Introduction to Machine Learning

Christophe Rappold
christophe.rappold@csic.es

IEM – CSIC

24 January 2025

What is Machine Learning ?

> A subset of artificial intelligence in the field of computer science that often uses statistical techniques to give computers the ability to "learn"with data, without being explicitly programmed
>
> Samuel Arthur –1959 –ML in Checkers

- Definition "to learn" from dictionary:

  "Gain knowledge or understanding of, or skill in by study, instruction or experience"

  - Learning a set of new facts
  - Learning *how* to do something
  - Improving ability of something already learned

What is Machine Learning ?

- Why learning ?
  - Machine learning is programming computers to optimize a performance criterion using example data or past experience
  - Learning is used when :
    - Human expertise does not exist
    - Humans are unable to explain their expertise
    - Amount of knowledge is too large for explicit encoding
    - Solution changes in time
    - Relationships can be hidden within large amounts of data
    - Solution needs to be adapted to particular cases
    - New knowledge is constantly being discovered by humans

The automatic extraction of semantic information from raw signal is at the core of many applications (object recognition, speech processing, natural language processing, planning, etc).

Can we write a computer program that does that?

- The (human) brain is so good at interpreting visual information that the gap between raw data and its semantic interpretation is difficult to assess intuitively:

This is a mushroom.

This is a mushroom.

In [1]:
```
from matplotlib.pyplot import imread
imread("mushroom-small.png")
```

Out[1]:
```
array([[[0.03921569, 0.03529412, 0.02352941, 1.        ],
        [0.2509804 , 0.1882353 , 0.20392157, 1.        ],
        [0.4117647 , 0.34117648, 0.37254903, 1.        ],
        ...,
        [0.20392157, 0.23529412, 0.17254902, 1.        ],
        [0.16470589, 0.18039216, 0.12156863, 1.        ],
        [0.18039216, 0.18039216, 0.14117648, 1.        ]],

       [[0.1254902 , 0.11372549, 0.09411765, 1.        ],
        [0.2901961 , 0.2509804 , 0.24705882, 1.        ],
        [0.21176471, 0.2       , 0.20392157, 1.        ],
        ...,
        [0.1764706 , 0.24705882, 0.12156863, 1.        ],
        [0.10980392, 0.15686275, 0.07843138, 1.        ],
        [0.16470589, 0.20784314, 0.11764706, 1.        ]],

       [[0.14117648, 0.12941177, 0.10980392, 1.        ],
        [0.21176471, 0.1882353 , 0.16862746, 1.        ],
        [0.14117648, 0.13725491, 0.12941177, 1.        ],
        ...,
        [0.10980392, 0.15686275, 0.08627451, 1.        ],
        [0.0627451 , 0.08235294, 0.05098039, 1.        ],
        [0.14117648, 0.2       , 0.09803922, 1.        ]],

       ...,
```

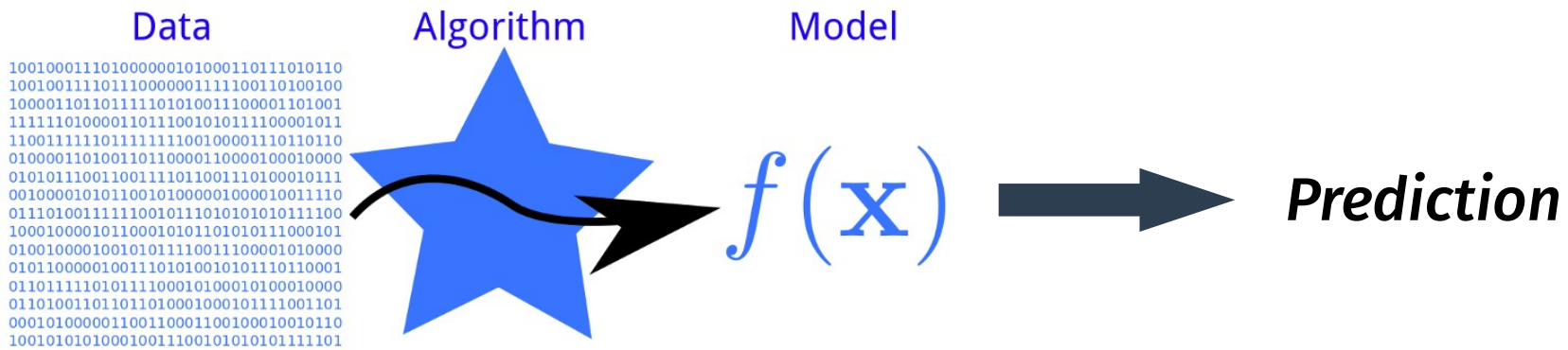
This is a mushroom.

- **Extracting semantic information requires models of high complexity.**
  - Cannot write a computer program that reproduces this process.
  - However, can write a program that learns the task of extracting semantic information.
- **A common strategy to solve this issue consists in:**
  - Defining a parametric model with high capacity
  - Optimizing its parameters by “making it work” on the training data

Learning → tuning the many parameters of the model

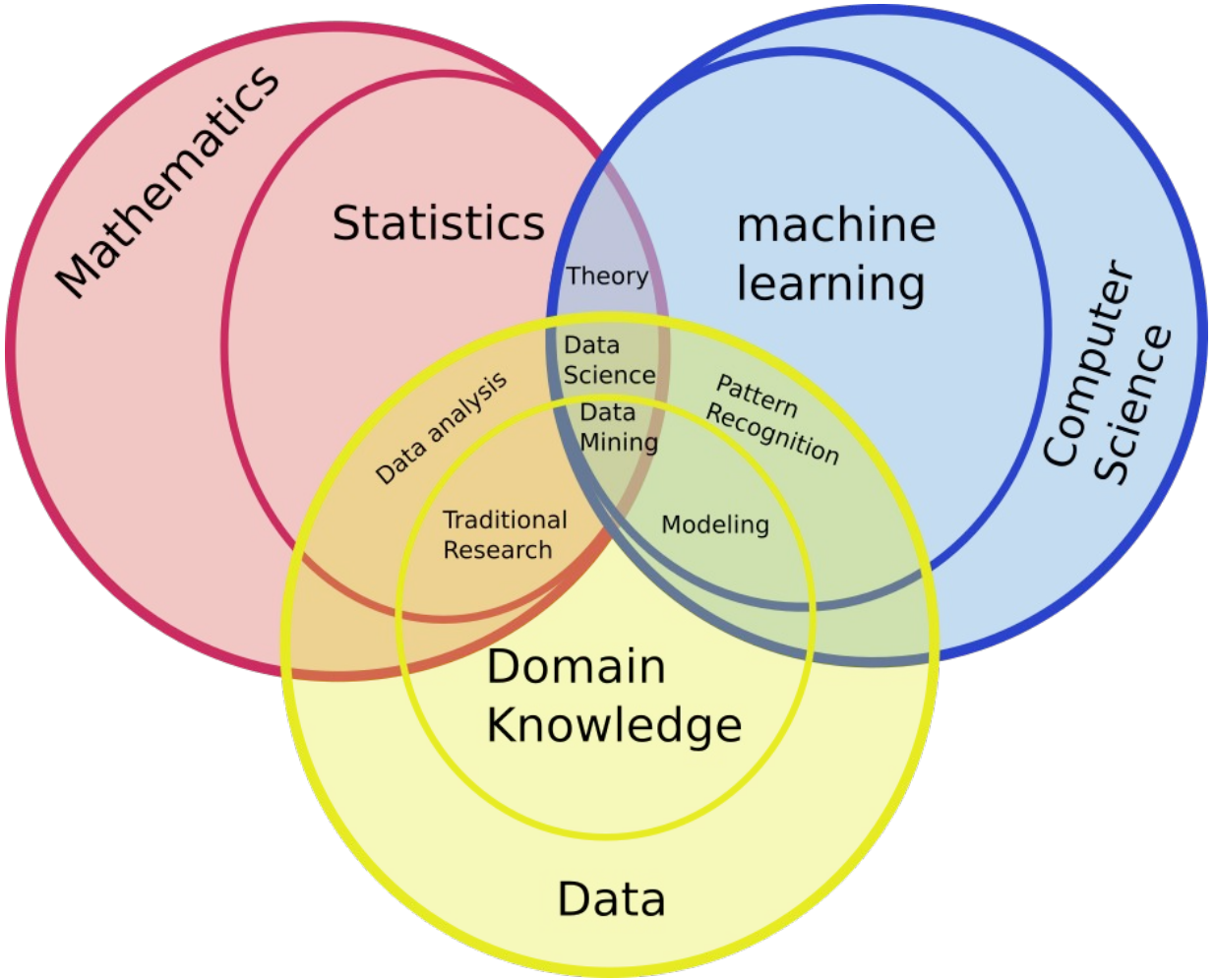
Machine learning is ...

- Finding patterns or associations that can be used to make prediction



- ML is general term → many algorithms / methods
- Big Picture Goal : Learning useful *generalizations*

Fields cross sections



Statistics vs Machine Learning

- Largely overlapping fields:
  - Both concerned with learning from data
  - Philosophical difference on ‘focus’ and ‘approach’.
- Statistics:
  - Founded in mathematics
  - Drawing valid conclusions based on analyzing existing data.
    - Making inference about a ‘population’ based on a ‘sample’
    - Tends to focus on fewer variables at once.
    - Precision and uncertainty are measures of model goodness.
- Machine Learning:
  - Founded in computer science
  - Focused on making predictions or seeking patterns (generalization).
    - Often considers a large number of variables at once.
    - Prediction accuracy to measure model goodness.

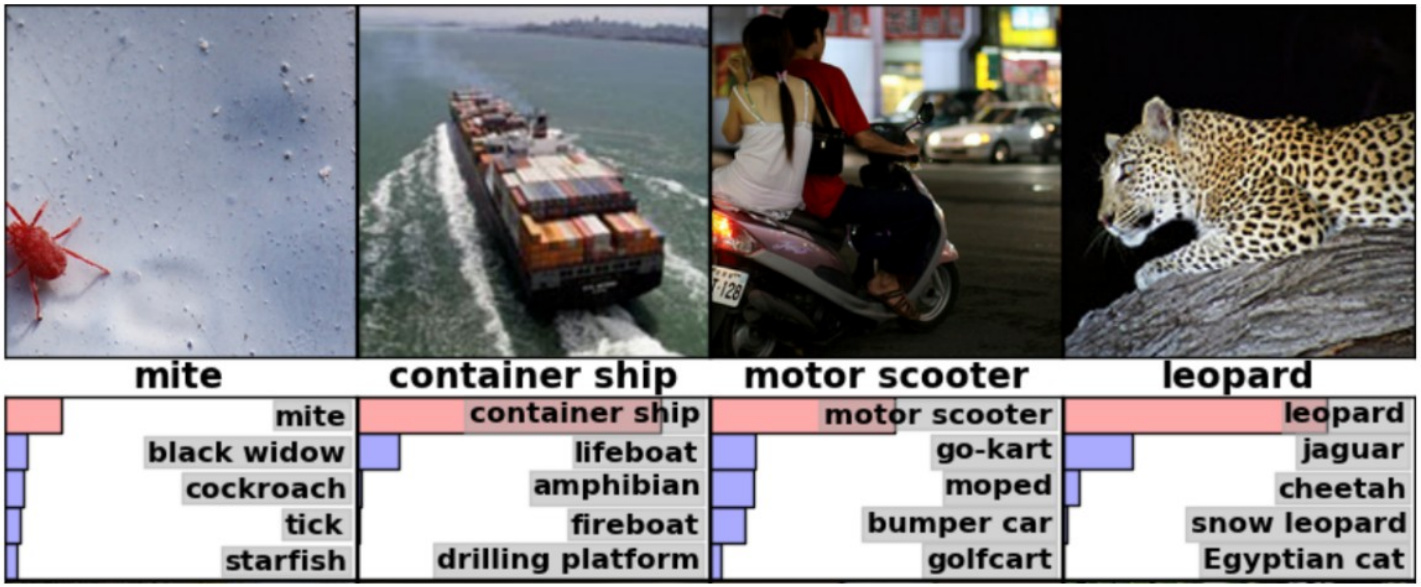
Classic example or has become a classic

- Recognition of handwritten digits
  - MNIST database (Modified National Institute of Standards and Technology database)
  - 60k training images and 10k testing images labeled with correct answer
  - 28 pixel x 28 pixel
  - Algorithms have reached "near-human performance"
  - Smallest error rate (2018): 0.18%

https://en.wikipedia.org/wiki/MNIST_database

Image recognition

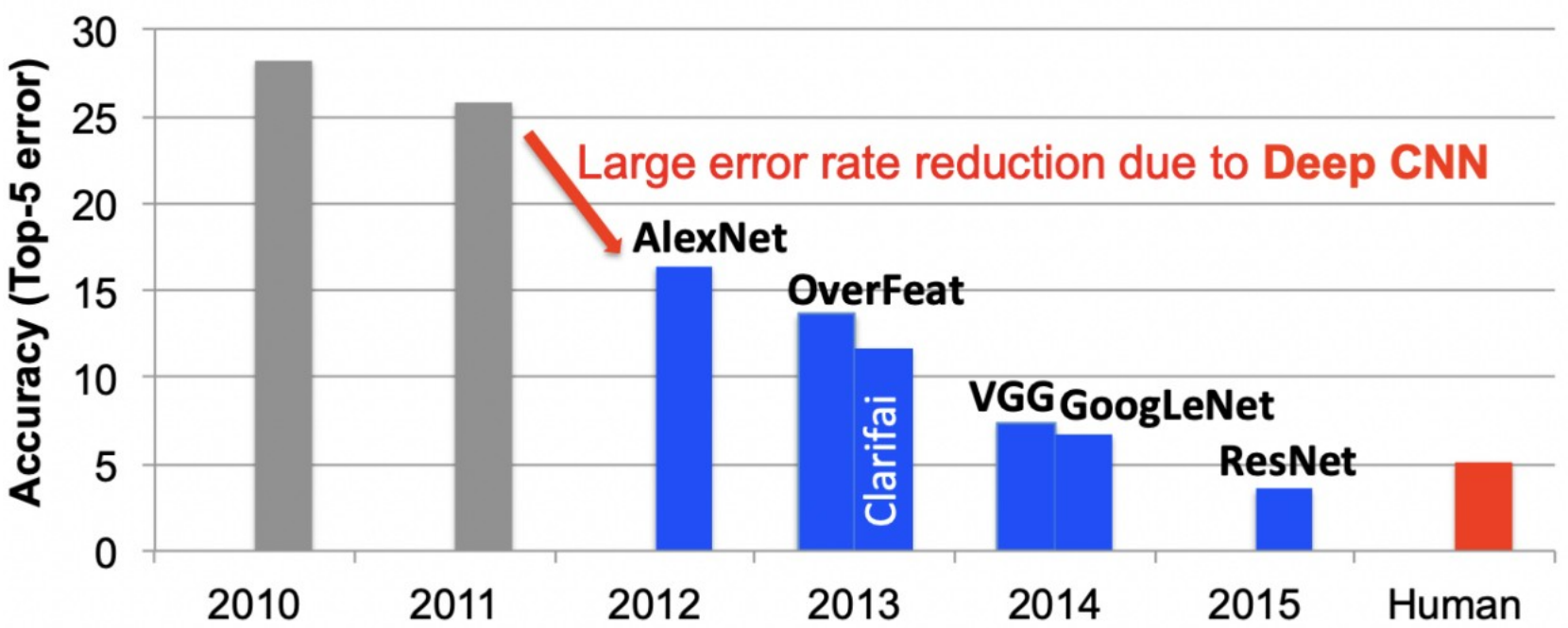
- ImageNet database:
  - 14 million images, 22,000 categories
  - Since 2010, the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC): 1.4 million images, 1000 categories
  - In 2017, 29 of 38 competing teams got less than 5% wrong



https://www.tensorflow.org/tutorials/image_recognition

ImageNet: Large Scale Visual Recognition Challenge

- Error rate in percent:



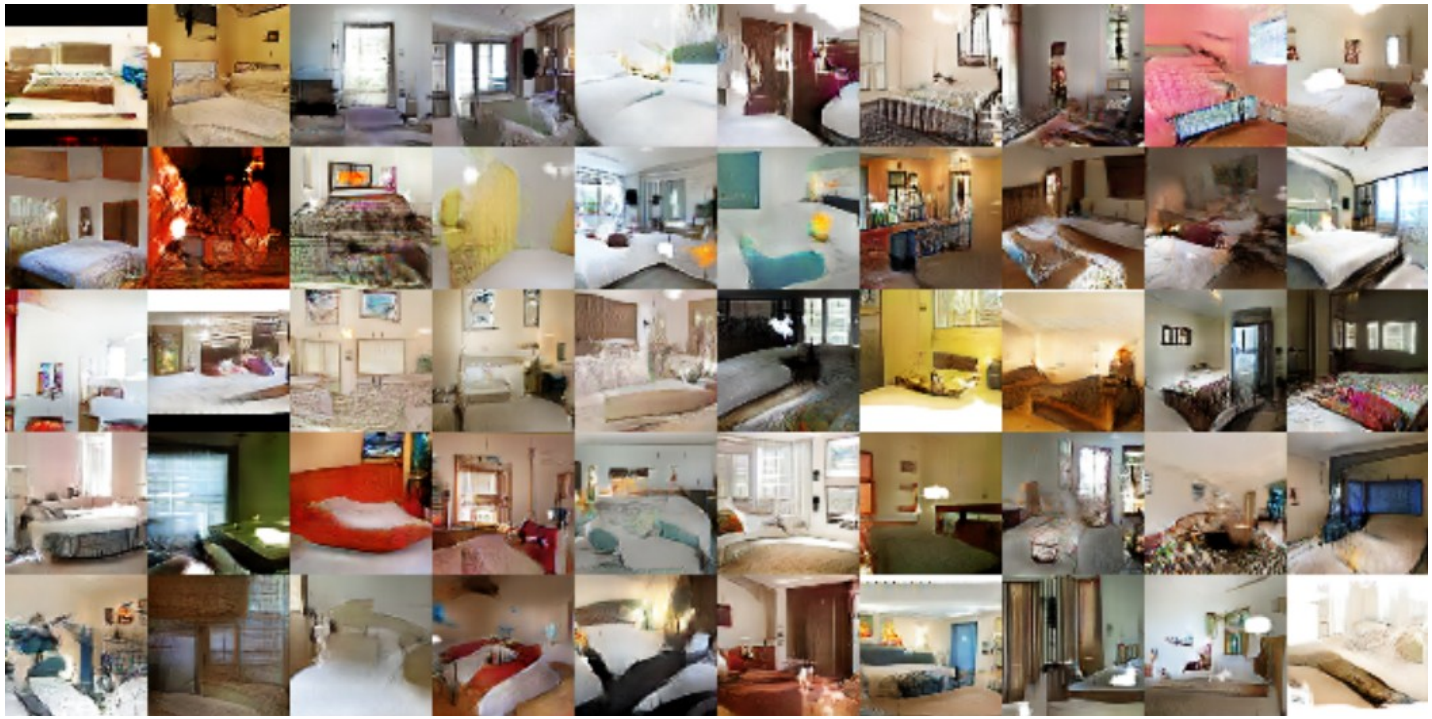
https://arxiv.org/pdf/1703.09039.pdf

Further progresses: some illustrative examples

Object detection and segmentation

K. He et al., *Mask R-CNN* (2017) arXiv:1703.06870

Further progresses: some illustrative examples



Data generation

M. Arjovsky et al, *Wasserstein GAN*, (2017) arXiv:1701.07875

Further progresses: some illustrative examples

a photograph of an astronaut riding a horse

Diffusion model

R. Rombach et al. *Stable diffusion* (2022)
ArXiv:2112.10752

OY can you give an short example of output of chatGPT for illustrating it to master students in nuclear physics

Here is an example of an output of ChatGPT that could be used to illustrate the capabilities of the model to master students in nuclear physics:

"ChatGPT is a powerful language model that can understand and generate scientific text. For example, it can generate a brief explanation of a nuclear physics concept such as beta decay:

"Beta decay is a type of radioactive decay in which a beta particle (an electron or a positron) is emitted from the nucleus of an atom. This process changes the atomic number of the atom, resulting in the formation of a different element or isotope. Beta decay is an important process in nuclear physics as it plays a role in the production of elements in stars and in the decay of certain radioactive isotopes used in medical and industrial applications."

This output provides a clear and concise explanation of beta decay, demonstrating the model's ability to understand and generate scientific text relevant to the field of nuclear physics.

Large Language models

OpenAI, *ChatGPT (2022)*
https://openai.com/models/gpt-3/

Some successes and unsolved problems in AI

- From a textbook in 2020:

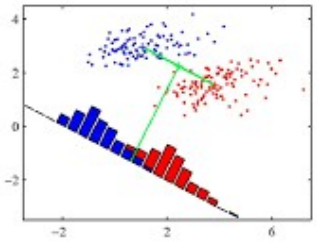
M. Woolridge,
The Road to Conscious Machines

Arithmetic (1945)	Easy
Sorting lists of numbers (1959)	
Playing simple board games (1959)	Solved, after a lot of effort
Playing chess (1997)	
Recognizing faces in pictures (2008)	
Usable automated translation (2010)	
Playing Go (2016)	
Usable real-time translation of spoken words (2016)	
Driverless cars	Real progress
Automatically providing captions for pictures	
Understanding a story & answering questions about it	Nowhere near solved
Human-level automated translation	
Interpreting what is going on in a photograph	
Writing interesting stories	
Interpreting a work of art	
Human-level general intelligence	

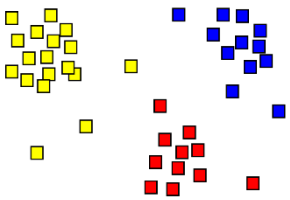
What was done in last 4 years !

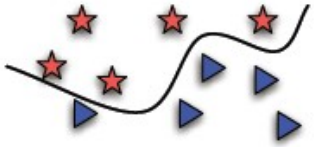
- Image recognition
- Speech recognition
- Recommendation systems
- Automated translation
- Chatbots based on Large Language Models
- AI agents

Types of Machine learning

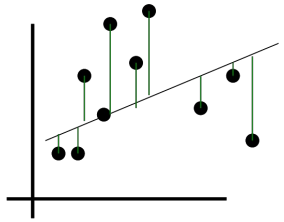


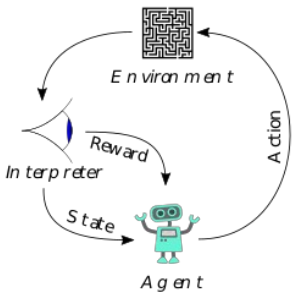
Unlabeled Data





Labeled Data





Machine Learning

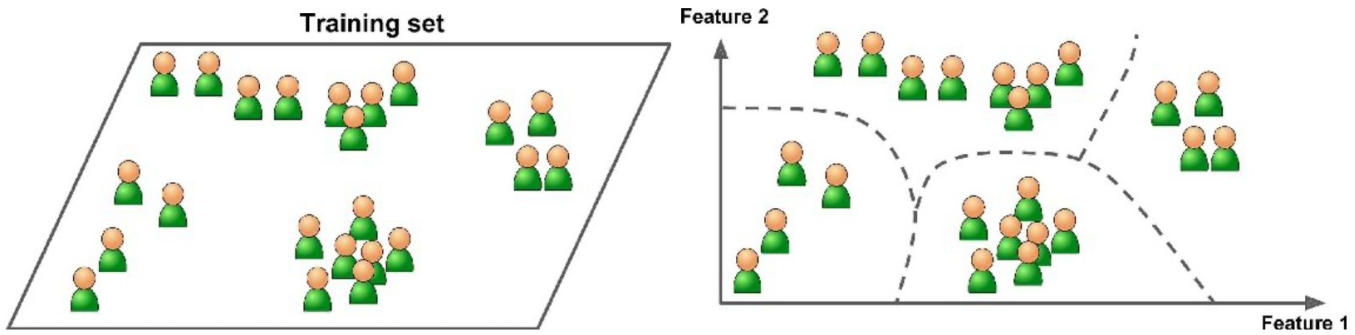
Unsupervised Learning
- Meaningful Compression
- Structure Discovery
- Dimensionality Reduction
- Big data Visualistaion
- Feature Elicitation
- Clustering
- Recommender Systems
- Targetted Marketing
- Customer Segmentation

Supervised Learning
- Classification
- Image Classification
- Customer Retention
- Idenity Fraud Detection
- Diagnostics
- Regression
- Advertising Popularity Prediction
- Weather Forecasting
- Market Forecasting
- Population Growth Prediction
- Estimating life expectancy

Reinforcement Learning
- Real-time decisions
- Game AI
- Robot Navigation
- Skill Acquisition
- Learning Tasks

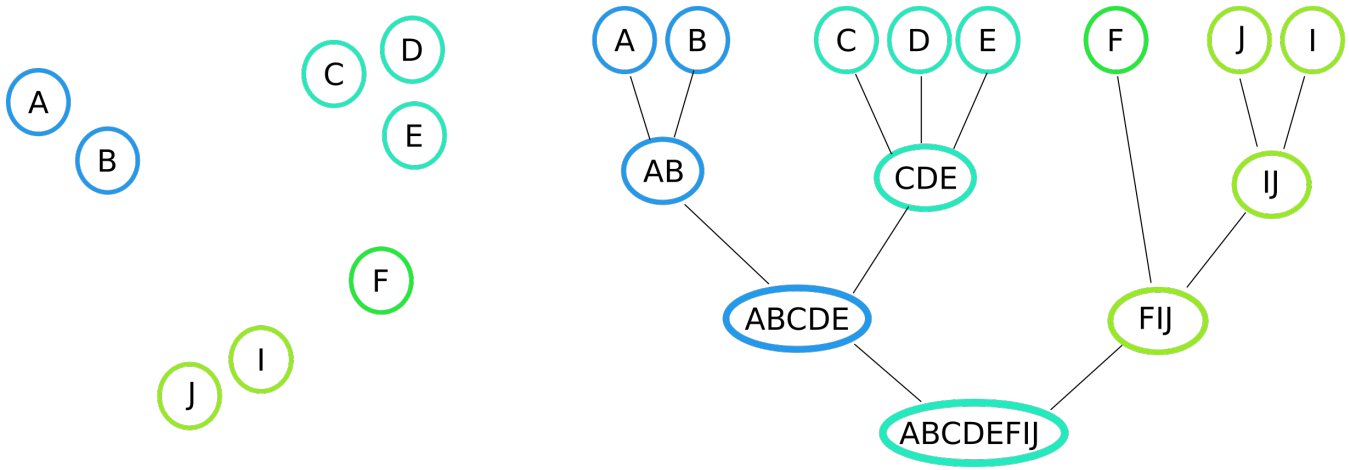
Unsupervised learning

- Important aspects :
  - No Labels or targets
  - No feedback
  - Find hidden structures



Unsupervised learning

- **Main algorithms:**
  - Clustering
    - Hierarchical cluster analysis
      - Needs one metric ($||.||_2$)
      - linkage criteria: d between clusters as a function of the d between observations (complete-linkage clustering $max\{d(a,b): a \in A, b \in B\}$)



Unsupervised learning

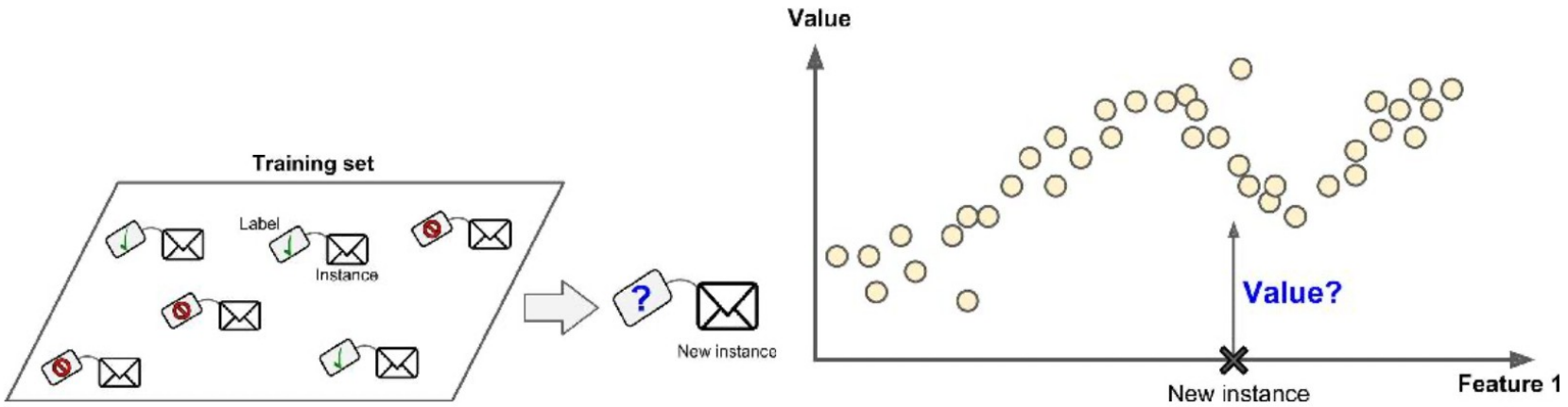
- Main algorithms:
  - Dimensionality reduction → Several aspects
    - high-dimensional datasets & the “curse of dimensionality”
      - When dimension UP, volume space unit hypercube UP, dataset become very sparse → problematic for statistics significance
        - 1D, unit interval & 100 uniformly distributed sample: distance spacing is 10^{-2}
        - 10D unit hypercube, for same lattice spacing needs 10^{20} samples.
    - Reduce dimension of dataset

      → Feature extraction: pre-processing steps for other algorithms

      → Data visualization: sometimes it is nice to also see the data

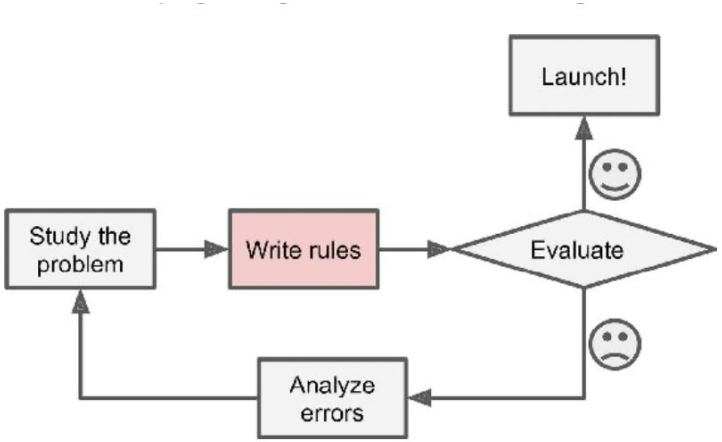
Supervised learning

- Important aspects :
  - Labeled data
  - Direct feedback
  - Predict outcome



Example : Spam detection

- Naive approach
  - Observe what is a spam and detect recurrent patterns
  - write an algorithm of these patterns
  - If a new email contains these patterns then classify it as a spam
  - iterate until convergence

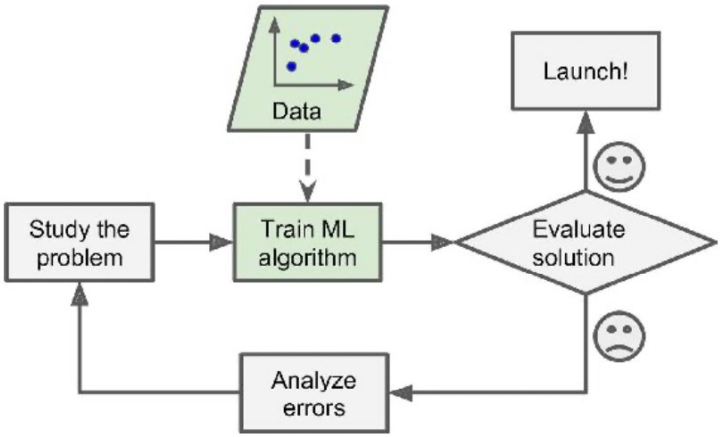


- Complex task
- High nb of rules
- Difficult to update

Example : Spam detection

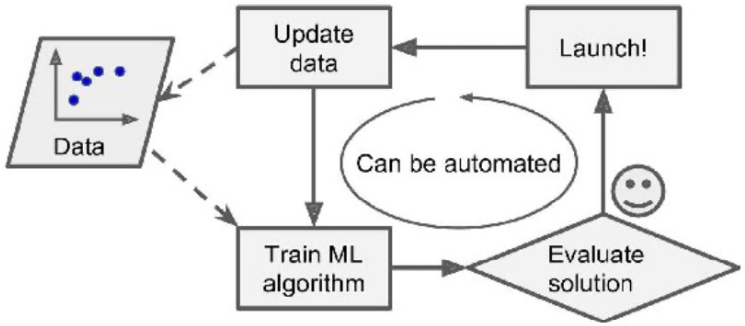
- Machine learning

    1. A ML spam filter automatically learns relevant patterns



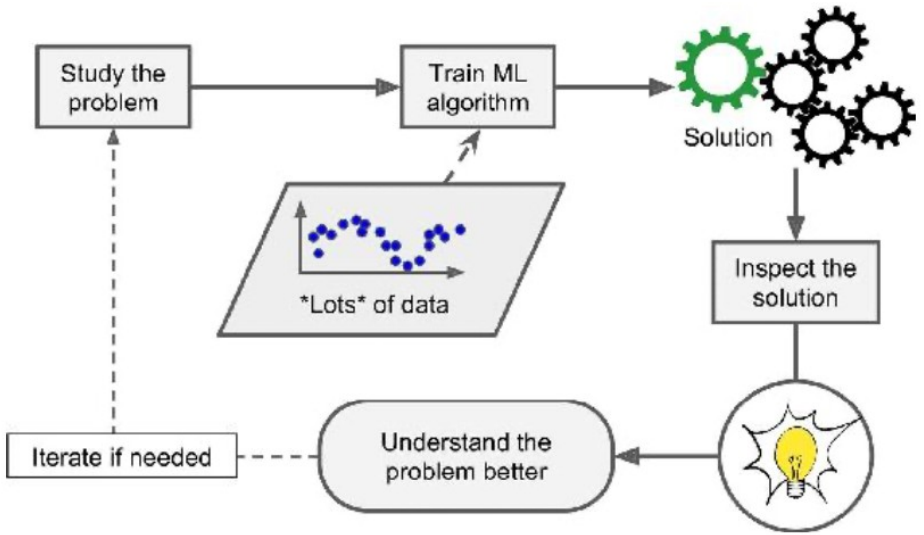
Example : Spam detection

- Machine learning
    1. A ML spam filter automatically learns relevant patterns
    2. Automatic adaptation



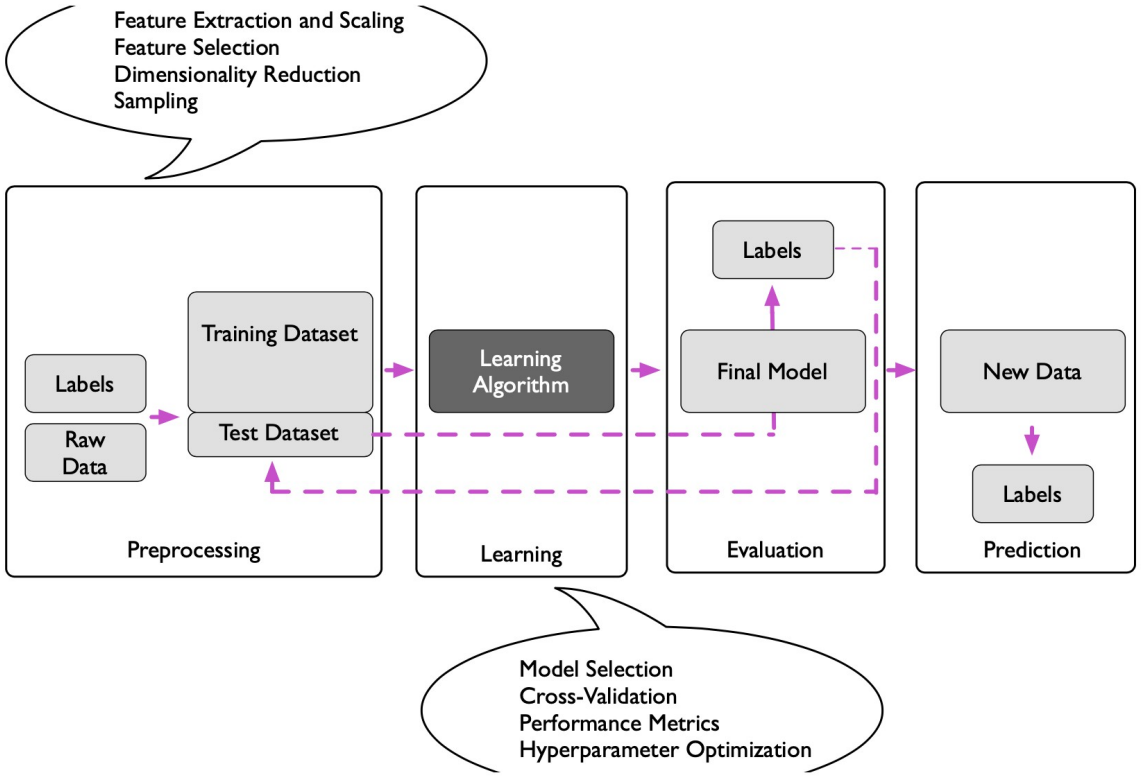
Example : Spam detection

- Machine learning
  1. A ML spam filter automatically learns relevant patterns
  2. Automatic adaptation
  3. Can help humans to learn
  → Data Mining



Supervised learning

- Workflow

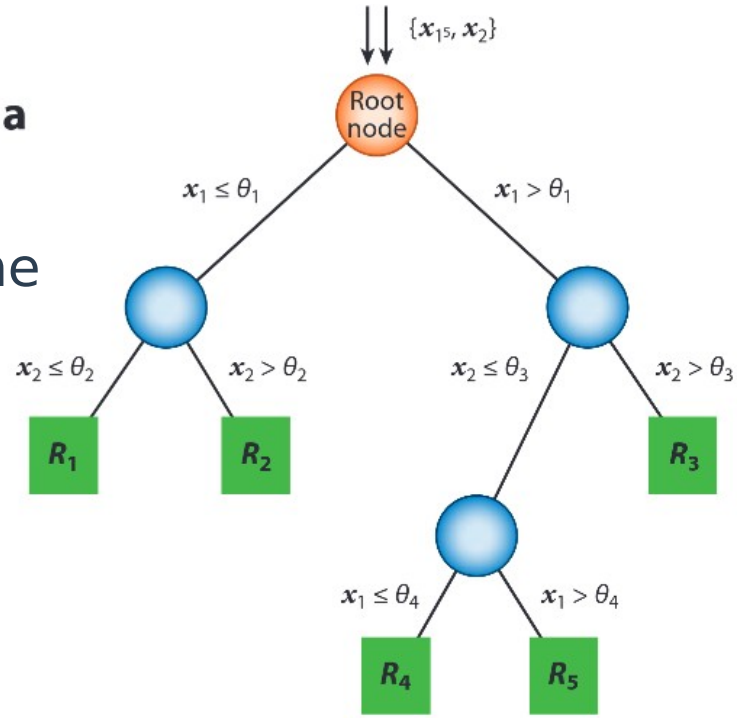


- Instance: *A specific observation of data.*
- Feature: *An measurable property of instance.*
- Criterion/Outcome: *The feature that you want to predict.*
- Model: *Representation or simulation of reality. Typically a simplification based on assumptions*

Supervised learning

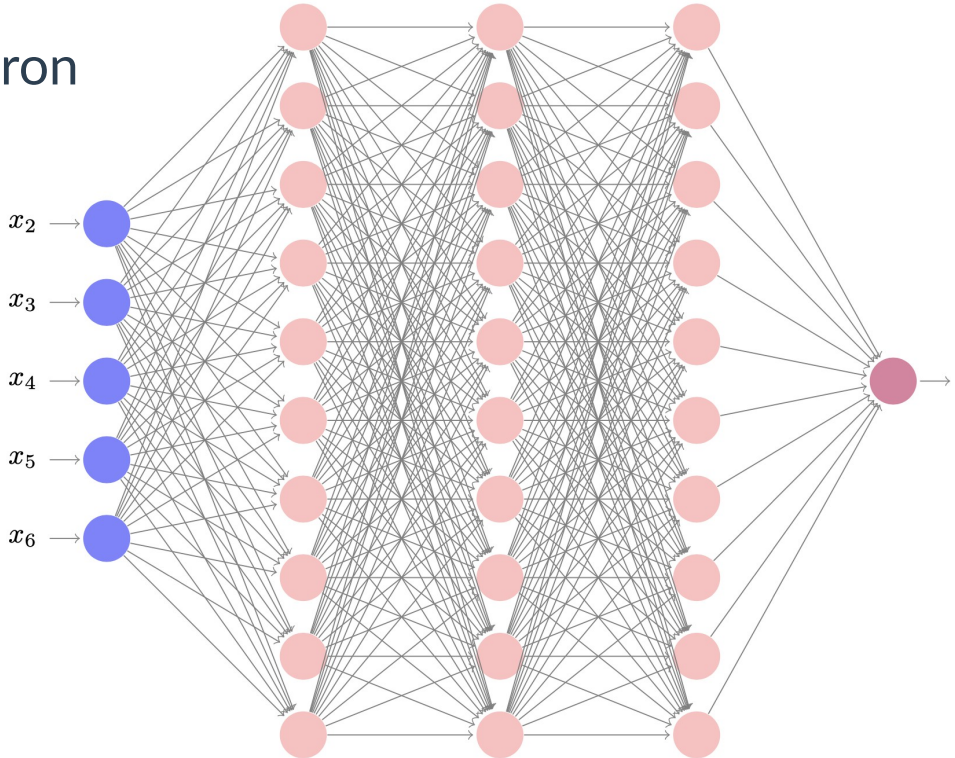
- Main algorithms:
  - Decision Trees :
    - The criterion is modeled as a sequence of logical TRUE or FALSE
    - Recursively partitions the feature space such that the samples with the same labels or similar target values are grouped together.
    - Minimize the impurity:

$$G = \frac{N^{left}}{N} H\left(Set_{left}\right) + \frac{N^{right}}{N} H\left(Set_{right}\right)$$



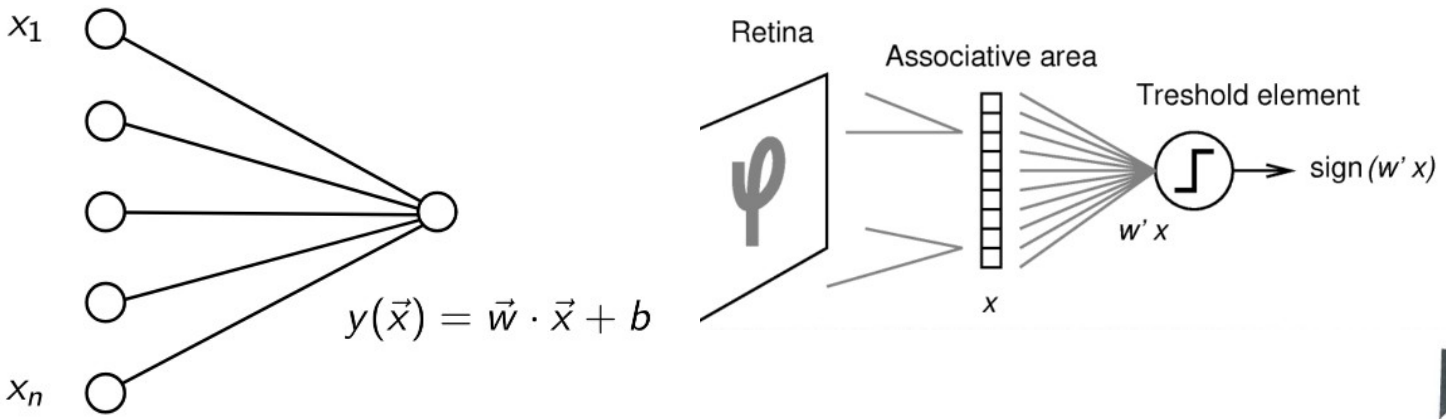
Supervised learning

- Main algorithms:
  - Artificial neural network
    - The biological inspiration: the neuron
    - C. elegans (roundworm):
      - 302 neurons
      - with ~ 25 synaptic connections
    - Human brain:
      - 10^{11} neurons
      - with ~ 7000 synaptic connections
    - Weighting Inputs signals
    - Passing through an activation

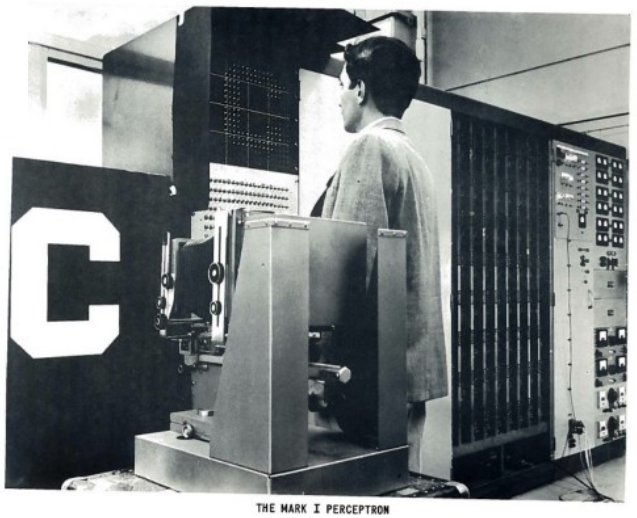


Perceptron

- Idea already from Rosenblatt, 1954:

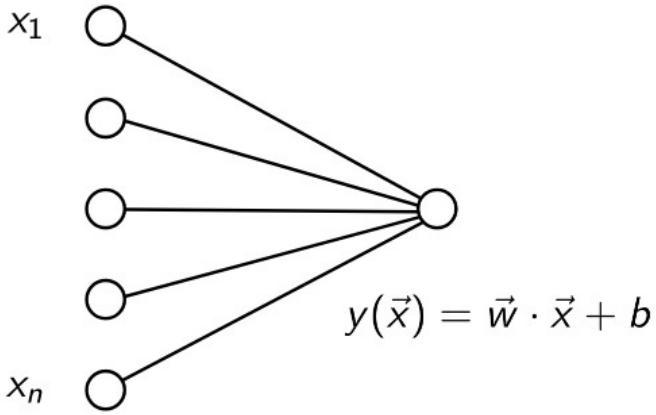


- Perceptron: designed for image recognition
  - It was first implemented in hardware with 400 photocells, weights = potentiometer settings
  - Based on the first mathematical model of a biological neuron of McCulloch–Pitts (1943)


THE MARK I PERCEPTRON

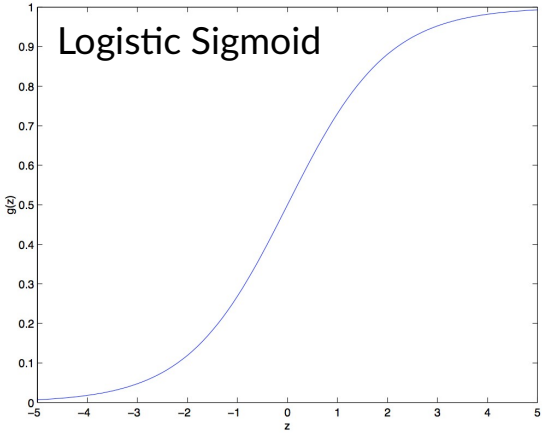
Improvements on the concept

- Non-linear transfer via activation function:

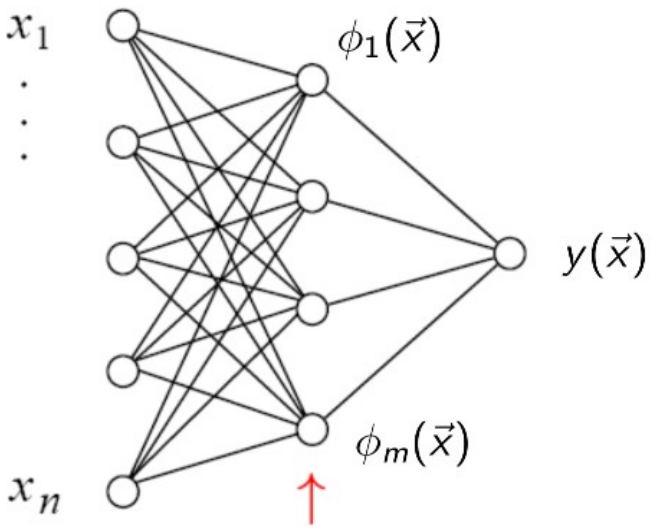


$$y(\vec{x}) = h\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)$$

- Example for h: sigmoid $\frac{1}{1+e^{-x}}$
- Non-linear activation function: when feature space is not linearly separable
- linear activation functions is just a perceptron



Feed Forward Neural Network



Hidden layer
Composed of *neurons*

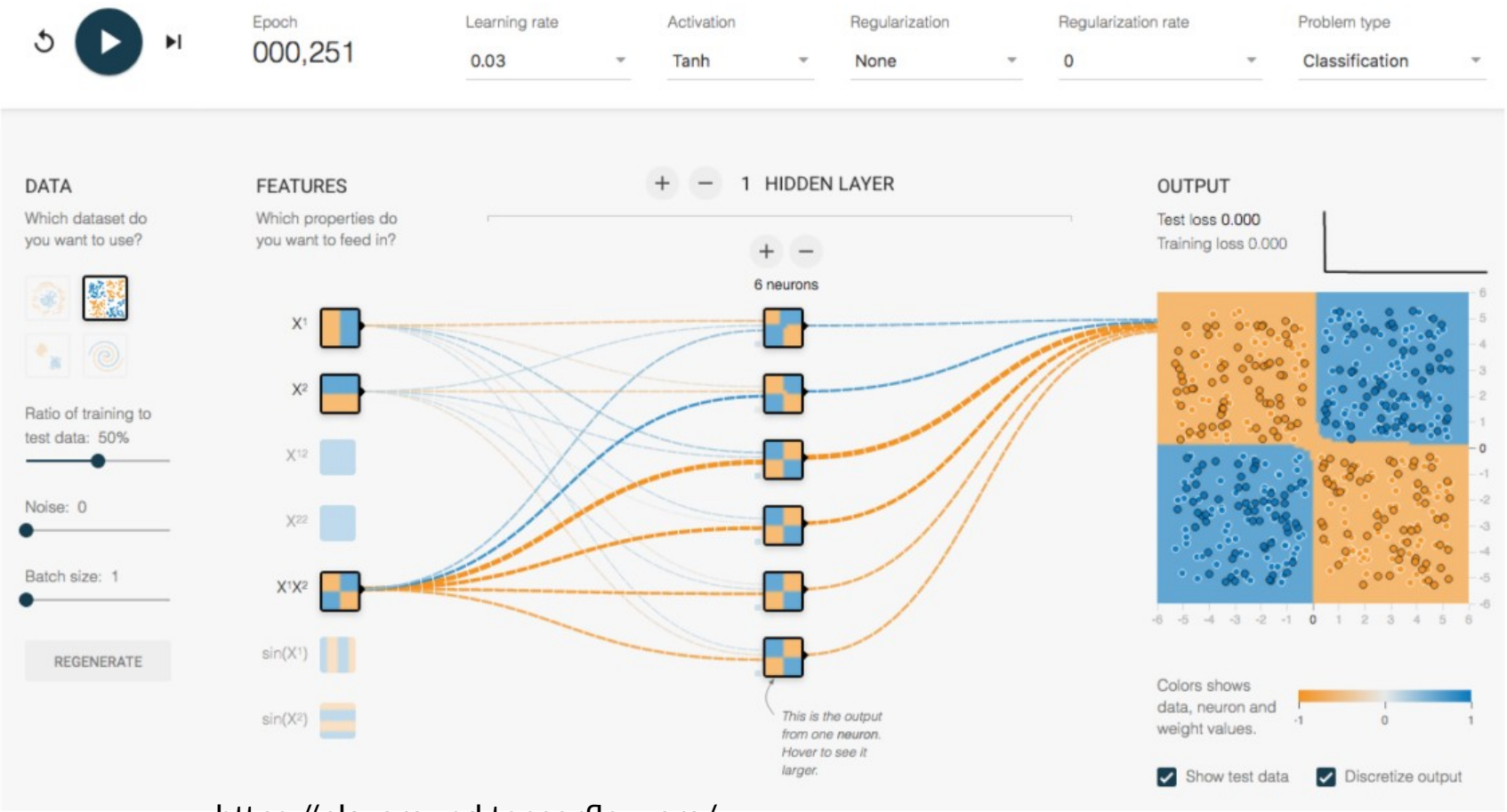
superscripts indicates layer number

$$\phi_i(\vec{x}) = h\left(w_{i0}^{(1)} + \sum_{j=1}^{n} w_{ij}^{(1)} x_j\right)$$

$$y(\vec{x}) = h\left(w_{10}^{(2)} + \sum_{j=1}^{m} w_{1j}^{(2)} \phi_j(\vec{x})\right)$$

Straightforward to generalize to multiple hidden layers

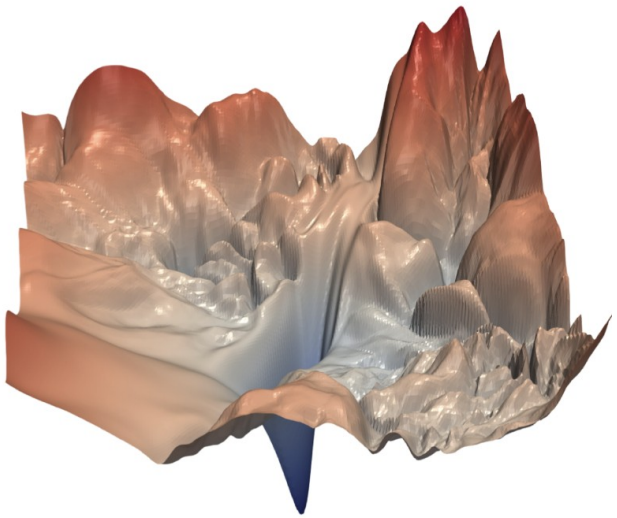
Try in the browser



https://playground.tensorflow.org/

Howto ?

- Network training:
  - An optimization problem: Find optimal weights to solve my problem
    → Need of a loss function on which we can action to find optimal
    - Example: Squared error loss (regression), Cross entropy (classification)
  - Usage of gradient descent $\vec{w}^{(\tau+1)} = \vec{w}^{(\tau)} - \eta \nabla E_a(\vec{w}^{(\tau)})$
  - Example of a loss landscape of a modern artificial neural network:

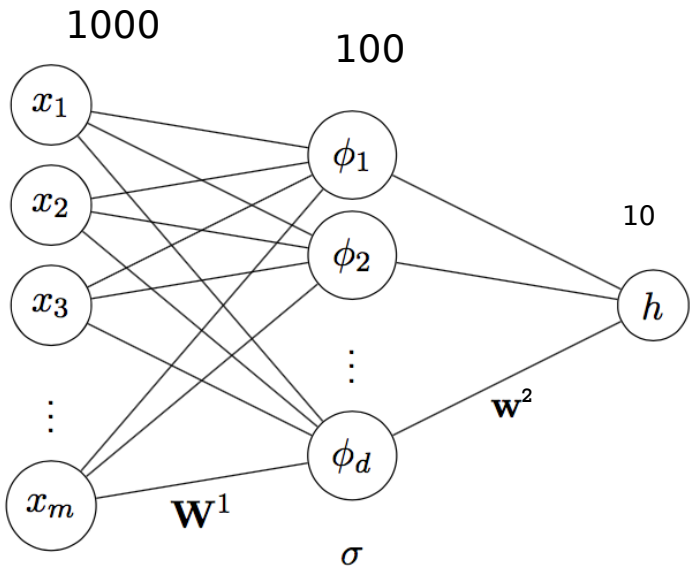


H. Li et al.
https://papers.nips.cc/paper_files/paper/2018/hash/a41b3bb3e6b050b6c9067c67f663b915-Abstract.html

Demystify neural networks

- Full implementation of training of 2-layer NN :

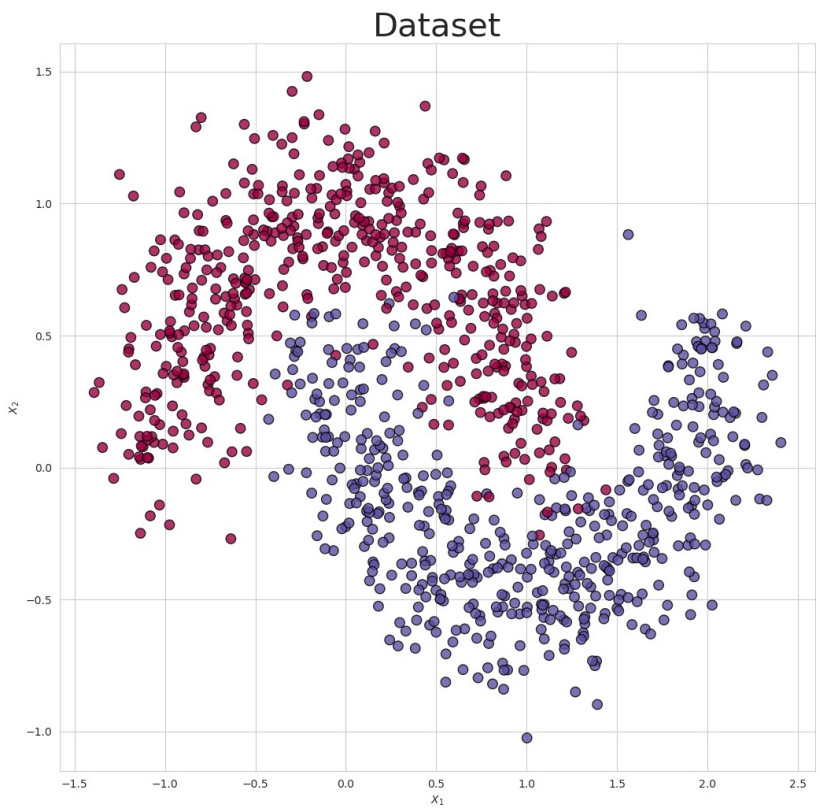
```
1   import numpy as np
2   from numpy.random import randn
3
4   N, D_in, H, D_out = 64, 1000, 100, 10
5   x, y = randn(N, D_in), randn(N, D_out)
6   w1, w2 = randn(D_in, H), randn(H, D_out)
7
8   for t in range(2000):
9     h = 1 / (1 + np.exp(-x.dot(w1)))
10    y_pred = h.dot(w2)
11    loss = np.square(y_pred - y).sum()
12    print(t, loss)
13
14    grad_y_pred = 2.0 * (y_pred - y)
15    grad_w2 = h.T.dot(grad_y_pred)
16    grad_h = grad_y_pred.dot(w2.T)
17    grad_w1 = x.T.dot(grad_h * h * (1 - h))
18
19    w1 -= 1e-4 * grad_w1
20    w2 -= 1e-4 * grad_w2
```



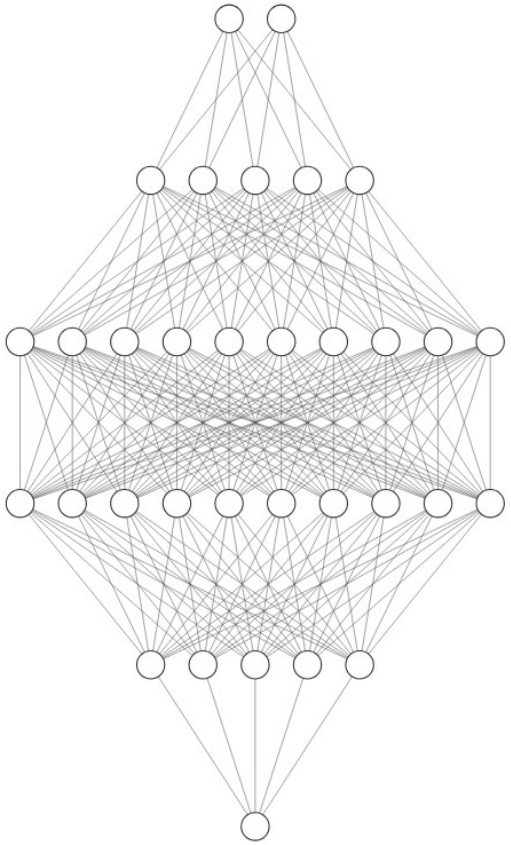
Optimization part:
gradient descent
via “back propagation”

Example of a training

Data



Model



(x,y)

Activation (25 n)

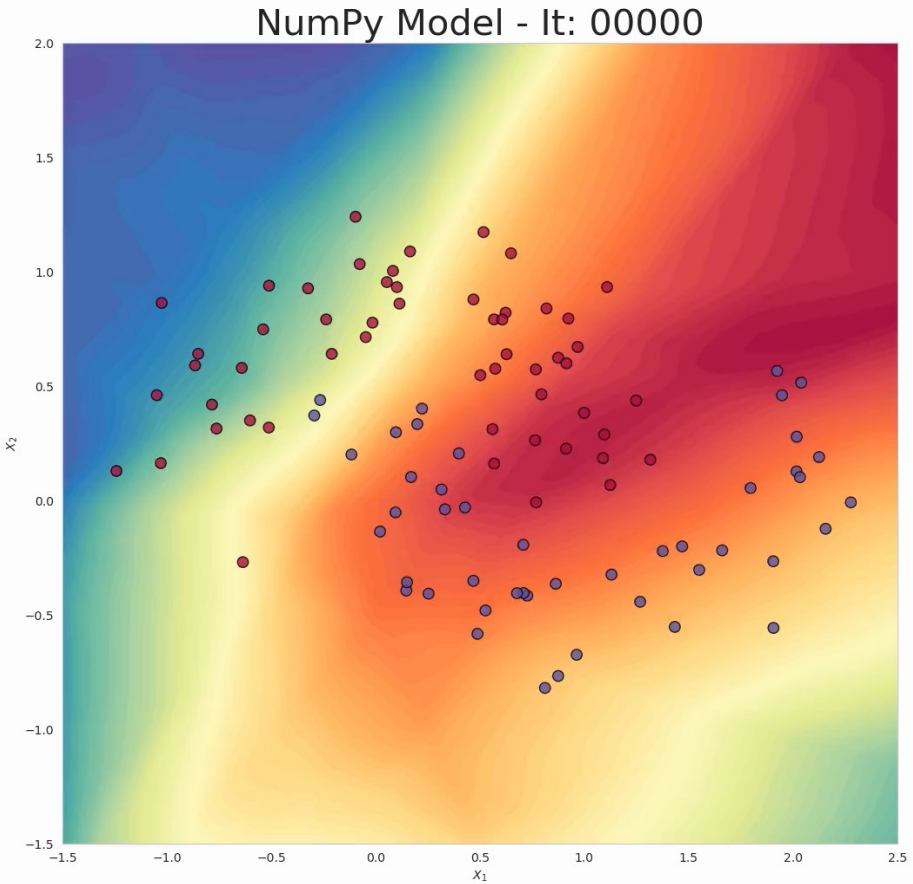
Activation (50 n)

Activation (50 n)

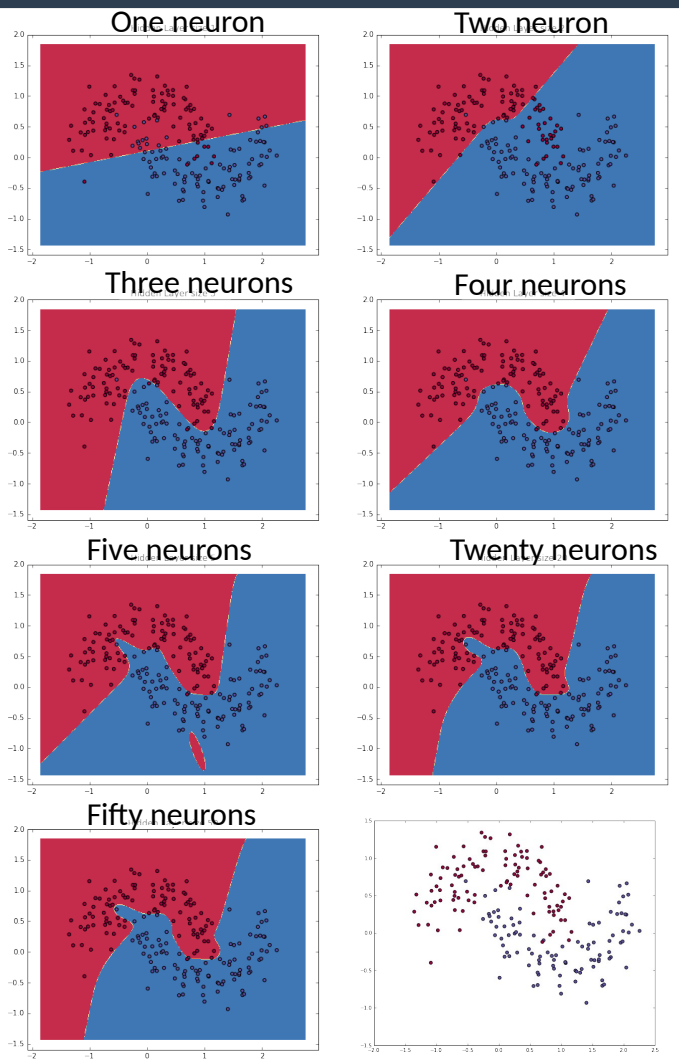
Activation (25 n)

Color (R,B)

Step by step evaluation of the training

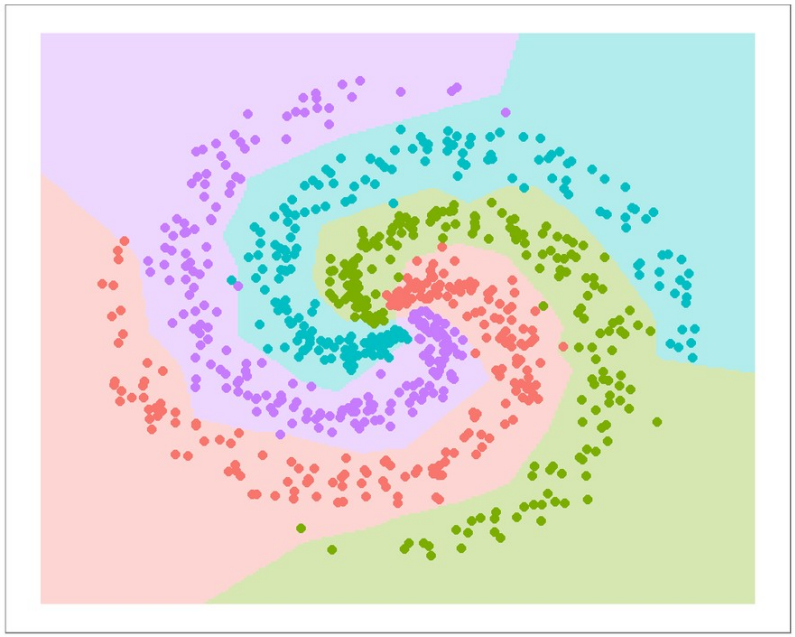


Neural Network Decision Boundaries

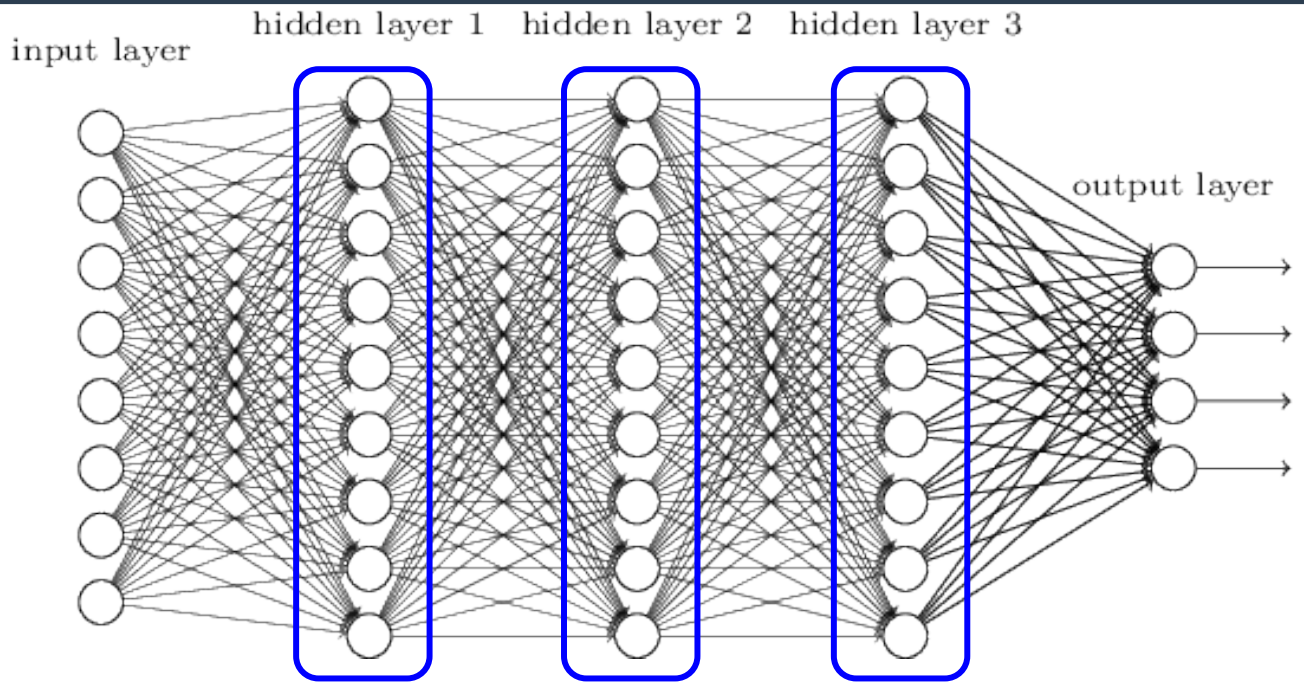


2-class classification
1-hidden layer NN
L2 norm regularization

4-class classification
2-hidden layer NN
ReLU activations
L2 norm regularization



Deep Neural Networks



- As data complexity grows, need exponentially large number of neurons in a single-hidden-layer network to capture all structure in data
- Deep neural networks factorize the learning of structure in data across many layers:
  - Universal approximation theorem (1989): https://link.springer.com/article/10.1007/BF02551274
- Challenges: Hard and slow to train & risk of overtraining

http://neuralnetworksanddeeplearning.com

Cooking recipe in ML

- Get data (loads of them) & good hardware
- Algorithm to choose ?
  - Structured data: "High level" features that have meaning
    - feature engineering + decision trees / Random forests / XGBoost
  - Unstructured data: "Low level" features, no individual meaning
    - deep neural networks / images → convolutional NN
- But pitfalls to be aware of:
  - Data quality : Garbage In → Garbage Out / Missing data ?
  - Underfitting / Overfitting
  - Simplicity don’t imply better generalization
  - Appropriate evaluation metric
  - Mistaking correlation for causation & confounding variables

Learning resources

- Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, free online https://www.deeplearningbook.org/
- Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, and Thomas B. Schön, Machine Learning – A First Course for Engineers and Scientists https://smlbook.org/
- Simon J.D. Prince, Understanding Deep Learning https://udlbook.github.io/udlbook/
- Kevin Patrick Murphy, Probabilistic Machine Learning, https://probml.github.io/pml-book/
- Aurélien Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow

Useful libraries

- scikit-learn, https://scikit-learn.org/
- PyTorch, https://pytorch.org/
- TensorFlow, https://www.tensorflow.org/
- XGBoost, https://xgboost.ai/

Any questions ?

christophe.rappold@csic.es