

From IA to Deep Learning

MOMI 2019

Frederic Precioso

Professor in Computer Science at Polytech'Nice
Laboratoire I3S, UMR CNRS 7271

Universite Cote d'Azur

precioso@univ-cotedazur.fr

Disclaimer

If any content in this presentation is yours but is not correctly referenced or if it should be removed, please just let me know and I will correct it.

Research group

- Former PhDs
 - Stéphanie Lopez, *Interactive Content-based Retrieval based from Eye-tracking*
 - Ameni Bouaziz, *Short message mining (tweet trends and events)*
 - Atheer Al-Nadji, *Multiple Clustering by consensus*
 - Romaric Pighetti, *Content-Based Information Retrieval combining evolutionary algorithms and SVM*
 - Katy Blanc, *Description, Analysis and Learning from Video Content*
 - Mélanie Ducoffe, *Active Learning for Deep Networks and their design*
- Current PhDs
 - **John Anderson Garcia Henao**, *Green Deep Learning for Health*
 - **Edson Florez Suarez**, *Deep Learning for Adversarial Drug Event detection*
 - **Miguel Romero Rondon**, *Network models and Deep Learning for new VR generation*
 - **Laura Melissa Sanabria Rosas**, *Video content analysis for sports video*
 - **Tianshu Yang**, *Machine learning for the revenue accounting workflow management*
 - **Laurent Vanni**, *Understanding Deep Learning for political discourse text analysis*
- Current Post-doc
 - **Sujoy Chatterjee**, *Multiple Consensus Clustering for Multidimensional and Imbalanced Data, AMADEUS*

Research group

- Former Post-docs
 - Geoffrey Portelli, Bio-Deep: A biology perspective for Deep Learning optimization and understanding, ANR Deep_In_France
 - Souad Chaabouni, From gaze to interactive classification using deep learning, ANR VISIIR
- Former Research Engineers
 - Lirone Samoun, 3D Object interactive search engine
 - Thomas Fisichella, Structure extraction from 3D point clouds
 - Lucas Malléus, *3D Object mining and recognition*
 - Ayattalah Aly Halim, *Human Action Recognition from 3D*
- Projects
 - ANR Recherche interactive d'image par commande visuelle (VISIIR) 2013-2018
 - ANR Deep_In_France 2017-2021
 - H2020 DigiArt: The Internet of Historical Things
 - Collaborations with Wildmoka, Amadeus, France Labs, Alcmeon, NXP, Renault, Bentley, Instant System, ESI France, SAP, Autodesk, Semantic Grouping Company...



Introduction: a new AI team @ UCA

To address those problems, [we are building a new research team](#) at UCA :

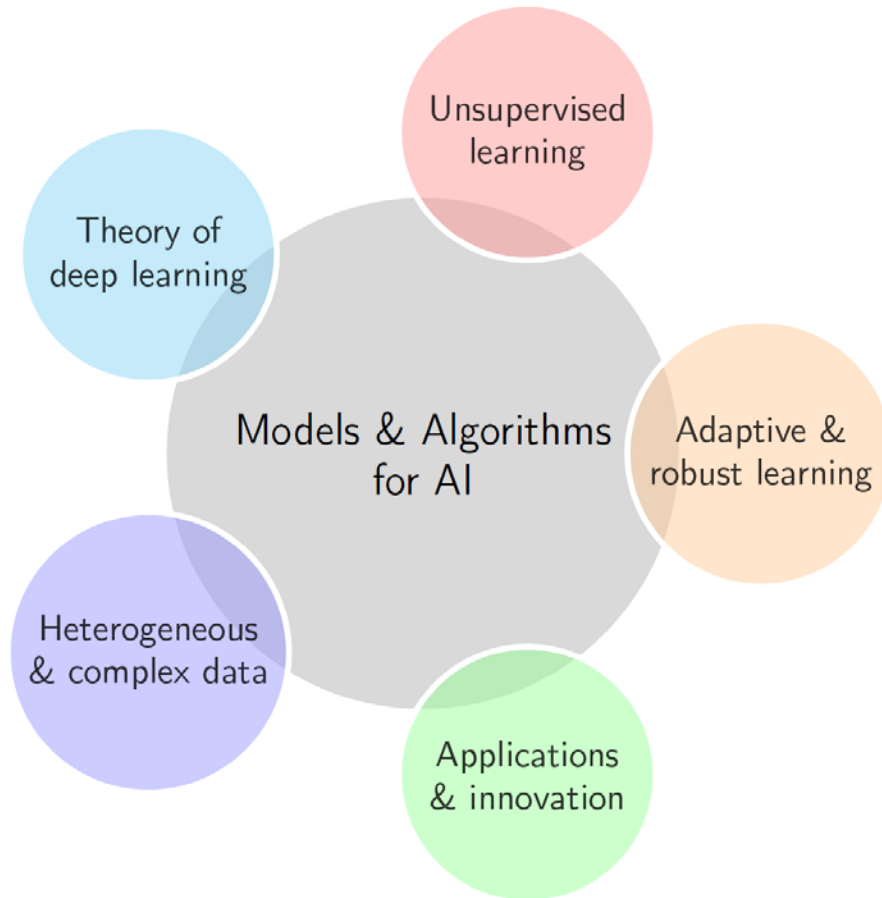


Figure: Scientific objectives of the (upcoming) Maasai team.



Overview

- **Context & Vocabulary**
 - *How is Artificial Intelligence defined?*
 - *Machine Learning & Data Mining?*
 - *Machine Learning & Data Science?*
 - *Machine Learning & Statistics?*
- Explicit supervised learning
- Implicit supervised learning

CONTEXT & VOCABULARY

HOW IS ARTIFICIAL INTELLIGENCE DEFINED?

How is Artificial intelligence defined?

- The term ***Artificial Intelligence***, as a research field, was coined at the conference on the campus of Dartmouth College in the summer of **1956**, even though the idea was around since antiquity (*Hephaestus built automatons of metal to work for him*, or the *Golem in Jewish folklore*, etc).
- For instance in the first manifesto of Artificial Intelligence, ***"Intelligent Machinery"***, in **1948** Alan Turing distinguished two different approaches to AI, which may be termed ***"top-down"*** or ***knowledge-driven AI*** and ***"bottom-up"*** or ***data-driven AI***

How is Artificial intelligence defined?

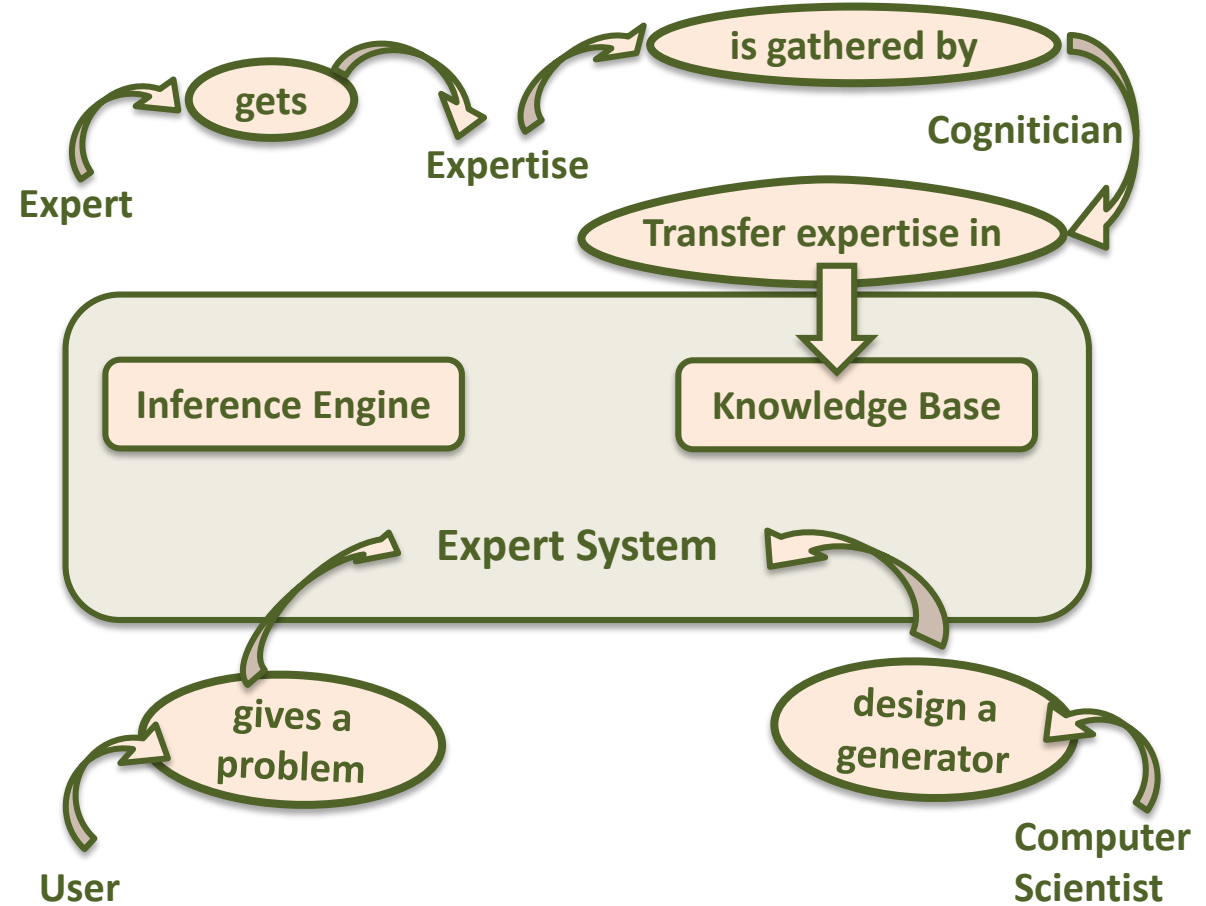
- **"top-down" or knowledge-driven AI**
 - cognition = high-level phenomenon, independent of low-level details of implementation mechanism, first neuron (1943), first neural network machine (1950), neucognitron (1975)
 - Evolutionary Algorithms (1954,1957, 1960), Reasoning (1959,1970), Expert Systems (1970), Logic, Intelligent Agent Systems (1990)...
- **"bottom-up" or data-driven AI**
 - opposite approach, start from data to build incrementally and mathematically mechanisms taking decisions
 - Machine learning algorithms, Decision Trees (1983), Backpropagation (1984-1986), Random Forest (1995), Support Vector Machine (1995), Boosting (1995), Deep Learning (1998/2006)...

How is Artificial intelligence defined?

- *AI is originally defined in 1956, by Marvin Lee Minsky:*
*“The construction of computer programs doing tasks, that are, **for the moment**, accomplished **more satisfyingly** by human beings because they require **high level mental processes** such as: learning. perceptual organization of memory and critical reasoning”.*
- There are so the “artificial” side with the usage of computers or sophisticated electronic processes and the side “intelligence” associated with its goal to imitate the (human) behavior.

Artificial Intelligence, Top-Down


- Example of an expert system:





Artificial Intelligence, Top-Down

- Expert system:

 AI Manufacturing Supply Chain Robo Dev Healthcare CRO Events All Topics ▾

Login | Join RBR Insider 🔍

A Cyclist's Encounter with an Indecisive Google Self-Driving Car

A bicyclist recently had a two-minute standoff with a Google self-driving car at a four-way stop in Austin, Texas. So what happened? We explain.

🕒 AUGUST 26, 2015 👤 STEVE CROWE

BRIEF

STAT: IBM's Watson gave 'unsafe and incorrect' cancer treatment advice

AUTHOR
Meg Bryant

PUBLISHED
July 26, 2018

SHARE IT

Dive Brief:

- A STAT review of internal IBM documents suggests the company's Watson supercomputer wrongly advised doctors on how to treat patients' cancers.
- The documents — slides presented by then-IBM Watson Health deputy chief health officer Andrew Norden in June and July of last year — include “multiple examples of unsafe and incorrect treatment



Why Artificial Intelligence is so difficult to grasp?

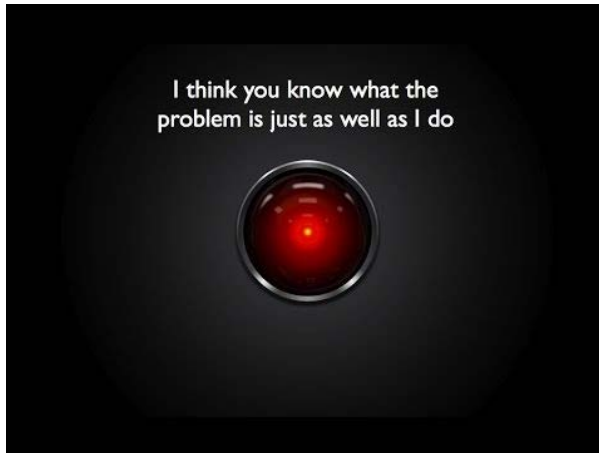
- Frequently, when a technique reaches mainstream use, it is no longer considered as artificial intelligence; this phenomenon is described as the AI effect: "AI is whatever hasn't been done yet." (*Larry Tesler's Theorem*) -> e.g. Path Finding (GPS), Checkers game, Chess electronic game, Alpha Go...

⇒ "AI" is continuously evolving and so very difficult to grasp.



How is Artificial intelligence defined?

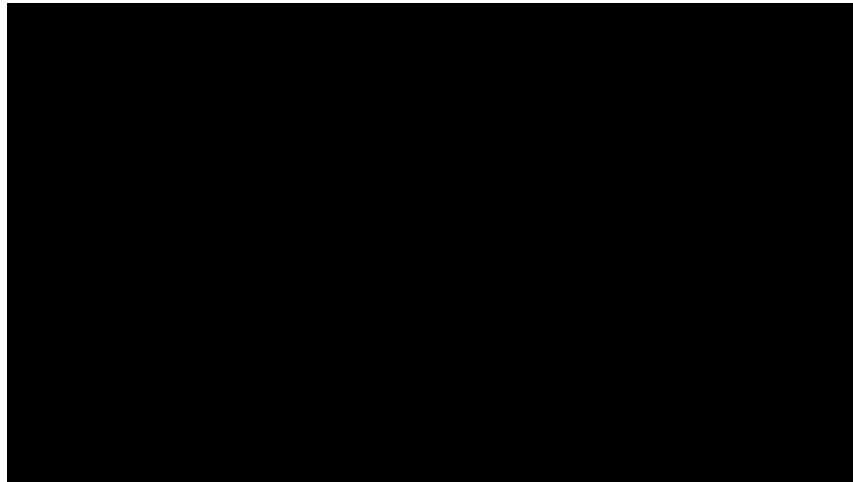
- The concept of ***strong artificial intelligence*** makes reference to a machine capable not only of producing intelligent behavior, but also to experience a feeling of a real sense of itself, “real feelings” (whatever may be put behind these words), and "an understanding of its own arguments”.





How is Artificial intelligence defined?

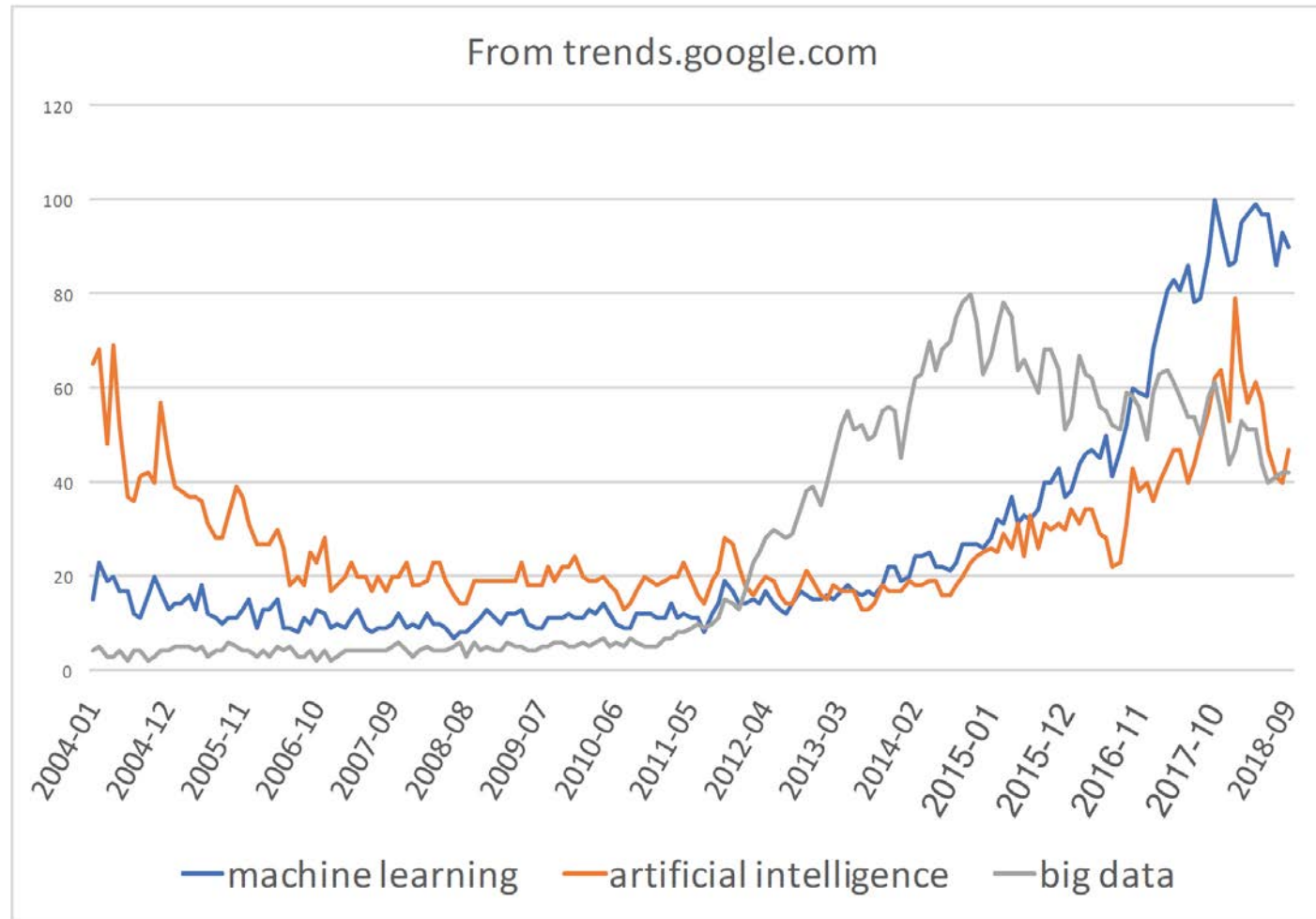
- The notion of ***weak artificial intelligence*** is a pragmatic approach of engineers: targeting to build more autonomous systems (to reduce the cost of their supervision), algorithms capable of solving problems of a certain class, etc. But this time, the machine *simulates* the intelligence, it seems to act as if it was smart.



Trolley dilemma (a two-year old solution)



An AI Revolution?

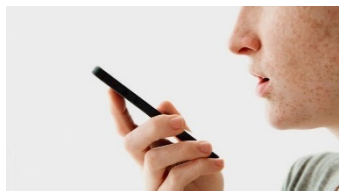


Credits F. Bach

MACHINE LEARNING

Machine Learning

$\begin{pmatrix} \mathbf{x} \end{pmatrix}$



$$\begin{pmatrix} \mathbf{x} \end{pmatrix} \xrightarrow{f(\mathbf{X}, \alpha) ?} y$$

Face Detection



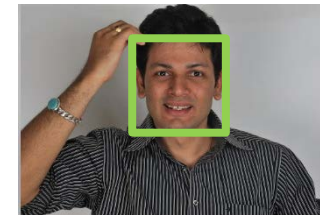
Betting on sports



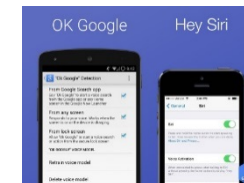
Speech Recognition



y

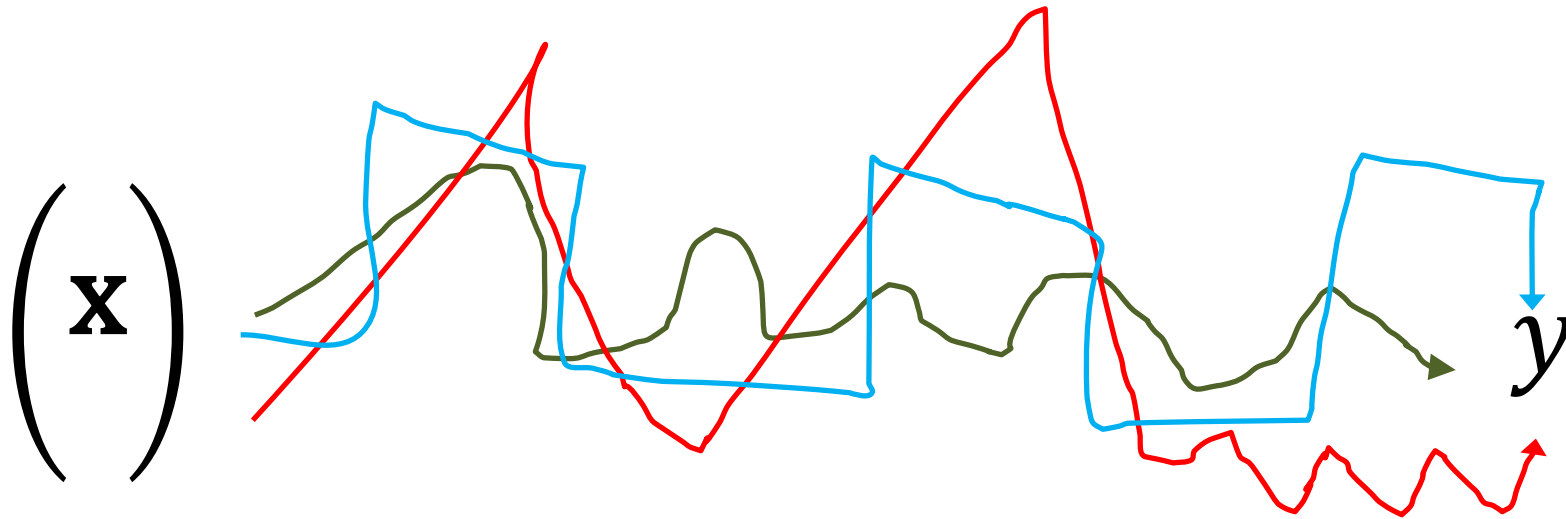


Scores, ranking...



Machine Learning

$$\begin{pmatrix} \mathbf{X} \end{pmatrix} \xrightarrow{f(\mathbf{X}, \alpha) ?} y$$



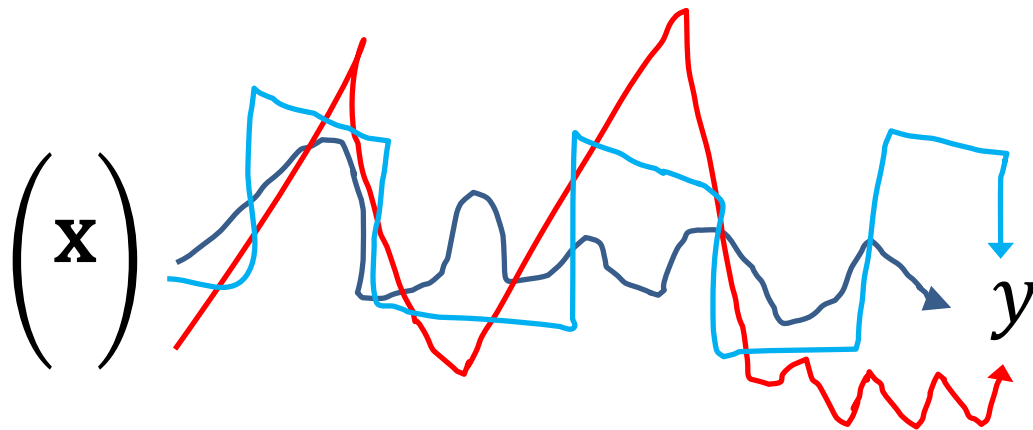
Support Vector Machines

Boosting

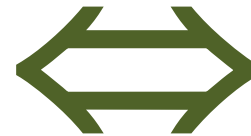
Artificial Neural Networks

Machine Learning

$$\begin{pmatrix} \mathbf{x} \end{pmatrix} \xrightarrow{f(\mathbf{X}, \alpha) ?} y$$



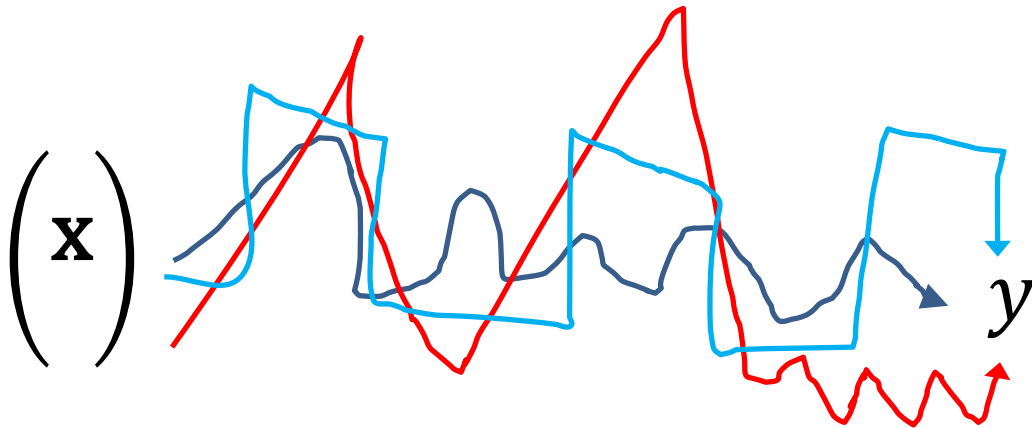
ML



“Weather Forecasting”

Machine Learning

$$\begin{pmatrix} \mathbf{x} \end{pmatrix} \xrightarrow{f(\mathbf{X}, \alpha) ?} y$$



ML

\neq

AI

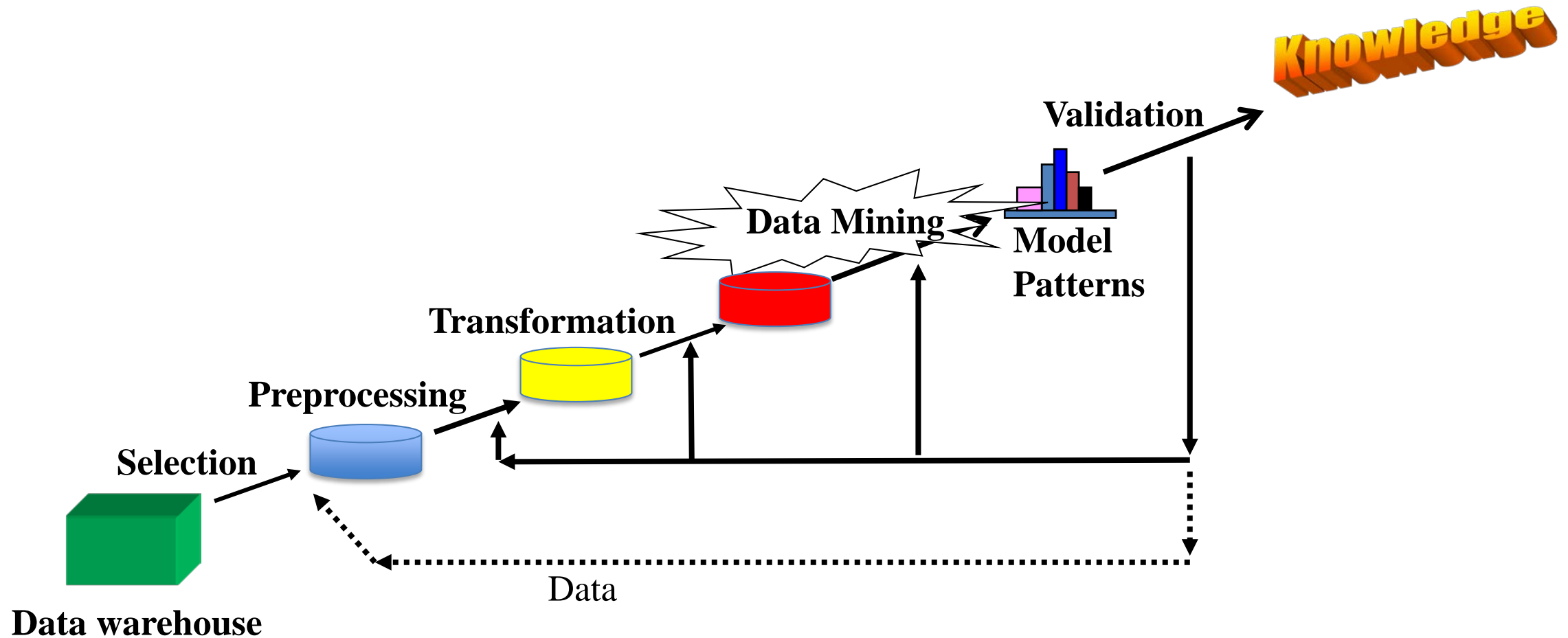
Francis Bach at *Frontier Research and Artificial Intelligence Conference*: “Machine Learning is not AI”

(https://erc.europa.eu/sites/default/files/events/docs/Francis_Bach-SEQUOIA-Robust-algorithms-for-learning-from-modern-data.pdf

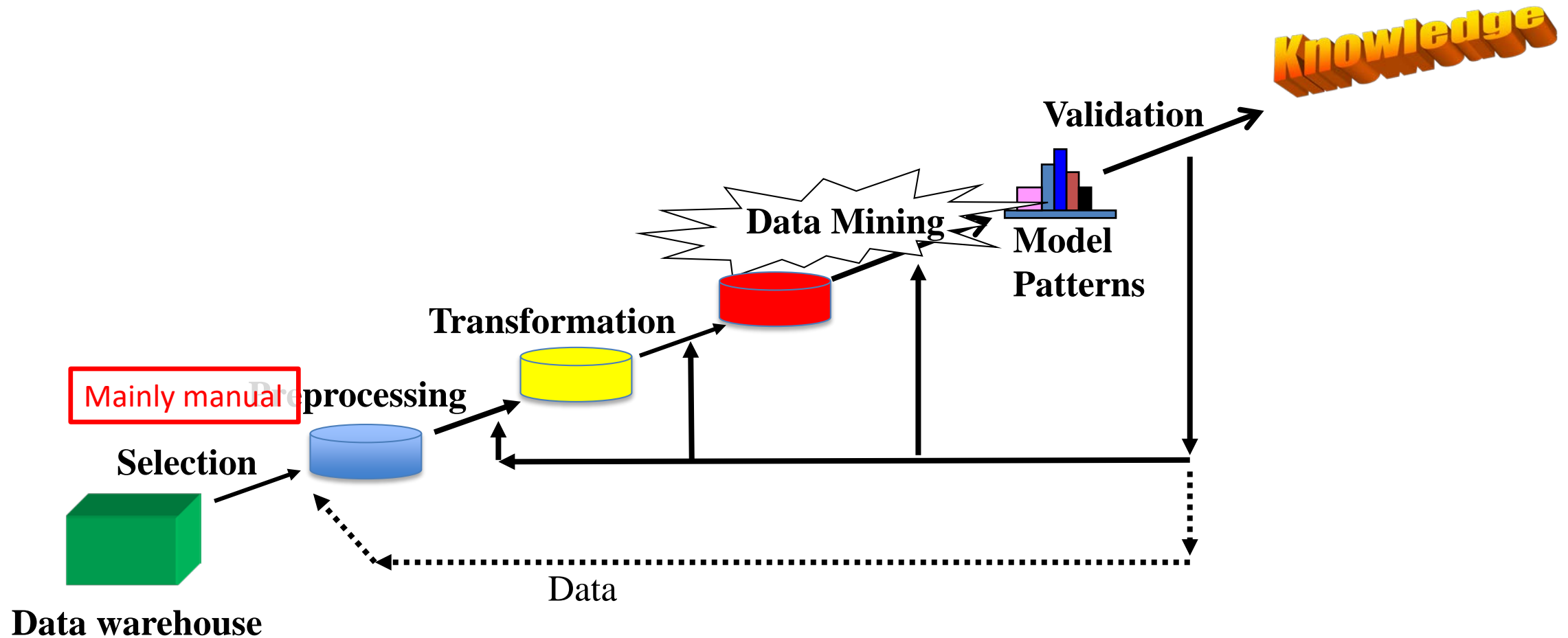
<https://webcast.ec.europa.eu/erc-conference-frontier-research-and-artificial-intelligence-25#>)

MACHINE LEARNING & DATA MINING?

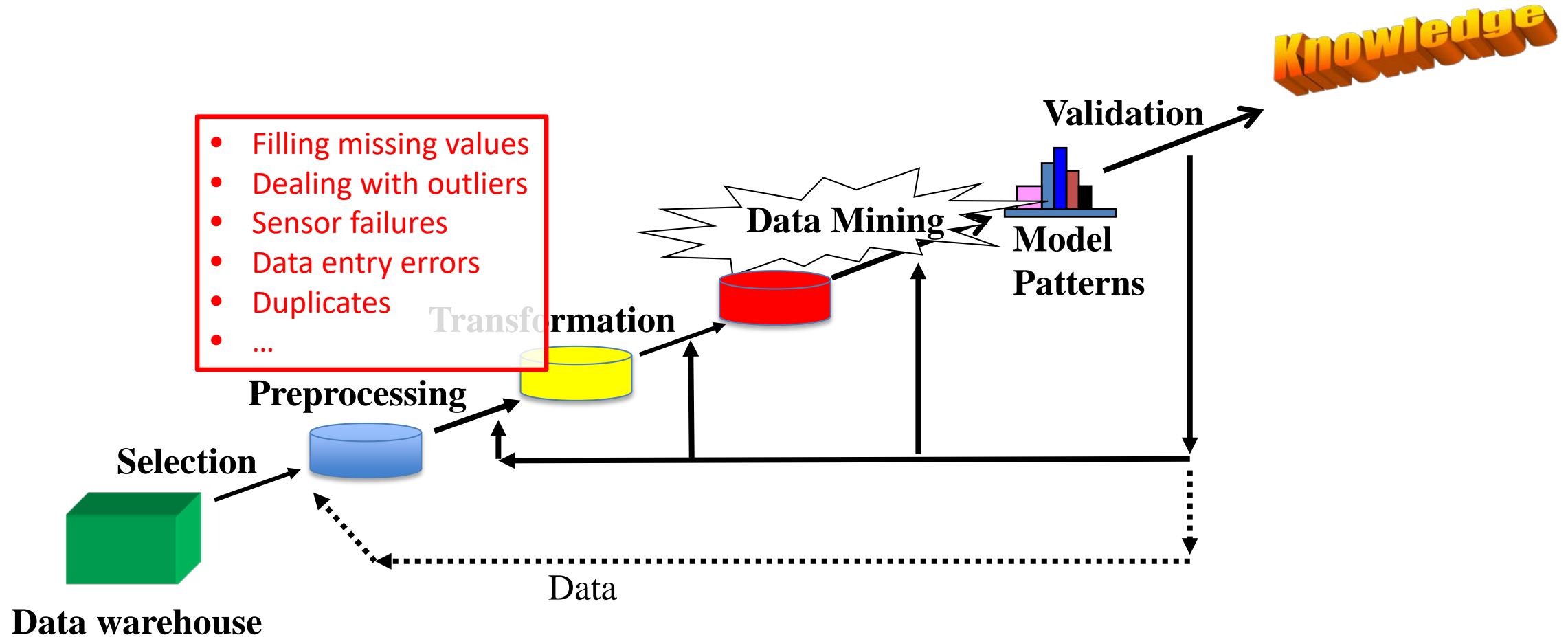
Data Mining Workflow



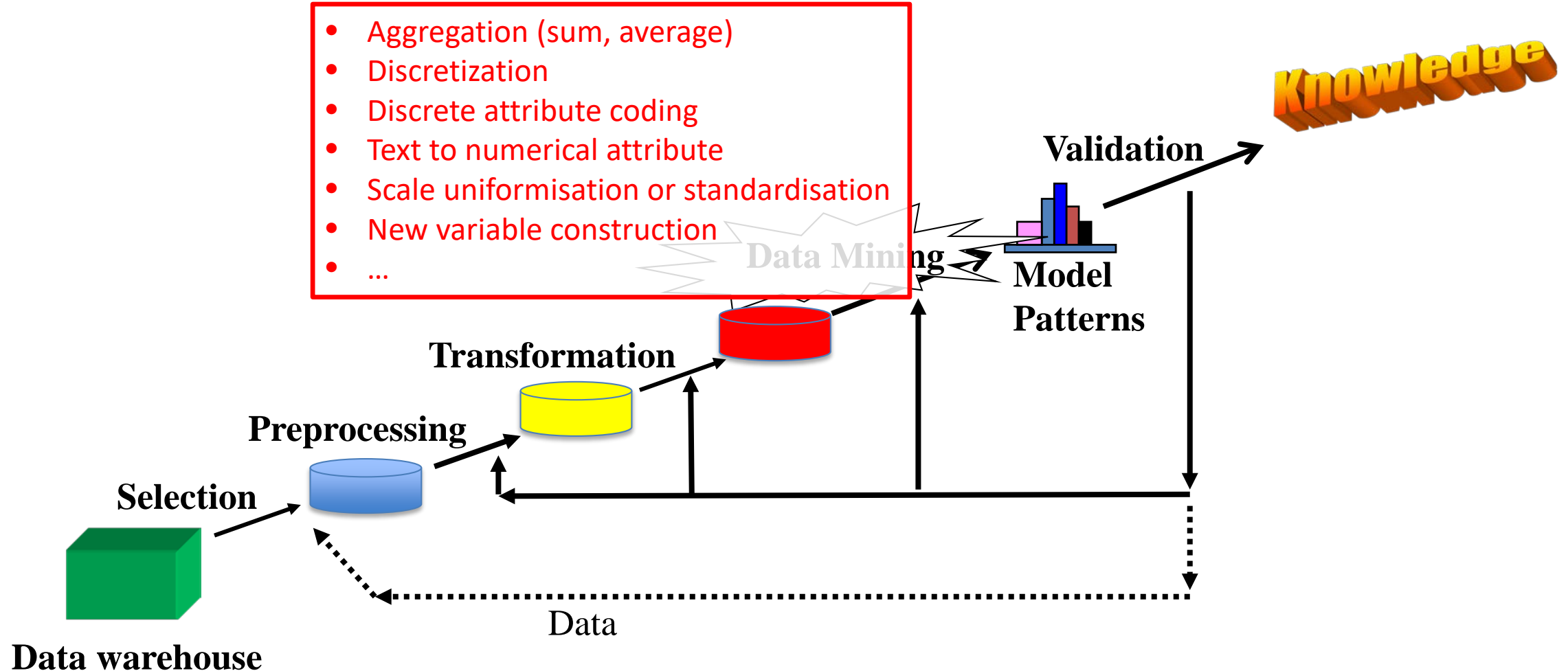
Data Mining Workflow



Data Mining Workflow

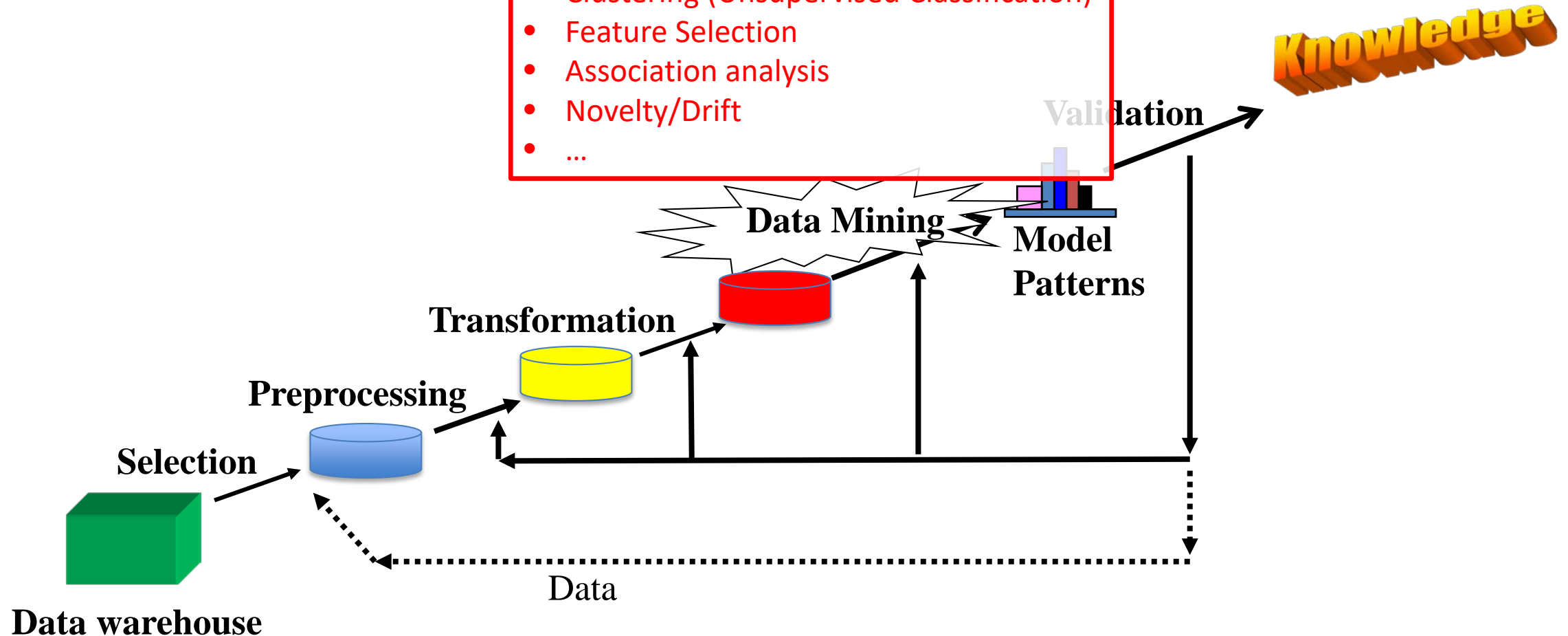


Data Mining Workflow



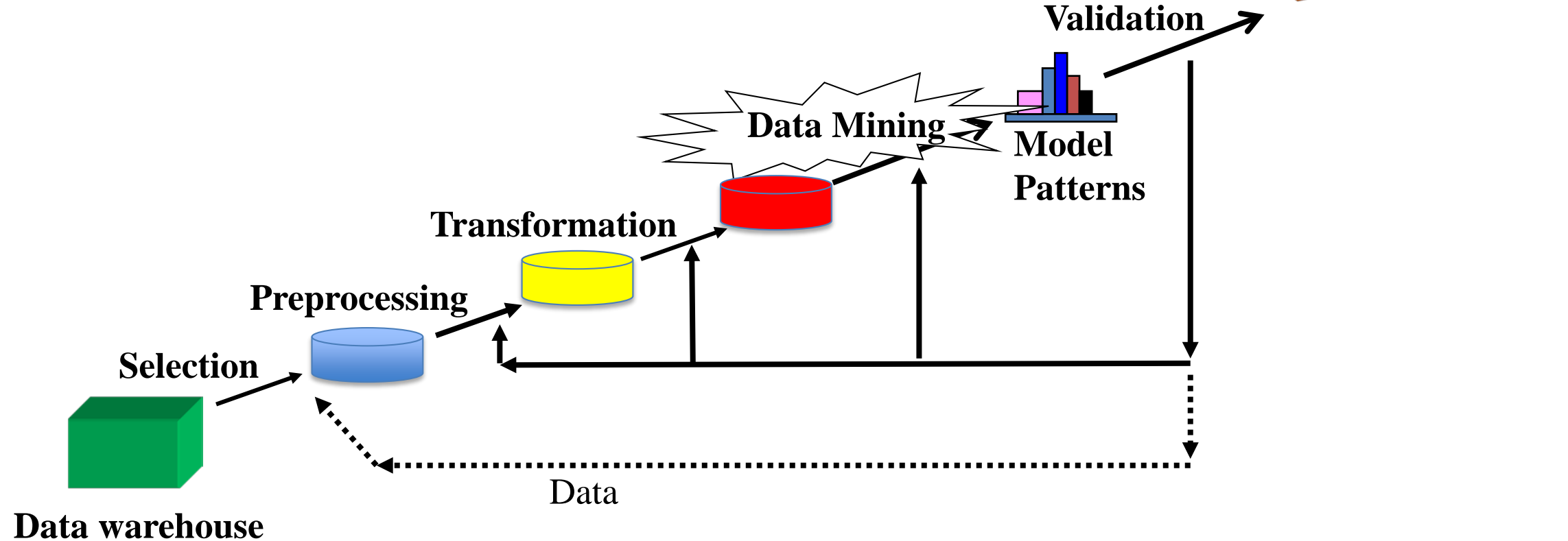
Data Mining Workflow

- Regression
- (Supervised) Classification
- Clustering (Unsupervised Classification)
- Feature Selection
- Association analysis
- Novelty/Drift
- ...



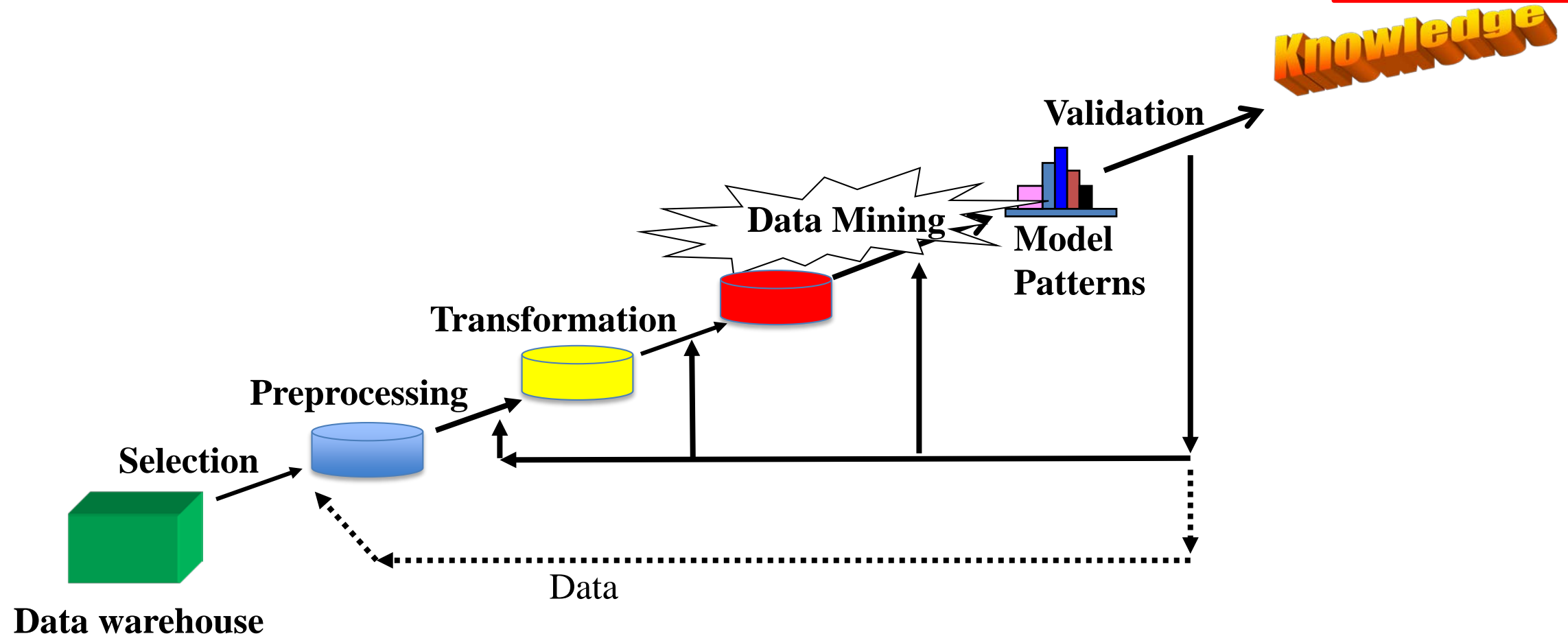
Data Mining Workflow

- Evaluation on Validation Set
- Evaluation Measures
- Visualization
- ...



Data Mining Workflow

- Visualization
- Reporting
- Knowledge
- ...



Data Mining Workflow

Problems

- Regression
- (Supervised) Classification
- Density Estimation / Clustering (Unsupervised Classification)
- Feature Selection
- Association analysis
- Anomaly/Novelty/Drift
- ...

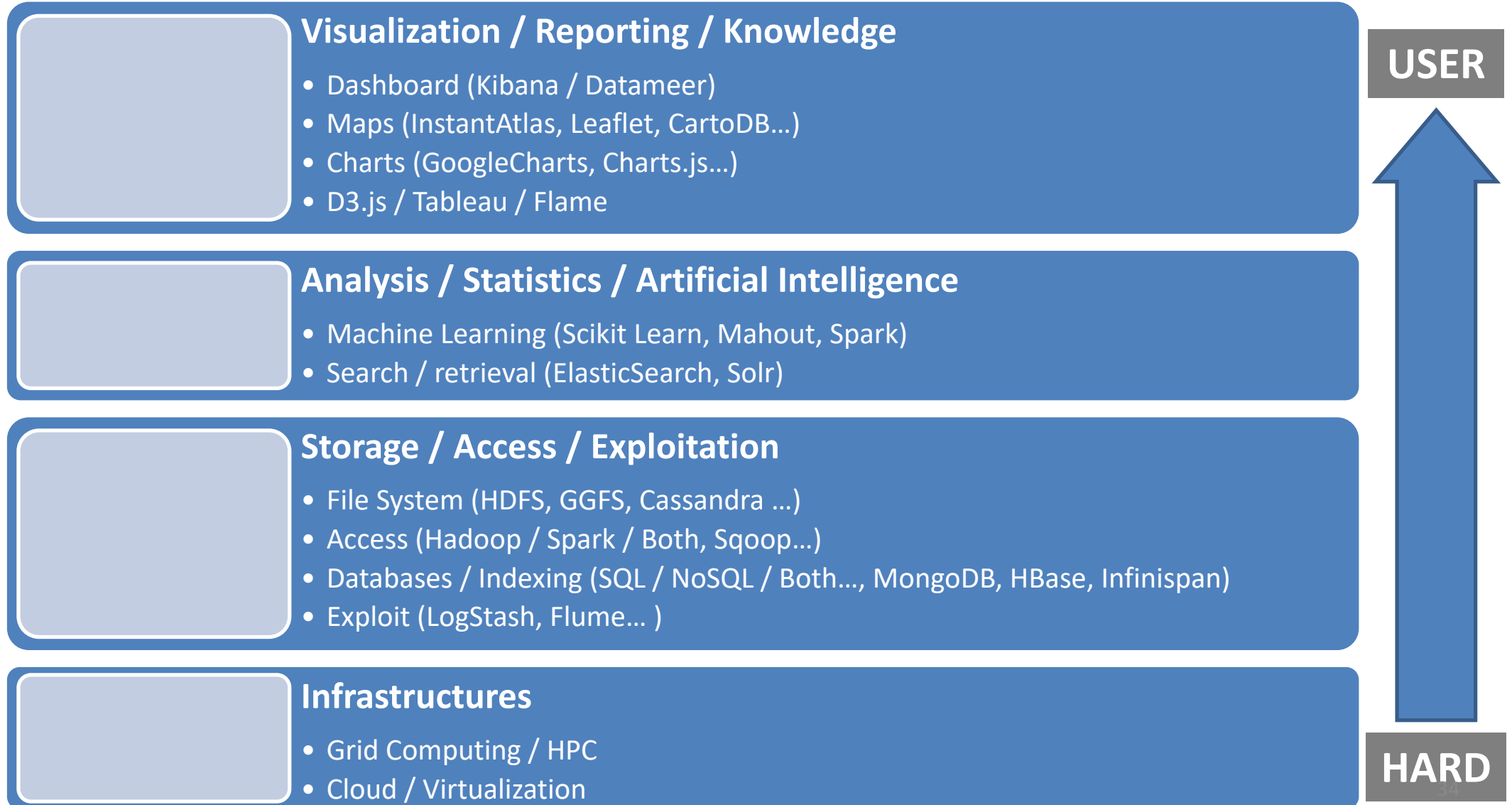
Possible Solutions

- Machine Learning
 - Support Vector Machine
 - Artificial Neural Network
 - Boosting
 - Decision Tree
 - Random Forest
 - ...
- Statistical Learning
 - Gaussian Models (GMM)
 - Naïve Bayes
 - Gaussian processes
 - ...
- Other techniques
 - Galois Lattice
 - ...

MACHINE LEARNING & DATA SCIENCE?

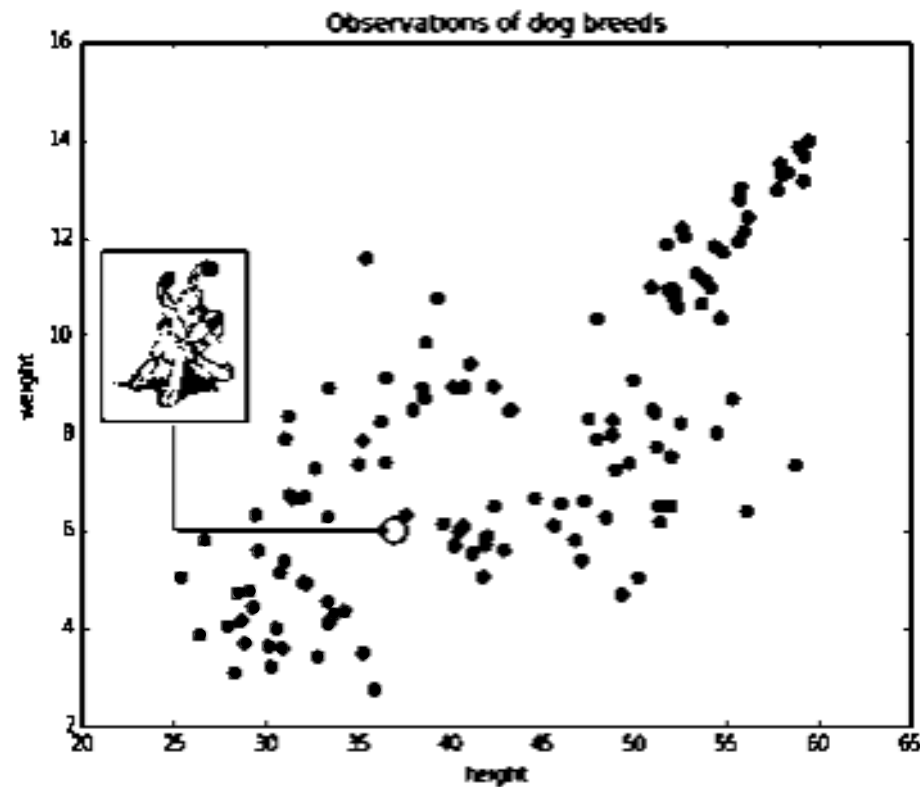


Data Science Stack



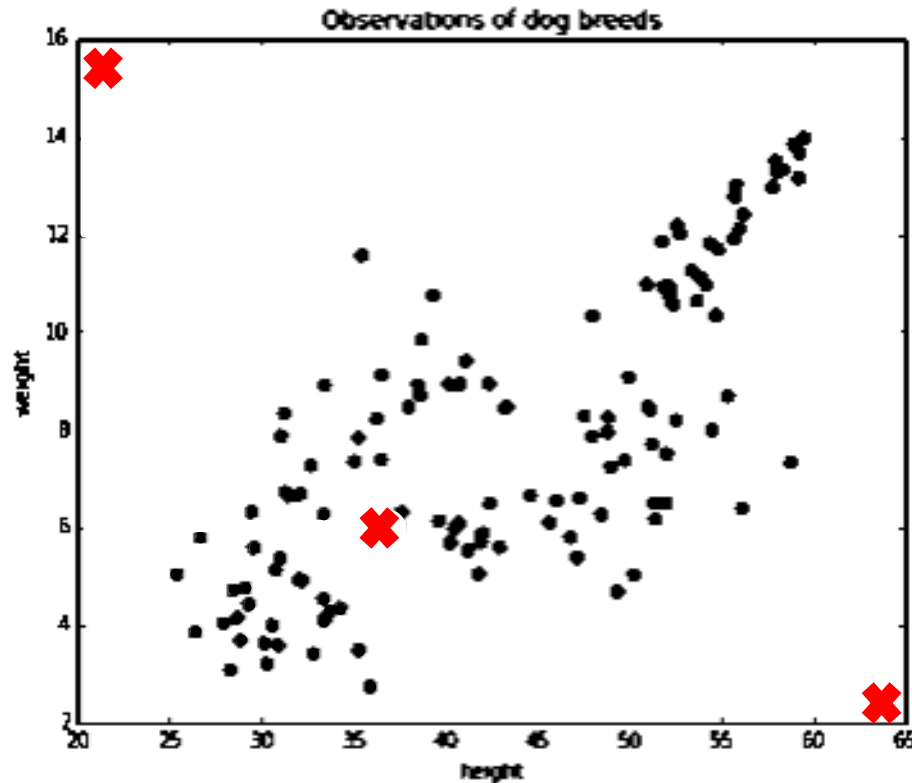
MACHINE LEARNING & STATISTICS?

What breed is that Dogmatix (Idéfix) ?



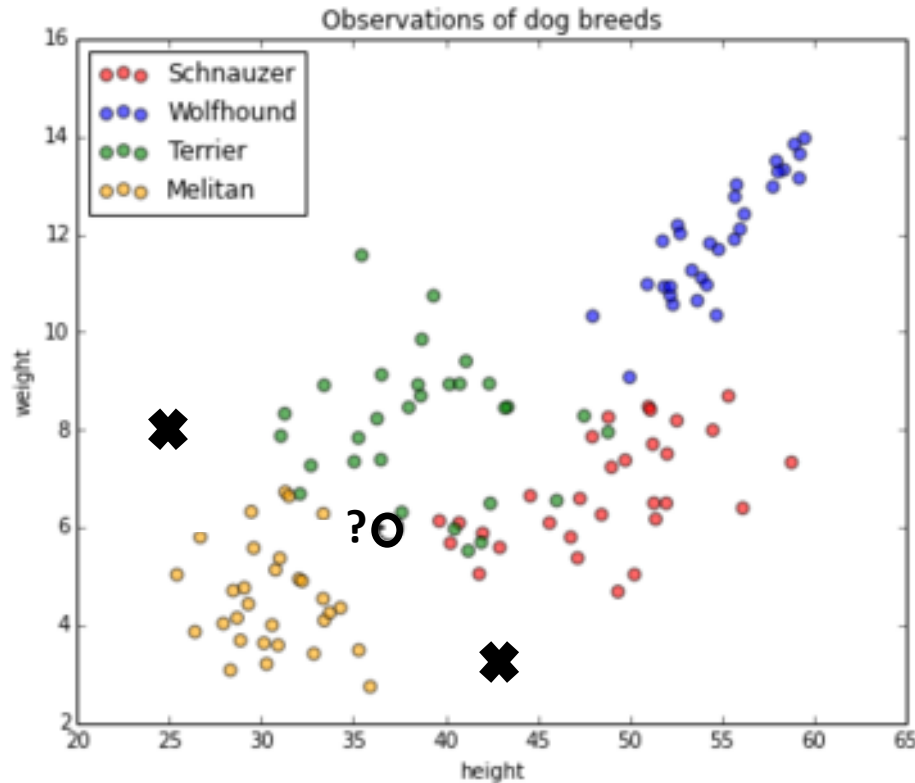
The illustrations of the slides in this section come from the blog "Bayesian Vitalstatistix: What Breed of Dog was Dogmatix?"

Does any real dog get this height and weight?



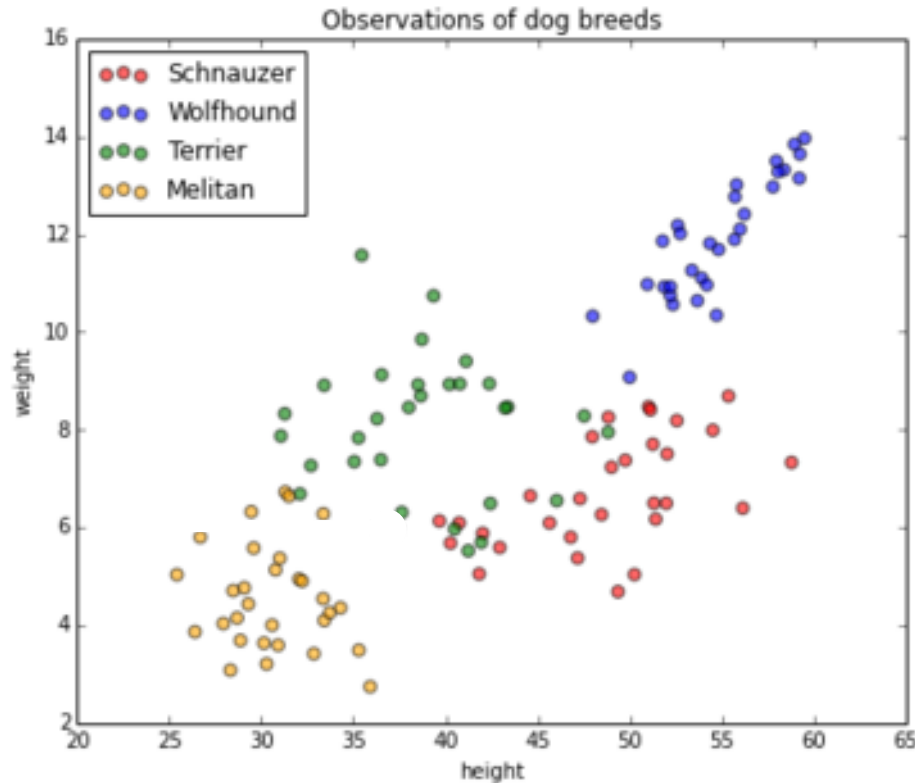
- Let us consider \mathbf{x} , vectors independently generated in \mathbf{R}^d (here \mathbf{R}^2), following a probability distribution fixed but *unknown* $P(\mathbf{x})$.

What should be the breed of these dogs?



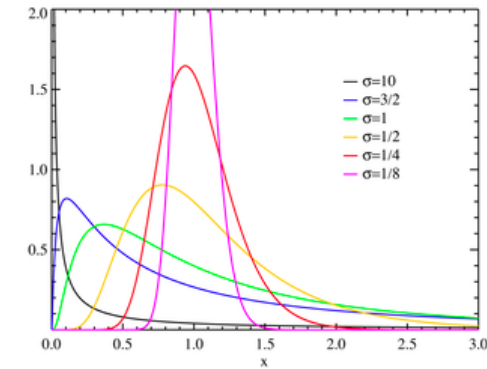
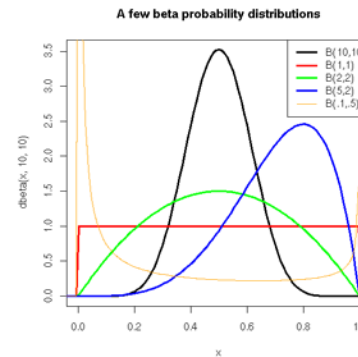
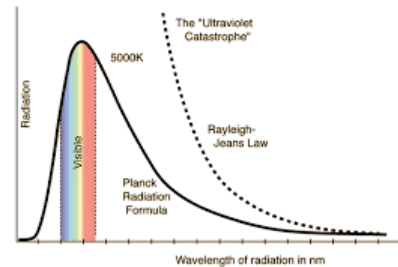
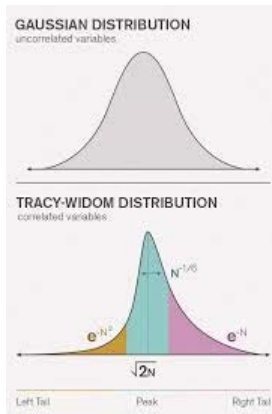
- An Oracle assigns a value y to each vector \mathbf{x} following a probability distribution $P(y/\mathbf{x})$ also fixed but *unknown*.

An oracle provides me with examples?

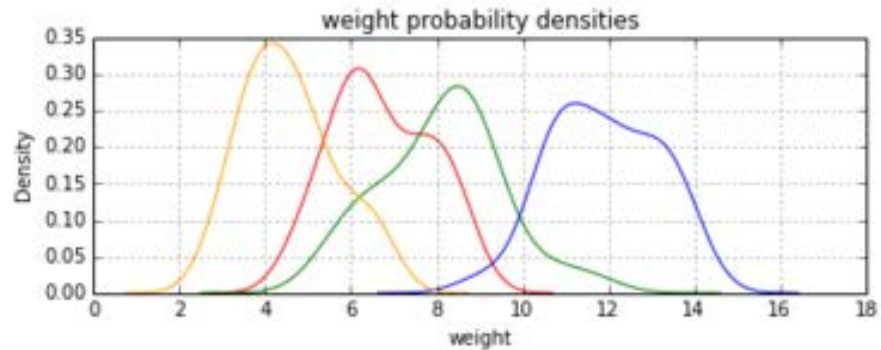
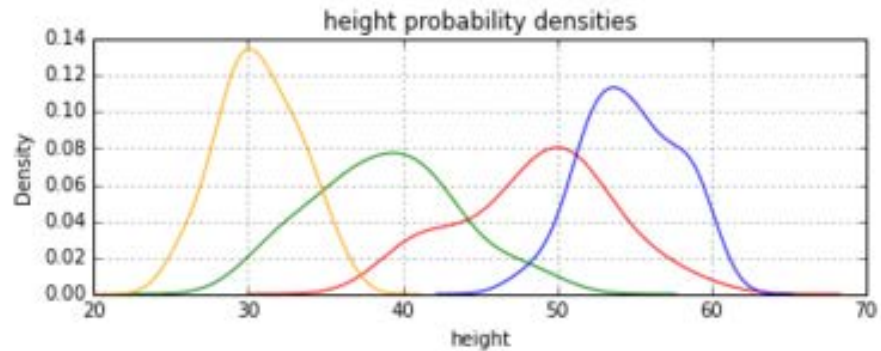
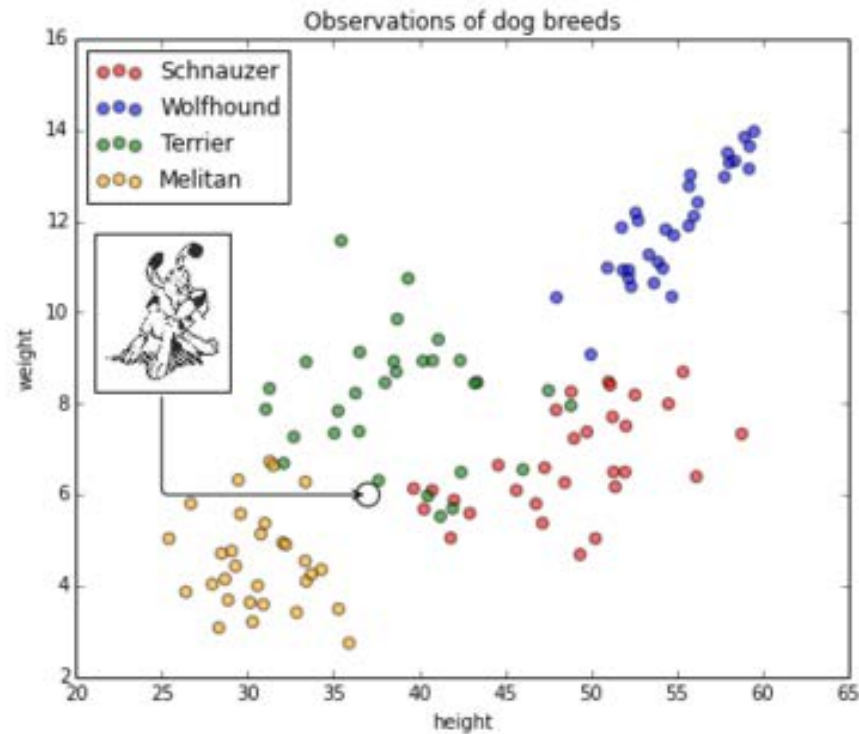


- Let S be a training set
 $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$,
with m **training samples i.i.d.** which
follow the **joint probability**
 $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$.

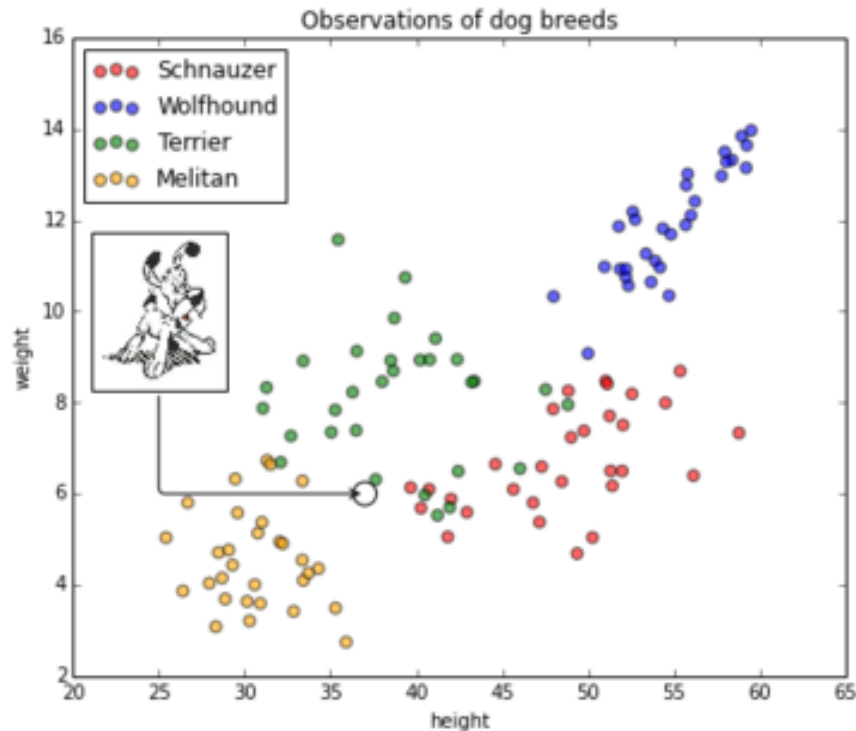
Statistical solution: Models, Hypotheses...



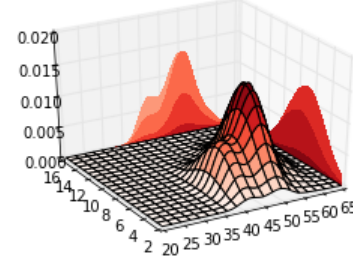
Statistical solution $P(\text{height, weight} \mid \text{breed})$...



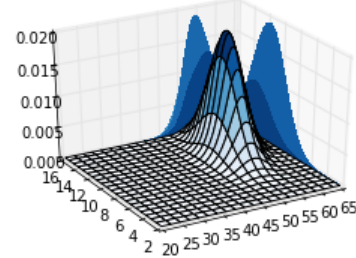
Statistical solution $P(\text{height, weight} \mid \text{breed})$...



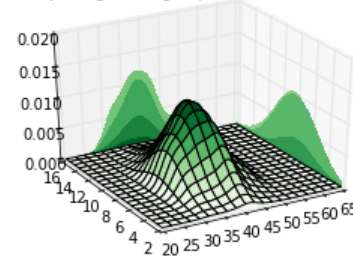
Joint Likelihood
 $p(\text{height, weight} \mid \text{breed} = \text{schnauzer})$



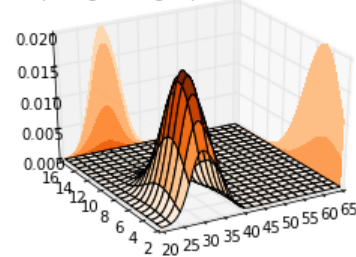
Joint Likelihood
 $p(\text{height, weight} \mid \text{breed} = \text{wolfhound})$



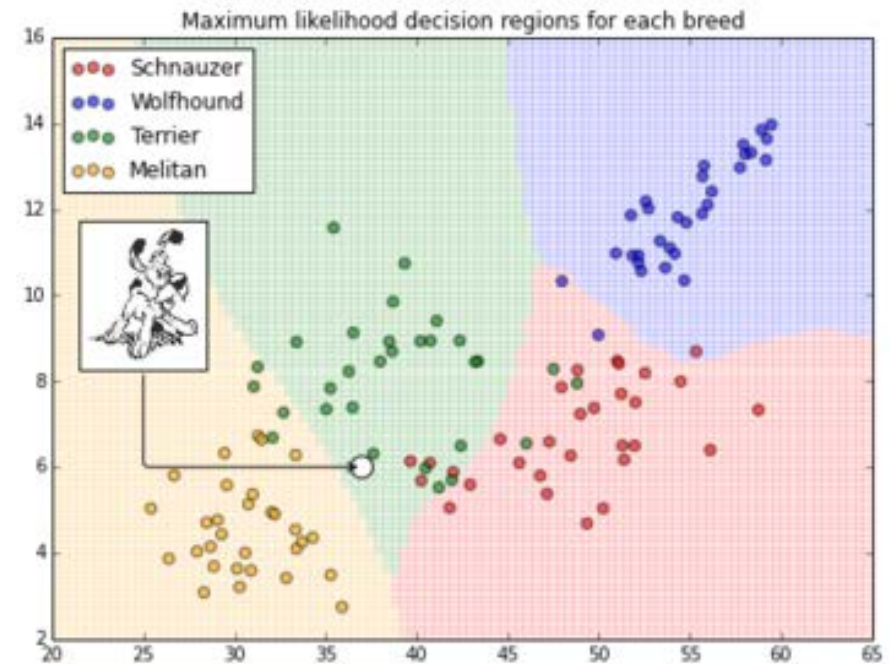
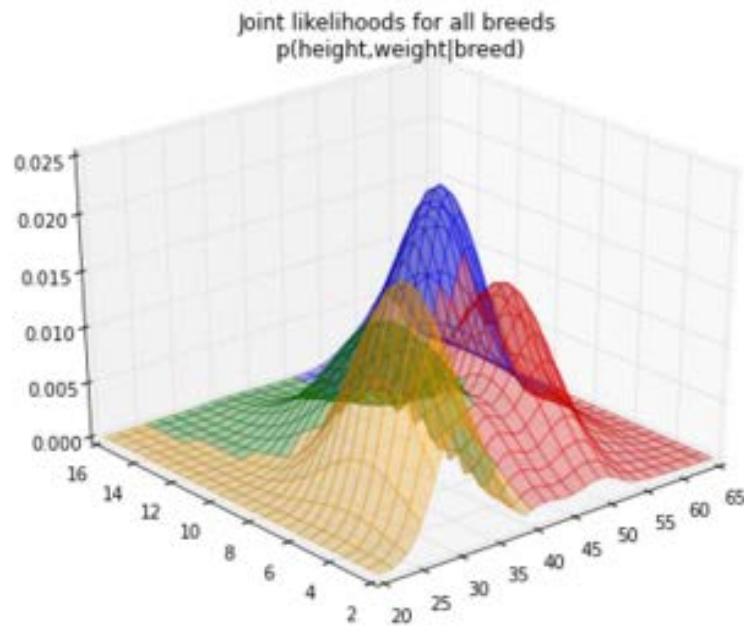
Joint Likelihood
 $p(\text{height, weight} \mid \text{breed} = \text{terrier})$



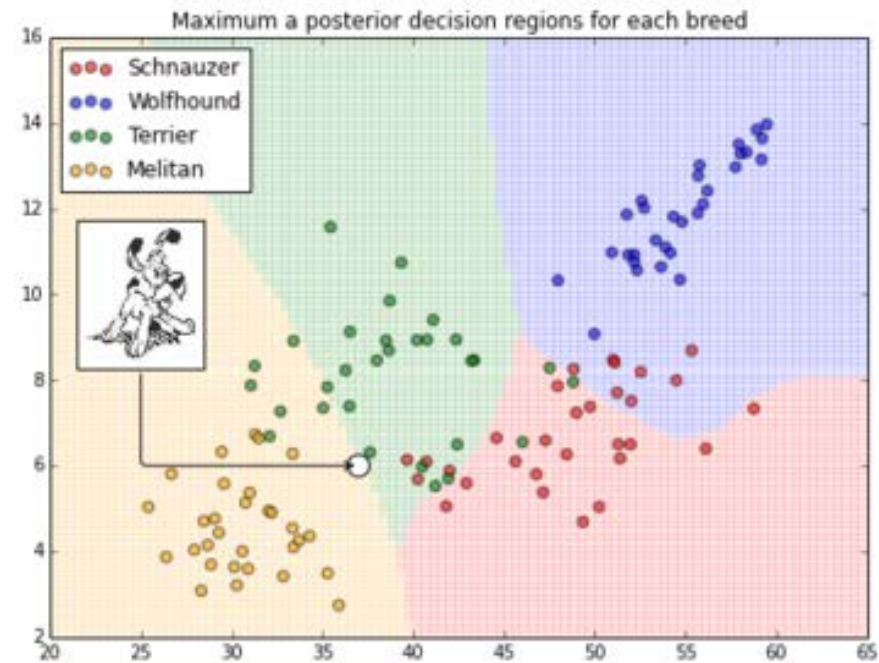
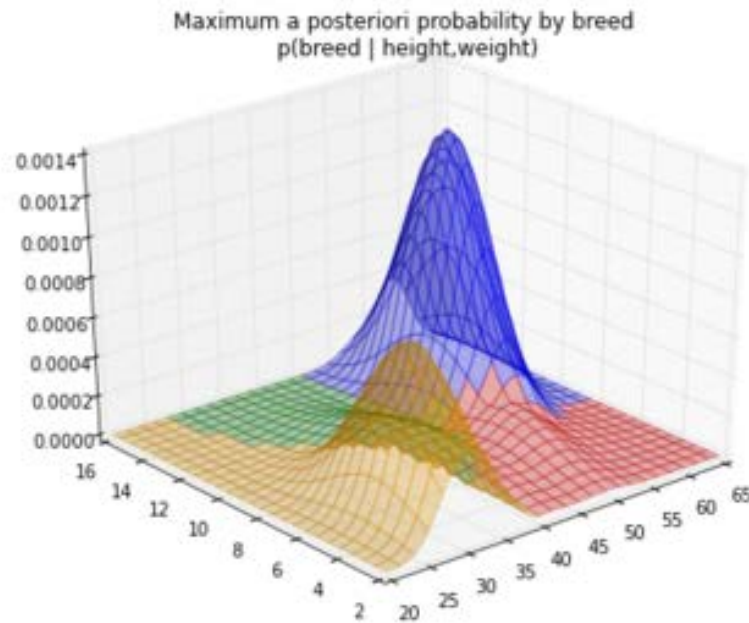
Joint Likelihood
 $p(\text{height, weight} \mid \text{breed} = \text{melitan})$



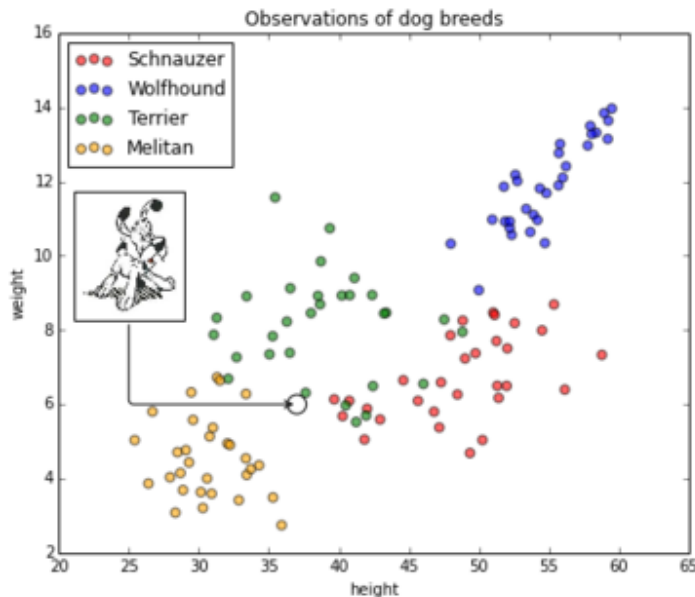
Statistical solution $P(\text{height, weight} | \text{breed})$...



Statistical solution: Bayes, $P(\text{breed} | \text{height, weight})$...



Machine Learning



- we have a learning machine which can provide a family of functions $\{f(\mathbf{x};\alpha)\}$, where α is a set of parameters.

$$\left(\mathbf{x} \right) \xrightarrow{f(\mathbf{X},\alpha) ?} y$$

The problem in Machine Learning

$$\left(\mathbf{x} \right) \xrightarrow{f(\mathbf{X}, \alpha) ?} y$$

- The problem of learning consists in finding *the model* (among the $\{f(\mathbf{x}; \alpha)\}$) which provides *the best approximation* \hat{y} of the true label y given by the Oracle.
- *best* is defined in terms of minimizing a specific (error) cost *related to your problem/objectives*
 $Q((\mathbf{x}, y), \alpha) \in [a; b]$.
- Examples of cost/loss functions Q : Hinge Loss, Quadratic Loss, Cross-Entropy Loss, Logistic Loss...

Loss in Machine Learning

- ***How to define the loss L (or the cost Q)?***

You should choose the right loss function based on your problem and your data (*here y is the true/expected answer, $f(x)$ the answer predicted by the network*).

Classification

- **Cross-entropy loss:** $L(x) = -(y \ln(f(x)) + (1-y) \ln(1-f(x)))$
- **Hinge Loss** (i.e. max-margin loss, i.e. 0-1 loss): $L(x) = \max(0, 1-yf(x))$
- ...

Regression

- **Mean Square Error** (or Quadratic Loss): $L(x) = (f(x)-y)^2$
- **Mean Absolute Loss:** $L(x) = |f(x)-y|$
- ...

If the loss is minimized but accuracy is low, you should check the loss function. Maybe it is not the appropriate one for your task.



The problem in Machine Learning

For Clarity sake, let us note $z = (\mathbf{x}, y)$.

- ▶ Thus, the objective is to minimize the **Risk**, i.e. the expectation of the error cost:

$$R(\alpha) = \int Q(z, \alpha) dP(z)$$

where $P(z)$ is unknown.

The training set $S = \{z_i\}_{i=1, \dots, m}$ is built through an *i.i.d.* sampling according to $P(z)$. Since we cannot compute $R(\alpha)$, we look for minimizing the **Empirical Risk** instead:

$$R_{emp}(\alpha) = \frac{1}{m} \sum Q(z_i, \alpha)$$

Machine Learning fundamental Hypothesis

For Clarity sake, let us note $z = (\mathbf{x}, y)$.

$S = \{z_i\}_{i=1,\dots,m}$ is built through an *i.i.d.* sampling according to $P(z)$.

Machine Learning  *Statistics*

Train through Cross-Validation

Machine Learning  *Statistics*

Training set & Test set have to be distributed according to the same law (i.e. $P(z)$).



Vapnik learning theory (1995)

Vapnik had proven the following equation $\forall m$ with a probability at least equal to $1 - \eta$:

$$R(\alpha_m) \leq R_{emp}(\alpha_m) + (b - a) \sqrt{\frac{d_{VC} (\ln(2m/d_{VC}) + 1) - \ln(\eta/4)}{m}}$$

Training Error

Generalization Error

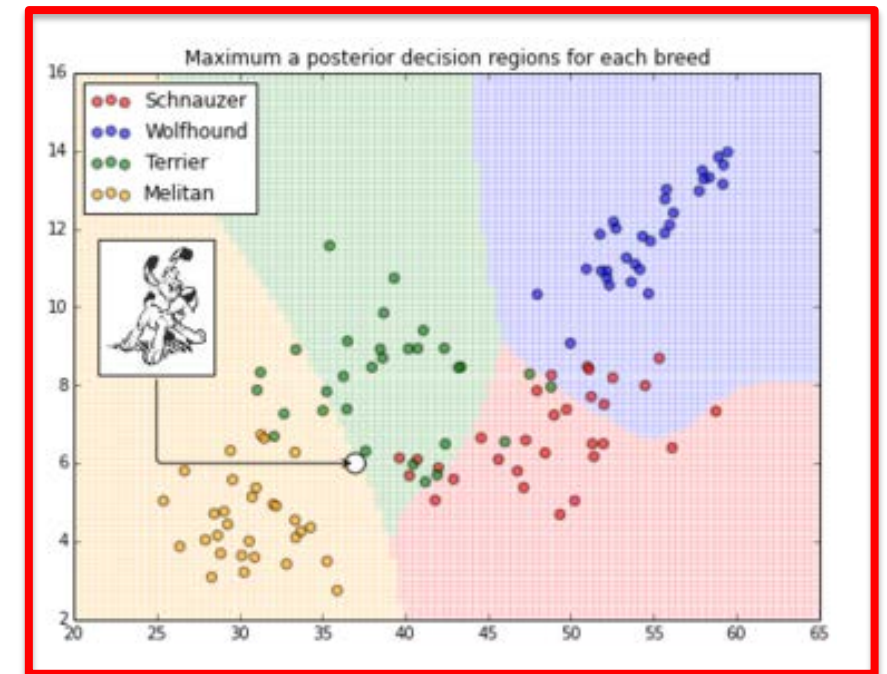
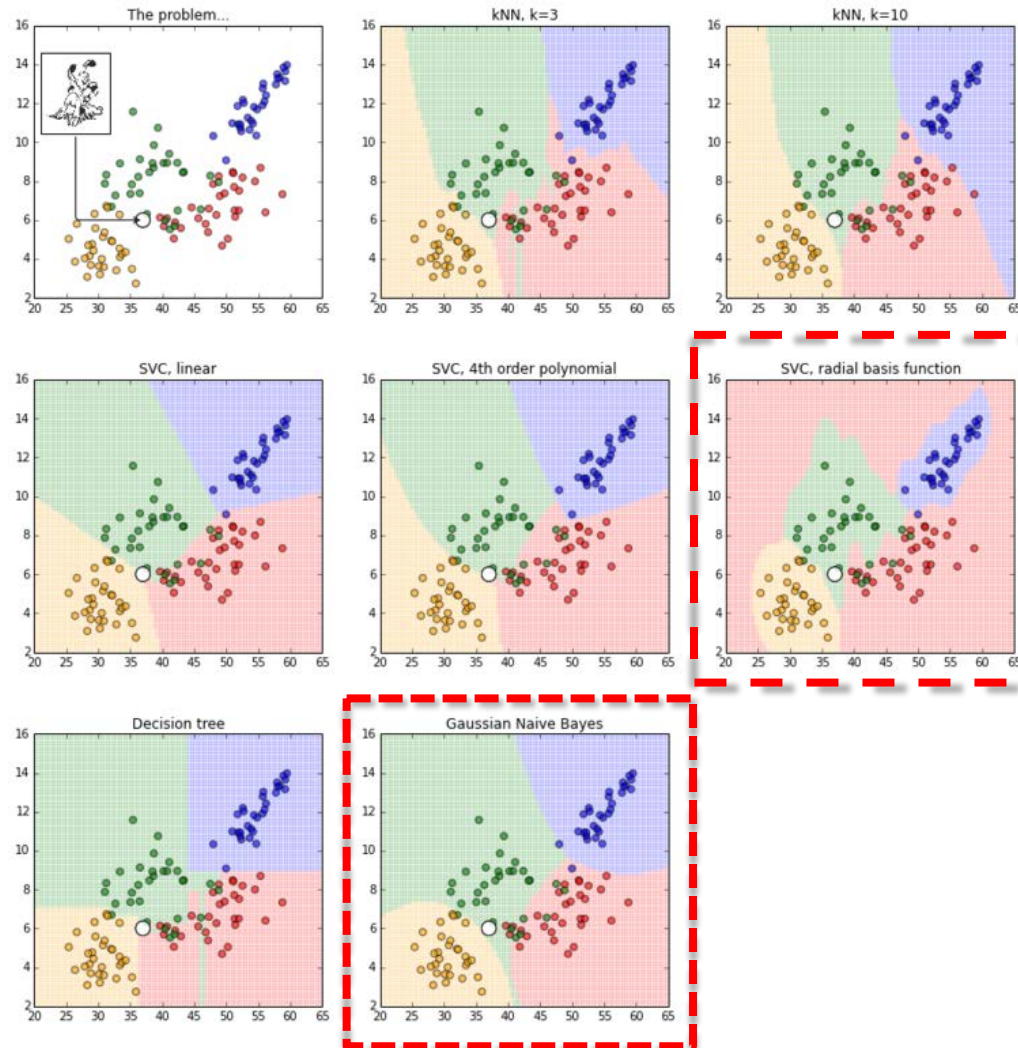
Thus minimizing the **Risk** depends on minimizing the **Empirical Risk** and the **confidence interval** which is linked to the term d_{VC} corresponding to the complexity of the model family chosen, i.e. the Vapnik-Chervonenkis dimension

Vapnik learning theory (1995)

In his learning theory [Vapnik, 1995], Vapnik defines 4 fundamental steps:

- Study the theory of consistence of learning processes
- Define bounds on convergence speed of learning processes
- Handle the generalization power of learning processes
- Design a theory to build learning algorithms in order to find a tradeoff between minimizing the ***Empirical Risk*** and the ***confidence interval*** \Rightarrow minimization of the ***Structural Risk***.

Machine Learning vs Statistics



Overview

- Context & Vocabulary
- **Explicit supervised learning**
- Implicit supervised learning

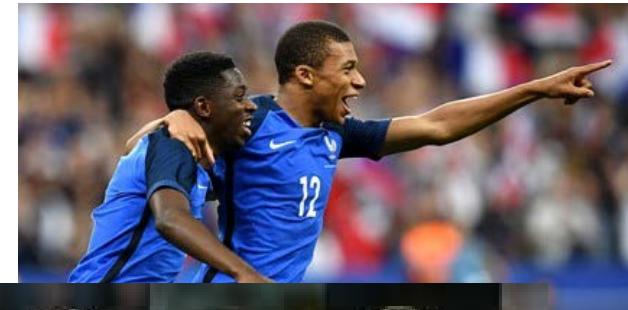
EXPLICIT SUPERVISED LEARNING

Ideas of boosting: Football Bets



If Mbappé and Dembélé are happy together,
French Football team wins.

If Lemar is not injured, French Football
team wins.



If French
France w



If Pogba is happy,
French Football team wins.

How to win?

- Ask to professional gamblers
- Lets assume:
 - That professional gamblers can provide one single decision rule simple and relevant
 - But that face to several games, they can always provide decision rules a little bit better than random
- Can we become rich?

Idea

- Ask heuristics to the expert
 - Gather a set of cases for which these heuristics fail (difficult cases)
 - Ask again the expert to provide heuristics for the difficult cases
 - And so one...
-
- Combine these heuristics
 - expert stands for weak learner



Questions

- How to choose games (i.e. learning examples) at each step?
 - Focus on games (examples) the most “difficult” (the ones on which previous heuristics are the less relevant)
- How to merge heuristics (decision rules) into one single decision rule?
 - Take a weighted vote of all decision rules



Boosting

- boosting = general method to convert several poor decision rules into one very powerful decision rule
- More precisely:
 - Let have a weak learner which can always provide a decision rule (even just little) better than random $\frac{1}{2} - \gamma$,
 - A boosting algorithm can build (theoretically) a global decision rule with an error rate ε as low as desired.
- A theorem of Schapire on “weak learning power” proves that **H** gets a higher relevance than a global decision rule which would have been learnt directly on all training examples.

Probabilistic boosting: AdaBoost

The standard algorithm is **AdaBoost** (*Adaptive Boosting*). 3 main ideas to generalize towards ***probabilistic boosting***:

1. A set of specialized experts and ask them to vote to take a decision.
2. Adaptive weighting of votes by multiplicative update.
3. Modifying example distribution to train each expert, increasing the weights iteratively of examples misclassified at previous iteration.

AdaBoost: the algorithm

- A training set: $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$
- $y_i \in \{-1, +1\}$ label (annotation) of example $\mathbf{x}_i \in S$
- A set of weak learners $\{h_t\}$
- For $t = 0, \dots, T$:
 - Give a weight to every sample in $\{1, \dots, m\}$ regarding its difficulty to be well classified by h_{t-1} : D_t
 - Find the weak decision (“heuristic”): $h_t : S \rightarrow \{-1, +1\}$
with **the smallest error** ε_t on D_t :

$$\varepsilon_t = \Pr_{D_t} [h_t(\mathbf{x}_i) \neq y_i] = \sum_{i: h_t(\mathbf{x}_i) \neq y_i} D_t(i)$$

- Compute the influence/impact of h_t
- Final decision H_{final} = a majority weighted vote of all the h_t



The AdaBoost Algorithm

What **goal** the AdaBoost wants to reach?

Given: $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in X, y_i \in \{-1, +1\}$

Initialization: $D_1(i) = \frac{1}{m}, i = 1, \dots, m$

For $t = 1, \dots, T$

- Find classifier $h_t : X \rightarrow \{-1, +1\}$ which minimizes error wrt D_t , i.e.,

$$h_t = \arg \min_{h_j} \varepsilon_j \text{ where } \varepsilon_j = \sum_{i=1}^m D_t(i) [y_i \neq h_j(x_i)]$$

- Weight classifier: $\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$

- Update distribution: $D_{t+1}(i) = \frac{D_t(i) \exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$, Z_t is for normalization

Output final classifier: $\text{sign} \left(H(x) = \sum_{t=1}^T \alpha_t h_t(x) \right)$

The AdaBoost Algorithm

What **goal** the AdaBoost?

They are goal dependent.

Given: $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in X$

Initialization: $D_1(i) = \frac{1}{m}, i = 1, \dots, m$

For $t = 1, \dots, T$

- Find classifier $h_t : X \rightarrow \{-1, +1\}$ which minimizes error wrt D_t , i.e.,

$$h_t = \arg \min_{h_j} \varepsilon_j \text{ where } \varepsilon_j = \sum_{i=1}^m D_t(i) [y_i \neq h_j(x_i)]$$

- Weight classifier:

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$$

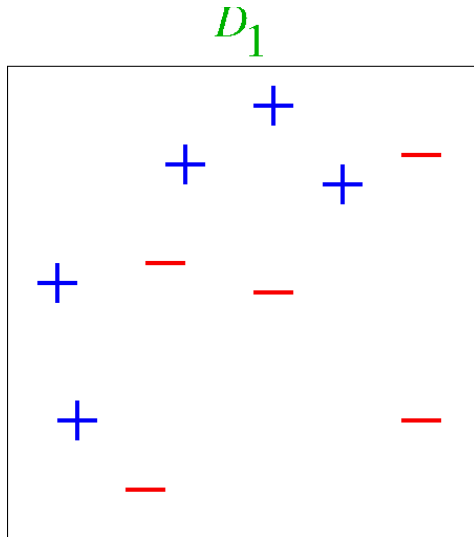
- Update distribution:

$$D_{t+1}(i) = \frac{D_t(i) \exp[-\alpha_t y_i h_t(x_i)]}{Z_t}, \text{ } Z_t \text{ is for normalization}$$

Output final classifier: $\text{sign} \left(H(x) = \sum_{t=1}^T \alpha_t h_t(x) \right)$



Example “Toy”

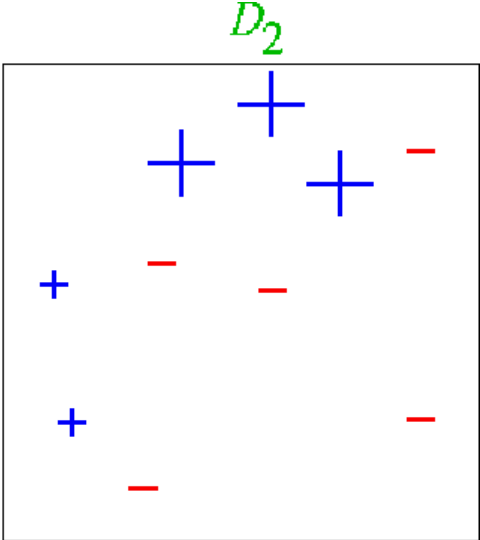
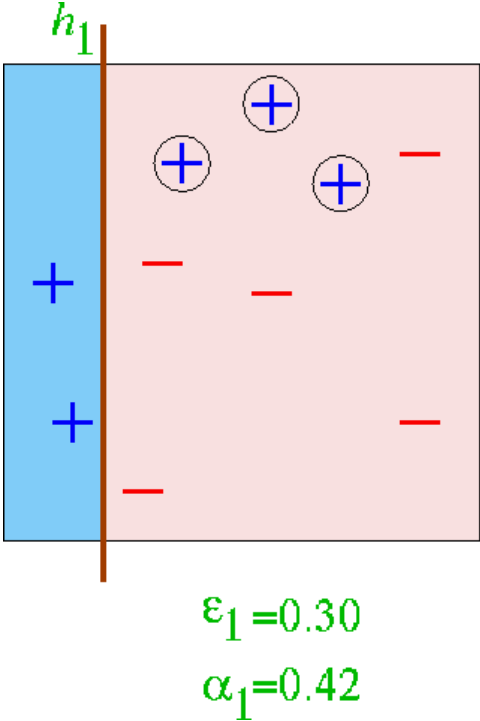


Very popular «Toy» example, 2D points in domain D_1 , to illustrate boosting schema.

(these points can be seen as vectors $\in \mathbb{R}^2$).

D1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Step 1



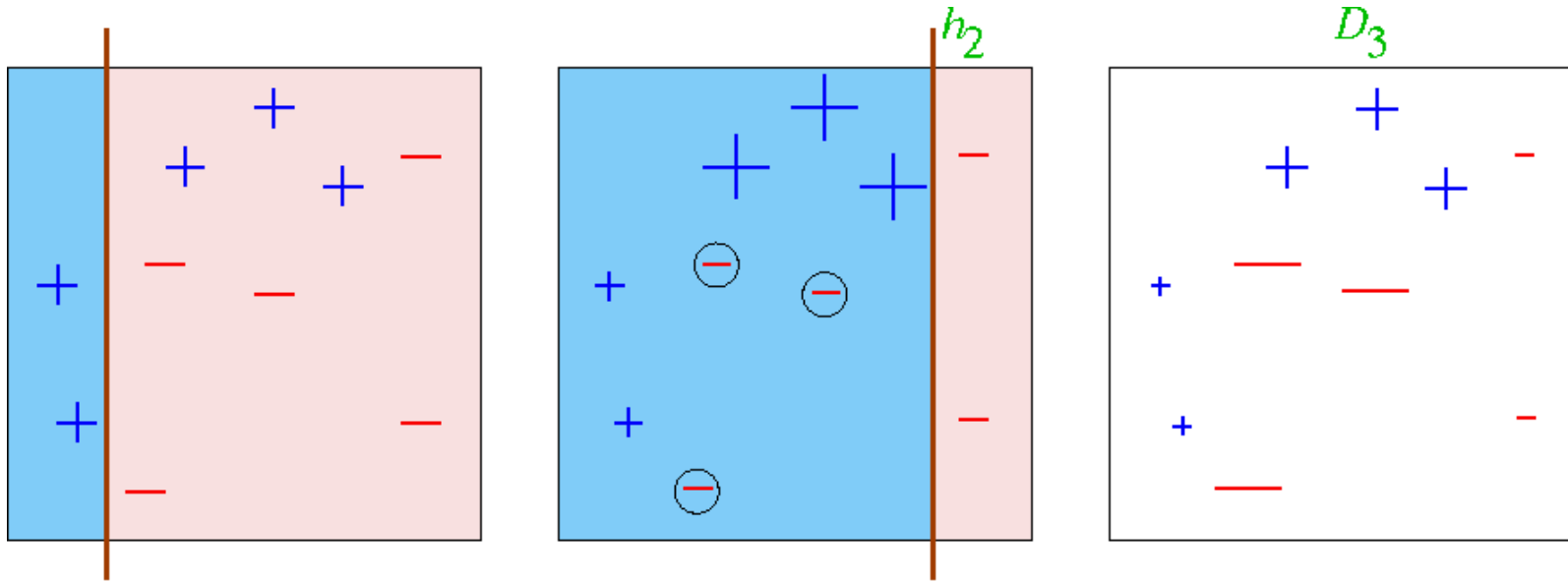
$$\begin{aligned}\alpha_1 &= \frac{1}{2} \ln \left(\frac{1 - \epsilon_1}{\epsilon_1} \right) = \frac{1}{2} \ln \left(\frac{1 - 0.3}{0.3} \right) \\ &= \frac{1}{2} \ln \left(\frac{0.7}{0.3} \right) = \ln \left(\sqrt{\frac{7}{3}} \right)\end{aligned}$$

$$D'_1 = \begin{cases} 0.1 e^{-\alpha_1} = 0.1 \sqrt{\frac{3}{7}} = 0.065 \\ 0.1 e^{\alpha_1} = 0.1 \sqrt{\frac{7}{3}} = 0.152 \end{cases}$$

$$Z_1 = 3 \times 0.152 + 7 \times 0.065 = 0.911$$

$D_2 = D'_1 / Z_1$	0.167	0.167	0.167	0.071	0.071	0.071	0.071	0.071	0.071	0.071
--------------------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Step 2

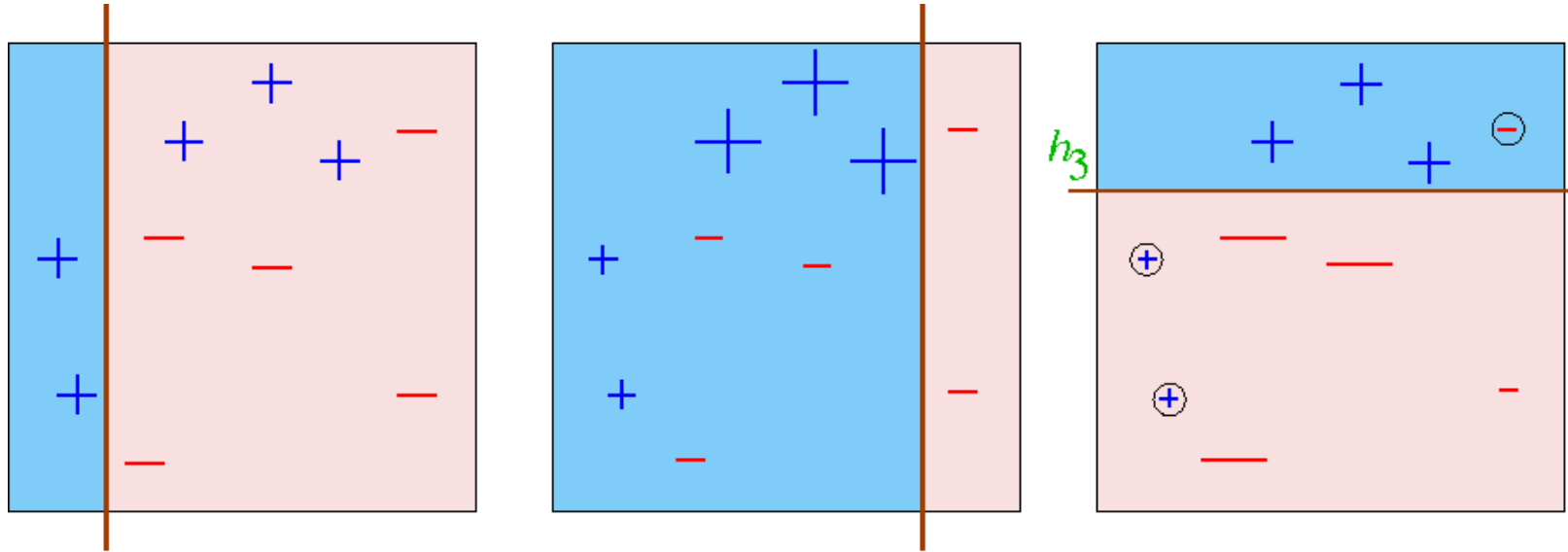


$$D'_2 = \begin{cases} 0.17 e^{-\alpha_2} = 0.0876 \\ 0.07 e^{-\alpha_2} = 0.036 \\ 0.07 e^{\alpha_2} = 0.1357 \end{cases} \quad \begin{matrix} \epsilon_2 = 0.21 \\ \alpha_2 = 0.65 \end{matrix} \quad Z_2 = 3 \times 0.0876 + 4 \times 0.036 + 3 \times 0.1357 = 0.814$$

$D_3 = D'_2 / Z_2$	0.107	0.107	0.107	0.044	0.044	0.044	0.044	0.166	0.166	0.166
--------------------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------



Step 3



$$\epsilon_3 = 0.14$$

$$\alpha_3 = 0.92$$



Final decision

$$H_{\text{final}} = \text{sign} \left(0.42 \begin{array}{|c|} \hline \text{blue} \\ \hline \end{array} + 0.65 \begin{array}{|c|} \hline \text{blue} \\ \hline \end{array} + 0.92 \begin{array}{|c|} \hline \text{blue} \\ \hline \end{array} \right)$$

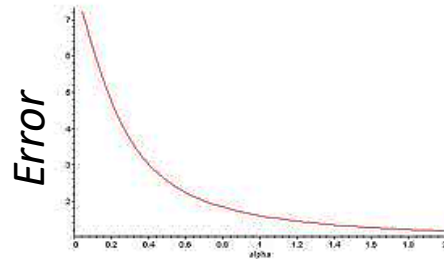
$$= \begin{array}{|c|c|c|} \hline \text{blue} & \text{blue} & \text{red} \\ \hline \end{array}$$

The final decision is a 2D plot with two vertical lines at x=0.42 and x=0.65, and one horizontal line at y=0.92. The regions are colored blue or red based on the sign of the weighted sum of the three weak classifiers. The regions are labeled with '+' for blue and '-' for red.

Error of generalization for AdaBoost

- Error of generalization of H can be bounded by:

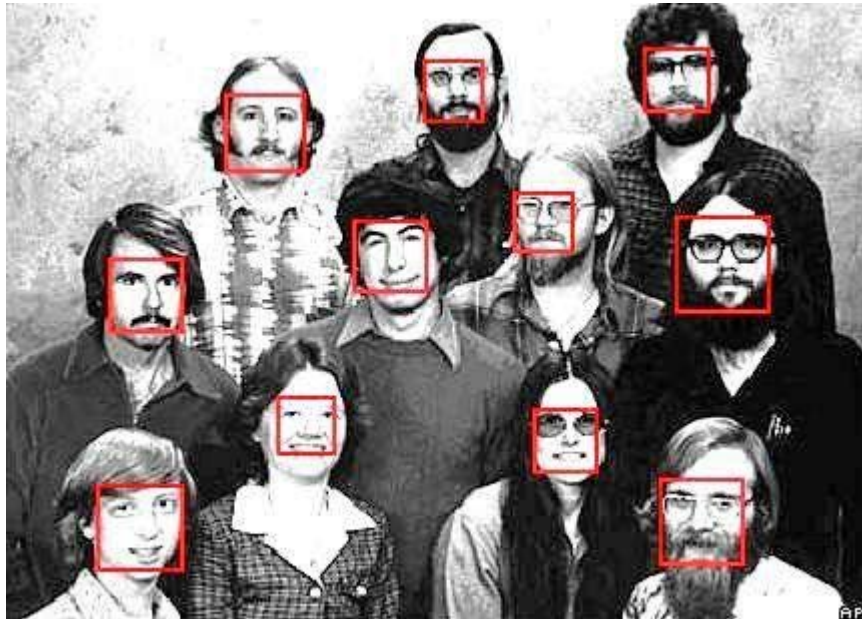
$$E_{\text{Real}}(H_T) = E_{\text{Empirical}}(H_T) + O\left(\sqrt{\frac{T \cdot d}{m}}\right)$$



- where
 - T is the number of boosting iterations
 - m the number of training examples
 - d the dimension of H_T space (“weak learner complexity”)



The Task of Face Detection



Many slides adapted from P. Viola

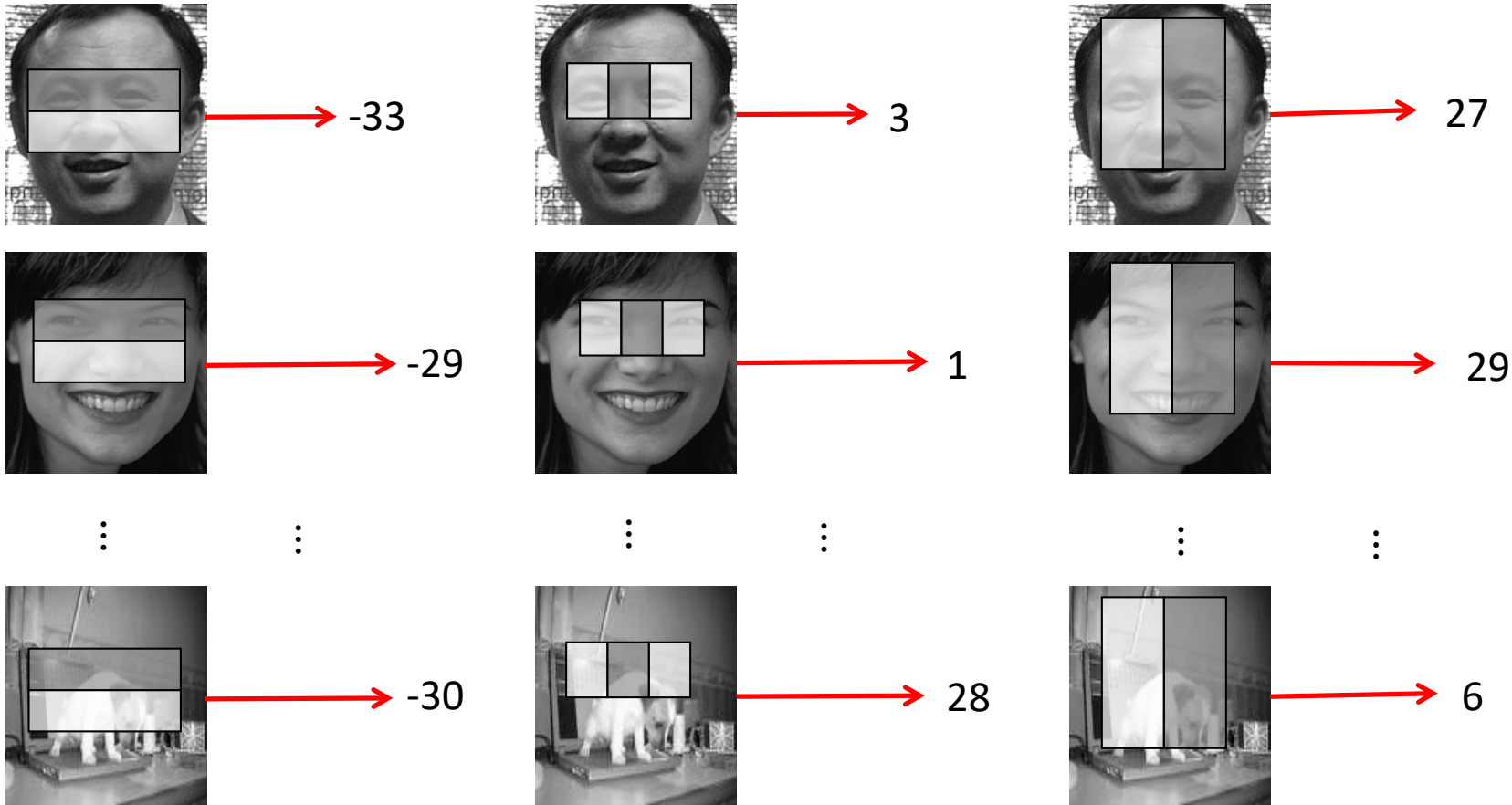
Basic Idea

- Slide a window across image and evaluate a face model at every location.



Image Features

$$\text{Feature Value} = \sum (\text{Pixel in white area}) - \sum (\text{Pixel in black area})$$

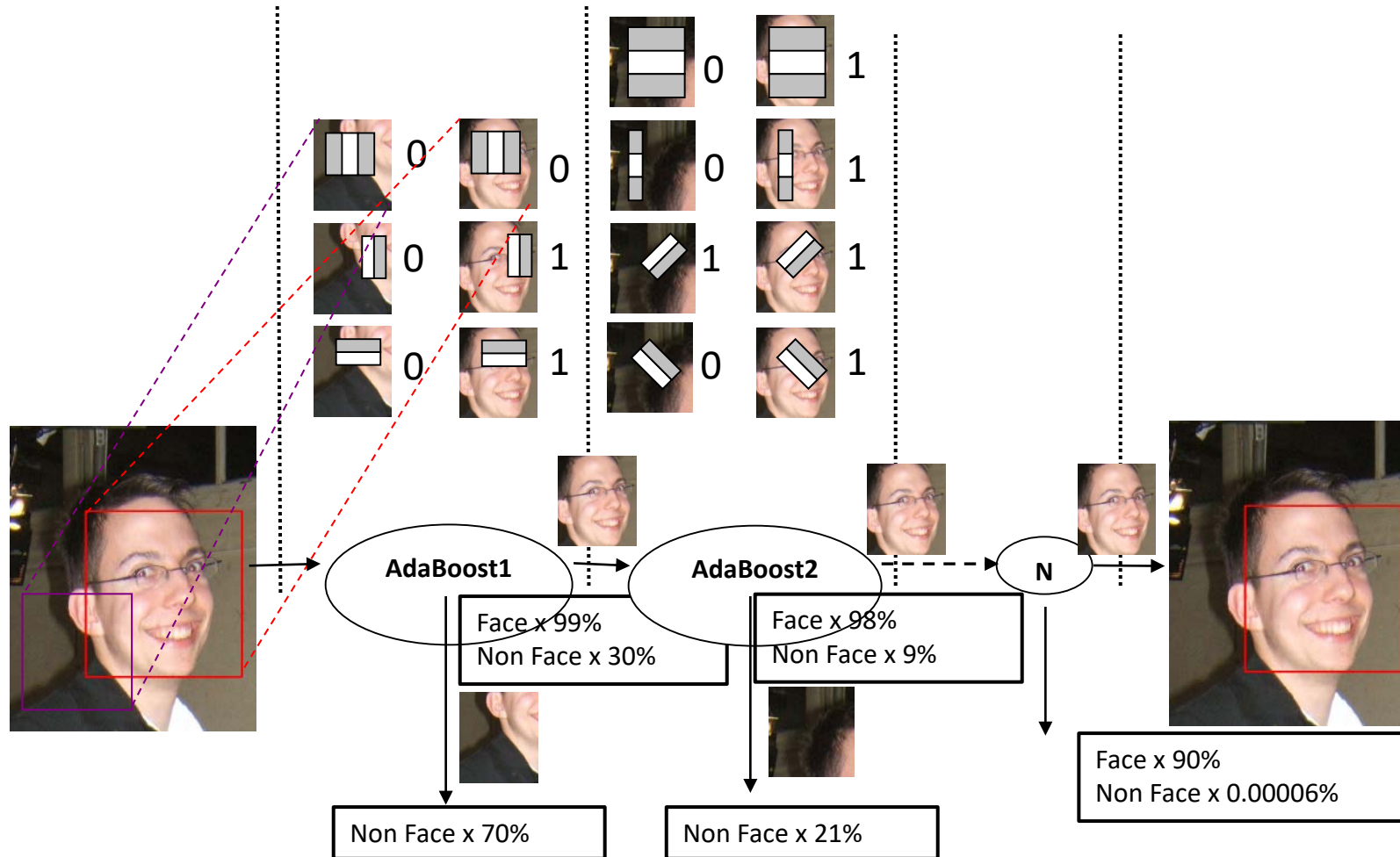


$$h_1\left(\begin{array}{|c|c|}\hline \text{white} & \text{black} \\ \hline\end{array}\right) = \begin{cases} 1 & \text{if } \begin{array}{|c|c|}\hline \text{white} & \text{black} \\ \hline\end{array} < -29 \\ 0 & \text{otherwise} \end{cases}$$

$$h_2\left(\begin{array}{|c|c|c|}\hline \text{white} & \text{black} & \text{white} \\ \hline\end{array}\right) = \begin{cases} 1 & \text{if } \begin{array}{|c|c|c|}\hline \text{white} & \text{black} & \text{white} \\ \hline\end{array} < 26 \\ 0 & \text{otherwise} \end{cases}$$

$$h_3\left(\begin{array}{|c|c|}\hline \text{white} & \text{black} \\ \hline\end{array}\right) = \begin{cases} 1 & \text{if } \begin{array}{|c|c|}\hline \text{white} & \text{black} \\ \hline\end{array} > 11 \\ 0 & \text{otherwise} \end{cases}$$

AdaBoost Cascade Principle



The Implemented System

- Training Data
 - 5000 faces
 - All frontal, rescaled to 24x24 pixels
 - 300 million non-faces sub-windows
 - 9500 non-face images
 - Faces are normalized
 - Scale, translation
- Many variations
 - Across individuals
 - Illumination
 - Pose





- Fixed images
- Video sequence



Frontal face



Left profile
face



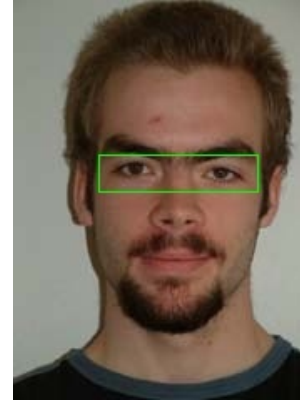
Right profile
face





Extension

- Fast and robust
- Other descriptors
- Other cascades (rotation...)
- Eye detection, Hand detection, Body detection...



Overview

- Context & Vocabulary
- Explicit supervised classification
- **Implicit supervised classification**
 - **Multi-Layer Perceptron**
 - Deep Learning

IMPLICIT SUPERVISED LEARNING

Thomas Cover's Theorem (1965)

“The Blessing of dimensionality”

Cover's theorem states: A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space.

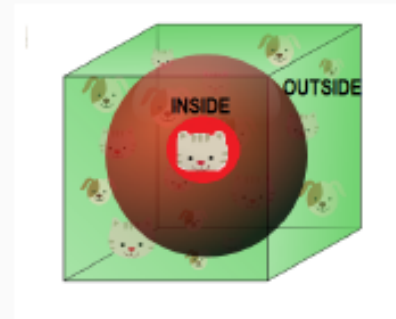
(repeated sequence of Bernoulli trials)



The curse of dimensionality [Bellman, 1956]

- Euclidian distance is not relevant in high dimension: $d \geq 10$
 - ① look at the examples at distance at most r
 - ② the hypersphere volume is too small: practically empty of examples

$$\frac{\text{volume of the sphere of radial } r}{\text{hypersphere of } 2r \text{ width}} \rightarrow_{d \rightarrow \infty} 0$$



- ③ need a number of examples exponential in d

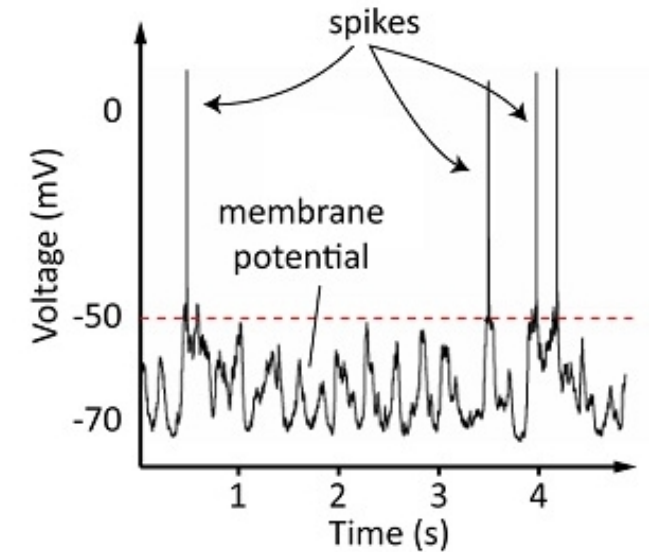
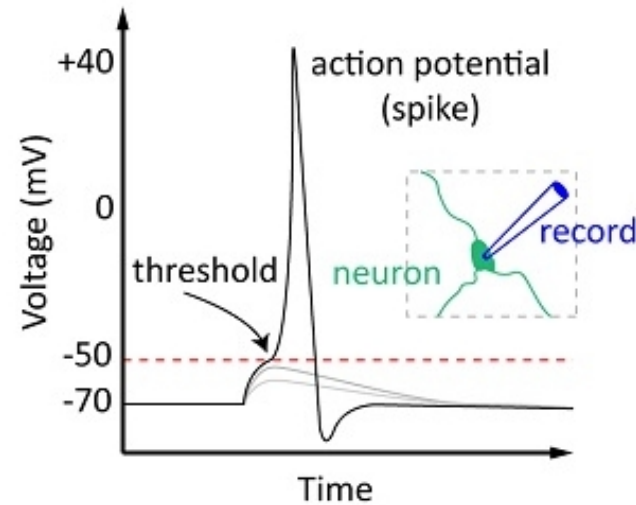
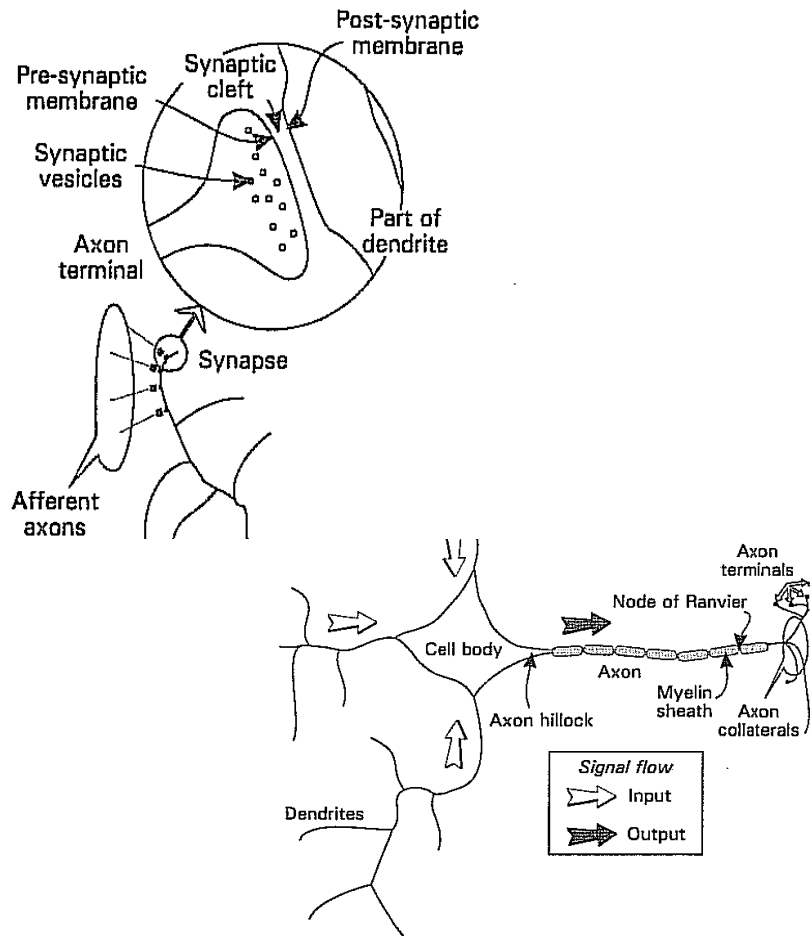
Remark

Specific care for data representation

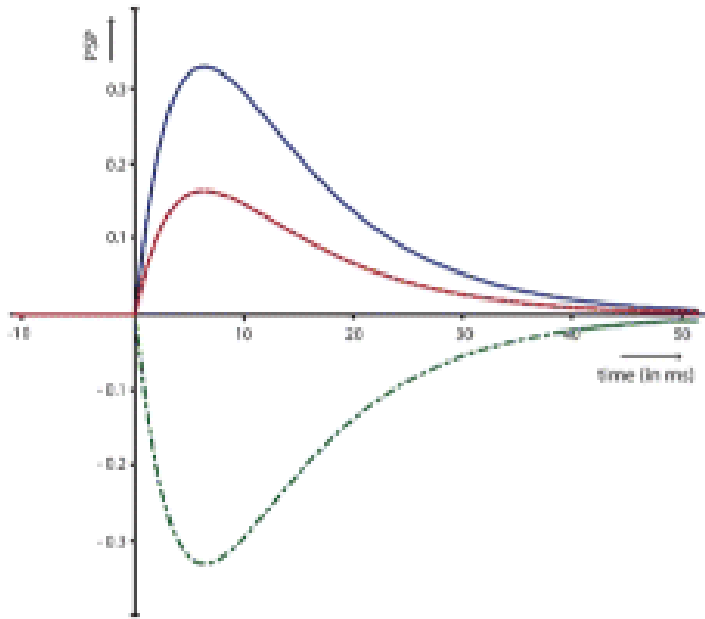
MULTI-LAYER PERCEPTRON

First, biological neurons

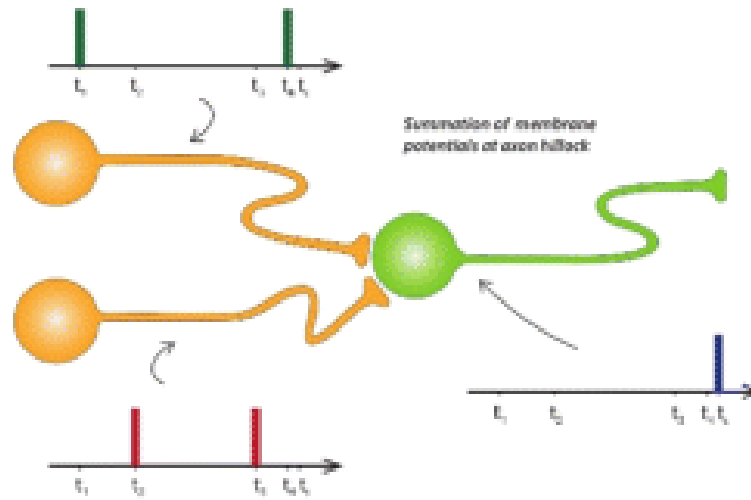
- Before we study artificial neurons, let's look at a biological neuron



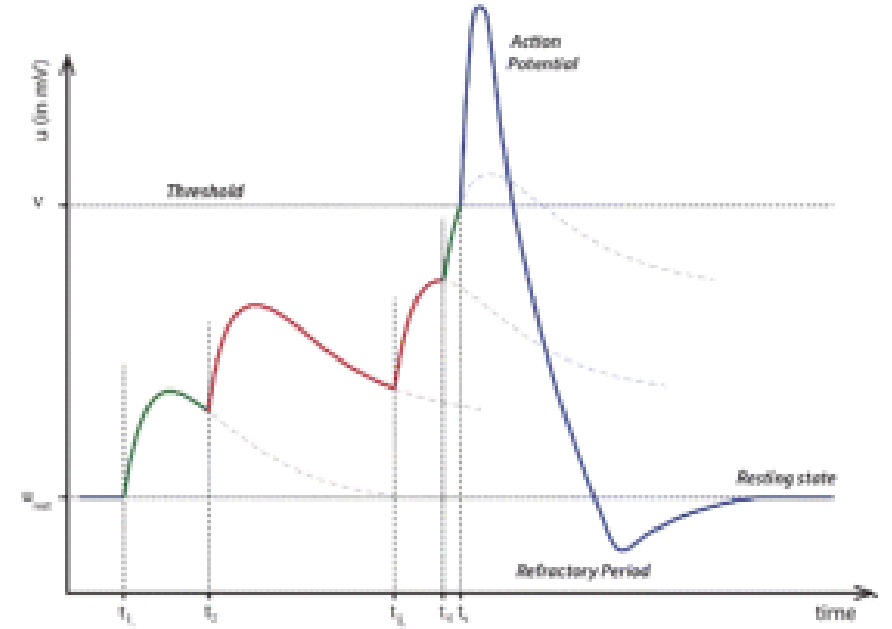
First, biological neurons



A



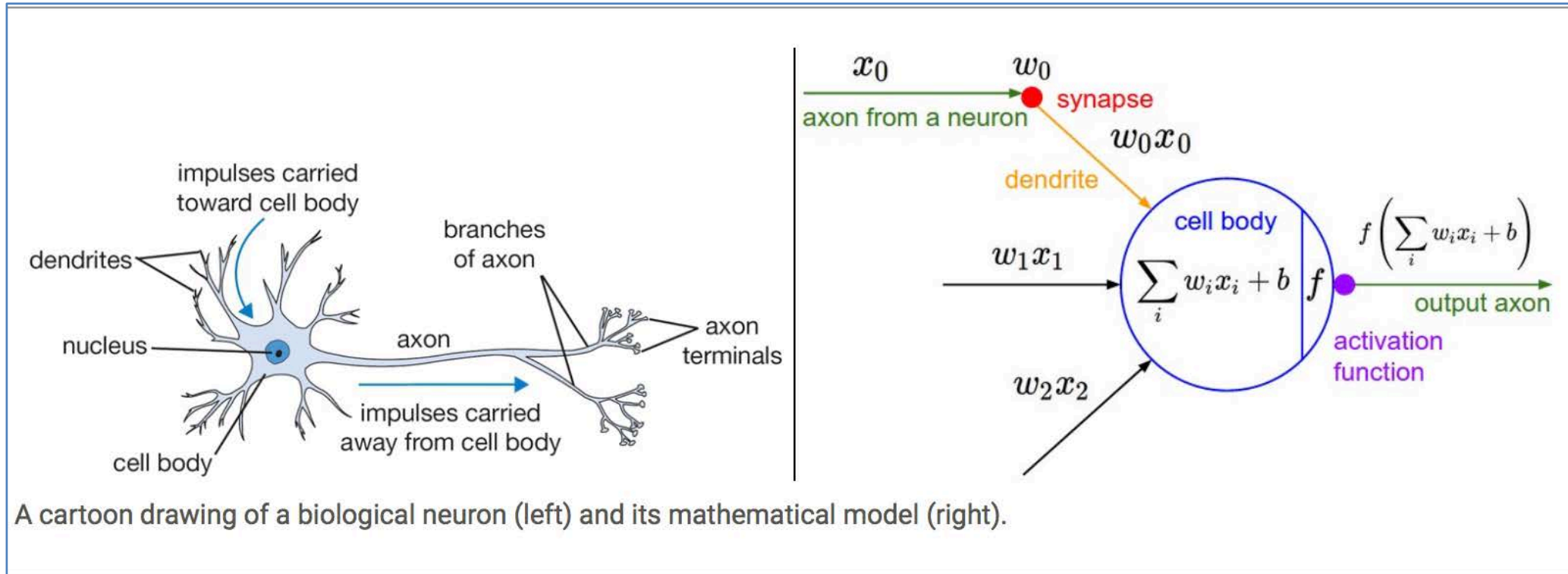
B



C

Postsynaptic potential function with weight dependency, as a function of time (ms) and weight value, being excitatory in case of red and blue lines, and inhibitory in case of a green line.

Then, artificial neurons

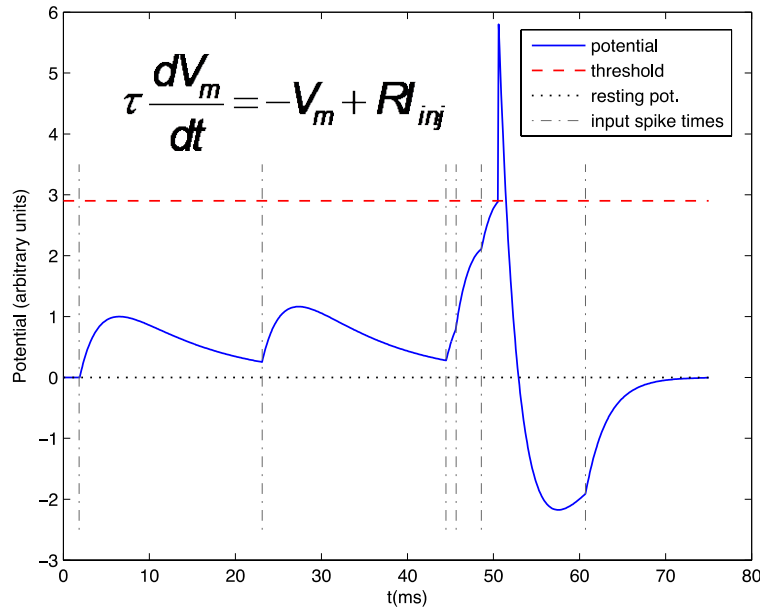


Pitts & McCulloch (1943), binary inputs & activation function f is a thresholding

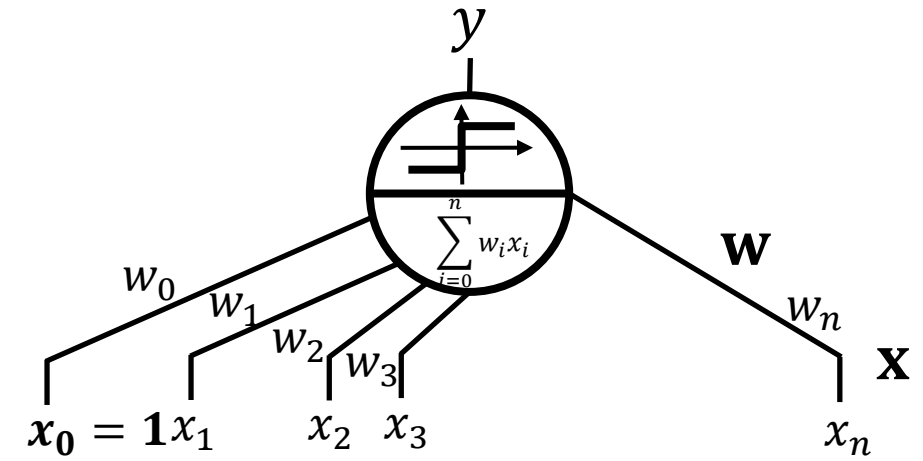
Rosenblatt (1956), real inputs & activation function f is a thresholding

Artificial neuron vs biology

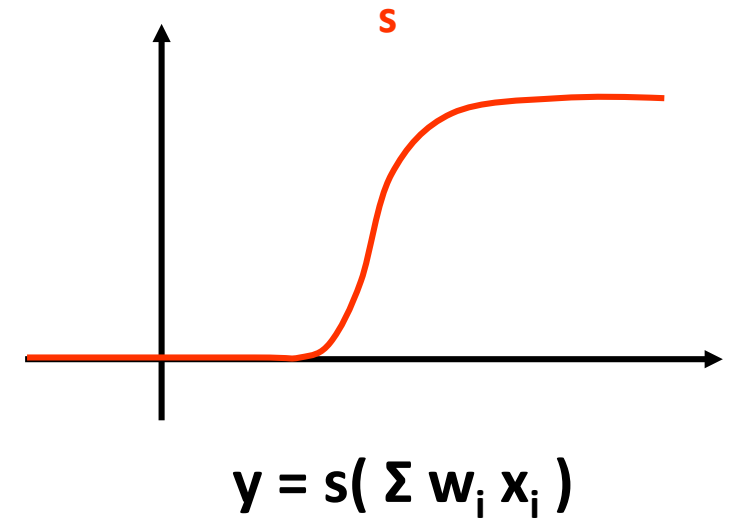
Spike-based description



Gradient descent: KO

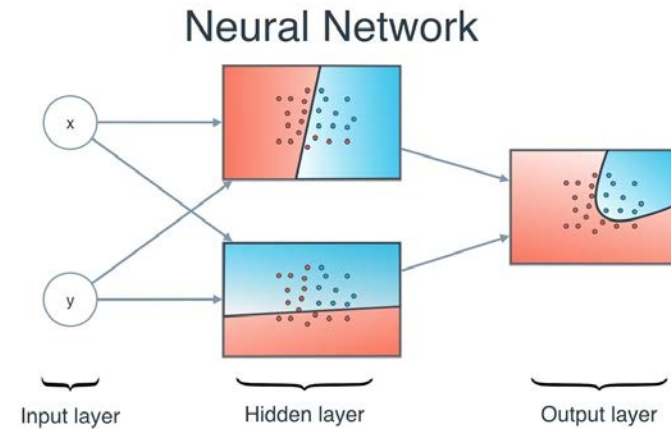
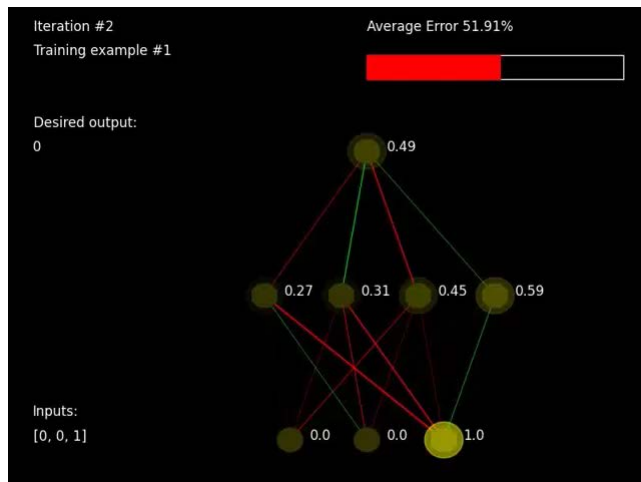
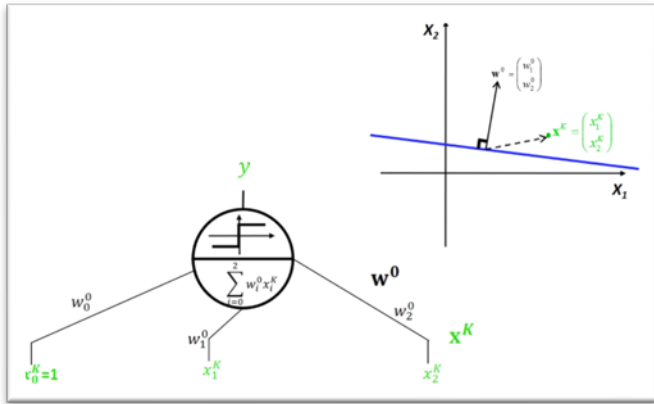


Rate-based description *Steady regime*



Gradient descent: OK

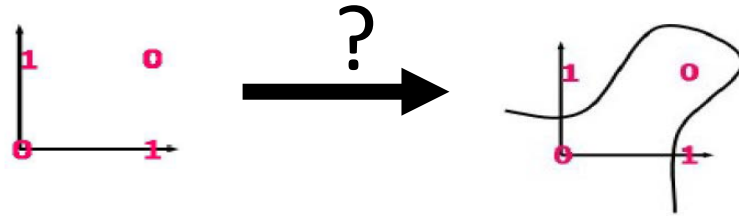
From perceptron to network



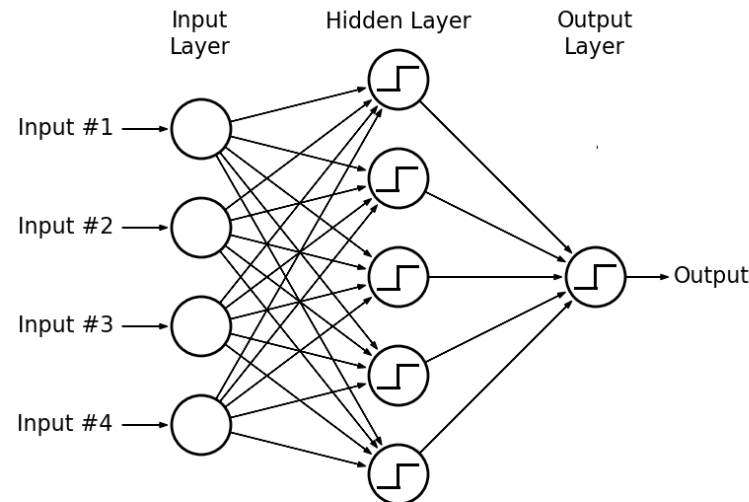
@tachyeonz: A friendly introduction to neural networks and deep learning.

Single Perceptron Unit

- **Perceptron** only learns linear function [Minsky and Papert, 1969]

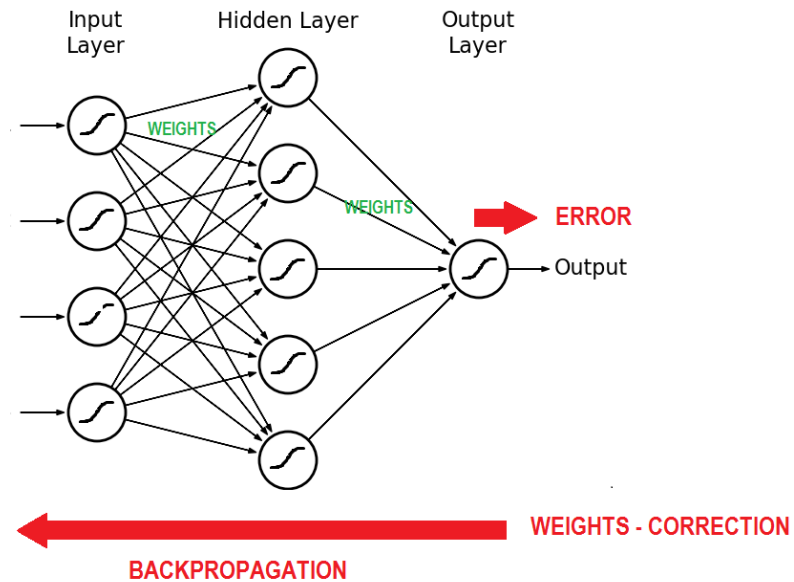
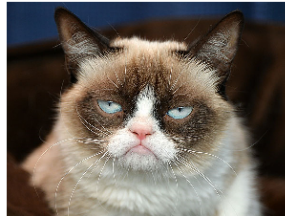


- Non-linear function needs layer(s) of neurons → Neural Network
- Neural Network = input layer + hidden layer(s) + output layer



Multi-Layer Perceptron

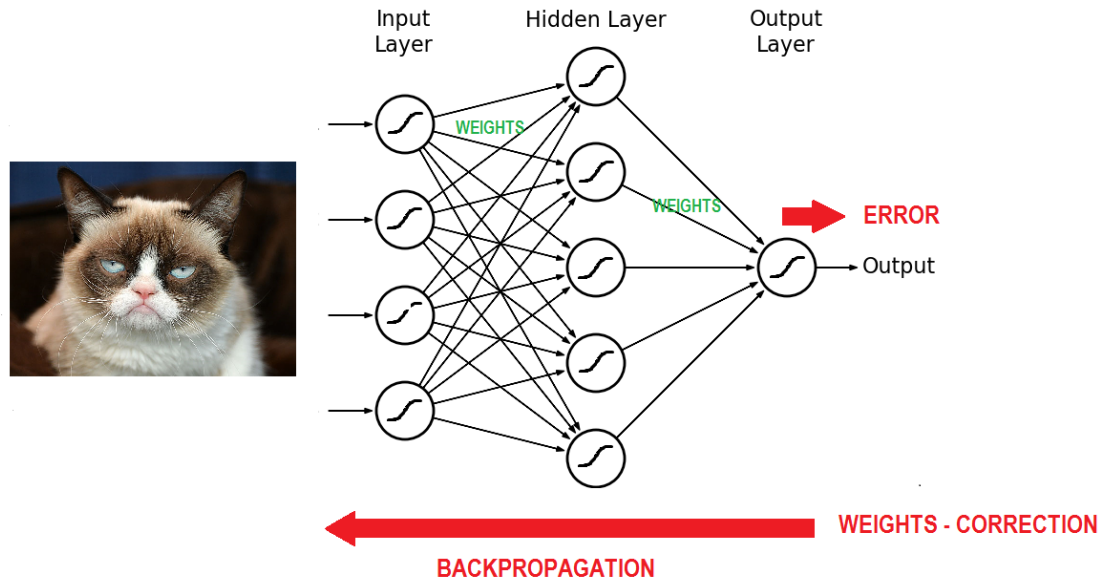
- Training a neural network [Rumelhart et al. / Yann Le Cun et al. 1985]
- **Unknown parameters: weights on the synapses**
- Minimizing a cost function: some metric between the predicted output and the given output



- Step function: non-continuous functions are replaced by a continuous non-linear ones

Multi-Layer Perceptron

- Minimizing a cost function: some metric between the predicted output and the given output



- Equation for a network of 3 neurons (i.e. 3 perceptrons):

$$y = s(w_{13}s(w_{11}x_1 + w_{21}x_2 + w_{01}) + w_{23}s(w_{12}x_1 + w_{22}x_2 + w_{02}) + w_{03})$$



Multi-Layer Perceptron

Theorem [Cybenko, 1989]

- A neural network with one single hidden layer is a **universal approximator**: it can represent any continuous function on compact subsets of \mathbf{R}^n
- 2 layers is enough ... theoretically:
“...networks with one internal layer and an arbitrary continuous sigmoidal function can approximate continuous functions with arbitrary precision providing that no constraints are placed on the number of nodes or the size of the weights”
- But ***no efficient learning rule*** is known and the size of the hidden layer is ***exponential*** with the complexity of the problem (which is unknown beforehand) to get an error ε , the layer must be infinite for an error 0.



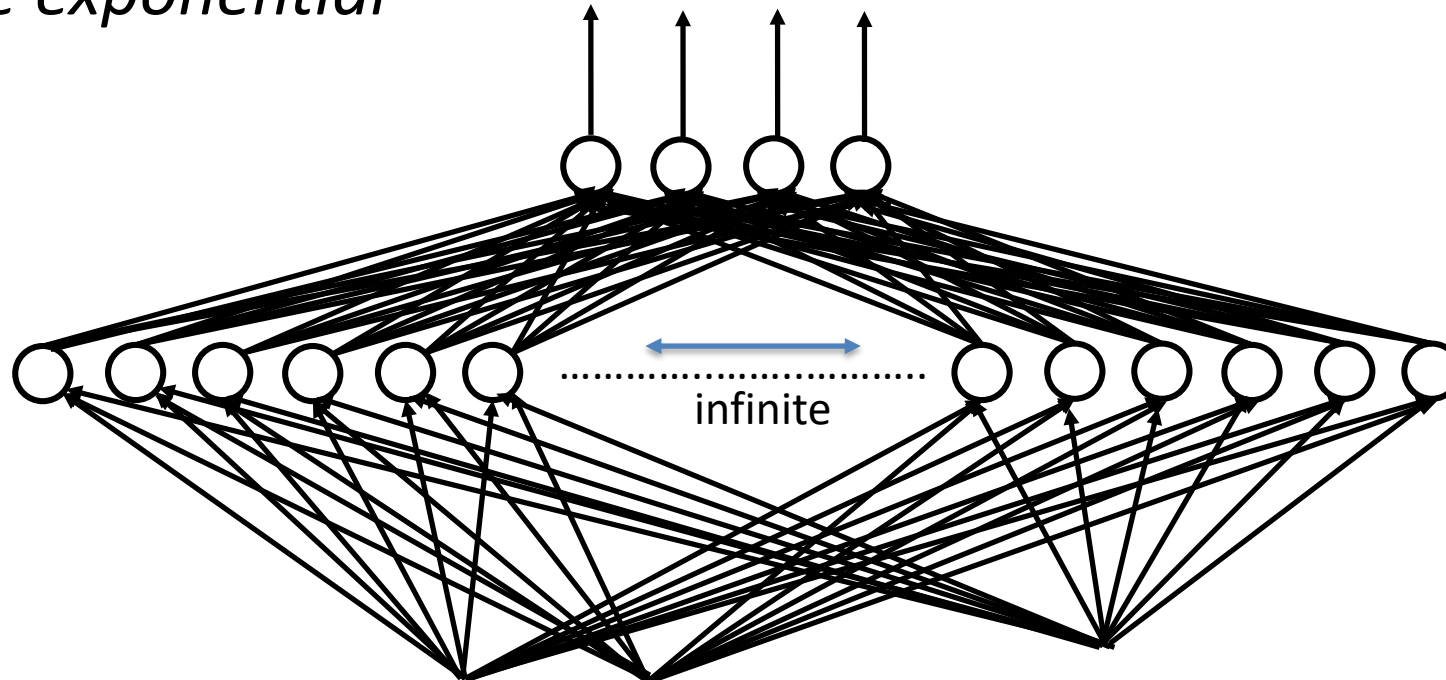
Overview

- Context & Vocabulary
- Explicit supervised classification
- **Implicit supervised classification**
 - Multi-Layer Perceptron
 - **Deep Learning**

DEEP LEARNING

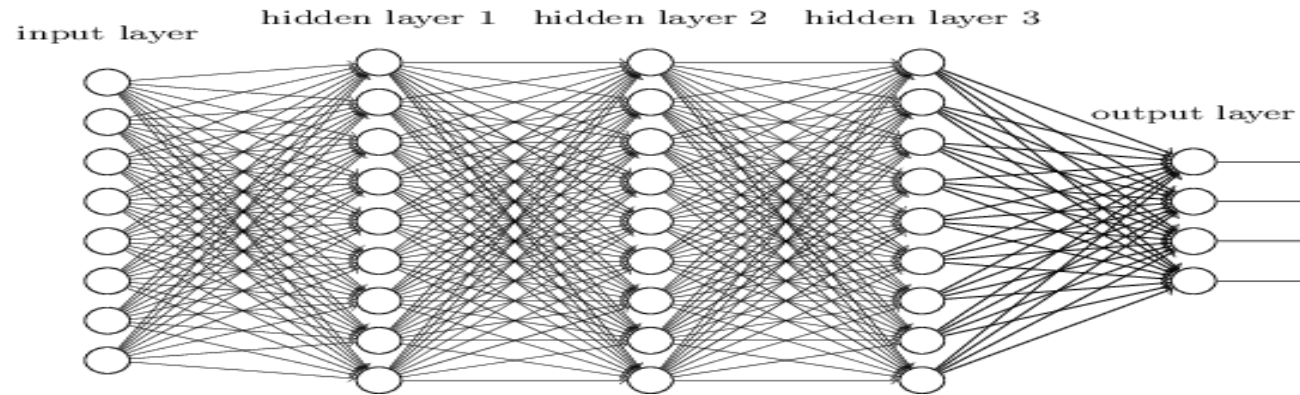
Deep representation origins

- **Theorem Cybenko (1989)** *A neural network with one single hidden layer is a universal “approximator”, it can represent any continuous function on compact subsets of $\mathbf{R}^n \Rightarrow 2$ layers are enough...but hidden layer size may be exponential*



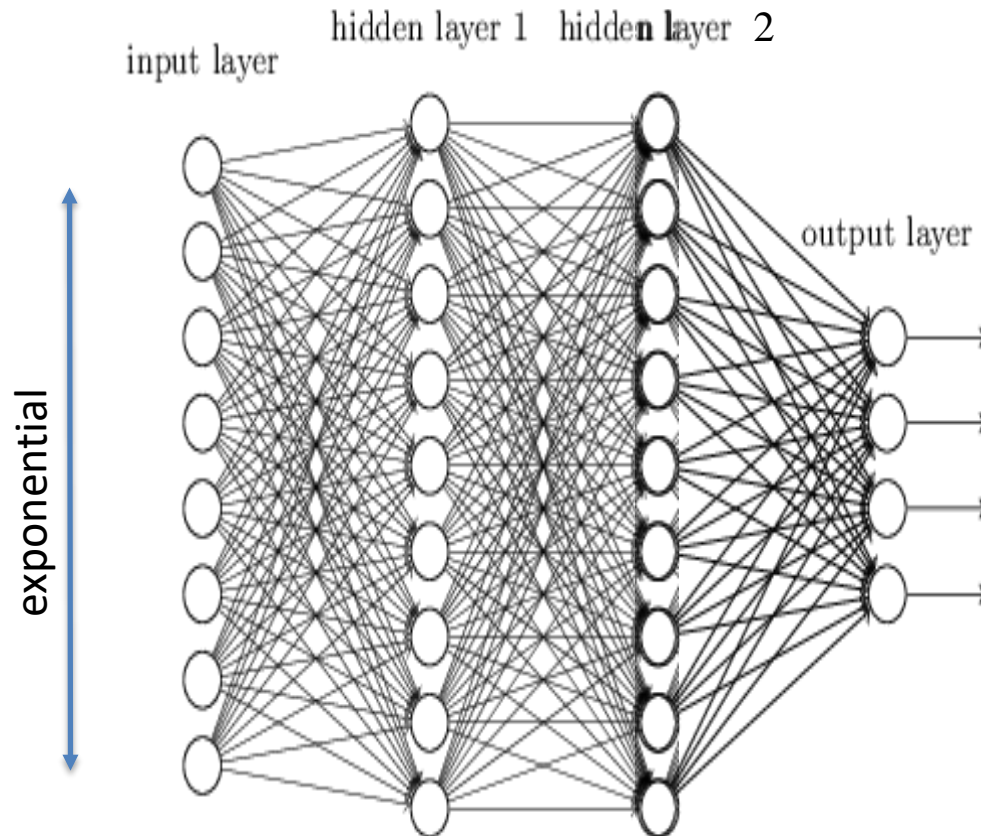
Deep representation origins

- **Theorem Hastad (1986), Bengio et al. (2007)** Functions representable compactly with k layers may require exponentially size with $k-1$ layers



Deep representation origins

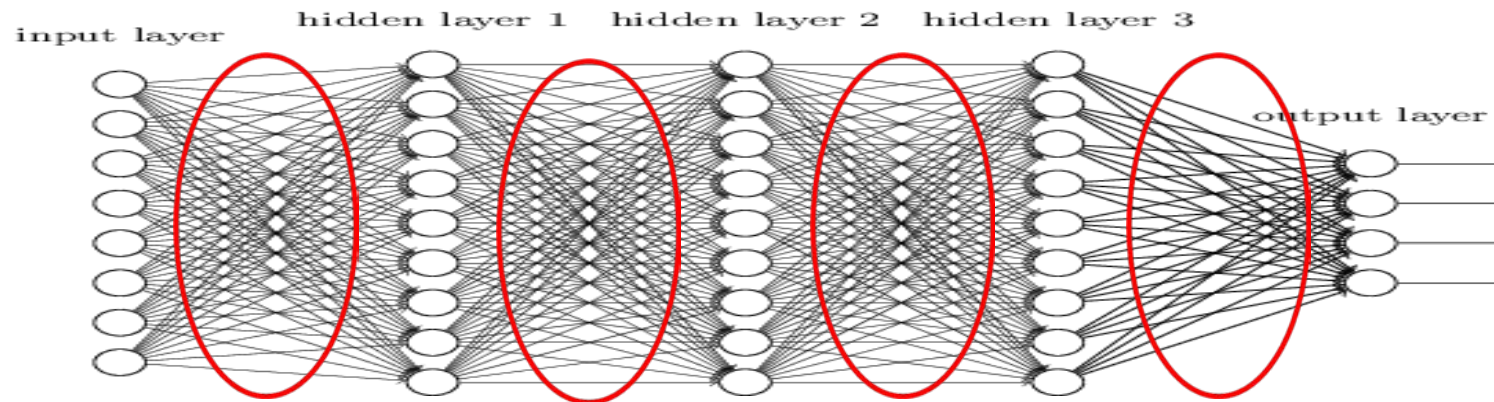
- **Theorem Hastad (1986), Bengio et al. (2007)** Functions representable compactly with k layers may require exponentially size with $k-1$ layers



Structure the network?

- Can we put any structure reducing the space of exploration and providing useful properties (invariance, robustness...)?

$$y = s(w_{13}s(w_{11}x_1 + w_{21}x_2 + w_{01}) + w_{23}s(w_{12}x_1 + w_{22}x_2 + w_{02}) + w_{03})$$



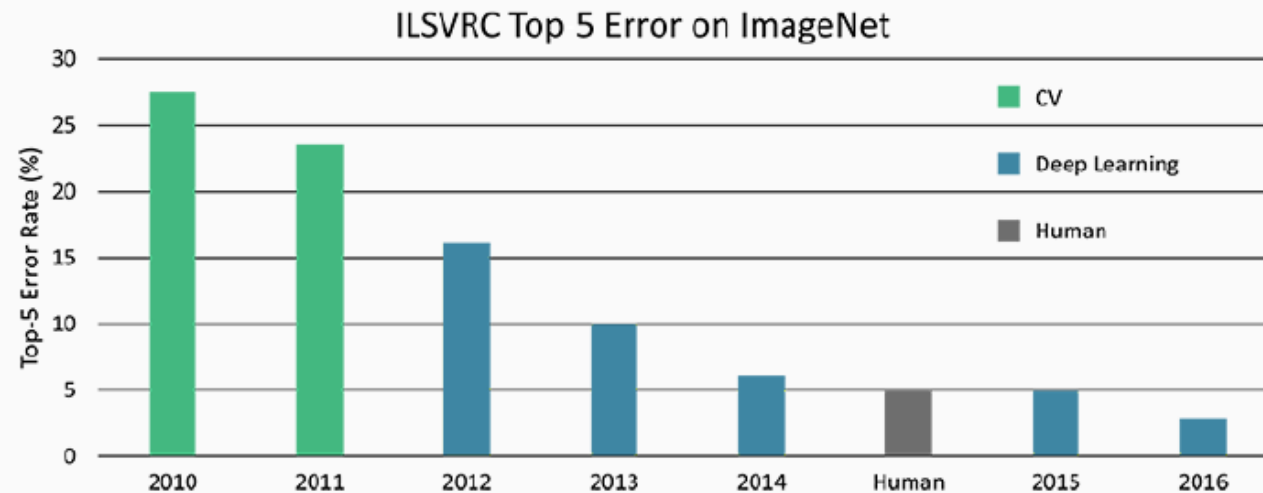
Enabling factors

- Why do it now ? Before 2006, training deep networks was unsuccessful because of practical aspects
 - faster CPU's
 - parallel CPU architectures
 - advent of GPU computing
- Hinton, Osindero & Teh « A Fast Learning Algorithm for Deep Belief Nets », *Neural Computation*, 2006
- Bengio, Lamblin, Popovici, Larochelle « Greedy Layer-Wise Training of Deep Networks », *NIPS'2006*
- Ranzato, Poultney, Chopra, LeCun « Efficient Learning of Sparse Representations with an Energy-Based Model », *NIPS'2006*
- Results...
 - 2009, sound, interspeech + ~24%
 - 2011, text, + ~15% without linguistic at all
 - 2012, images, ImageNet + ~20%

CONVOLUTIONAL NEURAL NETWORKS (AKA CNN, CONVNET)

Convolutional neural network

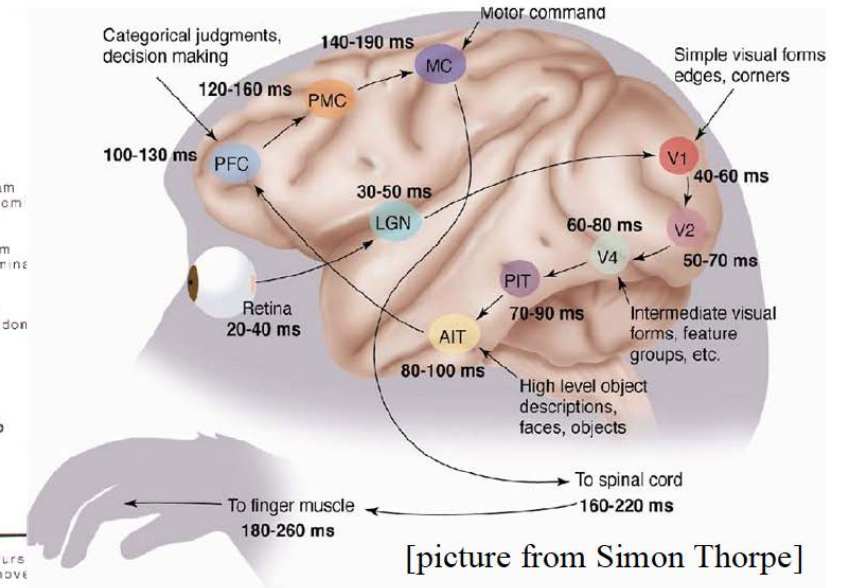
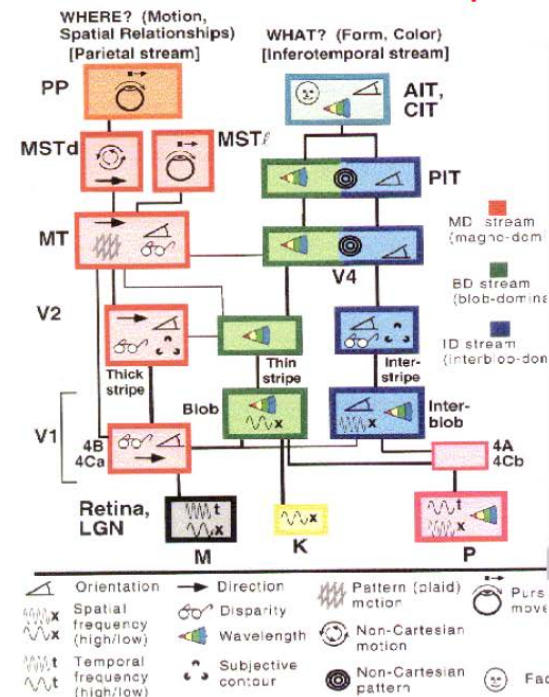
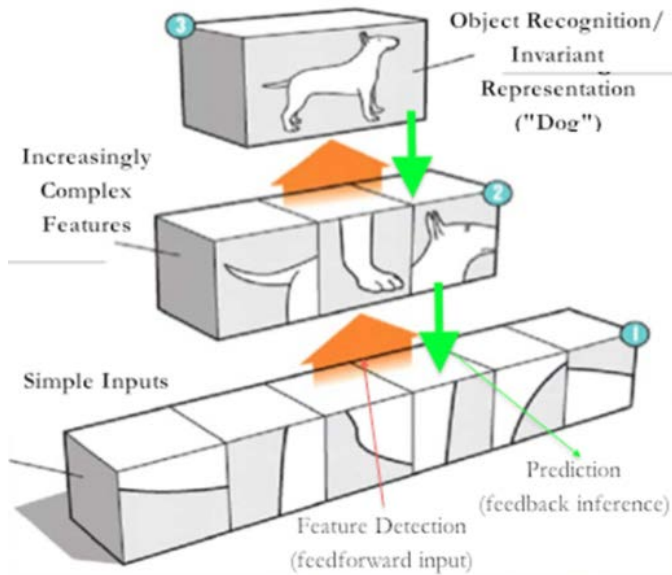
- Deep Networks are as good as humans at recognition, identification...



How much does a deep network understands those tasks?

Deep representation by CNN

- The ventral (recognition) pathway in the visual cortex has multiple stages
- Retina - LGN - V1 - V2 - V4 - PIT - AIT
- Lots of intermediate representations



[Gallant & Van Essen]

[picture from Simon Thorpe]

Convolution

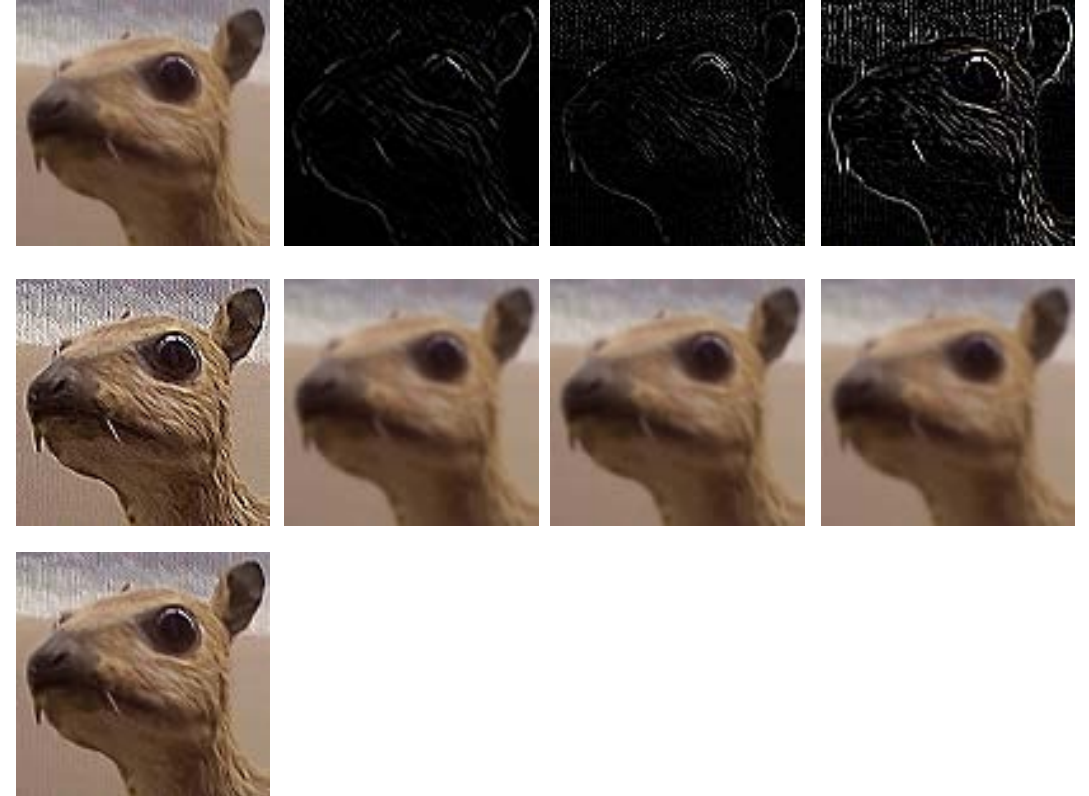
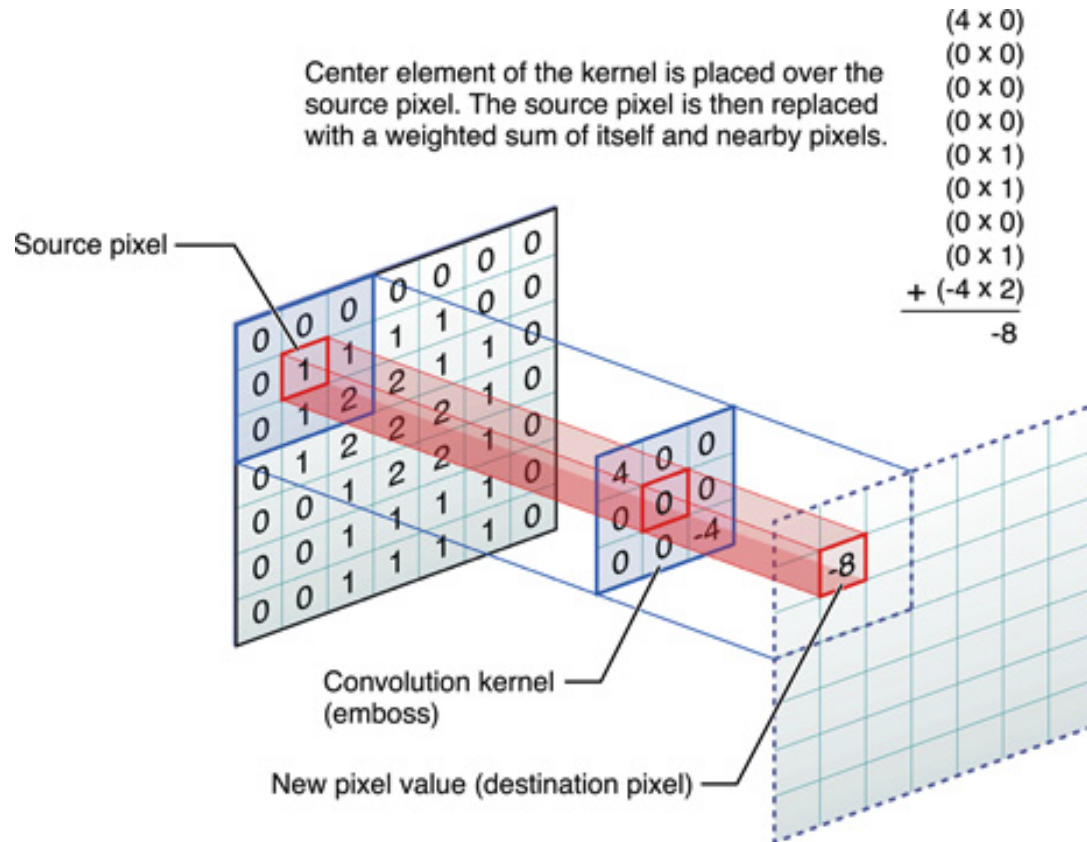
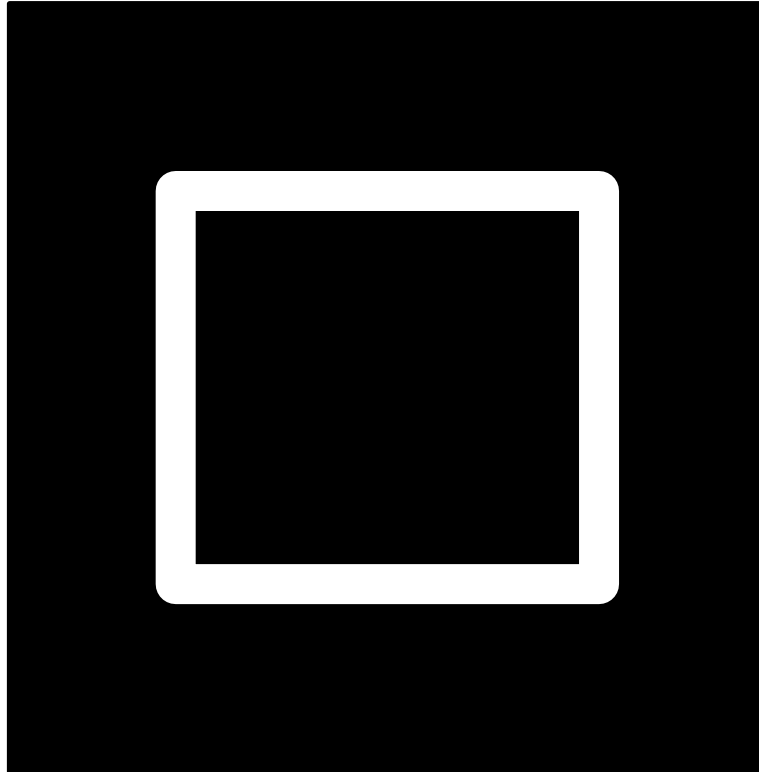


Image Convolution



Image

-1
0
1

Filter to extract
horizontal edges



Image Convolution

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Image

-1
0
1

Filter to extract
horizontal edges

Image Convolution

New Image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	1							

0
0
1

*

-1
0
1

Filter

$$= (0 \times -1) + (0 \times 0) + (1 \times 1) = 1$$

Image Convolution

New Image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	1	1						

0
0
1

*

-1
0
1

Filter

$$= (0 \times -1) + (0 \times 0) + (1 \times 1) = 1$$

Image Convolution

New Image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1					

0
0
1

*

-1
0
1

Filter

$$= (0 \times -1) + (0 \times 0) + (1 \times 1) = 1$$

Image Convolution

New Image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1				

0
0
1

*

-1
0
1

Filter

$$= (0 \times -1) + (0 \times 0) + (1 \times 1) = 1$$



Image Convolution

New Image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	-1	-1	-1	-1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	-1	0	0	0	0	-1	0	0
0	0	-1	-1	-1	-1	-1	-1	0	0
0	0	0	0	0	0	0	0	0	0

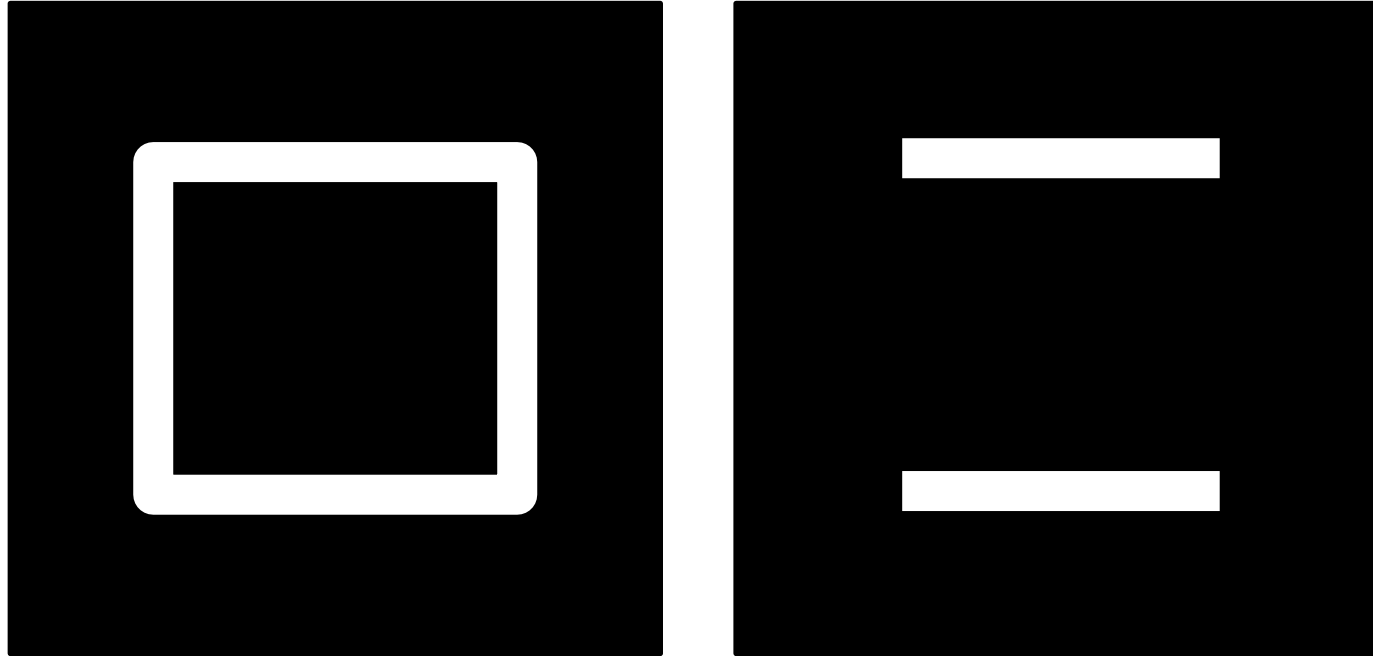


Horizontal Edge Detector



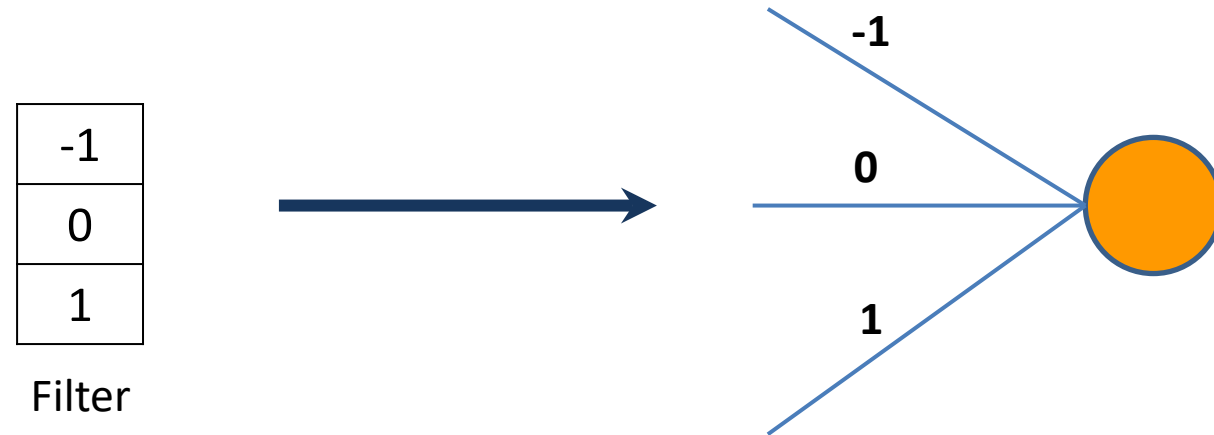
Image Convolution

New Image

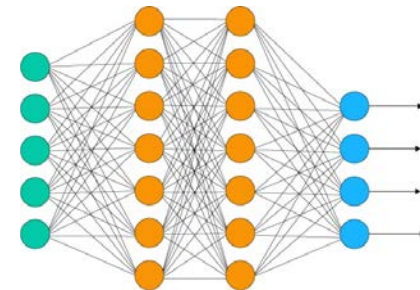


Horizontal Edge Detector

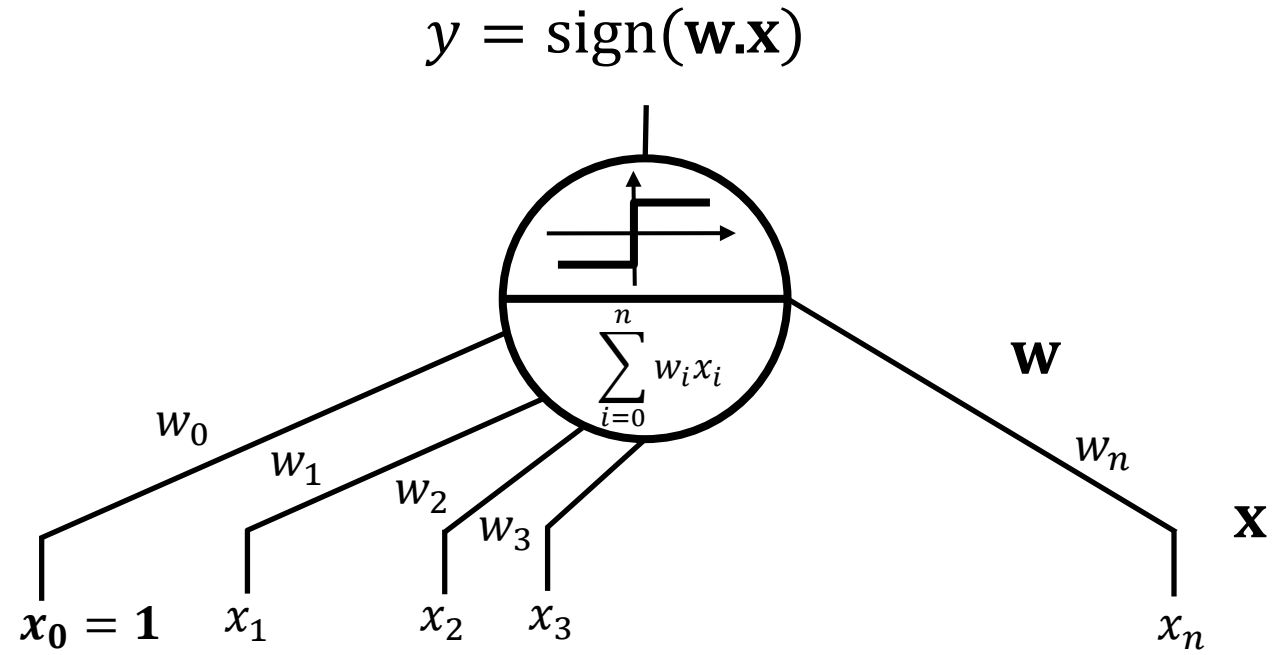
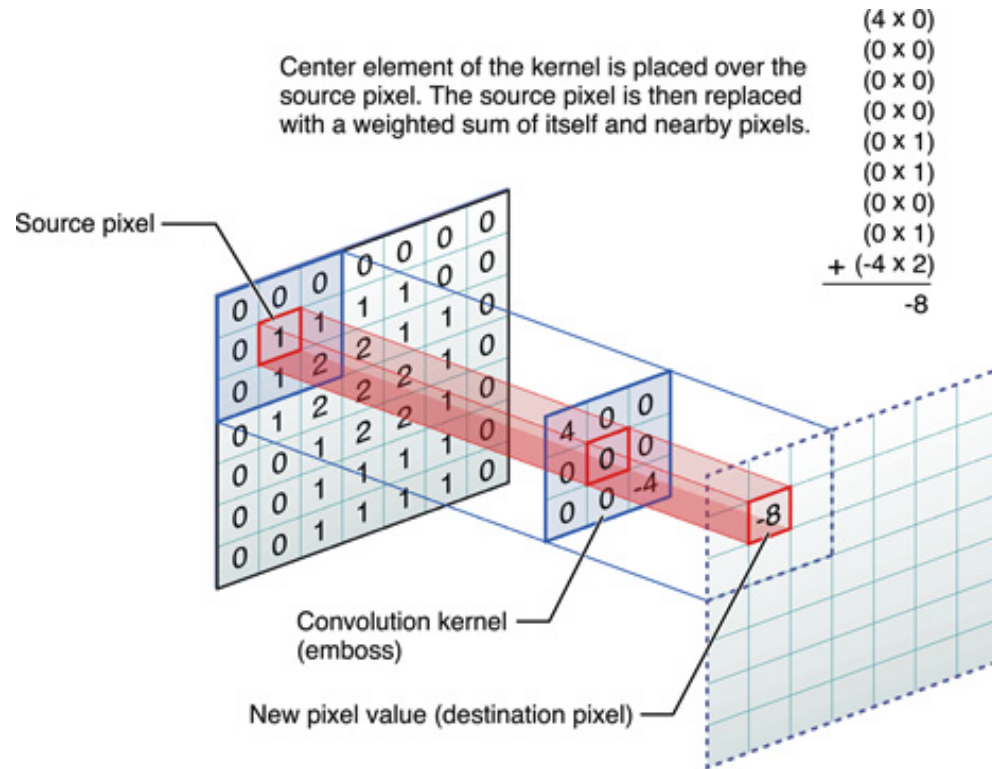
Filter in CNN



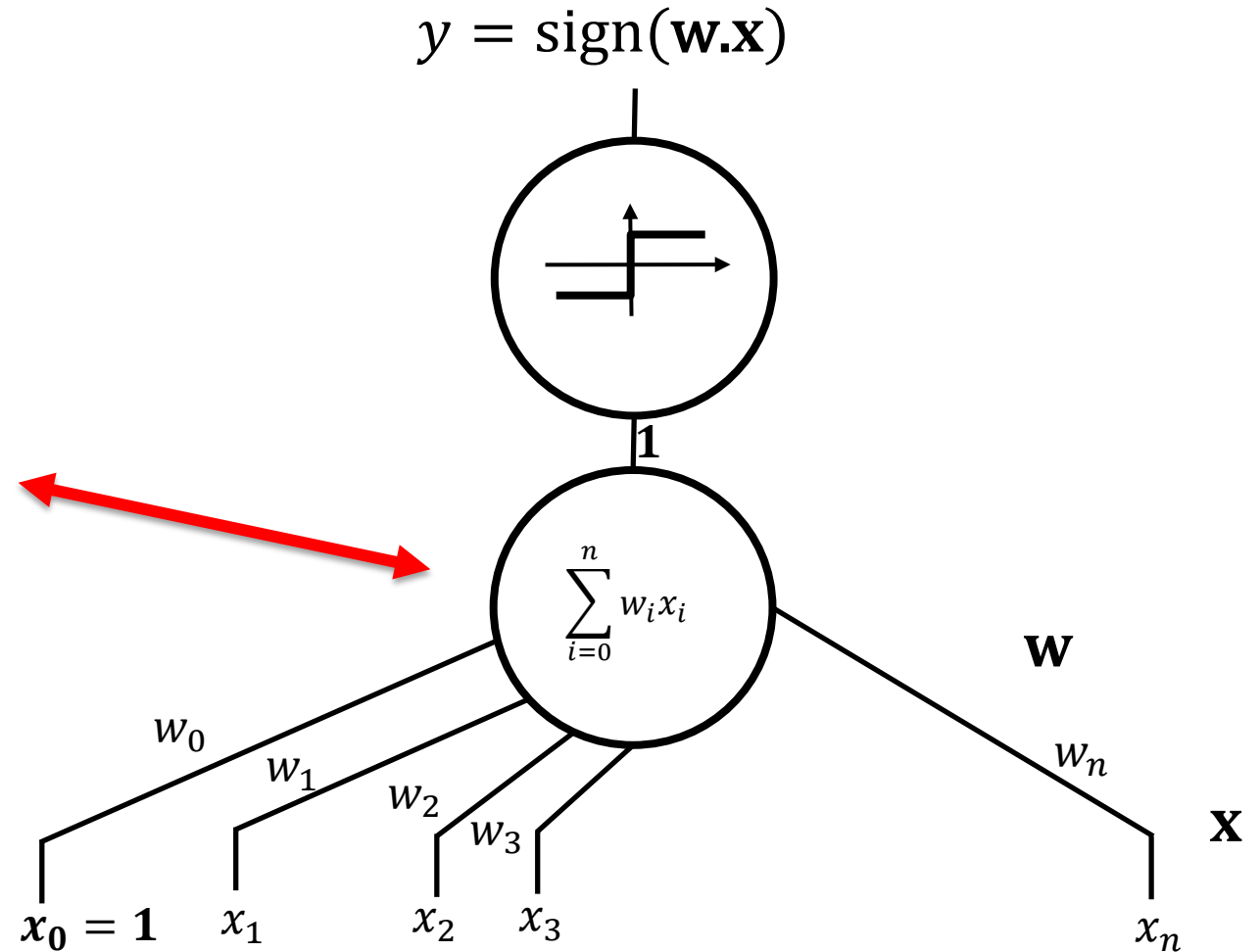
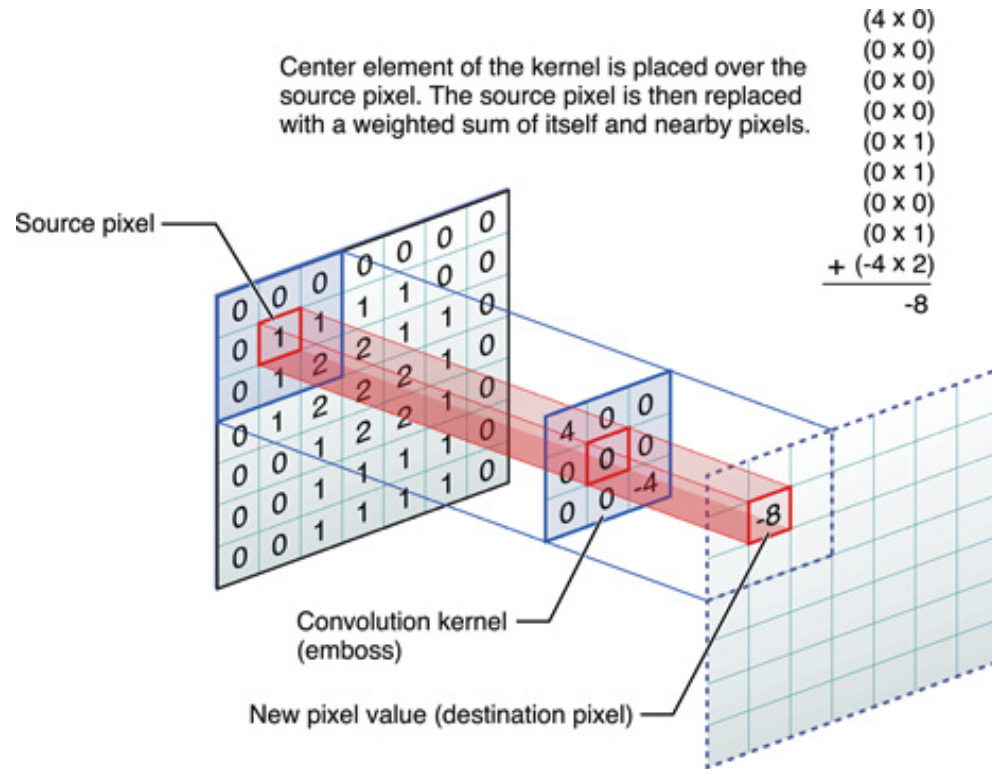
In CNN, the filter becomes a neuron



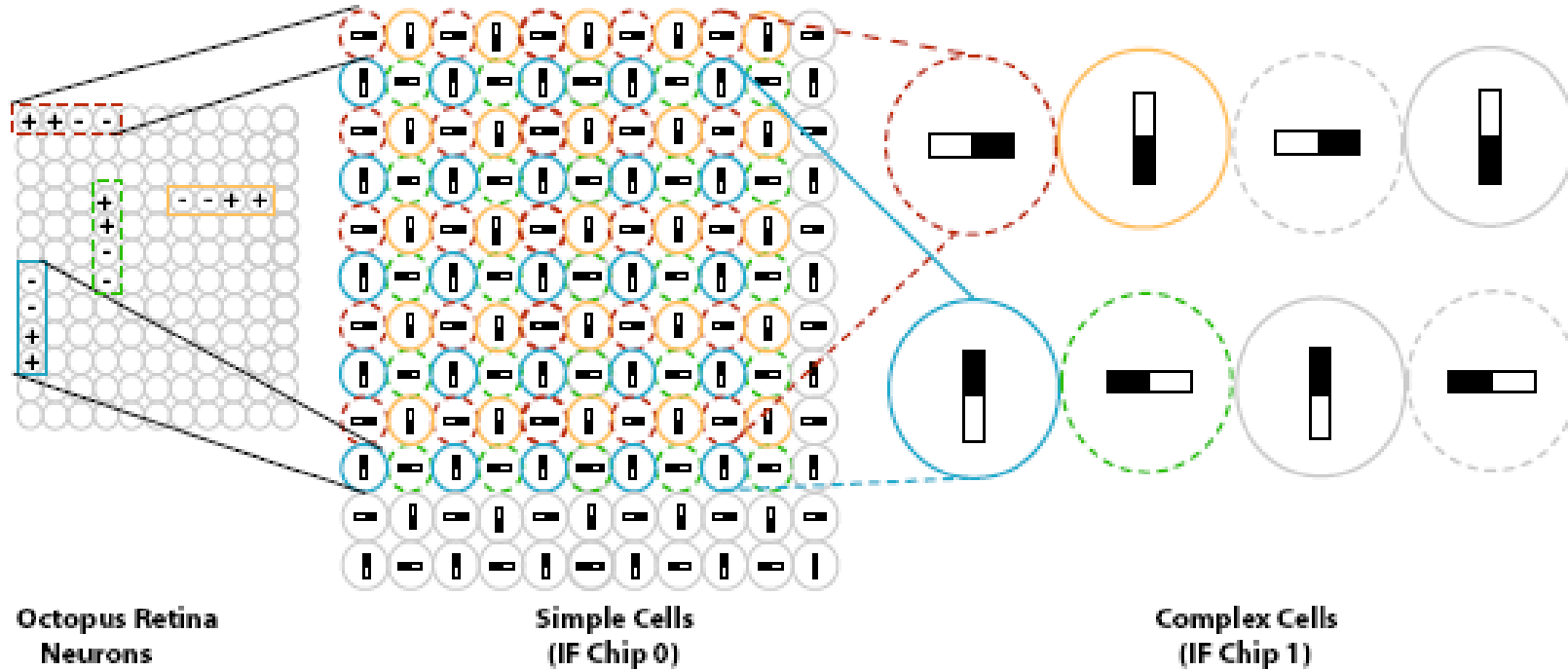
Convolution = Perceptron



Convolution = Perceptron

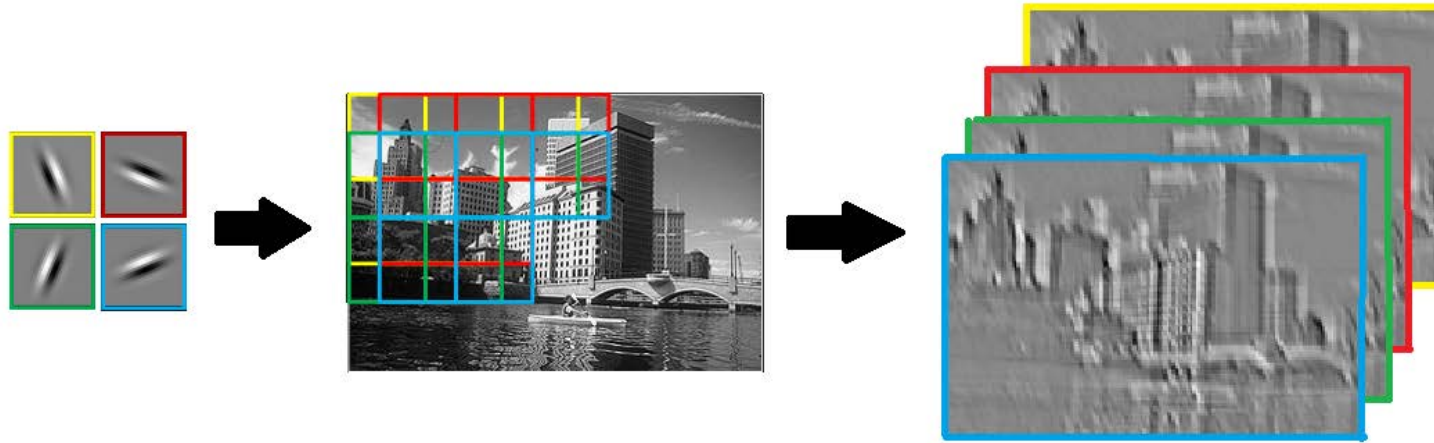


Convolution in nature

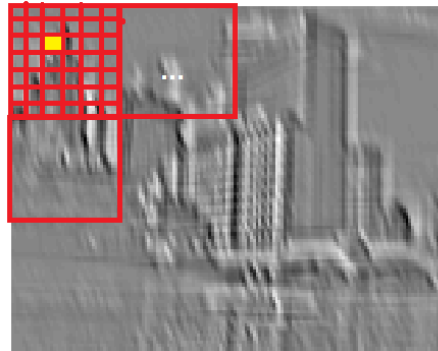


Convolution in nature

1. Hubel and Wiesel have worked on visual cortex of cats (1962)
2. Convolution

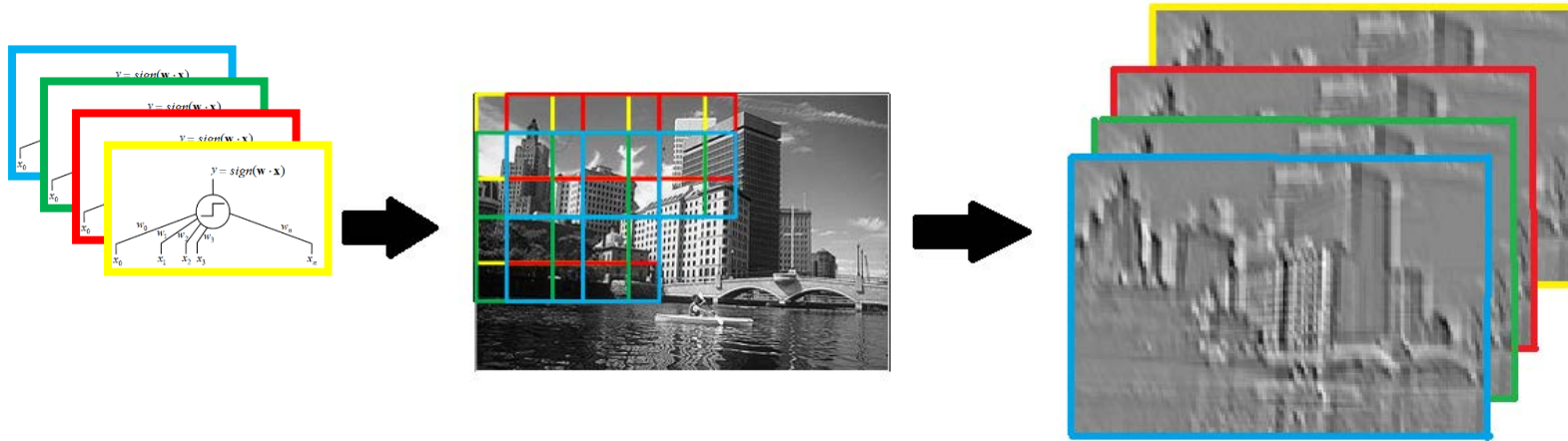


3. Pooling

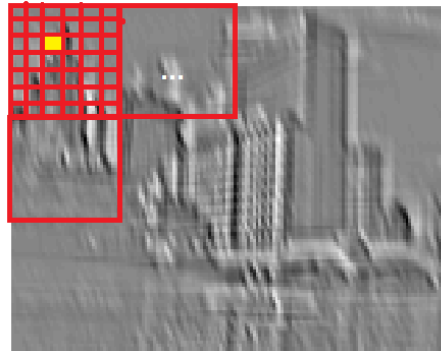


If convolution = perceptron

1. Convolution



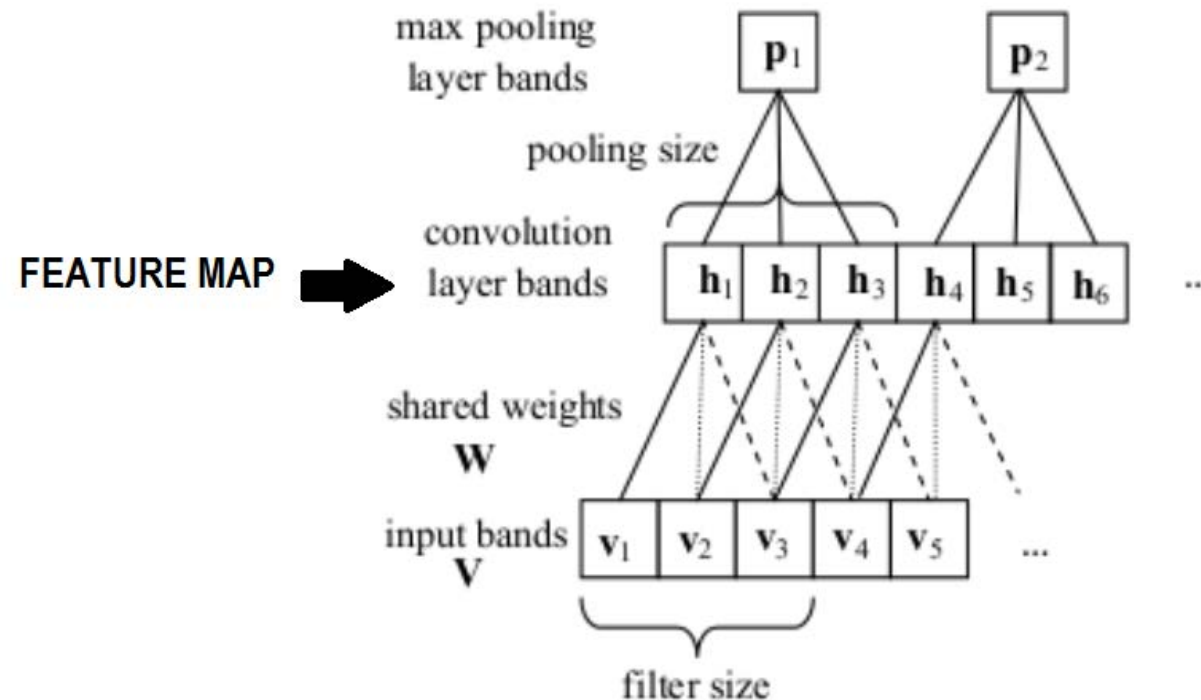
2. Pooling



Deep representation by CNN

Yann Lecun, [LeCun et al., 1998]

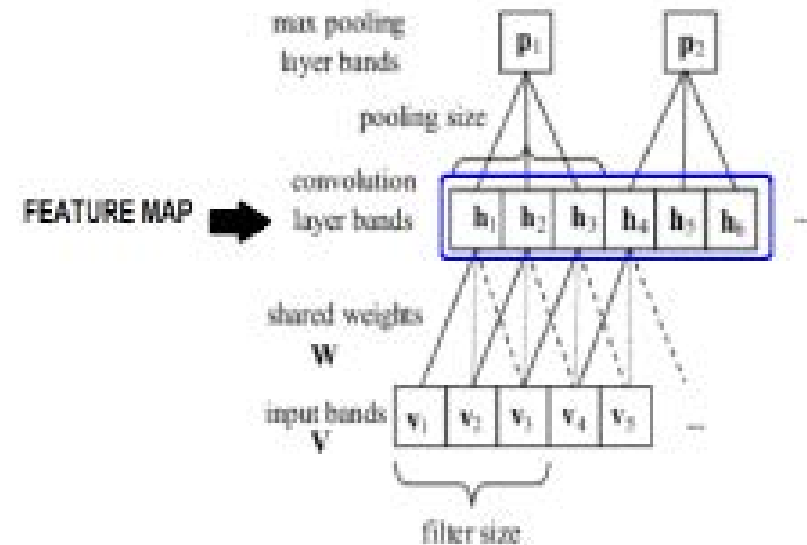
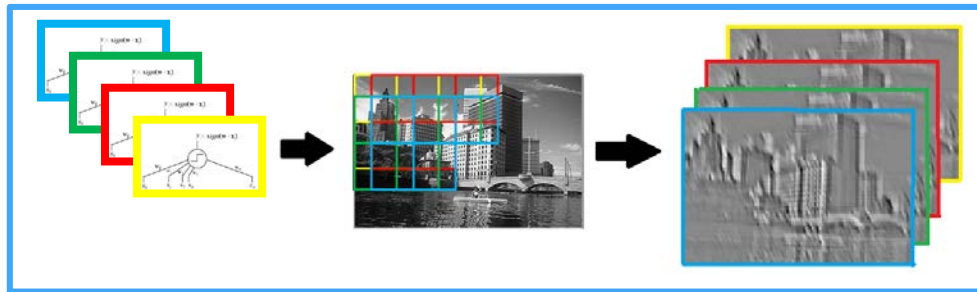
1. Subpart of the field of vision and translation invariant
2. S cells: convolution with filters
3. C cells: max pooling



Deep representation by CNN

Yann Lecun, [LeCun et al., 1998]

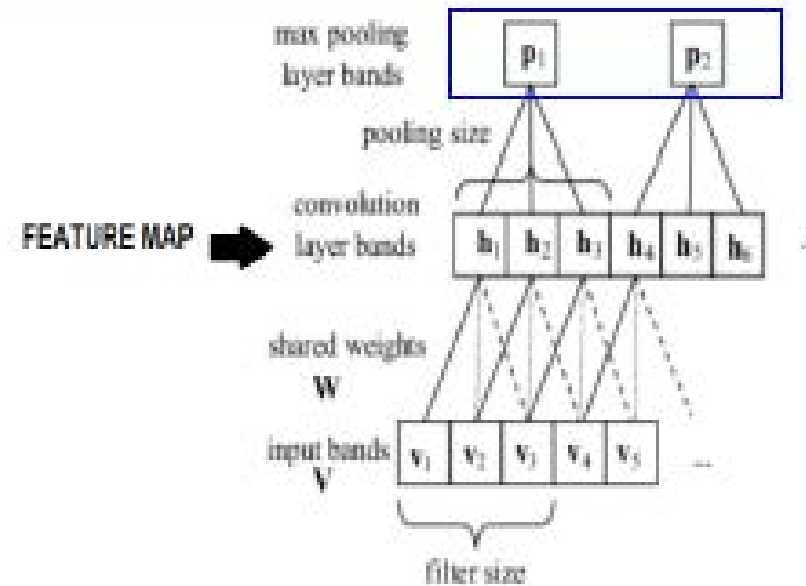
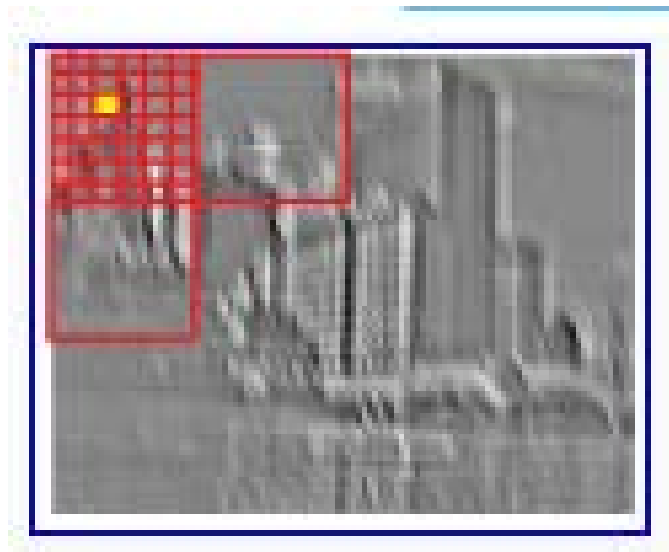
1. Subpart of the field of vision and translation invariant
2. S cells: convolution with filters
3. C cells: max pooling



Deep representation by CNN

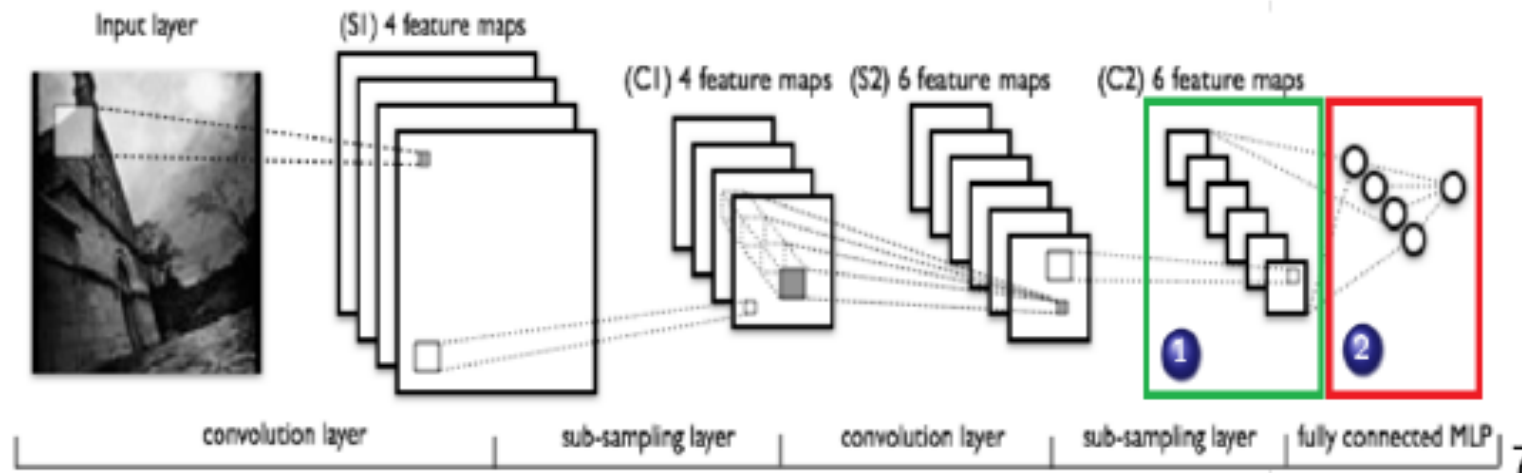
Yann Lecun, [LeCun et al., 1998]

1. Subpart of the field of vision and translation invariant
2. S cells: convolution with filters
3. C cells: max pooling



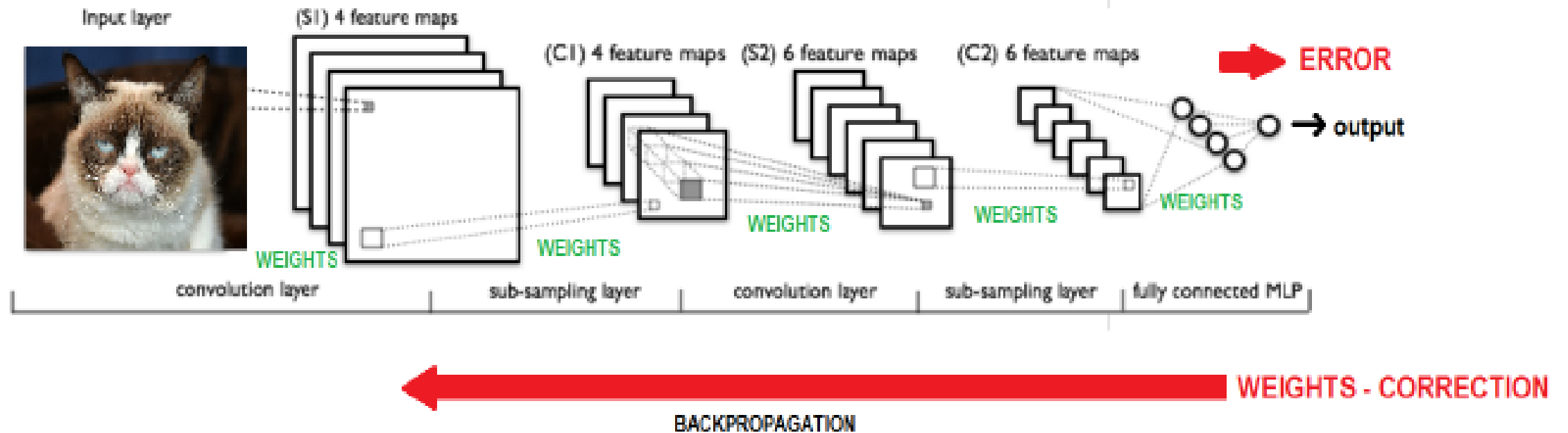
Deep representation by CNN

- feature map = result of the convolution
- convolution with a filter extract characteristics (*edge detectors*)
- extract parallelised characteristics at each layer

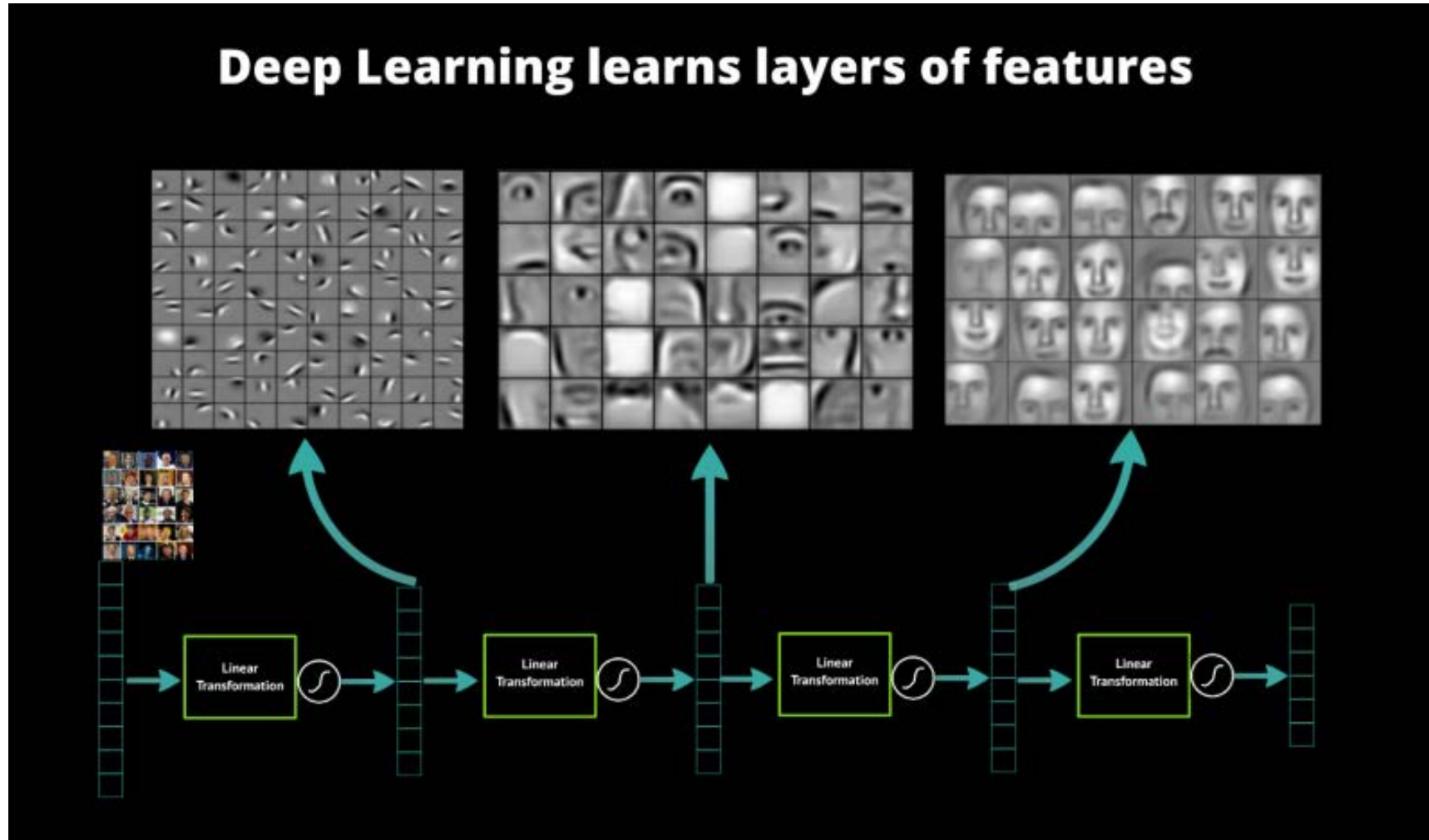


- ① final representation of our data
- ② classifier (MLP)

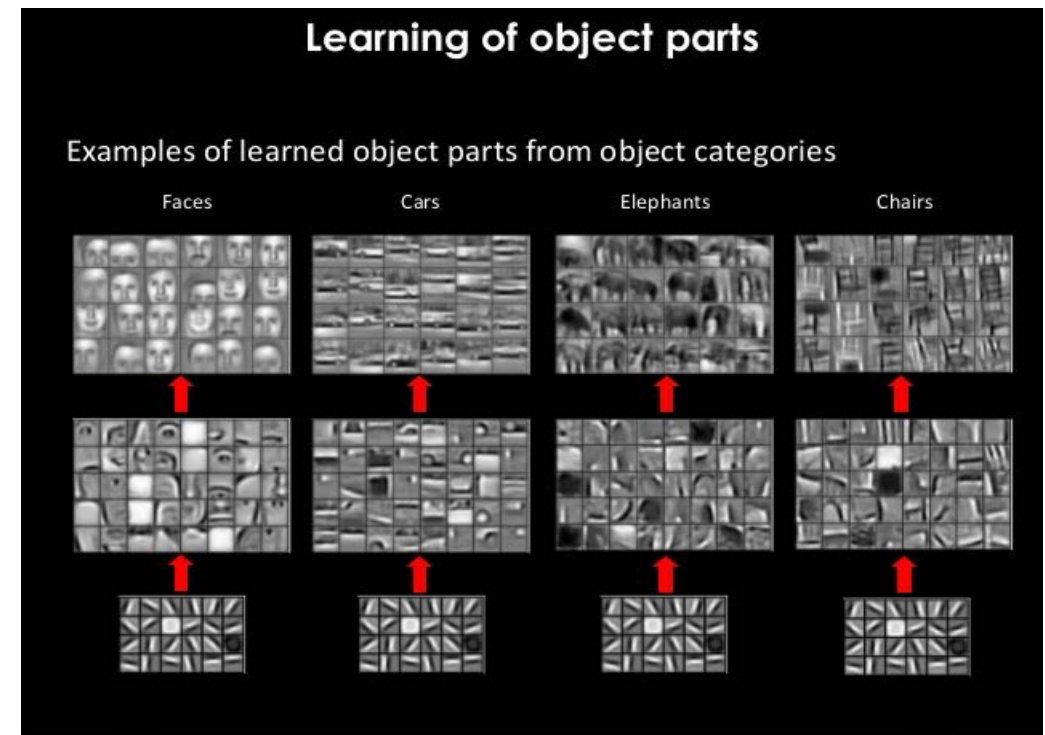
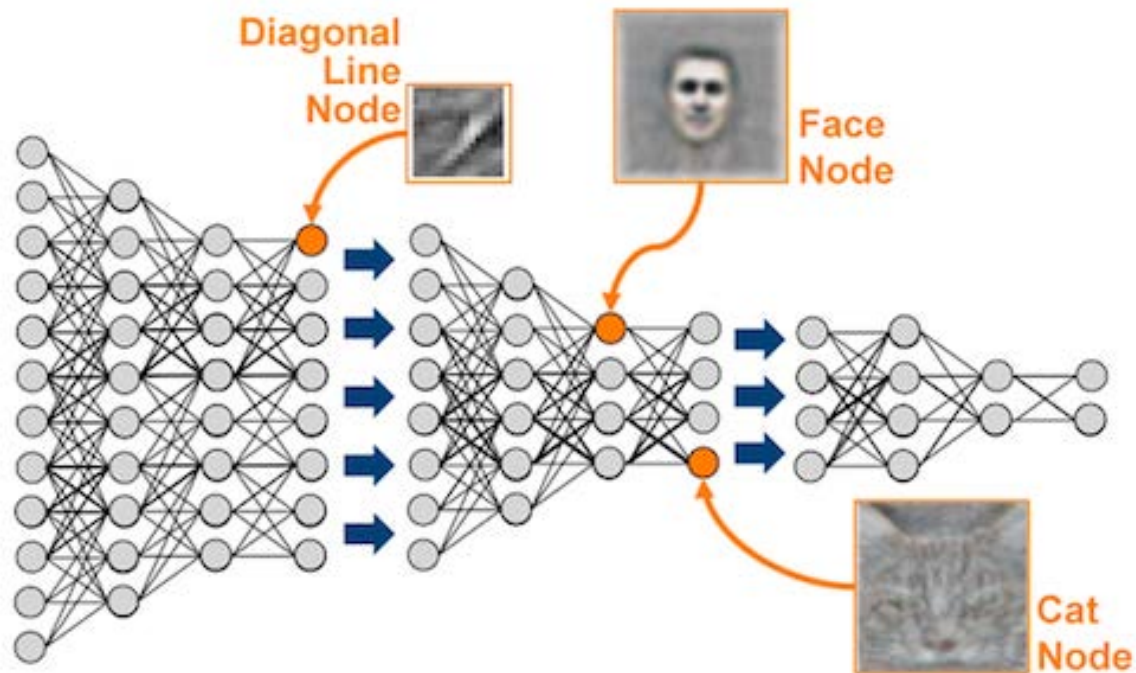
Deep representation by CNN



Deep representation by CNN



Deep representation by CNN



Transfer Learning!!

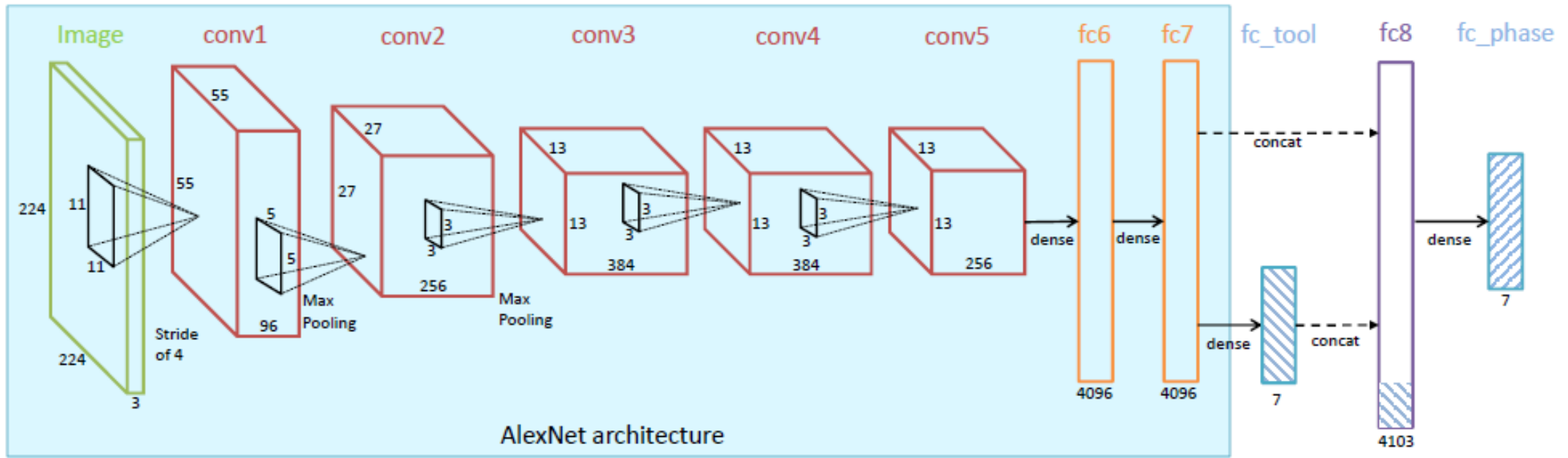


Fig. 2: EndoNet architecture (best seen in color). The layers shown in the turquoise rectangle are the same as in the AlexNet architecture.



Grasper



Bipolar



Hook



Clipper



Scissors



Irrigator



Specimen bag



Endoscopic Vision Challenge 2017

Surgical Workflow Analysis in the SensorOR

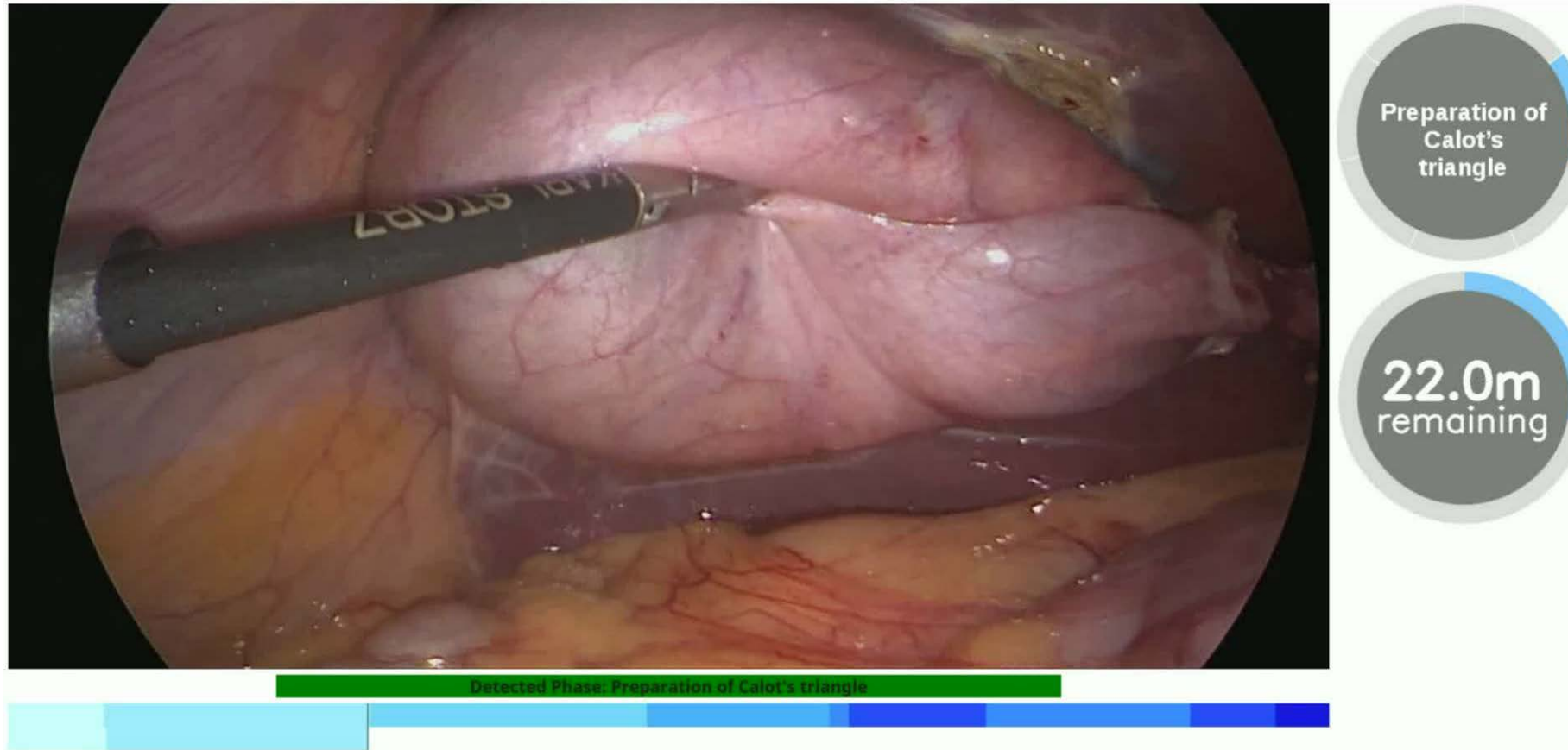
10th of September

Quebec, Canada



Clinical context: Laparoscopic Surgery

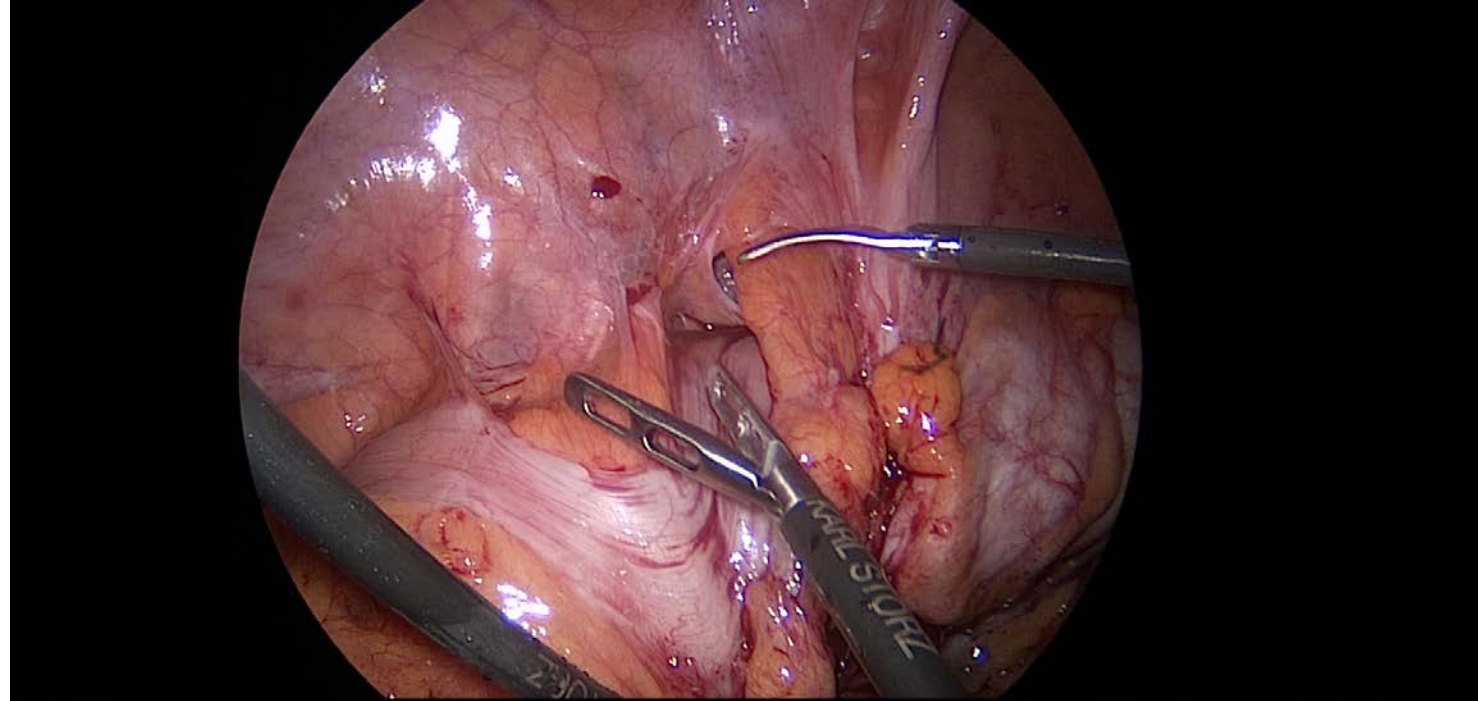
Surgical Workflow Analysis



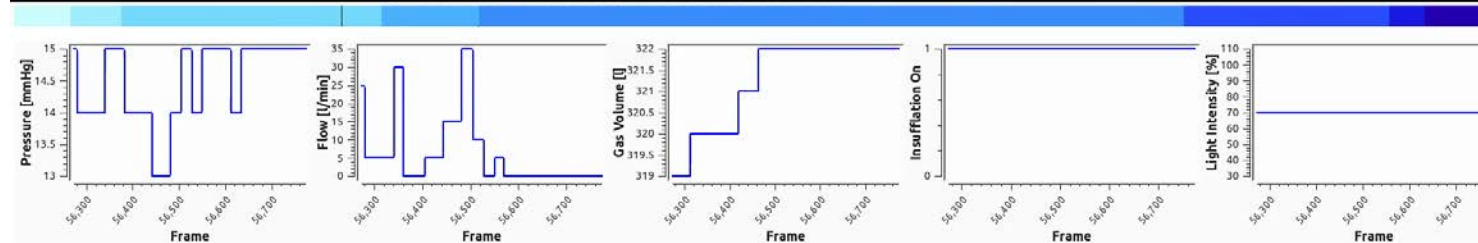
Task

Phase segmentation of laparoscopic surgeries

Video



Surgical
Devices





Dataset

30 colorectal laparoscopies

- Complex type of operation
- Duration: 1.6h – 4.9h (avg 3.2h)
- 3 different sub-types
 - 10x Proctocolectomy
 - 10x Rectal resection
 - 10x Sigmoid resection

Recorded at



Heidelberg University Hospital



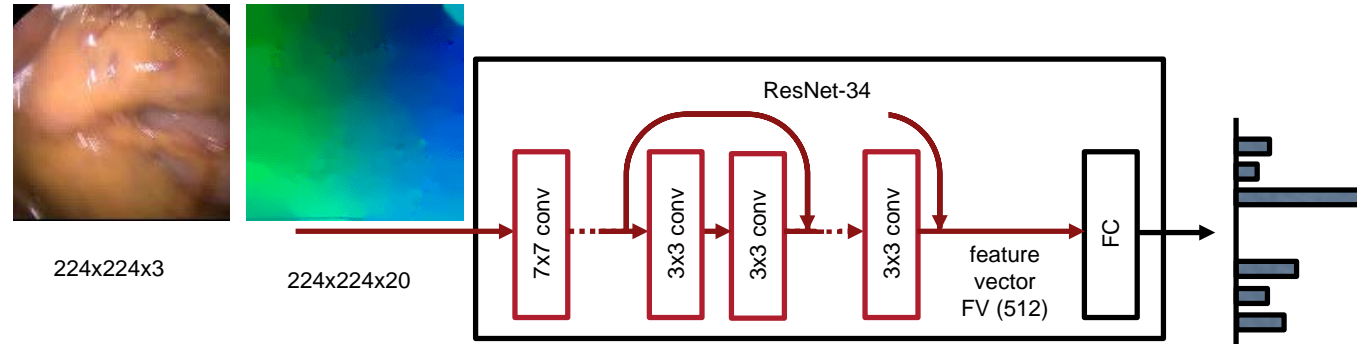
Annotation

Annotated by surgical experts, 13 different phases

Phase ID	Phase
0	Preparation and orientation at abdomen
1	Dissection of lymphnodes and blood vessels
2	Retroperitoneal preparation to lower pancreatic border
3	Retroperitoneal preparation of duodenum and pancreatic head
4	Mobilizing the sigmoid and the descending colon
5	Mobilizing the splenic flexure
6	Mobilizing the transverse colon
7	Mobilizing the ascending colon
8	Dissection and resection of rectum
9	Preparing the anastomosis extraabdominally
10	Preparing the anastomosis intraabdominally
11	Placing stoma
12	Finishing the operation
13	Exception (will be ignored during evaluation)



Temporal Network



Number of target classes:

Rectal resection: 11
Sigmoid resection: 10
Proctocolectomy: 12

Spatial network accuracy:

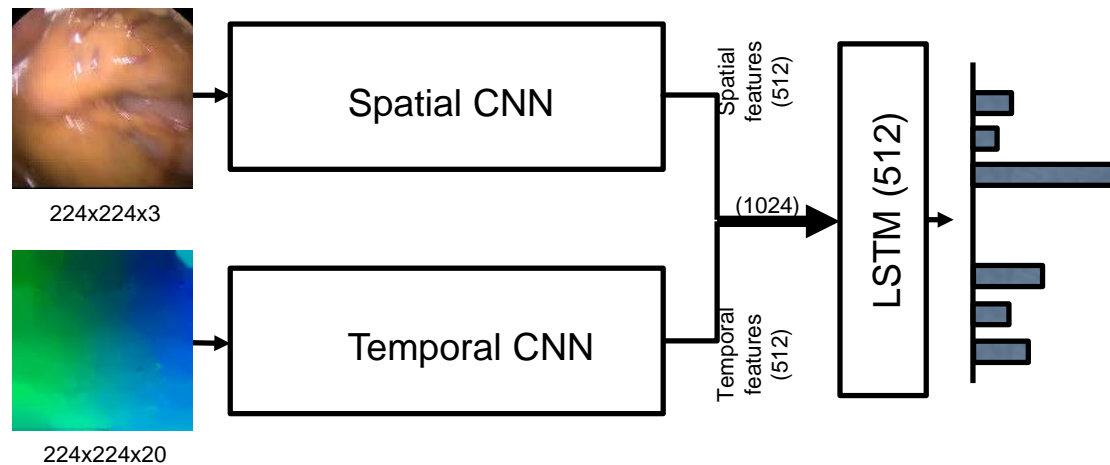
Rectal resection: 62.91%
Sigmoid resection: 63.01%
Proctocolectomy: 63.26%

Temporal network accuracy:

Rectal resection: 49.88%
Sigmoid resection: 48.56%
Proctocolectomy: 46.96%

Spatial Network

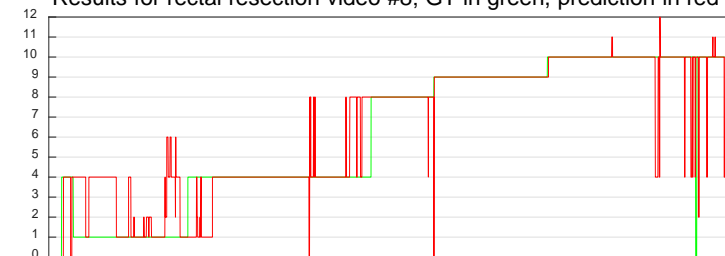
Final Network



Final Network Accuracy:

Rectal resection (8): 80.7%	sigmoid resection (7): 73.5%	Proctocolectomy (1): 71.3%
Rectal resection (6): 79.9%	sigmoid resection (1): 54.7%	Proctocolectomy (4): 73.9%

Results for rectal resection video #8; GT in green, prediction in red

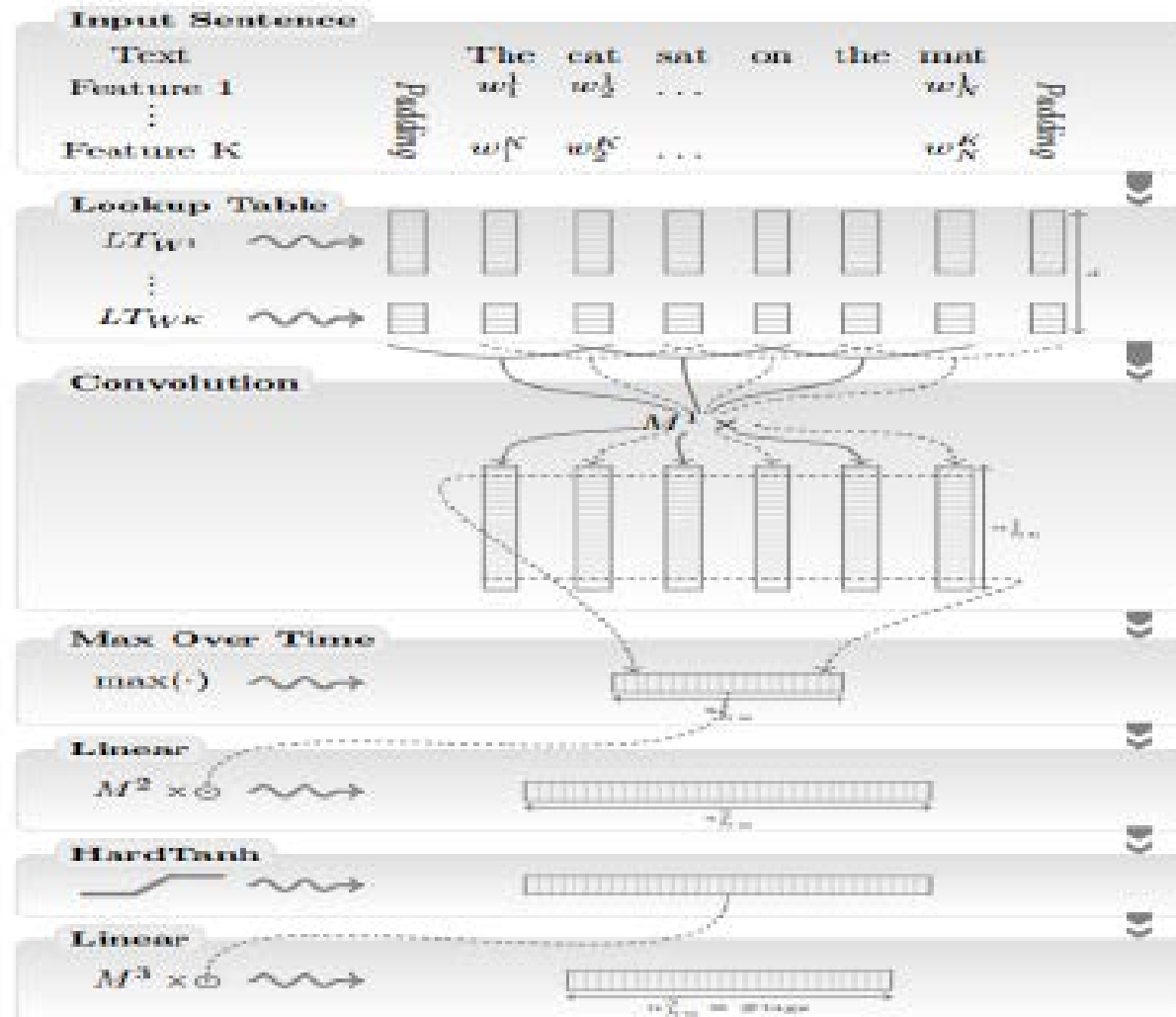




And the winner is...

	Data used	Average Jaccard	Median Jaccard	Accuracy	
1	Video	40%	38%	61%	Team UCA
2	Video + Device	38%	38%	60%	Team NCT
3	Video	25%	25%	57%	Team TUM
4	Device	16%	16%	36%	Team TUM
5	Video	8%	7%	21%	Team FFL

Extension to text





Extension to text

Task	Benchmark	Collobert
Part of Speech	97.24%	97.29%
Chunking	94.29%	94.32%
Named Entity Recognition	77.92%	75.49%
Semantic Role Labeling	89.31%	89.59%

Collobert is working quite well but:

- ① 852 million words
- ② 4 weeks

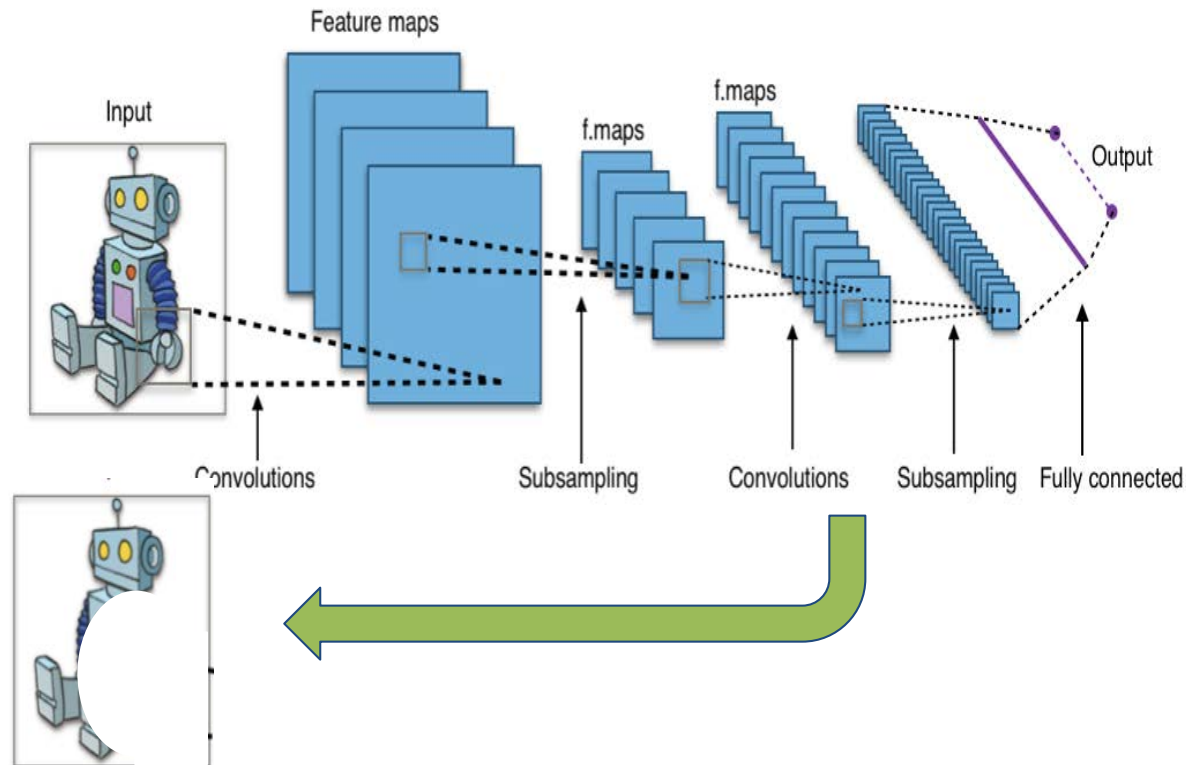
Extension to text

Extraction of higher order linguistic structures with CNN on texts...

L. Vanni, M. Ducoffe, D. Mayaffre, F. Precioso, D. Longrée, C. Aguilar, V. Elango, N. Santos, L. Galdo, J. Gonzalez, ***Text Deconvolution Saliency (TDS): a deep tool box for linguistic analysis***, in 56th Annual Meeting of the Association for Computational Linguistics (ACL), Melbourne, Australia, July 2018.

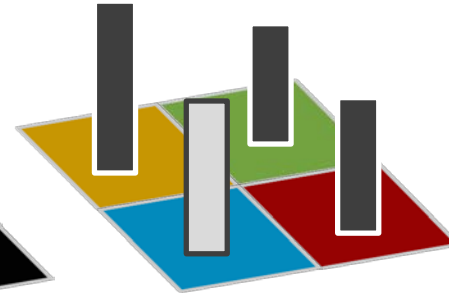
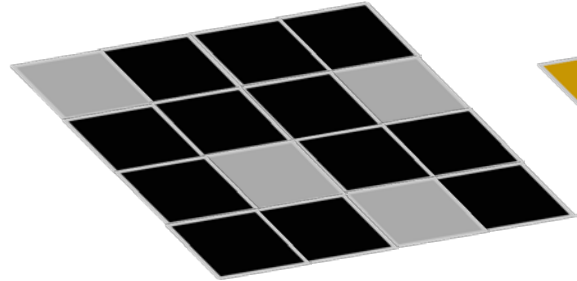
“Deconvolution” for CNNs

- Deconvolution = mapping the information of a CNN at layer k back into the input space
- Deconvolution for images => highlighting pixels
- Approximating the inverse of a convolution by a convolution
- “inverse of a filter” = “transpose” of a filter



Reversible Max Pooling

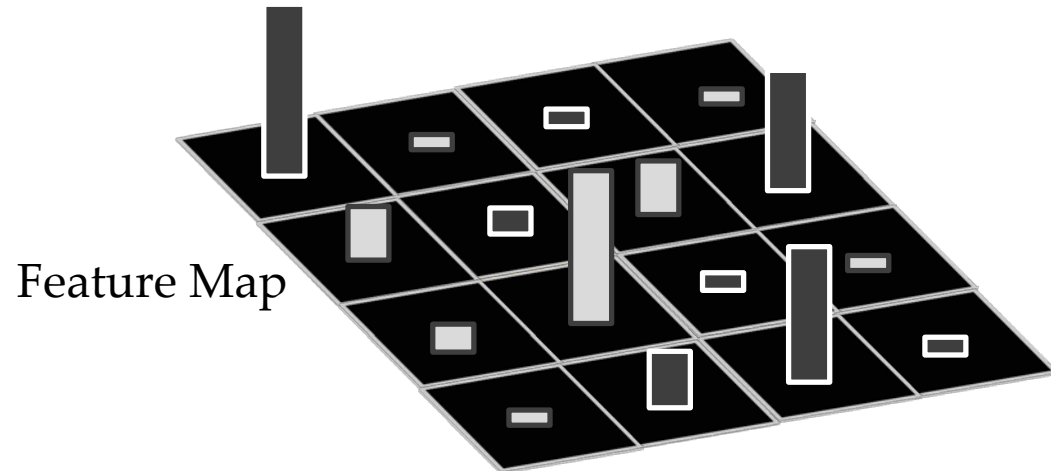
Max
Locations
“Switches”



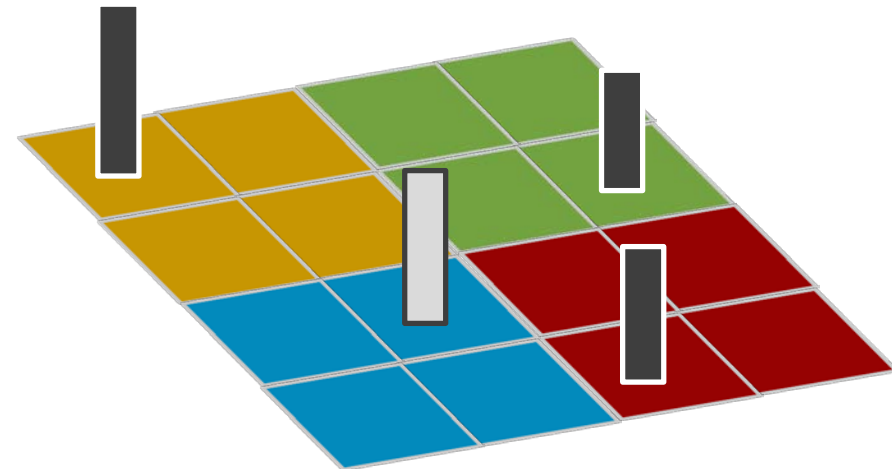
Pooled
Feature Maps

Pooling

Unpooling

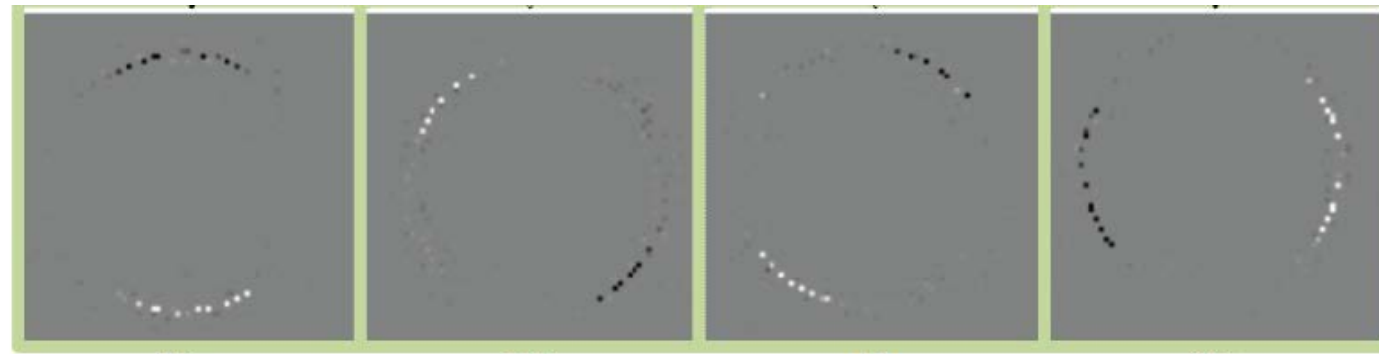


Feature Map



Reconstructed Feature Map

Toy Example

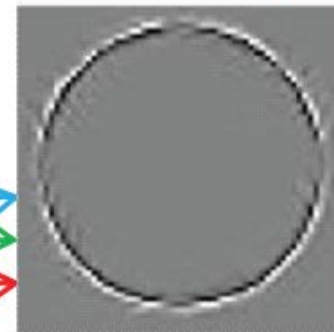
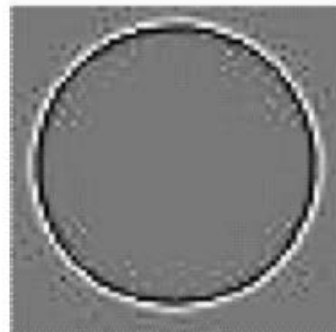


Feature
maps



Filters

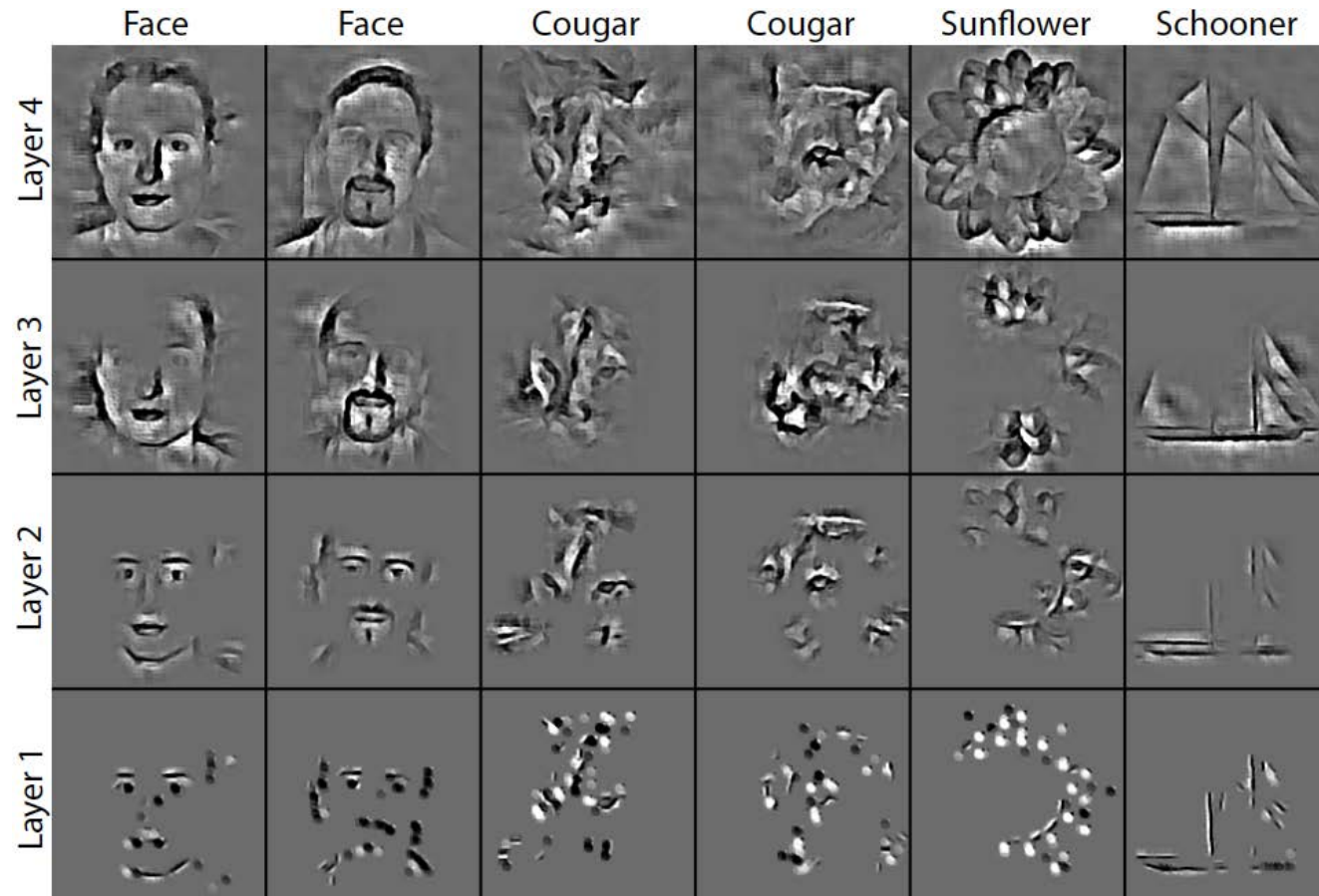
Input
Image
 y



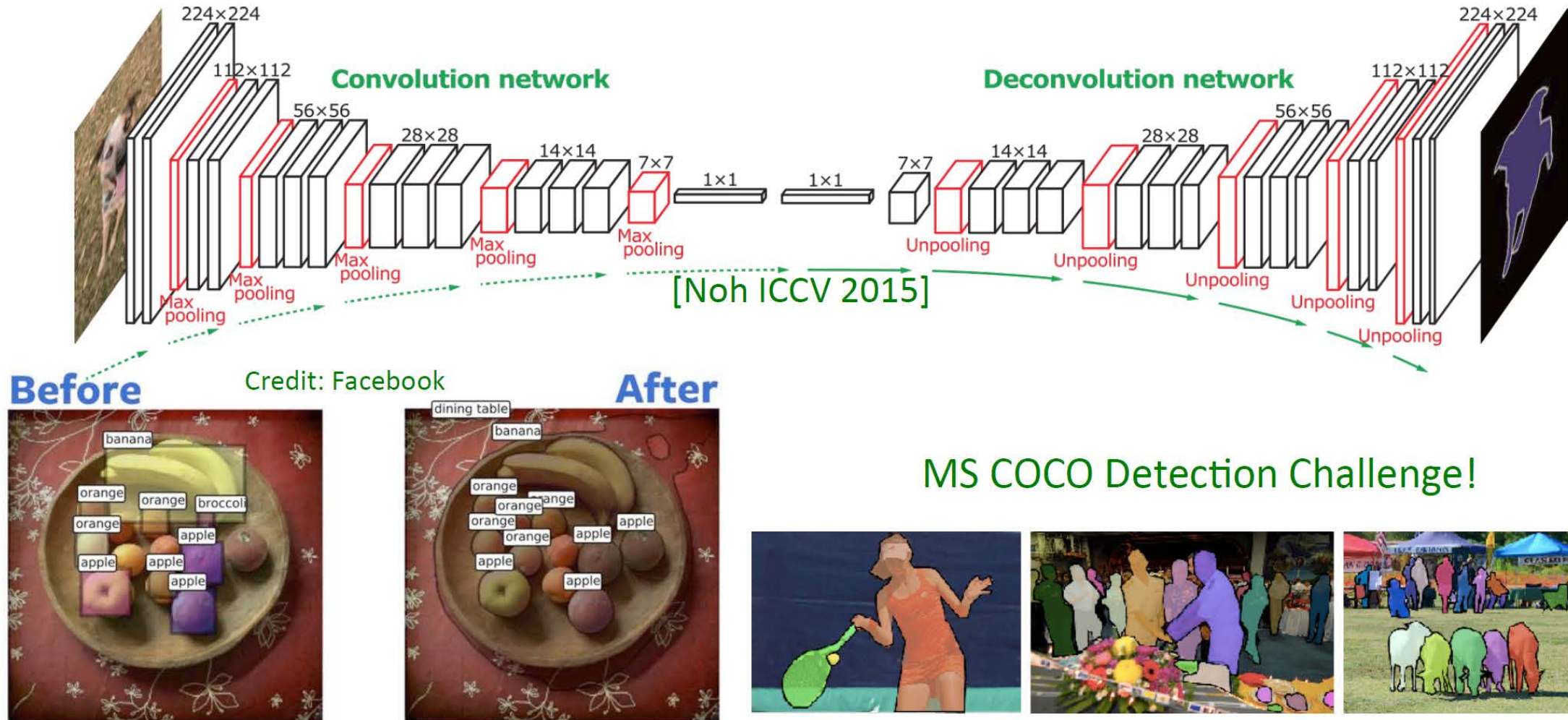
Reconstructed
Image
 \hat{y}

Top-down Decomposition

- Pixel visualizations of strongest features activated from top-down reconstruction from single max in top layer.



Supervised Image Segmentation Task



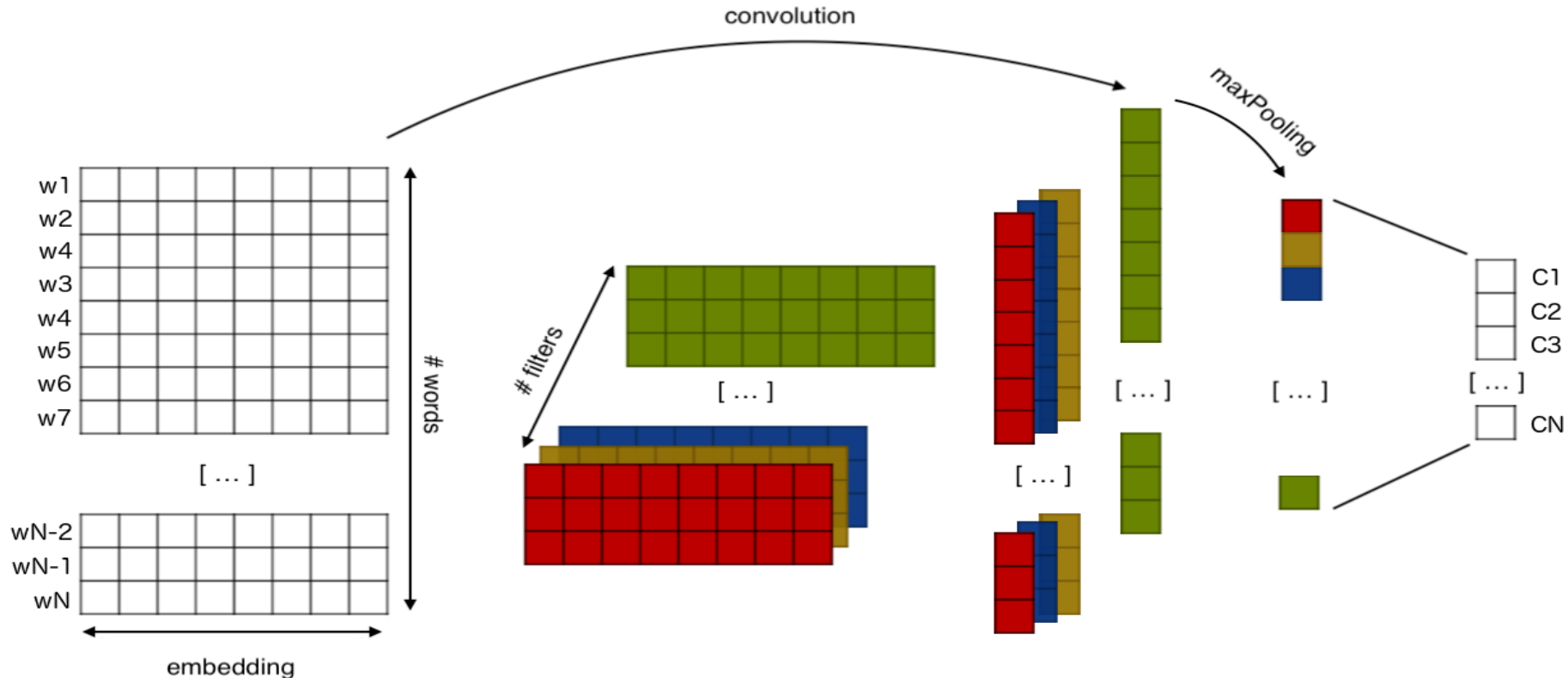
Credits Matthieu Cord

CNN for Sentence Classification

filters' width = |embedding|

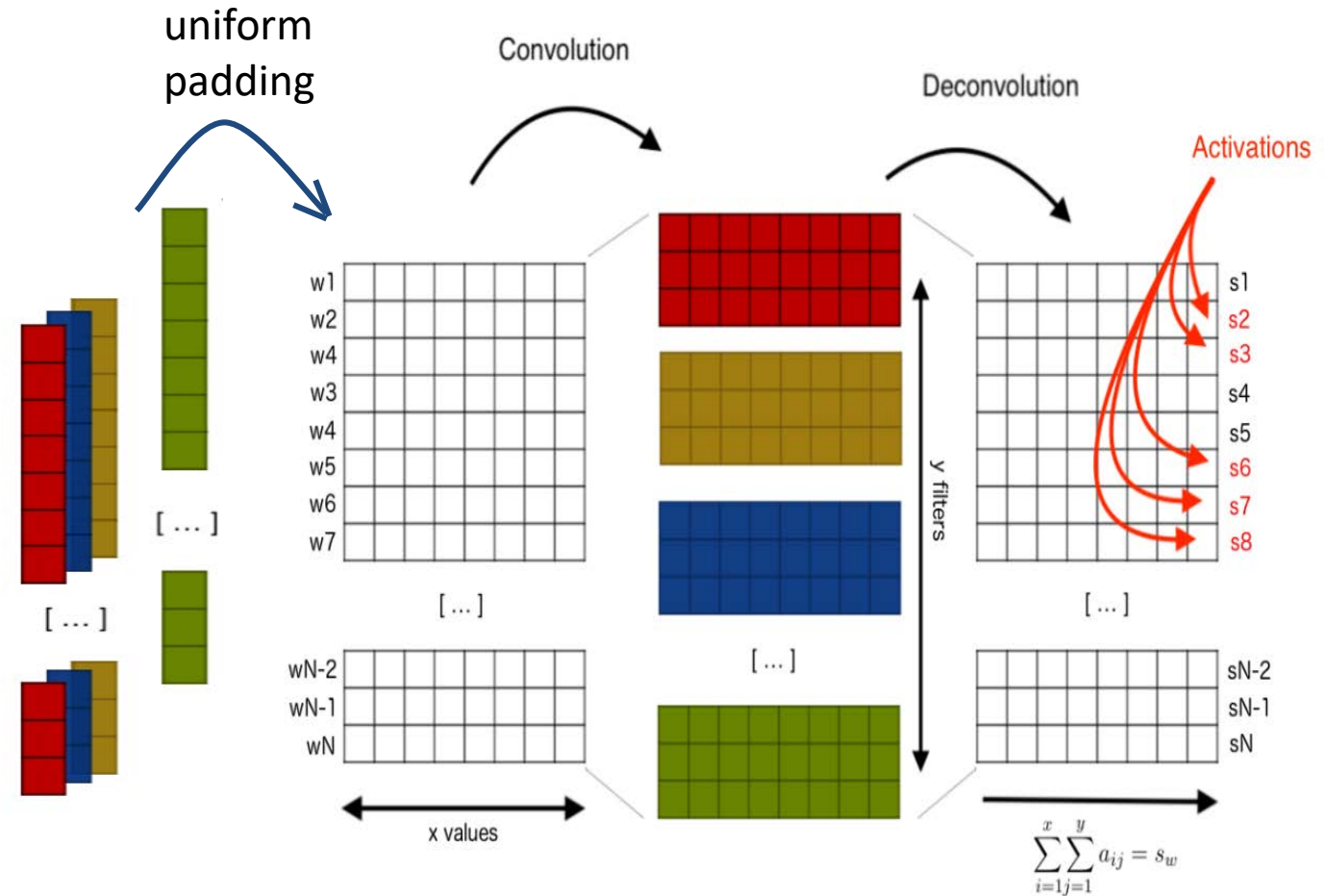
words + embeddings => non isotropic dimensions

No sense to transpose the filters for deconvolution



Deconvolution for CNNs [Vanni 2018]

- 1) apply uniform padding
- 2) repeat the convolution
- 3) sum the contribution along the embedding dimension



Empirical Evaluation of TDS

- threshold on the TDS score
- IMDb dataset, classification of positive and negative sentiments + french political discourses
- markers for classification, independently of the final prediction
- highlight co-occurrences : “transformations + profondes”, “transformations + subjonctif”

Quote 12.5.1: English Review

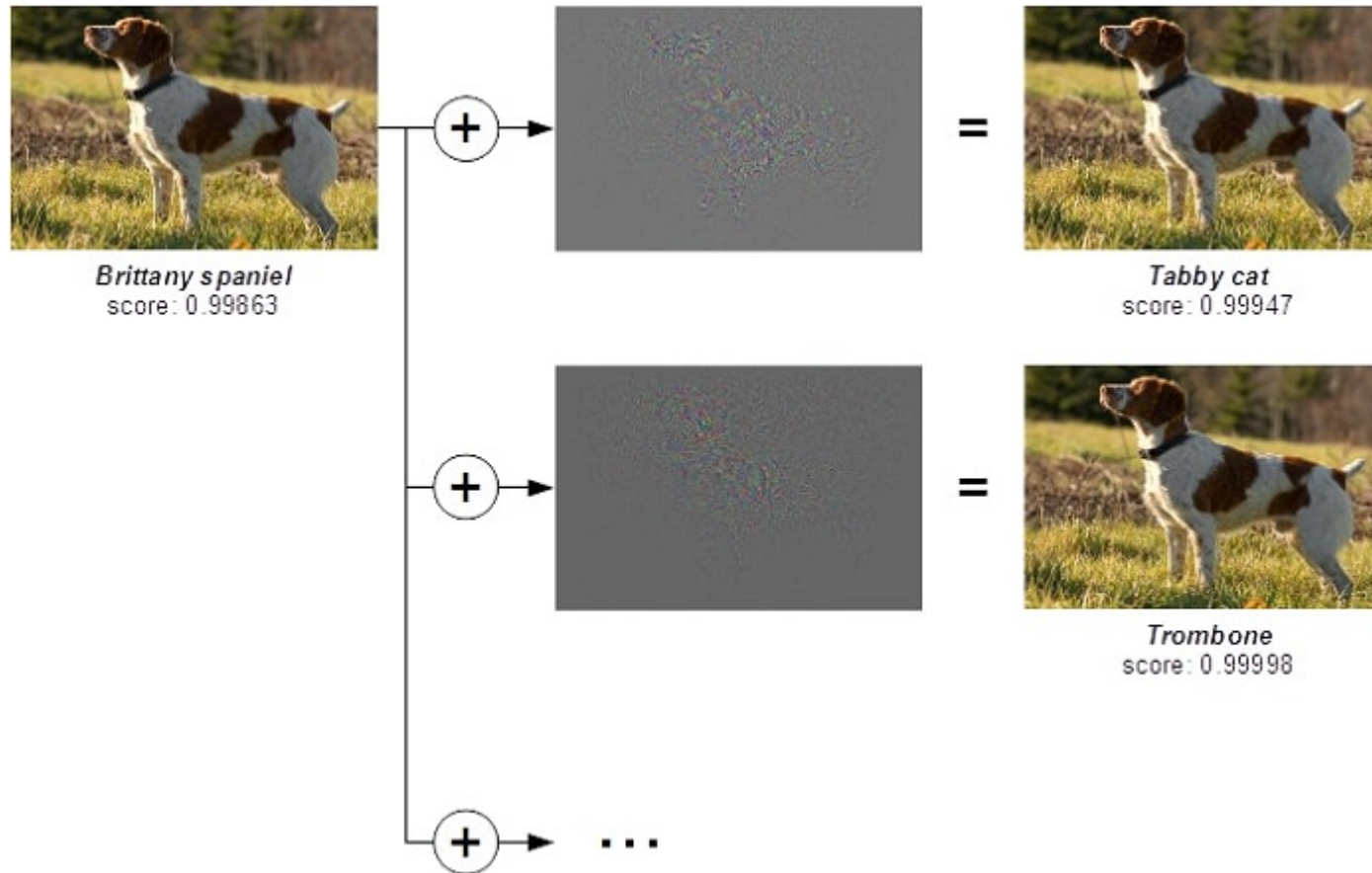
[...] i enjoyed three moments in the film in total , and if i am being honest and the person next to me fell asleep in the middle and started snoring during the slow space chasescenes . the story failed to draw me in and entertain me the way [...]

Quote 12.5.2: French discourse

[...] notre pays advienne à l'école pour nos enfants, au travail pour l'ensemble de nos concitoyens pour le climat pour le quotidien de chacune et chacun d'entre vous . Ces transformations profondes ont commencé et se poursuivront avec la même force le même rythme la même intensité [...]

AMAZING BUT...

Amazing but...be careful of the adversaries (as any other ML algorithms)



From Thomas Tanay

Amazing but...be careful of the adversaries (as any other ML algorithms)



Red Light Modified to
Green after 18 white pixels.
Probability: 59%



Red Light Modified to
Green after 9 green pixels.
Probability: 50.9%



Red Light Modified to
Green after 9 green pixels.
Probability: 53%



No Light Modified to Green
after 4 green pixels.
Probability: 51.9%

Amazing but...be careful of the adversaries (as any other ML algorithms)



stop

30m
speed
limit

80m
speed
limit

30m
speed
limit

go
right

go
straight

Confidence 0.999964 0.99

Amazing but...be careful of the adversaries

https://nicholas.carlini.com/code/audio_adversarial_examples/



Amazing but...be careful of the adversaries (as any other ML algorithms)

Intriguing properties of neural networks

C. Szegedy, w. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I.
Goodfellow, R. Fergus

arXiv preprint arXiv:1312.6199

2013

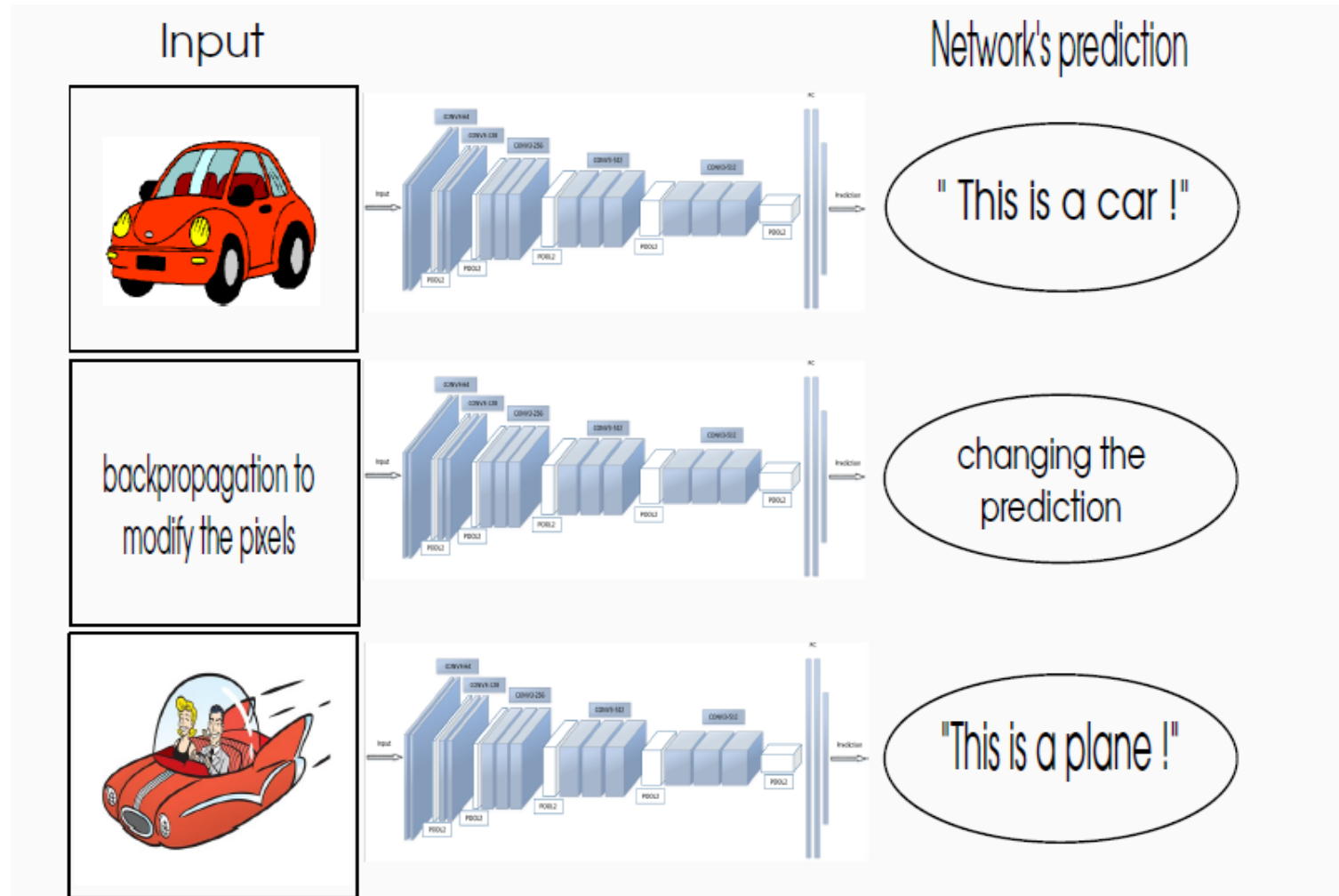
[1312.6199] Intriguing properties of neural networks - arXiv.org

<https://arxiv.org> > cs - Traduire cette page

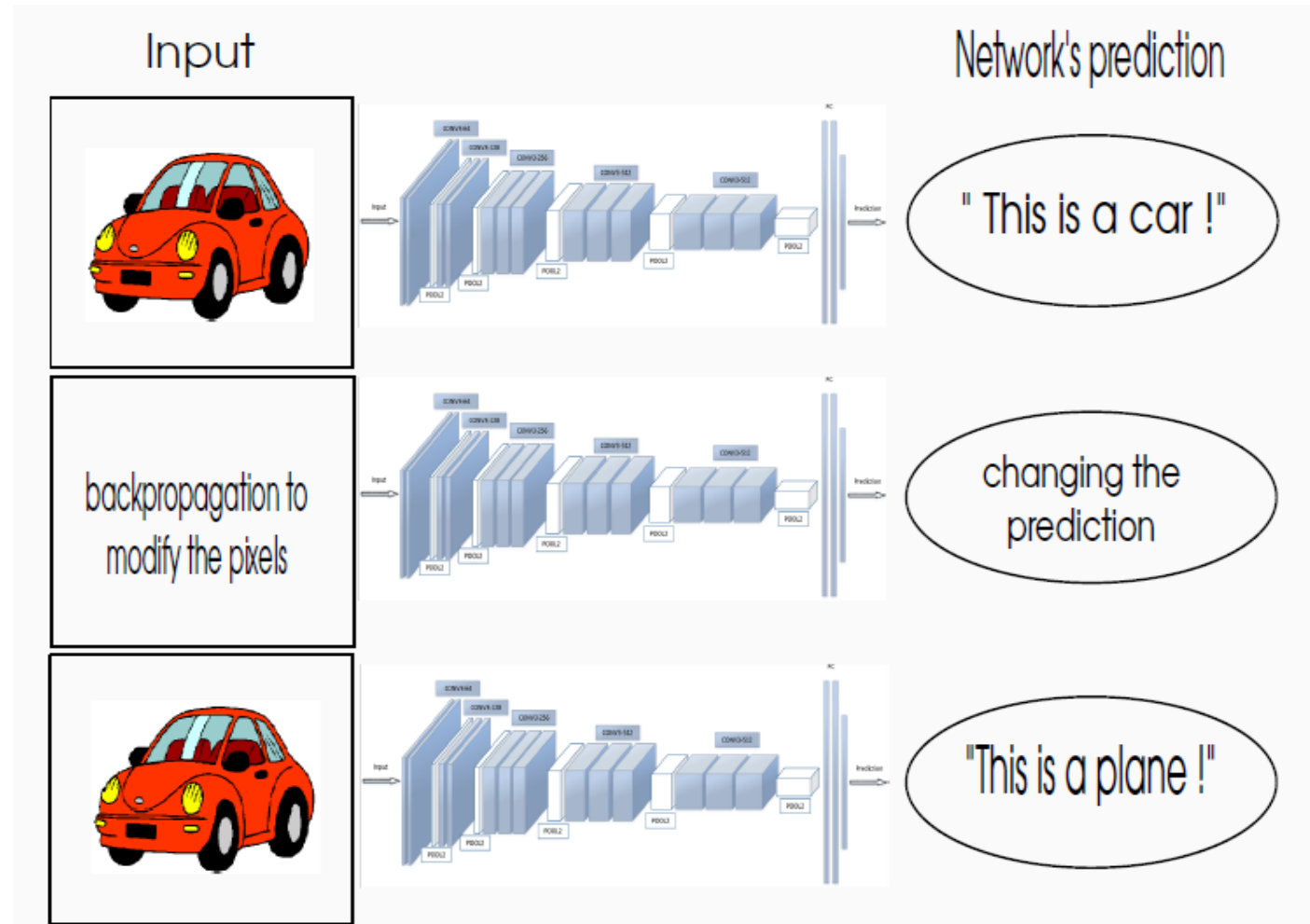
de C Szegedy - 2013 - Cité 449 fois - Autres articles

21 déc. 2013 - In this paper we report two such **properties**. First, we ... Second, we find that deep **neural networks** learn input-output mappings that are fairly ...

Amazing but...be careful of the adversaries (as any other ML algorithms)



Amazing but...be careful of the adversaries (as any other ML algorithms)





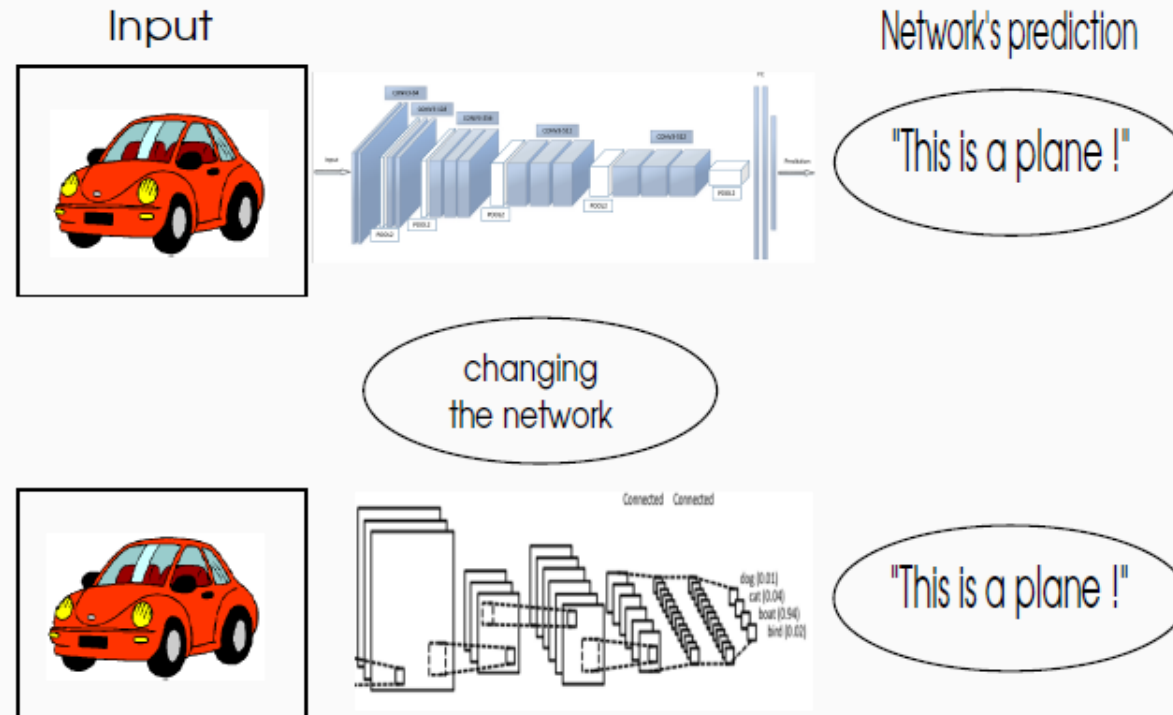
Amazing but...be careful of the adversaries (as any other ML algorithms)

Definition: \hat{x} is called adversarial iff:

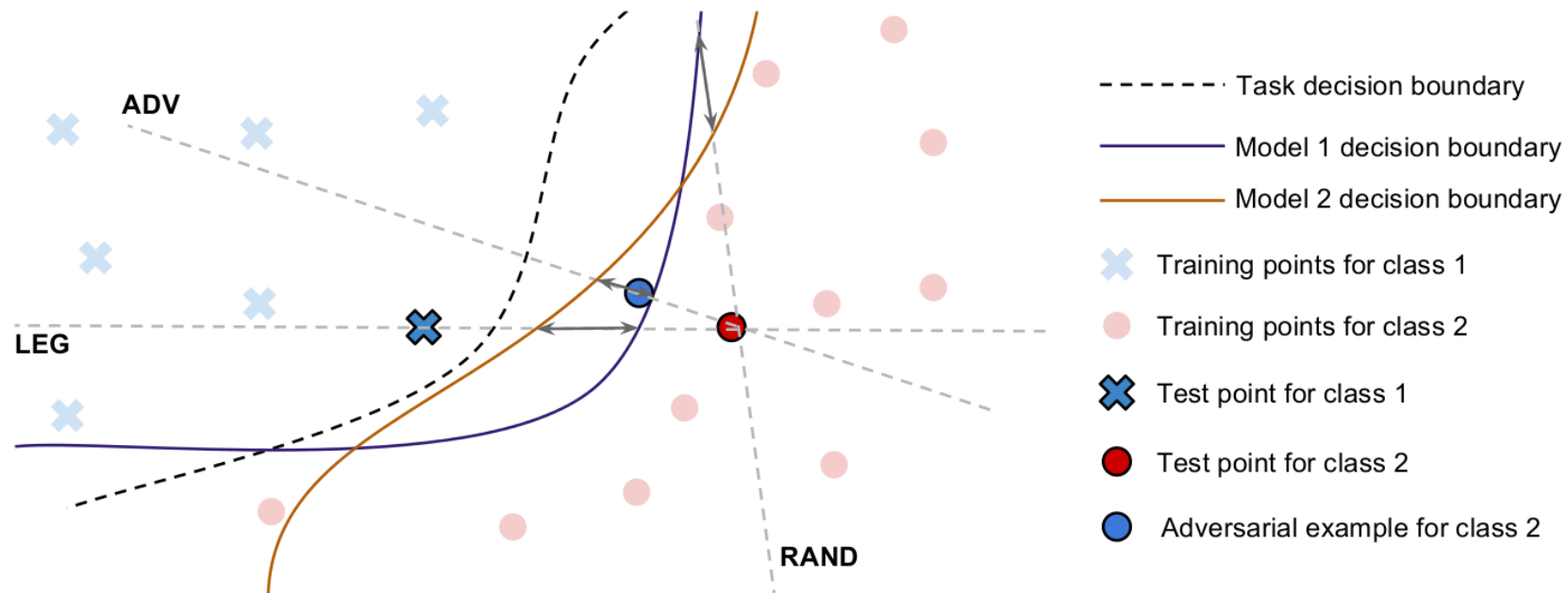
- given image x
- low distortion $\|x - \hat{x}\| < \epsilon$, ($\epsilon > 0$, few pixels)
- given network's probabilities $f_{\theta}(x)$
- **Different predictions!** $\operatorname{argmax}_{\theta}(x) \neq \operatorname{argmax}_{\theta}(\hat{x})$

Amazing but...be careful of the adversaries (as any other ML algorithms)

- \neq outliers
- regularization: correct one... find another
- high confidence predictions
- **Transferability**

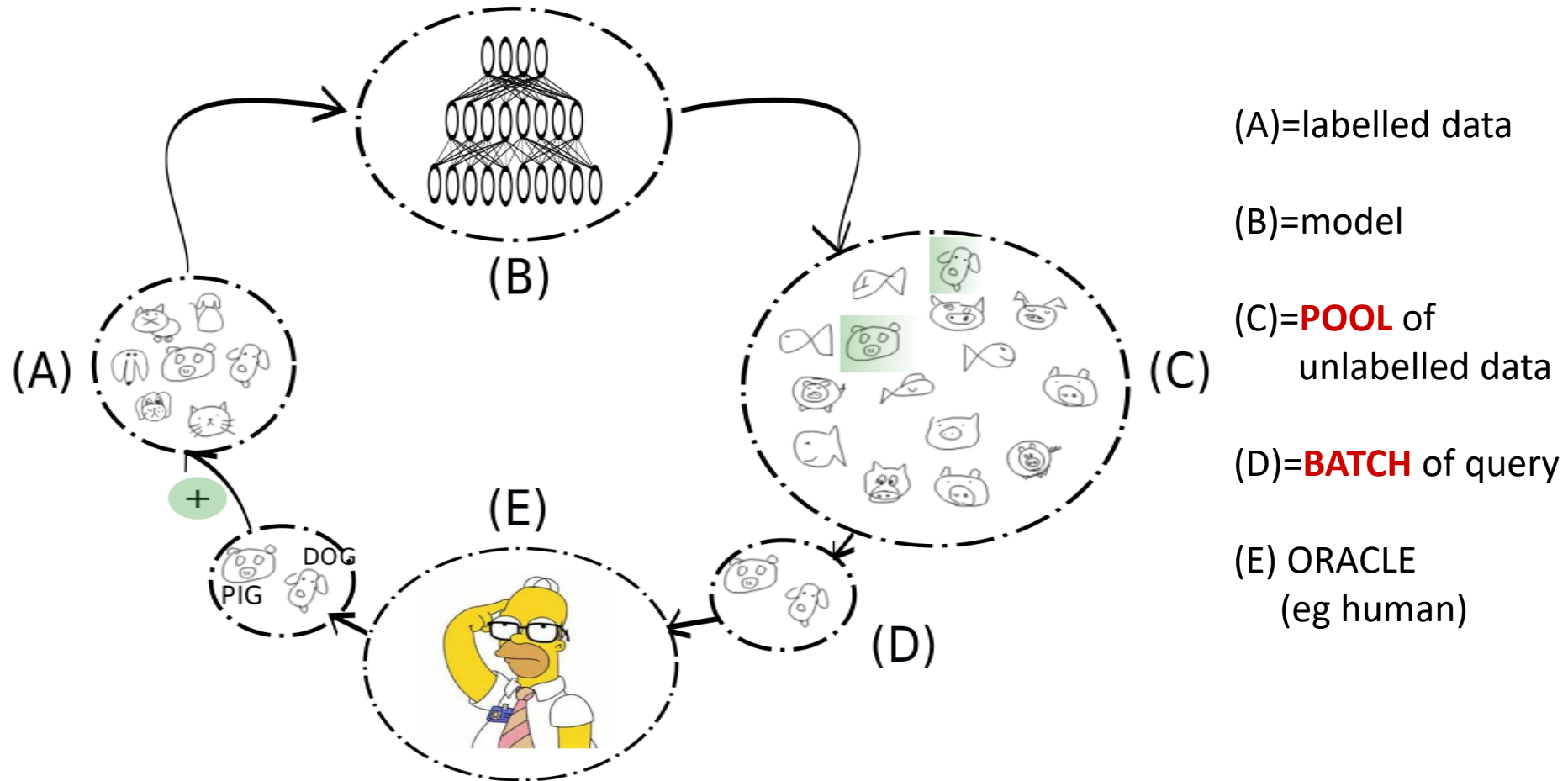


Adversarial examples...



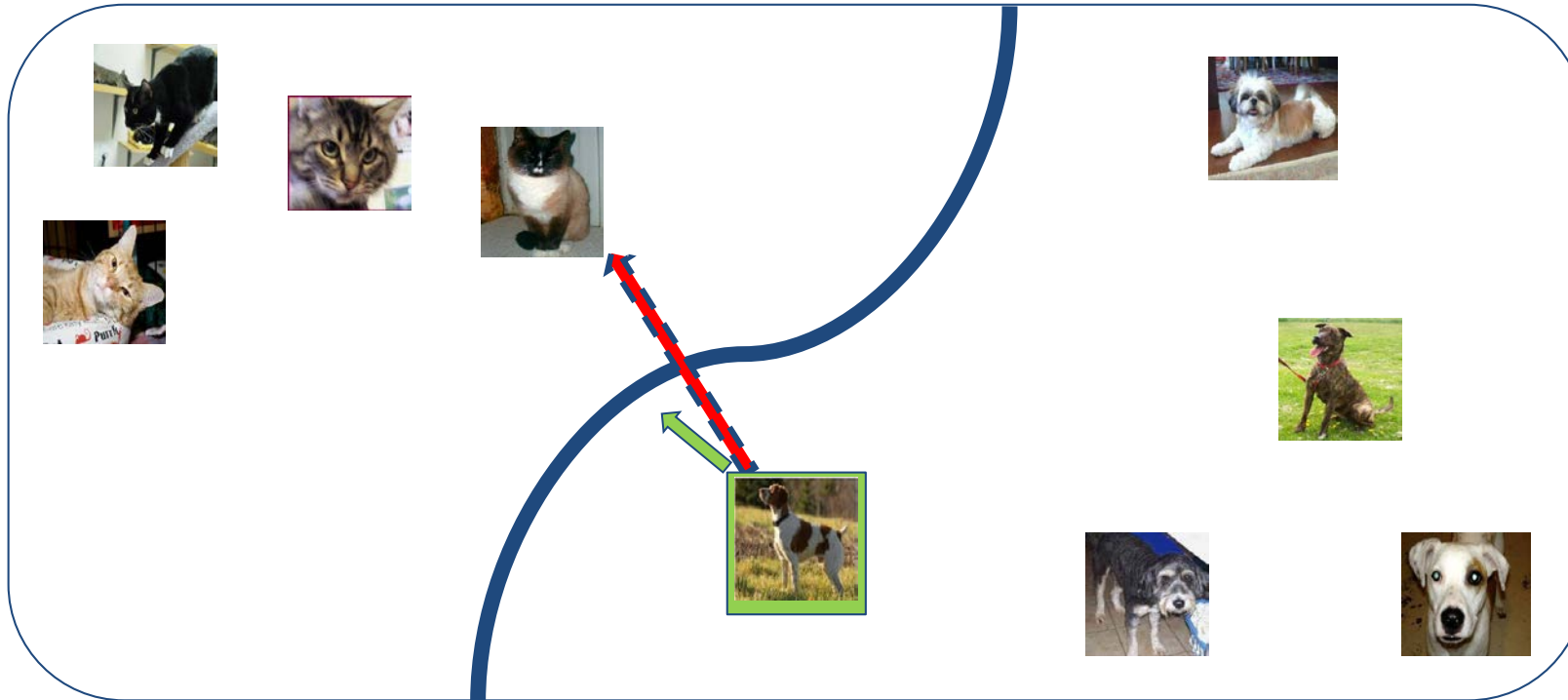
Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2017).
The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*.

ACTIVE Supervised Classification



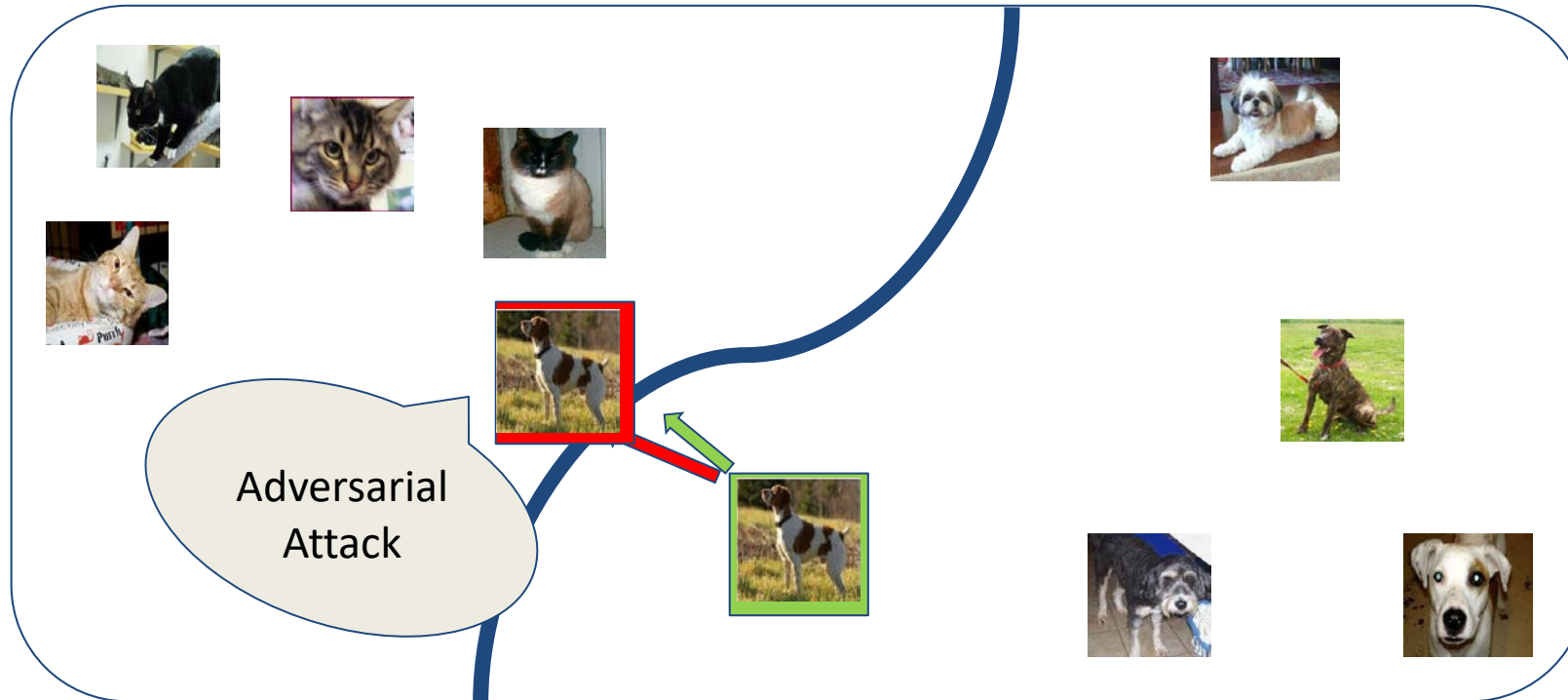
MARGIN BASED ACTIVE LEARNING?

- query => close to the decision boundary
- topology of the decision boundary unknown
- how can we approximate such a distance for neural networks ?
- *"Dimension + Volume + Labels"*




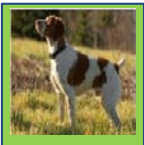

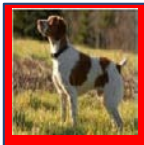




Adversarial attacks for MARGIN BASED ACTIVE LEARNING

- query → small adversarial perturbation



DeepFool Active Learning (DFAL)

	<p>query the top-k examples owing the smallest DeepFool adversarial perturbation</p> <div data-bbox="1462 514 1605 656">  </div> <p>= 'dog'</p>
	<p>query the adversarial attacks with the same label pseudo-labeling comes for free without corrupting the training set</p> <div data-bbox="1070 885 1212 1028">  </div> <p>= 'dog'</p> <div data-bbox="1462 921 1651 985">  </div> <div data-bbox="1712 885 1854 1028">  </div> <p>= 'dog'</p>
	<p>Transferability of adversarial examples</p> <div data-bbox="1676 1149 1717 1235">  </div> <p>Transferability of queries</p>

DFAL EXPERIMENTS 1/3

- Query the top-k samples owing the smallest adversarial perturbation (**DFAL 0**)
- **BATCH**= 10
- **|MNIST| = 60 000 |QuickDraw|=444 971**

		Accuracy (%)			
# annotations	100	500	800	1000	All
DFAL_0	85.08	95.89	97.79	98.13	–
BALD	53.73	91.47	94.32	94.32	–
CEAL	50.87	90.69	90.69	90.69	–
CORE-SET	78.80	96.68	97.46	97.88	–
EGL	37.92	91.84	93.99	93.99	–
uncertainty	45.57	88.36	94.27	94.60	–
RANDOM	69.79	91.96	94.05	94.46	98.98

% of Test accuracy
MNIST (VGG8)

		Accuracy (%)			
# annotations	100	500	800	1000	All
DFAL_0	78.62	91.35	92.44	93.14	–
BALD	82.00	89.94	91.92	92.87	–
CEAL	64.45	79.66	85.73	88.65	–
CORE-SET	66.71	89.93	92.28	92.62	–
EGL	63.12	86.80	90.06	90.06	–
uncertainty	52.77	88.05	89.31	91.03	–
RANDOM	78.28	88.13	89.71	89.94	96.75

% of Test accuracy
QuickDraw (VGG8)

DFAL EXPERIMENTS 2/3

- Pseudo labeling the adversarial samples of the queries
- **BATCH**= 10
- **|MNIST| = 60 000 |QuickDraw|=444 971**

	Accuracy (%)				
# annotations	100	500	800	1000	All
DFAL	84.28	96.90	97.98	98.59	–
DFAL_0	85.08	95.89	97.79	98.13	–
BALD	53.73	91.47	94.32	94.32	–
CEAL	50.87	90.69	90.69	90.69	–
CORE-SET	78.80	96.68	97.46	97.88	–
EGL	37.92	91.84	93.99	93.99	–
uncertainty	45.57	88.36	94.27	94.60	–
RANDOM	69.79	91.96	94.05	94.46	98.98

% of Test accuracy
MNIST (VGG8)

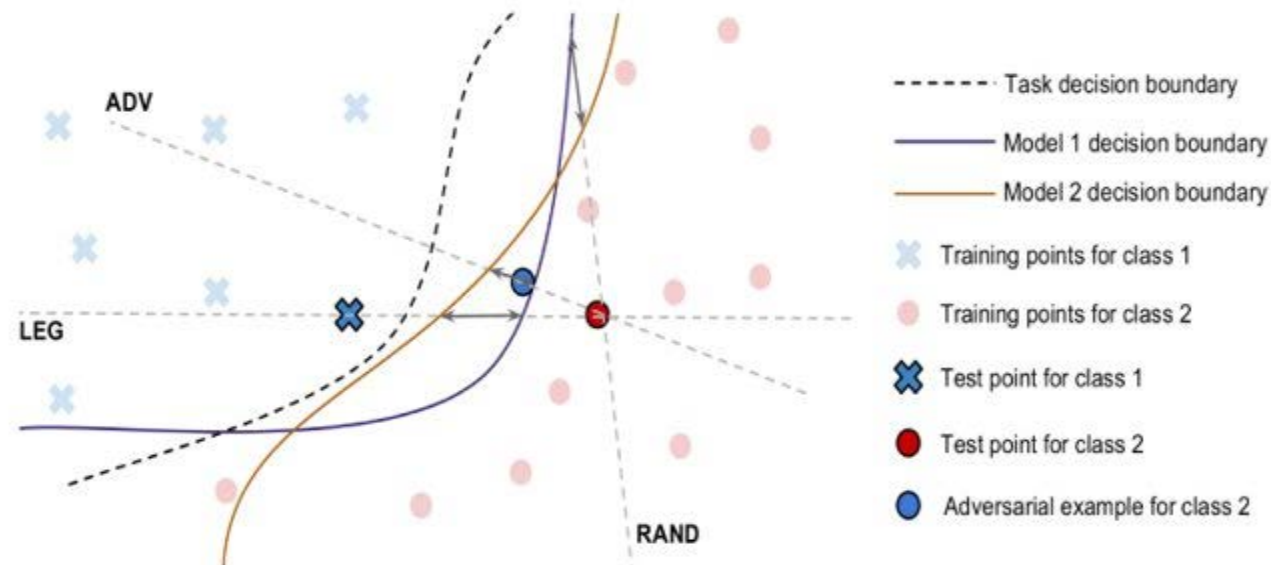
	Accuracy (%)				
# annotations	100	500	800	1000	All
DFAL	84.23	91.52	93.16	93.91	–
DFAL_0	78.62	91.35	92.44	93.14	–
BALD	82.00	89.94	91.92	92.87	–
CEAL	64.45	79.66	85.73	88.65	–
CORE-SET	66.71	89.93	92.28	92.62	–
EGL	63.12	86.80	90.06	90.06	–
uncertainty	52.77	88.05	89.31	91.03	–
RANDOM	78.28	88.13	89.71	89.94	96.75

% of Test accuracy
QuickDraw (VGG8)

TRANSFERABILITY: The ultimate threat of adversarial examples [Tramèr, 2017]

Adversarial space: contiguous, at least 2 dimensional. Dimension is proportional to the ratio increase in loss / perturbation

Different models with similar class boundary distances





DFAL EXPERIMENTS 3/3

- Transferability of non targeted adversarial attacks
- 1000 queries
- Model Selection
- **|MNIST| = 60 000 |ShoeBag|=184 792**

	DFAL	CORE-SET	RANDOM
LeNet5→ VGG8	97.80	96.90	94.46
VGG8→ LeNet5	97.93	97.40	95.31

% of Test accuracy

MNIST

	DFAL	CORE-SET	RANDOM
LeNet5→ VGG8	99.40	99.12	97.08
VGG8→ LeNet5	98.75	98.50	98.07

% of Test accuracy

Shoe Bag

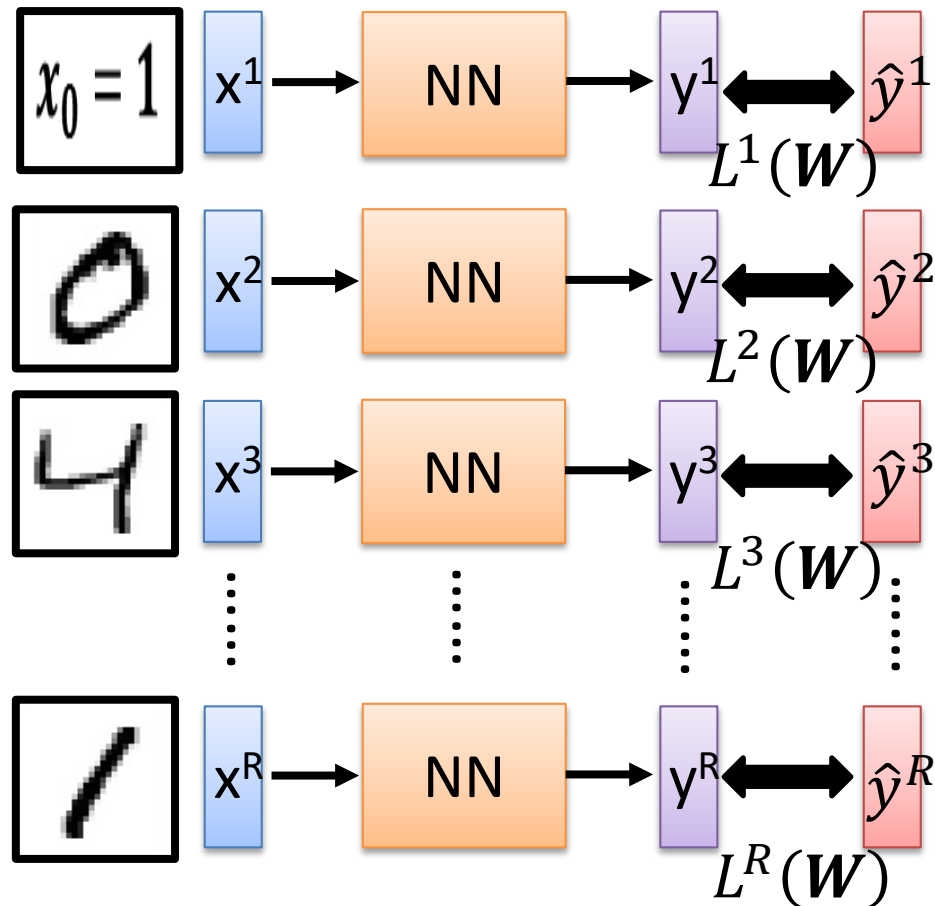
Based on Lectures from Hung-yi Lee (Laboratory Speech Processing and Machine Learning Laboratory, National Taiwan University)

OPTIMIZATION

Recipe of Deep Learning

• *Total Cost?*

For all training data ...



Total Cost:

$$C(W) = \sum_{r=1}^R L^r(W)$$

How bad the network parameters W is on this task

Find the network parameters W^* that minimize this value

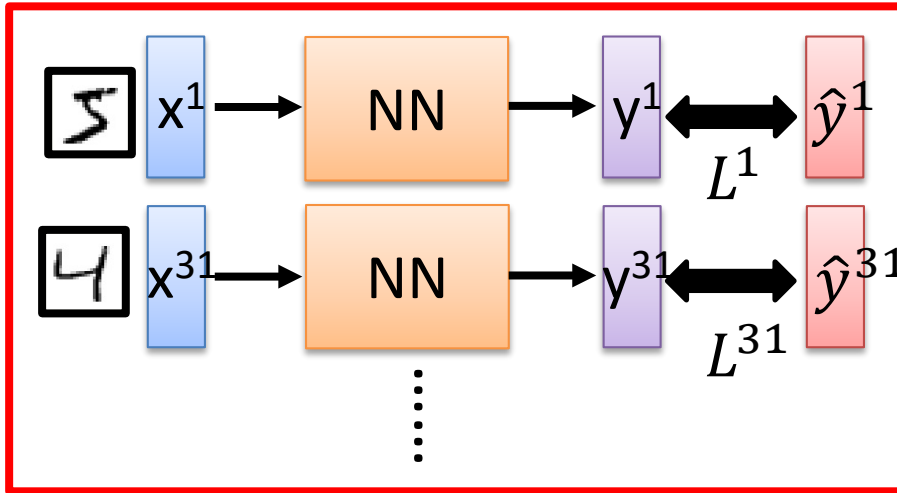
Recipe of Deep Learning

- *Mini-batches*

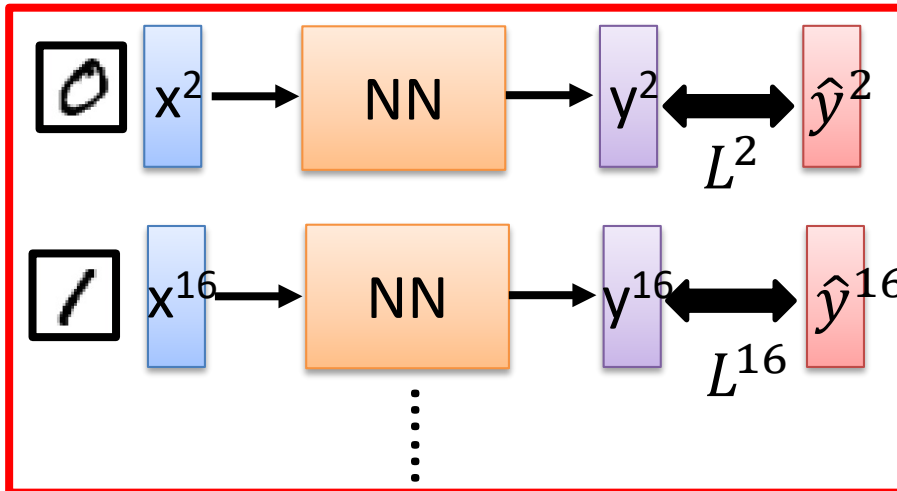
Faster

Better!

Mini-batch



Mini-batch



- Randomly initialize W^0

- Pick the 1st batch

$$C = L^1 + L^{31} + \dots$$

$$W^1 \leftarrow W^0 - \eta \nabla C(W^0)$$

- Pick the 2nd batch

$$C = L^2 + L^{16} + \dots$$

$$W^2 \leftarrow W^1 - \eta \nabla C(W^1)$$

 \vdots

- Until all mini-batches have been picked

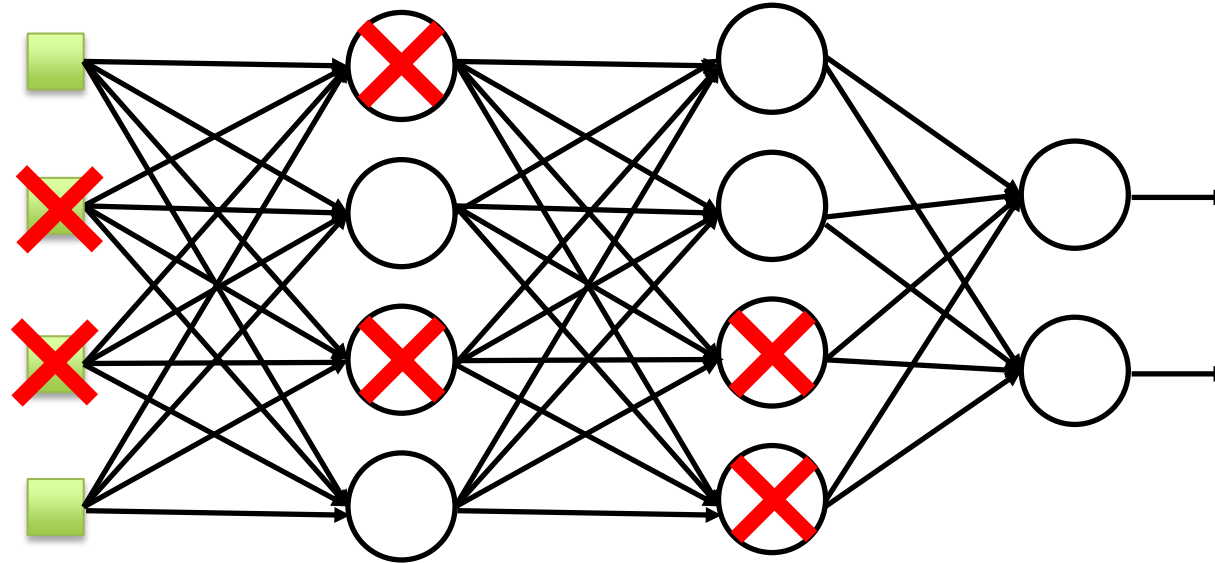
one epoch

Repeat the above process

Recipe of Deep Learning

- *Dropout*

Training:



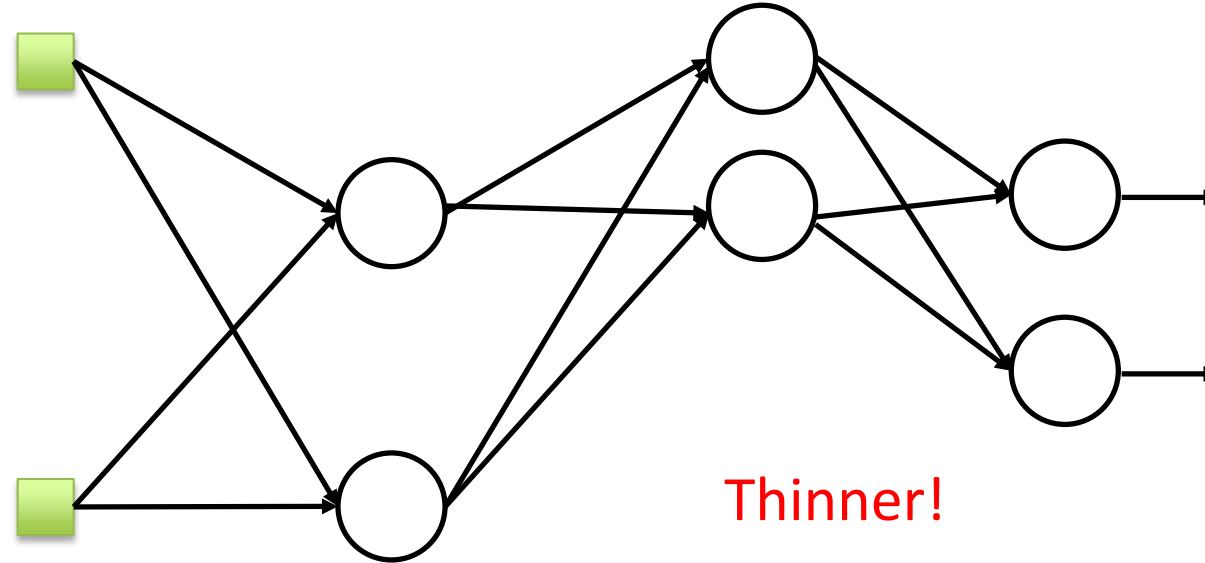
- Each time before updating the parameters
 - Each neuron has $p\%$ to dropout



Recipe of Deep Learning

- *Dropout*

Training:



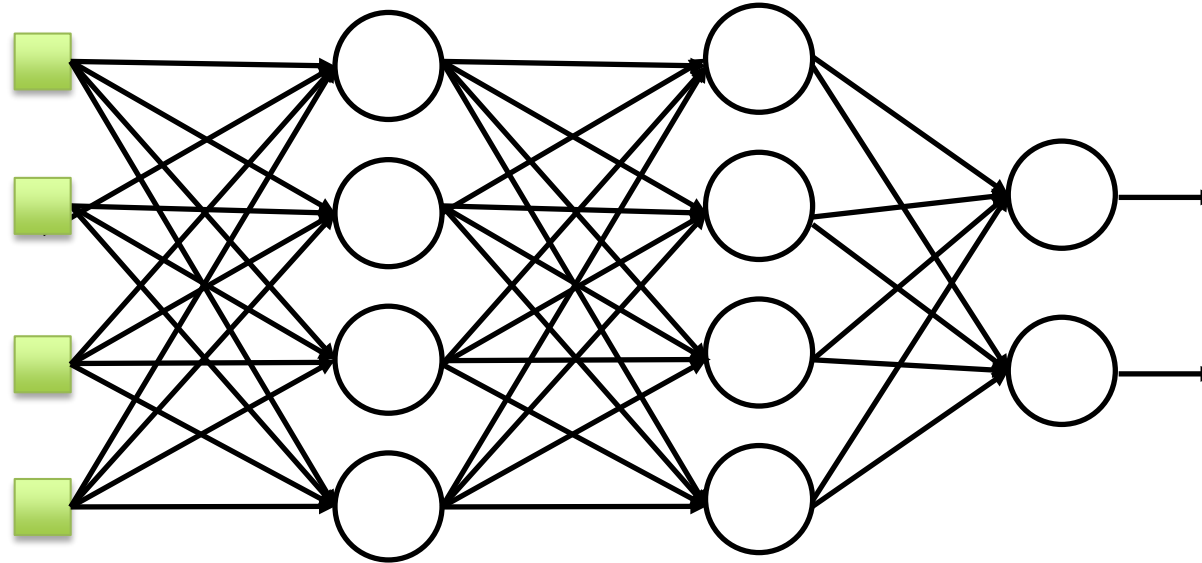
- Each time before updating the parameters
 - Each neuron has $p\%$ to dropout
 - ➡ **The structure of the network is changed.**
 - Using the new network for training

For each mini-batch, we resample the dropout neurons

Recipe of Deep Learning

- *Dropout*

Testing:

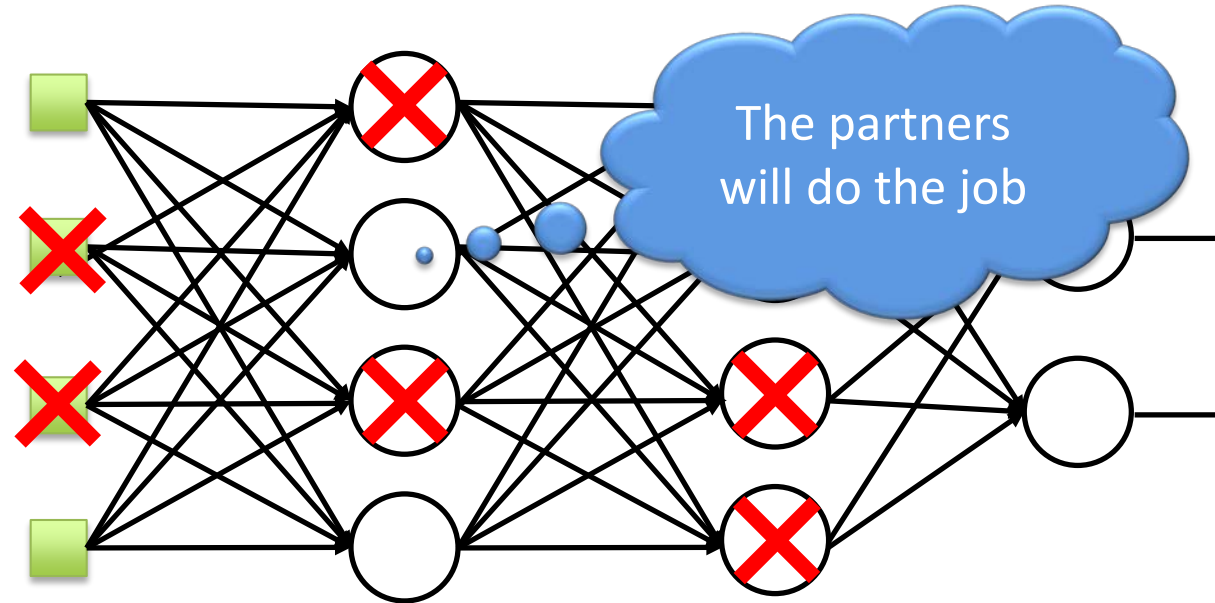


➤ **No dropout**

- If the dropout rate at training is $p\%$, all the weights times $1-p\%$
- Assume that the dropout rate is 50%.
If a weight $w = 1$ by training, set $w = 0.5$ for testing.

Recipe of Deep Learning

- *Dropout*



- When teams up, if everyone expect the partner will do the work, nothing will be done finally.
- However, if you know your partner will dropout, you will do better.
- When testing, no one dropout actually, so obtaining good results eventually.

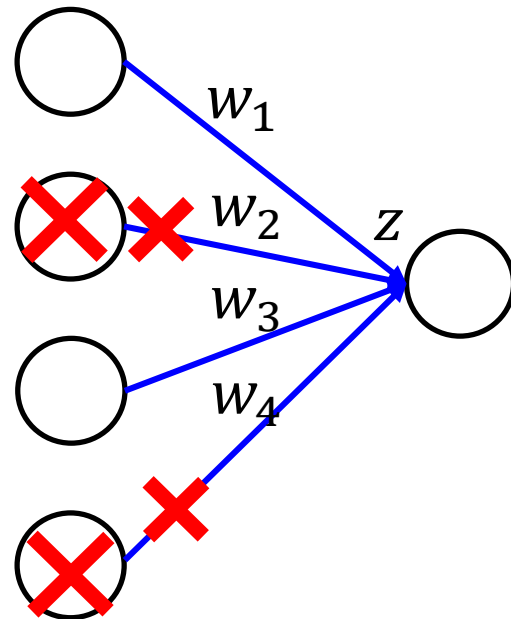
Recipe of Deep Learning

- *Dropout – intuitive reason*

- Why the weights should multiply $(1-p)\%$ (dropout rate) when testing?

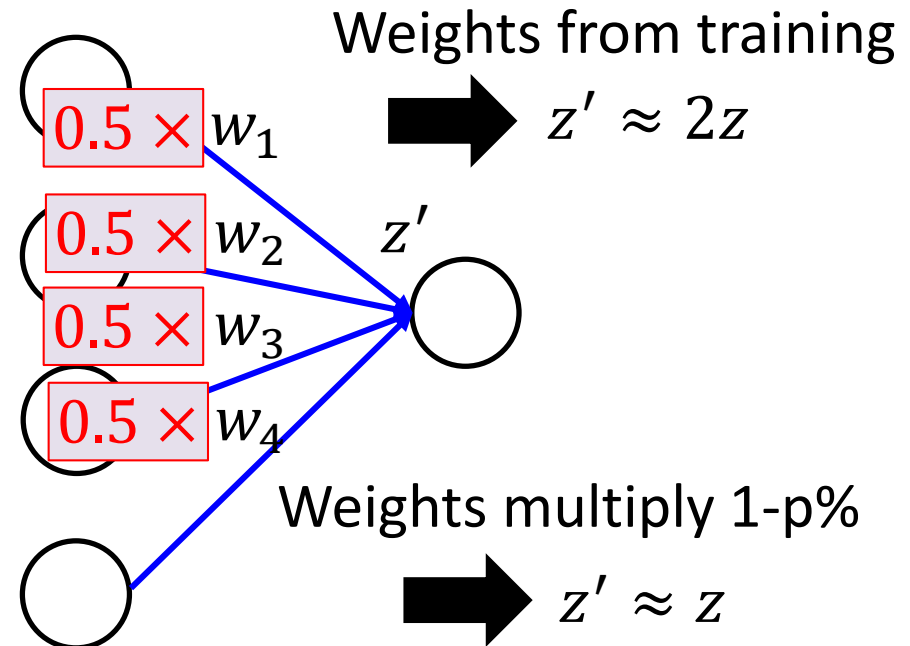
Training of Dropout

Assume dropout rate is 50%



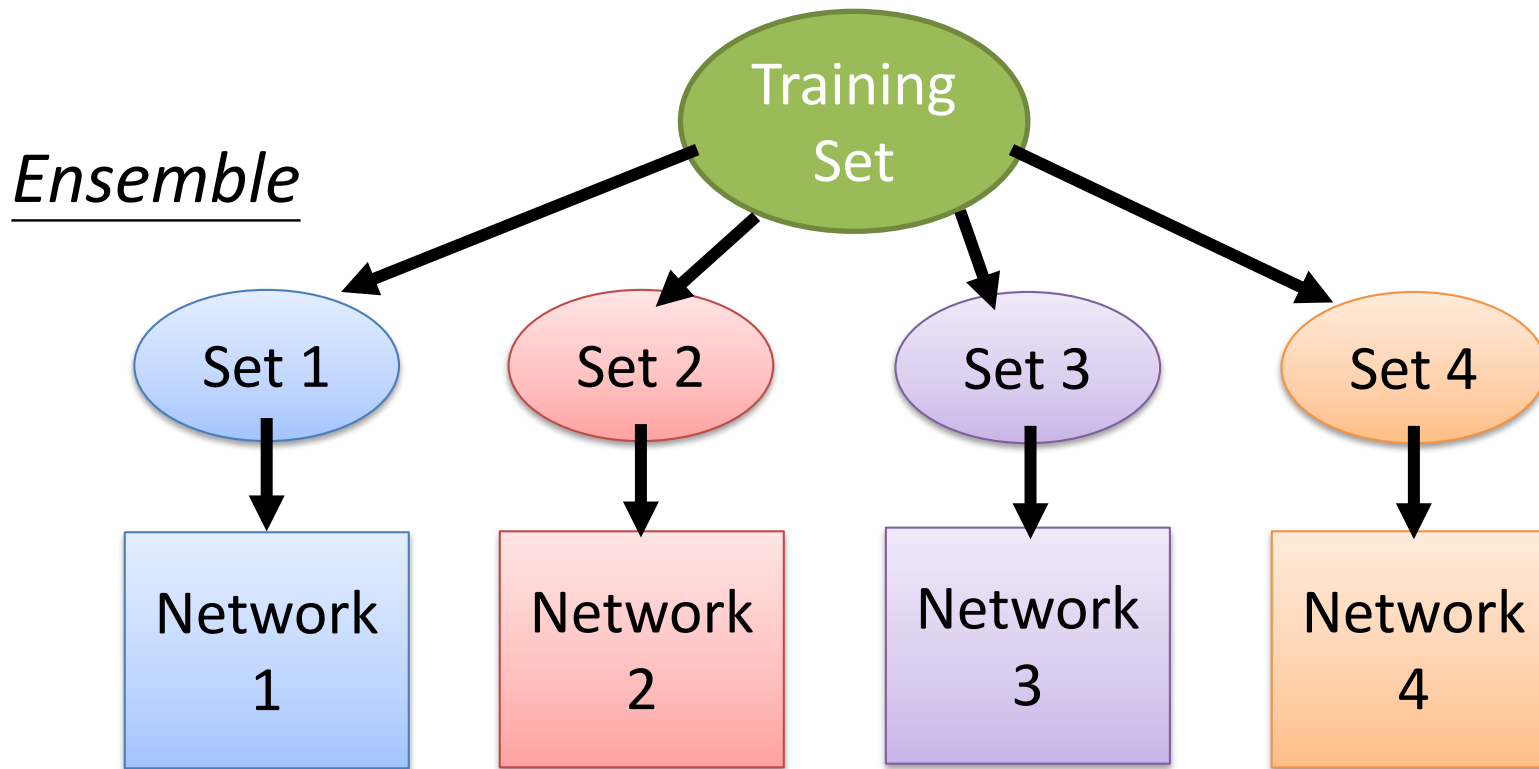
Testing of Dropout

No dropout



Recipe of Deep Learning

- *Dropout is a kind of ensemble*

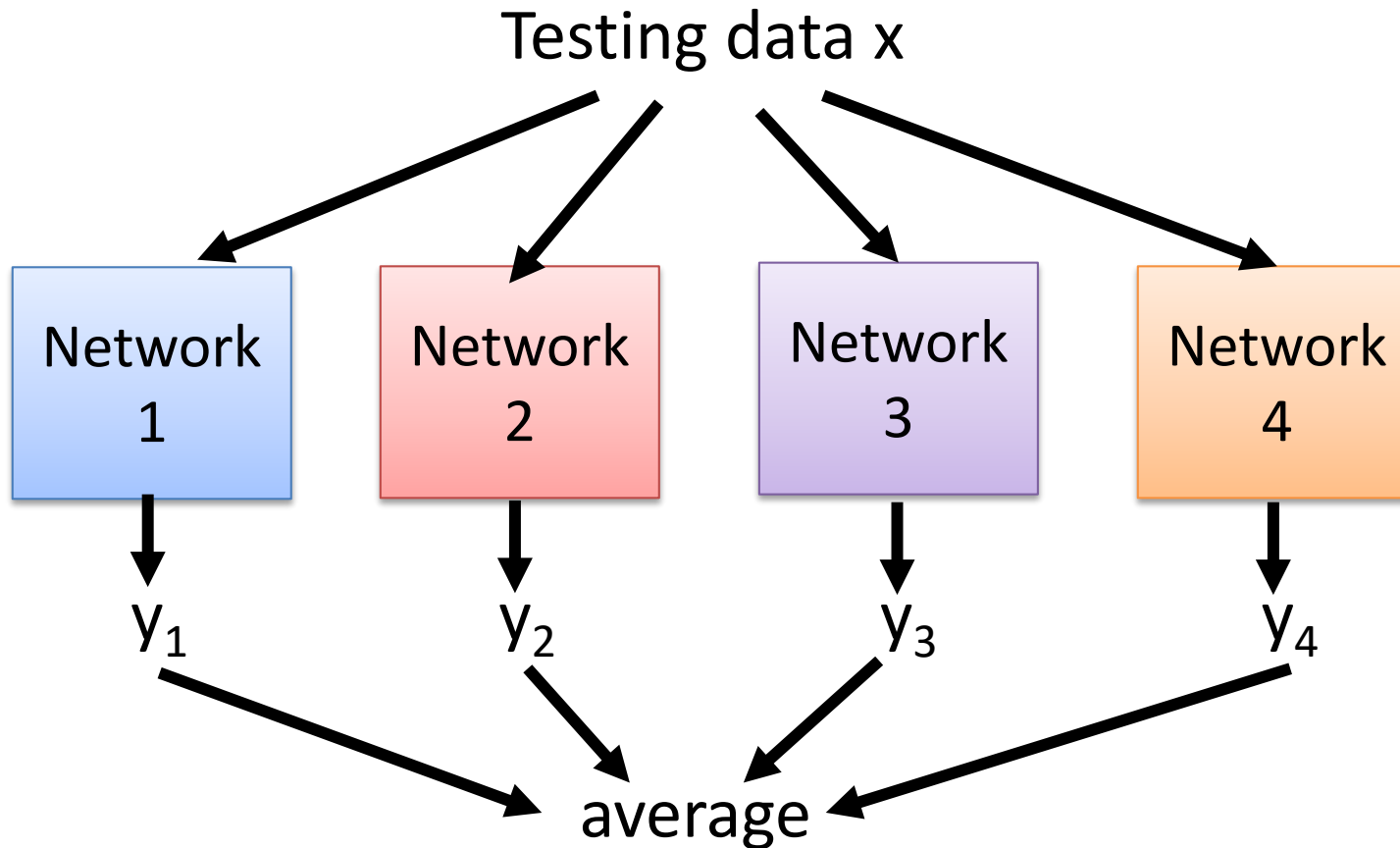


Train a bunch of networks with different structures

Recipe of Deep Learning

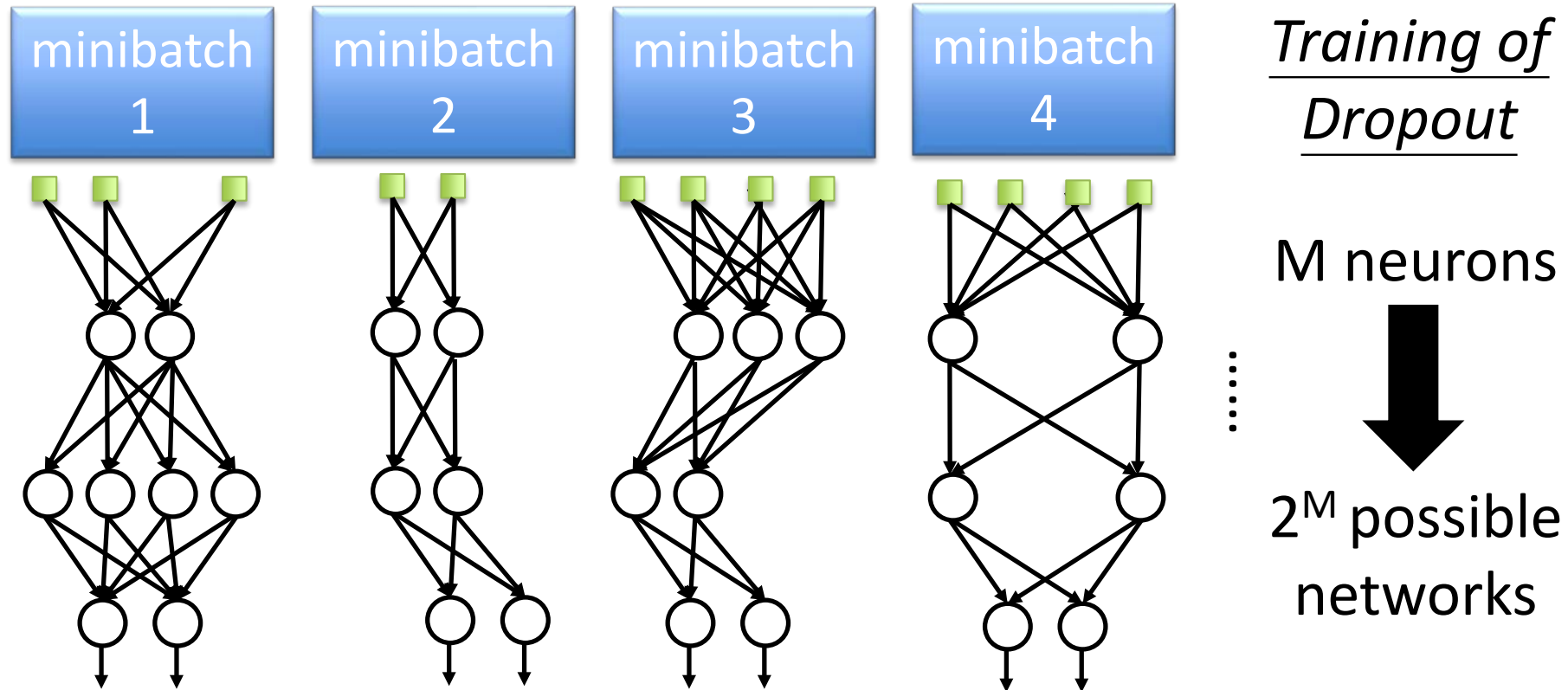
- *Dropout is a kind of ensemble*

Ensemble



Recipe of Deep Learning

- *Dropout is a kind of ensemble*

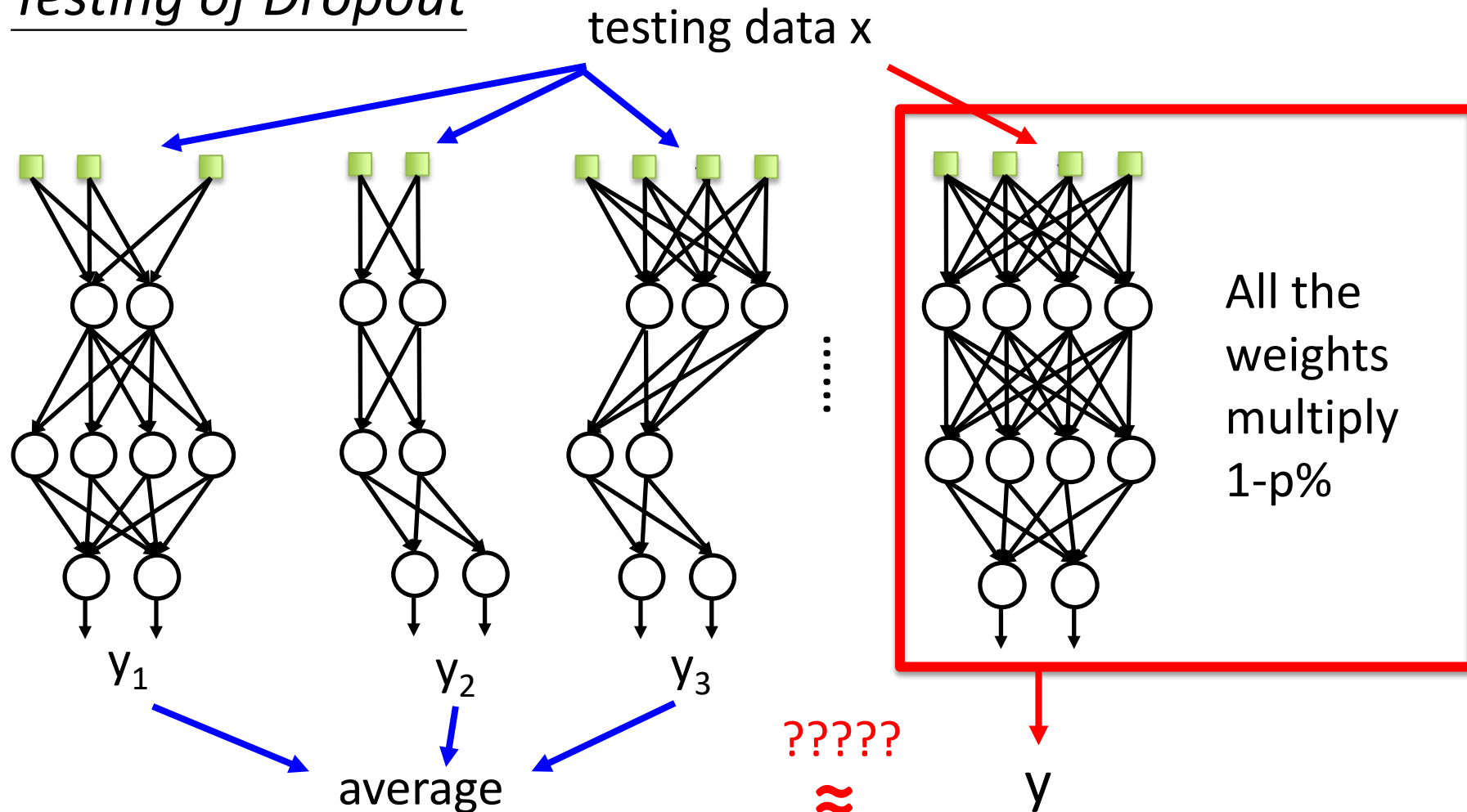


- Using one mini-batch to train one network
- Some parameters in the network are shared

Recipe of Deep Learning

- *Dropout is a kind of ensemble*

Testing of Dropout





univ-cotedazur.fr

Motivations: intialize and train the network

Real neural networks: e.g. the mammalian retina

- **initialized during developmental processes,**
such as spontaneous activation patterns

Those activation patterns are known to:

- **structure the network**
(retina and neural projections to the cortex)
- **pre-train the system by mimicing natural stimulations** and
lead to the emergence of gabor-like filters in V1 cortex area.

Simulated voltage map



Question:

Does artificial network can be pre-trained with images that mimic natural content ?

Biology mimicing for unsupervised pretraining

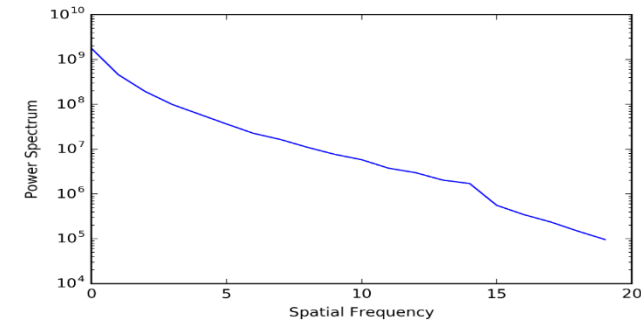
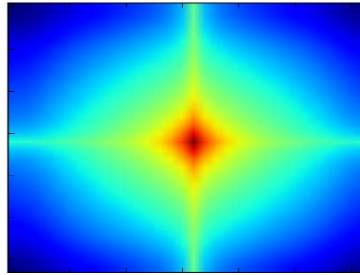
Mimic natural content:

Natural images



examples from STL-10

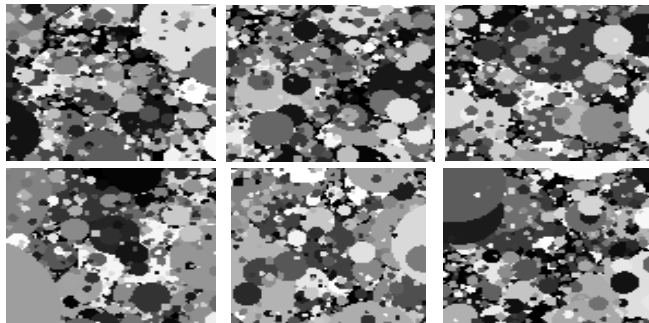
Average power spectrum



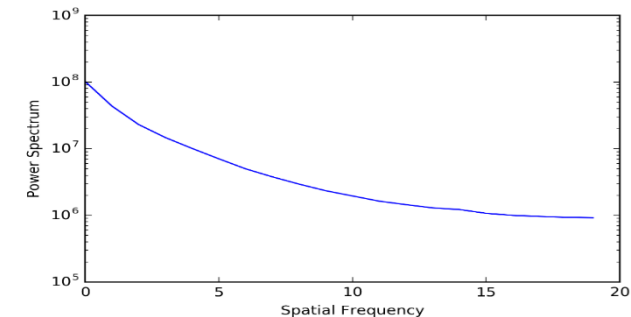
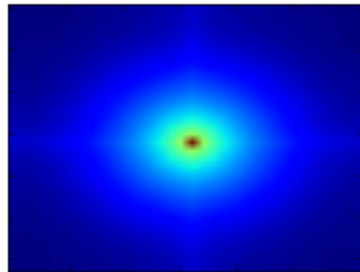
Synthetic images

Deadleaves model (DL): scale invariant, $1/f$ frequency distribution

Lee et al, Int Journal of Computer Vision, 2001

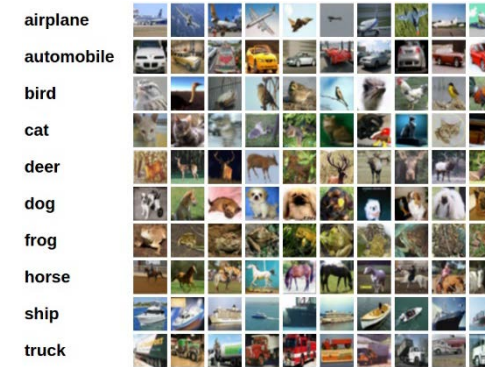


Average power spectrum

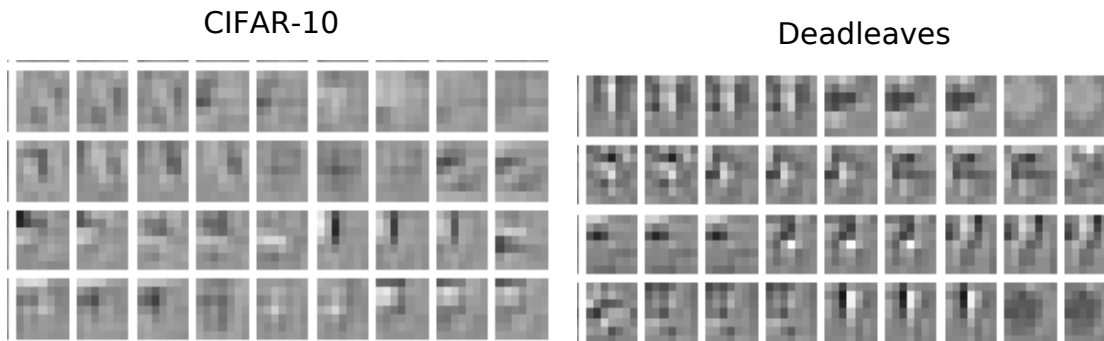


Experiments: Supervised training performances of a CNN, with or without unsupervised pre-training using Deadleaves

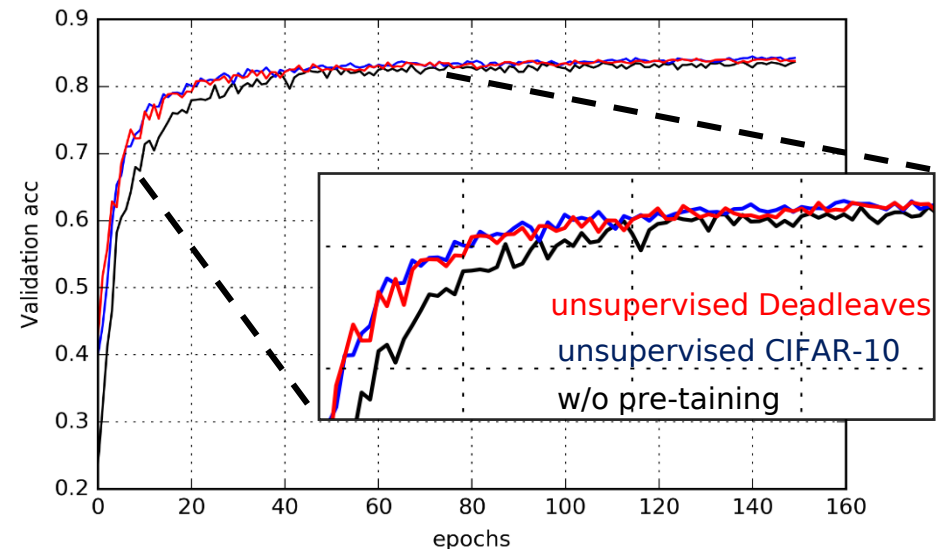
CIFAR-10: 10 classes, 50k training images, 10k test images, 32x32 px.



Learned filters during unsupervised pre-training, at the first layer of CNN:

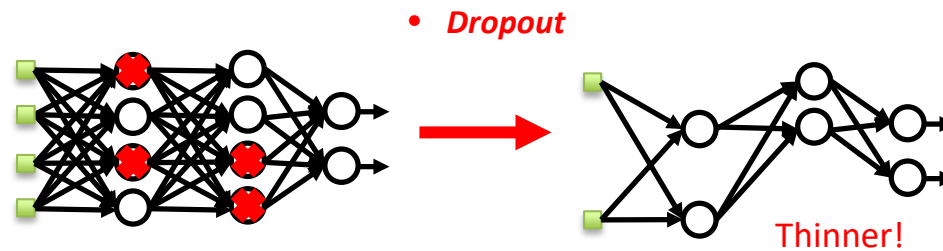


Training performances:

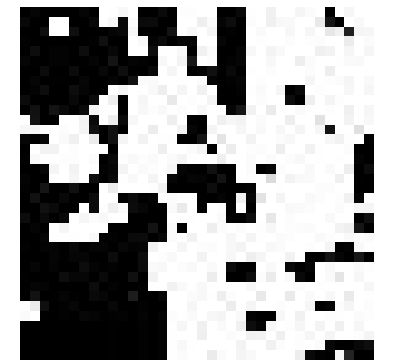


Biology mimicing dropout

- Bio-inspired optimization trick:



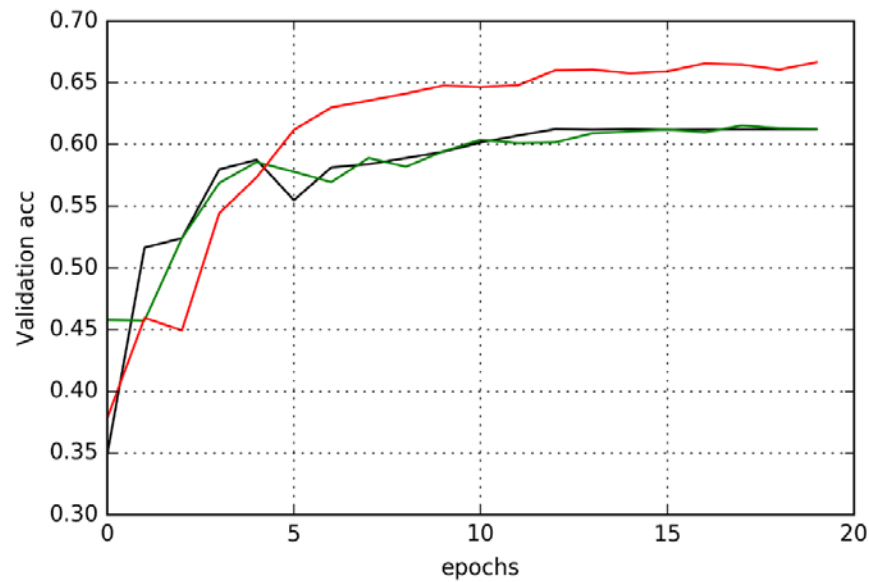
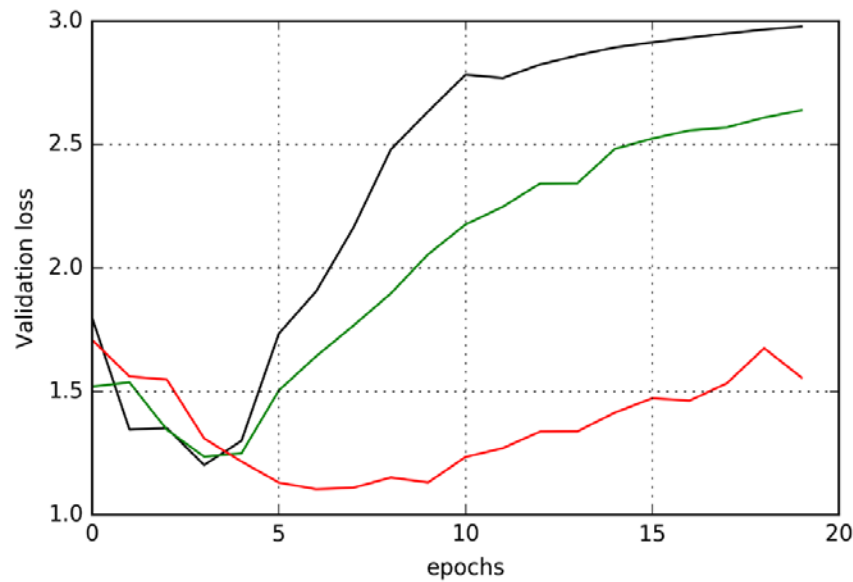
- * **Classical dropout:** dropout **randomly selects hidden unit activities to zero** with a given probability (from 0.2 to 0.5) during training.
- * Dropout probabilities follow a **binomial distribution**.
- * But one could suggest that **refining the probability distribution could lead to improvements** in the learning of deep architectures.
- *As **biology mimicing model can capture the main statistics of natural images**, one could use **retinal waves as dropout layer** in deep architectures.



Preliminary results:

Supervised training with CIFAR-10 following 3 conditions :

- w/o dropout
- classical dropout ($p=0.2$)
- dropout deadleaves





Biology mimicking dropout

univ-cotedazur.fr

Preliminary results:

Supervised training with CIFAR-10 following 3 conditions :

- classical dropout ($p=0.2$)
- dropout deadleaves
- dropout wave

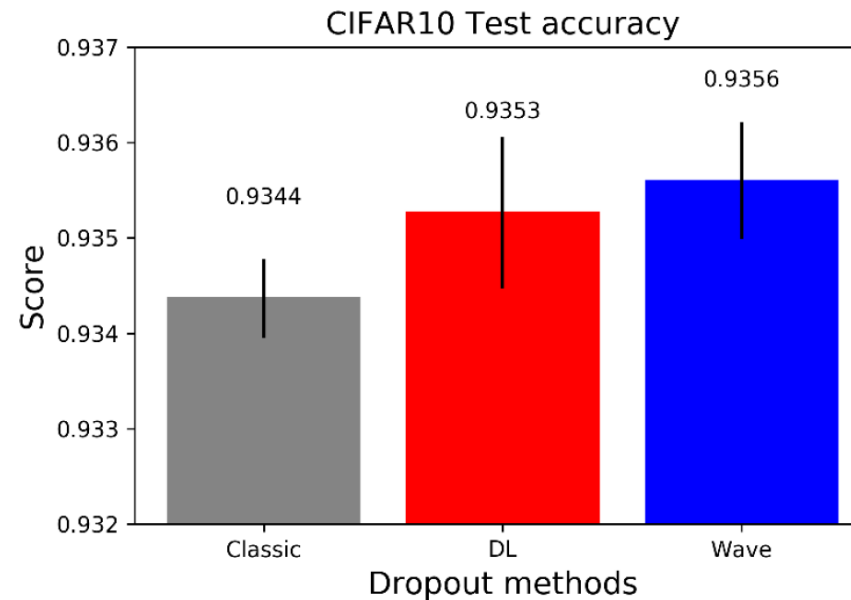


Figure 5: DenseNet test score

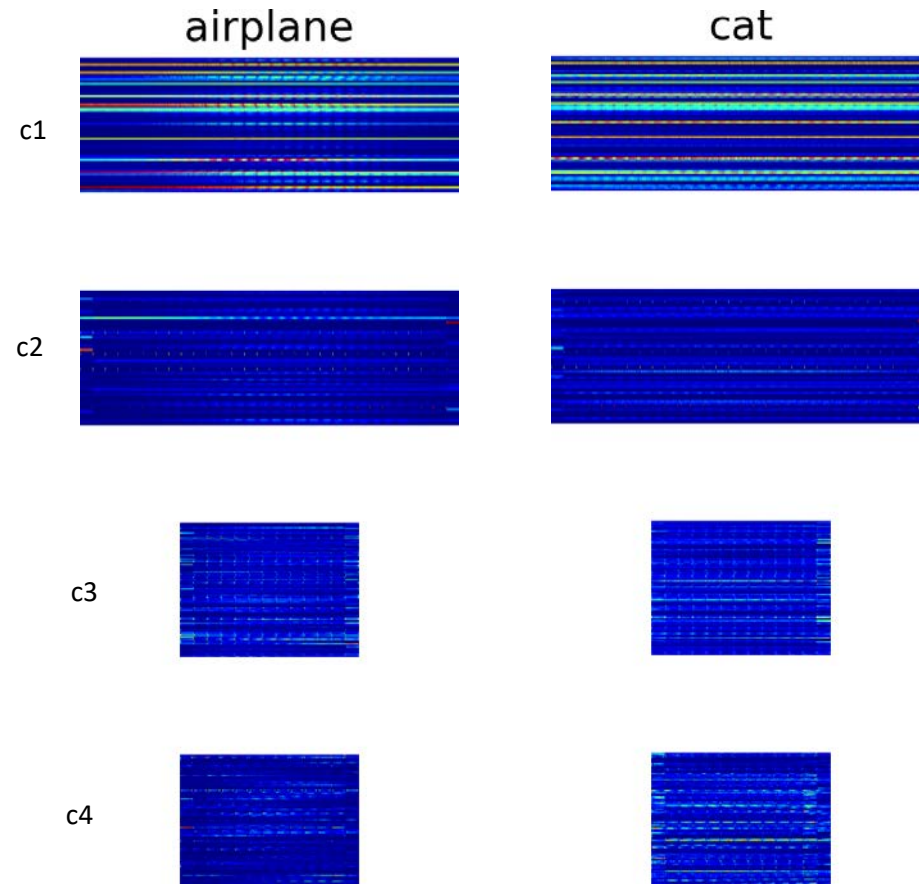
In biology, a given stimulus will activate a given neural subpopulation.

Hypotheses : in artificial neural network

- A given image would activate a specific subpopulation of artificial neurons (maps of activity)
- Hints of adversarial attacks could be tracked in the activity maps across the layers

Activity maps:

average across the category



Distinct categories have distinct activity maps across the network

Biology robustness to adversarial examples

Activity maps:
average across the category

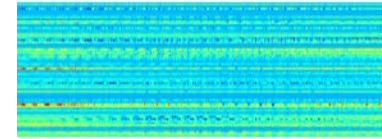
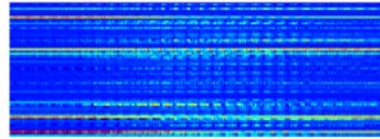
True images - adversarial images
recognized as the same category

Work in progress: identification and
quantification of differences between real
activities and adversarial activities

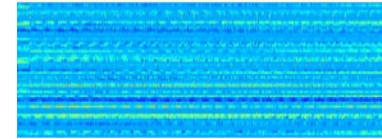
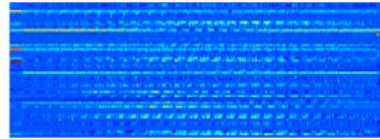
Airplane = (true - adv)

Cat = (true - adv)

c1

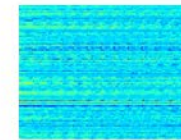
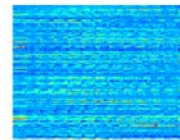


c2

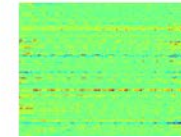
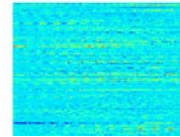


Szegedy attack

c3



c4





Adaboost to build Deep architectures: AdaNet

- AdaNet's approach is to optimize an objective that balances the trade-offs between the ensemble's performance on the training set and its ability to generalize to unseen data.
- The intuition is for the ensemble to include a candidate subnetwork only when it improves the ensemble's training loss more than it affects its ability to generalize.
- This guarantees that:
 - The generalization error of the ensemble is bounded by its training error and complexity.
 - By optimizing this objective, we are directly minimizing this bound.





Adaboost to build Deep architectures: AdaNet

- Block coordinate descent applied to convex objective: at each iteration,
 - a base subnetwork is selected (direction)
 - next, best step chosen by solving a convex optimization problem
- Convergence guarantees based on weak-learning assumption:
 - each network augmentation improves objective by a constant amount (-optimality condition) (*Raetsch et al., 2001; Luo & Tseng, 1992*)



Amazing, but...be careful of a little bias at the input

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings

Tolga Bolukbasi¹, Kai-Wei Chang², James Zou², Venkatesh Saligrama^{1,2}, Adam Kalai²

¹Boston University, 8 Saint Mary's Street, Boston, MA

²Microsoft Research New England, 1 Memorial Drive, Cambridge, MA

tolgab@bu.edu, kw@kwchang.net, jamesyzou@gmail.com, srv@bu.edu, adam.kalai@microsoft.com

From Nello Cristianini, at *Frontier Research and Artificial Intelligence Conference*:

https://erc.europa.eu/sites/default/files/events/docs/Nello_Cristianini-ThinkBIG-Patterns-in-Big-Data.pdf

Amazing, but...be careful of a little bias at the input

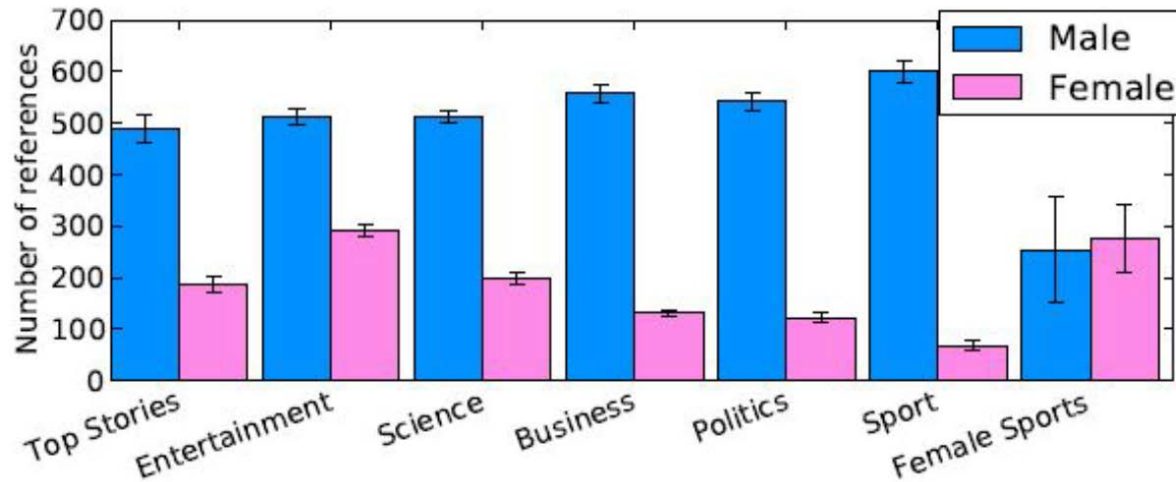


TABLE I: List of the top 10 occupations per gender by their association with gender.

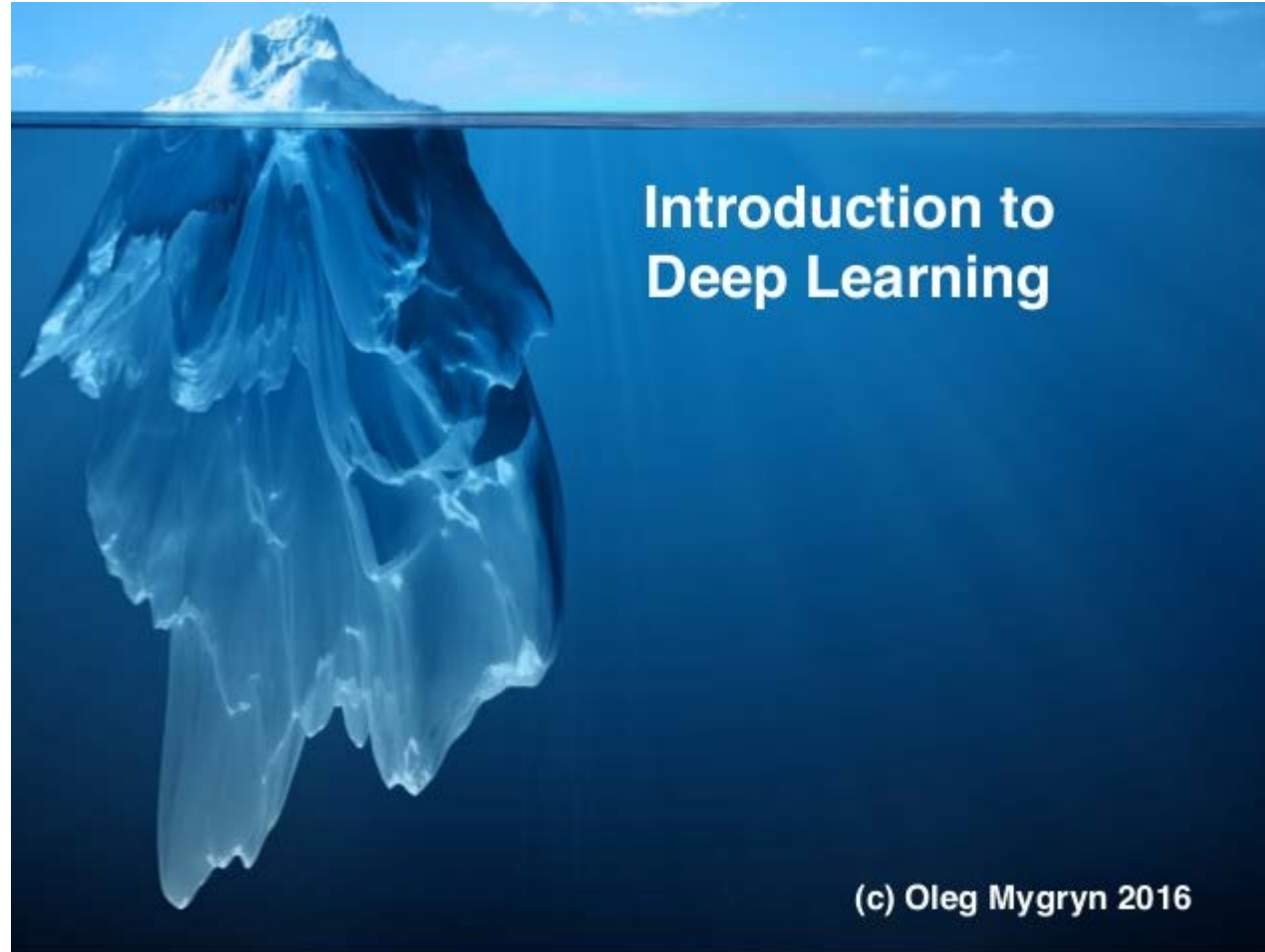
Gender	Occupations most associated with a gender
Male	Manager, Engineer, Coach, Executive, Surveyor, Secretary, Architect, Driver, Police, Caretaker, Director
Female	Housekeeper, Nurse, Therapist, Bartender, Psychologist, Designer, Pharmacist, Supervisor, Radiographer, Underwriter

From Nello Cristianini, at *Frontier Research and Artificial Intelligence Conference*:

https://erc.europa.eu/sites/default/files/events/docs/Nello_Cristianini-ThinkBIG-Patterns-in-Big-Data.pdf



This was just scratching the tip of the Deep Learning Iceberg



Theoretical understanding of Deep Networks is progressing constantly

How Does Batch Normalization Help Optimization?

Shibani Santurkar*
MIT
shibani@mit.edu

Dimitris Tsipras*
MIT
tsipras@mit.edu

Andrew Ilyas*
MIT
ailyas@mit.edu

Aleksander Mądry
MIT
madry@mit.edu

A Capacity Scaling Law for Artificial Neural Networks

Gerald Friedland*, Mario Michael Krell†
friedland1@llnl.gov, krell@icsi.berkeley.edu

September 5, 2018

UNDERSTANDING DEEP LEARNING REQUIRES RE-THINKING GENERALIZATION

Chiyan Zhang*
Massachusetts Institute of Technology
chiyan@mit.edu

Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

Benjamin Recht†
University of California, Berkeley
brecht@berkeley.edu

Oriol Vinyals
Google DeepMind
vinyals@google.com

Gradient Descent Finds Global Minima of Deep Neural Networks

Simon S. Du*¹, Jason D. Lee*², Haochuan Li*^{1,3,5}, Liwei Wang*^{4,5}, and Xiyu Zhai*⁶

¹Machine Learning Department, Carnegie Mellon University

²Data Science and Operations Department, University of Southern California

³School of Physics, Peking University

⁴Key Laboratory of Machine Perception, MOE, School of EECS, Peking University

⁵Center for Data Science, Peking University, Beijing Institute of Big Data Research

⁶Department of EECS, Massachusetts Institute of Technology

February 5, 2019

AdaNet: Adaptive Structural Learning of Artificial Neural Networks

Corinna Cortes¹ Xavier Gonzalvo¹ Vitaly Kuznetsov¹ Mehryar Mohri^{2,1} Scott Yang²

A Closer Look at Memorization in Deep Networks

Devansh Arpit*^{1,2} Stanisław Jastrzębski*³ Nicolas Ballas*^{1,2} David Krueger*^{1,2} Emmanuel Bengio⁴
Maxinder S. Kanwal⁵ Tegan Maharaj^{1,6} Asja Fischer⁷ Aaron Courville^{1,2,8} Yoshua Bengio^{1,2,9}
Simon Lacoste-Julien^{1,2}

QUESTIONS?