Université de Montréal

Département d'informatique et de recherche opérationnelle

# IFT6390
# Fondements de l'apprentissage machine

First class:

**INTRODUCTION**

Professor: Ioannis Mitliagkas

Slides adapted from: Pascal Vincent

Institut des algorithmes d'apprentissage de Montréal

MILA

# Today

- Getting to know the class (very general discussion, hardly any technical content).
- Webpage on Studium
  https://studium.umontreal.ca
- The objectives, the plan of the class, the evaluation and grading...
- Informal presentation of the domain of machine learning.

# Other machine learning classes offered at DIRO

Automonne

Graduate version of this class (in English)

IFT6390 Fondements de l'apprentissage machine

Automne

Advanced course presenting a formalism and essential techniques for learning

IFT6269 Modèles graphiques probabilistes et apprentissage

Limited capacity, very interesting premise

Automne

IFT6757 Autonomous vehicles (Duckietown)

Data Science class

Automne

IFT6758 Science des données

# Other machine learning classes offered at DIRO

Advanced course presenting cutting-edge research in artificial neura
networks and deep learning

Hiver

IFT6266 Algorithmes d'Apprentissage
/ Apprentissage de représentations

Course that goes deeper into the fundamental math

Hiver

IFT6085 Theoretical principles for deep learning

Theory course: Linear and Multi-linear algebra in ML

IFT6760A Linear algebra/Spectral Methods in ML

Hiver

New Reinforcement Learning course

IFT6760?

FIRST PART

# Course plan

## and other practical information

# You can find information..

On the StudiUM webpage:
**https://studium.umontreal.ca**

If you are registered, in the menu «mes cours» (my courses) you should see:

IFT6390-A-A19
Fondements de l'apprentissage machine
If not, your registration might not be yet finalized.

# Flavor of the class

**The course requires some comfort on both mathematics & computer science**

- Linear algebra
  (vectors, matrices, ...)
- Probability, statistics
  (random variable, distribution, expectaton, ...)
- Analysis
  (partial derivatives...)
- Algorithms
  (and complexity)
- Data structures
- Programming
  (Python environment +numpy+matplotlib...)

There will be reminders for essential mathematical notions ...

# THIS CLASS IS HARD

Goal for first 2-3 weeks:
Decide if this class is for you

I will point you to resources to cover any gaps in math and programming.

Labs will help!!

==> HEAVY first month for some of you

Overall, graduate class with lots of work

# Grade breakdown

Homework (20%):
3 sets of homework + data competition

Lab midterm exam (15%):
Individual programming exam (Oct 9th)

Midterm exam (20%):
Individual theory exam (Oct 30th)

Group project (20%):
Runs from late September to end of the semester

Final exam (20%):
Theory exam on all material (likely date Monday, December 16th, will announce)

# Flavor of the class

- Educational material comes from various sources (pay attention to mathematical notations!).
- The course will be given in english
  ⟶ Details in next slide

# Language

- Graduate class
  - Largely international students
  - Research performed in English
  - Conferences, workshops, journals
- All of the material will be in English, however we can accommodate:
  - Homework and exams can be submitted in French

SECOND PART

# Informal presentation of the domain of machine learning

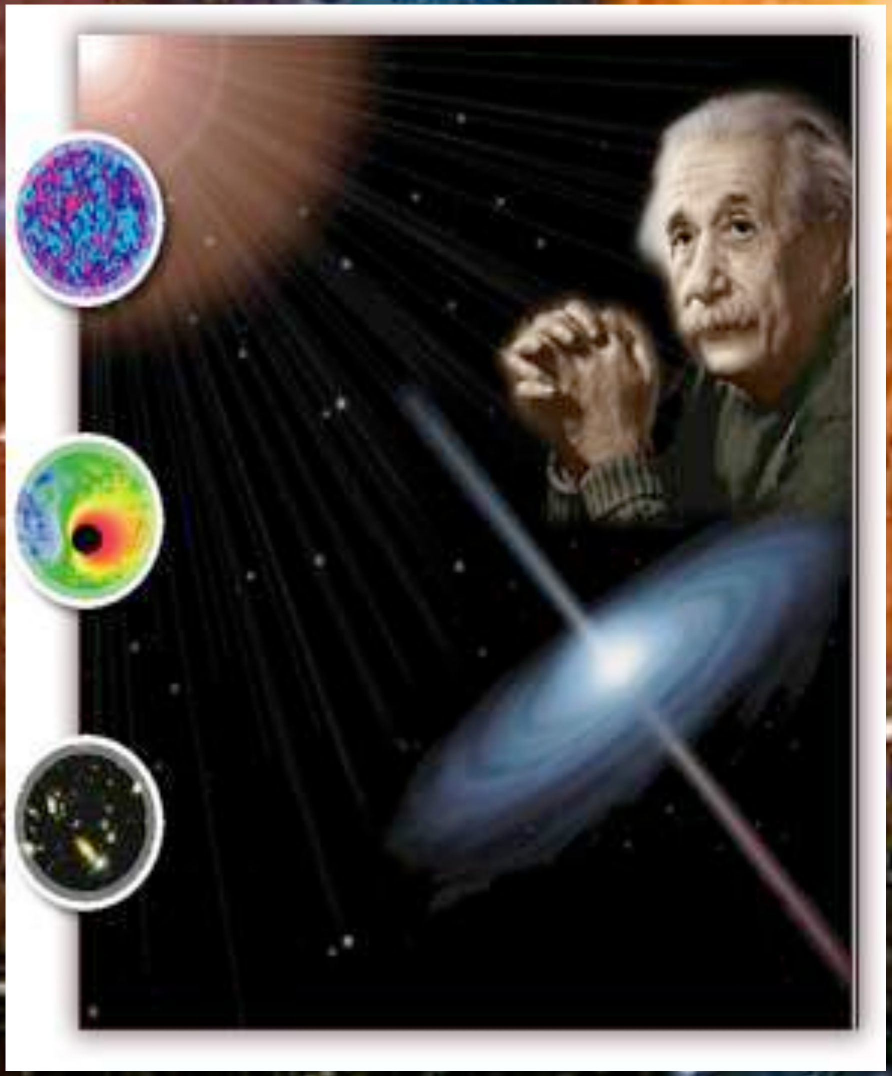
# On the schedule today

- The role of learning in modern Artificial intelligence.
- The founding disciplines of learning.
- The domains of application of learning.
- Examples of types of problems in learning.
- Data représentations.

# Scientific curiosity

# Three great mysteries

The universe,
space / time
energy / matter



Life

Intelligence,
Consciousness

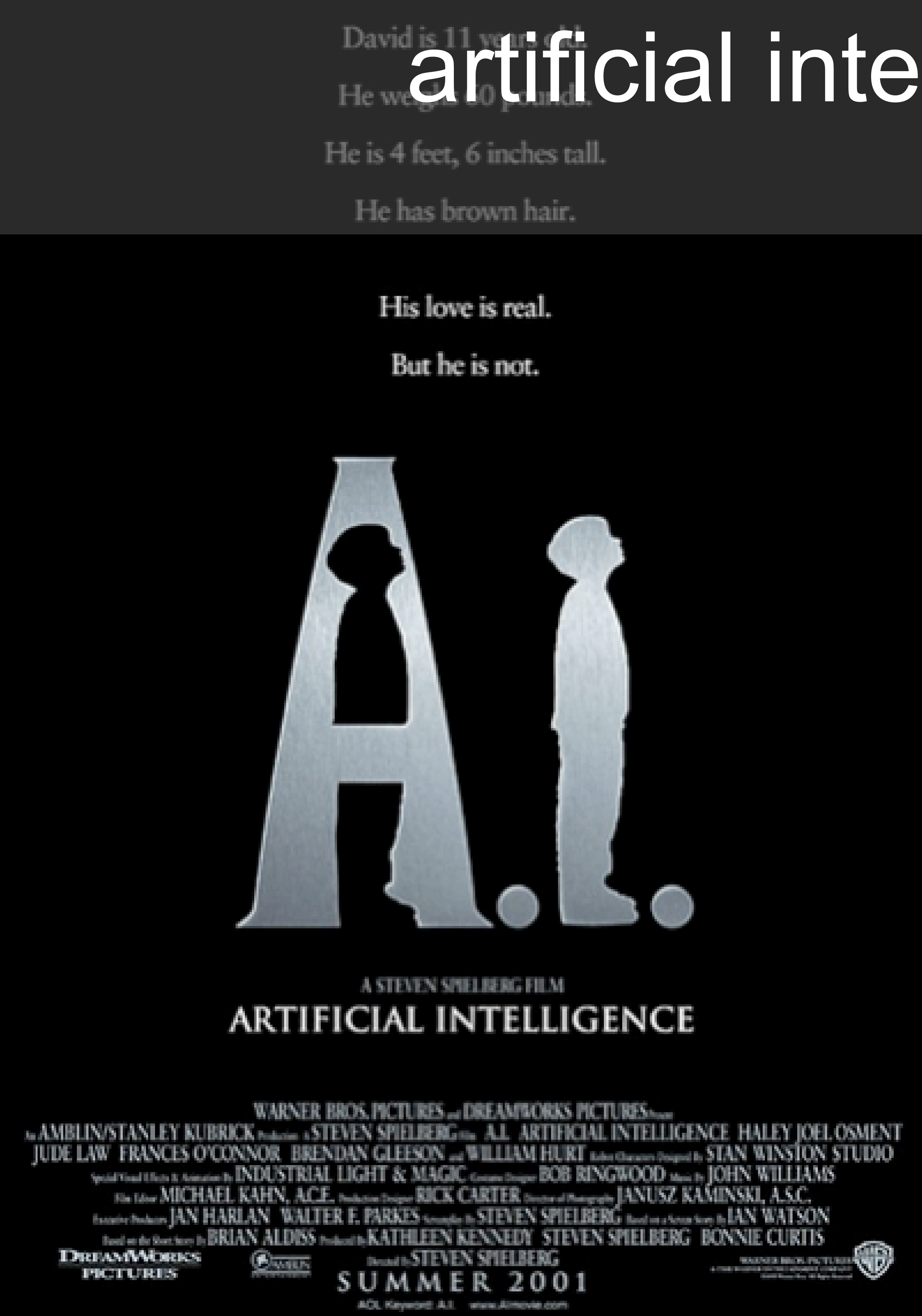
# Natural intelligence: a brain that learns, adapts

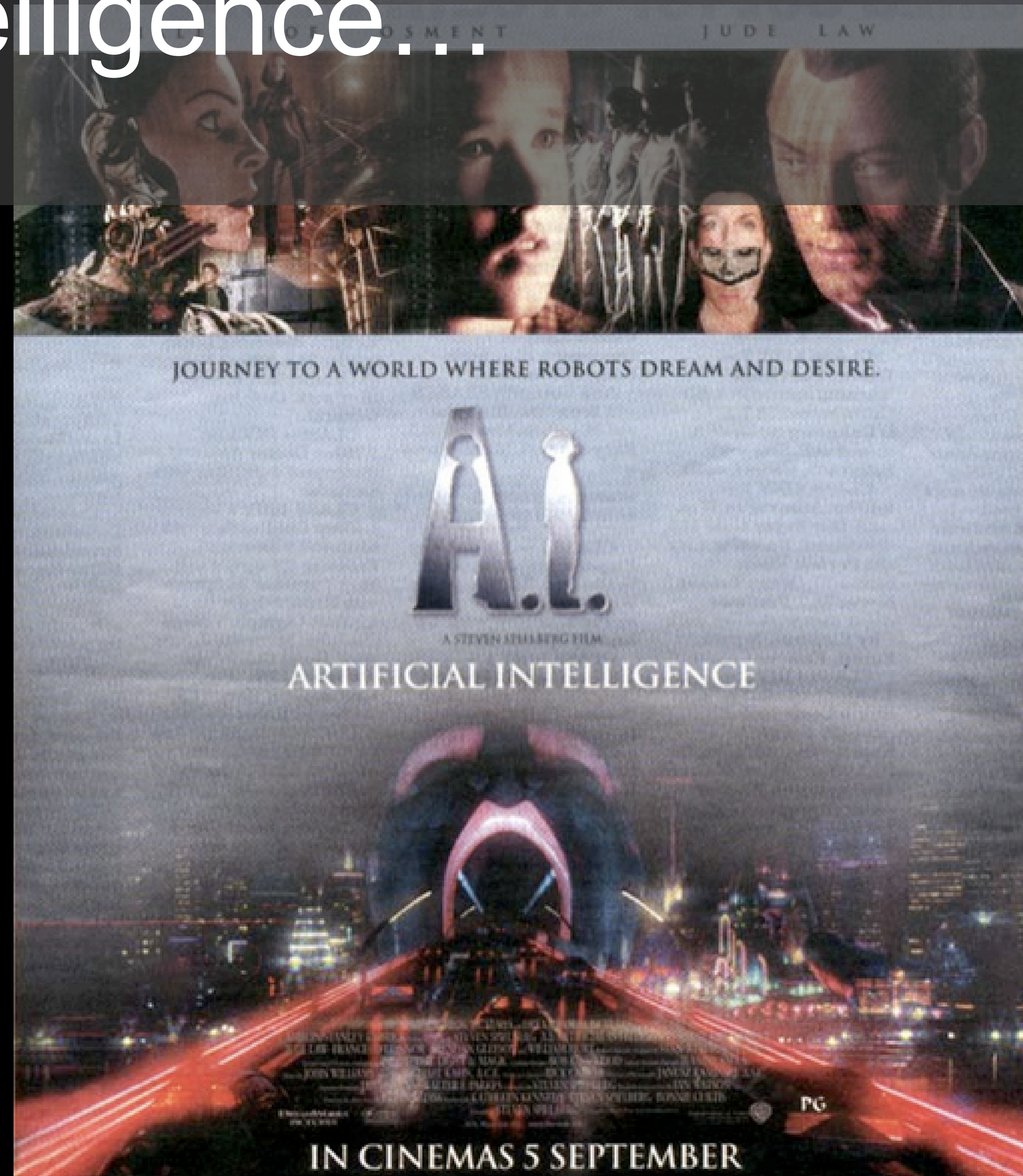
- $10^{11}$ neurons,
  $10^{14}$ synapses

- Complex network of neurons

- Learning: the modification/adaptation of synapses

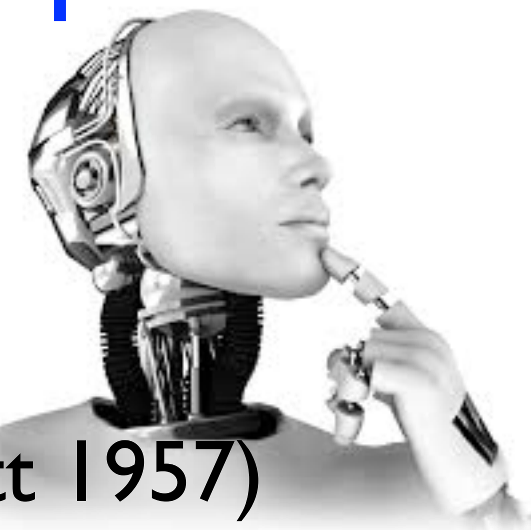
# artificial intelligence…

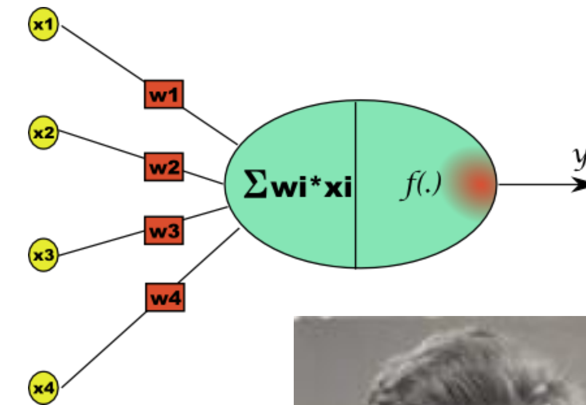


# The origins of machine learning

## Historical perspective






- Born from an ambitious goal:
  **Artificial Intelligence**

- <u>Founding project:</u>
  The **Perceptron** (Frank Rosenblatt 1957)
  First artificial neuron **learning** from examples

- Two approaches historically different in AI

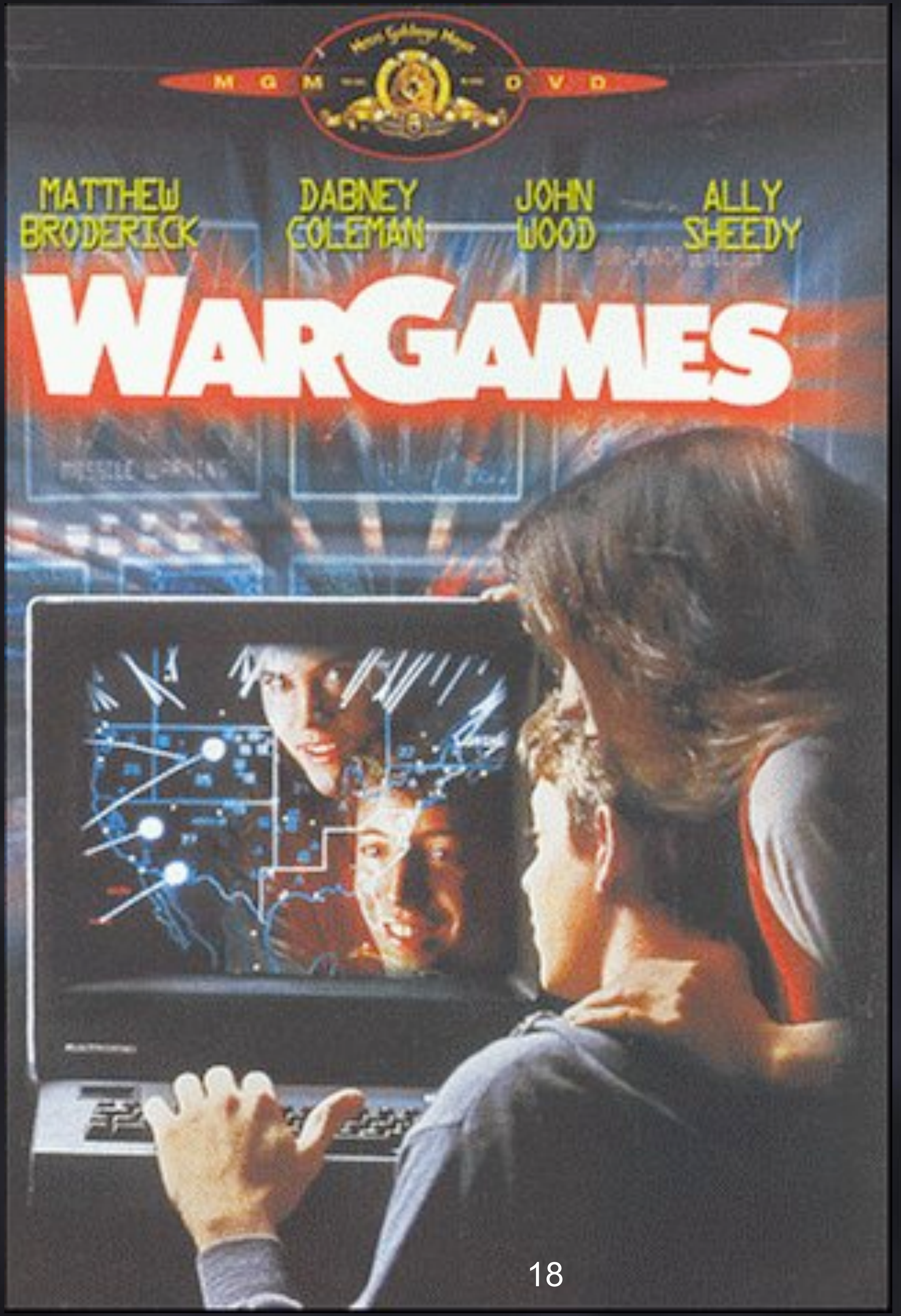
**Inspired by the brain:**
➪ network of neurones
learning from examples
➪ artificial perception.

**«Classic» AI is symbolic:**
Centered around logical reasoning
➪ <u>No learning</u> (hand-coded rules)
➪ <u>no handling of uncertainty</u>
Finally added with Bayes Nets...

Learning and probabilistic models have largely won
➪ machine learning (apprentissage machine)

# A.I. in science fiction

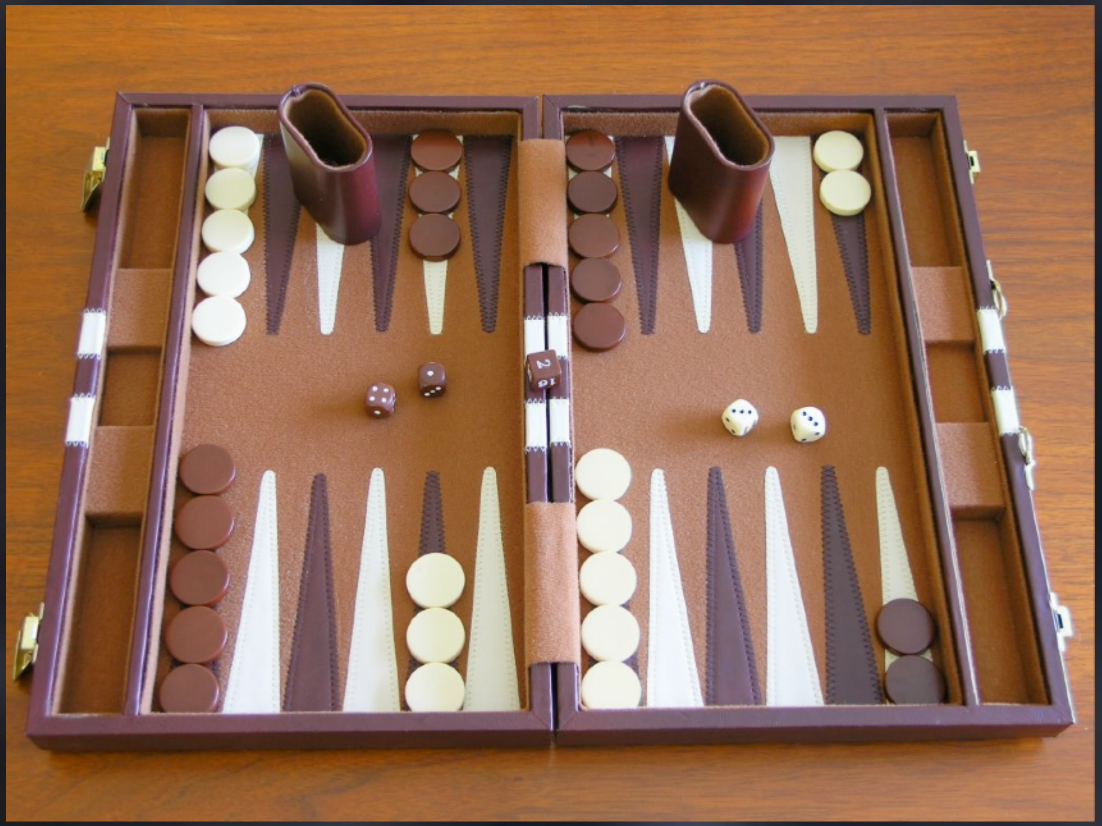
- 1983: In WarGames, a computer learns by playing against itself to play tic-tac-toe and do “global thermonuclear war”.



# in reality…

## Backgammon

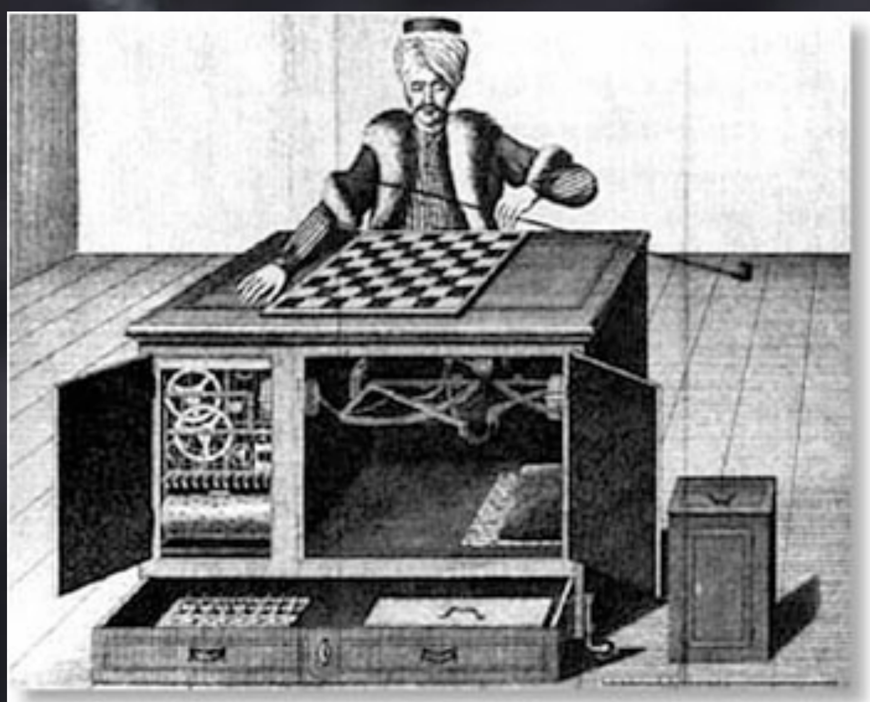
- 1995: TD-gammon, an artificial neural network trained by playing 200 000 games of backgammon against itself, plays at a level equivalent to the best players in the world (Tesauro 1995).



# In chess

(Example of success of a “classic” AI approach)

1770: «Mechanical turk» automatic chess player



Won against Napoleon Bonaparte and Benjamin Franklin

A hoax!

1997: Garry Kasparov vs «Deep Blue» of IBM

# At Geopardy

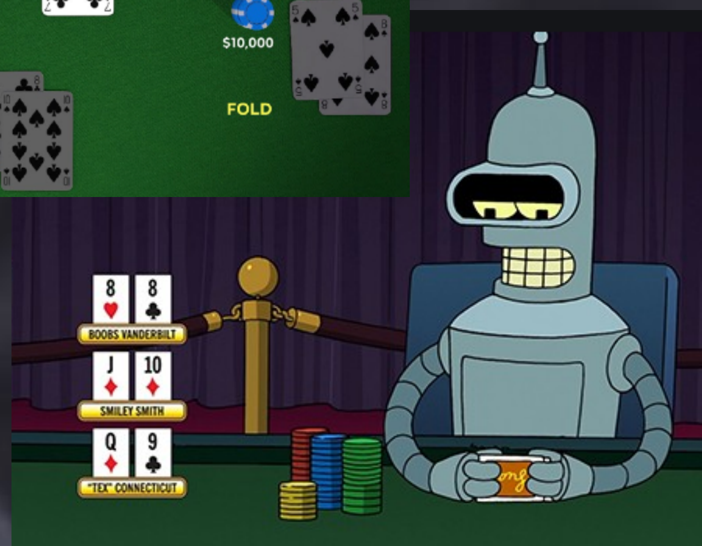
- February 2011: Watson, an IBM system, defeats the human champions of Geopardy.

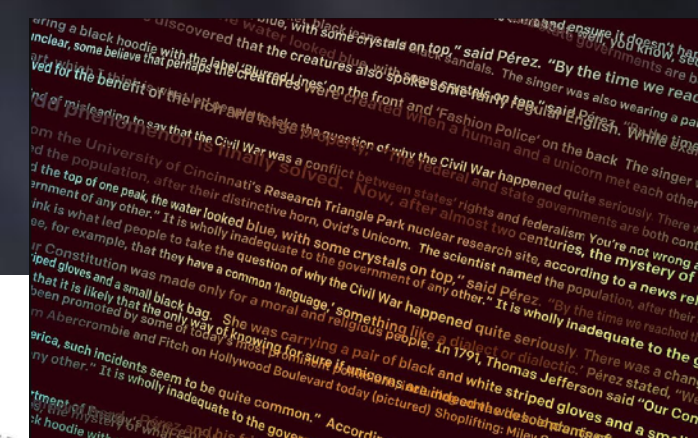

- Used machine learning on large database of textual data.

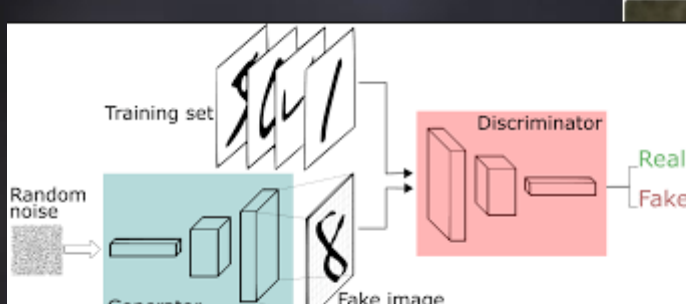
# March 2016

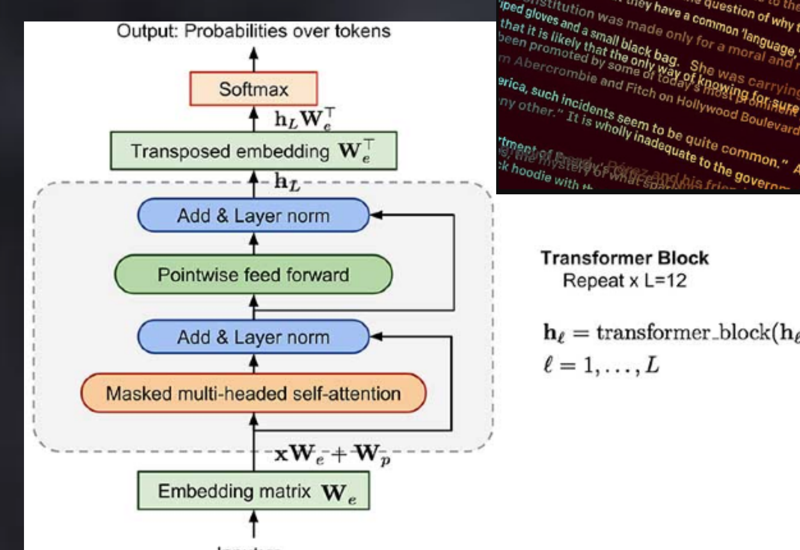

GO: one of the rare table games where human champions still had a large lead.
Not anymore!

# More recent advances



# Learning is at the heart of modern successes of AI

## And it is not just for games

- Google search
- Computer vision systems
- Voice recognition (eg. Siri)
- Smart product recommendations: Netflix / Amazon / ...
- Autonomous vehicles
- Autonomous robots etc...

# Applications

- Traditional applications: recognizing forms/ patterns
  - handwriting, speech, fingerprints
  - expert humans do these well
  - **moderate** amount of data, number of attributes, numbers of classes
  - **moderate** noise and ambiguity

# Applications

- Modern applications:
  - data mining, large scale text mining, financial predictions, ranking web hits (Google), analysis of genetic expression
  - expert humans do not exist
  - **enormous** amount of data, number of attributes, numbers of classes
  - **increased** noise and ambiguity

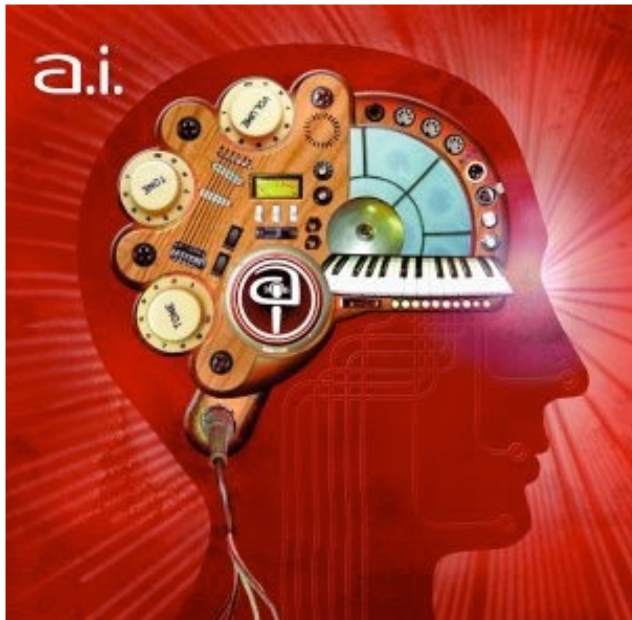
# Applications

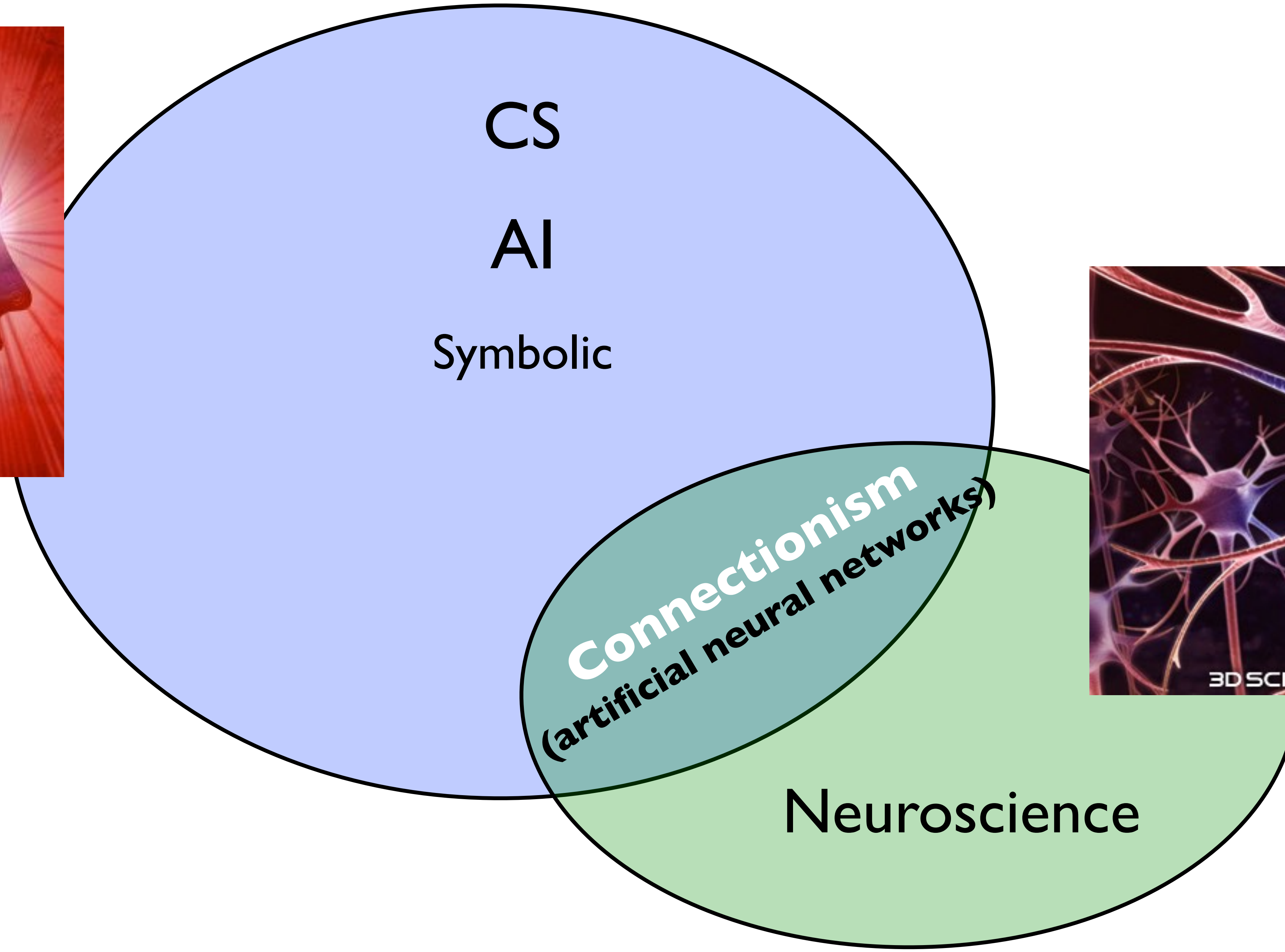
- Pattern recognition
  - handwriting
  - speech
  - fingerprints
  - images
- Mining text
  - Google
  - text classification
- Natural language processing
  - predicting the next word
  - disambiguation of meaning
  - predicting the part of speech (POS)
- Software engineering
  - Predicting stability
  - Better testing (*)
  - Security (*)

# Applications

- Data mining
  - direct marketing
  - prediction of bonuses
- Bioinformatics
  - predicting the risk of cancer
  - detecting disease
  - drug discovery
- Collaborative filtering
  - product recommendations
  - personalized medicine

# The vision of AI
# in **1957** (Rosenblatt, “Perceptron”)




# The role of learning in modern AI

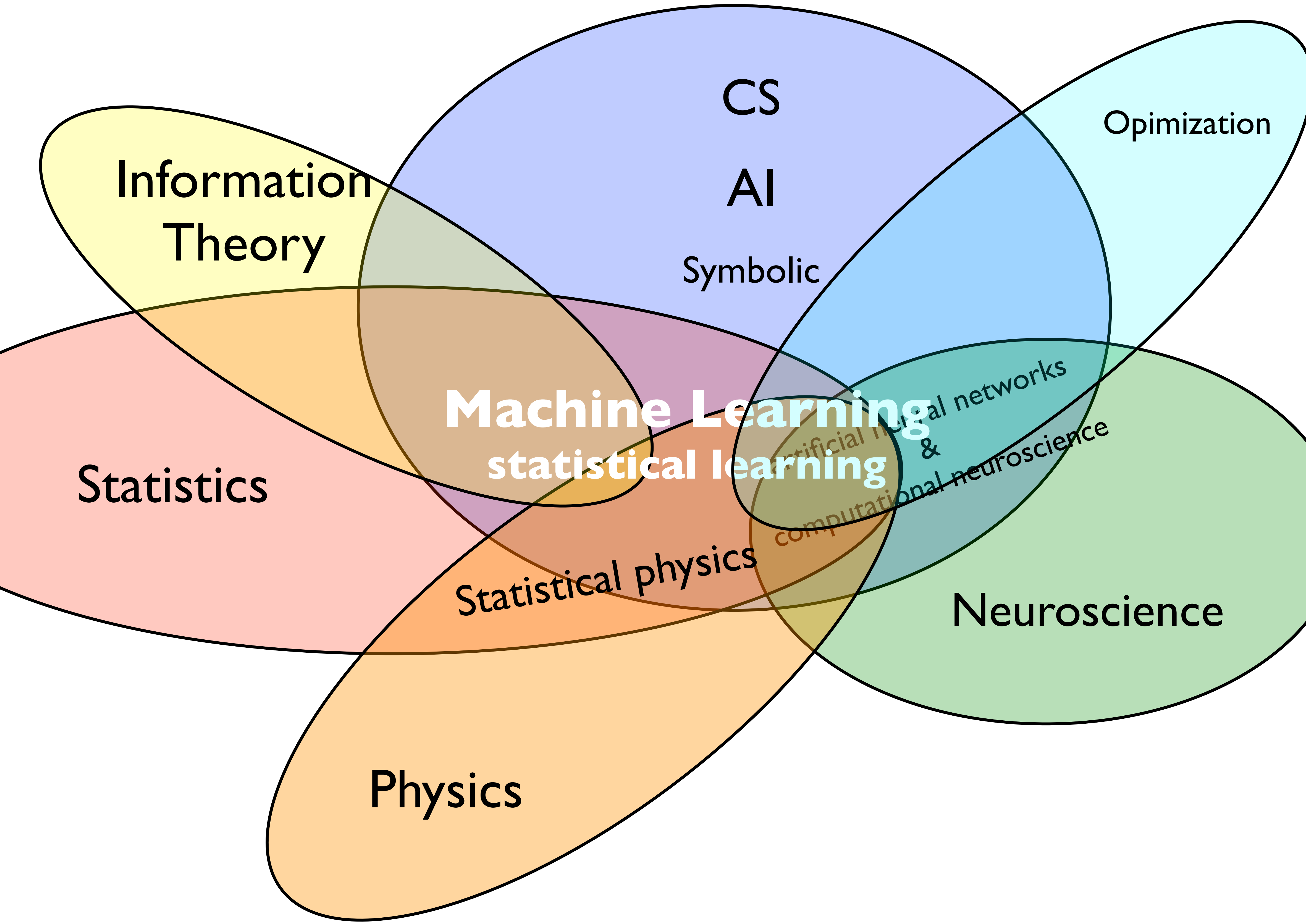
- "Connectionist" AI has matured, is mathematical, has given rise to machine learning — neural networks are part of ML.
- "Classic AI", having integrated uncertainty, gave rise to probabilistic graphical models (Bayesian networks), whose parameters can be learned.

apprentissage

machine

- The fundamental role of learning and a probabilistic approach is largely recognized.

# Current vision of founding disciplines



# What is ML?

Perspective of a (hypnotized) user

- A field of scientific study (=witchcraft) which
- researches the fundamental principles (magic formulas)
- and **develops the algorithmes** (magical incantations/spells)
- capable of using the collected data to (automagically) **produce** *predictive* functions to apply on similar data (in the future!)

# The basic ingredient of ML is...


- Collected from nature, from the internet, industrial processes.
- Arrive in many formats, structured / unstructured, rarely clean, often messy.
- In learning we want to see our data as a **list of examples** (or we will  transform them in that form)
- ideally ***many*** *examples* of the *same nature*.

➡ preferably with each example, *a vector of numbers* (or we will transform them in that form)

# Learn from **examples!**

“horse”

“horse”

“horse”

Principle much more general than to write by hand, starting from scratch, an algorithm to recognize a horse …

You know how to program: how would you do it?

# “Classic” algorithms vs learning

- Classic approach:
  - formal description of the constraints of entry and desired output
  - understanding of the computational problem
  - **design** of an algorithmic solution, based on this understanding
  - **increased** noise and ambiguity
- Problems:
  - Incomplete understanding
  - Algorithmic solution can be very expensive

# “Classic” algorithms vs learning

- Learning approach:
  - data (examples) of the form **(input, output)**
  - **partial** understanding of the computational problem: a priori knowledge
  - **learning:** searching within a large class of functions

- Important:

For ML to work, we need (sometimes a lot of) data.
The more data we have, the better results we get.

# Learning

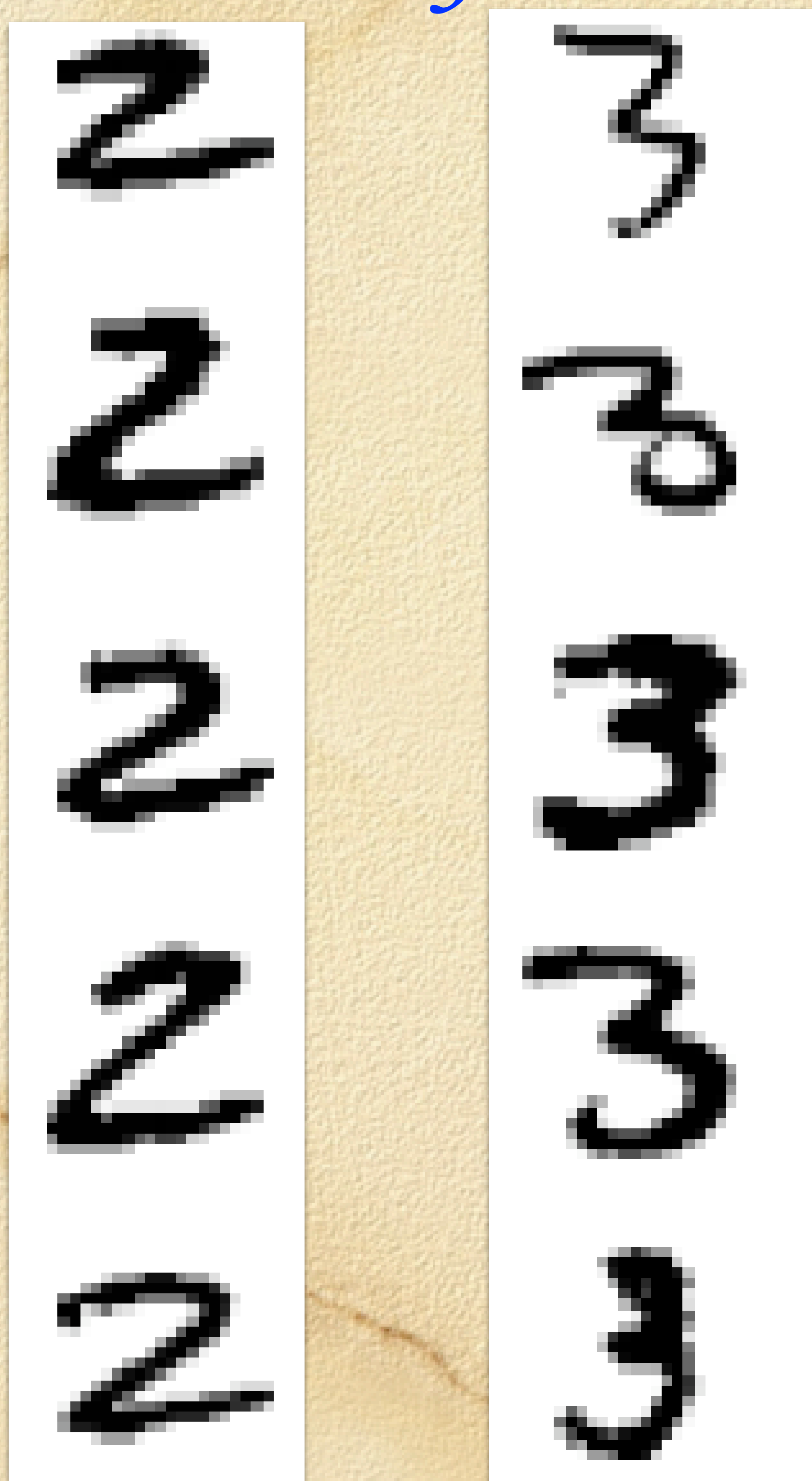
- An essential characteristic of natural intelligence
- Learning **by heart** vs **inductive** reasoning
- Key word: **generalization**
- Typical learning situation:

1. We are given examples (data)
2. We are then presented with a **new example** and we have to make a decision/prediction.

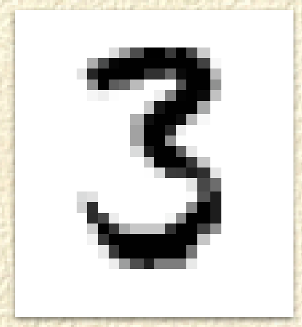
# Ex: Character recognition

Training set

2 3



Test point



2 or 3 ?

*Learning is not simply* *memorizing…*

*It is to be able to* *generalize!*

on new examples that we have not before seen

# Categories of learning problems

- Classification
    - Task: **Classify** new examples in discrete categories
- Regression
    - Task: make a **real-valued prediction** for each example
- Density estimation
    - Task: decide if new example resembles seen examples

# Machine learning? or Statistics?

a lot in common but with a difference in point of view

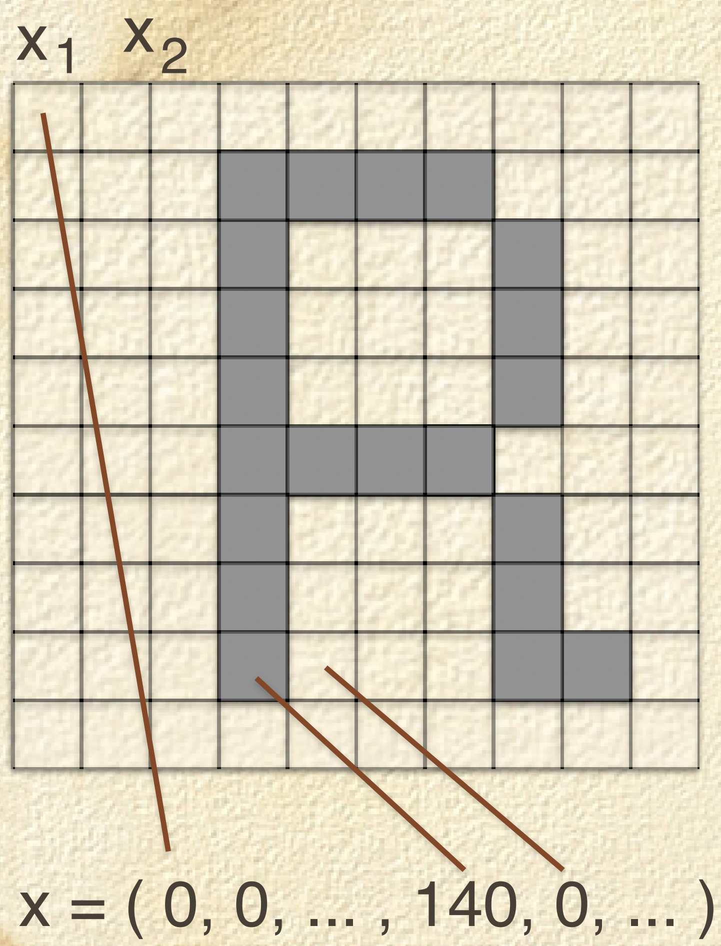
- Statistics: branch of mathematics
  - Importance of rigor and theoretical guarantees
  - Strong assumptions and hypothesis tests
- Machine learning:
  branch of Artificial Intelligence (computer science)
  - Grand ambition: intelligence!
  - We take inspiration from everything we can
    (neuroscience, statistics, physics, information theory, ...)
  - The important fact is that ML works! ☺️
    Pragmatic approach.

# Data-mining?

## or machine learning? or statistics?

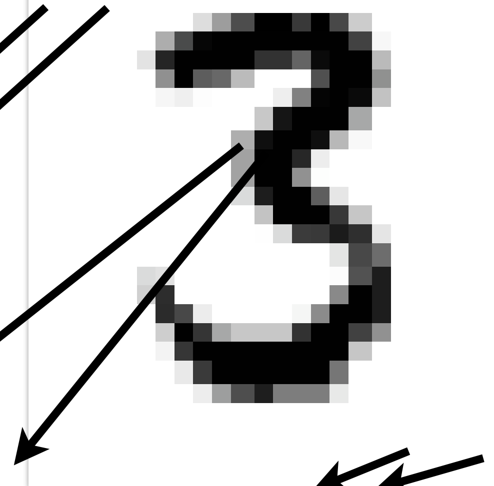
- Statistics and machine-learning = theoretical studies and algorithmic developments for data analysis / learning.
- data-mining = use of these techniques on big industrial problems (big datasets).
    - Challenges related to size: scaling problems
    - Practical approach.

# Vector representation of a data example




x vector in $R^d$

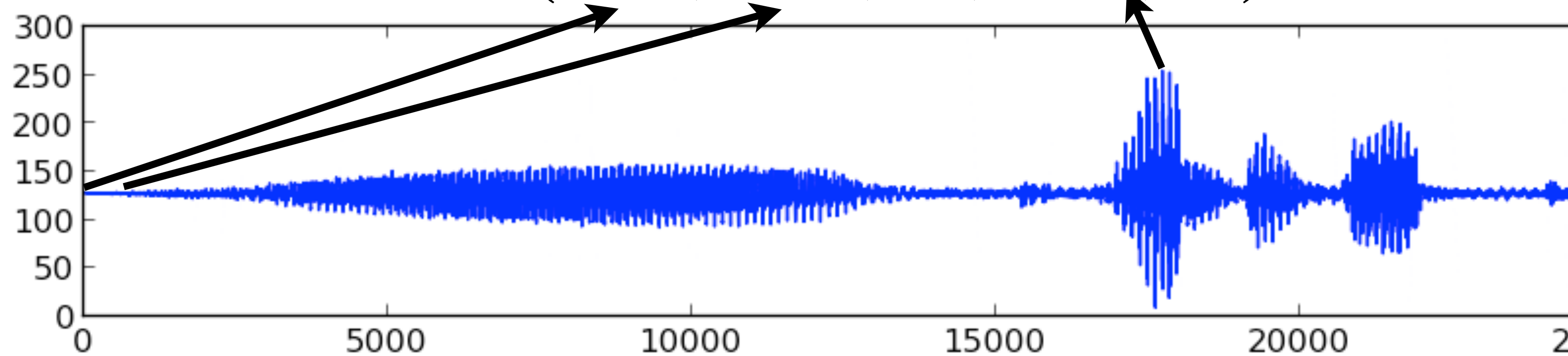
# Transform an example into a vector representation $\mathbf{x} \in \mathbb{R}^d$



## Gross representation:

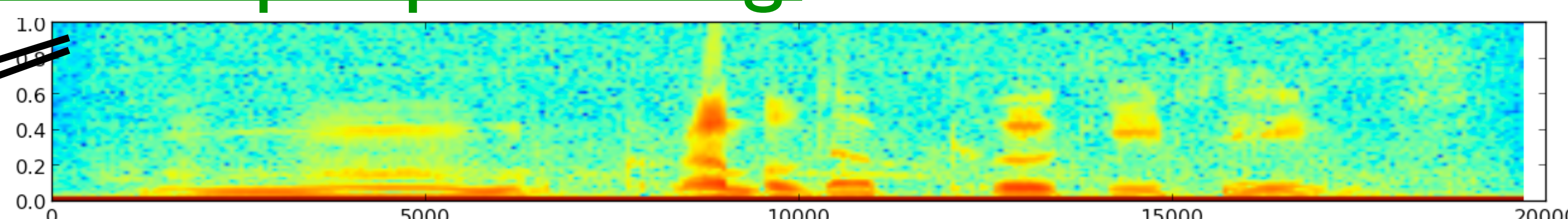
$x = (0, 0, \ldots, 54, 120, \ldots, 0, 0)$

$x = (125, 125, \ldots, 250, \ldots)$




## Or feature extraction from pre-processing:

$x = (\ ,\ \ ,\ \ ,\ \ , \ldots.)$




we the jumped jumping run elephant dog cat horse

Bag of words for «The cat jumped»: $x = (\ldots 0 \ldots, 0, 1, \ldots 0 \ldots, 1, 0, 0, \ldots., 0, 0, 1, 0, \ldots 0 \ldots$

## OR vector of handmade features:

$x = (\text{feature } 1, \ldots, \text{feature } d)$

ex: Histograms of Oriented Gradients

# Data set seen as a scatter plot in a high-dimensional vector space

(input)

$\mathbf{x} \in \mathbb{R}^d$

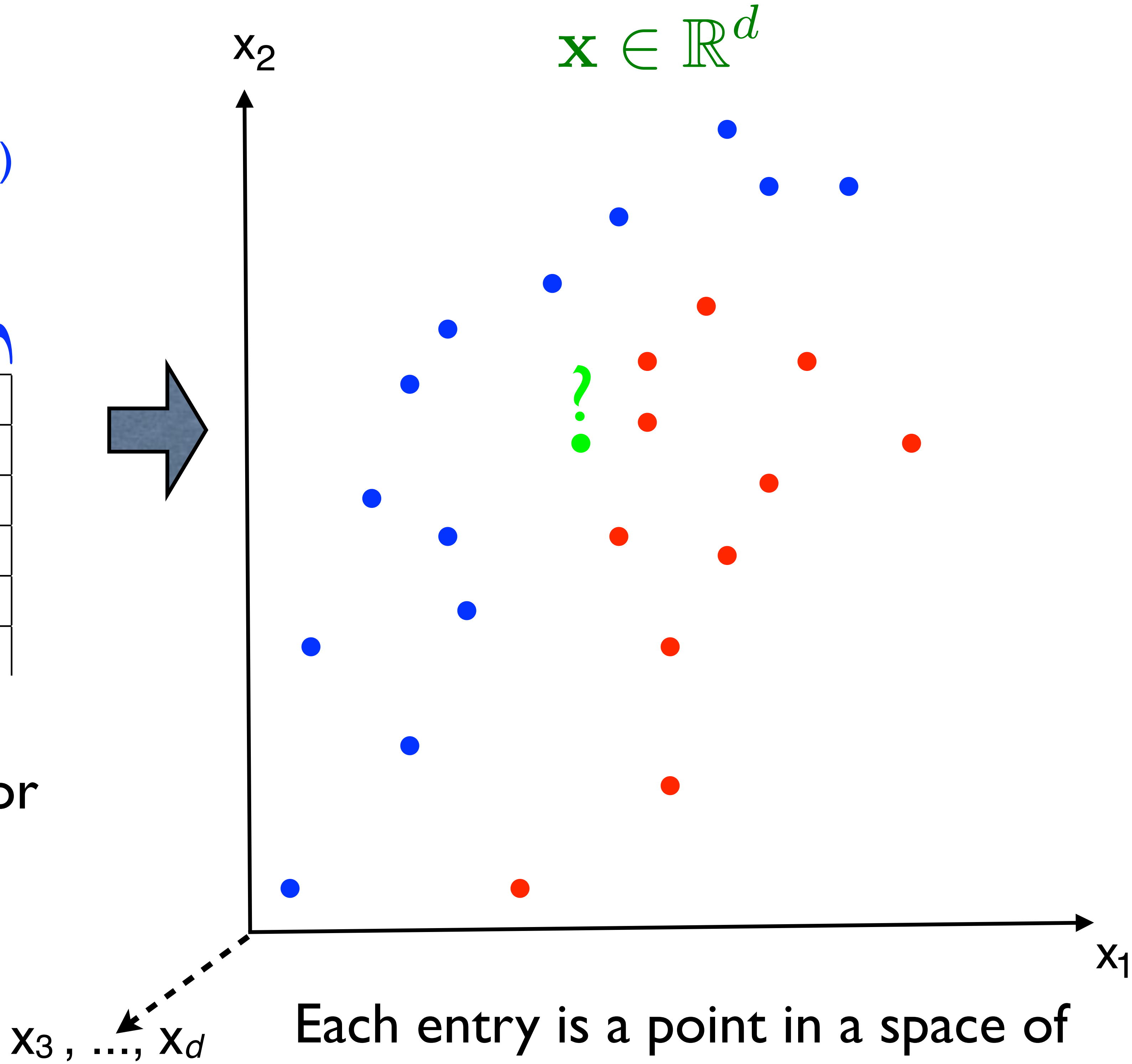
(label)

$y$

n examples

| $\mathbf{x}_1$ | $\mathbf{x}_2$ | $\mathbf{x}_3$ | $\mathbf{x}_4$ | $\mathbf{x}_5$ | |
|---|---|---|---|---|---|
| 0.32 | -0.27 | +1 | 0 | 0.82 | 1 |
| -0.12 | 0.42 | -1 | 1 | 0.22 | 0 |
| 0.06 | 0.35 | -1 | 1 | -0.37 | 1 |
| 0.91 | -0.72 | +1 | 0 | -0.63 | 1 |
| ... | ... | ... | ... | ... | ... |

Each example (row) is a vector of $d$+1-dimensions



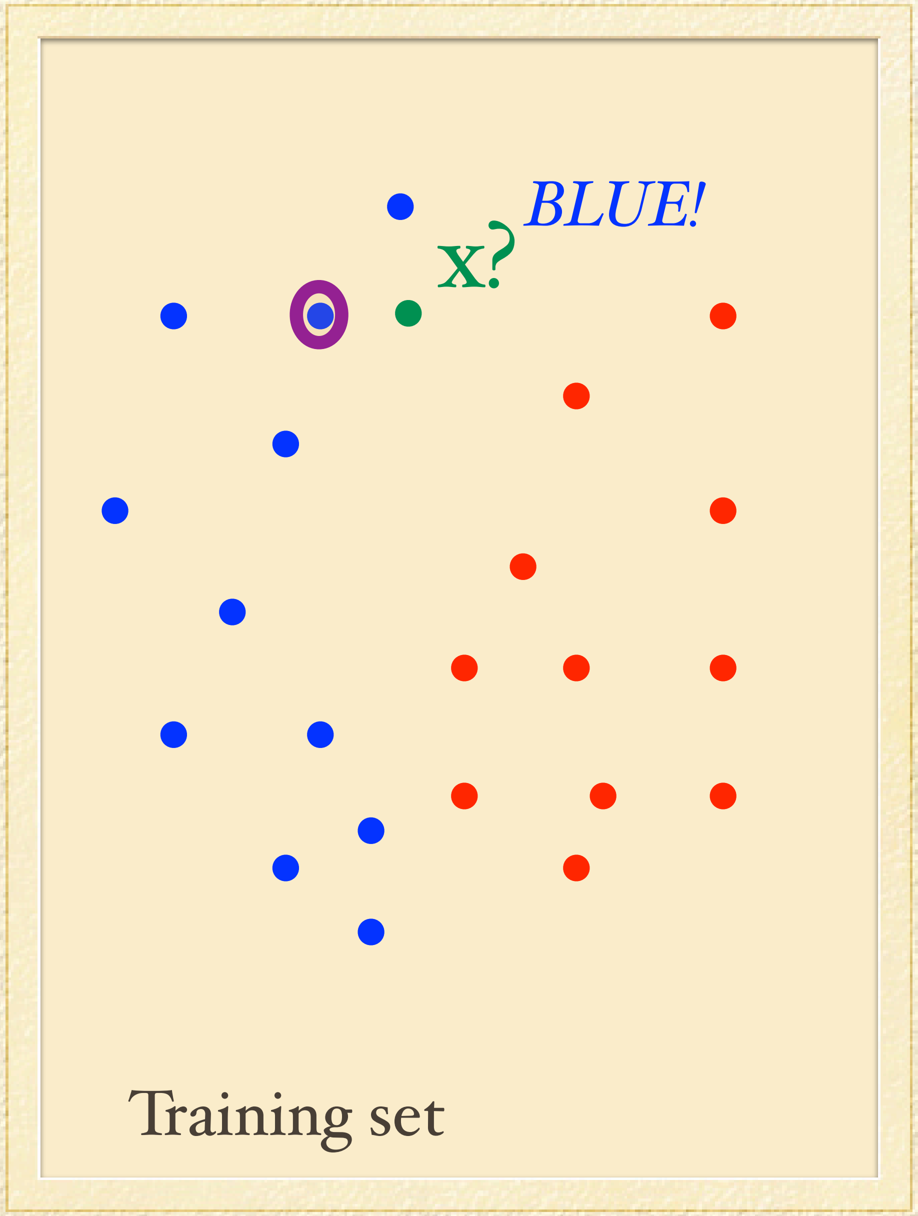

Each entry is a point in a space of $d$-dimensions

# Ex: the nearest neighbor algorithm

For a test point x:

- We find the **nearest neighbour** of x within the training set apprentissage by some measure of distance (eg Euclidean distance).

- We associate x with the class of this nearest neighbor.



# Représentation des données:

## Notion de niveau de représentation