# An introduction to Machine Learning

## Acquiring key notions

E. Rachelson

# Discussion

Your expectations for this course?

Let’s hope they match what I have in store.

I’ll try to adapt to fit your specific needs.

# Contents

| | |
|---|---|
| What this class is about: | key concepts in Machine Learning (ML) |
| What it is not about: | data processing, AI, big data, etc. |

What this class is useful for:

Evaluate statements/projects/proposals about ML or using ML

Lay the ground for deeper exploration of technical ML topics

Steps towards this goal:

- Defining vocabulary, debunking misconceptions, naming tasks and problems
- Situating ML within data analysis
- Diving into the technicalities of 4 different perspectives on ML
  - A geometrical perspective (support vector machines and more)
  - A probabilistic perspective (Bayesian modeling methods)
  - A connexionist perspective (neural networks and deep learning)
  - An ensemblist perspective (incl. random forests)

# Schedule (flexible)

**day 1**

| | |
|---|---|
| 09:00 - 10:00 | Introduction |
| | Context and definitions |
| 10:00 - 10:30 | The importance of pre-processing |
| | A practical example on text data |
| 10:45 - 12:15 | A geometrical approach to Machine Learning |
| | An intro to support vector machines and a bit of kernel theory |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 17:30 | A probabilistic approach to Machine Learning |
| | Bayesian modeling, naive Bayes blassifiers and Gaussian Processes |

**day 2**

| | |
|---|---|
| 09:00 - 12:15 | A connexionist perspective on Machine Learning |
| | An introduction to Artificial Neural Networks and to Deep Learning |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 16:30 | Ensemble of explainable models |
| | Decision trees, bagging and random forests |
| 16:30 - 17:30 | Wrap-up and discussion |

# A word on the instructor

- https://erachelson.github.io/
- Professor in ML/AI at ISAE-SUPAERO, previously at various places.
- Engineer, PhD, HDR
- Reinforcement Learning, SuReLI team leader
- Founder of Data Science curricula at ISAE-SUPAERO
  MVA Lecturer (ENS Paris-Saclay)

# ML and you

Your ML keywords

Problems and examples you want to use ML for.
Keywords associated to ML.
Let’s list all that on the white board.

# Buzz-words and definitions

**AI** ML is only a small (currently fashionable) part of Artificial Intelligence.

**BD** Big Data refers to working with datasets that have large Volume, Variety, Velocity (, Veracity, and Value).

**ML** Field of computer science that gives computer systems the ability to “learn” (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed.

**DL** Deep Learning is Machine Learning with Deep Neural Networks.

**GenAI** ML task of generating convincing content.

**threat** [Provocative thought] ML / Data Science / Big Data are as much of a threat (to jobs, the society, the economy. . . ) as the combustion engine was in the XIXth century.

**ethics** Technical problems are not just technical problems and solutions *always* imply some tradeoff. Who bears the (moral) responsibility?

October 2012 Issue

DATA

## Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren’t seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, “It was like arriving at a conference reception and realizing you don’t know anyone. So you just stand in the corner sipping your drink—and you probably leave early.”

# Machine Learning

Machines that learn?
Let's try to give a general definition.

Machine learning is a field of computer science that gives computer systems the ability to "learn" (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed.

(Wikipedia)

Task? Performance? Data?
We'll clarify this through examples.

# Applicative examples

- Predictive maintenance
- Market segmentation
- Demand forecast
- Preliminary design studies
- Clinical diagnosis
- Documentation management
- Satellite imaging
- ...

# From field tasks, to data, to ML

Let’s take the example of Predictive Maintenance.

We would like to build automated tools for the following tasks:

- Visualize system state
- Identify anomalies
- Predict Remaining Useful Life (RUL) / Time To Failure (TTF)
- Predict failure occurrence or probability at a given horizon

All this, in order to base our maintenance strategy on the (inferred) system state, rather than a general statistical trend.

Can you relate this task decomposition to the other use-cases?

Traditionally, all this is based on user expertise.
Let’s take a data-driven approach.

# Data analysis workflow

1. Collect
2. Analyze
3. Predict
4. Decide

- Sensors deployment
- Historical data collection
- Integrated storage (datawarehouses) and retrieval issues

→ Extract-Transform-Load (ETL) process

More on ETL: [link].

The *data engineer*'s job: data quality, management, availability.

# Data analysis workflow

1. Collect
2. Analyze
3. Predict
4. Decide

- data cleaning
- feature selection / engineering
- performance criteria
- algorithm selection
- parameters tuning

The *data analyst* or *data scientist*'s job.
But can't be disconnected from field engineers on the task.

# Data analysis workflow

1. Collect
2. Analyze
3. Predict
4. Decide

- Make predictions on new test cases
- Deploy solution in your operational process
- Make things usable

# Data analysis workflow

1. Collect
2. Analyze
3. Predict
4. Decide

- Improve your decisions

End-user.
Job title depends on your professional field.

# Data analysis workflow

1. Collect
2. Analyze
3. Predict
4. Decide

Need to automate as many steps as possible in this workflow

→ data-driven approaches

→ Machine Learning for step 2 (and 3)

# A word on data quality

- amount of data: data is often abundant but crucial data is often scarce
- noise, errors, missing data, outdated data: reliability
- high-dimensional data
- class imbalance
- heterogeneous data (scalars, booleans, time series, images, text, . . . )

All these will influence your algorithmic design or choices.

So let’s talk about algorithms to see how we can solve the problems listed earlier.

# Machine Learning

Recall the general definition of ML:

Machine learning is a field of computer science that gives computer systems the ability to “learn” (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed.

(Wikipedia)

We have clarified *data*. What about *task* and *performance*?

# ML examples

Given the available *data*. . .

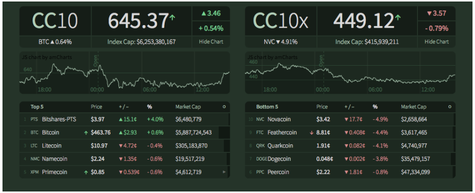
- Will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?
- Is this e-mail a spam?
- Can I cluster together customers? press articles? genes?
- What is the best strategy when playing Counter Strike? or poker?

Image sources: Wikimedia commons

# ML examples

Given the available *data*. . .

- Will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?
- Is this e-mail a spam?
- Can I cluster together customers? press articles? genes?
- What is the best strategy when playing Counter Strike? or poker?



CC10 645.37↑ ▲3.46 +0.54%

BTC ▲0.64% Index Cap: $6,253,380,167 Hide Chart

| | | Top 5 | | Price | +/– | % | Market Cap |
|---|---|---|---|---|---|---|---|
| 1 | PTS | Bitshares-PTS | | $3.97 | ▲15.1¢ | +4.0% | $6,480,779 |
| 2 | BTC | Bitcoin | ↑ | $463.76 | ▲$2.93 | +0.6% | $5,887,724,543 |
| 3 | LTC | Litecoin | | $10.97 | ▼4.72¢ | -0.4% | $305,183,870 |
| 4 | NMC | Namecoin | | $2.24 | ▼1.35¢ | -0.6% | $19,517,219 |
| 5 | XPM | Primecoin | ↑ | $0.85 | ▼0.539¢ | -0.6% | $4,612,719 |

CC10x 449.12↑ ▼3.57 -0.79%

NVC ▼4.91% Index Cap: $415,939,211 Hide Chart

| | | Bottom 5 | | Price | +/– | % | Market Cap |
|---|---|---|---|---|---|---|---|
| 10 | NVC | Novacoin | | $3.42 | ▼17.7¢ | -4.9% | $2,658,664 |
| 9 | FTC | Feathercoin | ↓ | 8.81¢ | ▼0.408¢ | -4.4% | $3,617,465 |
| 8 | QRK | Quarkcoin | | 1.91¢ | ▼0.082¢ | -4.1% | $4,740,977 |
| 7 | DOGE | Dogecoin | | 0.048¢ | ▼0.002¢ | -3.8% | $35,479,157 |
| 6 | PPC | Peercoin | | $2.22 | ▼1.81¢ | -0.8% | $47,334,099 |

Image sources: Wikimedia commons

# ML examples

Given the available *data*. . .

- Will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?
- Is this e-mail a spam?
- Can I cluster together customers? press articles? genes?
- What is the best strategy when playing Counter Strike? or poker?

Image sources: [link]

# ML examples

Given the available *data*. . .

- Will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?
- Is this e-mail a spam?
- Can I cluster together customers? press articles? genes?
- What is the best strategy when playing Counter Strike? or poker?

✉ **Enlarge your thesis!**

Image sources: Iconfinder

# ML examples

Given the available *data*...

- Will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?
- Is this e-mail a spam?
- Can I cluster together customers? press articles? genes?
- What is the best strategy when playing Counter Strike? or poker?

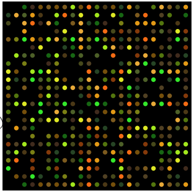


Image sources: People.jpg / Writing to Discuss: Use of a Clustering Technique / DNA microarray

# ML examples

Given the available *data*. . .

- Will this patient have a second heart attack in the next 5 years?
- What price for this stock, 6 months from now?
- Is this handwritten number a 7?
- Is this e-mail a spam?
- Can I cluster together customers? press articles? genes?
- What is the best strategy when playing Counter Strike? or poker?

COUNTER STRIKE SOURCE
THE MOST POPULAR ONLINE ACTION GAME OF ALL TIME!
ALSO INCLUDES: HALF-LIFE 2: DEATHMATCH DAY OF DEFEAT: SOURCE
VALVE

Image sources: CS:source / poker

# ML tasks

What does ML do? 3 main tasks.

| **Task** | Supervised Learning | Unsupervised Learning | Reinforcement Learning |
|---|---|---|---|
| **Goal** | Learn a function, $f(x) = y$ | Find groups and correlations, $x \in C$ | Optimal control, $f(x) = u \ / \ \max \sum r$ |
| **Data** | $\{(x,y)\}$ | $\{x\}$ | $\{(x,u,r,x')\}$ |
| **Sub-task** | Classification, Regression | Clustering, Density estimation, Dimensionnality reduction | Value estimation, Policy optimization |
| **Algo ex.** | Neural Networks, SVM, Random Forests | k-means, PCA, HCA | Q-learning |

# Evaluation criteria

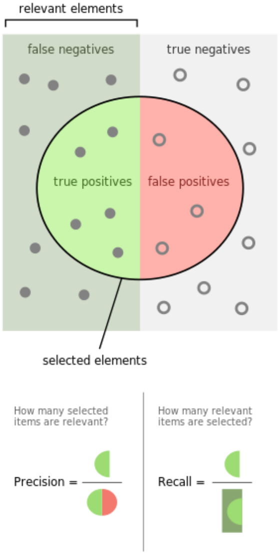
Evaluating ML methods? What do we really want?

Ability to fit the training data:

- Regression: Mean Square Error
- Classification: Accuracy, TP, FP, ROC, AUC...
  cf. this Wikipedia article
- Clustering: similarity scores

Ability to generalize:

- Goal: filter out noise, avoid overfitting, generalize to unseen cases.
- ML Notions:
  - maximize margin
  - minimize difference btw class distributions (cross-entropy)



# Empirical risk minimization

A generic framework: empirical risk minimization.

| | |
|---|---|
| Task, data distribution | $x, y \sim p(x, y)$ |
| Cost of error, *loss* function | $\ell(f(x), y)$ |
| Performance for the task, *Risk* | $R(f) = \mathbb{E}_{x,y\sim p}[\ell(f(x), y)]$ |

Ideal solution to ML problem:
$\min_f R(f)$

But only finite amount of data!
*Empirical risk:* $\bar{R}(f) \frac{1}{N} \sum_i \ell(f(x_i), y_i)$
*Empirical risk minimization*: $f^* = \arg\min_f \bar{R}(f)$

Generalization gap: $|R(f^*) - \bar{R}(f)|$

# Short break

Now that we have cleared up the fog, let’s take a look at these buzz-words again.

# Buzz-words and definitions

**AI** ML is only a small (currently fashionable) part of Artificial Intelligence.

**BD** Big Data refers to working with datasets that have large Volume, Variety, Velocity (, Veracity, and Value).

**ML** Field of computer science that gives computer systems the ability to “learn” (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed.

**DL** Deep Learning is Machine Learning with Deep Neural Networks.

**GenAI** ML task of generating convincing content.

**threat** [Provocative thought] ML / Data Science / Big Data are as much of a threat (to jobs, the society, the economy. . . ) as the combustion engine was in the XIXth century.

**ethics** Technical problems are not just technical problems and solutions *always* imply some tradeoff. Who bears the (moral) responsibility?

October 2012 Issue

DATA

Data Scientist: The Sexiest Job of the 21st Century

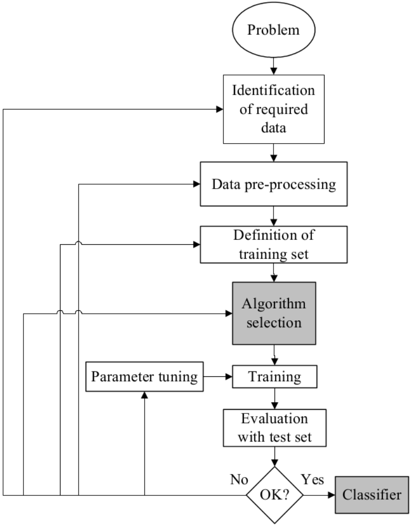
by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren’t seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, “It was like arriving at a conference reception and realizing you don’t know anyone. So you just stand in the corner sipping your drink—and you probably leave early.”

# The analysis pipeline

Recall the data analysis workflow?
ML algorithms are only a small (crucial) part of the analysis and decision pipeline!



From **Supervised Machine Learning: A Review of Classification Techniques**, S. B. Kotsiantis, *Informatica*, 31:249–268, 2007.

# Relating field tasks and ML tasks

Back to the example of Predictive Maintenance tasks.

- Visualizing system state

  → Dimensionnality reduction (Unsupervised learning)
- Detecting anomalies

  → Density estimation (Unsupervised learning)
- Predicting RUL or TTF

  → Regression (Supervised learning)
- Predicting failure in $N$ cycles

  → Classification (Supervised learning)

# Relating field tasks and ML tasks

Back to the example of Predictive Maintenance tasks.

- Visualizing system state
  → Dimensionnality reduction (Unsupervised learning)
- Detecting anomalies
  → Density estimation (Unsupervised learning)
- Predicting RUL or TTF
  → Regression (Supervised learning)
- Predicting failure in *N* cycles
  → Classification (Supervised learning)

Thinking like a Maintenance Engineer:
How can I monitor my system to manage my maintenance operations?
Thinking like a Data Scientist:
Is this a supervised or an unsupervised problem? What available data?

Relate this example to your own field.
Now you can start discussing with data scientists to design together the most appropriate method for your data and your problem.

# A word on ML software and the ML ecosystem

Software:

- Many free software libraries: scikit-learn, tensorflow, pytorch...
  check `www.mloss.org`!
- Free environments: Weka, RStudio...
- Commercial embedded solutions (more or less specialized): Matlab, IBM, Microsoft...

Short “time to market”, high innovation pace, open knowledge practices.
Value is in 1) the data, 2) the advanced expertise, 3) the implementation tricks.
But *not* in the IP.

# A word on libraries

Scikit-learn = general purpose Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license
- Well documented, with lots of examples

`http://scikit-learn.org`

Let's take a look at the documentation's table of contents to grasp a few more keywords.

We will also look at PyTorch (deep neural networks) later in the course.
`https://pytorch.org/`

# Contents

What this class is about: key concepts in Machine Learning (ML)

What it is not about: AI in general,
data storage and manipulation, big or small data,
data storytelling,
specific applications of ML, etc.

I'm happy to discuss these along the class and during the breaks, but they are beyond our focus.

What this class is useful for:

Evaluate statements/projects/proposals about ML or using ML
Lay the ground for deeper exploration of technical ML topics

Steps towards this goal:

- Defining vocabulary, debunking misconceptions, naming tasks and problems
- Situating ML within data analysis
- Diving into the technicalities of 4 different perspectives on ML
  - A geometrical perspective (support vector machines and more)
  - A probabilistic perspective (Bayesian modeling methods)
  - A connexionist perspective (neural networks and deep learning)
  - An ensemblist perspective (incl. random forests)

# What you should expect in the remainder of this class

- As many intuitive notions as possible,
- . . . but also quite a bit of (hopefully painless) math,
- . . . and a fair amount of hands-on manipulations and demos.

# Schedule (flexible)

**day 1**

| | |
|---|---|
| 09:00 - 10:00 | Introduction<br>Context and definitions |
| 10:00 - 10:30 | The importance of pre-processing<br>A practical example on text data |
| 10:45 - 12:15 | A geometrical approach to Machine Learning<br>An intro to support vector machines and a bit of kernel theory |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 17:30 | A probabilistic approach to Machine Learning<br>Bayesian modeling, naive Bayes blassifiers and Gaussian Processes |

**day 2**

| | |
|---|---|
| 09:00 - 12:15 | A connexionist perspective on Machine Learning<br>An introduction to Artificial Neural Networks and to Deep Learning |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 16:30 | Ensemble of explainable models<br>Decision trees, bagging and random forests |
| 16:30 - 17:30 | Wrap-up and discussion |

# The importance of data pre-processing

Images, text, video, sound, measurement time series, continuous or discrete variables, missing data...

→ filtering out noise and irrelevant data.
*scaling, filtering, reducing...*

→ data- and application-specific procedures.
*domain knowledge leverages non-representative datasets.*

→ source of potential harm.
*keep goals in mind, to make informed tradeoffs that might induce bias.*

⇒ Crucial elements for a good start.

Never neglect the pre-processing.

# Schedule (flexible)

| | |
|---|---|
| **day 1** | |
| 09:00 - 10:00 | Introduction<br>Context and definitions |
| 10:00 - 10:30 | The importance of pre-processing<br>A practical example on text data |
| 10:45 - 12:15 | A geometrical approach to Machine Learning<br>An intro to support vector machines and a bit of kernel theory |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 17:30 | A probabilistic approach to Machine Learning<br>Bayesian modeling, naive Bayes blassifiers and Gaussian Processes |
| **day 2** | |
| 09:00 - 12:15 | A connexionist perspective on Machine Learning<br>An introduction to Artificial Neural Networks and to Deep Learning |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 16:30 | Ensemble of explainable models<br>Decision trees, bagging and random forests |
| 16:30 - 17:30 | Wrap-up and discussion |

# A geometrical approach to ML

1. Draw a line that sits as far as possible from the data points → Support Vector Machines
2. Send all data points in a higher dimension space where they are linearly separable → kernel trick

⇒ SVM + kernel trick = Find the optimal separating hyperplane in this higher dimension space, without ever computing the mapping.

- SVM try to separate data by maximizing a geometrical margin
- They are computed offline
- They offer a sparse, robust to class imbalance, and easy to evaluate predictor
- Kernels are a way of enriching (lifting) the data representation so that it becomes linearly separable
- SVMs + kernels offer a versatile method for classification, regression and density estimation
- Link to documentation in scikit-learn

# Schedule (flexible)

| | |
|---|---|
| **day 1** | |
| 09:00 - 10:00 | Introduction |
| | Context and definitions |
| 10:00 - 10:30 | The importance of pre-processing |
| | A practical example on text data |
| 10:45 - 12:15 | A geometrical approach to Machine Learning |
| | An intro to support vector machines and a bit of kernel theory |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 17:30 | A probabilistic approach to Machine Learning |
| | Bayesian modeling, naive Bayes blassifiers and Gaussian Processes |
| **day 2** | |
| 09:00 - 12:15 | A connexionist perspective on Machine Learning |
| | An introduction to Artificial Neural Networks and to Deep Learning |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 16:30 | Ensemble of explainable models |
| | Decision trees, bagging and random forests |
| 16:30 - 17:30 | Wrap-up and discussion |

# A probabilistic approach to ML

Bayesian approach: find $y$ that maximizes $\mathbb{P}(Y = y | \textrm{data}, X = x)$

This problem of Bayesian inference is hard to solve without additional hypothesis.

# A probabilistic approach to ML

## Naive Bayes classifiers

- Make a naive, counter-intuitive hypothesis of conditional independence of the feature variables;
- Compute each class' probability for a new example using this hypothesis and picks the most probable one;
- Are a simple, scalable, online method;
- Despite their simplicity, perform surprisingly well and are competitive in many applications.

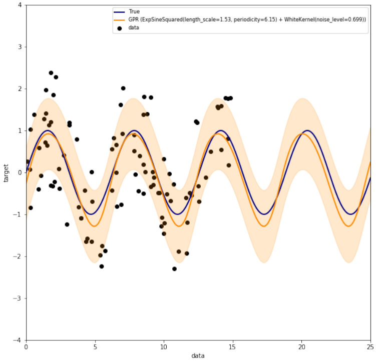
# A probabilistic approach to ML

## Gaussian Processes

- Compute the most probable function that passes through the data points, given a priori information about how related two data points are (through a covariance kernel);
- Also provide a measure of prediction uncertainty in each point;
- Are computed offline and require an $N \times N$ matrix inversion for $N$ data points in the training set (computationnally costly);
- Careful engineering of covariance kernels can help incorporate priori knowledge into Gaussian Processes;
- Are suitable both for regression and classification.

# A probabilistic approach to ML

Note that Gaussian Processes are widely used in preliminary design phases, especially as surrogate models that replace physics computations.



# Schedule (flexible)

| | |
|---|---|
| **day 1** | |
| 09:00 - 10:00 | Introduction<br>Context and definitions |
| 10:00 - 10:30 | The importance of pre-processing<br>A practical example on text data |
| 10:45 - 12:15 | A geometrical approach to Machine Learning<br>An intro to support vector machines and a bit of kernel theory |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 17:30 | A probabilistic approach to Machine Learning<br>Bayesian modeling, naive Bayes blassifiers and Gaussian Processes |
| **day 2** | |
| 09:00 - 12:15 | A connexionist perspective on Machine Learning<br>An introduction to Artificial Neural Networks and to Deep Learning |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 16:30 | Ensemble of explainable models<br>Decision trees, bagging and random forests |
| 16:30 - 17:30 | Wrap-up and discussion |

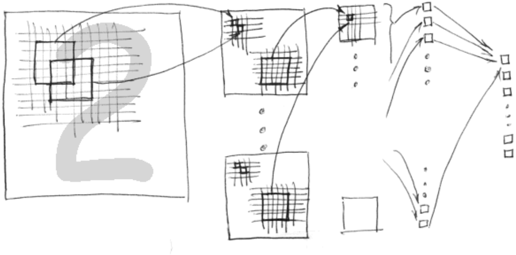
# Artificial Neural Networks

Keywords:

- Computation graph $f_\theta(x)$
- Forward pass and gradient backpropagation
- Online training
- Minibatches
- The vanishing gradient problem
- Tensorflow, Pytorch
- Avoiding overfitting: dropout, regularization, data augmentation
- Convolutional neural networks

# Artificial Neural Networks

- Versatile, online training
- State-of-the-art performance on many benchmarks
- But fragile and hard to tune
- Lots of "recipes" still today
- CNNs = method of choice for structured data (images, sound, time series. . . )
- Lots of flavors of deep learning: RNNs, generative models, language models, auto-encoders, etc.



# Schedule (flexible)

**day 1**

| | |
|---|---|
| 09:00 - 10:00 | Introduction<br>Context and definitions |
| 10:00 - 10:30 | The importance of pre-processing<br>A practical example on text data |
| 10:45 - 12:15 | A geometrical approach to Machine Learning<br>An intro to support vector machines and a bit of kernel theory |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 17:30 | A probabilistic approach to Machine Learning<br>Bayesian modeling, naive Bayes blassifiers and Gaussian Processes |

**day 2**

| | |
|---|---|
| 09:00 - 12:15 | A connexionist perspective on Machine Learning<br>An introduction to Artificial Neural Networks and to Deep Learning |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 16:30 | Ensemble of explainable models<br>Decision trees, bagging and random forests |
| 16:30 - 17:30 | Wrap-up and discussion |

# Decision trees

- Easy to interpret and to explain
- Poor representative power
- Greedy growth procedure $\Rightarrow$ suboptimal resulting tree
- Offline training
- Very sensitive to noise in the input data

# Random Forests

- RF = decision trees + random feature selection + Bagging
- Robust, scalable, out-of-the-box classifier

⇒ excellent multi-purpose benchmarking algorithm!

# Schedule (flexible)

**day 1**

| | |
|---|---|
| 09:00 - 10:00 | Introduction<br>Context and definitions |
| 10:00 - 10:30 | The importance of pre-processing<br>A practical example on text data |
| 10:45 - 12:15 | A geometrical approach to Machine Learning<br>An intro to support vector machines and a bit of kernel theory |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 17:30 | A probabilistic approach to Machine Learning<br>Bayesian modeling, naive Bayes blassifiers and Gaussian Processes |

**day 2**

| | |
|---|---|
| 09:00 - 12:15 | A connexionist perspective on Machine Learning<br>An introduction to Artificial Neural Networks and to Deep Learning |
| 12:15 - 13:30 | Lunch break |
| 13:30 - 16:30 | Ensemble of explainable models<br>Decision trees, bagging and random forests |
| 16:30 - 17:30 | Wrap-up and discussion |

# Contents

What this class is about: key concepts in Machine Learning (ML)
What it is not about: data processing, AI, big data, etc.

What this class is useful for:

Evaluate statements/projects/proposals about ML or using ML

Lay the ground for deeper exploration of technical ML topics

Steps towards this goal:

- Defining vocabulary, debunking misconceptions, naming tasks and problems
- Situating ML within data analysis
- Diving into the technicalities of 4 different perspectives on ML
  - A geometrical perspective (support vector machines and more)
  - A probabilistic perspective (Bayesian modeling methods)
  - A connexionist perspective (neural networks and deep learning)
  - An ensemblist perspective (incl. random forests)