LIÈGE université
GIGA
CRC In vivo Imaging

Machine Learning principles

What are we talking about?

fnrs
LA LIBERTÉ DE CHERCHER

Christophe Phillips, Ir Ph.D.

Outline

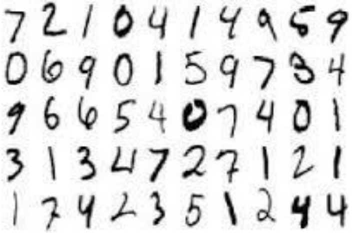
- Introduction
- Deep learning concepts
- Clustering
- Linear predictive models
- Conclusions

Pattern recognition

Pattern recognition aims

- to find patterns/regularities in the data that can be used
- to produce an output (e.g. make predictions or take actions).

Digit Recognition



Face Recognition

Recommendation Engines

NETFLIX amazon

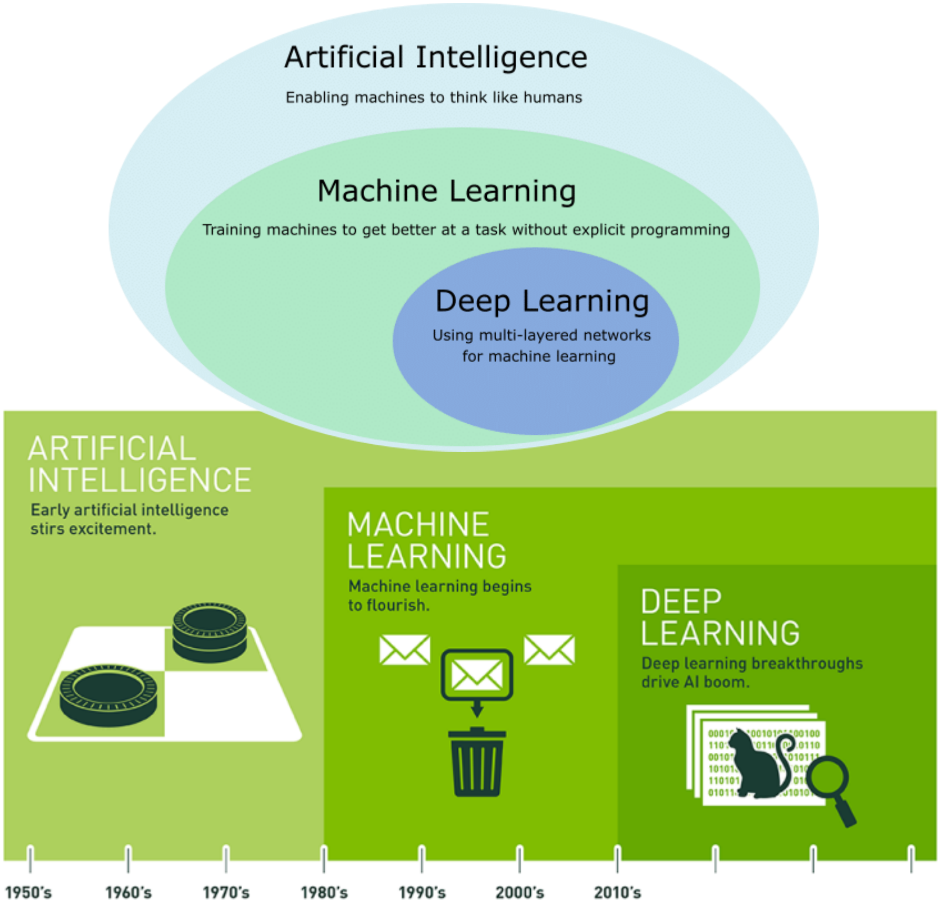
Types of Learning:

- *Supervised* learning:
  trained with labeled data (classification/regression)

- *Unsupervised* learning:
  trained with unlabeled data (clustering)

- *Reinforcement* learning:
  actions and rewards (maximize cumulative reward)

AI vs. Machine learning vs. Deep learning


Artificial Intelligence
Enabling machines to think like humans
Machine Learning
Training machines to get better at a task without explicit programming
Deep Learning
Using multi-layered networks for machine learning
ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

AI & ML:

- Self driving cars
- Playing chess, Go, poker,...
- Text/speech translation
- Info bots & text generation
- Protein folding solving
- Banking and admin duties
- Medical diagnostic & prognostic
- …

https://www.7wdata.be/big-data/ai-vs-machine-learning-vs-deep-learning/

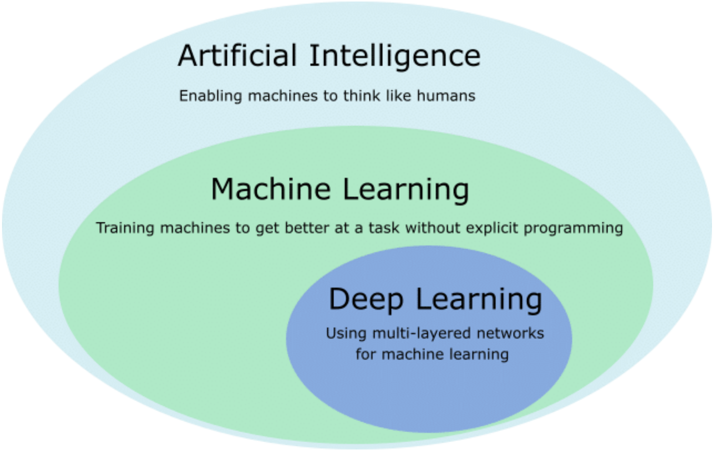
Example: chess & go

“Deep(er) blue” vs. Kasparov (1996 & 1997):

- Relies on dedicated hardware & software
- “Brute force” approach

“Stockfish” (2004 – now)

- Open source & free chess engine
  → download and play against it!
- Relies on (pre-trained) neural networks
- Best official ELO ranking ever


Artificial Intelligence

Enabling machines to think like humans

Machine Learning

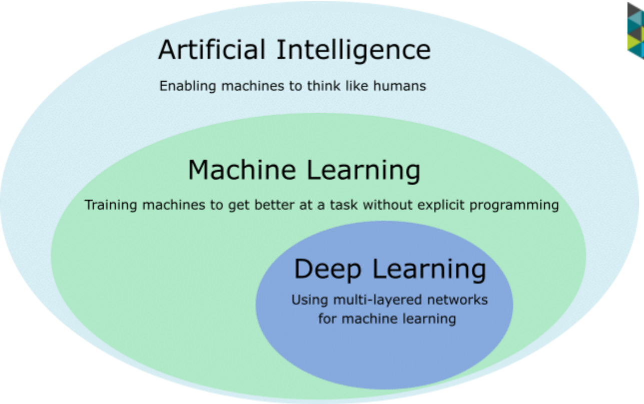
Training machines to get better at a task without explicit programming

Deep Learning

Using multi-layered networks for machine learning

https://en.wikipedia.org/wiki/Deep_Blue_versus_Garry_Kasparov
https://fr.wikipedia.org/wiki/Stockfish_(programme_d%27%C3%A9checs)

Example: chess & go



Artificial Intelligence

Enabling machines to think like humans

Machine Learning

Training machines to get better at a task without explicit programming

Deep Learning

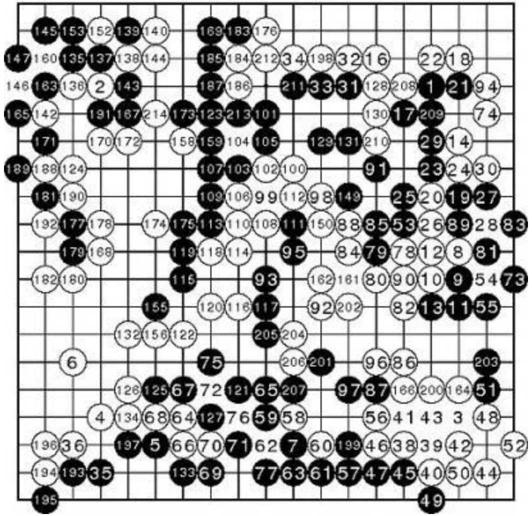
Using multi-layered networks for machine learning

“AlphaGo” (2017)

- From DeepMind company (now part of Google)
- beat Lee Seedol, Go world champion

“AlphaZero” (2018 – now)

- Can play *chess*, *shogi* and *go*
- Relies on neural networks
- Superhuman level of play in a few hours of training
- **No guiding, just self-training !**
- Training requires heavy computing,
- Playing with a “simple” computer



https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol
https://en.wikipedia.org/wiki/AlphaZero

Example: text/image generator

OpenAI company (founded in 2015)

- chatGPT (2022 – now), https://chatgpt.com/auth/login
- DALL-E (2021 – now), https://openai.com/index/dall-e-3/

Google/Deepmind

- Gemini 2.0 (2022 – now)

« Il peut comprendre et interagir avec l'audio et la vidéo, et générer du texte (poésie, scripts, pièces musicales, courriels, lettres, etc.), du code, des traductions (entre plus de 100 langues). Il peut produire divers types de contenu créatif (images, dessins, sons, musique, vidéos...), aider des chercheurs en analysant des données ou en générant des hypothèses. »

- https://deepmind.google/technologies/gemini/

https://fr.wikipedia.org/wiki/Google_Gemini
https://fr.wikipedia.org/wiki/ChatGPT

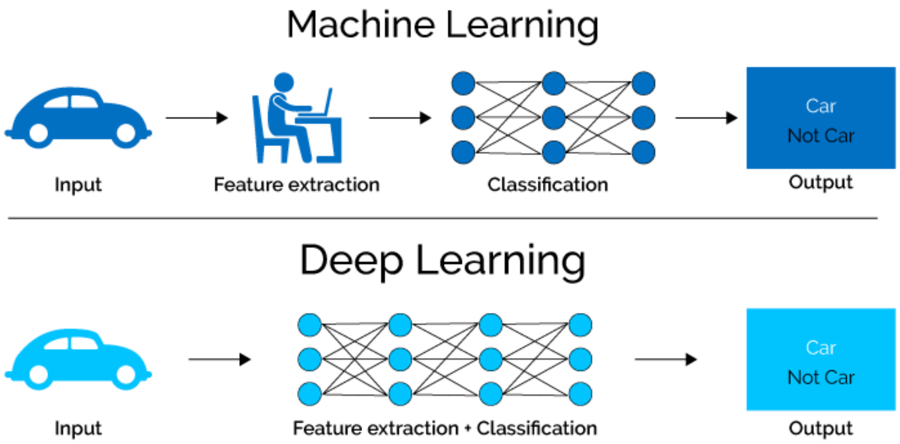
Outline

- Introduction
- Deep learning concepts
- Clustering
- Linear predictive models
- Conclusions

What about deep learning

Deep learning constraints & features:

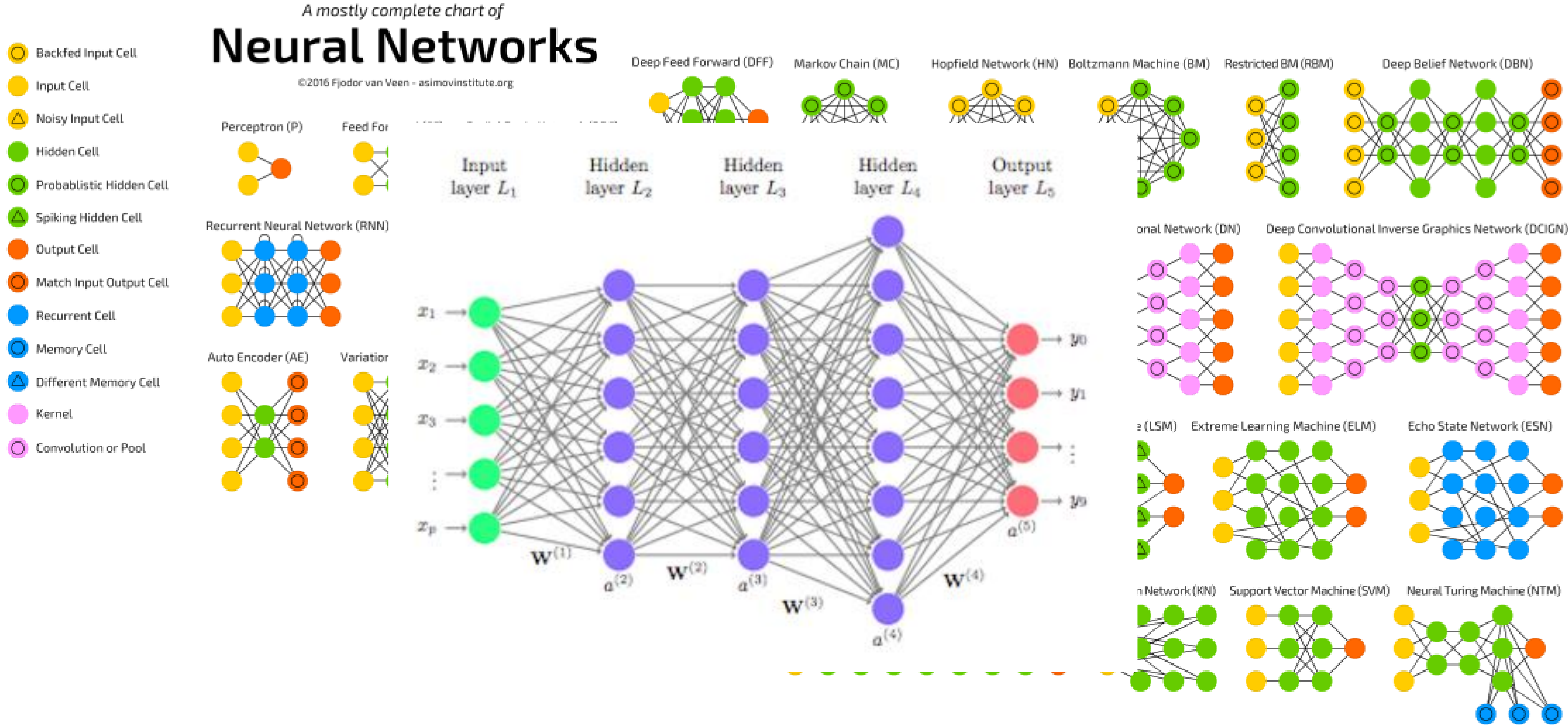
- Architecture choice → depends on data type and problem
- Amount of data → **enormous** (& well curated)
- Application & objective
  → playing a game
  ≠ medical diagnosis
  ≠ driving a car
  ≠ generating text/images
- Raw data input



Schulz et al. (2020)

https://www.datasciencecentral.com/profiles/blogs/concise-visual-summary-of-deep-learning-architectures

Deep learning architectures



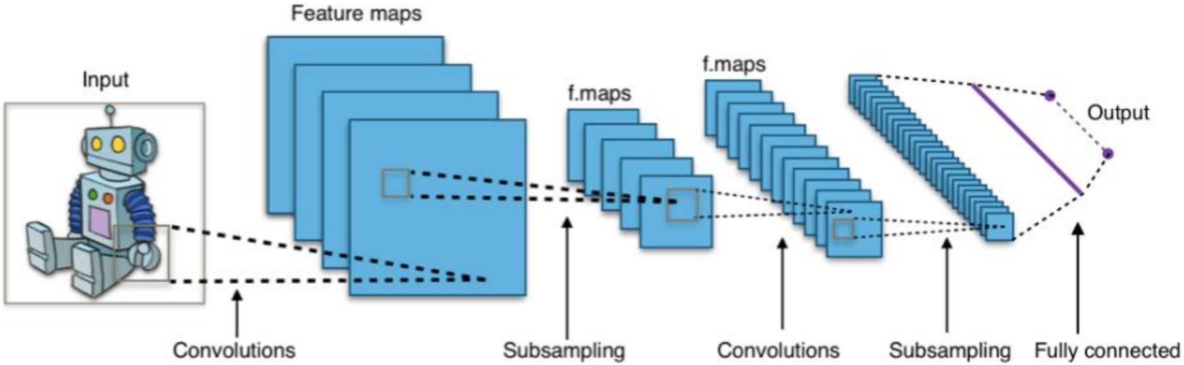
Schulz et al. (2020)

https://www.datasciencecentral.com/profiles/blogs/concise-visual-summary-of-deep-learning-architectures

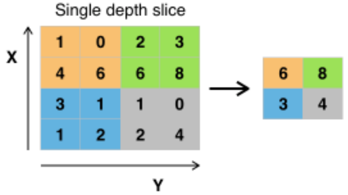
Convolutional neural network, CNN

Most commonly applied to analyze visual imagery

- Building blocks:
  - Subsampling
  - Convolution
  - Pooling
  - …
- Still 1000’s of parameters to optimize!



Max pooling with a 2x2 filter and stride = 2

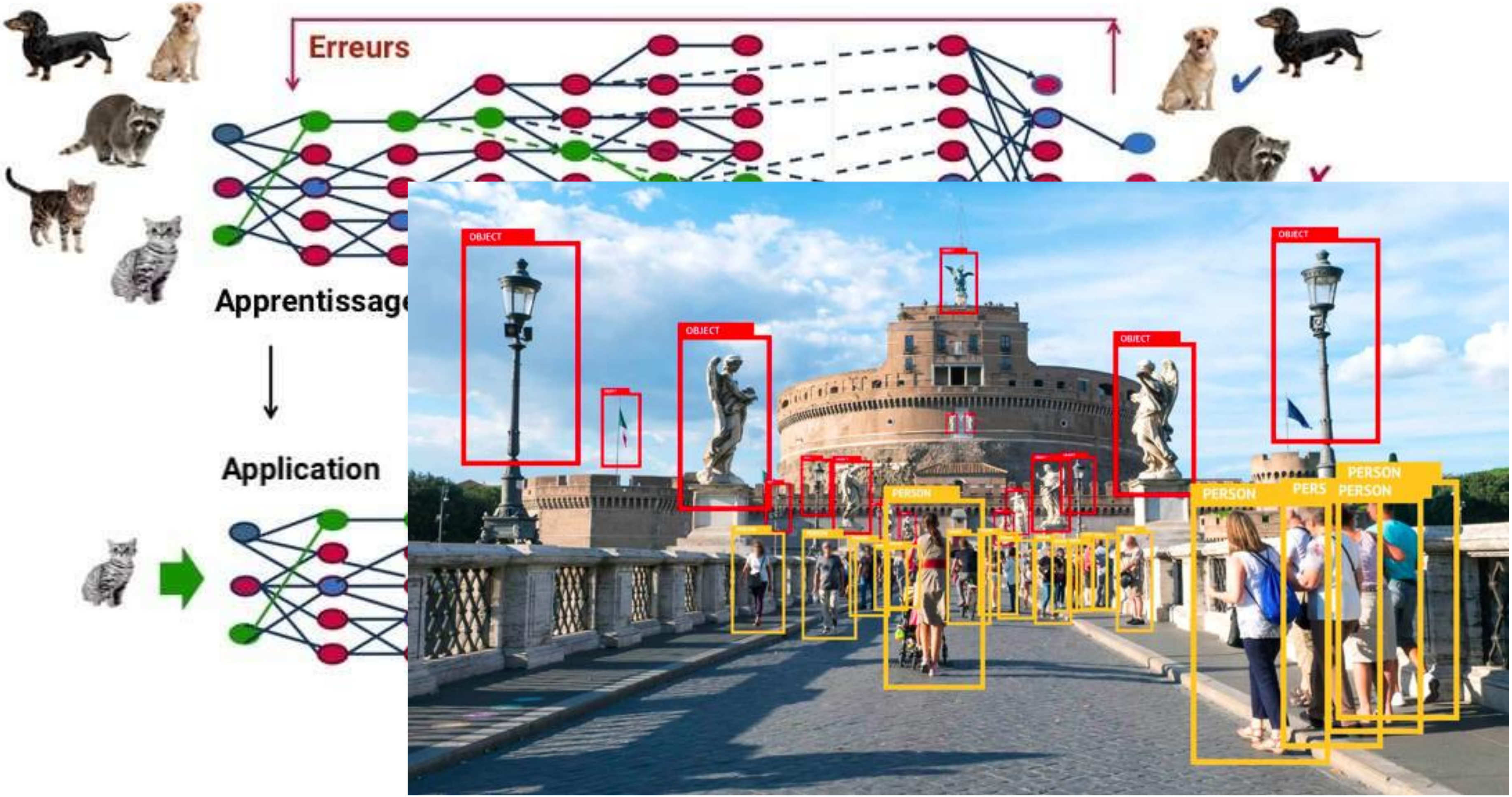


1	0	2	3
4	6	6	8
3	1	1	0
1	2	2	4

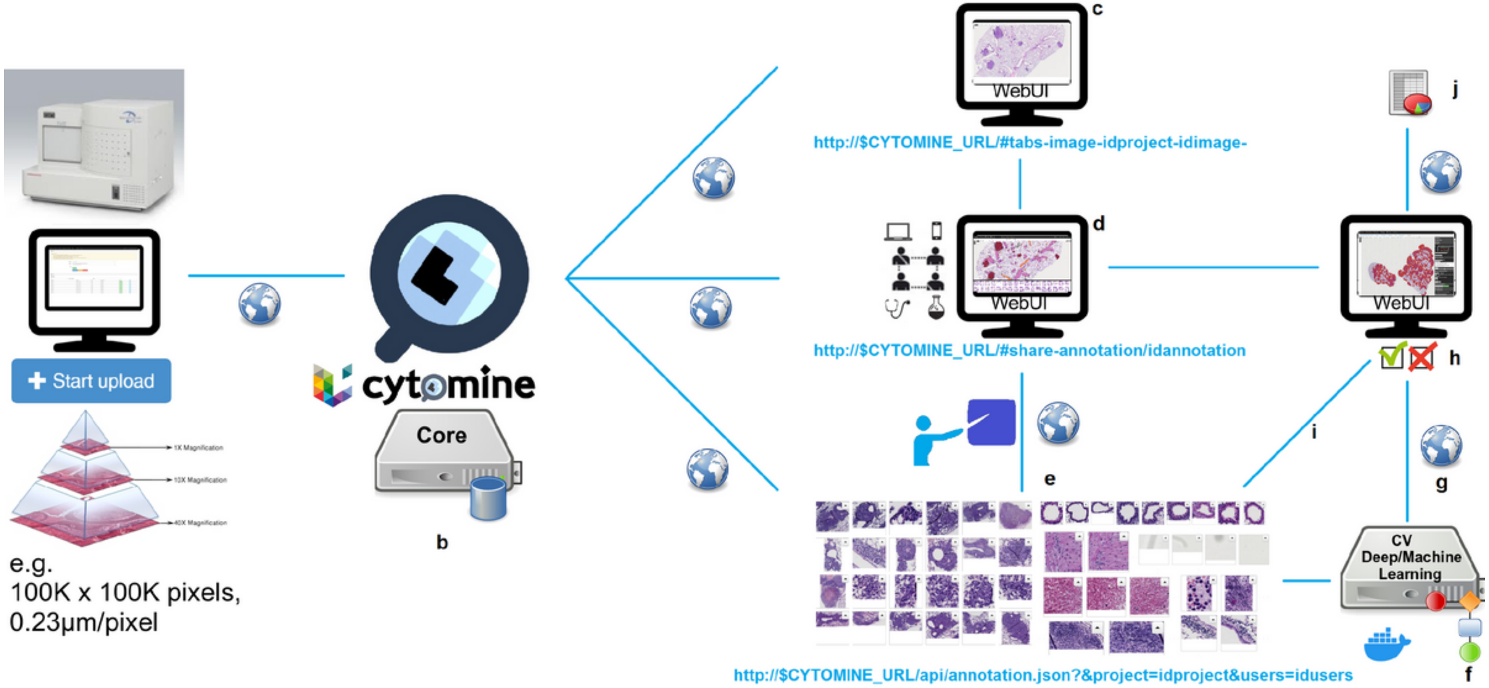
→

6	8
3	4

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


Erreurs
Apprentissag
Application
OBJECT
OBJECT
OBJECT
OBJECT
PERSON
PERSON
PERS
PERSON
PERSON

Medical application


Start upload
1X Magnification
10X Magnification
40X Magnification
e.g.
100K x 100K pixels,
0.23µm/pixel
cytomine
Core
b
WebUI
c
http://$CYTOMINE_URL/#tabs-image-idproject-idimage-
WebUI
d
http://$CYTOMINE_URL/#share-annotation/idannotation
e
WebUI
j
h
i
g
CV
Deep/Machine
Learning
f
http://$CYTOMINE_URL/api/annotation.json?&project=idproject&users=idusers

(Marée et al., Bioinformatics 2016 ; Rubens et al., Proteomics Clin Appl. 2019 ; Rubens et al., Cell Patterns, 2020)

DL challenges

Deep learning neural networks learn a mapping function from training input/output examples.

- **Training = 1000000's of parameters to optimize!**
- Nonlinear activation functions ⟶ not a convex problem
- No way to directly
  - compute the optimal set of weights for a model,
  - nor global convergence guarantees to find an optimal set of weights
- Use "stochastic gradient descent" through "backpropagation"

https://machinelearningmastery.com/a-gentle-introduction-to-the-challenge-of-training-deep-learning-neural-network-models/

DL training

- **Training = 1000000's of parameters to optimize!**
- Use "stochastic gradient descent" through "backpropagation"
- Huge computational load
  - loads of "small operations", one per node
  - loads of iterations

→ parallel operations

→ use GPU's (and "Tensor chip" from Google)

https://machinelearningmastery.com/a-gentle-introduction-to-the-challenge-of-training-deep-learning-neural-network-models/

DL application for language generation

Large Language Models (LLMs)

- ChatGPT from OpenAI
- Gemini from Google-DeepMind
- Deepseek from « China », https://www.deepseek.com/

→ *Generate realistic textual answers to "any questions"!*

…but "*ChatGPT sometimes writes plausible-sounding but incorrect or nonsensical answers.*"

Further read: https://www.newyorker.com/tech/annals-of-technology/chatgpt-is-a-blurry-jpeg-of-the-web

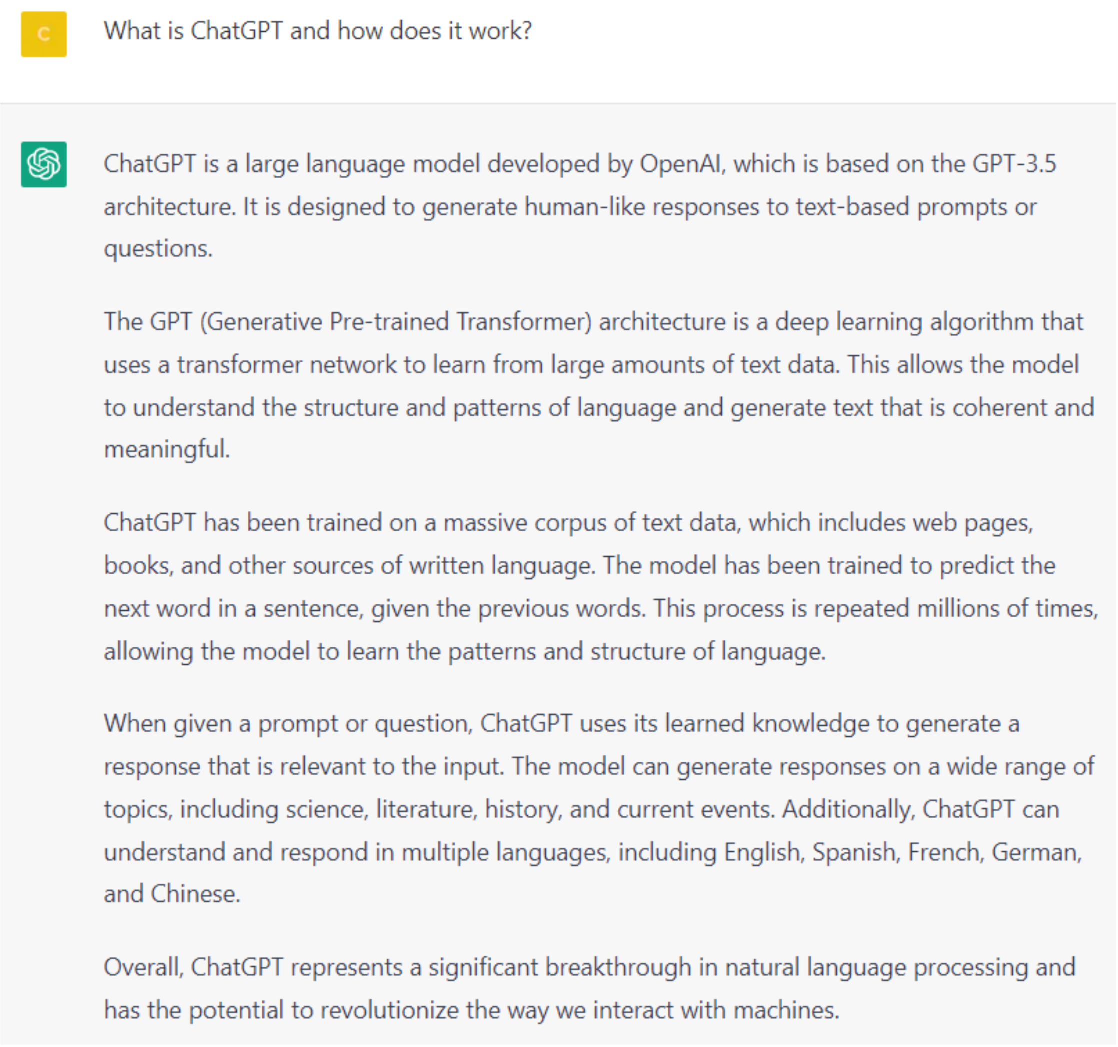
https://fr.wikipedia.org/wiki/ChatGPT

What is ChatGPT?

Let’s ask ChatGPT to describe itself…

https://chat.openai.com/chat

Let’s try it on any other question!



What is ChatGPT and how does it work?

ChatGPT is a large language model developed by OpenAI, which is based on the GPT-3.5 architecture. It is designed to generate human-like responses to text-based prompts or questions.

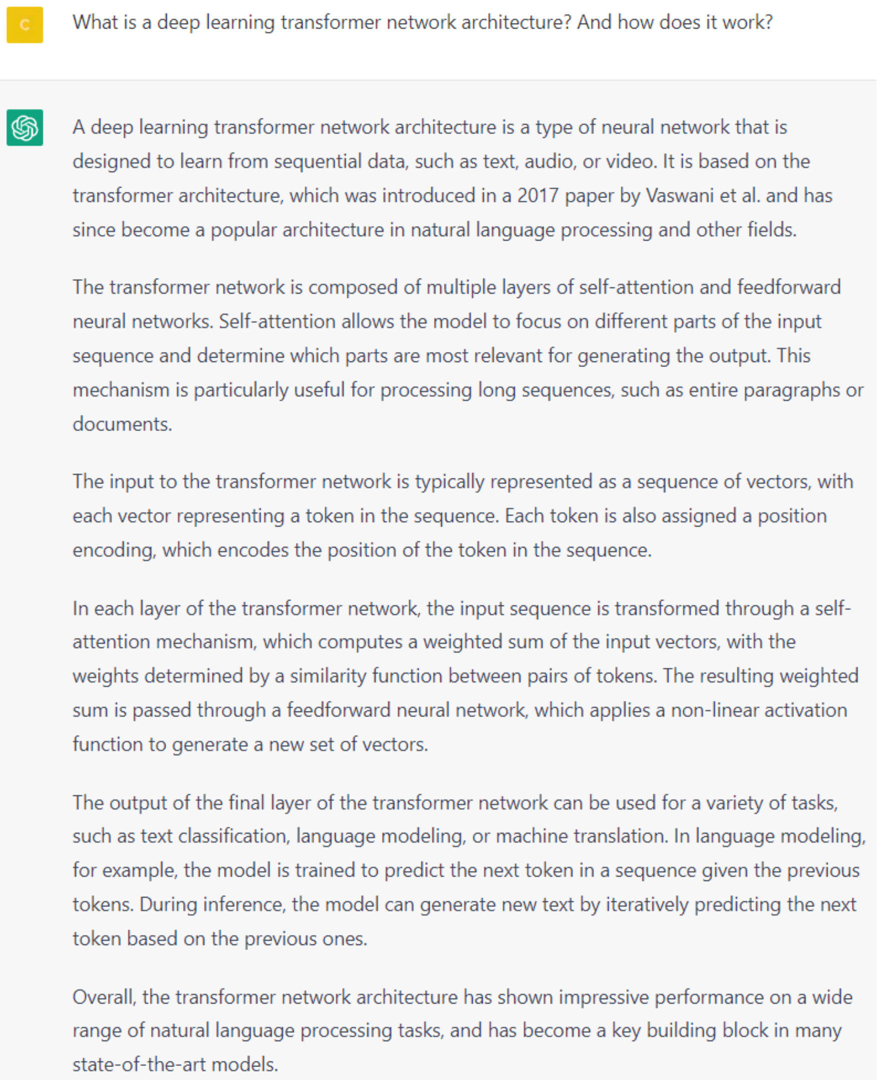
The GPT (Generative Pre-trained Transformer) architecture is a deep learning algorithm that uses a transformer network to learn from large amounts of text data. This allows the model to understand the structure and patterns of language and generate text that is coherent and meaningful.

ChatGPT has been trained on a massive corpus of text data, which includes web pages, books, and other sources of written language. The model has been trained to predict the next word in a sentence, given the previous words. This process is repeated millions of times, allowing the model to learn the patterns and structure of language.

When given a prompt or question, ChatGPT uses its learned knowledge to generate a response that is relevant to the input. The model can generate responses on a wide range of topics, including science, literature, history, and current events. Additionally, ChatGPT can understand and respond in multiple languages, including English, Spanish, French, German, and Chinese.

Overall, ChatGPT represents a significant breakthrough in natural language processing and has the potential to revolutionize the way we interact with machines.

DL transformer network architecture?



What is a deep learning transformer network architecture? And how does it work?

A deep learning transformer network architecture is a type of neural network that is designed to learn from sequential data, such as text, audio, or video. It is based on the transformer architecture, which was introduced in a 2017 paper by Vaswani et al. and has since become a popular architecture in natural language processing and other fields.

The transformer network is composed of multiple layers of self-attention and feedforward neural networks. Self-attention allows the model to focus on different parts of the input sequence and determine which parts are most relevant for generating the output. This mechanism is particularly useful for processing long sequences, such as entire paragraphs or documents.

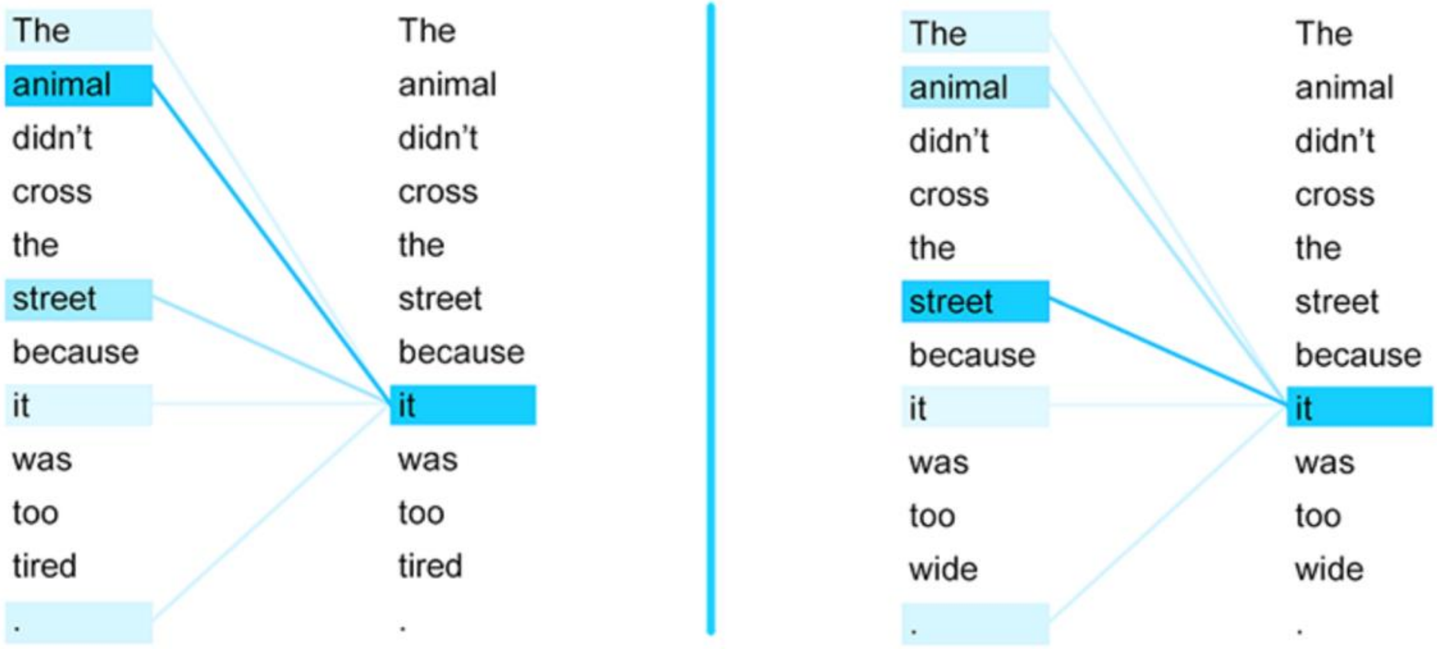
The input to the transformer network is typically represented as a sequence of vectors, with each vector representing a token in the sequence. Each token is also assigned a position encoding, which encodes the position of the token in the sequence.

In each layer of the transformer network, the input sequence is transformed through a self-attention mechanism, which computes a weighted sum of the input vectors, with the weights determined by a similarity function between pairs of tokens. The resulting weighted sum is passed through a feedforward neural network, which applies a non-linear activation function to generate a new set of vectors.

The output of the final layer of the transformer network can be used for a variety of tasks, such as text classification, language modeling, or machine translation. In language modeling, for example, the model is trained to predict the next token in a sequence given the previous tokens. During inference, the model can generate new text by iteratively predicting the next token based on the previous ones.

Overall, the transformer network architecture has shown impressive performance on a wide range of natural language processing tasks, and has become a key building block in many state-of-the-art models.

Transformers & “attention mechanisms”



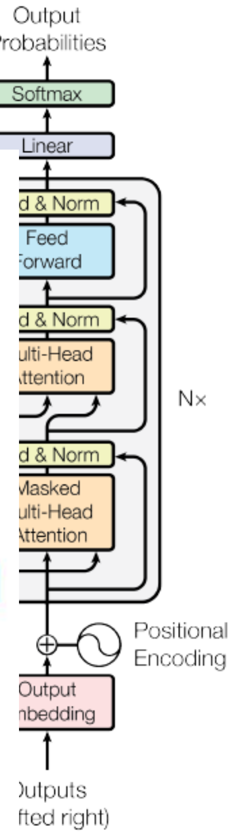


Figure 1: The Transformer - model architecture.

Vaswani et al., 2017, Attention Is All You Need
https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

DL application for image generation

Generative Adversarial Network (GAN) architecture to generate images

- Midjourney, https://www.midjourney.com/
- DALL-E from OpenAI
- « Stable Diffusion » from Stability AI

→ *Generate realistic images to "any textual descriptions"!*

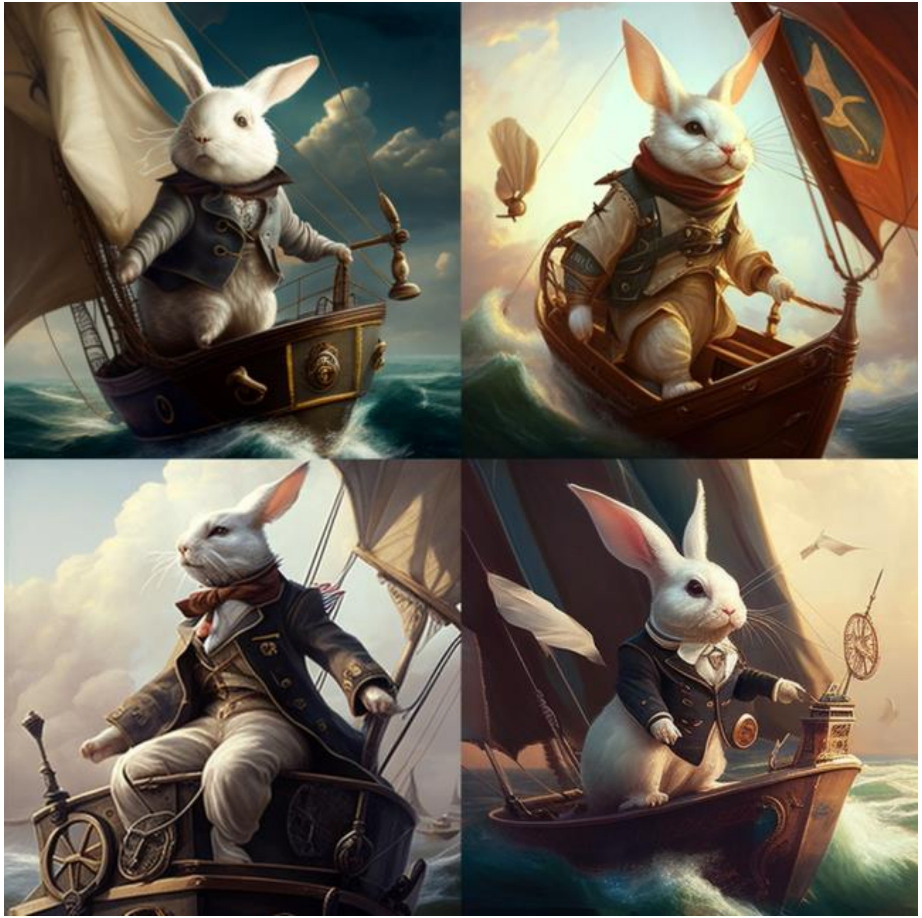
…no limit?

https://en.wikipedia.org/wiki/Midjourney
https://en.wikipedia.org/wiki/DALL-E
https://en.wikipedia.org/wiki/Stable_Diffusion

Test Midjourney

Test:

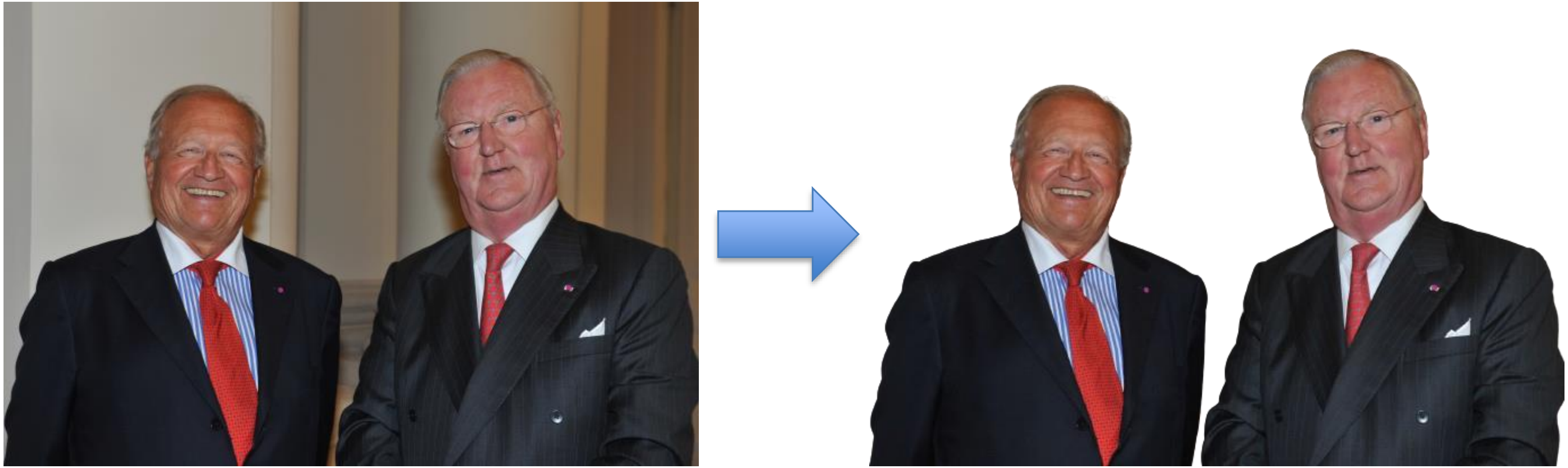
/imagine
a suited white rabbit at the helm of a flying sail ship



https://www.midjourney.com/

Image segmentation

For example, automatic background removal



https://www.remove.bg/

DL/ChatGPT for medicine?

Try « Glass AI », https://www.glass.ai/ :

…to draft a differential diagnosis or clinical plan based on a diagnostic problem representation. It is intended for the purpose of supporting or providing recommendations to a healthcare provider about the prevention, diagnosis or treatment of a disease or condition.

https://glass.health/ai

DL/ChatGPT for research ?

Try « Elicit», https://elicit.com/ :

…Analyze research papers at superhuman speed

Automate time-consuming research tasks like summarizing papers, extracting data, and synthesizing your findings..

https://glass.health/ai

New stuff… always!

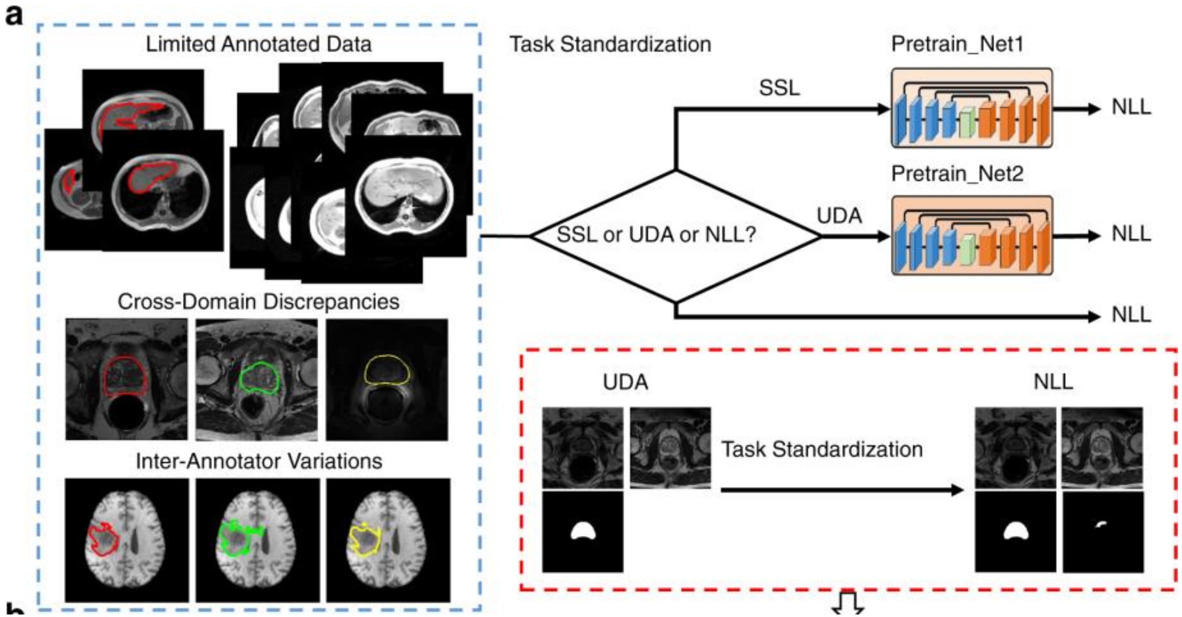
Liquid Neural Nets (LNNs, 2020), aka. "time-continuous recurrent neural network models"

> *…promises more compact and dynamic neural nets for* ***time series prediction****. LNNs offer a new approach to tasks like weather prediction, speech recognition, and autonomous driving. The primary benefit LNNs offer is that they* ***continue adapting to new stimuli after training****. Additionally, LNNs are* ***robust in noisy conditions*** *and are* ***smaller and more interpretable*** *than their conventional counterparts.*

https://news.mit.edu/2021/machine-learning-adapts-0128
https://medium.com/@hession520/liquid-neural-nets-lnns-32ce1bfb045a
https://doi.org/10.48550/arXiv.2006.04439

Medical imaging applications

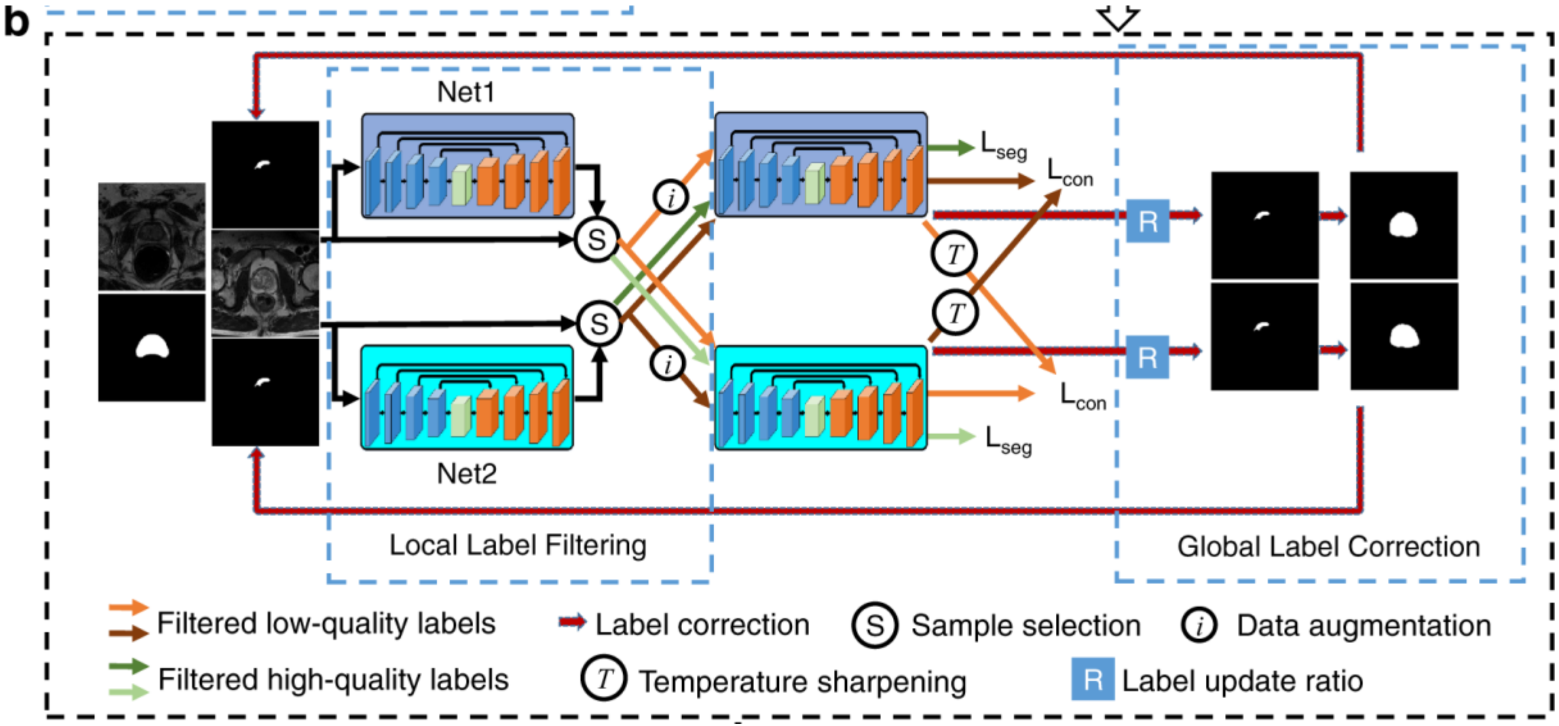
Wang et al., « Annotation-efficient deep learning for automatic medical image segmentation » , Nat. Com., 2021.



a The three challenges (semi-supervised learning (SSL), unsupervised domain adaptation (UDA), and noisy label learning (NLL)) that AIDE addresses and the proposed task standardization method

https://doi.org/10.1038/s41467-021-26216-9

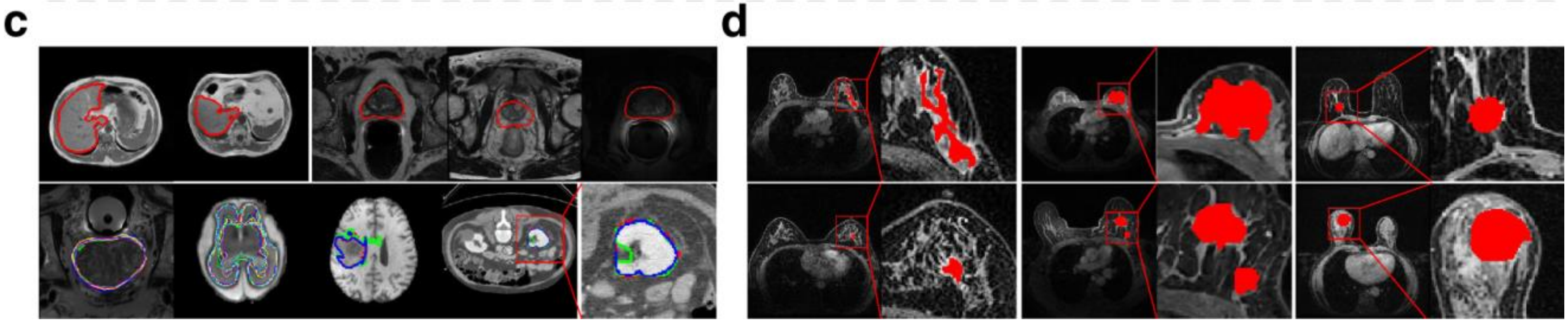
Medical imaging applications



b The overall framework of AIDE, which comprises three major elements: local label filtering, global label correction, and cross-model co-optimization.

https://doi.org/10.1038/s41467-021-26216-9

Medical imaging applications

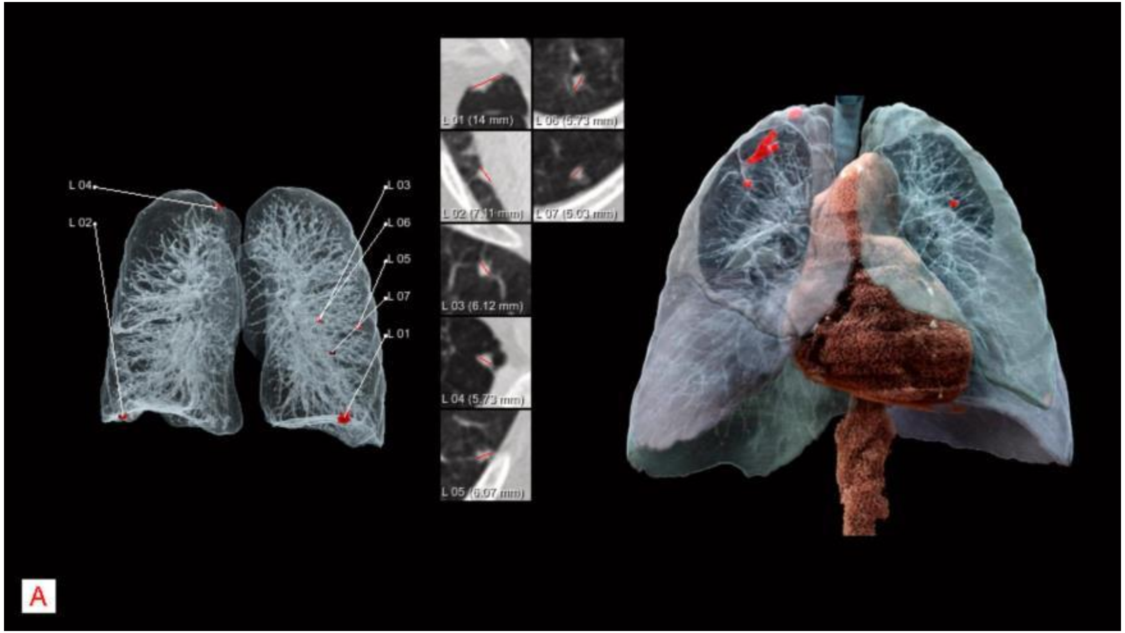


c Example images of the open datasets. The top left two images are from the CHAOS dataset. The top right three images are from the three domains of prostate datasets. The bottom images are from the QUBIQ datasets, where the first four correspond to the four subtasks and the last one is an enlarged view of the fourth image. Color lines indicate the target regions. **d** Example images of the breast datasets. From left to right, the three columns correspond to the images collected from the three medical centers. Red color regions show the breast tumors.

https://doi.org/10.1038/s41467-021-26216-9

Medical imaging applications

…and companies, e.g. Siemens.



Automated highlighting and quantification of certain anatomies and abnormalities within a chest CT exam.

…or Intel.

https://www.siemens-healthineers.com/medical-imaging/digital-transformation-of-radiology/ai-in-radiology
https://www.intel.com/content/www/us/en/customer-spotlight/stories/siemens-healthineers-customer-story.html

Outline

- Introduction
- Deep learning concepts
- Clustering
- Linear predictive models
- Conclusions

Clustering

- Automatic partitioning of data samples into “clusters”
- Samples in a cluster more similar than across clusters
- **No single solution but many algorithms!**

For example

- Hierarchical clustering
- k-means clustering
- Distribution-based clustering

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

Hierarchical clustering

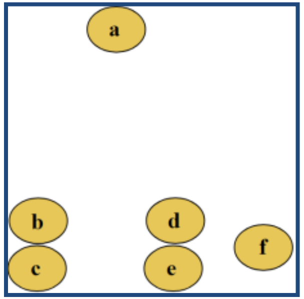
- Agglomerative or divisive approach
  - "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
  - "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.
- Relies on
  - “cluster dissimilarity” = distance between pairs of observations
  - “linkage criterion” = the dissimilarity of sets as a function of the pairwise distances of observations in the sets
- Usually presented in a “dendrogram”

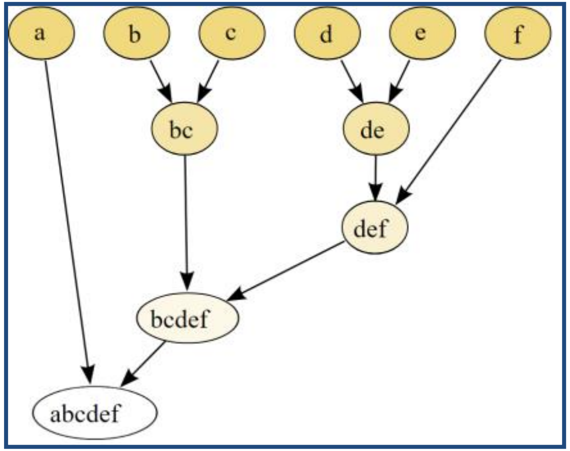
https://en.wikipedia.org/wiki/Hierarchical_clustering

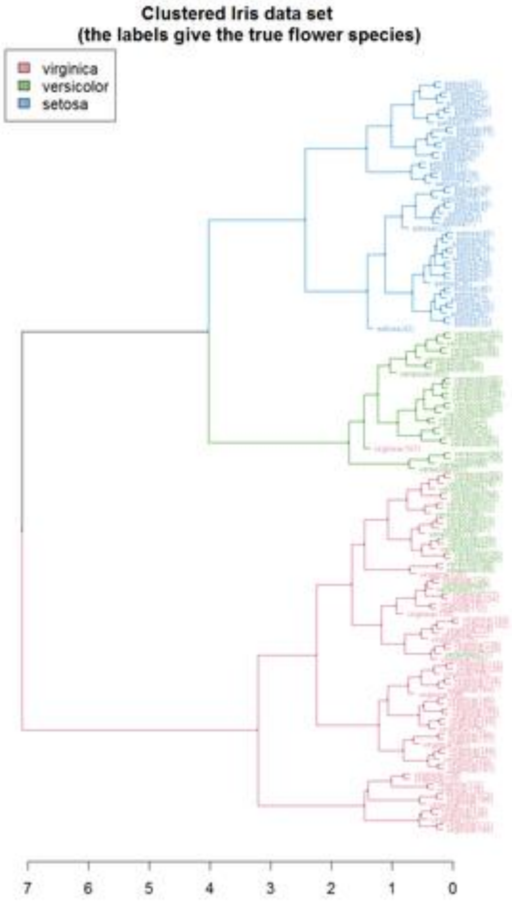
Hierarchical clustering

Raw data to cluster:

- 2 features
- Euclidian distance metric
- Agglomerative approach






Clustered Iris data set
(the labels give the true flower species)

Question: How many clusters?

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

k-means clustering

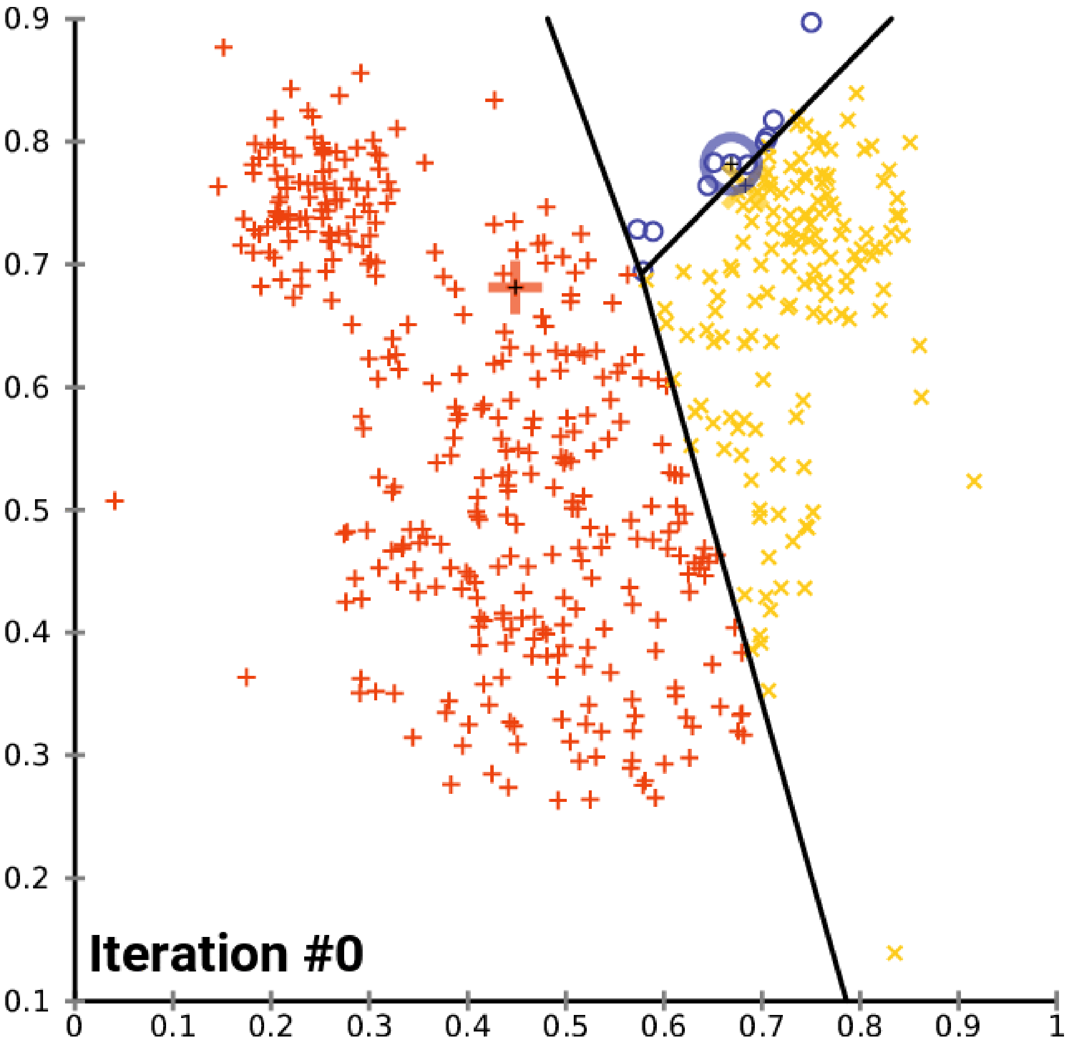
- Partition `n` observations into `k` clusters
- Each observation belongs to the cluster with the nearest "mean" (cluster centers or cluster centroid)
- Iteratively alternates between
  - clustering observations
  - updating clusters' mean

https://en.wikipedia.org/wiki/K-means_clustering

k-means clustering

Simple example

- 2D observations
- A priori 3 clusters
- Random initialization



https://en.wikipedia.org/wiki/K-means_clustering

k-means clustering

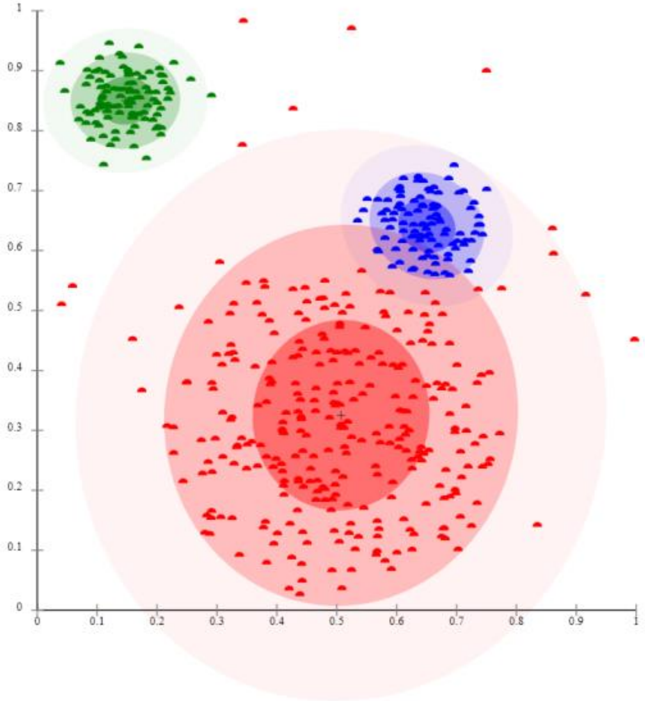
- Partition `n` observations into `k` clusters
- Each observation belongs to the cluster with the nearest "mean" (cluster centers or cluster centroid)
- Iteratively alternates between
  - clustering observations
  - updating clusters' mean

- Tends to find clusters of comparable spatial extent
- **How many clusters? → fixed *a priori*…**

https://en.wikipedia.org/wiki/K-means_clustering

Distribution-based clustering

- Clusters defined as objects belonging most likely to the same distribution.
- Iteratively alternates between
  - clustering observations
  - updating clusters' parameters
- Example "Gaussian mixture models" (GMMs)
- K-means vs GMM



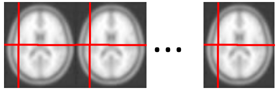
https://en.wikipedia.org/wiki/Cluster_analysis

Outline

- Introduction
- Deep learning concepts
- Clustering
- Linear predictive models:
  - Classification and regression
  - Regularization & kernel methods
  - Validation & inference
- Conclusions

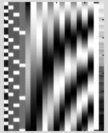
Mass-univariate vs Pattern recognition in NI

Standard Statistical Analysis (mass-univariate)



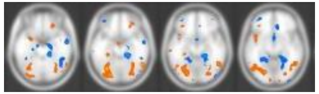
Patients vs controls

Voxel-wise GLM model estimation



Independent statistical test at each voxel

Correction for multiple comparisons

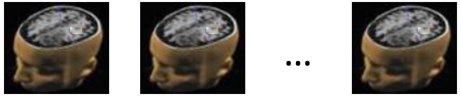


Statistical Parametric Map

Very different meaning!

Pattern Recognition Analysis (multivariate & predictive)

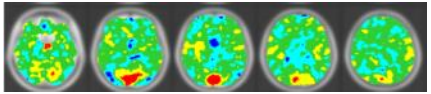
Volumes from group 1



Volumes from group 2

New example

Training Phase



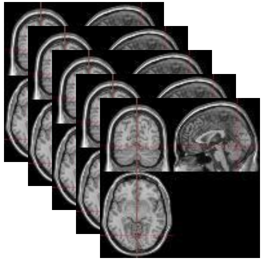
Linear model predictive map
(classification or regression weights)

Testing Phase

Predictions:
group 1 or group 2

Classification model in NI

Class 1



Label = patient

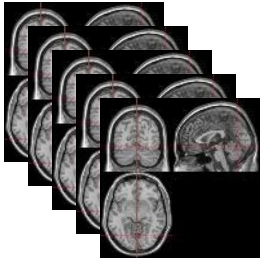
Label = patient

Label = patient

Label = patient

Label = patient

Class 2



Label = control

Label = control

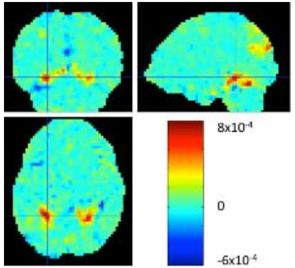
Label = control

Label = control

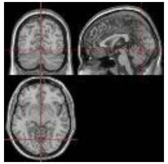
Label = control

Training

Predictive function: *f*



New subject



Testing

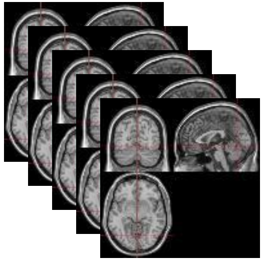
Prediction:

Class membership

(patient/control)

Regression model in NI

Class 1



Score = 11

Score = 8

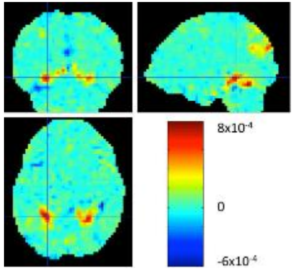
Score = 22

Score = 17

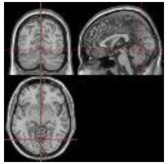
Score = 30

Training

Predictive function: f



New subject



Testing

Prediction:
Score = 28

Advantages of Pattern Recognition

- **Multivariate analysis**: It can be more sensitive to detect spatially distributed effects.

  …but no local inferences.

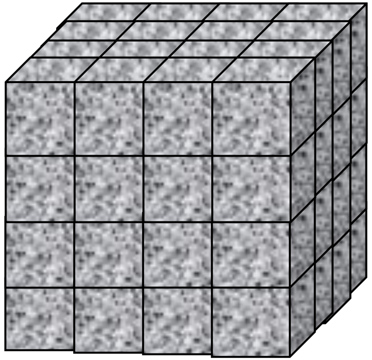
- **Predictive framework**: Provides predictions for new examples (e.g. new subjects/images).

  …but it typically requires more data!

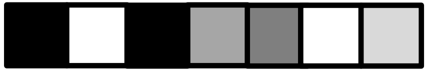
Extracting features from brain data

Whole brain volume

3D matrix of voxels



Feature vector



Data dimensionality
= number of voxels

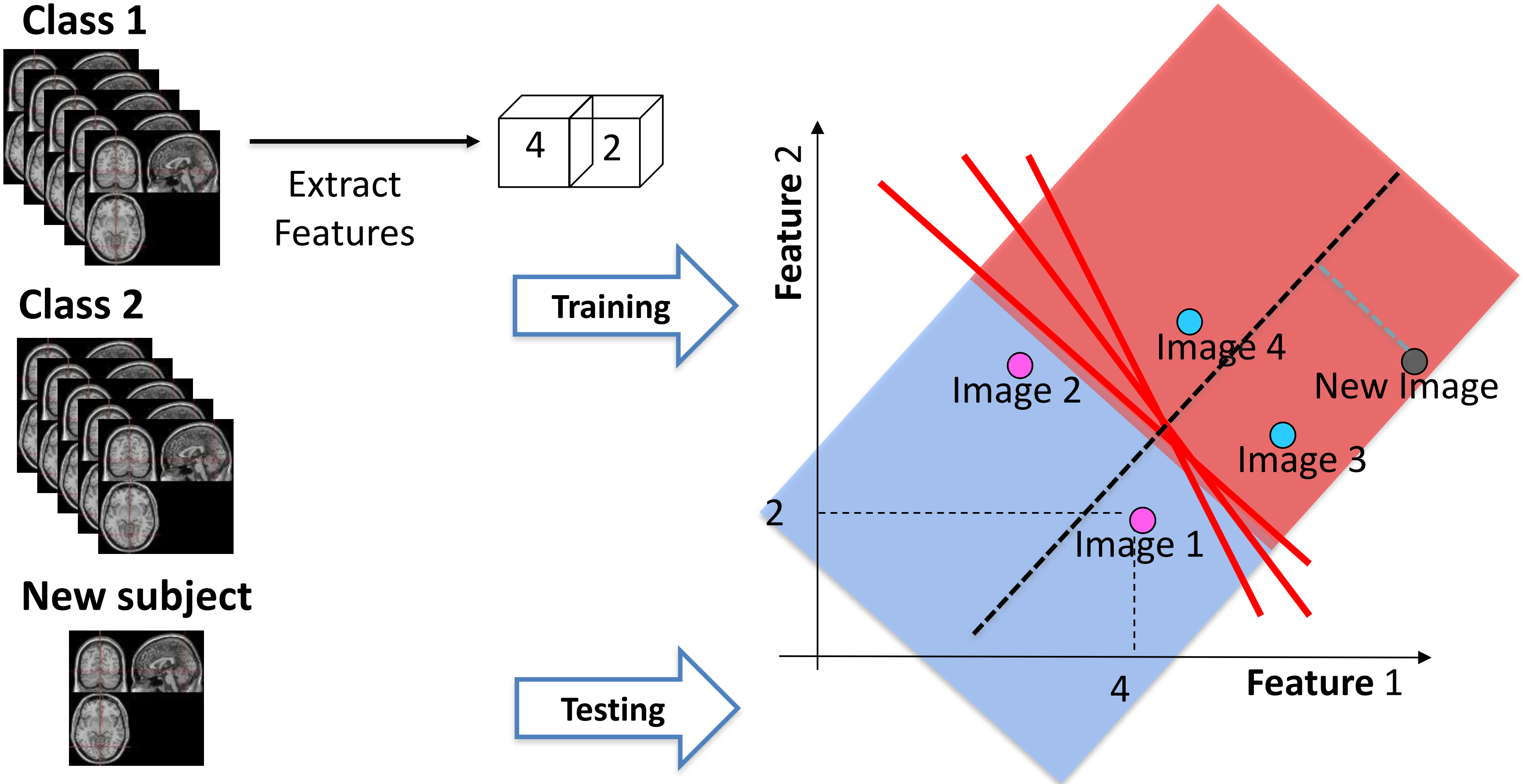
Feature meaning depends on the data type:

- BOLD signal activation
- local GM volume
- radiotracer concentration
- EEG time-frequency pattern
- …

Other type of features:

- Volumes of regions of interest (ROIs)
- Connectivity measures
- Cortical thickness
- …

Classification model



Class 1

Extract Features

Class 2

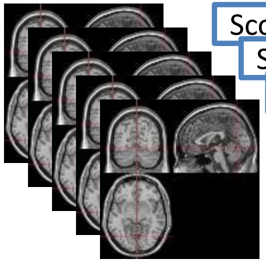
New subject

Training

Testing

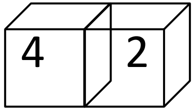
Regression model

Class 1



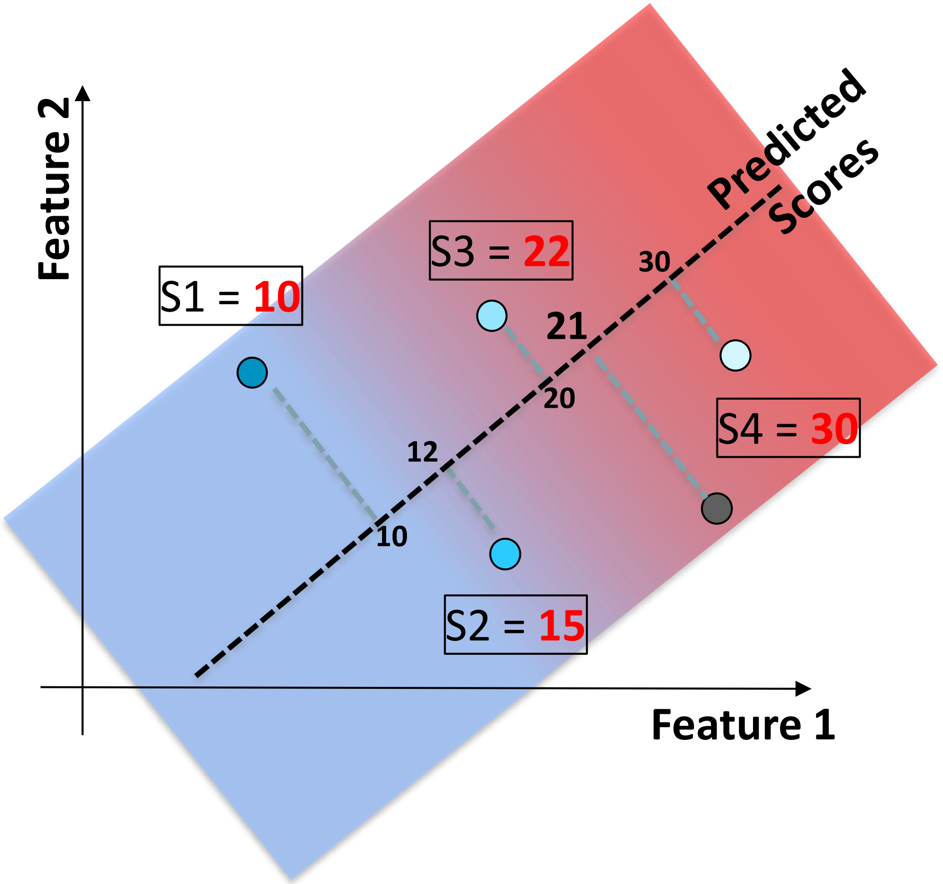
Score = 10
Score = 15
Score = 22
Score = 30

Extract Features

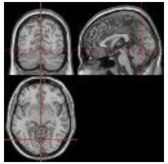


Training

Testing



New subject



Linear predictive models

- Linear predictive models (classifier or regression) are parameterized by a weight vector $\boldsymbol{w}$ and a bias term b.

- The general equation for making predictions for a test example $\boldsymbol{x}_*$ is:

Parameters learned/estimated from training data

$$f(\boldsymbol{x}_*) = \boldsymbol{w} \cdot \boldsymbol{x}_* + b$$

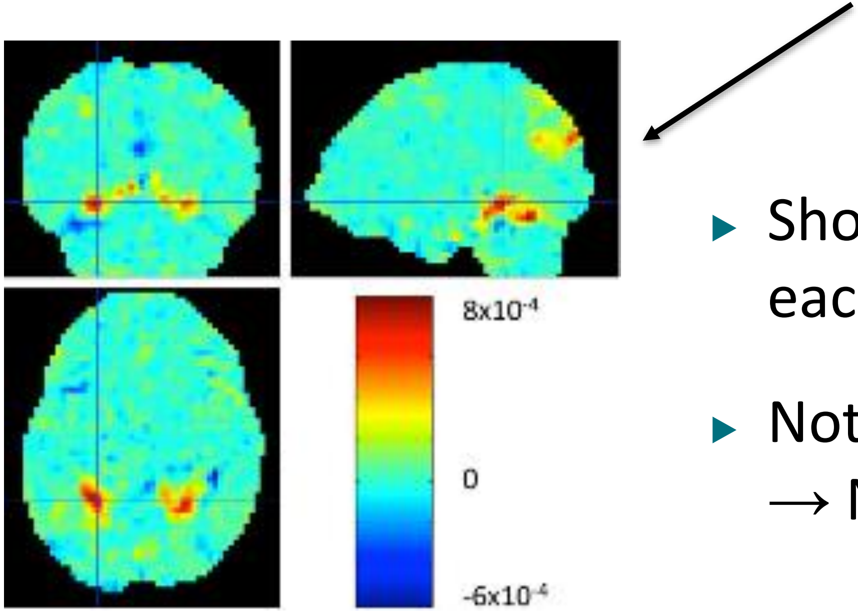
- In the linear case $\boldsymbol{w}$ can be expressed as a linear combination of training examples $\boldsymbol{x}_{\boldsymbol{i}}$ (N = number of training examples)

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i$$

Weight maps or predictive patterns

Linear prediction model:

$$f(\boldsymbol{x}_*) = \boldsymbol{w} \cdot \boldsymbol{x}_* + b$$



- Shows the relative contribution of each feature for the decision
- Not a statistical map
  → No local inferences can be made!

Outline

- Introduction
- Deep learning concepts
- Clustering
- Linear predictive models:
  - Classification and regression
  - Regularization & kernel methods
  - Validation & inference
- Conclusions

Pattern recognition in neuroimaging

Common issue with neuroimaging applications:

#features (e.g. voxels) $\gg$ **#samples (e.g. subjects)**

$\Rightarrow$ ill-conditioned problems!

Possible solutions:

- Decrease the number of features
  - *a priori* “Region of Interest” (ROI)
  - feature selection strategies **(DANGER of double dipping!)**
  - “searchlight”, i.e. shift all around a small ROI
- Regularization + Kernel Methods

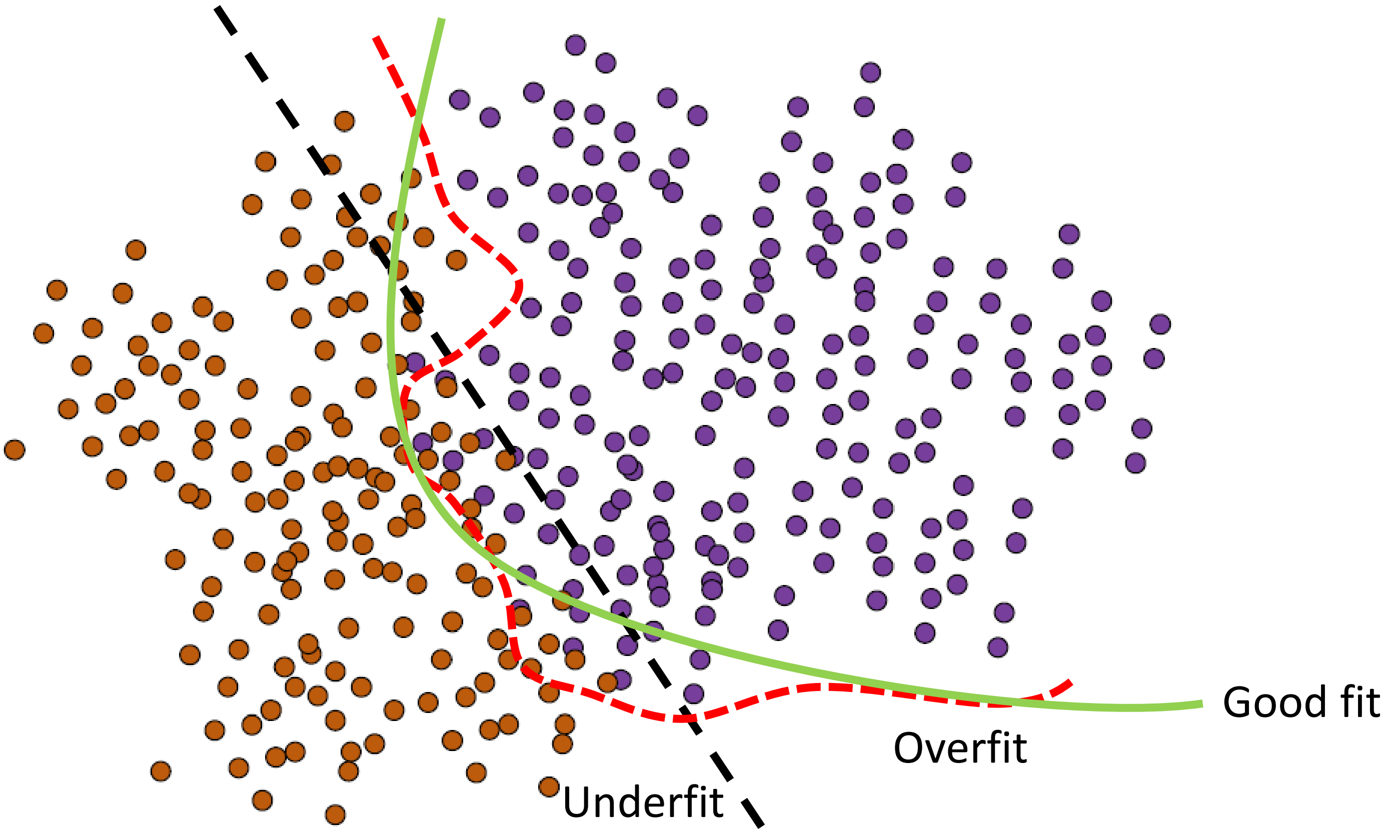
Regularization

- To find a unique solution & avoid overfitting
- Balance between data-fit L & penalty J terms

  $\min_{\boldsymbol{w} \in \mathbb{R}^p} \{L(\boldsymbol{w}) + \lambda\, J(\boldsymbol{w})\}$
- Different choices of L and J lead to different solutions!

Baldassarre et al.(2017)

Overfitting & model complexity



Regularization

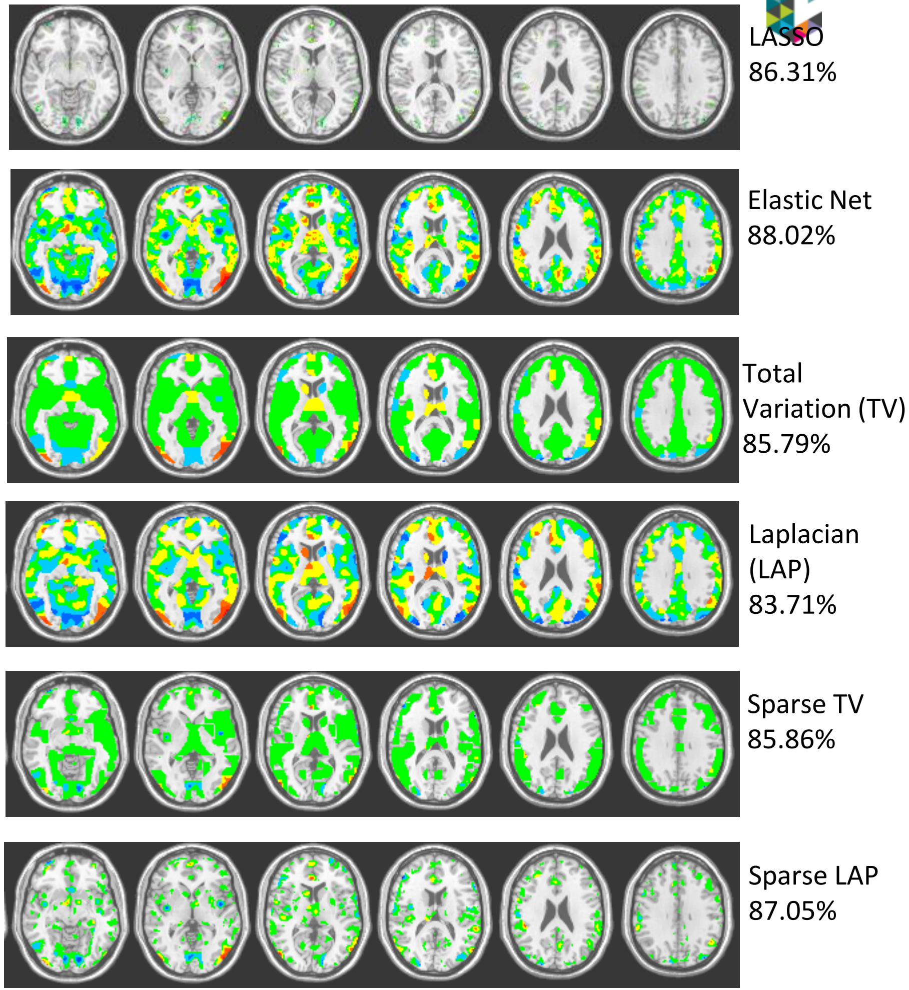
- To find a unique solution & avoid overfitting
- Balance between data-fit L & penalty J terms

$$\min_{\boldsymbol{w}\in\mathbb{R}^p}\{L(\boldsymbol{w}) + \lambda\, J(\boldsymbol{w})\}$$

- Different choices of L and J lead to different solutions!

Example:

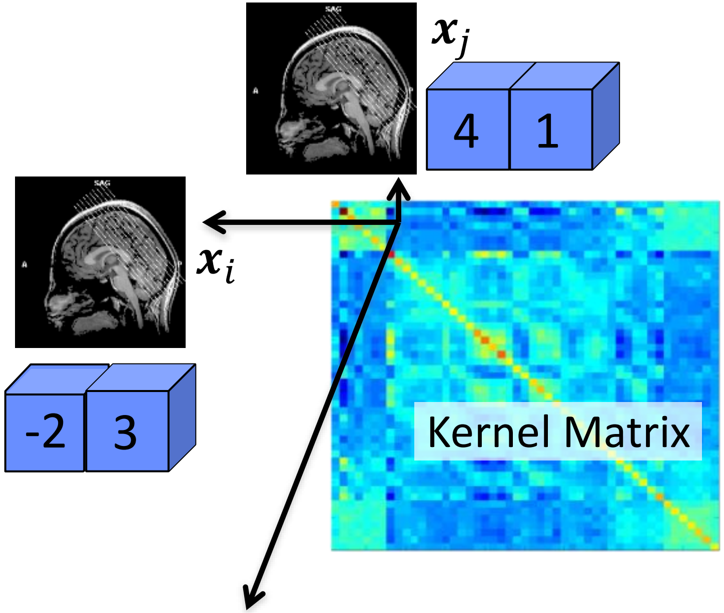
- square loss L + different J
- about the same accuracy!



Baldassarre et al.(2017)

Kernel Methods

- General framework for classification & regression models
- Relies on 2 parts
  - kernel function $k_{ij} = \mathrm{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$
  - algorithm relying on kernel formalism
- Advantages
  - general approach for regularization
  - computational efficiency
  - “kernel trick” (linear & non-linear kernels) to measure “sample similarity”



Linear kernel ≡ dot product

$$k_{ij} = \boldsymbol{x}_i \cdot \boldsymbol{x}_j = (4 * -2) + (1 * 3) = -5$$

Kernel methods & Multi-kernel learning

- Linear kernel approach:

$f(\boldsymbol{x}_*) = \boldsymbol{w} \cdot \boldsymbol{x}_* + b$ where $\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i$

→ $f(\boldsymbol{x}_*) = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i \cdot \boldsymbol{x}_* + b$

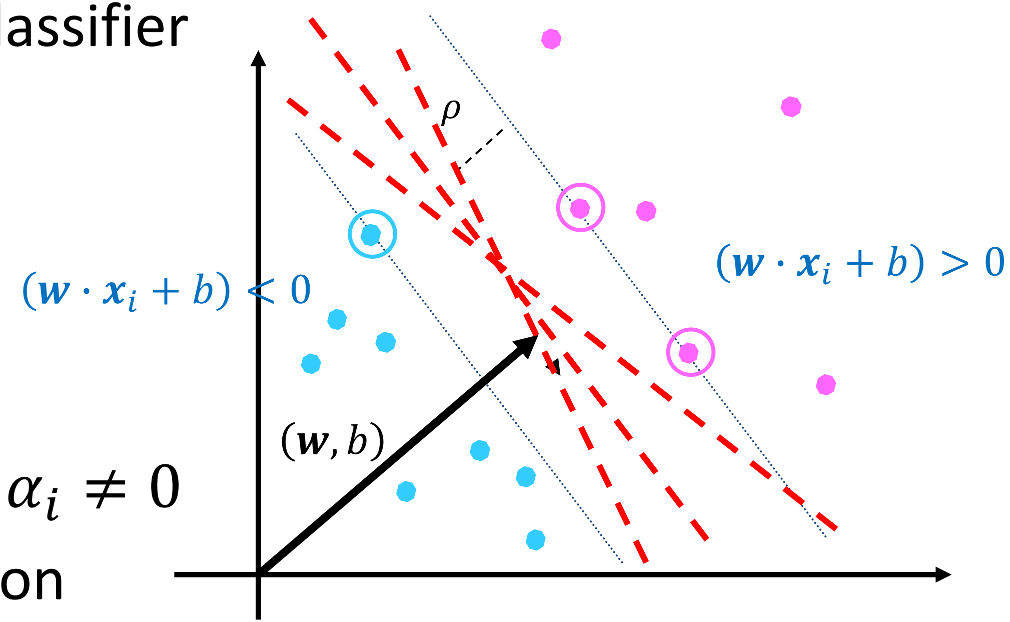
→ $f(\boldsymbol{x}_*) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_*) + b$ with $k_{ij} = \mathrm{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$

- Example of kernel methods:

Support Vector Machines (SVM), Kernel Ridge Regression (KRR), Gaussian Process (GP), Kernel Fisher Discriminant, Relevance Vector Regression,…

Support Vector Machine

- Relies on kernel representation
- "maximum margin" ρ classifier


ρ
$(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) < 0$
$(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) > 0$
$(\boldsymbol{w}, b)$

- "Support vectors" have $\alpha_i \neq 0$
- Fast & resilient estimation
- …but only "hard binary" prediction!

Kernel methods & Multi-kernel learning

- Linear kernel approach:

$f(\boldsymbol{x}_*) = \boldsymbol{w} \cdot \boldsymbol{x}_* + b$ where $\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i$

→ $f(\boldsymbol{x}_*) = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i \cdot \boldsymbol{x}_* + b$

→ $f(\boldsymbol{x}_*) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_*) + b$ with $k_{ij} = \mathrm{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$

- Example of kernel methods:

SVM, KRR, GP, RVR,…

- “Multi-kernel learning” ≡ combine M sub-kernels

$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{M} d_m \, K_m(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with $d_m \geq 0$ and $\sum_{m=1}^{M} d_m = 1$

then learn kernel weight d_m and decision function $(\boldsymbol{w}, b)$.

Kernel methods & Multi-kernel learning

- "Multiple-kernel learning" (MKL) can be applied to combine different sources of information for prediction, like
  - multimodal imaging, e.g. multiple fMRI contrasts, different quantitative maps, etc.
  - a set of ROIs, e.g. derived from an anatomical/functional atlas
  - both multimodal imaging and set of ROIs!

$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{M} d_m K_m(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with $d_m \geq 0$ and $\sum_{m=1}^{M} d_m = 1$

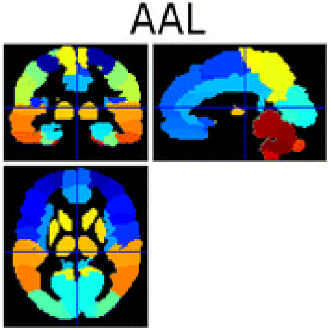
- MKL models simultaneously learn the kernel weights d_m and the associated decision function $(\boldsymbol{w}, b)$ in supervised learning settings, with
  - L1-norm on $d_m \longrightarrow$ sparse solution for the set of kernels
  - L2-norm on features $\longrightarrow$ smooth weight maps $\boldsymbol{w}$

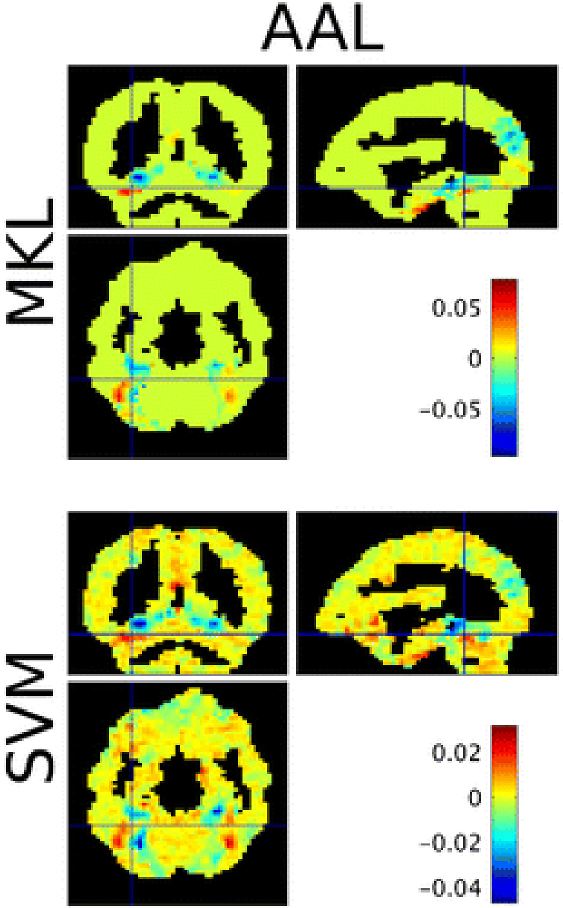
Schrouff et al. (2018)

Kernel methods & Multi-kernel learning

Example:

- Single subject fMRI, visual stimulation
- "Faces vs. Houses" classification
- AAL atlas





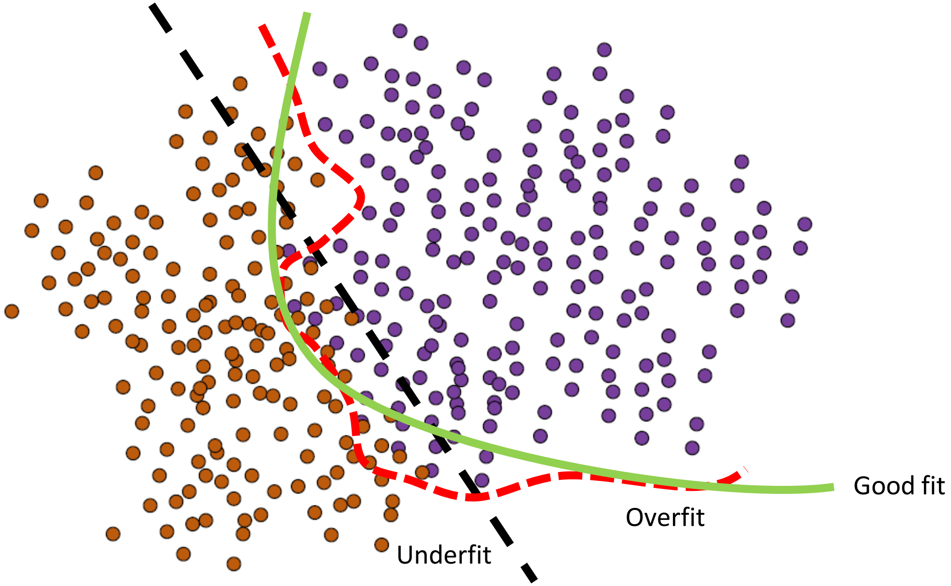
Schrouff et al. (2018)

Outline

- Introduction
- Deep learning concepts
- Clustering
- Linear predictive models:
  - Classification and regression
  - Regularization & kernel methods
  - Validation & inference
- Conclusions

Validation principle

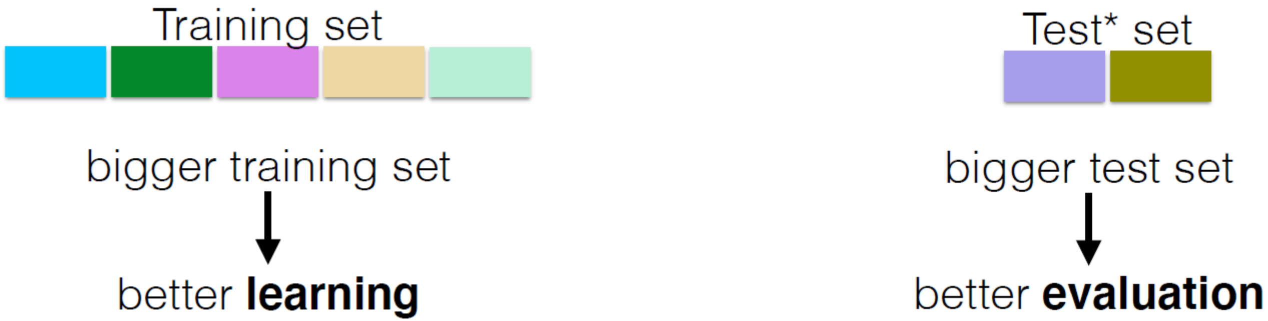
- How accurate is my prediction?
- Does this generalizes to *new* samples?



Accuracy and generalizability!

For the specified model… with *appropriate* complexity!

Validation principle



Training set

bigger training set

better **learning**

Test* set

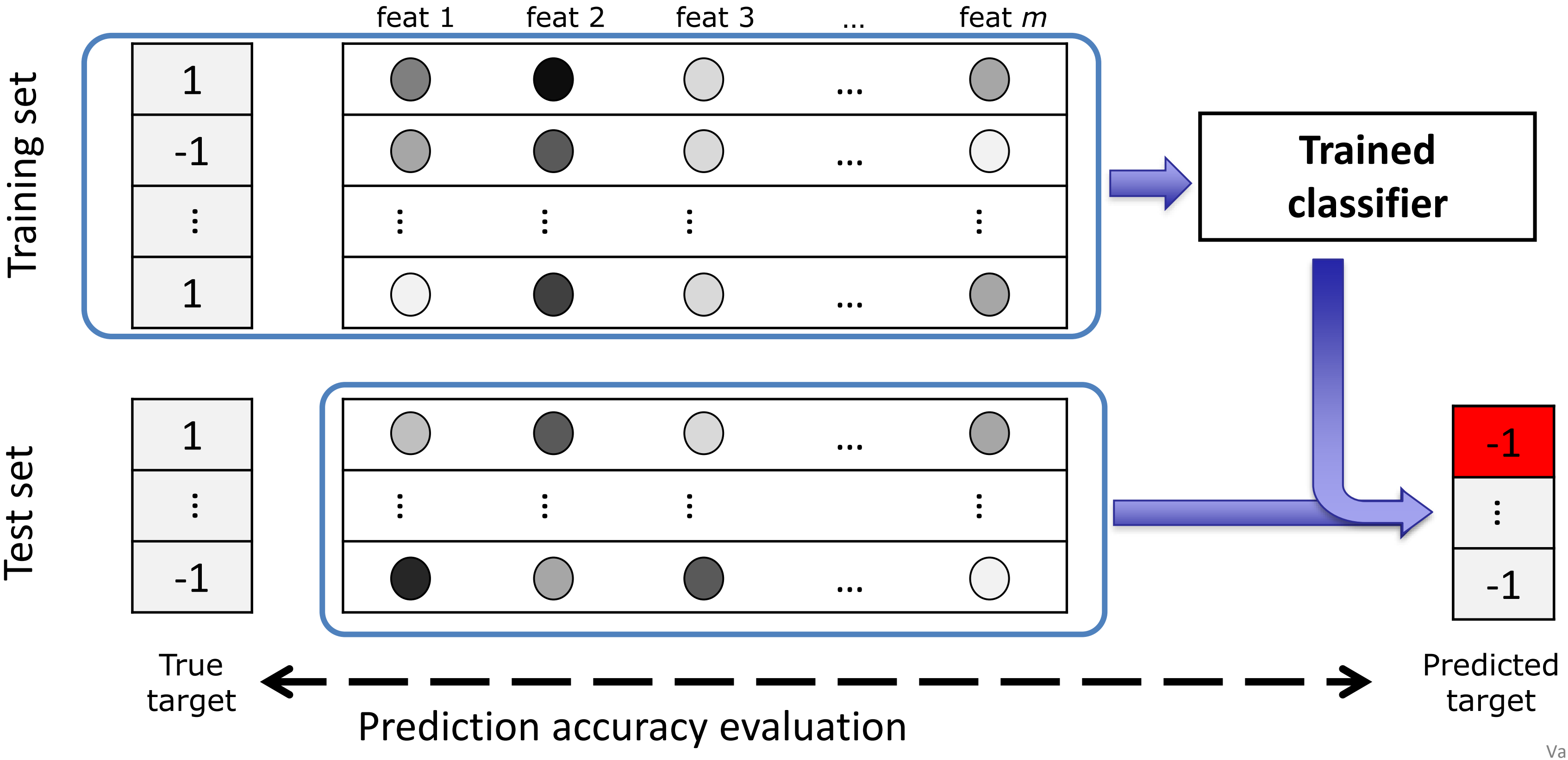
bigger test set

better **evaluation**

- Key: Train & test sets must be disjoint.
  - And the dataset or sample size is fixed.
  - They grow at the expense of each other!

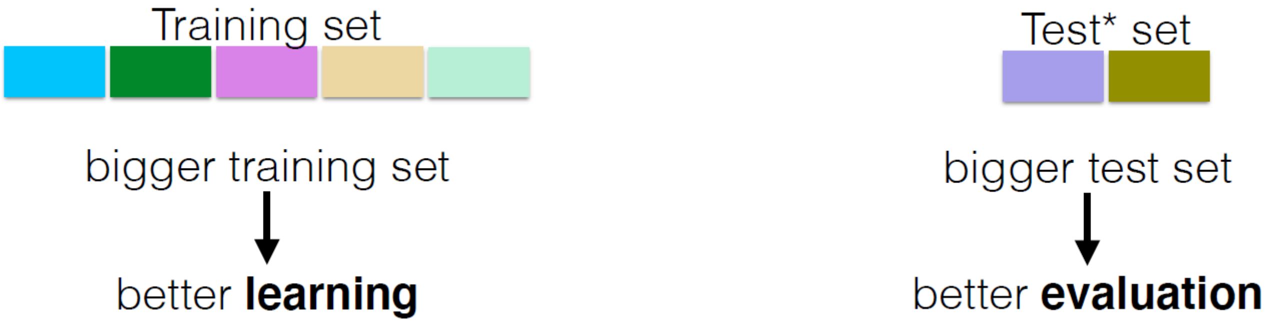
Validation principle

→ Out of sample prediction!



Varoquaux (2018)

Validation principle



Training set

bigger training set

↓

better **learning**

Test* set

bigger test set

↓

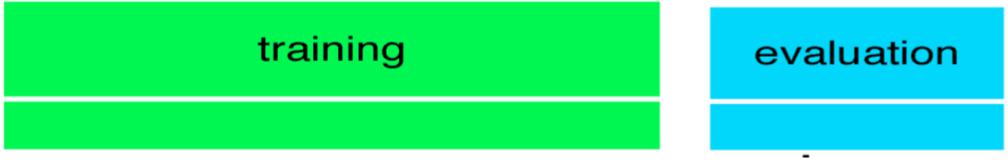
better **evaluation**

- Key: Train & test sets must be disjoint.
  - And the dataset or sample size is fixed.
  - They grow at the expense of each other!
- Cross-validation for “small” data sets

M-fold cross-validation

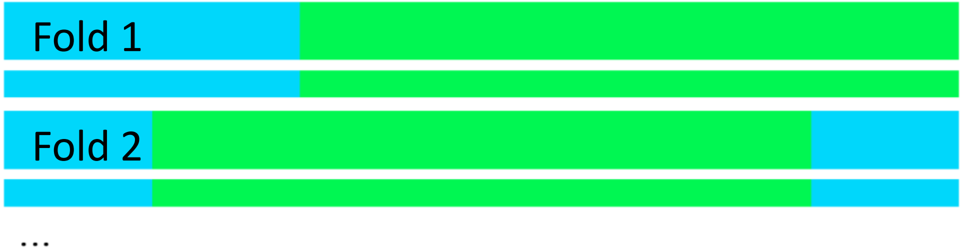
- Split data in 2 sets: “train” & “test”

  ➔ evaluation on 1 “fold”



- Rotate partition and repeat

  ➔ evaluations on M “folds”



- Applies to scans/events/blocks/subjects/…

  ➔ “Leave-some-out” (LSO) approach, **beware of dependences!**

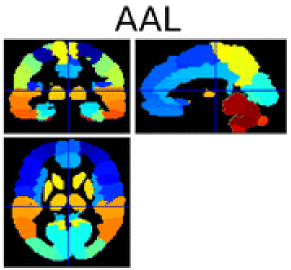
- Accumulates metric over the M “folds”.

Varoquaux (2018)

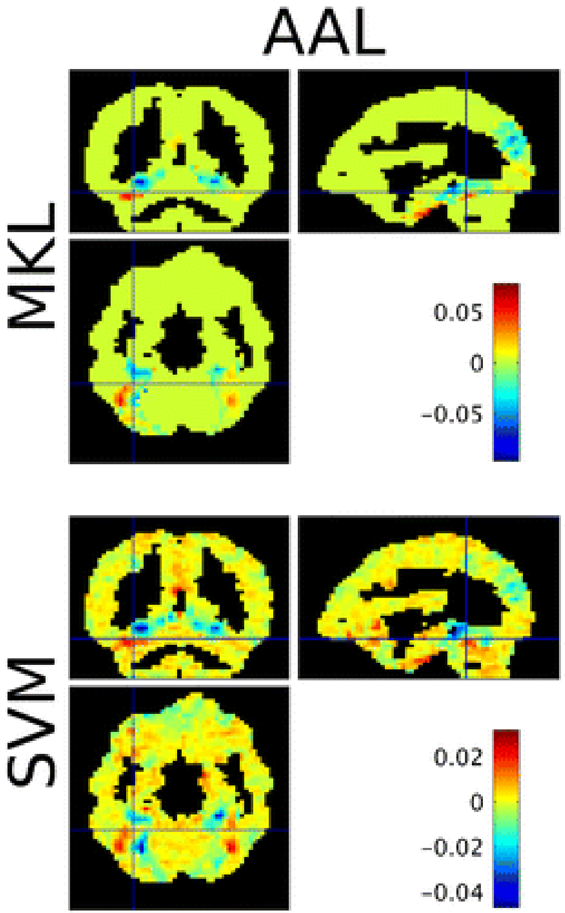
Kernel methods & Multi-kernel learning

Example:

- Single subject fMRI, visual stimulation
- “Faces vs. Houses” classification
- AAL atlas



- Balanced accuracy
  - MKL 98,15%
  - SVM 93,06%



Schrouff et al. (2018)

Prediction assessment

Classification → confusion matrix

- *Accuracy: total, class specific, or balanced*
  - $A_{tot} = \frac{TP+TN}{TP+FN+FP+TN}$,
  - $A_{c1} = \frac{TP}{TP+FN}$ & $A_{c0} = \frac{TN}{FP+TN}$
  - $A_{bal} = \frac{A_{c1}+A_{c0}}{2}$ (useful for unbalanced dataset)
- Sensitivity $S_e = A_{c1}$ & specificity $S_p = A_{c0}$
- Positive/negative predictive value

  $PPV = \frac{TP}{TP+FP}$ and $NPV = \frac{TN}{TN+FN}$

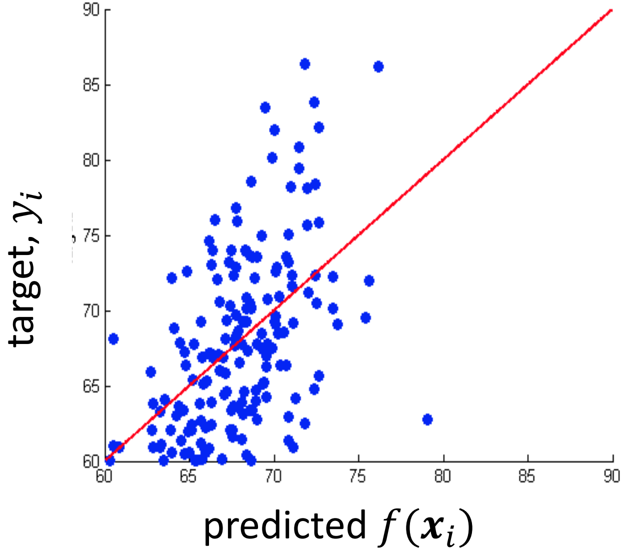
		Predicted class	
		1	0
Actual class	1	TP	FN
	0	FP	TN

Prediction assessment

Regression ⟶ mean squared error

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\boldsymbol{x}_i))^2$$

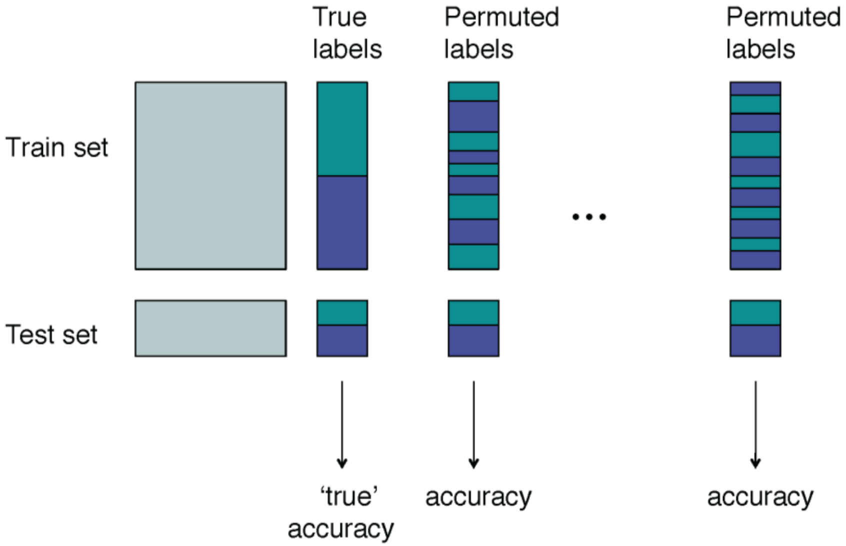
(or correlation between true & predicted scores)



Model inference

- H_0: “no link between features and target”
- Test statistic, e.g. cross-validation (CV) accuracy A
- Estimate distribution of test statistic under H_0
  - → Random permutation of labels
  - → Estimate CV accuracy, A_m
  - → Repeat M times
- Calculate p-value p as

$$p = \frac{1}{M} \sum_{m=1}^{M} \#(A_m \geq A)$$



Outline

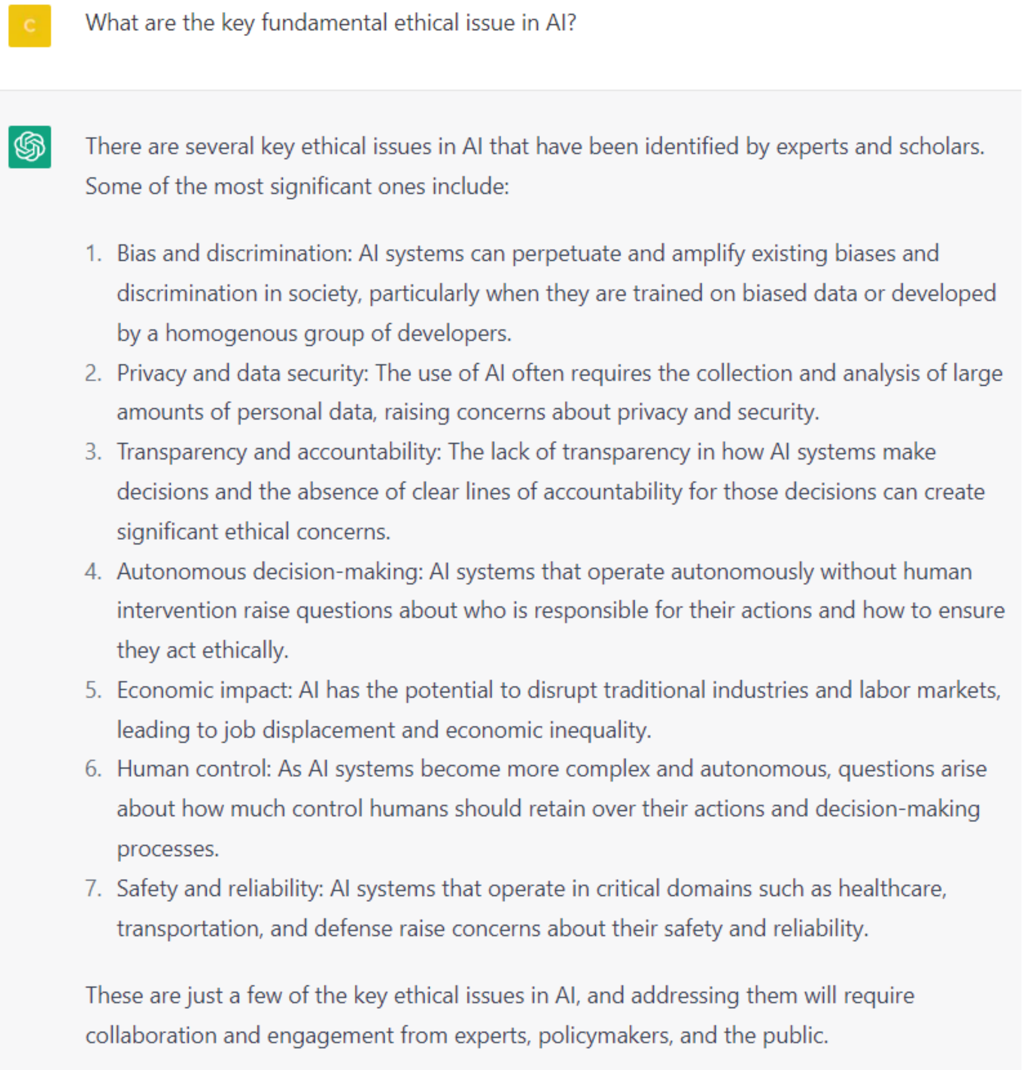
- Introduction
- Deep learning concepts
- Clustering
- Linear predictive models:
  - Classification and regression
  - Regularization & kernel methods
  - Validation & inference
- Conclusions

Outline

- Introduction
- Deep learning concepts
- Clustering
- Linear predictive models
- Conclusions

Conclusions

Ethical issues in AI…



C

What are the key fundamental ethical issue in AI?

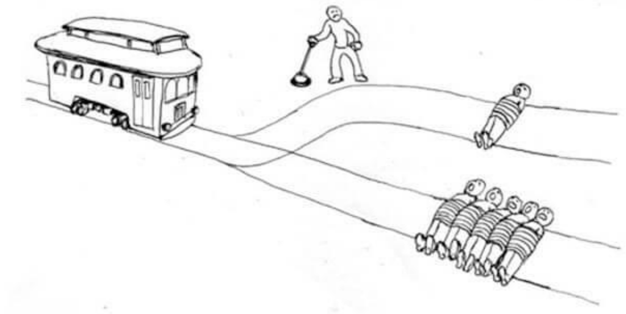
There are several key ethical issues in AI that have been identified by experts and scholars. Some of the most significant ones include:

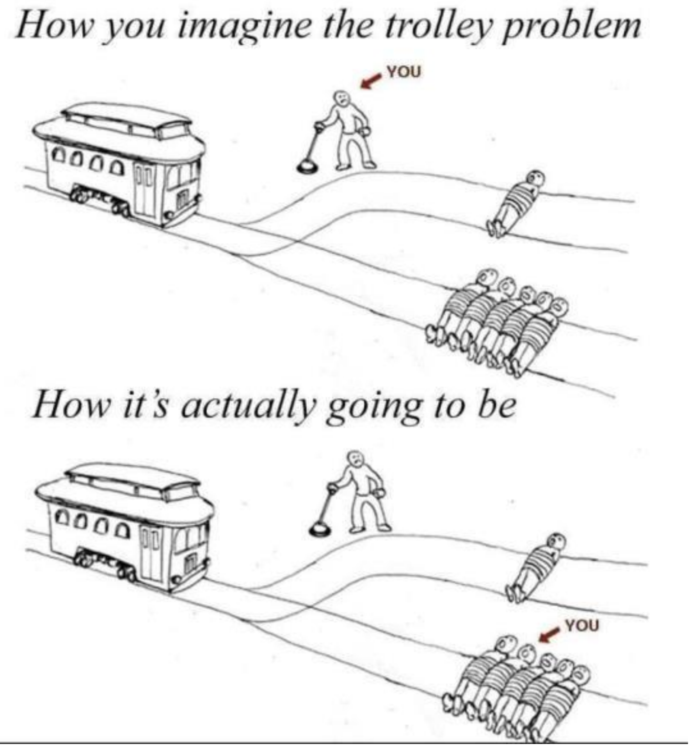
1. Bias and discrimination: AI systems can perpetuate and amplify existing biases and discrimination in society, particularly when they are trained on biased data or developed by a homogenous group of developers.
2. Privacy and data security: The use of AI often requires the collection and analysis of large amounts of personal data, raising concerns about privacy and security.
3. Transparency and accountability: The lack of transparency in how AI systems make decisions and the absence of clear lines of accountability for those decisions can create significant ethical concerns.
4. Autonomous decision-making: AI systems that operate autonomously without human intervention raise questions about who is responsible for their actions and how to ensure they act ethically.
5. Economic impact: AI has the potential to disrupt traditional industries and labor markets, leading to job displacement and economic inequality.
6. Human control: As AI systems become more complex and autonomous, questions arise about how much control humans should retain over their actions and decision-making processes.
7. Safety and reliability: AI systems that operate in critical domains such as healthcare, transportation, and defense raise concerns about their safety and reliability.

These are just a few of the key ethical issues in AI, and addressing them will require collaboration and engagement from experts, policymakers, and the public.

The « trolley problem »

"The ethical dilemma where an onlooker can save five lives from a rogue trolley by diverting it to kill just one person – illustrates why making decisions about who lives and dies are inherently moral judgments so they can't be relegated to unfeeling machines"




How you imagine the trolley problem
YOU
How it's actually going to be
YOU

Global references

- Books:
  - Hastie , Tibishirani, Friedman (2003) Elements of Statistical Learning. Springer
  - Shawe-Taylor and Christianini (2004) Kernel Methods for Pattern Analysis. Cambridge: Cambridge University Press.
  - Bishop, Jordan, Kleinberg, Schölkopf (2006) Pattern Recognition and Machine learning. Springer
- Machines:
  - Burges (1998) A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2:121–167.
  - Rasmussen, Williams (2006) Gaussian Processes for Machine Learning. The MIT Press.
  - Tipping (2001) Sparse Bayesian Learning and the Relevance Vector Machine Journal of Machine Learning Research, 1, 211-244
  - Breiman (1996) Bagging Predictors Machine Learning, 24, 123-140

Specific references

- Reviews:
  - Haynes and Rees (2006) Decoding mental states from brain activity in humans. Nat. Rev. Neurosci., 7, 523-534
  - Pereira, Mitchell, Botnivik (2009) Machine learning classifiers and fMRI: a tutorial overview. Neuroimage,45, S199-S209
- ML and NI:
  - M.-A. Schulz et al. (2020) Different scaling of linear models and deep learning in UKBiobank brain images versus machine-learning datasets. Nat. Com., 11,4238. https://doi.org/10.1038/s41467-020-18037-z
  - L. Baldasare (2017) Sparsity Is Better with Stability: Combining Accuracy and Stability for Model Selection in Brain Decoding. Frontiers in Neuroscience, 11, 62. https://doi.org/10.3389/fnins.2017.00062
  - G. Varoquaux (2018) Cross-validation failure: Small sample sizes lead to large error bars. Neuroimage, 180, 68-77. https://doi.org/10.1016/j.neuroimage.2017.06.061
  - J. Schrouff et al. (2018) Embedding Anatomical or Functional Knowledge in Whole-Brain Multiple Kernel Learning Models. Neuroinformatics. https://doi.org/10.1007/s12021-017-9347-8
- Software:
  - Pattern Recognition for Neuroimaging Toolbox (PRoNTo), http://www.mlnl.cs.ucl.ac.uk/pronto/
  - Nilearn: Machine learning for Neuro-Imaging in Python, https://nilearn.github.io/

Thank you for your attention!

@CodeWisdom

"A computer is like a mischievous genie.
It will give you exactly what you ask for,
but not always what you want."

- Joe Sondow