirements

- No mandatory obligations
- Commission and Board to encourage drawing up of codes of conduct intended to foster the **voluntary application of requirements to low-risk AI systems**

Lucilla Sioli, A European Strategy for Artificial Intelligence, 2021

The EU risk-based Approach to AI Regulation



Lucilla Sioli, A European Strategy for Artificial Intelligence, 2021

The 7 European Key Requirements for Trustworthy AI

1. **Human agency and oversight**, including fundamental rights, human agency and human oversight
2. **Technical robustness and safety**, including resilience to attack and security, fall back plan and general safety, accuracy, reliability and reproducibility
3. **Privacy and data governance**, including respect for privacy, quality and integrity of data, and access to data
4. **Transparency**, including traceability, explainability and communication
5. **Diversity, non-discrimination and fairness**, including the avoidance of unfair bias, accessibility and universal design, and stakeholder participation
6. **Societal and environmental wellbeing**, including sustainability and environmental friendliness, social impact, society and democracy
7. **Accountability**, including auditability, minimization and reporting of negative impact, trade-offs and redress

https://ec.europa.eu/futurium/en/ai-alliance-consultation/guidelines/1.html

Take-home Message

- We are living exciting time for AI…
  - …Our work feeds a lot of **consumer technologies** for **personal** applications…

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

Where Do These *Security Risks* Come From?

We will address this point in the next lectures

First, we recap some fundamental concepts on machine learning for pattern classification…

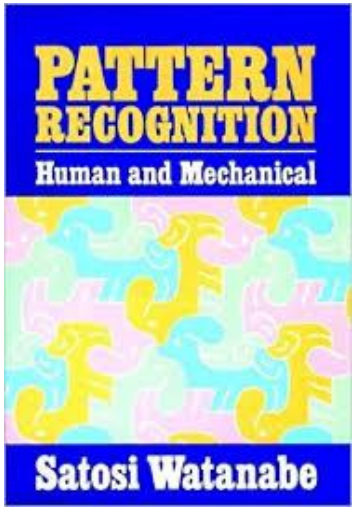
Pattern recognition as "classification"

- This course focuses on **pattern classification**. We use the term recognition instead of classification if the context makes the meaning clear and there is no ambiguity.
  - Pattern Classification: assigning a *"pattern" (a particular grouping of data)* to a category/class

In this picture, the pattern is the particular **grouping** of pixels that represent the number seven !

What is a pattern…

- **Satosi Watanabe** defined pattern recognition as *"seeing one in many"*, namely, the capability of recognizing the *unity* in the *multiplicity*, the capability of recognizing one face in a huge collection of pixels or recognizing the concept of tree despite the huge variety of sizes, shapes, and colours of the different individualities.



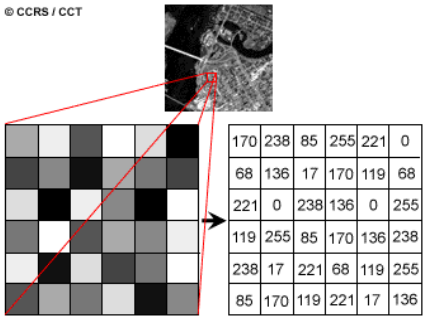
[Satosi Watanabe, Pattern Recognition: Human and Mechanical, 1985]

Pattern recognition as "classification"

Pattern classification is about assigning class *labels* to patterns.

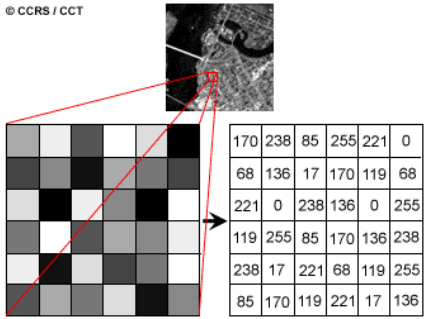
Patterns are described by a set of measurements called also **features** (or attributes, inputs).

- If we are working with image data, **feature values** could correspond simply to the **brightness** of each **pixel**.



Basic concepts: class and feature

In this course, we assume that each pattern is described by a feature vector with "d" elements: $\mathbf{x} = (x_1, x_2, \ldots, x_d)$.



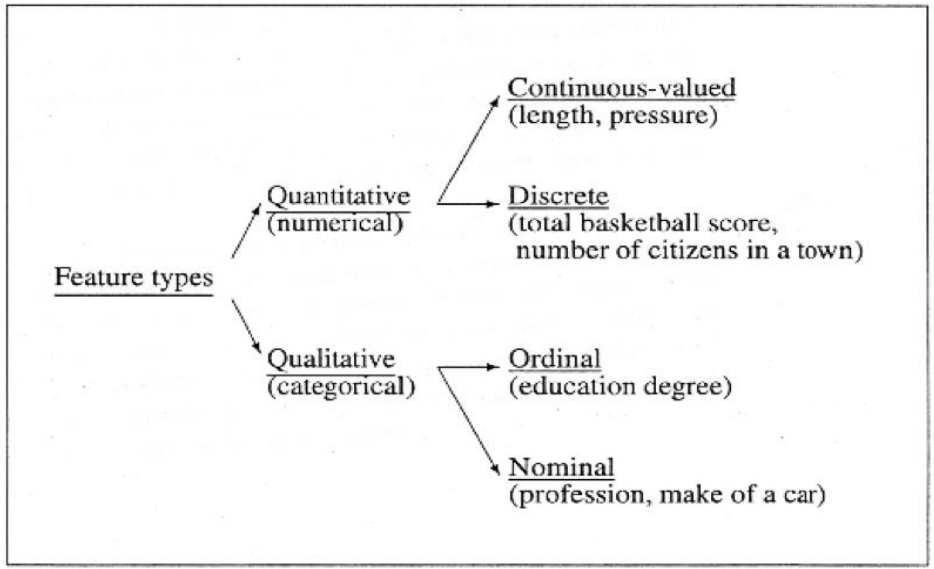
$\mathbf{x} = (x_1, x_2, \ldots, x_d) = (170, 238, 85 \ldots 136)$

Class: intuitively, a class contains similar patterns, whereas patterns from different classes are dissimilar (e.g., dogs and cars).

In this course, we assume that there are *c* possible classes, and we denote that as: $\mathbf{\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}}$, each pattern belongs to one of the "c" classes of the set Ω. We say that each pattern has a class **label**.

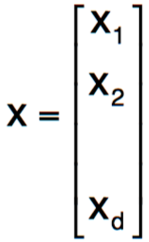
Different feature types

[L. Kuncheva, Combining pattern classifiers, Wiley, 2004]

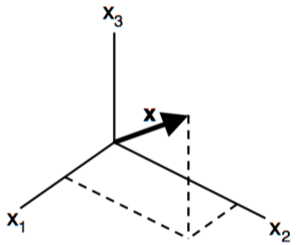


- Statistical pattern classification uses **numerical** features.

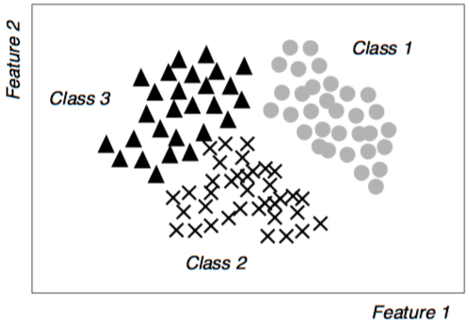
Basic concepts: feature vector, feature space



Feature vector

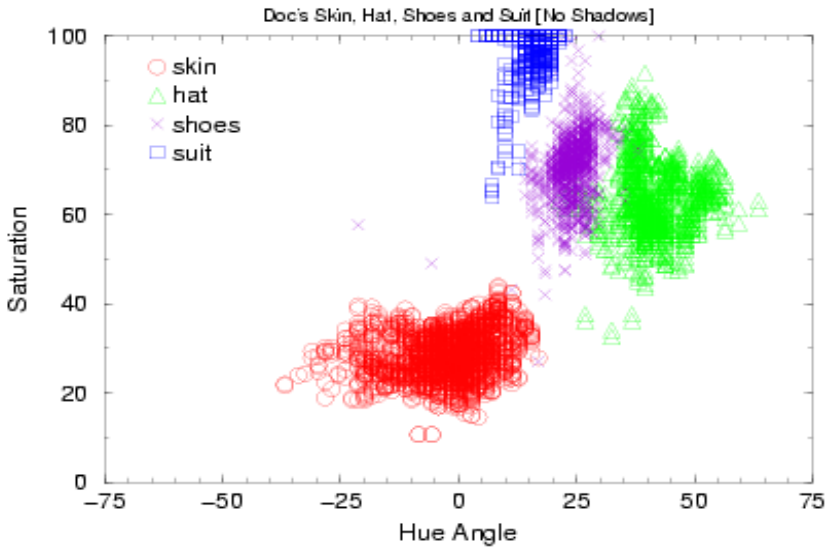


Feature space (3D)



Scatter plot (2D)

Basic concept: feature space

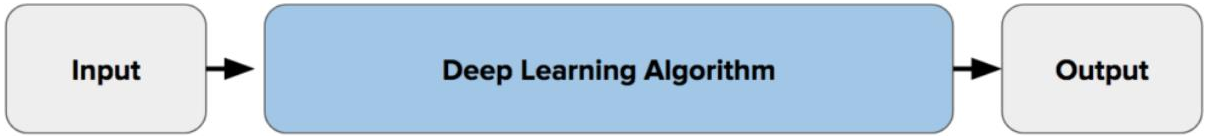


The feature values are arranged as a d-dimensional vector.
The real space is called the **feature space**, each axis corresponding to a physical feature.

Hand-crafted vs. non-handcrafted (learned) features

- In the previous example, we have seen what is named **«handcrafted»** features that are manually engineered by the human designer.
- Today, we can extract **non-handcrafted** features that are automatically learned from a machine learning algorithm.

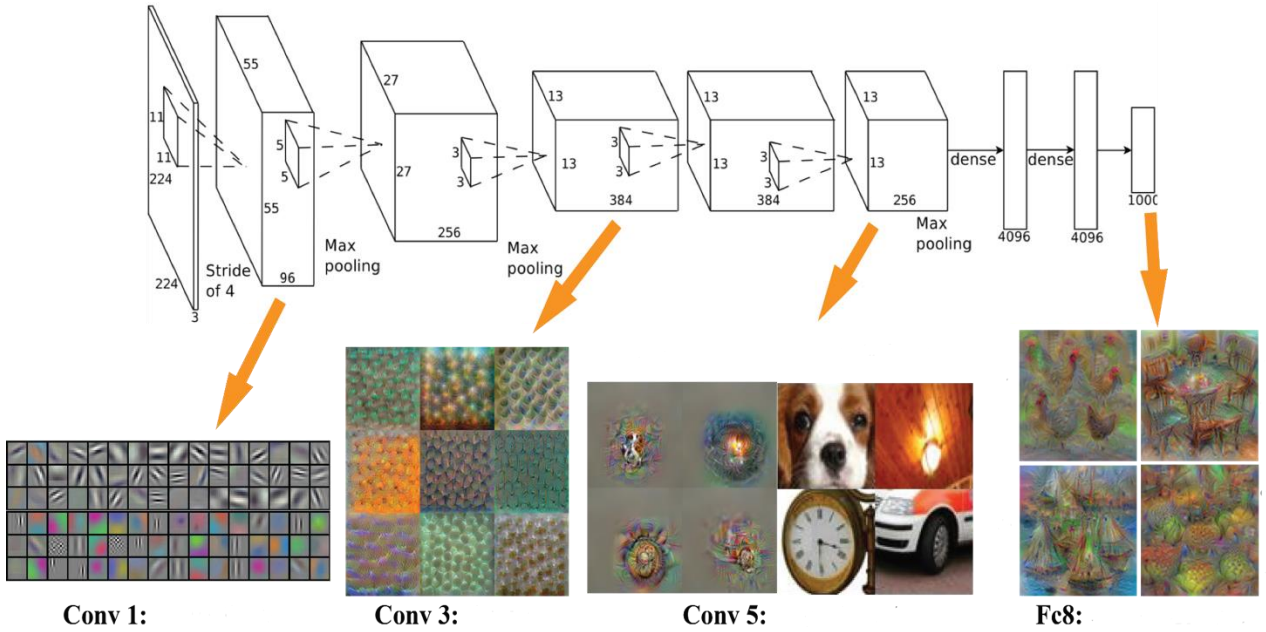
Processing flow for extraction of **handcrafted»** features



Processing flow for learning **non-handcrafted** features («learned» features)

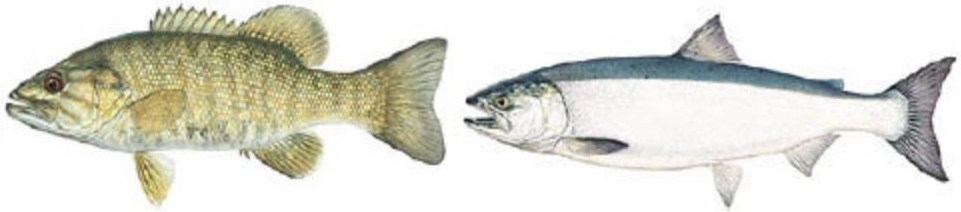
Learning non-handcrafted features

- **Non-handcrafted** features can be automatically learned with **deep neural networks**



Classification Model

- **Classification**: after the extraction of a set of features to characterize patterns, we should select a classification model using such features to classify patterns
- Let us assume that want to recognize 2 classes of fish: *salmon* and *sea bass*
- We use only one feature: length value (random variable *L*).



Sea bass Salmon

Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork, John Wiley & Sons, 2000

Classification Model

- A very simple classification model based on a simple heuristic rule could be:
  - *A sea bass is generally longer than a salmon*

- We can rewrite more formally this heuristic rule as follows:
  - if $L > L^*$ then fish=sea bass , else fish=salmon

- The threshold value L^* can be an heuristic value that we know, otherwise we should estimate it

- How can we estimate L^* ? We need a set of samples/examples of the two fish types (called "design o **training set**")

Basic Concept: Design or Training Dataset

- The information to design a pattern classifier is usually in the form of a labeled data set $\boldsymbol{D}$ (called design or training set):

$$\boldsymbol{D} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}]$$

$$\boldsymbol{x_i} = (\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, \ldots, \boldsymbol{x_{id}}) \text{ i=1,..,n}$$

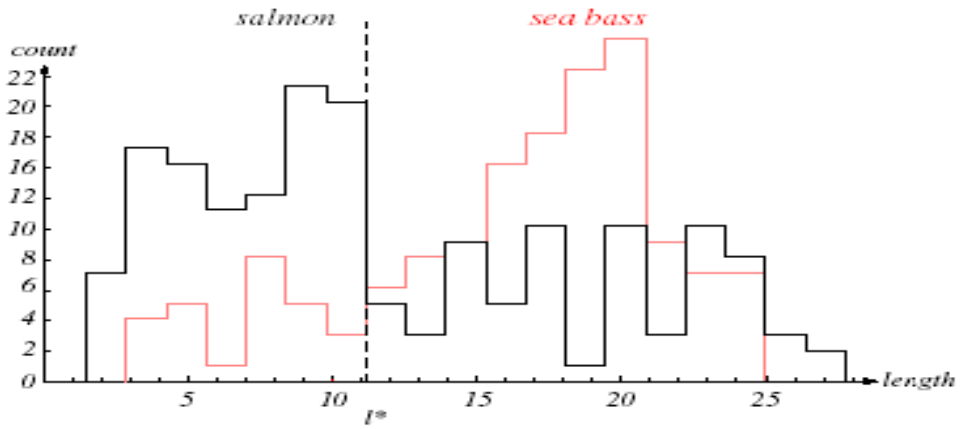
$\boldsymbol{x_i}$ belongs to one of the "c" classes ($\boldsymbol{x_i} \varepsilon \omega_j$ $j=1,\ldots,c$)

- In the previous example, $\boldsymbol{D}$ is the data set used to compute the empirical distributions of the length of the two fish types.

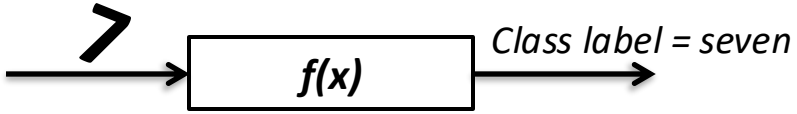
- This allows us to estimate the threshold value l* that discriminates between salmon and sea bass.

Classification Models

- This simple example suggests us a more general classification model. We could estimate the two probability functions:
  - P(length / salmon) and P(length / sea bass)
  - and then make a probabilistic decision…



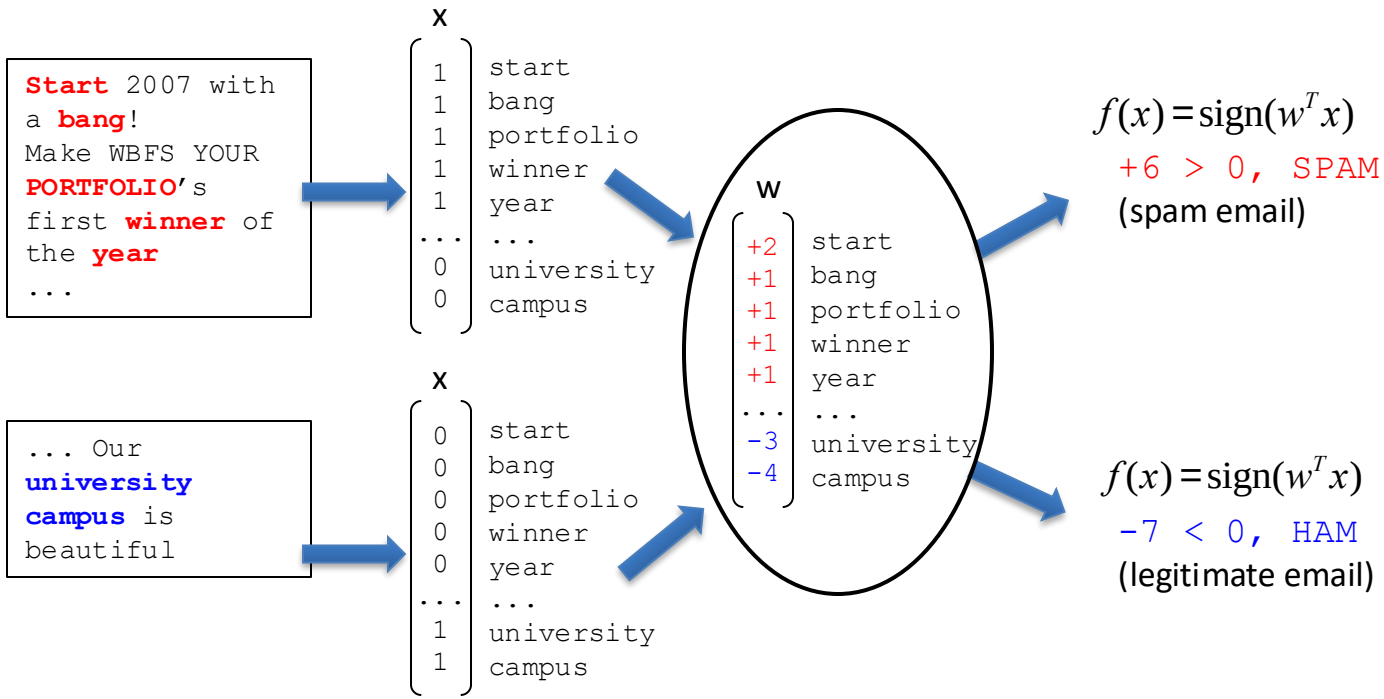
Classification models in general…



In general, a classification model can be regarded as a **function** $f(\boldsymbol{x})$, that takes as input the vector $\boldsymbol{x}$ (representing the pattern) and provides as output the classification (class label)

For example, the classification model could be a **linear function**: $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b = \sum_{j=1}^{d} w_j x_j + b$

Example: Spam Filtering



Learning as Optimization

- We said that machine learning is "learning from experience".
  - i.e., improving classification performances over time

- How do we evaluate if we are improving?

- In order to develop a formal mathematical system of learning machines, we need to have formal measures of how good (or bad) our models are.

- To this end, we use ***loss functions*** (or *cost functions)* to evaluate how good (or bad) our classification models are.

Example of loss function

$$L(D, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i; \boldsymbol{\theta}))$$

- D: training set containing «n» examples
- y_i: is the class label for training example $\boldsymbol{x}_i$
- $f(\boldsymbol{x}_i; \boldsymbol{\theta})$ is the classification model

- $\ell(y_i, f(\boldsymbol{x}_i; \boldsymbol{\theta}))$ could be the zero-one loss function
  - equal to 0 for correct predictions and 1 otherwise

$$\ell\big(y_i, f(\boldsymbol{x}_i; \boldsymbol{\theta})\big) = \begin{cases} 0, \text{classification is correct} \\ 1, \text{classification is incorrect} \end{cases}$$

Learning as an Optimization Problem

- Given a linear function $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = \Sigma_{j=1}^{\mathrm{d}} w_j x_j + b$
  - How do we estimate the classifier parameters $\boldsymbol{w}$ and b?

- Modern approaches formulate the learning problem as an **optimization problem**
  - This is generally true also for nonlinear classification functions $f(\boldsymbol{x}; \boldsymbol{\theta})$, including modern deep-learning approaches and neural networks

$$\boldsymbol{w}^\star, b^\star = \underset{\boldsymbol{w},b}{\operatorname{argmin}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i))}_{\text{loss term } L(D,\boldsymbol{\theta})} + \underbrace{\lambda\, \Omega(\boldsymbol{w})}_{\text{regularization term } \Omega(\boldsymbol{w})}$$

λ: regularization hyperparameter

Learning as an Optimization Problem

- The loss function $\ell(y_i, f(\boldsymbol{x}_i))$ measures how much a prediction is wrong
  - e.g., the zero-one loss is 0 if points are correctly predicted, and 1 if they are not

- The regularization term $\Omega(\boldsymbol{\theta})$ imposes a penalty on the magnitude of the classifier parameters to avoid overfitting and promote smoother functions, i.e., functions that change more gradually as we move across the feature space

- The hyperparameter λ tunes the trade-off between the training loss and regularization
  - Larger values tend to promote more regularized functions but with a larger training error
  - Smaller values tend to reduce the training error but learn more complex functions

Optimization Algorithms

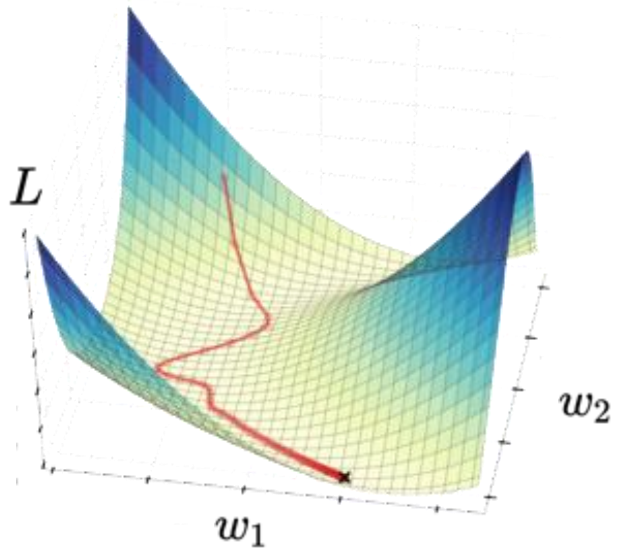
- In machine learning, we need an **optimization algorithm** (also called *solver*) capable of finding the best possible parameters that minimize the loss function

- The most popular optimization algorithms follow an approach called **gradient descent**

$$\boldsymbol{w}^\star, b^\star = \underset{\boldsymbol{w},b}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) + \lambda\, \Omega(\boldsymbol{w})$$

The Workhorse of ML: *Gradient Descent* (again!)

1: $\mathbf{w} \leftarrow \mathbf{w}_0$
2: $i \leftarrow 0$
3: **while** $i < maxiter$ **do**
4: $\quad \mathbf{w} \leftarrow \mathbf{w} - \eta \, \nabla_{\mathbf{w}} L(\mathbf{X}, \mathbf{y})$
5: $\quad i \leftarrow i + 1$
6: **end while**
7: **return** $\mathbf{w}$

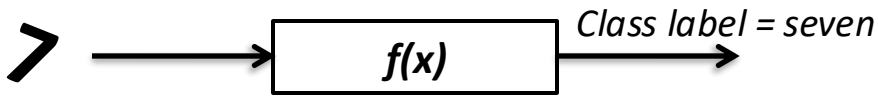


Generalization Error - Overfitting

- The best values of the model's parameters are learned by minimizing the loss incurred on a ***training set*** consisting of some number of *examples* collected for trainin
- However, doing well on the training data does not guarantee that we will do well on (**unseen**) **test** data
- So we split the available data into two partitions: the training data (for fitting model parameters) and the test data (which is held out for evaluation), and then measure:
  - **Training Error:** The error on that data on which the model was trained.
  - **Test Error:** This is the error incurred on an **unseen** test set (**generalization error**). This can deviate significantly from the training error. When a model performs well on the training data but fails to generalize to unseen data, we say that it is ***overfitting***.

Two Main Kinds of Machine Learning

- **Supervised learning**
  - in this course, we mainly focus on this case



- **Unsupervised learning**
  - Learning from a set of unlabeled samples. The goal of **unsupervised learning** (also called "clustering") is basically to find groupings in the data ("clusters") which actually reflect the ground truth and the "natural properties" of the domain the data comes from.

Other Kinds of Machine Learning Problems

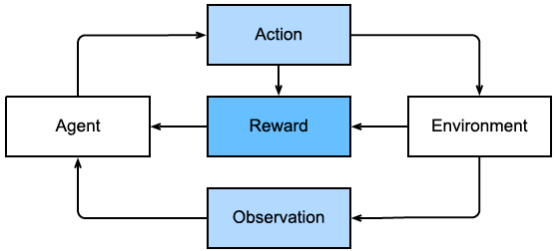
- **Regression**
  - for example, predicting the rating that a user will assign to a movie can be thought of as a regression problem
- **Tagging / Multi-label classification**
  - for example, assigning muliple lables to one image can be thought of as a tagging problem
- **Search and ranking**
  - for example, determining whether a particular web page is relevant for a user's query can be thought of as a search and ranking problem
- **Recommendation**
  - for example, providing movie recommedations to web users can thought of as a recommendation problem

https://d2l.ai, Chapter 1

Other Kinds of Machine Learning Problems

- **Sequence learning**
  - When you have a sequence of inputs and you have to provide a sequence of outputs; for example, speech recognition, text-to-speech, language translation, can be thought of as a sequence learning problems

- **Reinforcement learning (learning by interacting with an environment)**
  - Game of chess, driving a car, can be thought of as reinforcement learning problems



https://d2l.ai, Chapter 1

Course Objectives and Outcomes

- **Objectives.** Fundamental elements of machine learning security in the context of different application domains
  - Concepts and methods of adversarial machine learning, from threat modeling to attacks and defenses
  - Methods to properly evaluate adversarial robustness of a machine learning model against different attacks

- **Outcomes.** An understanding of fundamental concepts and methods of machine learning security and its applications
  - Ability to analyse and evaluate attacks and defenses in the context of application-specific domains
  - Ability to design and evaluate robust machine learning models with Python and test them on benchmark data sets

Machine Learning Security - Tentative Course Outline

Security of Machine Learning

1. Introduction and ML basics
2. Threat Modeling and Attacks on ML
3. Evasion Attacks and Defenses + lab
4. Evaluation Issues + lab
5. Poisoning Attacks and Defenses + lab
6. Backdoor and Privacy Attacks
7. Explainability + lab
8. Adversarial Malware + lab

Course Grading and Material – MLSec 5 CFU

Course Grading

- Reading group exercise / written examination (2/3 of the final mark)
- Python project (1/3 of the final mark)

Material

- A. Joseph, B. Nelson, B. Rubinstein, D. Tygar, Adversarial machine learning, Cambridge University Press, 2018
- B. Biggio, F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition 84 (2018): 317-331.

Website / Repository

- https://github.com/unica-mlsec/mlsec

Course Instructors

Instructor

- Battista Biggio

Teaching Assistants

- Maura Pintor
- Ambra Demontis
- Angelo Sotgiu

Thanks!

Battista Biggio
battista.biggio@unica.it