

# *INTELLIGENZA ARTIFICIALE*

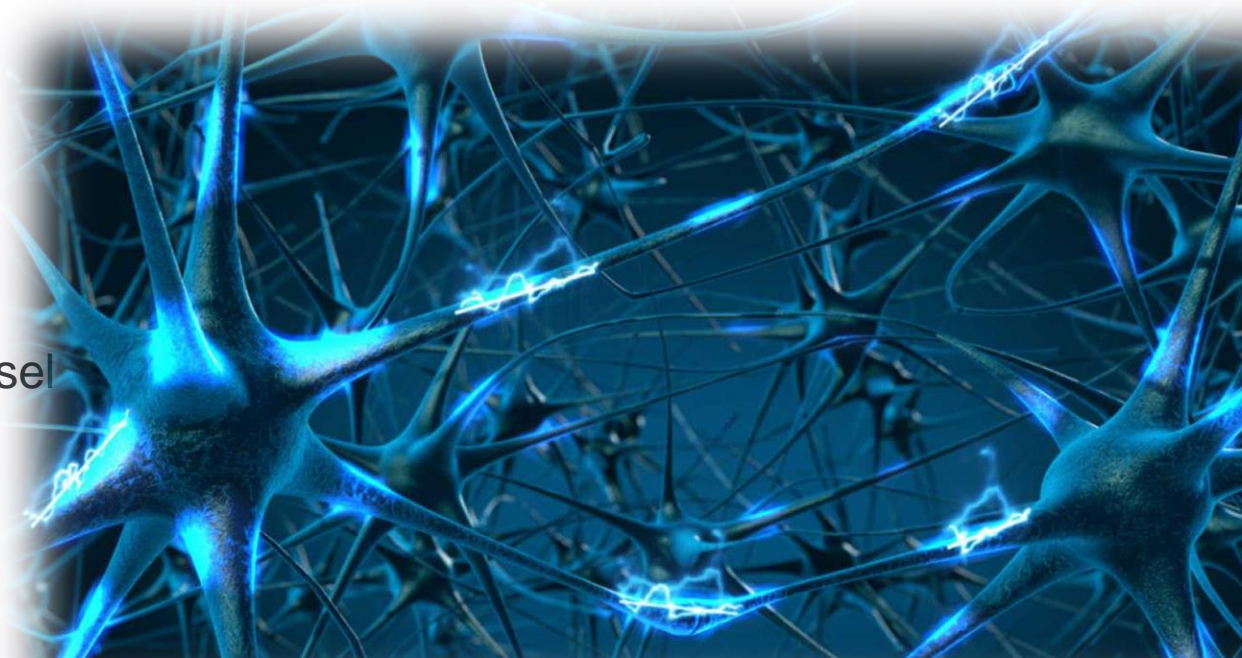
## *INTRODUZIONE ALLE RETI NEURALI*

---

Corsi di Laurea in Informatica, Ing. Gestionale, Ing. Informatica,  
Ing. di Internet  
(a.a. 2023-2024)

Roberto Basili

(\*) dalle *slides* di S. Russel



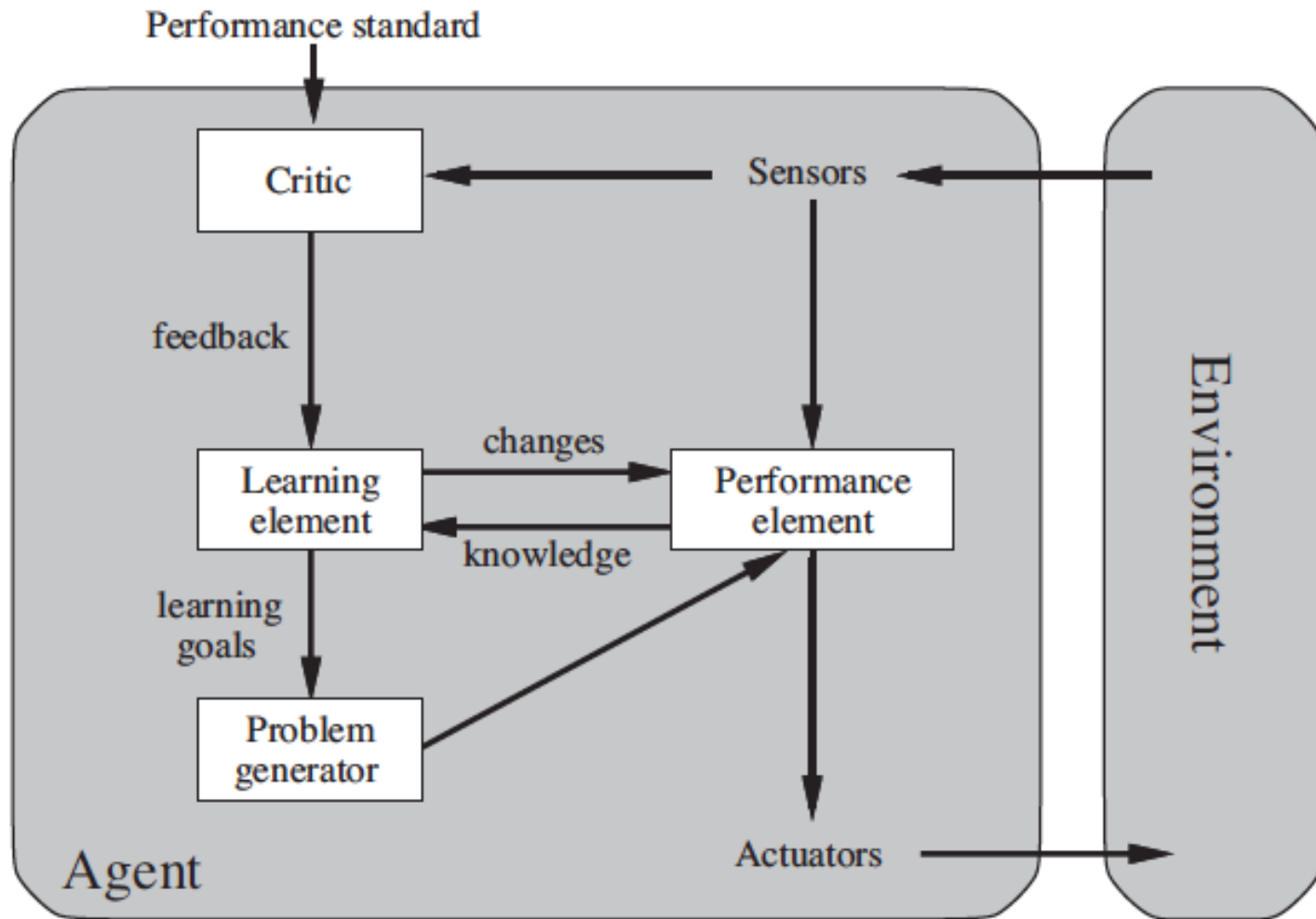
# Recap (AIMA chpt. 18.1-18.4)

- Agents & machine learning
- Learning from examples:
  - Complexity and Expressiveness
  - The definition of model selection
- An example: Decision Tree learning
  - Recursive search among Boolean formulas
  - Attribute Selection in DT: Information Gain
- Learning methodology: design, experiment/  
evaluation and model selection
  - Cross validation

# Introduction to machine learning

- Introduction to machine learning
  - When appropriate and when not appropriate
  - Task definition
- Learning methodology: design, experiment, evaluation
- Learning issues: representing hypothesis
- Learning paradigms
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning

# AIMA learning architecture



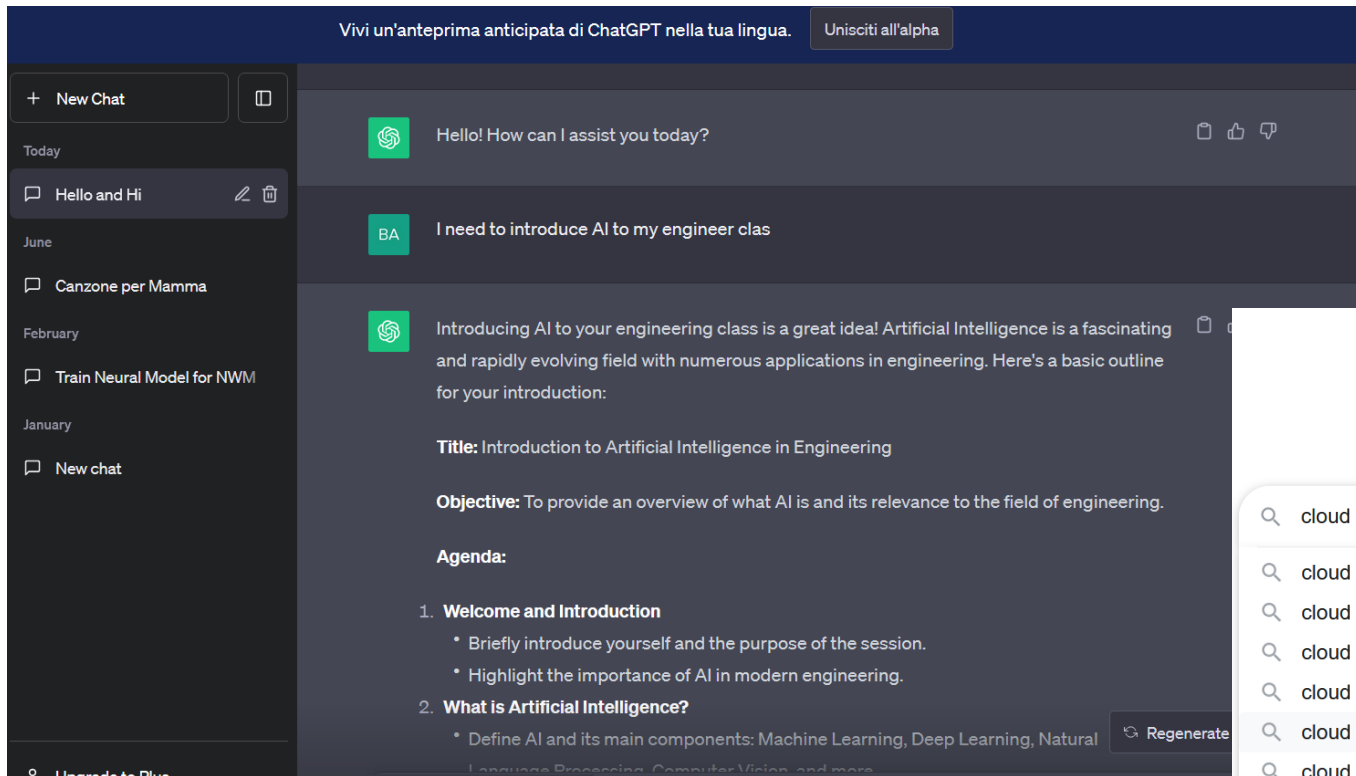
# Machine learning: definition

- *A computer program is said to **learn** from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$  [Mitchell]*
- Problem definition for a learning agent
  - Task  $T$
  - Performance measure  $P$
  - Experience  $E$

# ESEMPI DI APPLICAZIONI DI RETI NEURALI

---

# Esempi della nozione di "Task"



Vivi un'anteprima anticipata di ChatGPT nella tua lingua. Unisciti all'alpha

+ New Chat

Today

Hello and Hi

June

Canzone per Mamma

February

Train Neural Model for NWM

January

New chat

Upgrade to Plus

Hello! How can I assist you today?

BA I need to introduce AI to my engineer clas

Introducing AI to your engineering class is a great idea! Artificial Intelligence is a fascinating and rapidly evolving field with numerous applications in engineering. Here's a basic outline for your introduction:

**Title:** Introduction to Artificial Intelligence in Engineering

**Objective:** To provide an overview of what AI is and its relevance to the field of engineering.

**Agenda:**

- Welcome and Introduction**
  - Briefly introduce yourself and the purpose of the session.
  - Highlight the importance of AI in modern engineering.
- What is Artificial Intelligence?**
  - Define AI and its main components: Machine Learning, Deep Learning, Natural Language Processing, Computer Vision, and more.

Regenerate



cloud container

cloud container

cloud containers

cloud container **security**

cloud container **services**

cloud container **engine**

cloud containers **examples**

cloud container **registry**

cloud container **distribution**

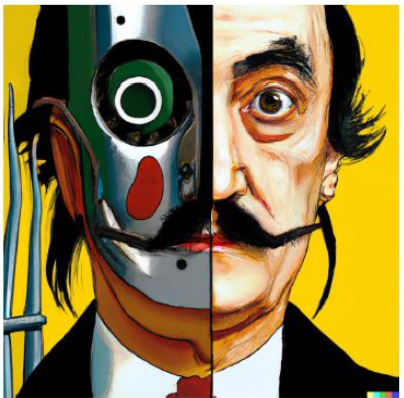
cloud container **isation**

cloud container **technologies**

Cerca con Google

Mi sento fortunato

# Altri Tasks (2)



vibrant portrait painting of Salvador Dalí with a robotic half face



a shiba inu wearing a beret and black turtleneck



a close up of a handpalm with leaves growing from it

 DALL-E History Collections

Edit the detailed description

Impressionist painting of a Web server



an espresso machine that makes coffee from human souls, artstation



panda mad scientist mixing sparkling chemicals, artstation



a corgi's head depicted as an explosion of a nebula





# Neural encoding-decoding for DALL-E

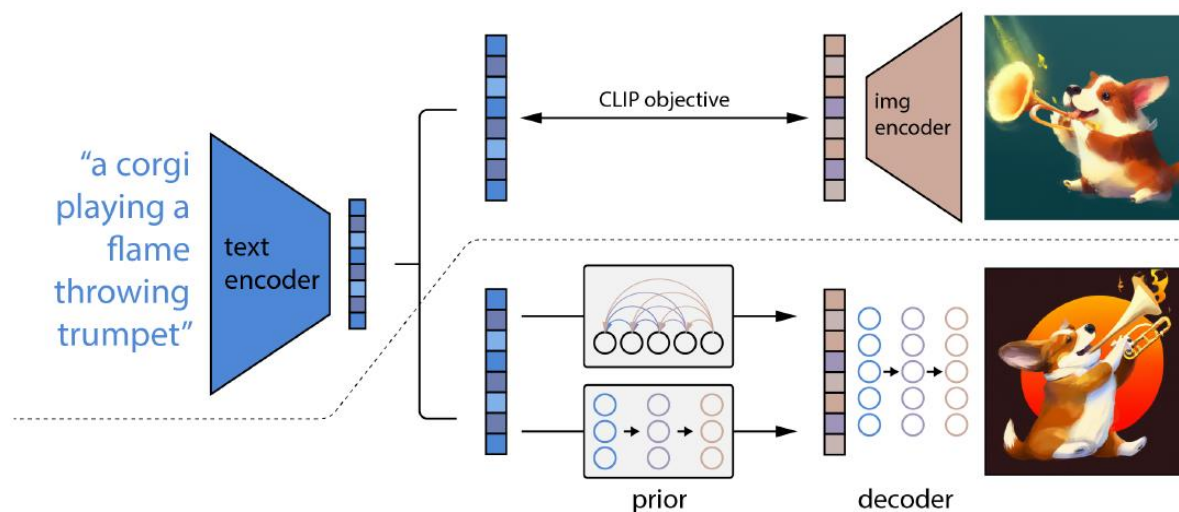


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive or diffusion prior to produce an image embedding, and then this embedding is used to condition a diffusion decoder which produces a final image. Note that the CLIP model is frozen during training of the prior and decoder.

# METODI DI MACHINE LEARNING

---

# Metodi di Machine Learning



## Artificial Intelligence Definitions

**Intelligence** might be defined as the ability to learn and perform suitable techniques to solve problems and achieve goals, appropriate to the context in an uncertain, ever-varying world. A fully pre-programmed factory robot is flexible, accurate, and consistent but not intelligent.

**Artificial Intelligence (AI)**, a term coined by emeritus Stanford Professor John McCarthy in 1955, was defined by him as “the science and engineering of making intelligent machines”. Much research has humans program software agents to behave in a clever way, like playing chess, but, today, we emphasize agents that can learn, as human beings navigating our changing world do.

**Autonomous systems** can independently plan and decide sequences of steps to achieve a specified goal without micro-management. A hospital delivery robot must autonomously navigate busy corridors to succeed in its task. In AI, autonomy doesn't have the sense of being self-governing that is common in politics or biology.

**Machine Learning (ML)** is the part of AI studying how computer systems can improve their perception, knowledge, decisions, or actions based on experience or data. For this, ML draws from computer science, statistics, psychology, neuroscience, economics, and control theory.

In **supervised learning**, a computer learns to predict human-given labels, such as dog breed based on labeled dog pictures; **unsupervised learning** does not require labels, sometimes making its own prediction tasks such as trying to predict each successive word in a sentence; **reinforcement learning** lets an agent learn action sequences that optimize its total rewards,

such as winning games, without explicit examples of good techniques, enabling autonomy.

**Deep Learning** is the use of large multi-layer (artificial) neural networks that compute with continuous (real number) representations, a little like the hierarchically-organized neurons in human brains. It is currently the most successful ML approach, usable for all types of ML, with better generalization from small data and better scaling to big data and compute budgets.

An **algorithm** lists the precise steps to take, such as a person writes in a computer program. AI systems contain algorithms, but often just for a few parts like a learning or reward calculation method. Much of their behavior emerges via learning from data or experience, which is a sea change in system design that Stanford alumnus Andrej Karpathy dubbed **Software 2.0**.

**Narrow AI** is intelligent systems for particular tasks, e.g., **speech** or **facial recognition**. **Human-level AI**, or **Artificial General Intelligence (AGI)**, seeks broadly intelligent, context-aware machines. It is needed for effective, adaptable **social chatbots** or **human-robot interaction**.

**Human-Centered Artificial Intelligence** is AI that seeks to augment the abilities of, address the societal needs of, and draw inspiration from human beings. It researches and builds effective partners and tools for people, such as a robot helper and companion for the elderly.

Text by Professor Christopher Manning, v 1.1, November 2020

Learn more at [hai.stanford.edu](http://hai.stanford.edu)

**Intelligence** might be defined as the ability to learn and perform suitable techniques to solve problems and achieve goals, appropriate to the context in an uncertain, ever-varying world. A fully pre-programmed factory robot is flexible, accurate, and consistent but not intelligent.

**Artificial Intelligence (AI)**, a term coined by emeritus Stanford Professor John McCarthy in 1955, was defined by him as “the science and engineering of making intelligent machines”. Much research has humans program software agents to behave in a clever way, like playing chess, but, today, we emphasize agents that can learn, as human beings navigating our changing world do.

**Autonomous systems** can independently plan and decide sequences of steps to achieve a specified goal without micro-management. A hospital delivery robot must autonomously navigate busy corridors to succeed in its task. In AI, autonomy doesn't have the sense of being self-governing that is common in politics or biology.

# Metodi di Machine Learning

**Machine Learning (ML)** is the part of AI studying how computer systems can improve their perception, knowledge, decisions, or actions based on experience or data. For this, ML draws from computer science, statistics, psychology, neuroscience, economics, and control theory.

In **supervised learning**, a computer learns to predict human-given labels, such as dog breed based on labeled dog pictures; **unsupervised learning** does not require labels, sometimes making its own prediction tasks such as trying to predict each successive word in a sentence; **reinforcement learning** lets an agent learn action sequences that optimize its total rewards,

**Deep Learning** is the use of large multi-layer **(artificial) neural networks** that compute with continuous (real number) representations, a little like the hierarchically-organized neurons in human brains. It is currently the most successful ML approach, usable for all types of ML, with better generalization from small data and better scaling to big data and compute budgets.

An **algorithm** lists the precise steps to take, such as a person writes in a computer program. AI systems contain algorithms, but often just for a few parts like a learning or reward calculation method. Much of their behavior emerges via learning from data or experience, which is a sea change in system design that Stanford alumnus Andrej Karpathy dubbed **Software 2.0**.

**Narrow AI** is intelligent systems for particular tasks, e.g., **speech** or **facial recognition**. **Human-level AI**, or **Artificial General Intelligence (AGI)**, seeks broadly intelligent, context-aware machines. It is needed for effective, adaptable **social chatbots** or **human-robot interaction**.

# Inductive learning

- Simplest form: learn a function from examples

$f$  is the **target function**

An **example** is a pair  $(x, f(x))$

Problem: find a **hypothesis**  $h$   
such that  $h \approx f$   
given a **training set** of examples

(This is a highly simplified model of real learning:

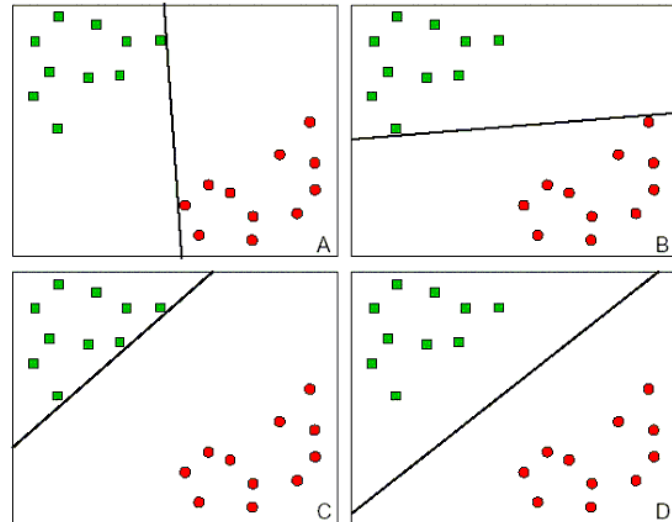
- Ignores prior knowledge
- Assumes examples are given)

# Metodi di ML: selezione dei modelli

- **Approcci discriminativi**

- Lineari

- $h(\mathbf{x}) = \text{sign}(\mathbf{W} \cdot \mathbf{x} + \mathbf{b})$



- **Approcci probabilistici**

- Stima delle probabilità  $p(\mathcal{C}_k|\mathbf{x})$  attraverso un training set
- Modello generativo ed uso della inversione Bayesiana

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}.$$

# Apprendimento e Classi di Algoritmi

- Acquisizione di:
  - Funzioni logiche booleane, (ad es., alberi di decisione)
  - Induzione: determinazione ricorsiva delle CNES che caratterizzano i diversi sottogruppi .
- Approcci probabilistici:
  - Funzione target di Probabilità, (ad es., classificatore Bayesiano)
  - Induzione: Stima delle probabilità (in quanto parametri).
- Approcci geometrici
  - Funzioni di separazione in spazi vettoriali (lineari e non)
    - KNN
    - Funzioni Lineari, perceptroni, Neural Networks, Support Vector Machines,...
    - Embeddings, analisi spettrale (trasformazioni di spazio)
  - Induzione: parametrizzare la funzione appartenente ad una certa classe (ad es. polinomi di grado  $n$ )

# ML: una introduzione visuale

- See URL: <http://www.r2d3.us/visual-intro-to-machine-learning-part-1/>



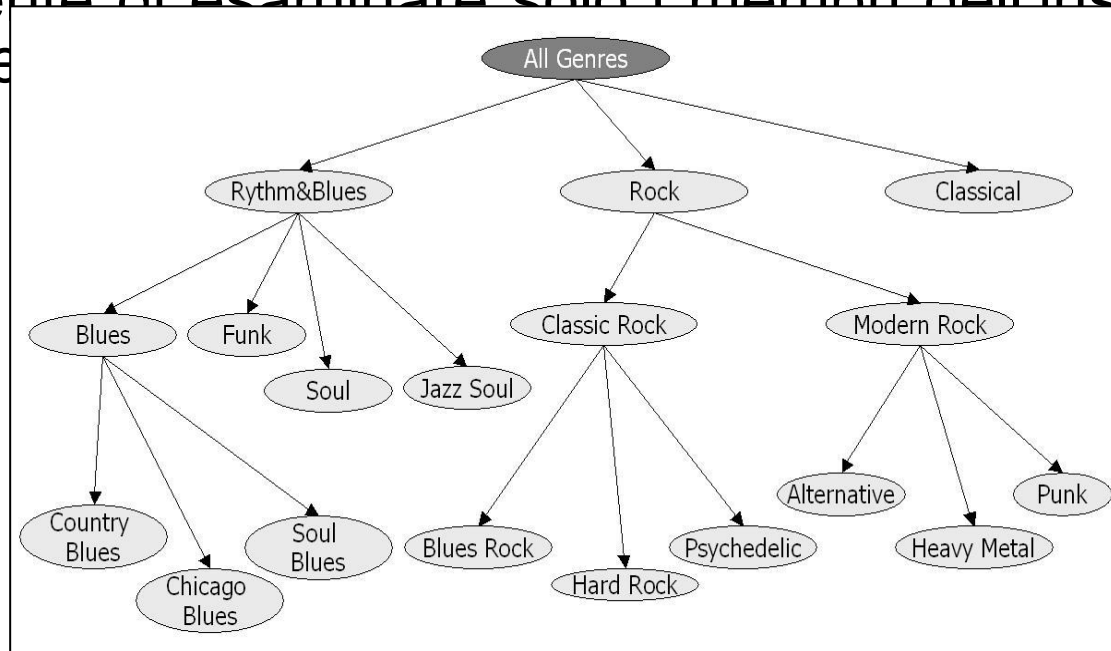


# Apprendimento senza supervisione

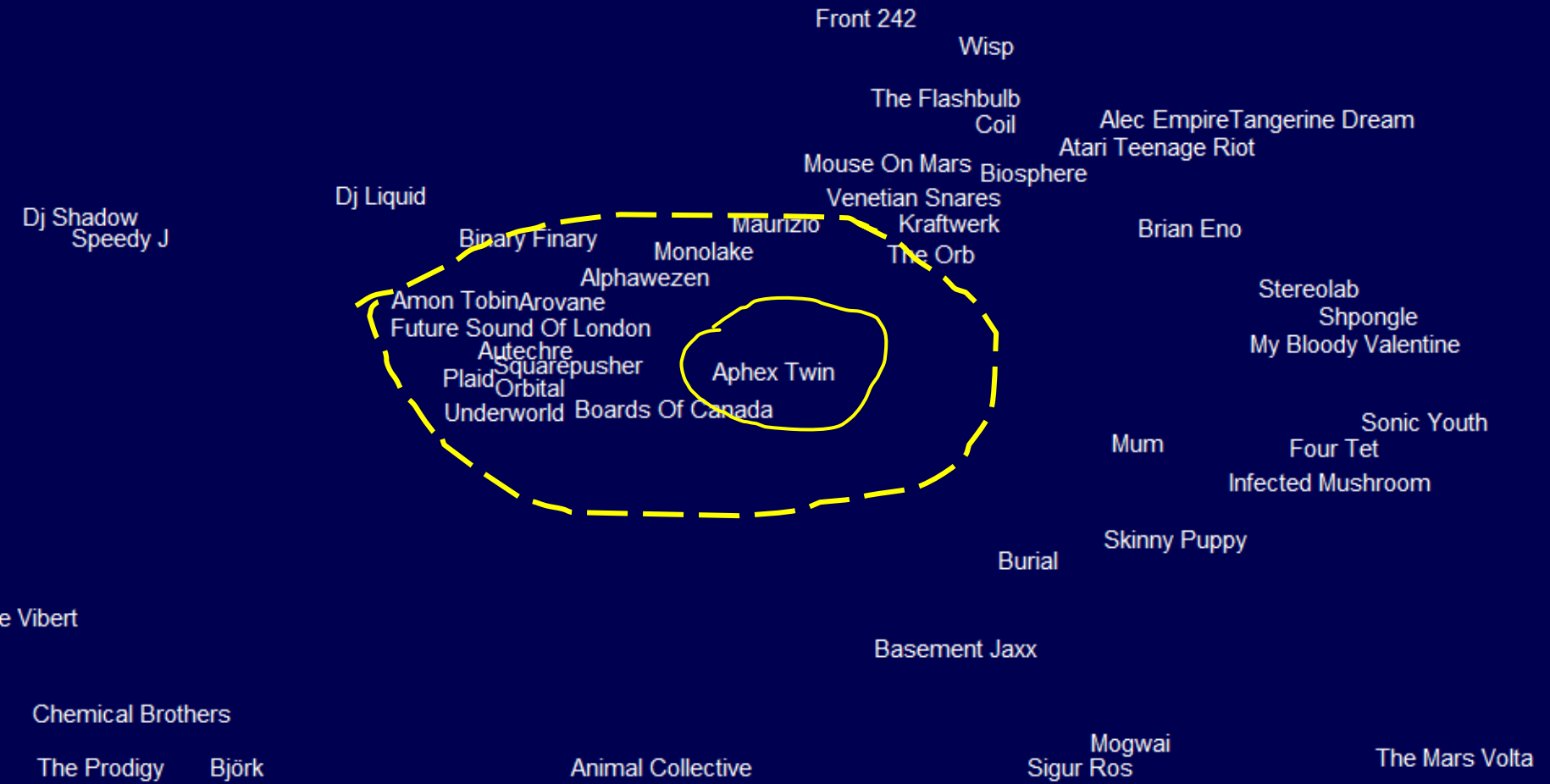
- In assenza di un oracolo o di conoscenze sul task esistono ancora molti modi di migliorare le proprie prestazioni, ad es.
  - Migliorando il proprio modello del mondo (acquisizione/*discovery* della conoscenza)
  - Migliorando le proprie prestazioni computazionali (ottimizzazione)

# Apprendimento senza supervisione

- Es Al termine del processo di acquisizione il sistema
- dispone di un sistema di classi e relazioni indotti che migliora la sua interazione futura con l'ambiente operativo (ad es. l'utente)
- Il miglioramento avviene quindi almeno rispetto agli algoritmi di ricerca: la organizzazione gerarchica consente di esaminare solo i membri dell'insieme in alcune



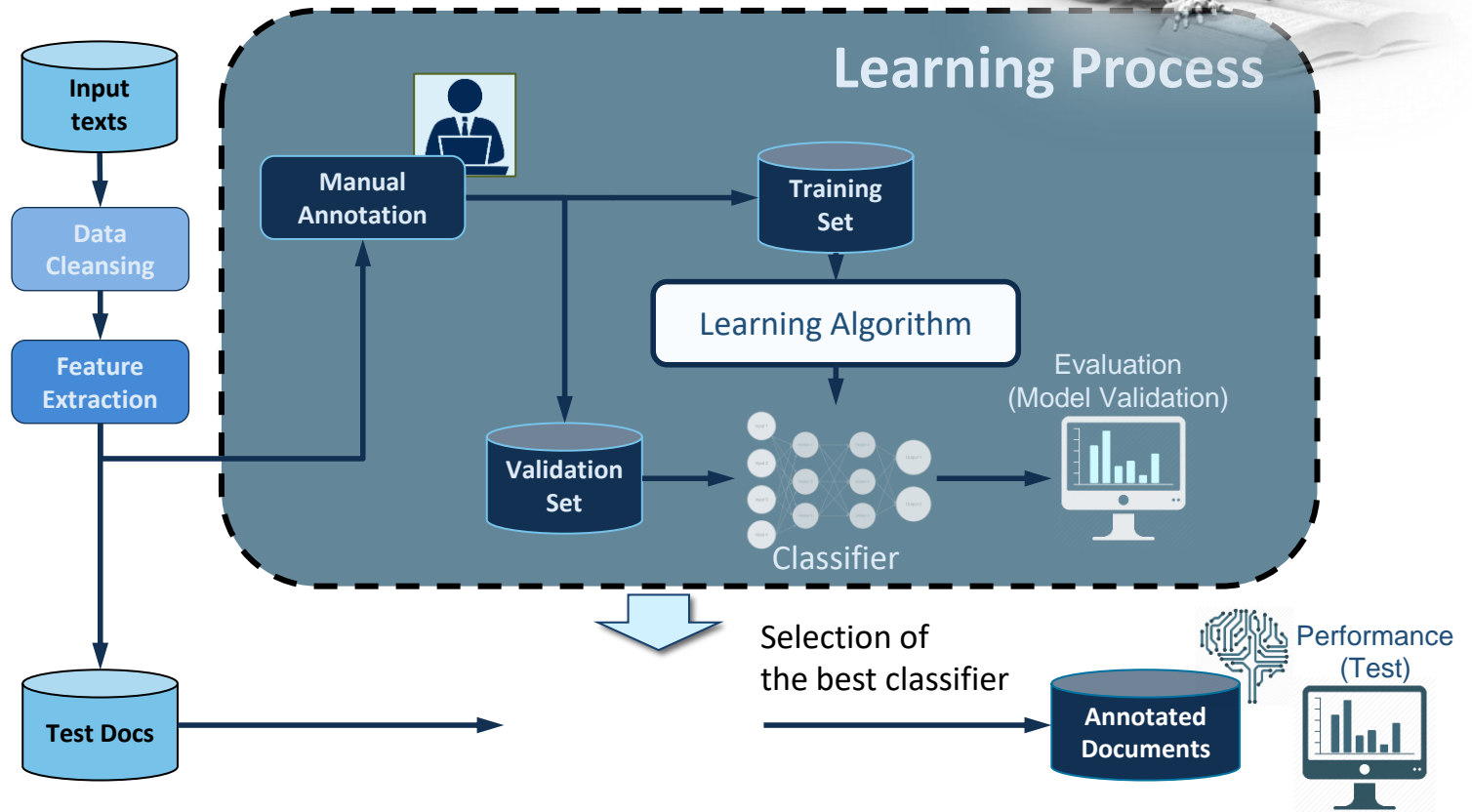
# map



# MACHINE LEARNING: PARAMETRIZZAZIONE

---

# Machine Learning workflow



# Evaluation of a ML system

- Performance Evaluation Metrics
  - Classifier Evaluation Metrics
- Tuning and Evaluation Methods

# Classifier Evaluation: Confusion Matrix

		PREDICTED VALUE		
		Class A	Class B	Class C
ACTUAL VALUE	Class A	38	12	0
	Class B	5	43	2
	Class C	6	0	44

$$accuracy = \frac{\#correct\ classifications}{\#classifications} = \frac{38 + 43 + 44}{150} = 83.33\%$$

$$error\ rate = \frac{\#incorrect\ classifications}{\#classifications} = \frac{12 + 5 + 2 + 6}{150} = 16.67\%$$



# Evaluation with skewed data

- Accuracy is not a suitable metric for task with imbalanced classes (for instance a spam detector)

		PREDICTED VALUE	
		Spam	Non-Spam
ACTUAL VALUE	Spam	0	10
	Non-Spam	0	9990

Very bad performance on the Spam class, that is the target of the classifier!! ... nonetheless ...

$$\text{accuracy} = \frac{\# \text{correct classifications}}{\# \text{classifications}} = \frac{9990}{10000} = 99.9\%$$

# Single Class Metrics

		PREDICTED VALUE	
		Class C	Not Class C
ACTUAL VALUE	Class C	<b>TP</b> True Positive	<b>FN</b> False Negative
	Not Class C	<b>FP</b> False Positive	<b>TN</b> True Negative

$$precision = \frac{TP}{TP + FP}$$

what percentage of instances the classifier labeled as positive are actually positive?

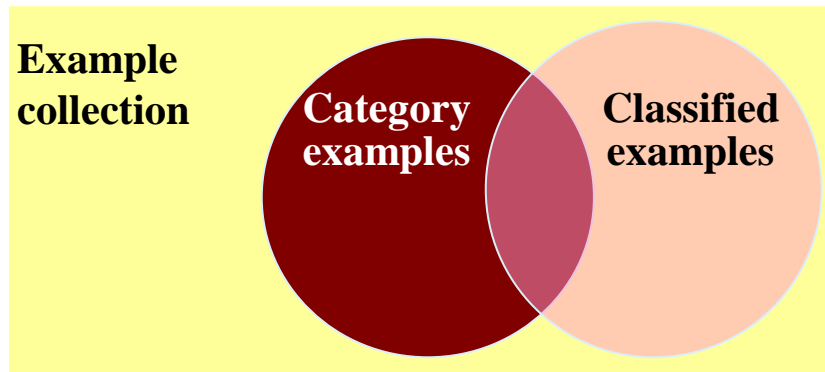
$$recall = \frac{TP}{TP + FN}$$

what percentage of positive instances did the classifier label as positive?

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

F-measure is the harmonic mean of precision and recall

# Class-based evaluation



Members	Classified	Classified & Members
	Rejected	Rejected but Members
Not Members	Classified	Classified but not Members
	Rejected	Rejected & not Members

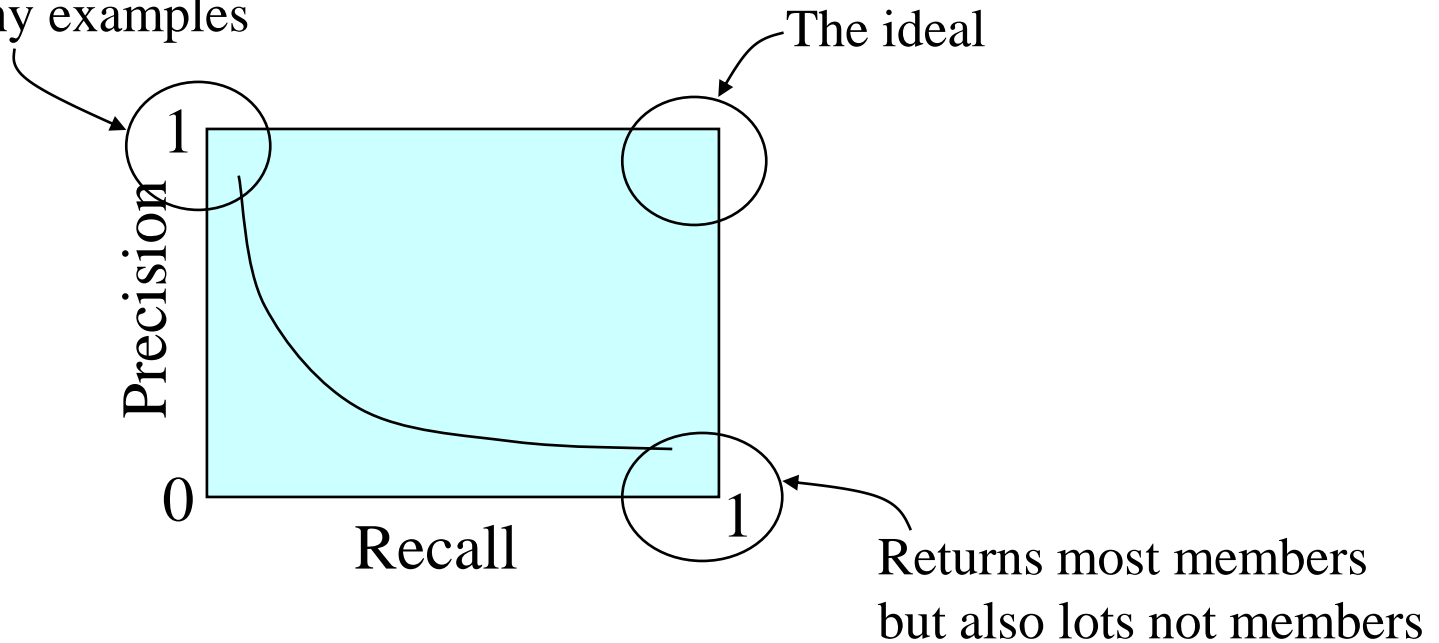
$$\textit{precision} = \frac{\# \textit{ of Members Classified}}{\# \textit{ of Members Classified} + \# \textit{ of Classified not Members}}$$

$$\textit{recall} = \frac{\# \textit{ of Members Classified}}{\# \textit{ of Members Classified} + \# \textit{ of Rejected Members}}$$

*What about accuracy???*

# Trade-off between Precision and Recall

Classify members but still misses many examples



# Other class based measures

## Precision and Recall of $C_i$

- a, corrects ( $TP_i$ )
- b, mistakes ( $FP_i$ )
- c, instances of a  $Class_i$  that are not actually retrieved, ( $FN_i$ )

The *Precision* and *Recall* are defined by the above counts:

$$Precision_i = \frac{a_i}{a_i + b_i}$$

$$Recall_i = \frac{a_i}{a_i + c_i}$$

		PREDICTED VALUE		
		Class A	Class B	Class C
ACTUAL VALUE	Class A	38	12	0
	Class B	5	43	2
	Class C	6	0	44

- $\text{Precision}_A = 38/(38+5+6)=38/49$
- $\text{Recall}_A = 38/(38+12)=38/50$
- $\text{Precision}_B = 43/(43+12)=43/55$
- $\text{Recall}_C = 44/(44+6)=44/50$

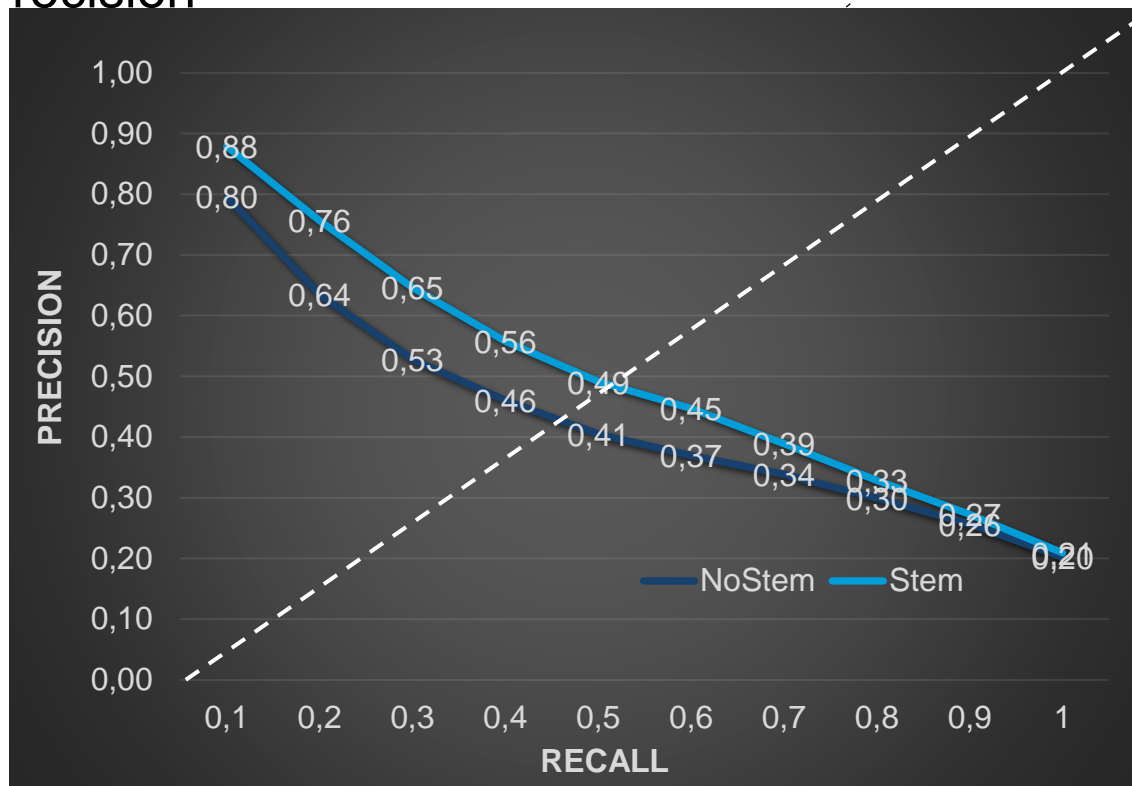
# Performance Measurements (cont'd)

- Breakeven Point
  - Find thresholds for which  
Recall = Precision
  - Interpolation
- F-measure 
$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$
  - Harmonic mean between precision and recall
- Global performance on more than two categories
  - Micro-average
    - The counts refer to classifiers
  - Macro-average (average measures over all categories)



# Break-even Point

- The BEP is the interpolated estimate of the value for which Recall=Precision

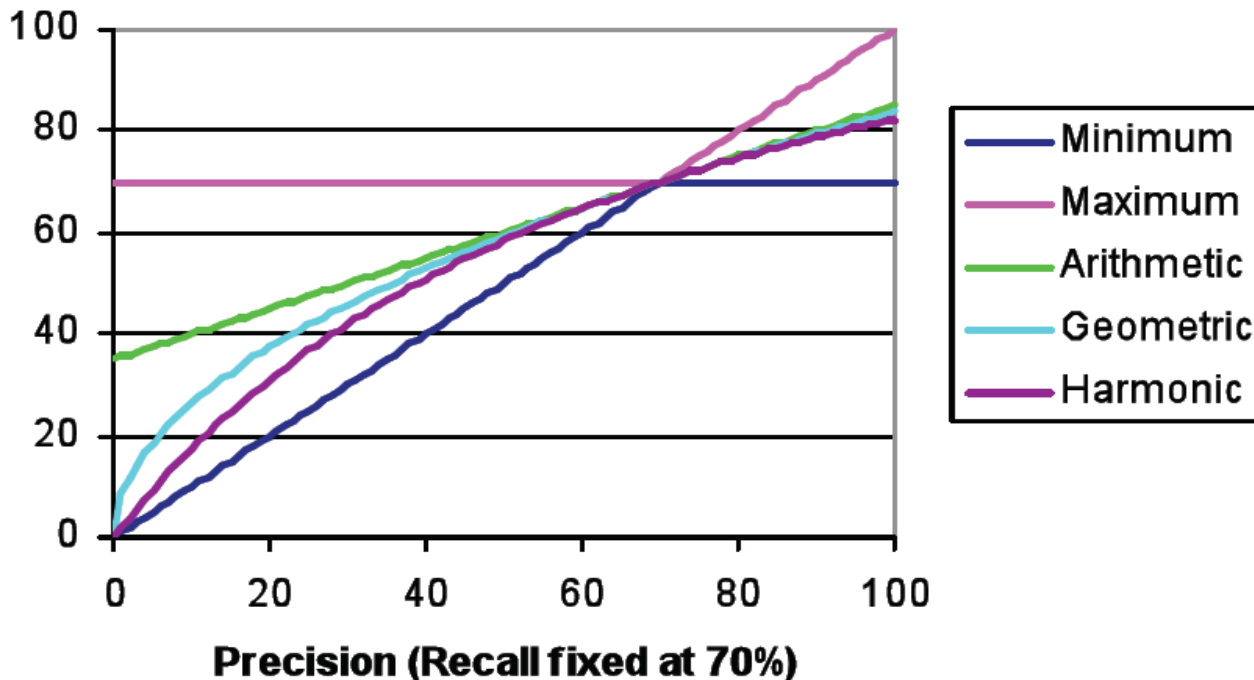


- It shows the superiority of methods whose behavior is closer to the (1,1) ideal performance

# Averaging Precision & Recall: comparison

A

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$



$$\min(p, r)$$
$$\max(p, r)$$

$$arithM(p, r) = \frac{p + r}{2}$$

$$geomM(p, r) = \sqrt{p \cdot r}$$

$$harmM(p, r) = \frac{2}{p^{-1} + r^{-1}}$$

# Averaging Precision & Recall: cross-categorical analysis

- Individual scores characterize the performance about each specific class
- Simple **macro** averaging can be applied to have

$$MPrecision = \sum_{i=1}^n Precision_i$$

$$MRecall = \sum_{i=1}^n Recall_i$$

$$MF_1 = \frac{2 \cdot MPrecision \cdot MRecall}{MPrecision + MRecall}$$

# F-measure e MicroAverages

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$\mu Precision = \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n a_i + b_i}$$

$$\mu Recall = \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n a_i + c_i}$$

$$\mu BEP = \frac{\mu Precision + \mu Recall}{2}$$

$$\mu f_1 = \frac{2 \times \mu Precision \times \mu Recall}{\mu Precision + \mu Recall}$$

		PREDICTED VALUE		
		Class A	Class B	Class C
ACTUAL VALUE	Class A	38	12	0
	Class B	5	43	2
	Class C	6	0	44

- $\text{Precision}_A = 38 / (38 + 5 + 6) = 38 / 49$
- $\text{Precision}_B = 43 / (43 + 12) = 43 / 55$
- Segue che:

$$M_{\text{precision}} = 1/3(38/49 + 43/55 + \dots)$$

		PREDICTED VALUE		
		Class A	Class B	Class C
ACTUAL VALUE	Class A	38	12	0
	Class B	5	43	2
	Class C	6	0	44

- $\text{Precision}_A = 38 / (38 + 5 + 6) = 38 / 49$
- $\text{Precision}_B = 43 / (43 + 12) = 43 / 55$
- Segue che:  
 $\mu\text{Precision} = (38 + 43 + 44) / (38 + 43 + 44 + 11 + 12 + 2)$

# Overview

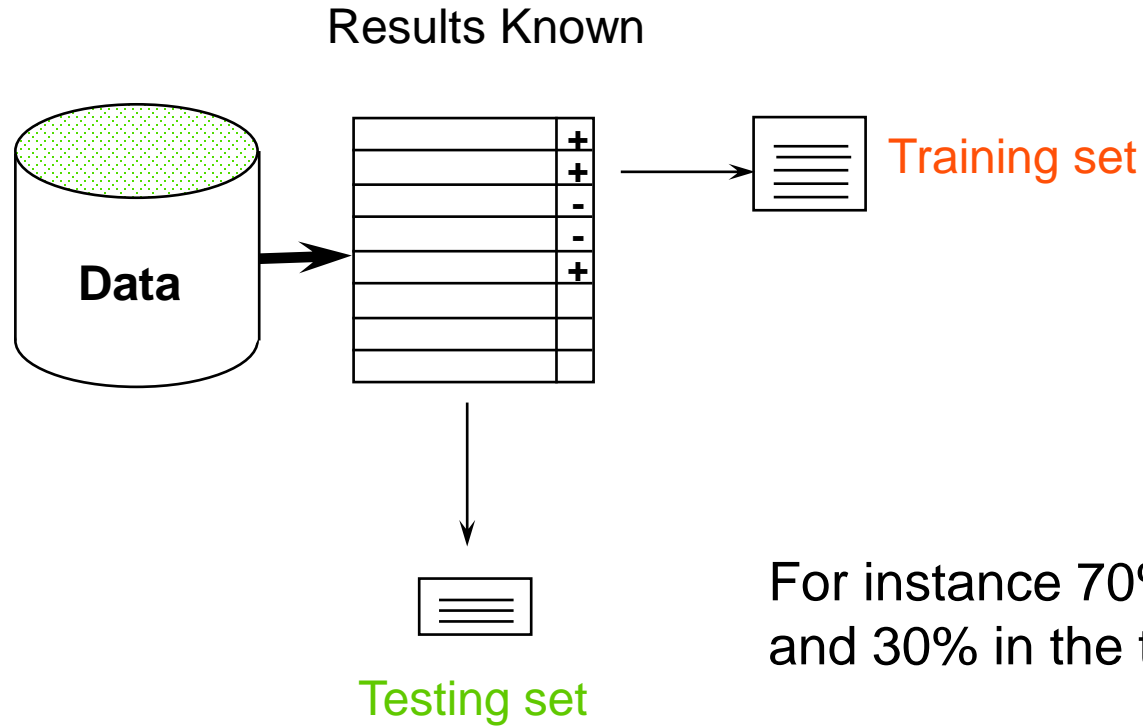
- Performance Evaluation Metrics
  - Classifier Evaluation Metrics
  - Information Retrieval Systems Evaluation Metrics
- **Tuning and Evaluation Methods**
- Error Diagnostics

# Testing Data

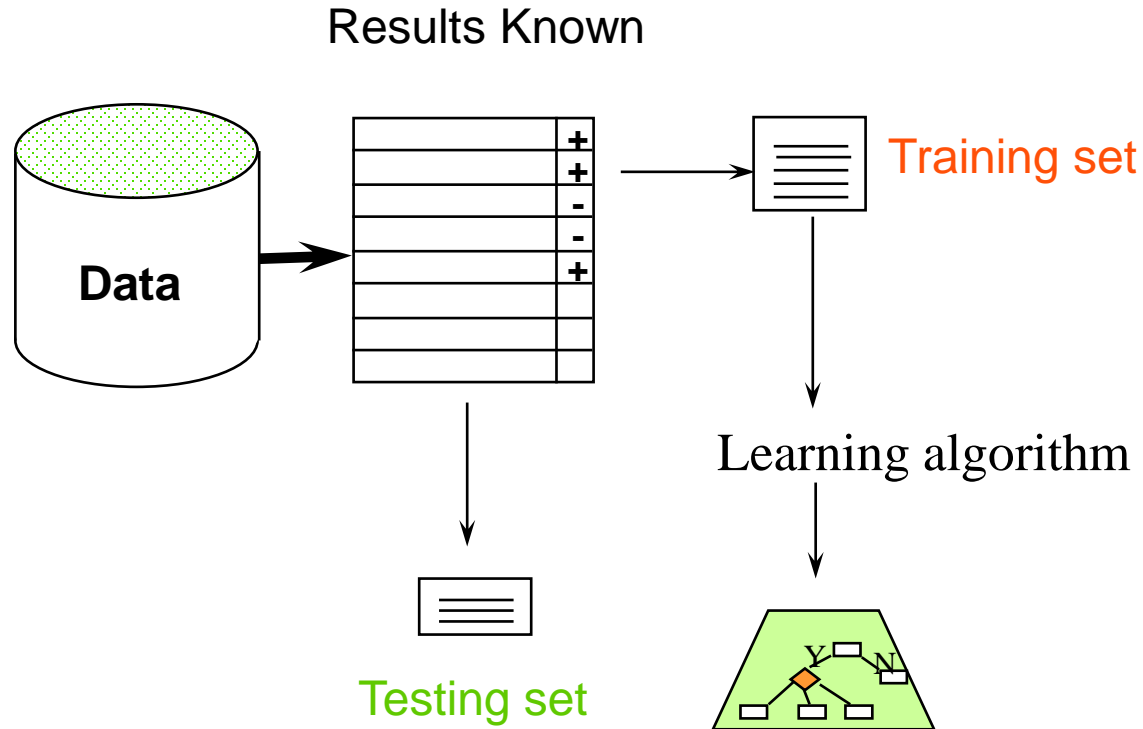
- To obtain a reliable estimation, test data **must be instances not employed for the training** step:
  - Error on the training data is *not* a good indicator of performance on future data, because new data will probably not be **exactly** the same as the training data!
  - **Overfitting** – fitting the training data too precisely - usually leads to poor results on new data
  - We want to evaluate how much accurate predictions of the model we learned are, and not other computational aspects (e.g. its memorization capability)



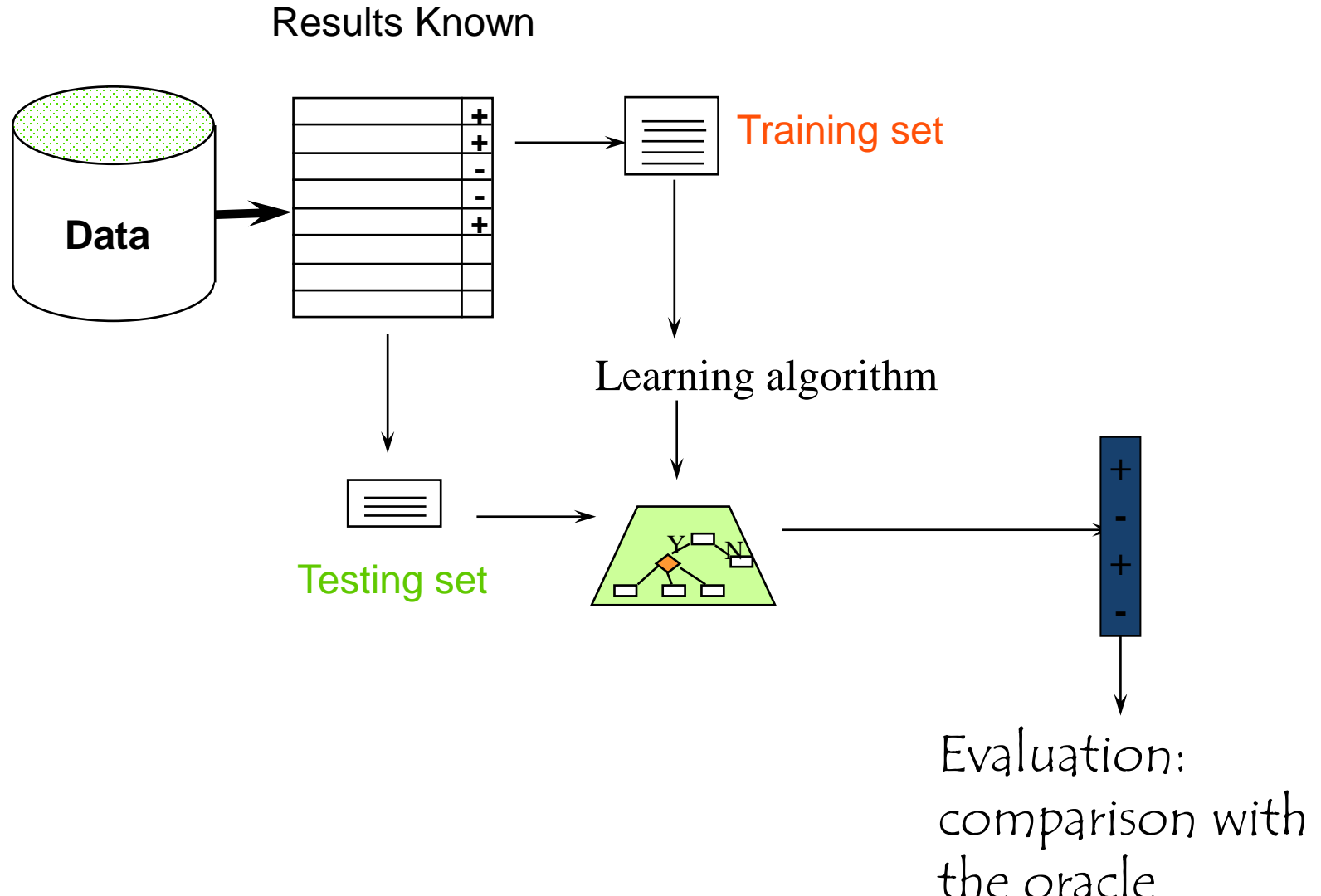
# Step 1: dataset splitting



# Step 2: learning phase



# Step 3: testing the model



# Evaluation on Few Data

- When data is scarce (totally or for a single class), a single evaluation process could not be enough representative
  - The testing set could contain too few instances to produce a reliable result
- **SAMPLING:** The evaluation process must be repeated with different splitting

# N-Fold Cross Validation

- Data is split into  $n$  subsets of equal size
- Each subset in turn is used for testing and the remainders  $n-1$  for training
- The metrics estimated in each round are averaged

5 fold  
splitting



Testing fold

Round 1



...

Round 5



Testing fold

# Tuning a Classifier

- Most of ML algorithms depends on some parameters
  - Examples:  $k$  in KNN,  $w_i$  in Rocchio,  $p(w_i | c_j)$  for NB
- The best configuration must be chosen after a proper tuning stage:
  - A set of configurations must be established (for instance,  $k=1, 2, 5, 10, \dots, 50$ )
  - Each configuration must be evaluated on a validation (or tuning) set

# INTRODUZIONE ALLE RETI NEURALI

---

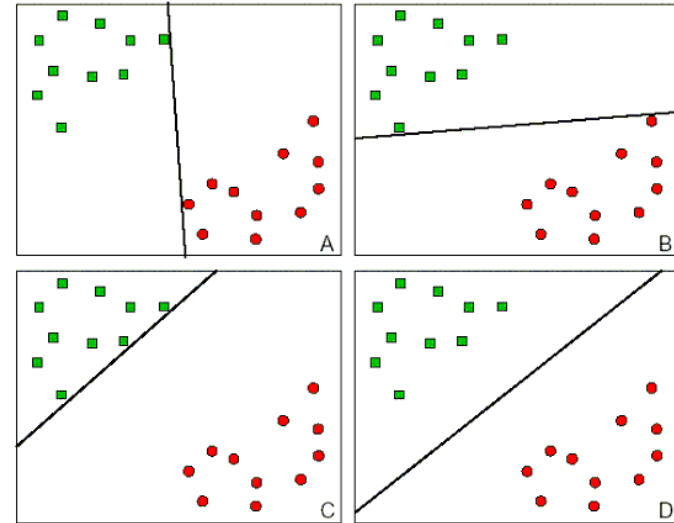
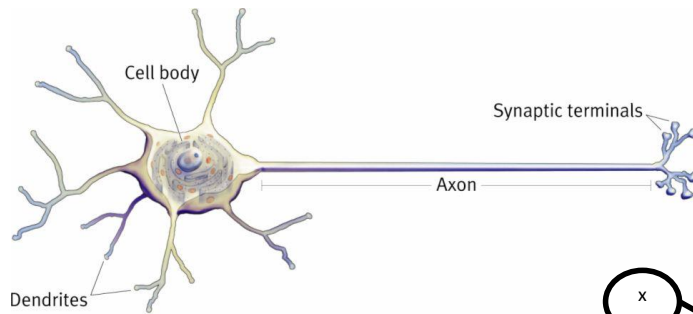
# Introduzione alle Reti neurali

- Dal perceptrone alle reti multistrato
- Addestramento in una rete multistrato
- Deep Learning:
  - Dalla Ingegneria delle Features (caratteristiche) ...
  - all'Apprendimento delle Rappresentazioni
- Applicazioni
  - Natural Language Processing
  - Image Classification e Object Recognition
- Language Modeling e sistemi di Chat

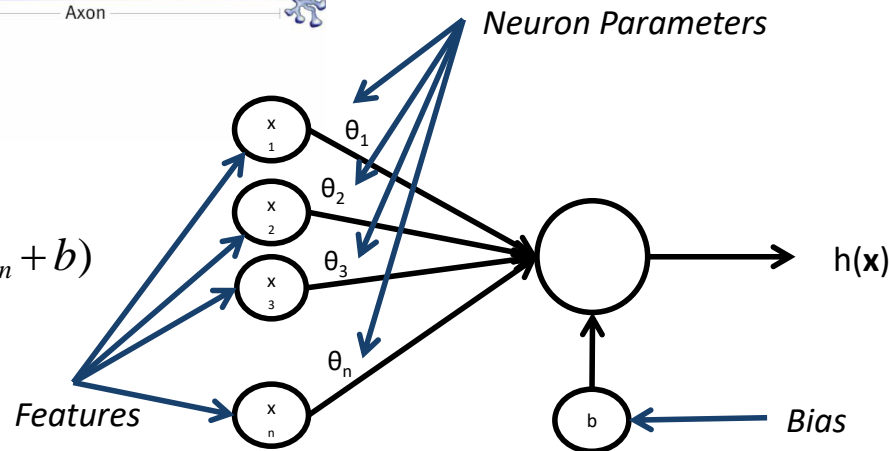


# Perceptron (Rosenblatt, 1958)

- Linear Classifier mimicking a neuron



$$h(\vec{x}) = g\left(\sum_n \theta_n x_n + b\right)$$



# Adding Layers ...

- From simple linear laws ...

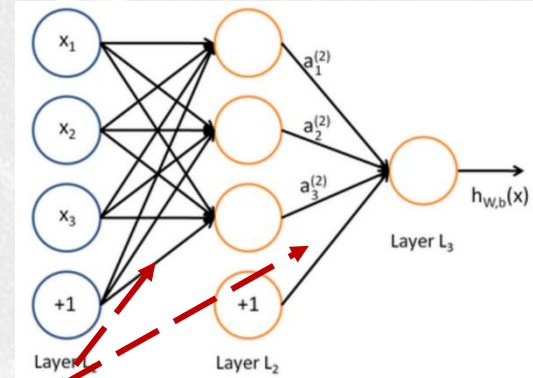
$$h(\vec{x}) = g(\vec{x}; \vec{\theta}, b) = g\left(\sum_n \theta_n x_n + b\right)$$

- to feedforward structures. It can be made dependent on a sequence of functions  $g^{(1)}$  and  $g^{(2)}, \dots, g^{(k)}$  that give rise to a structured hypothesis:

$$\begin{aligned} h(\vec{x}) &= g^{(2)}\left(g^{(1)}(\vec{x}; \vec{\theta}^{(1)}, b^{(1)}); \vec{\theta}^{(2)}, b^{(2)}\right) = \\ &= W^{(2)} g^{(2)}\left(g^{(1)}(W^{(1)} \cdot \vec{x} + b^{(1)}) + b^{(2)}\right) \end{aligned}$$

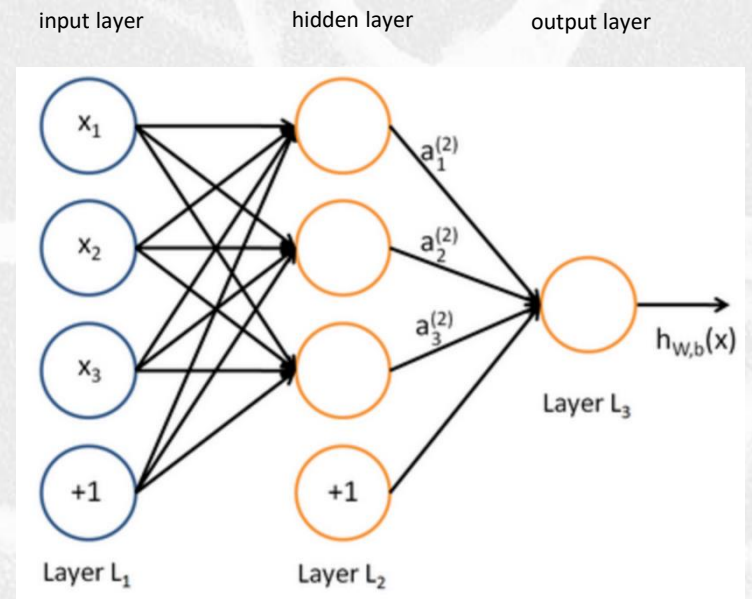
- Hidden layers

$$h^{(1)}(\vec{x}) = g^{(1)}(W^{(1)} \vec{x} + b^{(1)})$$



# Neural Networks

- Each circle represent a **neuron** (or unit)
  - 3 **input**, 3 **hidden** and 1 **output**
- $n_f=3$  is the **number of layers**
- $S_l$  denotes the **number of units in layer l**
- Layers:
  - Layer l is denoted as  $L_l$
  - Layer l and l+1 are connected by a matrix  $W^{(l)}$  of parameters
    - $W^{(l)}_{ij}$  connects neuron  $j$  in layer  $l$  with neuron  $i$  in layer  $l+1$
- $b^{(l)}_i$  is the **bias** associated to neuron  $i$  in layer  $l+1$

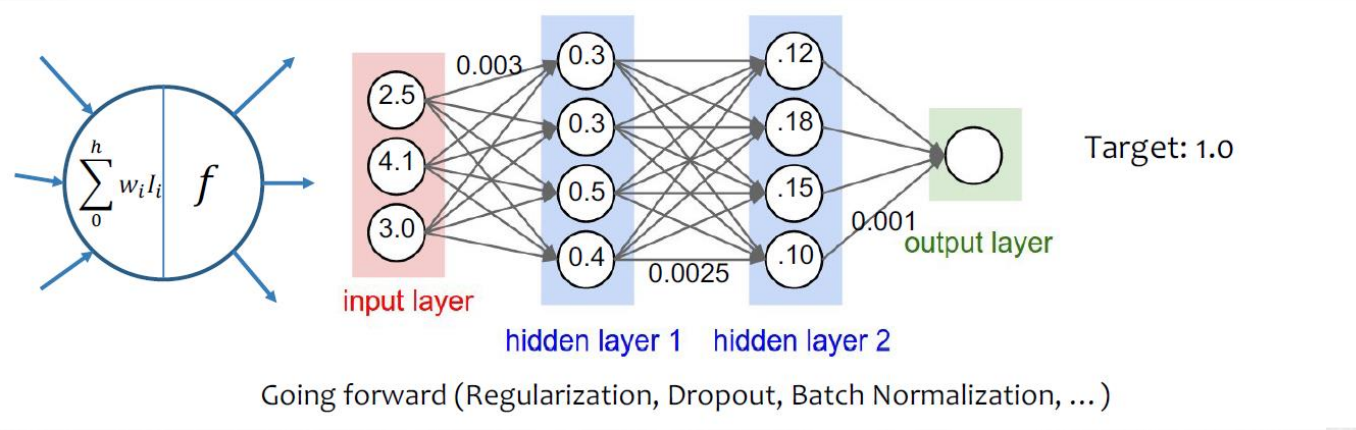
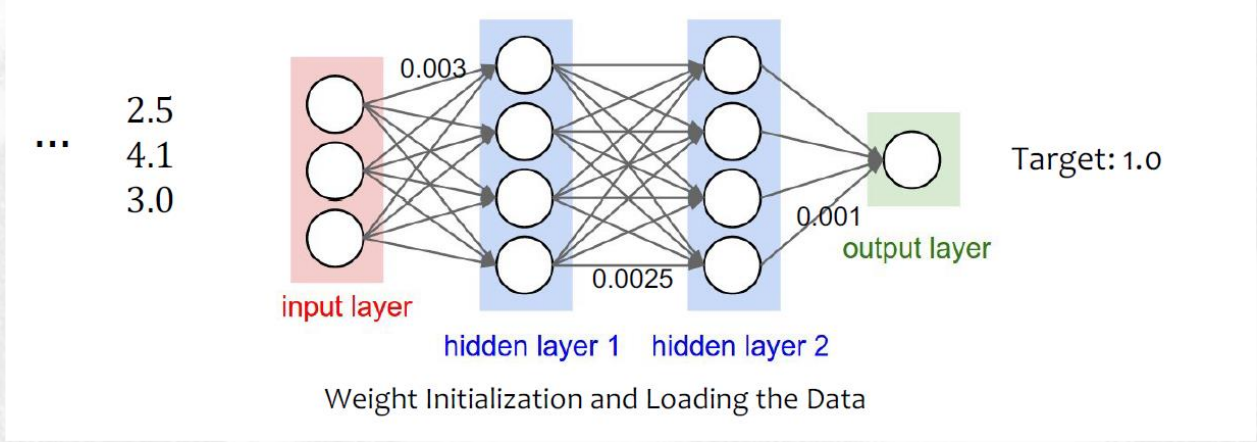


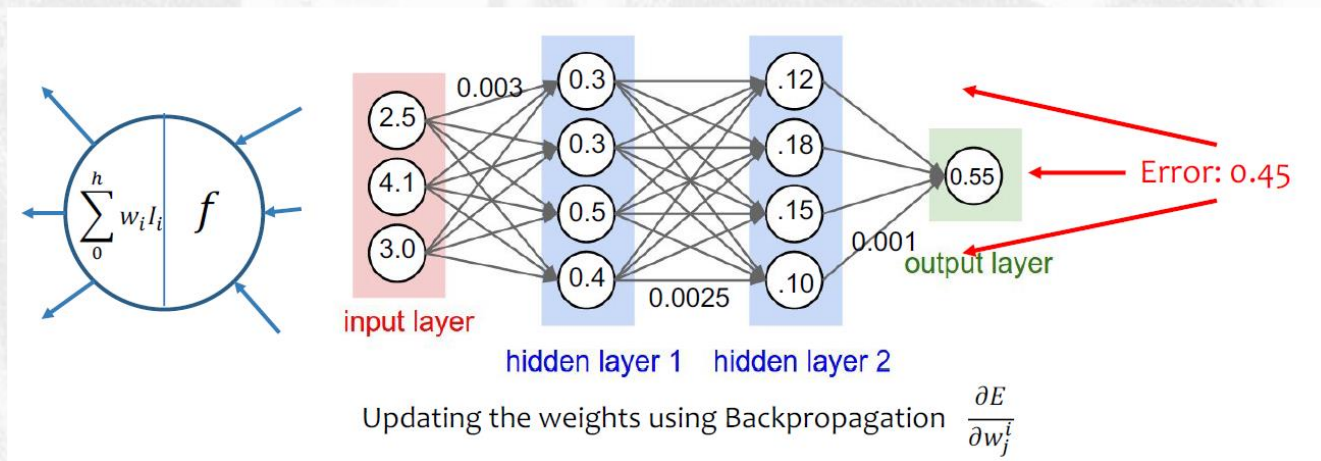
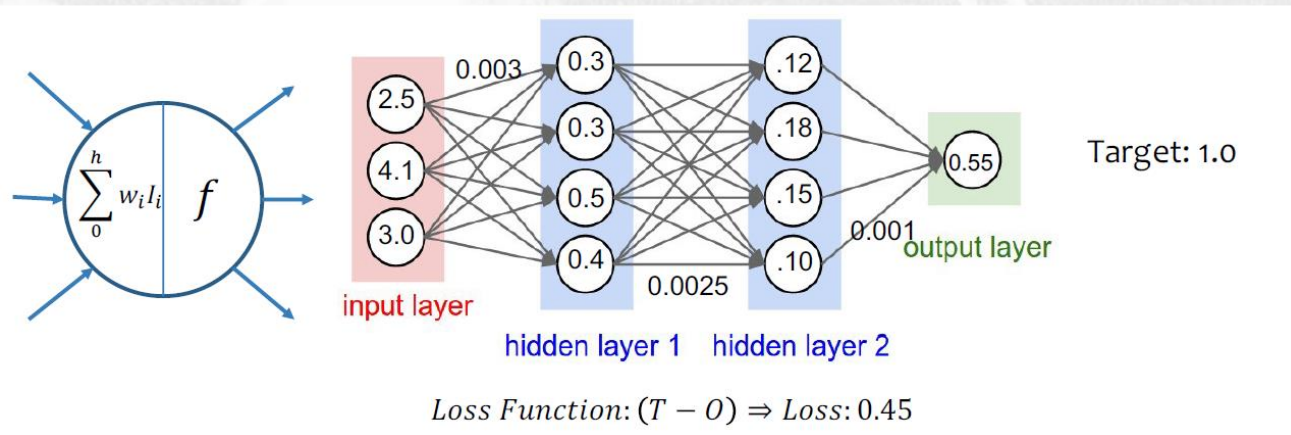
# Training MLPs: Back-propagation

- How are parameters of the tw-layer network, i.e.  $W$ ,  $w$  and  $c$ ,  $b$  defined?
- This is the role of the training algorithm for which:

$$f(\mathbf{x}; \mathbf{W}, \mathbf{c}, w, b) = w^\top \max\{0, \mathbf{W}^\top \mathbf{x} + \mathbf{c}\} + b \approx f^*(\mathbf{x}).$$

- The learning process in MLPs is based on two notions:
  - The optimization local to individual neurons
  - The adjustments to the overall network by propagation backwards from the output (where the error manifests) through all the hidden layers.





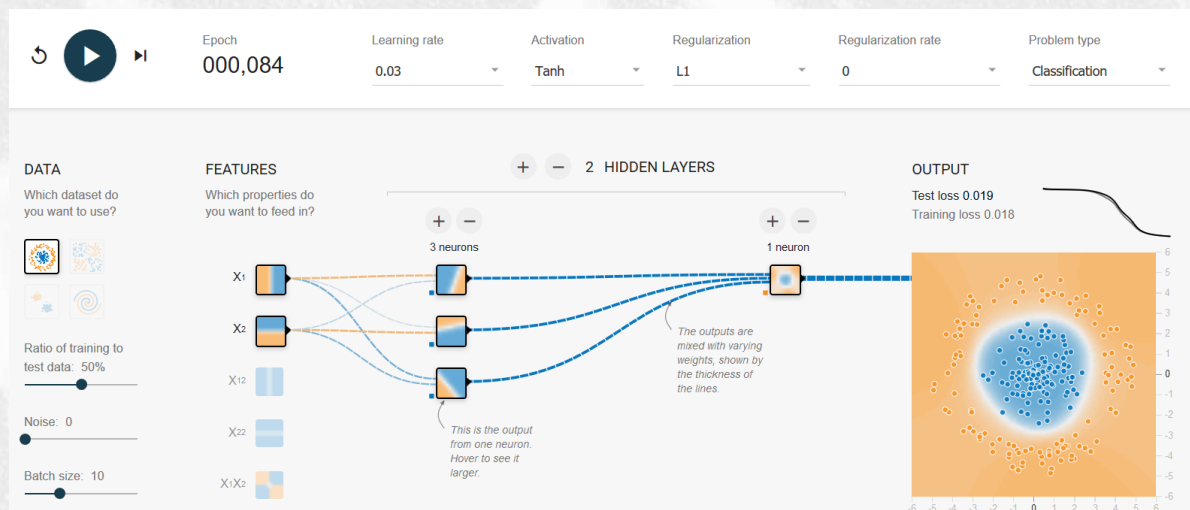
# How to induce the hypothesis $h$ from examples

- Learn the parameters  $\vartheta$  and  $b$
- To find these we look at the past data (i.e. training data) optimizing an objective function
- Objective function: the error we make on the training data
  - the sum of differences between the decision function  $h$  and the label  $y$
  - also called Loss Function or Cost Function

$$J(\theta, b) = \sum_{i=1}^m (h(x^{(i)}; \theta, b) - y^{(i)})^2$$

# A simple demo on TensorFlow

- Look at: <https://playground.tensorflow.org/>





# Backp

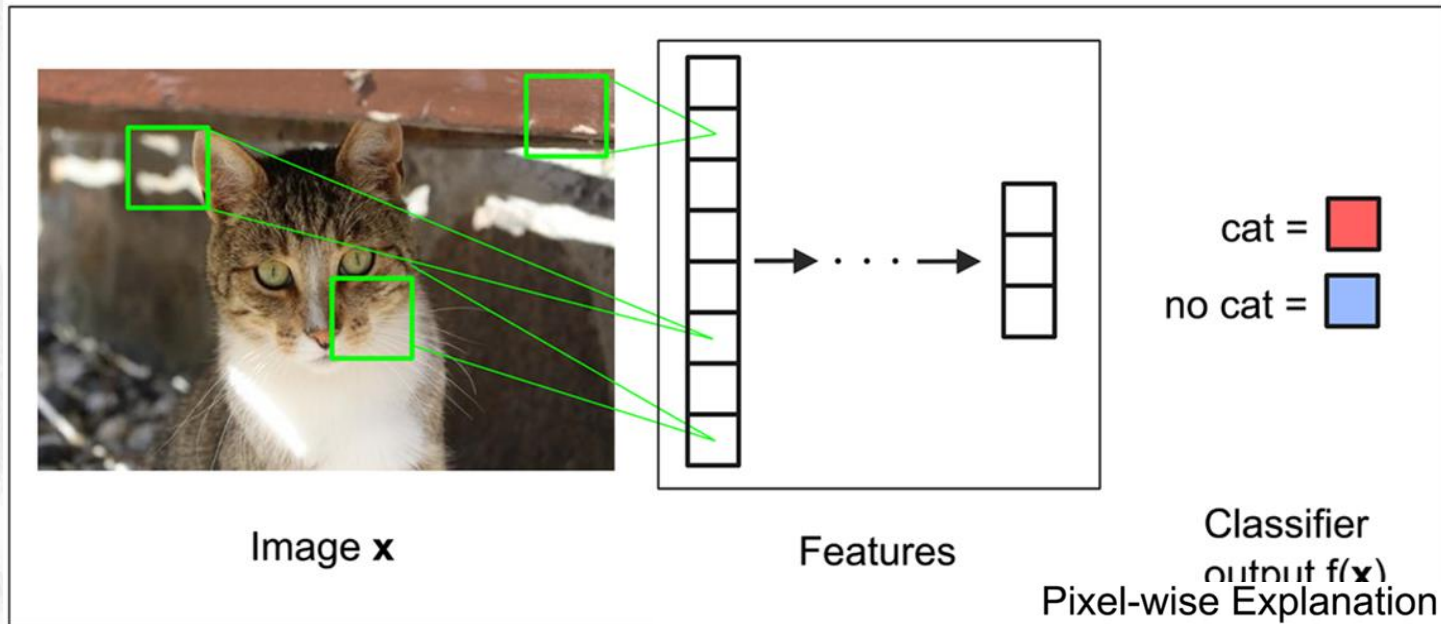
```
function BACK-PROP-LEARNING(examples, network) returns a neural network
  inputs: examples, a set of examples, each with input vector  $\mathbf{x}$  and output vector  $\mathbf{y}$ 
           network, a multilayer network with  $L$  layers, weights  $w_{i,j}$ , activation function  $g$ 
  local variables:  $\Delta$ , a vector of errors, indexed by network node

  repeat
    for each weight  $w_{i,j}$  in network do
       $w_{i,j} \leftarrow$  a small random number
    for each example  $(\mathbf{x}, \mathbf{y})$  in examples do
      /* Propagate the inputs forward to compute the outputs */
      for each node  $i$  in the input layer do
         $a_i \leftarrow x_i$ 
      for  $\ell = 2$  to  $L$  do
        for each node  $j$  in layer  $\ell$  do
           $in_j \leftarrow \sum_i w_{i,j} a_i$ 
           $a_j \leftarrow g(in_j)$ 
      /* Propagate deltas backward from output layer to input layer */
      for each node  $j$  in the output layer do
         $\Delta[j] \leftarrow g'(in_j) \times (y_j - a_j)$ 
      for  $\ell = L - 1$  to  $1$  do
        for each node  $i$  in layer  $\ell$  do
           $\Delta[i] \leftarrow g'(in_i) \sum_j w_{i,j} \Delta[j]$ 
      /* Update every weight in network using deltas */
      for each weight  $w_{i,j}$  in network do
         $w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$ 
  until some stopping criterion is satisfied
  return network
```

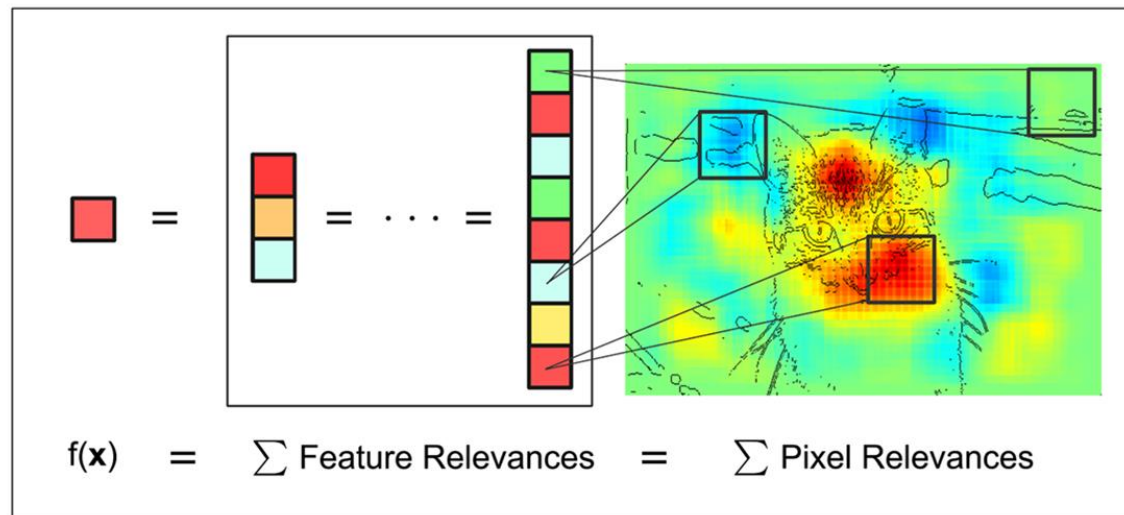
**Figure 18.24** The back-propagation algorithm for learning in multilayer networks.

# NN for Image Classification

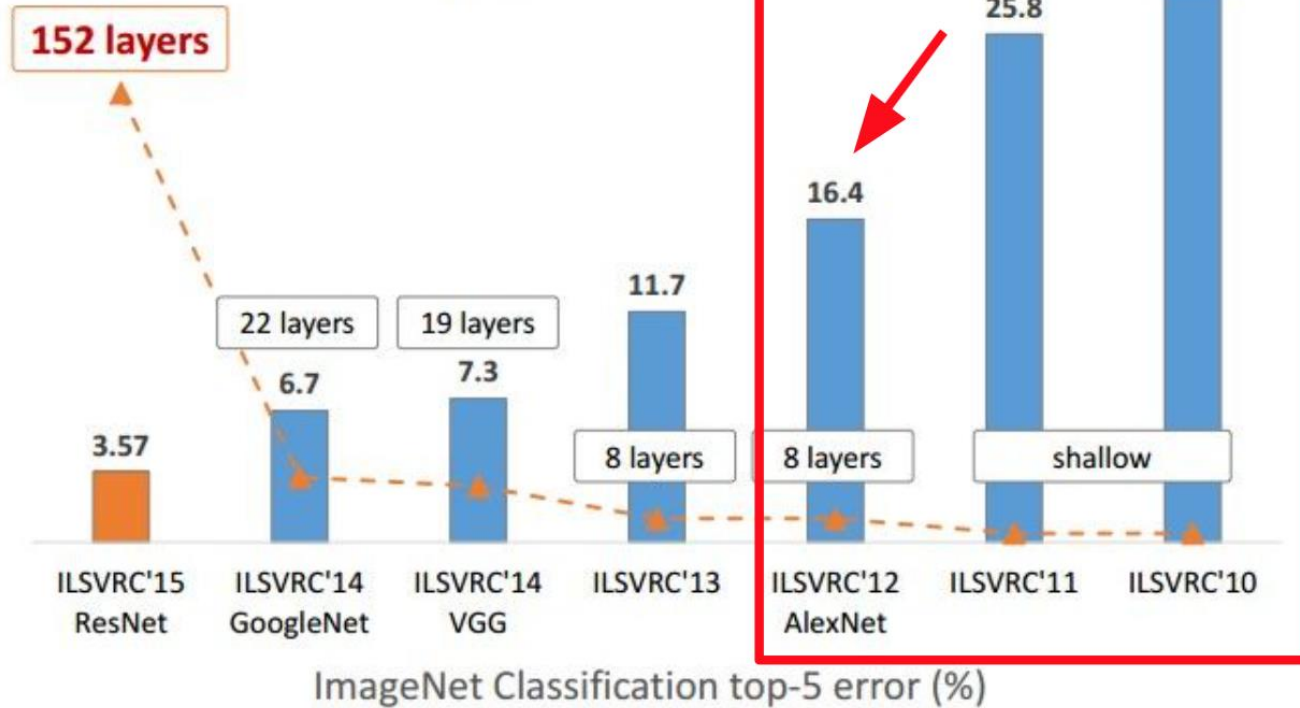
## Classification



Classifier output  $f(x)$   
Pixel-wise Explanation

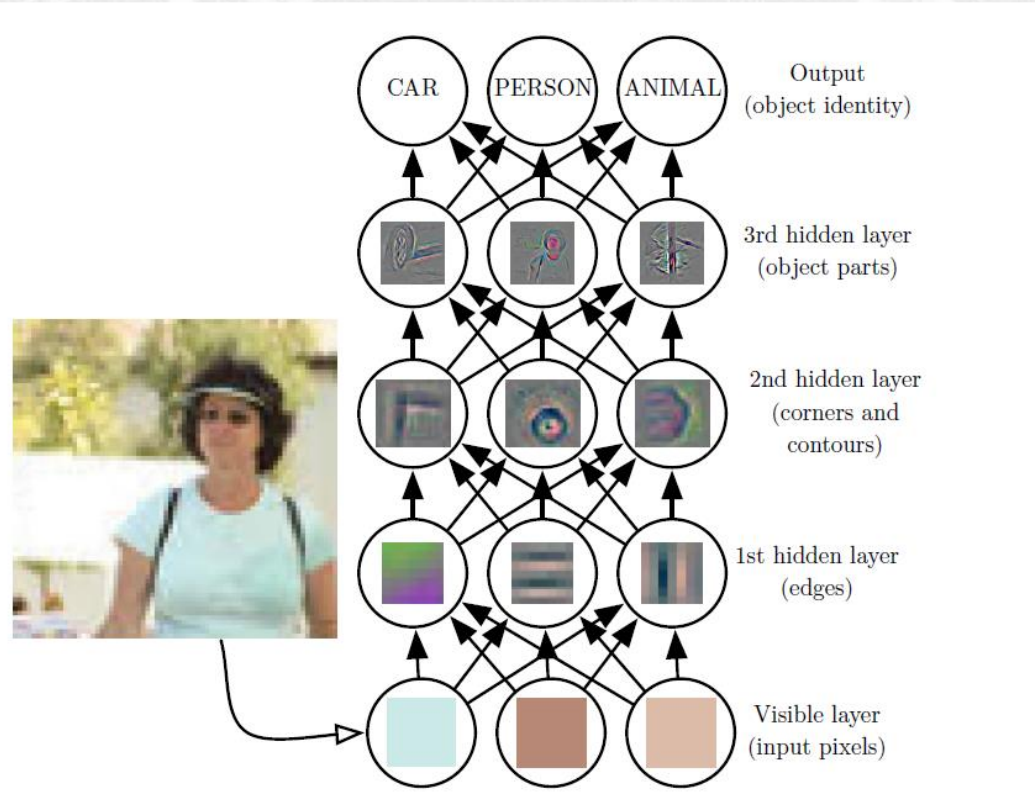


# Revolution of Depth



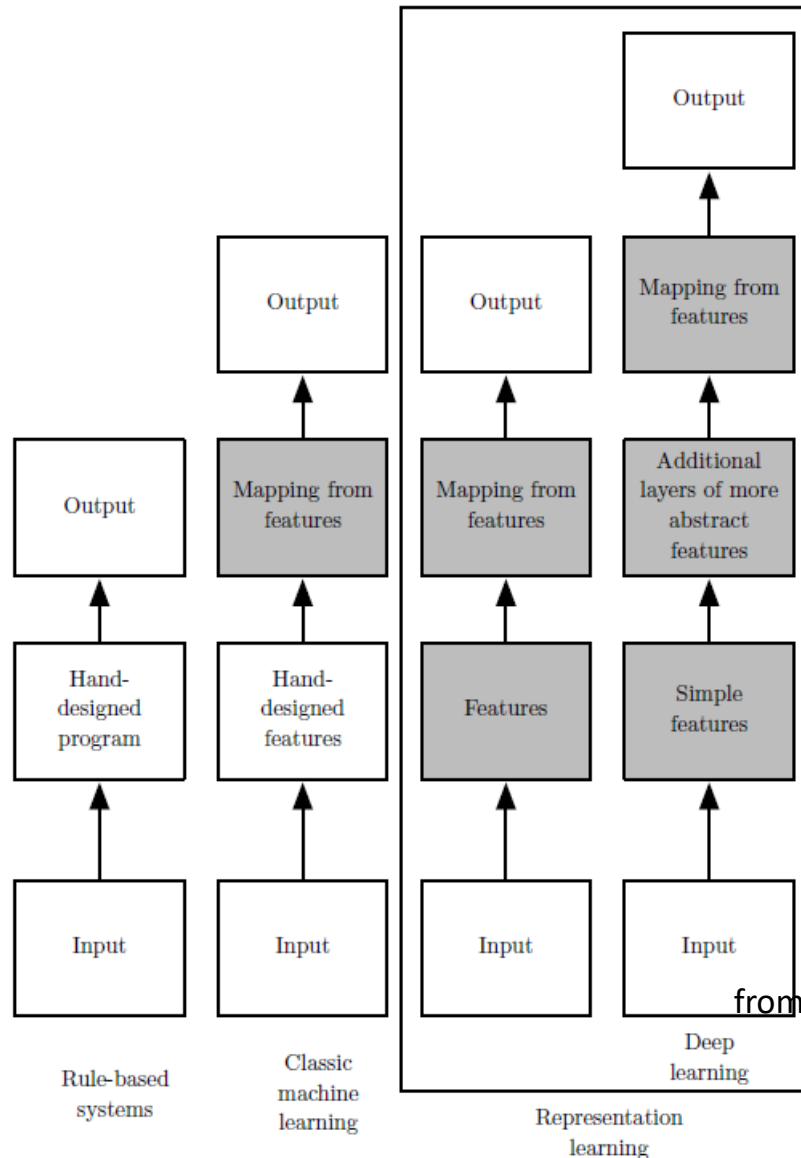
(slide from Kaiming He's recent presentation)

# Rappresentazione e Apprendimento: il ruolo della profondità



Zeiler and Fergus (2014)

# Learning Multiple Components

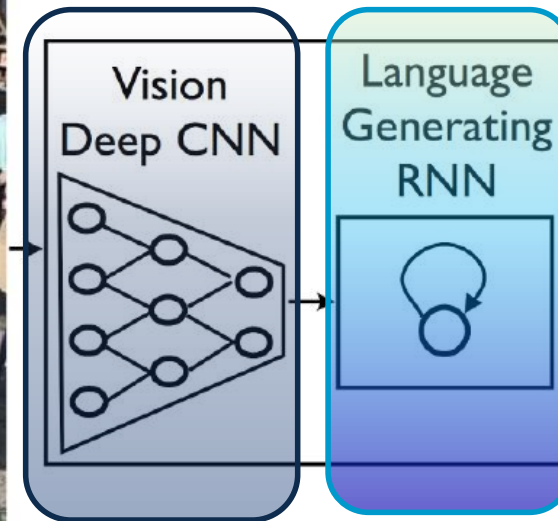


from Goodfellow et al., DL MIT book

# Applicazioni delle reti neurali

- **Acquisizione Automatica di Lessici Semantici da testi (non annotati)**
  - Modelli linguistici del linguaggio scritto (Italiano, Inglese, ...)
  - Modelli vettoriali della semantica lessicale
  - Sistemi di pre-addestramento per la inizializzazione dei classificatori supervised (ad es. Machine Translation systems)
- **Riconoscimento di oggetti o di pattern (ad es. emozioni dai visi) nelle immagini**
- **Acquisizione integrata di conoscenza da immagini, video e testi scritti**
- **Sistemi di Question Answering o Dialogo su immagini**

# Automatic image captioning: neural networks *at work*



**A group of people  
shopping at an  
outdoor market.**

**There are many  
vegetables at the  
fruit stand.**

# Relationship with other areas of AI

- Neural networks for **faster inference over large knowledge bases** wrt to logical approaches
- Neural networks for **high quality and cost-effective complex tasks**
  - Pattern recognition
  - Language and Image/Video processing
  - Complex rewriting tasks, e.g. Machine Translation
- **Learning to acquire knowledge**
  - Machine Reading for QA
  - Fast Indexing and retrieval from large document bases or Web sources
- Processing **Time Series**
  - Predictive Analytics tasks that depend on time
  - Sequence labeling tasks
- Complex inference on **hybrid** (i.e structured and unstructured) **data**



# Riferimenti Bibliografici

- *AIMA*, Chapter 18
- **READING.** *Machine Learning*, Tom Mitchell, Mc Graw-Hill International Editions, 1997 (Cap 3).
- **L'Algoritmo Definitivo**, Pedro Domingos, Bollato Boringhieri, 2016

