

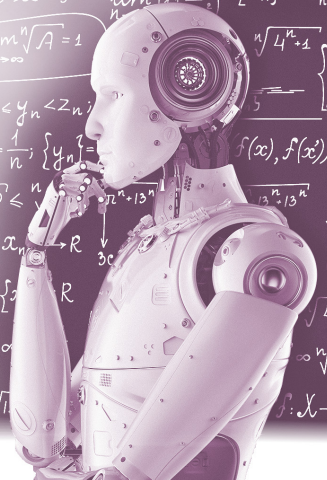


d'après une image de Mike MacKenzie

Mention **SDI-Metz**

Data and Information Sciences
Metz Mention

<https://sdi.metz.centralesupelec.fr>





Why SDI-Metz?

Expectations and opportunities

If you ???, get trained at **METZ** and become ???

are challenged by the promises of AI



if you want to study CompSci without giving up Maths

if you want to study Maths without giving up CompSci



you think coding is creating

have the courage to travel to 'Province'



you better see the challenge of tidying up the data than your room



a solid data scientist/engineer in a datablab



researcher in Machine Learning (R&D), public) after a PhD



source of innovation in AI and machine learning in the industry



consultant-analyst in services (banking insurance, etc.)



leader in a data science startup





How?

Mathematics and Computer Science for Data Science and Machine Learning

On the **maths** side

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Cours	Seq	ECTS
Machine Learning	SD9	2.5
Statistical models	SD9 SG10	4
Statistical Learning Theory	SG11	2
Reinforcement Learning	SG11	1
Sparse Models	SG11	1
Total	-	10.5

On the **computer science** side



Cours	Seq	ECTS
Advanced C++ programming	SD9	2
Deep Learning	SG10	2
Data Science Algorithms	SG10	2
Computational Models for Big Data	SD9	2
Software Application Engineering	SD9	1.5
GPU Programming	SG11	1
Total	-	10.5

On the **application** side



Cours	Seq	ECTS
End-of-studies project (CEI/PFE)		9
Natural Language Processing	SG11	2
Image Processing	SG10	1
Sound Processing	SG10	1
Seminars		0
Total	-	13





Do it for real!

- More that 50% of teachings are labworks & tutorials.
- Three projects:
 - The *Tweetoscope* project from Sept. to Dec.
 - The Deep Learning challenge from Dec. to Feb.
 - The end-of-study project (PFE) from Nov. to April.
- Acquire operational programming and technical skills:
 - C++ and Python programming languages
 - Deep learning, Stat. Programming and Big Data frameworks
 - CI/CD toolchain (git, Docker, Kubernetes, etc)
 - GPU programming (CUDA)
- Use our GPU & CPU clusters (See <https://dce.pages.centralesupelec.fr>).





How?

Double Diplomas



UNIVERSITÉ
DE LORRAINE

UL Master of Mathematics :

1 possible track

- Fundamental and Applied Mathematics (MFA)

UL Master of Computer Science :

4 possible tracks

- Machine Learning, Vision and Robotics (AVR)
- Optimization and Algorithms (OPAL)
- Software Engineering (IL)
- Information and Systems Security (SIRAV)



UNIVERSITÉ
DE LORRAINE

IAE METZ
School of
Management

IAE/UL Master: double diploma in management

3 possible tracks

- M2 Business Management
- Business Development & Entrepreneurship
- Project Management






Where?

Campus at 1h30 from Paris centre by train





Where?
The city of Metz





Where?

Research at <https://www.loria.fr/>



Visit <https://sdi.metz.centralesupelec.fr>

To know more about courses, masters, practical information, etc



Fasten you seat belt! Hard work in perspective... but it is worth it!

- Companies (especially Big Tech) are looking for people having both math and computer science skills.
- PhD. students in IA/DS who can handle technical aspects and contribute to high level science are needed.





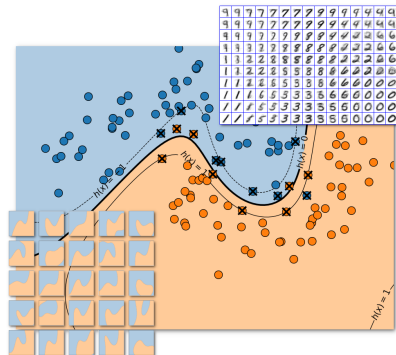
Open the black boxes of machine learning algorithms!

An overview of the whole field

- Different ways of learning
- Introduction to the statistical approach (risks, overfitting, ...)
- Main principles of data manipulation (dimensionality reduction, ...)

A comprehensive approach of algorithms

- Support vector machines and kernel methods
- Ensemble methods (boosting, bagging, décision trees)
- Vector quantization (growing neural gas, self-organizing maps)





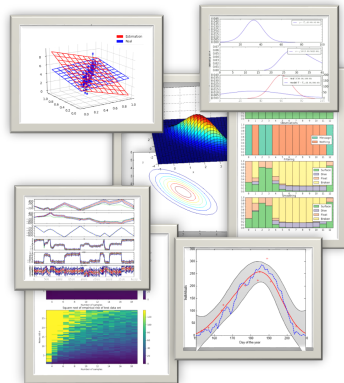
Become a data scientist solid and credible in statistics

Get a good grounding in statistics.

- Understand fundamental notions (frequentist and Bayesian estimation, belief networks, exp. models, causality, stochastic processes, etc).
- Gather these notions as theoretical ingredients to build up statistical models and efficient algorithms to learn their parameters from data.
- Discover useful models (NB, LDA/QDA, GLM, GP, HMM, KF, etc) and their application scopes.

Become a practitioner in statistical modeling.

- Play and tune models on datasets.
- Implement and test by yourself some models in Python.
- Design your own statistical models.





Efficient programming can be smart!

Strong typing in C++

- C++11/14/17/20... offers smart concepts. Typing is crucial.
- Standard Template Library design is efficient and close to theoretical algorithms.
- Type checking annoys you at compiling time... but makes your code robust!
- Get the power of templates, **it sounds like maths**.
- Think functional!

System

- Learn to invoke system features for **efficient executions**.
- Understand how memory works.
- Use threads and shared data.

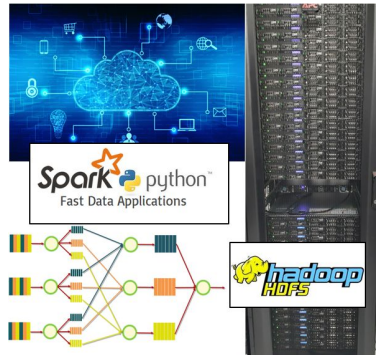




Large-scale data analysis requires the use of distributed platforms for data storage and processing, such as "Spark clusters" or "Clouds". Specific distributed programming models must then be used, and optimization of algorithms and codes is necessary to successfully "scale up".

Content

- Hadoop distributed file system (HDFS)
- Spark programming model on PC cluster
- Data analysis framework on Cloud
- "Scaling" metric

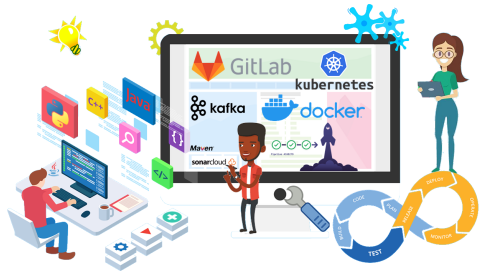




Develop your data science application like a pro!

Topics

- software application architecture, service-oriented architecture, middleware
- continuous delivery, continuous deployment, cloud, virtualization, containers
- continuous integration



Approach

- 4 lectures to present concepts
- 5 tutorials to get you started on widely used tools (Kafka, Docker, Kubernetes, GitLab)
- 6 lab sessions on a live wire example (track most active Wikipedia pages over time), deployed on a cluster





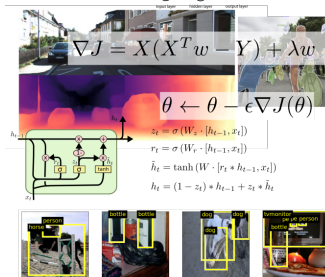
Learn and practice with deep neural networks

Understanding every single piece

- Neural network architectures (FFNN, CNN, RNN)
- Computational graphs
- Initialization and learning with gradient descents

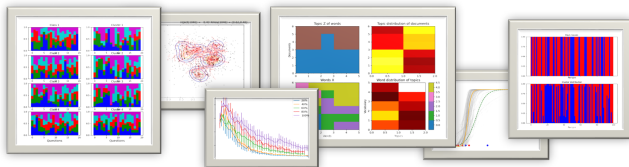
Practice on the GPUs with pytorch

- Introduction to deep learning with pytorch
- Object detection with convolutional neural networks
- Sequence processing with recurrent neural networks
- Generative neural networks





Discover how to apply statistical models to complex and large problems.



Get the fundamental ingredients

- Learn algorithmic techniques for approximate estimation based on sampling (MC, MCMC, PF, etc) and variational inference.
- Discover applications of these techniques to different problems (Latent Dirichlet Alloc., Bayesian deep learn., variat. autoencoders, etc)
- Apply these methods to solve problems using statistical programming framework Pyro/Pytorch.

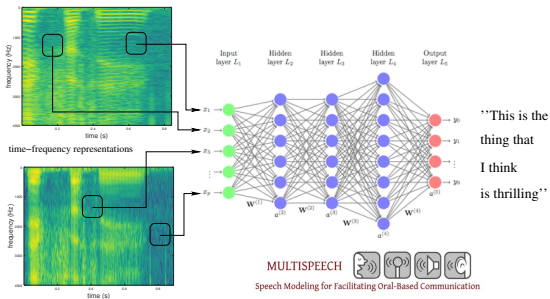
Learn fundam. algorithms in data science

- Information retrieval
- Recommender systems
- Social network analysis
- Stream processing and data sketching
- Data mining



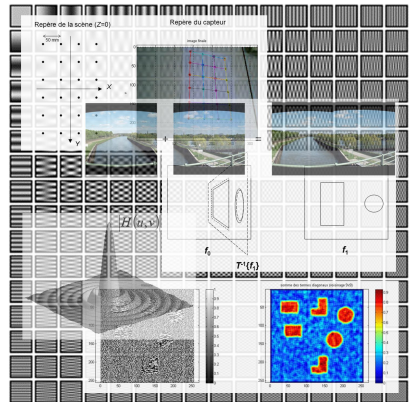
Learn and practice numerical sound processing.

- Sound perception and production
- Speech processing, including speech recognition, acquisition, manipulation, storage, transfer and synthesis
- Human machine communication



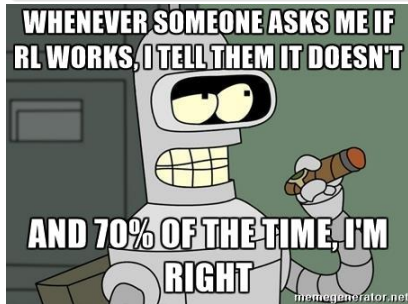


- **Visual perception, photometry:** Description of a scene. Definition of physical quantities in the field of optics.
- **Image sensor:** Physical and geometric modeling. Calibration problem. Generalization.
- **Continuous systems, digitization:** Linear modeling of optical systems. Sampling.
- **Geometric transformations:** Optimization methods for image registration. Choice of degrees of freedom.
- **Linear transformations:** Orthogonal transformations, wavelets, filter banks. Application to coding and compression.
- **Features on images** Problem of detection and classification. Statistical, geometrical features.





Using **Deep Reinforcement Learning**, computers can beat human experts at video games (Atari, Starcraft, Dota) and even at the game of Go. Such achievements were not within reach only 5 years ago. **What has changed?** What is the theory behind all this? And still, why is the following picture quite right?



This course will try to make you able to **answer these questions** and, thus, shine in salon conversation.





- Formalization of supervised learning problems
- PAC learning capacity and uniform convergence
- The bias-complexity trade-off
- The VC (Vapnik-Chervonenkis) dimension of a hypothesis space
- Two fundamental theorems of PAC learning



Alexey Chervonenkis

Statistical learning theory
Supervised learning
Generalization
PAC learning
 $\dim VC < \infty$



Vladimir Vapnik

 x_1 x_2 x_3

$$f(x_1) = f(x_2) = f(x_3) = -1$$

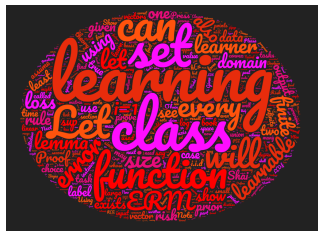
 x_4 x_5 x_6

$$f(x_4) = f(x_5) = f(x_6) = 1$$

 x_7

$$f(x_7) = ?$$

Learning f from a dataset seems difficult, what is required?



GPU computing increases the application performance and their power efficiency. Unfortunately, it requires original algorithms and programming, and some computations still remain more efficient on the CPU. Hence the need to develop "hybrid" algorithms and codes running on couple CPU + GPU.

Content

- Massively parallel computing on GPU
- Hybrid computing on CPU+GPU
- CUDA programming
- Clustering algorithms on CPU+GPU architecture





Sparse models for data science

- Data processing tasks are **complicated in high-dimension**, noise, etc.
- **Sparsity** (parsimony aka Occam's razor) is useful as a **problem-solving principle**.
- Meaningful data **feature extraction & dimension reduction**.
- **Efficient statistical models** for data analysis & visualization.

