A Short Introduction to Machine Learning

Introduction to Machine Learning
Lect.s 2 and 3

Alessio Micheli

micheli@di.unipi.it

Dipartimento di Informatica
Università di Pisa - Italy

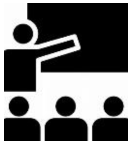
Computational Intelligence & Machine Learning Group

About ML

Dip. Informatica
University of Pisa

- **Machine Learning (ML)**
  - *Master Programme in Computer Science*
  - *Master Programme in Data Science and Business Informatics*
  - *Master Programme in Digital Humanities*
- **Code: 654AA Credits (ECTS): 9 Semester: 1**
- **Lecturer: Alessio Micheli:** micheli@di.unipi.it

Practical information



In class:

- Please, max **silence** during the lecture (to avoid noise) *recording in progress!* (of course, you can make questions)

Connect to ML:

- **Material: Moodle https://elearning.di.unipi.it/**
- **Streaming & recordings of lectures: Teams platform**
  - See lecture 1 and Moodle: «FAQ and general information»
  - The enrolling students mechanism for attendance "in presence" (and **to connect to Teams**) is through the App "Didactic Agenda" ("Agenda Didattica") for 654AA 24/25
  - Please, remember to fill the poll (see INTRO-curricula22)

Introduction to ML: plan of the next lectures

Dip. Informatica
University of Pisa

- Introduction aims:
  - Critical contextualization of the ML in comp. science [lect 1 and 2]
  - **Overview and Terminologies** [lect 2, 3, 4]
    - the relevant concepts will be developed later in the course
  - First basic models and learning algorithms [lect 5, 6, 7]
  - Then, we will start with Neural Networks!

See the "Course structure" slide!

Learning

*The problem of **learning** is arguably at the very core of the problem of **intelligence**, both biological and artificial*

Poggio, Shelton, *AI Magazine 1999*

i.e. Learning as a major challenge and a strategic way to provide intelligence into the systems

Machine Learning (I)

Dip. Informatica
University of Pisa

We restrict to the *computational* framework:

- Principles, methods and algorithms for learning and prediction:
  - Learning by the system of the experience (known data) to approach a defined computational task
  - Build a model (or hypothesis) to be used for predictions
    - (see examples on email-spam or face recognition)

Most common specific framework :

- Infer a model / *function* from a set of examples which allows the ***generalization*** (to provide accurate response on new data)

Machine Learning (II): When?

Dip. Informatica
University of Pisa

Opportunity (if useful) and *awareness* (needs and limits)

- Utility of predictive models: (in the following cases)
  - **no (or poor) theory** (or knowledge to explain the phenomenon)
  - **uncertain, noisy or incomplete data** (which hinder formalization of solutions)
- Requests:
  - source of training experience (representative data)
  - tolerance on the precision of results

Machine Learning (III): When?

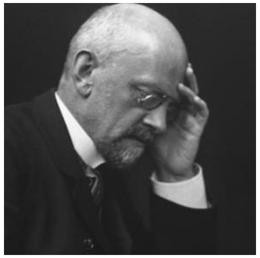
Dip. Informatica
University of Pisa

- Models to solve real-world problems that are difficult to be treated with traditional techniques (*complementary* to analytical models based on previous knowledge, algorithms and imperative programming, classical AI, ...)
- Examples of appropriate applications versus standard programming:
  - Knowledge is too difficult (to be formalized by 'hand-made' algorithm)
    - e.g. face recognition: humans can do it but cannot describe how they do it
    - e.g. voice automatic telephone answering service
  - Not enough human knowledge
    - e.g., predicting binding strength of molecules to proteins
  - Personalized behavior
    - scoring email messages or web pages according to user preferences
    - individualized (intelligent) human-computer interfaces
- Due to this flexibility ML applicative area is very large: see lecture 1

General challenges

- Build autonomous Intelligent/learning systems:
  - Robotics, HRI, search engines, …
- Build powerful tools for emerging challenges in intelligent data analysis
  - Tools for the "data scientist"
- Open new areas of applications in CS: innovative interdisciplinary open problems (more in general, "machine learning scientist")
  - Fantasy is your limitation !
  - ML in the era of "changing of paradigm in science, in which scientific advances are becoming more and more *data-driven*"
  - *Growing data sources opens up a huge application area for ML and related areas (Web, Social Net., IoT, BioMed, …)*

An useful framework:
Learning as an approximation of an unknown function from examples



Hilbert spaces

Specific vision but widespread in ML

For us:

- **Different tasks seen in uniform framework**
- **Enables a rigorous formulation**

→ Intro guided by intuitive examples

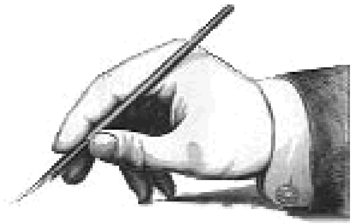
Please, note that the following example was already introduced in Lect 1

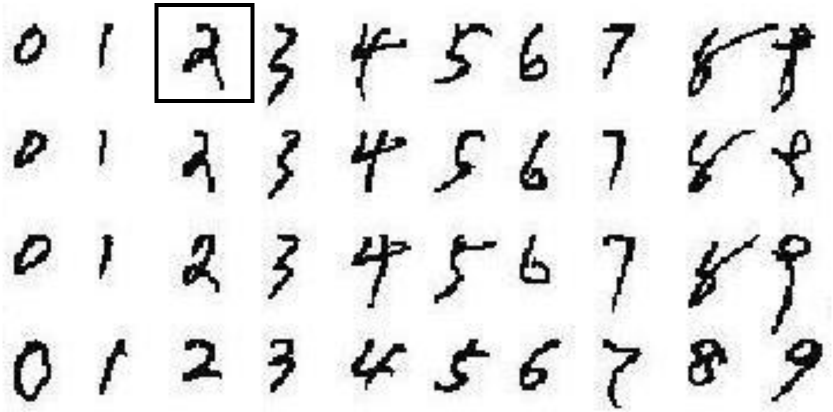
An Example

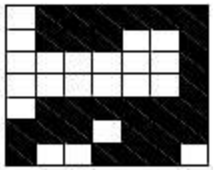
Dip. Informatica
University of Pisa

- A pilot example: *recognition of handwritten digits*

- Input: collection of images of handwritten digits (arrays/matrix of values)
- Problem: build model that receives in input an image of handwritten digit and "predict" the digits



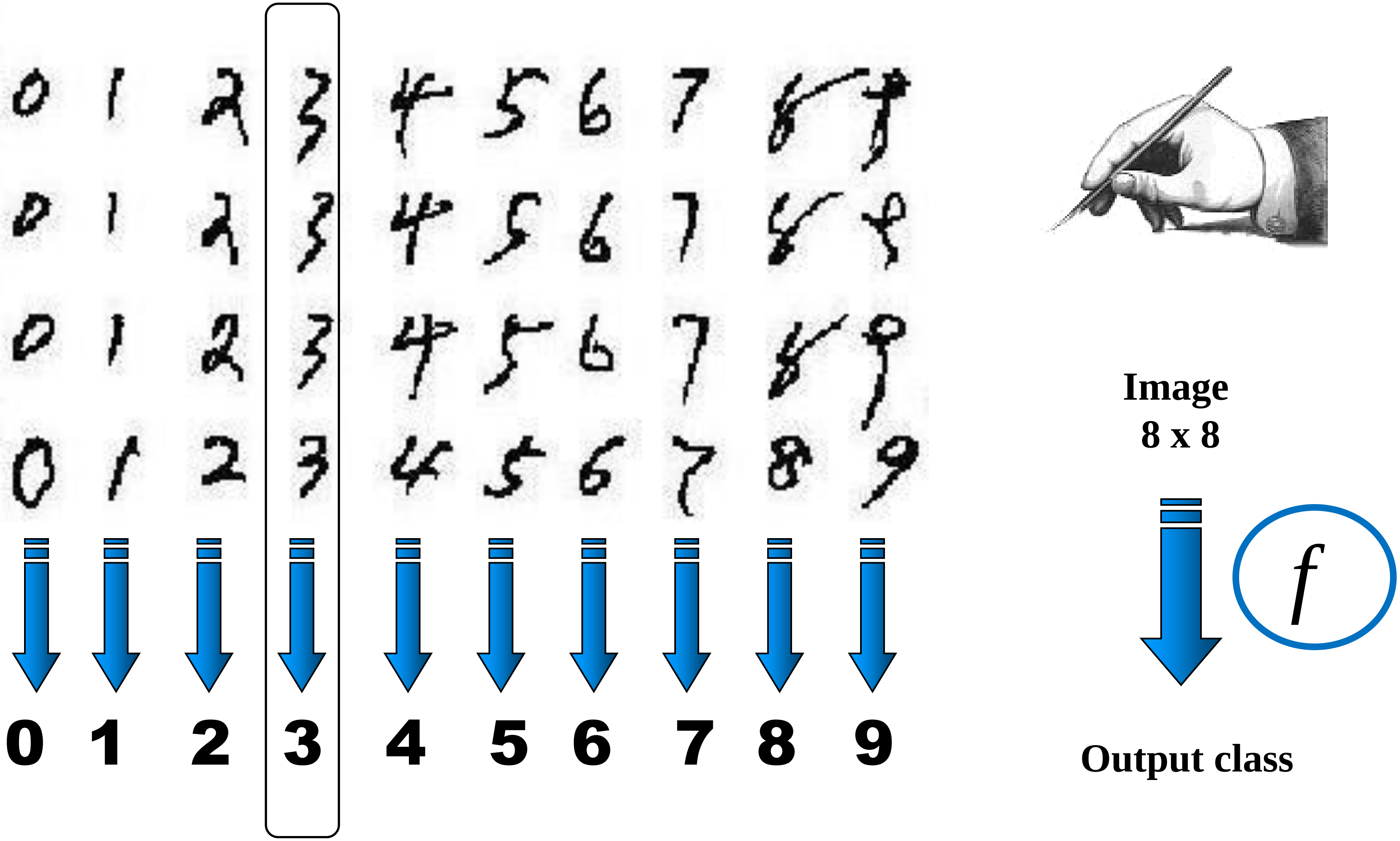




8 x 8

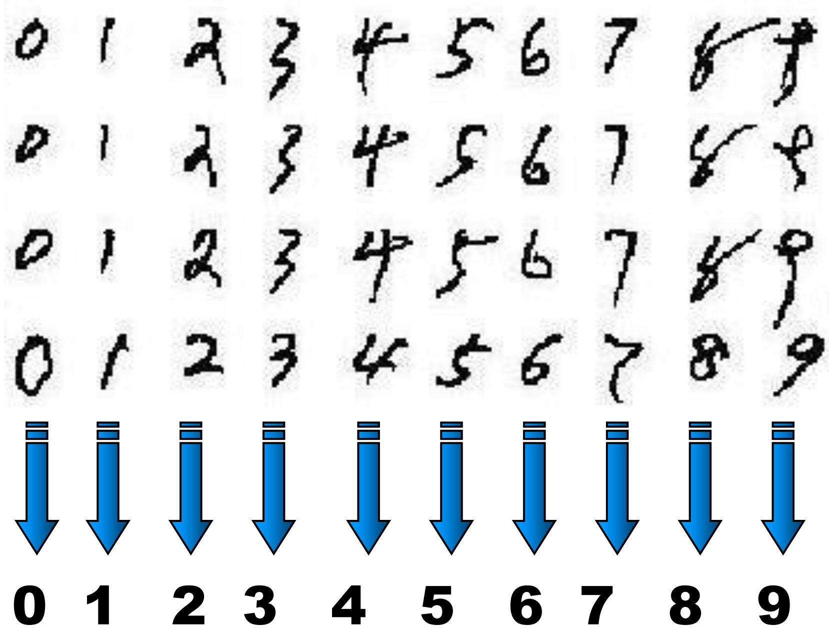
Build a function from examples

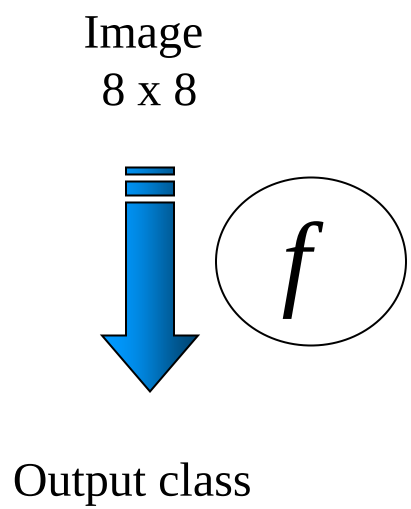
Dip. Informatica
University of Pisa



Handwritten Digits Recognition

Dip. Informatica
University of Pisa





Classification problem

- Difficult to formalize exactly the solution of the problem: Possible presence of noise and ambiguous data;
- Relatively easy to collect a set of labeled examples

=> Example of successful application of the ML!

Machine Learning

Dip. Informatica
University of Pisa

A new extended definition (looking to the pilot example)

- The ML studies and proposes methods to build (infer) dependencies / **functions** / hypotheses from examples of observed data
  - that ***fits*** the know examples
  - able to ***generalize***, with reasonable accuracy for new data
    - According to verifiable results
    - Under statistical and computational conditions and criteria
  - Considering the expressiveness and algorithmic complexity of the models and learning algorithms

Examples of *x* - *f(x)*

Dip. Informatica
University of Pisa

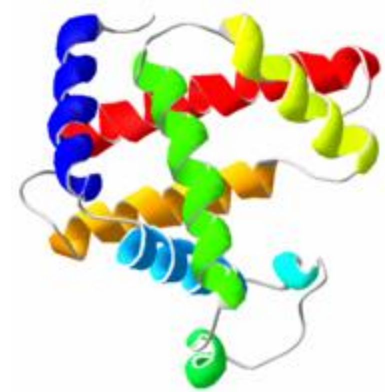
Inferring general functions from know data:

- Handwriting Recognition
  - ***x***: Data from pen motion.
  - *f(**x**)*: Letter of the alphabet.
- Disease diagnosis (from database of past medical records)
  - ***x***: Properties of patient (symptoms, lab tests)
  - *f(**x**)*: Disease (or maybe, recommended therapy)
  - TR Training Set: <***x***,*f(**x**)*>: database of past medical records
- Face recognition
  - ***x***: Bitmap picture of person's face
  - *f(**x**)*: Name of the person.
- Spam Detection
  - ***x***: Email message
  - *f(**x**)*: Spam or not spam.

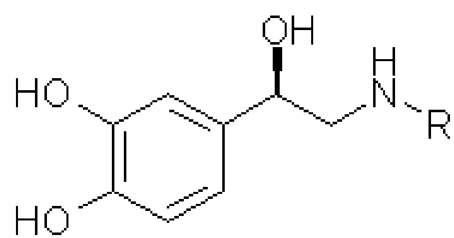
Complex data

Dip. Informatica
University of Pisa

- Protein folding
  - ***x***: sequence of amino acids
  - *f(**x**)*: sequence of atoms' 3D coordinates
  - TR <***x***,*f(**x**)*>: known proteins
  - Type of ***x***: string (variable length)
  - Type of *f(**x**)*: sequence of 3D vectors

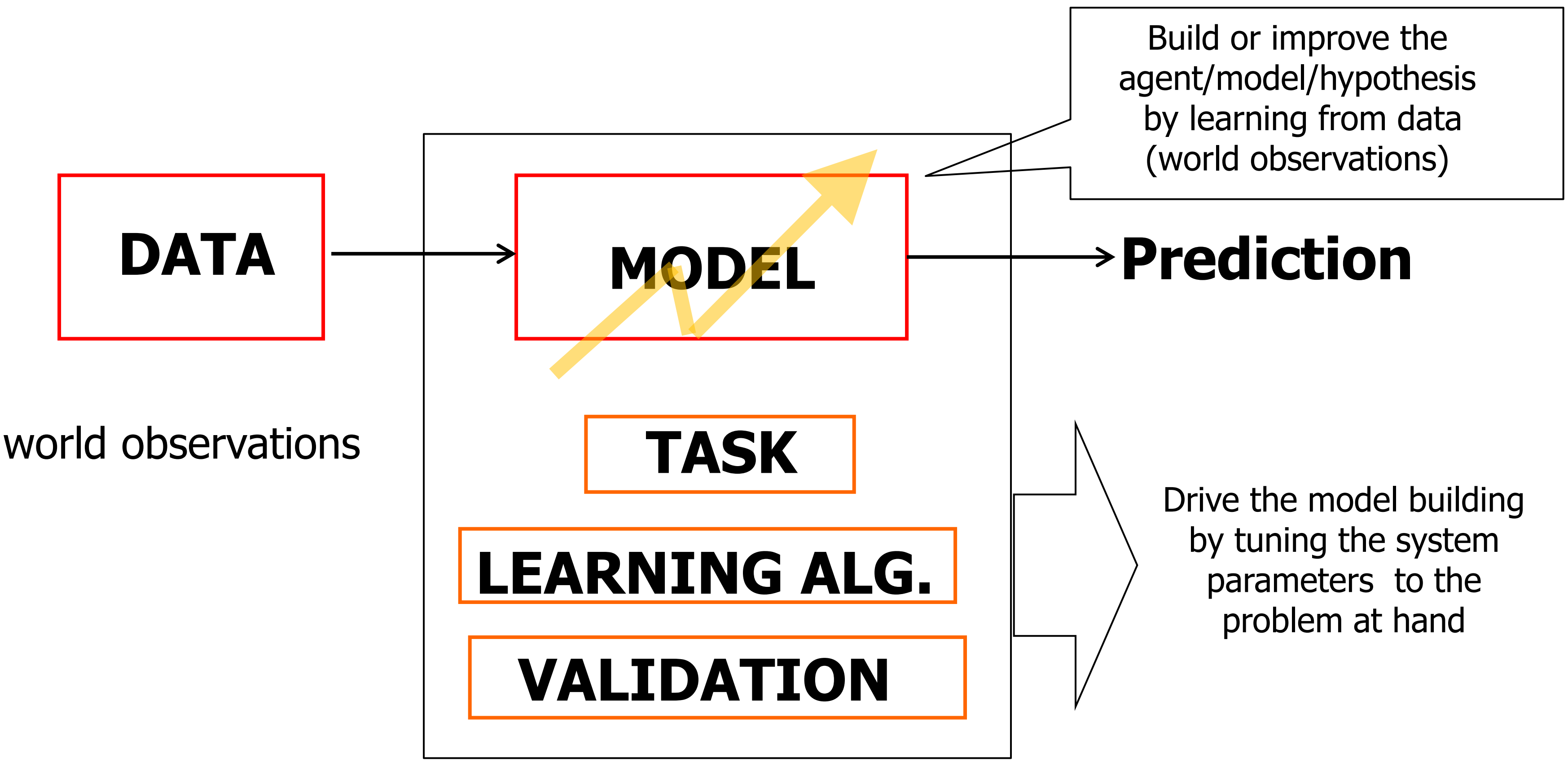


- Drug design
  - ***x***: a molecule
  - *f(**x**)*: binding strength to HIV protease
  - TR <***x***,*f(**x**)*>: molecules already tested
  - Type of **x**: a graph or a relational description of atoms/chemical bonds
  - Type of *f(**x**)*: a real number



Overview of a ML (predictive) System

Dip. Informatica
University of Pisa



Also as a guide to the **key design choices**
(ML system **"ingredients"**)

DATA

Dip. Informatica
University of Pisa

- The data *represent* the available facts (*experience).*
  - Representation problem: to capture the structure of the analyzed objects

Types: Flat, Structured, …

- **Flat** (*attribute-value language*):

fixed-size vectors of properties (*features)*, single table of tuple (measurements of the objects)

Fruits	Weight	Cost $	Color	Bio
Fruit 1 (lemon)	2.1	0.5	y	1
Fruit 2 (apple)	3.5	0.6	r	?

Attributes (categorical/discrete or continuous)

missing data

Data can be subject to
preprocessing: e.g. Variable scaling, encoding*, feature selection…

Examples and terminologies

Dip. Informatica
University of Pisa

Medical records

i

Patients	Age	Smoke	Sex	Lab Test
Pat 1	101	0.8	M	1
Pat 2	30	0.0	F	?

p $\mathbf{x}_p$

Attributes (discrete/continuous)

- Each row ($\mathbf{x}$, vector in bold): example, pattern, instance, sample,….
- Dimension of data set: number of examples l
- Dimension (of the input x): number of features n
- If we will index the features/inputs/variables by *i* or *j* : variable x_i is (typically) the *i-th* feature/property/attribute/element/component of $\mathbf{x}$.

(but may be to simplify we need to use subscript index for other meanings)

- $\mathbf{x}_p$ (or $\mathbf{x}_i$) is (typically) the *p-th* (or *i-th)* pattern/example/raw (vector)
- $x_{p,i}$ (for example) can be the attribute i of the pattern p

DATA Encoding

Dip. Informatica
University of Pisa

Flat case:

- **Numerical encoding for categories: e.g.**
  - 0/1 (or –1/+1) for 2 classes
  - More classes:
    - 1,2,3... Warning: grade of similarity (1 vs 2 or 3): useful for "order categorical" variables (e.g small, medium, large)
    - *1-of-k* (or *1-hot*) encoding: useful for symbols

A	**1**	0	0
B	0	**1**	0
C	0	0	**1**

It will be useful for the project !

Useful both for input or output variables

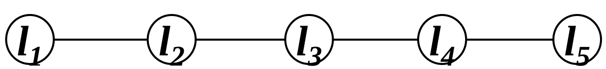
DATA : Structures

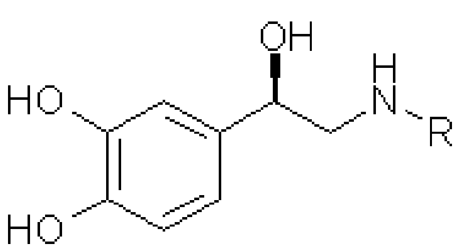
Dip. Informatica
University of Pisa

- **Structured:** Sequences (lists), trees, graphs, Multi-relational data (table) (in DB)

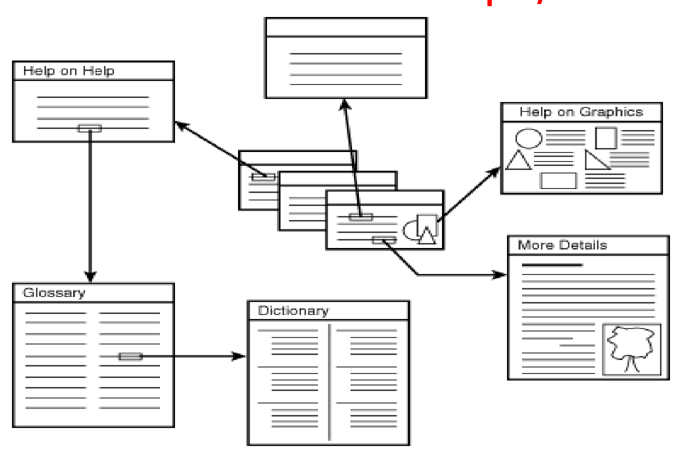
Examples: images, microarray, temporal data, strings of a language, DNA e proteins, hierarchical relationships, molecules, hyperlink connectivity in web pages, ...

Which natural representation?





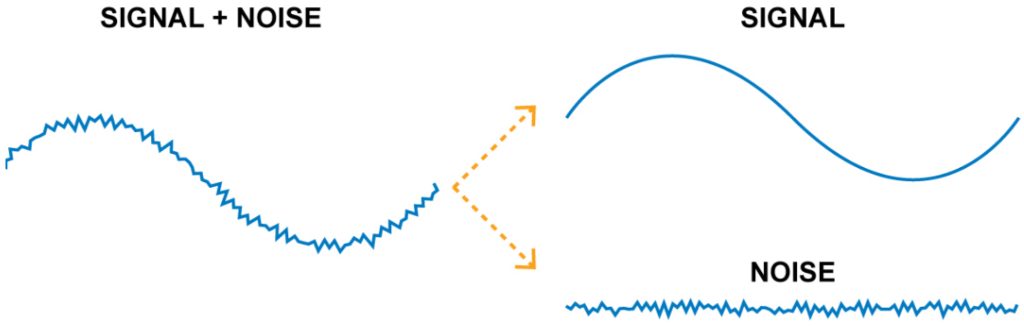
Graph/network data



DATA
Further terminologies

Dip. Informatica
University of Pisa

- ***Noise***: addition of external factors to the stream of (target) information (*signal*); due to randomness in the measurements, not due to the underlying law: e.g. Gaussian noise



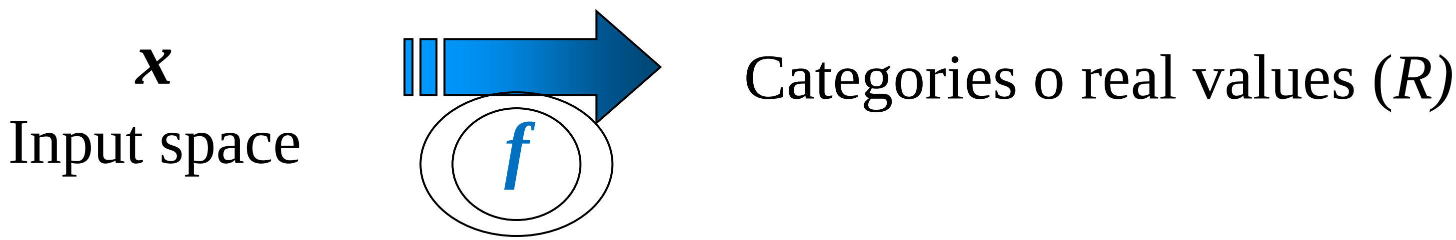
- ***Outliers***: are unusual data values that are not consistent with most observations (e.g. due to abnormal measurements errors)
  - outlier detection – preprocessing: removal
  - Robust modeling methods
- ***Feature selection***: selection of a small number of informative features: it can provide an optimal input representation for a learning problem

TASKS

- The task defines the purpose of the application:
  - Knowledge that we want to achieve? (e.g. pattern in DM or model in ML)
  - Which is the helpful nature of the result?
  - What information are available?

Mainly in the ML course

- **Predictive** (Classification, Regression): function approximation



E.g. recall the *"pilot" example on handwritten digits:* Build a *function* from examples

- **Descriptive** (Cluster Analysis, Association Rules): find subsets or groups of unclassified data

Tasks: Supervised Learning

- Given: Training examples as $<input,output>=<\mathbf{x},d>$ (***labeled* examples**)

  Def

  for an unknown function f (known only at the given points of example)
  - Target value: desiderate value d or t or y … is given by the teacher according to $f(\mathbf{x})$ to label the data
- Find: A *good* approximation to f (a hypothesis h that can used for prediction on unseen data $\mathbf{x}'$, i.e. that is able to generalize)

$\boldsymbol{x}$ Input space $\xrightarrow{f}$ Categories o real values (R)

- Target d (or t or y): a categorical or numerical *label*
  - ***Classification:*** discrete value outputs:

    $f(\mathbf{x}) \in \{1,2,\ldots,K\}$ *classes* (*discrete-valued function*)
  - ***Regression:*** real continuous output values (approximate a real-valued target function, in R or R^K)

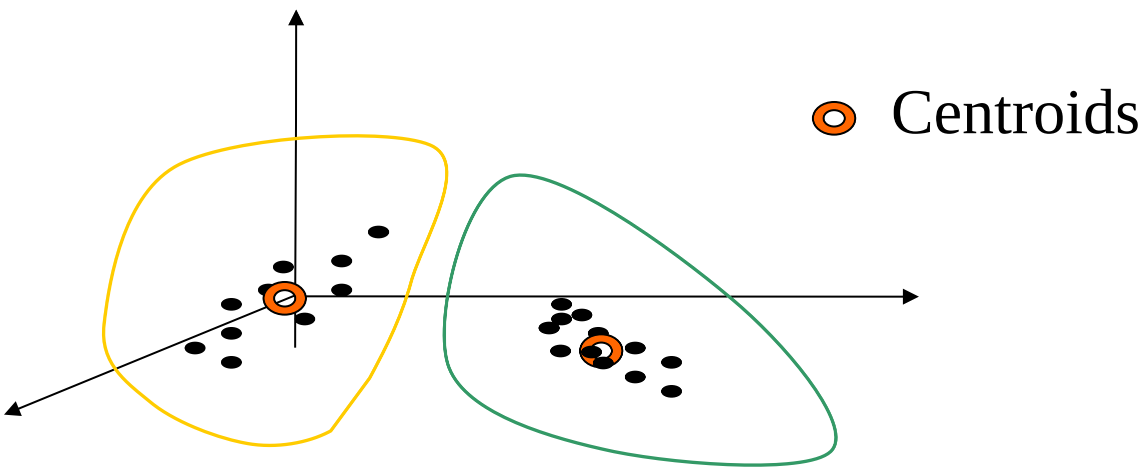
Unified vision thanks to the formalism of a **function approximation** task

Tasks: Unsupervised Learning

Dip. Informatica
University of Pisa

Unsupervised Learning: No teacher!

- TR (Training Set)= set of unlabeled data <x>
- E.g. to find *natural groupings* in a set of data
  - Clustering
  - Dimensionality reduction/ Visualization/Preprocessing
  - Modeling the data density



- Clustering:

Partition of data into clusters (subsets of "similar" data)

Tasks: Classification

Dip. Informatica
University of Pisa

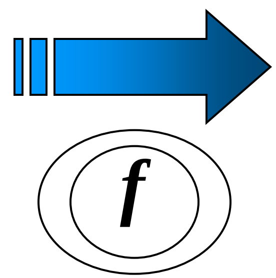
(Supervised) Classification: Patterns (features vectors) are seen as members of a class and the goal is to assign the patterns observed classes (label)

- *Classification*: $f(\mathbf{x})$ return the correct class for $\boldsymbol{x}$
- Number of classes:
  - **=2** : $f(\mathbf{x})$ is a Boolean function: binary classification, **concept learning** (T/F or 0/1 or –1/+1 or negative/positive),
  - **> 2**: multi-class problem (C_1,C_2,C_3 ....C_K)

Example

From DATA to TASK (e.g. classification)

Patients	Age	Smoke	Sex	Lab Test
Pat 1	101	0.8	M	1
Pat 2	30	0.0	F	?



Target: diagnose
+
-

$\boldsymbol{x}$: Input space

Terminology in statistics:

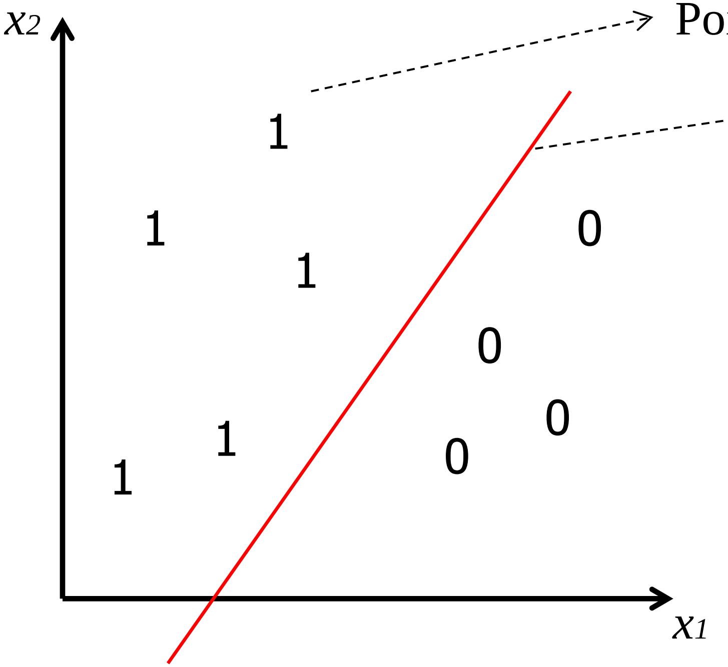
- Inputs are the "independent variables"
- Outputs are the "dependent variables" or "responses"

Tasks: Classification

Dip. Informatica
University of Pisa

The classification may be viewed as the allocation of the input space in decision regions (e.g. **0/1**)

<u>Example</u>: graphical illustration of a linear separator on a instance space $\mathbf{x}^T=(x_1,x_2)$ in IR^2 , $f(\mathbf{x})=0/1$ (or $-1/+1$)



Point belonging to class 1

Separating (hyper)plane : $\mathbf{x}$ s.t.

PREVIEW

$$\mathbf{w}^T\mathbf{x} + w_0 = w_1x_1 + w_2x_2 + w_0 = 0$$

$$\boxed{h(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad \mathbf{w}^T\mathbf{x} + w_0 \geq 0 \\ 0 & \text{otherwise} \end{cases}}$$

or

$$\boxed{h(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + w_0)}$$

Linear threshold unit (LTU)

Indicator functions

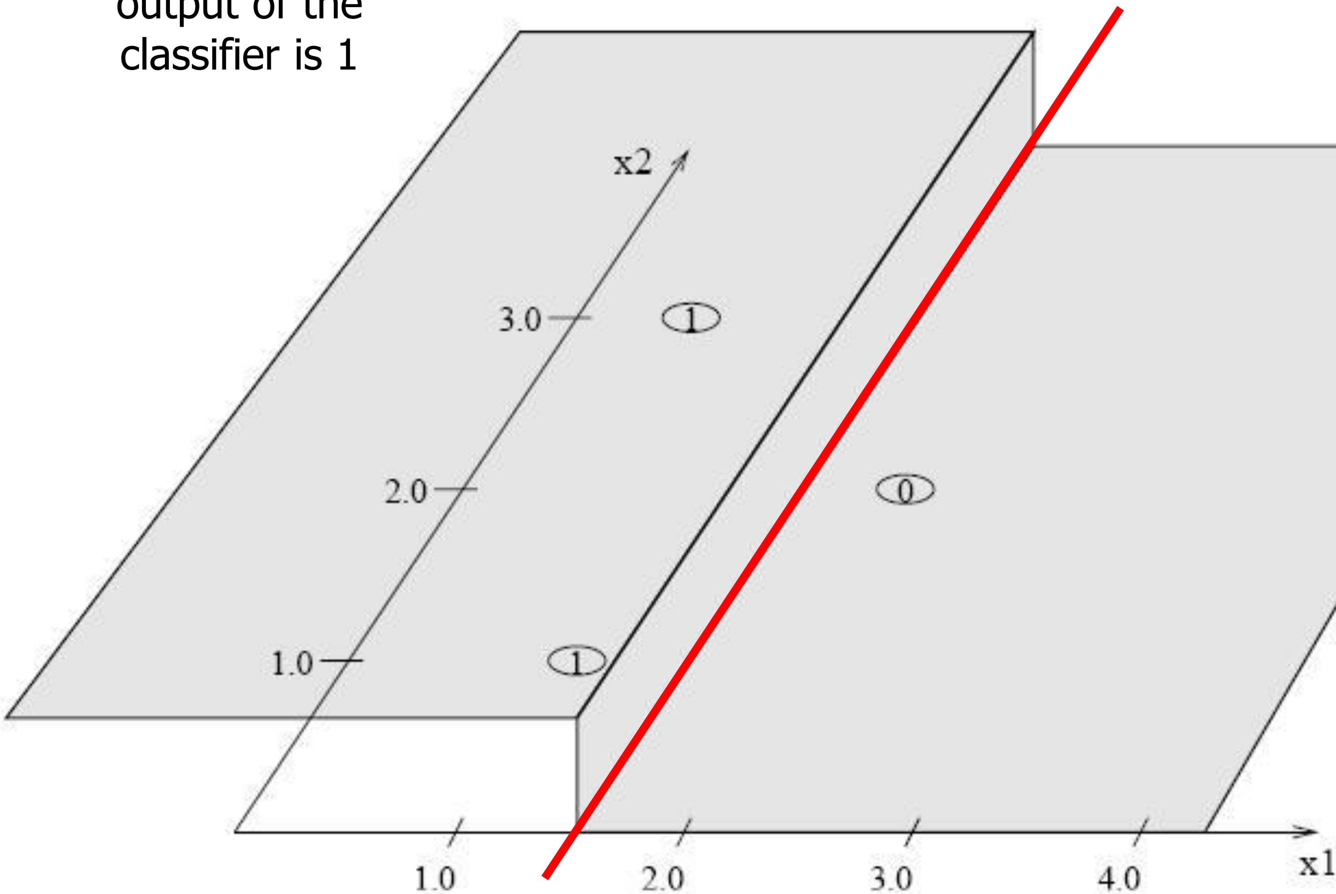
How many? (H): set of dichotomies induced by hyperplanes

Geometrical 3D (pre)view: Classifier

Dip. Informatica
University of Pisa

The 0/1 classification function in 3D
(on a 2D input space)

Region where the
output of the
classifier is 1



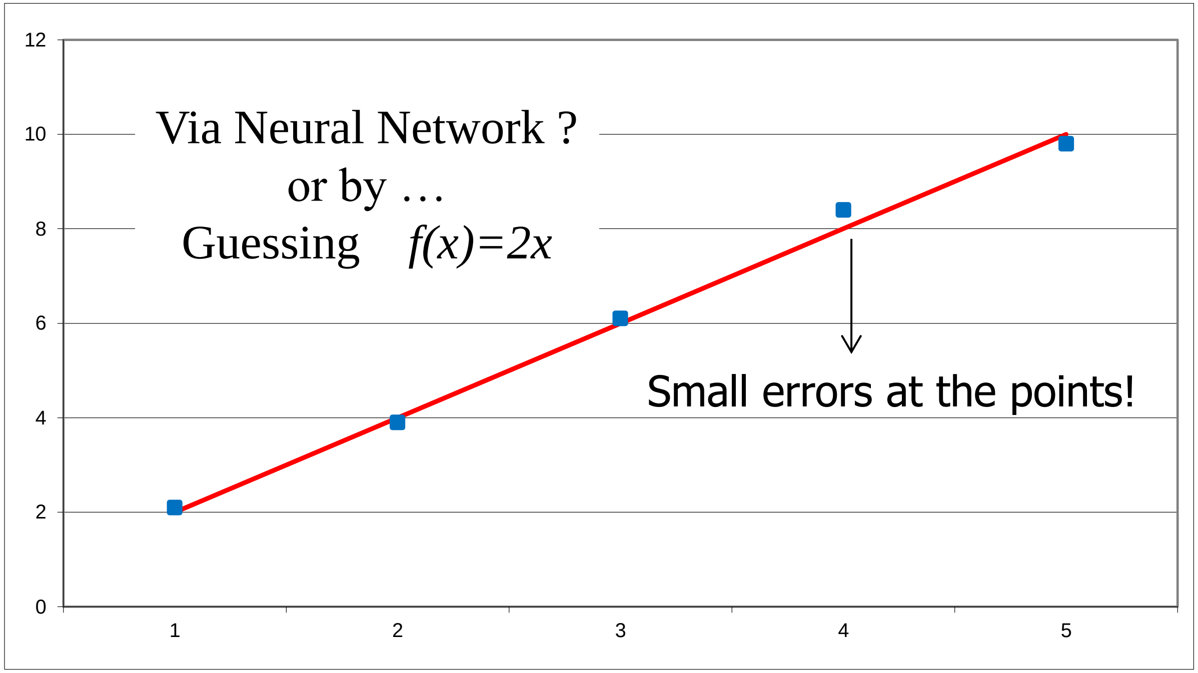
Tasks: Regression: example

Dip. Informatica
University of Pisa

- Process of estimating of a real-value function on the basis of finite set of noisy samples (supervised task)
  - known pairs (***x***, *f(**x**)+random noise*)

Task (exercise): find *f* for the data in the following table:

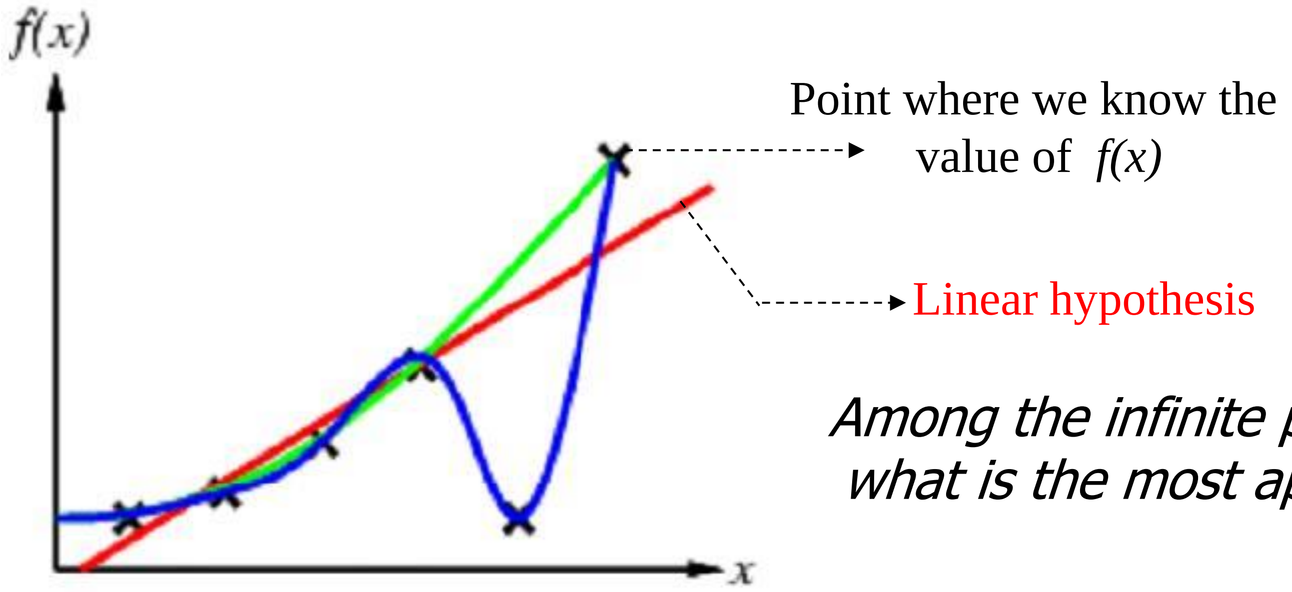
x	*target*
1	2.1
2	3.9
3	6.1
4	8.4
5	9.8
...	...



Dip. Informatica
University of Pisa

Tasks: regression

- *Regression:* $\mathbf{x}$ = variables (e.g. real values), $f(\mathbf{x})$ *real values*: curve fitting (x is 1-dim in the example but it becomes *k-dim* in general)
- Process of estimating of a real-value function on the basis of finite set of noisy samples
  - known pairs ($\mathbf{x}$, $f(\mathbf{x})$+*random noise*)



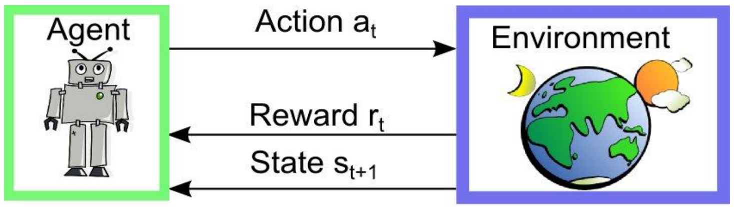
Among the infinite possibilities, what is the most appropriate?

An example (linear hypothesis): $h_w(x)=w_1x+w_0=0.2\ x\ -0.4$

Tasks: Other Topics ...

- **Semi-supervised learning**
  - combines both labeled and unlabeled examples to generate an appropriate function or classifier.

- **Reinforcement Learning** (learning with right/wrong critic).
  - Adaptation in *autonomous systems*
  - "the algorithm learns a *policy* of how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback that guides the learning algorithm".
  - Not step by step examples
  - Toward decision-making aims
  - Useful in modern AI



Reinforcement Learning Setup

Models
and survey of useful concepts

Dip. Informatica
University of Pisa

- MODEL:
  - Aim: to capture/describes the relationships among the data (on the basis of the task) by a "language" (numerical, symbolic, …)
  - The "language" is related to the *representation* used to get knowledge
  - The model defines the class of functions that the learning machine can implement (*hypotheses space*)
    - E.g. set of functions $h(\boldsymbol{x},\boldsymbol{w})$, where $\boldsymbol{w}$ is the (abstract) parameter

- **Training example** (superv.): An example of the form ($\boldsymbol{x}$, $f(\boldsymbol{x})$+*noise*)
  $\boldsymbol{x}$ is usually an input vector of features, *(d or t or)* $y=f(\boldsymbol{x})+noise$ is called the target value
- **Target function**: The true function *f*
- **Hypothesis**: A proposed function *h* believed to be similar to *f*. An expression in a given *language* that describes the relationships among data
- **Hypotheses space** H: The space of all hypotheses (specific models) that can, in principle be output by the learning algorithm

Models: few trivial examples....

Dip. Informatica
University of Pisa

Just to have a preview of different *representation* of hypothesis (because you already know the *language* of equations, logic, probability):

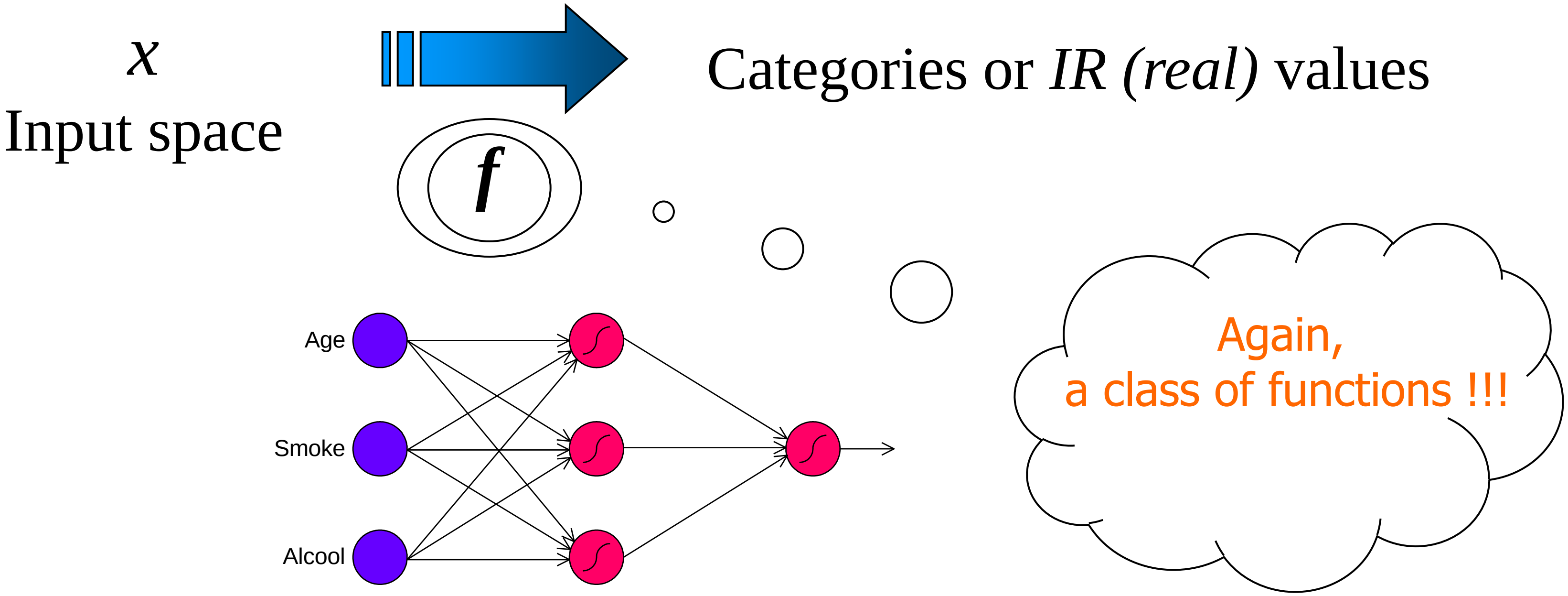
- **Linear models** (representation of H defines a **continuously** parameterized space of potential hypothesis);
  each assignment of w is a different hypothesis, e.g:
  - $h(\mathbf{x}) = sign(\mathbf{w}^T \boldsymbol{x} + w_0)$ *binary classifier*

    $h_{\mathbf{w}}(x)=w_1x+w_0$ E.g. $h_{\mathbf{w}}(x)= 2x+150$ *simple linear regression*
- **Symbolic Rules**: (hypothesis space is based on **discrete** representations); different rules are possible , e.g:
  - *if* $(x_1=0)$ *and* $(x_2=1)$ *then* $h(\mathbf{x})=1$
  - *else* $h(\mathbf{x})=0$

  binary classifier
- **Probabilistic models**: estimate $p(x,y)$
- K Nearest neighbor regression: Predict mean y value of nearest neighbors (memory-based)

Neural Networks (just a look)

An example: we will see a **neural networks**, beyond the *neurobiological inspiration*, as a computational model for the treatment of data, capable of approximating complex (*non-linear*) relationships between inputs and outputs



Paradigms and methods (Languages for H)

Dip. Informatica
University of Pisa

- Symbolic and Rule-based (or *discrete* H)
  - Conjuction of literals*, Decision trees (propositional rules)
  - Inductive grammars, Evolutionary algorithms, …
  - Inductive Logic Programming (first order logic rules)
- Sub-symbolic (or *continuous* H)
  - Linear discriminant analysis, Multiple Linear Regression*, LTU
  - Neural networks
  - Kernel methods (SVMs, gaussian kernels, spectral kernels, etc)
- Probabilistic/Generative
  - Traditional parametric models (density estimation, discriminant analysis, polynomial regression,…)
  - Graphical models: Bayesian networks, Naïve Bayes, PLSA, Markov models, Hidden Markov models, …
- Instance-based
  - Nearest neighbor*

Note: Underlined – >ML
1. Some models can be expressed by different languages
2. * Next lectures

How many models?

- Theory (**No Free Lunch Theorem)** : *there is no universal "best" learning method (without any knowledge, for any problems,…):*

  if an algorithm achieves superior results on some problems, it must pay with inferiority on other problems. In this sense there is no free lunch.

  E.g. Devroye (1982), Wolpert and Macready (1997), and others

→ The course provides a
  - set of models and the
  - critical instruments to compare them

- However, not all the models are equivalent:
  - Important differences are for the **flexibility** of the approaches, toward models that can in principle approximate arbitrary functions (e.g. no just linear approximation seen in the examples)
  - Important differences are for the **control of the complexity** (we will see later)
  - Use of flexible models and principia for the control of the complexity are the core of ML

Learning Algorithms

Dip. Informatica
University of Pisa

- LEARNING ALGORITHM   Basing on data, task and model

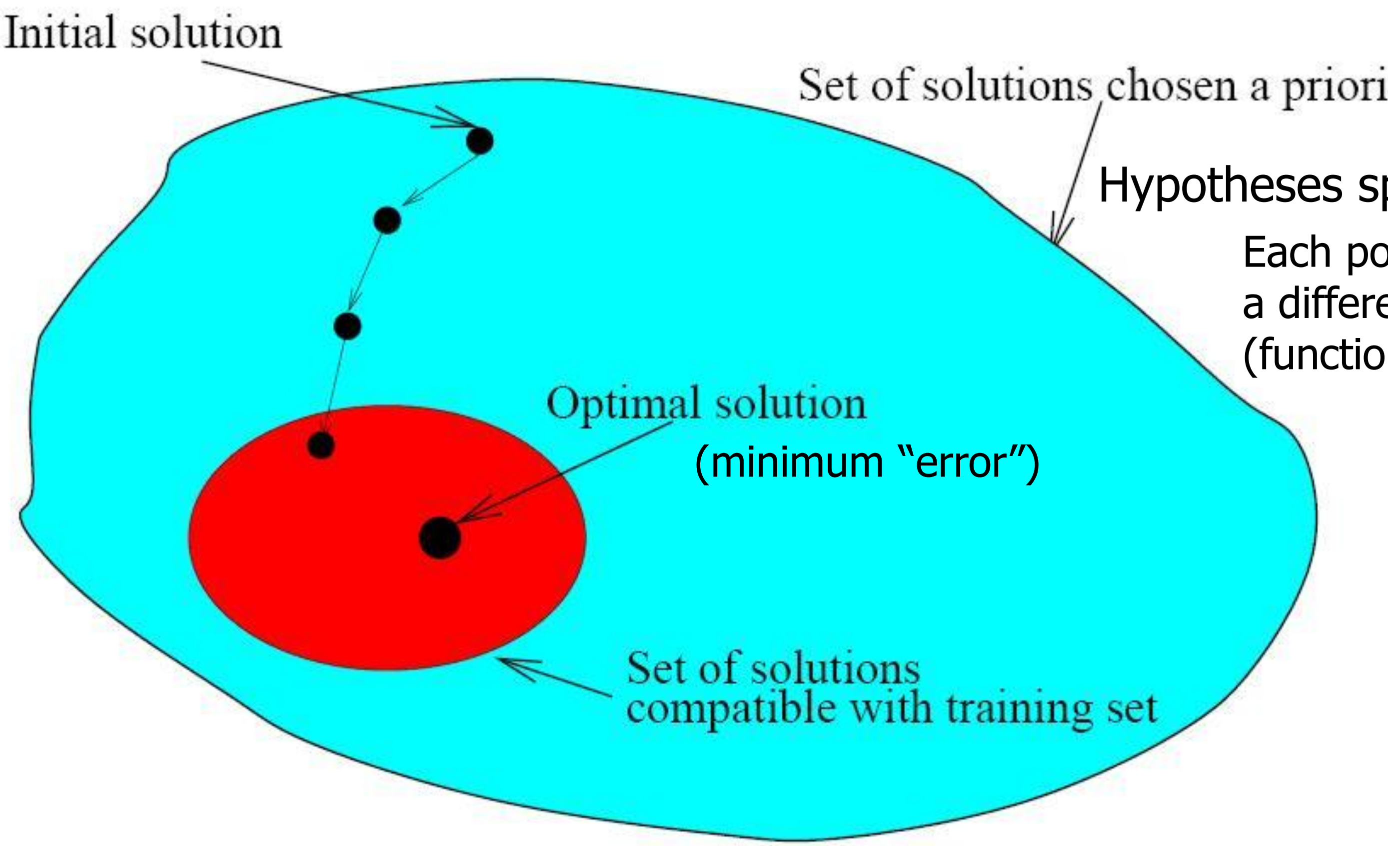
- (Heuristic) *search through the hypothesis space* **H** *of the* ***best hypothesis***
  - i.e. the best approximation to the (unknown) target function
  - Typically searching for the *h* with the *minimum "error"*

  - E.g. free parameters of the model are fitted to the task at hand:
  - Examples: best $\mathbf{w}$ in linear models, best rules for symbolic models, ...
  - Remember the regression example, we proposed $h(x)=2x$, for $h_{\mathbf{w}}(x)=w_1x+w_0$ assuming $w_1=2$ and $w_0=0$ as the best parameter value: *how?*

- **H** may not coincide with the set of all possible functions and the search can not be exhaustive: we need to make assumptions → (we will see the role of) *Inductive bias*

Learning Algorithms: search

Dip. Informatica
University of Pisa



Hypotheses space *H*

Each point represents a different hypothesis (function)

Typically local search approaches

Learning (terminologies)

Dip. Informatica
University of Pisa

According to the different paradigms/contexts "learning" can be differently termed or have different acceptations:

- Inference (statistics)
- Inference: Abduction/Induction (logic)
- Adapting (biology, systems)
- Optimizing (mathematics)
- Training (e.g. Neural Networks)
- Function approximations (mathematics)

Can be more specifically found in other sub-fields:

- Regression analysis (statistics), curve fitting (math, CS), …
- Or using other terminologies e.g. "*Fitting* a multivariate function"

Recap and next topics

Dip. Informatica
University of Pisa

After the introduction of the first four ingredients (Data, Task, Model and Learning Alg.), we need to focus on three mentioned *relevant concepts* not yet discussed so far:

1. The *inductive bias* (examples in discrete hypothesis spaces)
2. The *loss,* used to measure the quality of our approximation
3. The concept of *generalization* and *validation* (next lecture)

1. The Role of the Inductive Bias

Dip. Informatica
University of Pisa

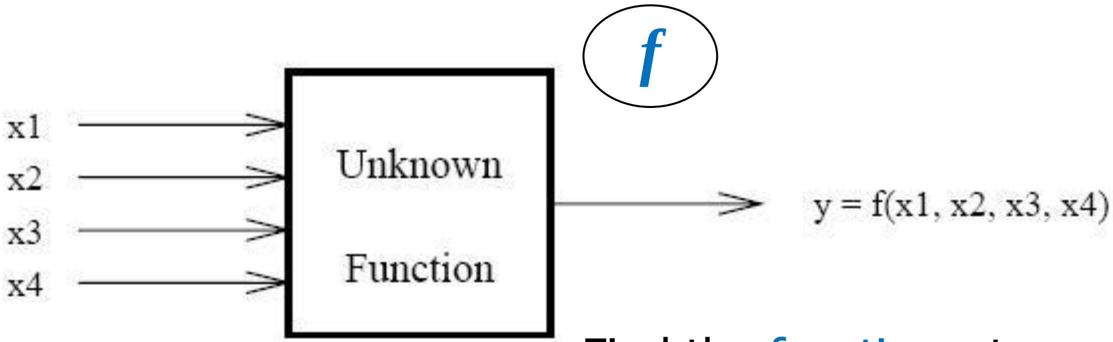
In order to set up a model and a learning algorithm we can make *assumptions* (about the nature of the target function) concerning either

- Constraints in the model (in the hypothesis space H, due to the set of hypotheses that we can express or consider) (**Language Bias**)
- Constraints or preferences in learning algorithm/search strategy (**Search Bias**)
- Or **Both**.

- We will see that such assumptions are strictly need to obtain an useful model for the ML aims, i.e. a model with generalization capabilities
- We start to discuss it within examples in *discrete hypotheses spaces (rules)*, ***learning a concept*** (a Boolean function) [Mitchell chapt. 2]
  - E.g. $\boldsymbol{x}$ is a "cat" if $h_{cat}(\boldsymbol{x}) = 1$, otherwise is 0 for $\boldsymbol{x}$ in "animals"

An example:
Learning Boolean functions

Dip. Informatica
University of Pisa



Find the function s.t.

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Table 1

This is an **ill posed** (inverse) problem:
We may violate either existence, **uniqueness**, stability of the solution or solutions

Learning Boolean functions: ill-posed

Dip. Informatica
University of Pisa

- There are $2^{16} = 2^{2^4} = 65536$ possible Boolean functions over four input features. We can not figure out which one is correct until we have seen every possible input-output pair.
- After 7 examples, we still have 2^9 possibilities.

- In the general case, in this discrete hypothesis space H: $|H| = 2^{\text{\#-input-instances}} = 2^{2^n}$

  for binary inputs/outputs, *n= input dimension*

  Lookup table *model* →

- I.e. a rote learner: Store/memorize examples, classify ***x*** if and only if it matches a previously observed example (else ”no answer”).
  - No inductive bias → *no generalization!*

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

Another discrete H space: Conjutive rules

Dip. Informatica
University of Pisa

- As second example of discrete H, we can image to learn a discrete function with discrete inputs assuming ***conjunctive rules*** (propositions with AND among literals, a language bias)
- i.e. using a *language bias* to work with a restricted hypothesis space
- E.g. $h_1 = l_2$, $h_2 = (l_1$ ***and*** $l_2)$, $h_3 = true$, $h_4 =$ **not**(l_1) ***and*** l_2 , …
  - Rules such as *if* $l_2(=true)$ *then* $h(\mathbf{x})=true$, *else* $h(\mathbf{x})=false$
    or equivalently *if* $(x_2=1)$ *then* $h(\mathbf{x})=1$, *else* $h(\mathbf{x})=0$

- With n binary inputs we had $|H| = 2^{\#\text{-input-instances}} = 2^{2^n}$
- With only conjunctive rules:

#semantically distinct hypotheses (conjunctions):

3^n (for each of the n positions we can have l_i, **not**(l_i), *don't care*) + 1

(+1 because all h with (l_i *AND* **not**(l_i)) are equivalent to "*false*")

(e.g. from 65536 to just $3^4+1=82$ in the example with $n=4$)

Find the Version Space

Dip. Informatica
University of Pisa

- Given the def.: a hypothesis h is **consistent** with the TR, if $h(\mathbf{x})=d(\mathbf{x})$ for each training example $<\mathbf{x},d(\mathbf{x})>$ in TR.

- It is possible to perform a *complete search* (finding the set of *all* h consistent with the TR set) *in an efficient way* in this reduced space (of conjunctive rules) by cleverer algorithms (Mitchell chap. 2)
  - Instead of searching enumerating all the possible combination of literals, i.e. every h in H

- We are only interested to say that these algorithms find the VS:
- Call the **version space**, $VS_{H,TR}$, with respect to hypothesis space H, and training set TR, the *subset of hypotheses* from H *consistent* with all training examples

Unbiased Learner I

Dip. Informatica
University of Pisa

- Hence, this conjunctive assumption for H leads to an efficient solution in finding a VS.
  However, using only conjunctive rules may be ***too restrictive***: if the target concept is not in H, it *cannot be* represented in H.
  - e.g. *if* $(x_1=1)$ **or** $(x_2=1)$ *then* $h(\mathbf{x})=1$, *else* $h(\mathbf{x})=0$
- **Idea**: Choose H that expresses every teachable concept (among propositions), that means H is the set of all possible subsets of X (*instance* or *input* space): the power set P(X)
- E.g. n=10 binary inputs $|X|= 2^{10}=1024$, $|P(X)|=2^{1024} \sim 10^{308}$ distinct concepts (much more than the num. of atoms in the universe)
- H = disjunctions, conjunctions, negations
- H surely contains the target concept.
- *What for generalitazion ?*

Unbiased Learner II (formal)

Dip. Informatica
University of Pisa

Recall that the **version space**, $VS_{H,TR}$, with respect to hypothesis space H, and training set TR, is the subset of hypotheses from H consistent with all training examples

The only examples that are unambiguolsy classified by an ***unbiased learner*** represented with the VS are the training examples themselves I.e. the *lookup table* !

Property: An unbiased learner is *unable to generalize* (on new instances):
Proof: Each unobserved instance will be classified 1 (positive) by precisely half the hypothesis in VS and 0 (or negative) by the other half (*rejection:* no answer is made by the VS for new input instances).
Indeed:
$\forall$h consistent with x_i (test), $\exists$ h' identical to h except $h'(x_i) <> h(x_i)$,
$h \in VS \rightarrow h' \in VS$ (because they are identical on TR)

Futility of Bias-Free Learning

Dip. Informatica
University of Pisa

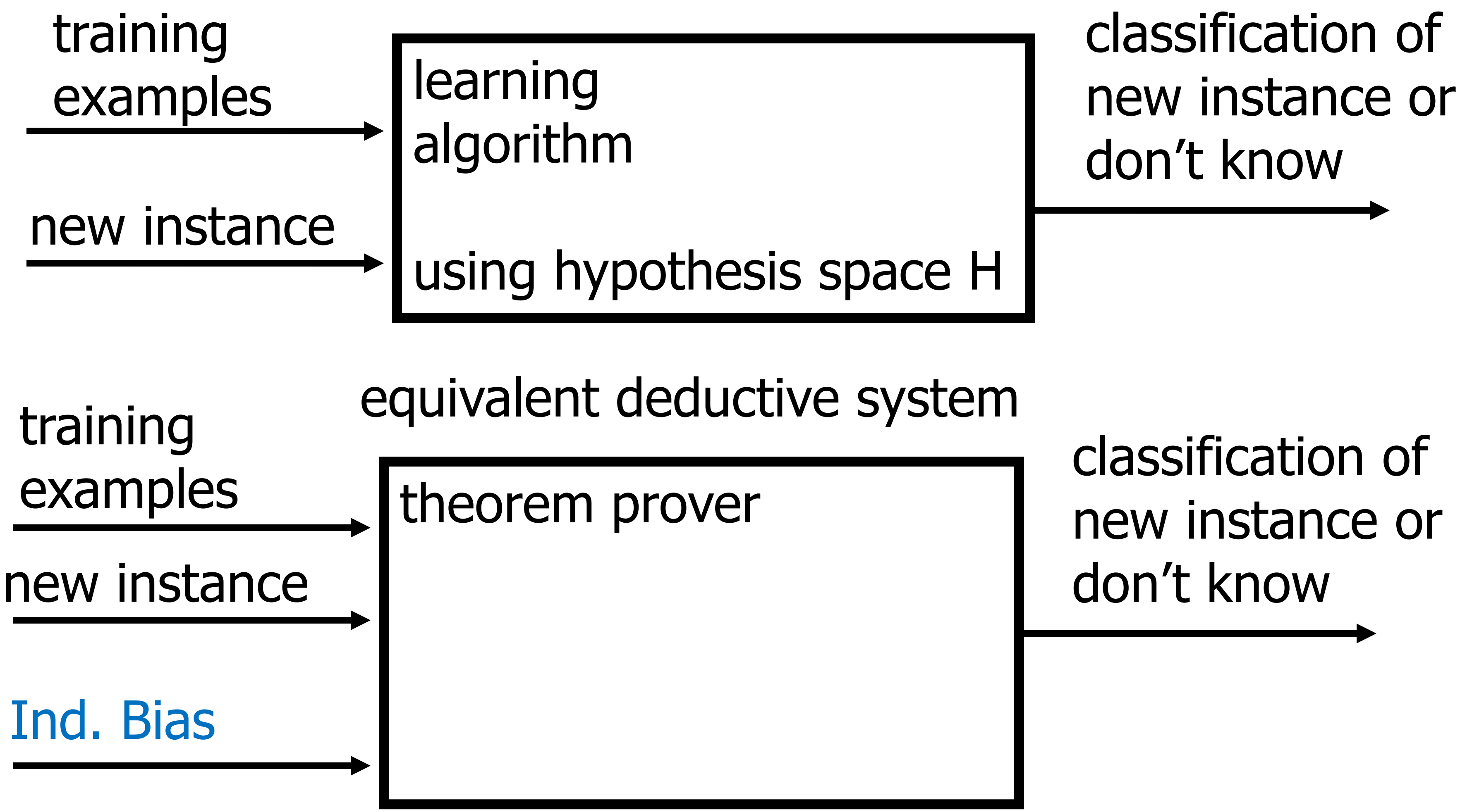
- A learner that makes no prior assumptions regarding the identity of the target function/concept has no rational basis for classifying any unseen instances.
- (Restriction, preference) bias *not* only assumed for *efficiency*, *it is needed for the generalization capability*
  - However, it does no tell us (quantify) which one is the best solution for generalization yet
- **Trivial Example** (TR= Training Set, TS= Test Set): **:**

	X	*d(x)*	
			H={*x*, not(*x*), **0**, **1**}
TR	0	0	VS={*x*,**0**}
TS	1	?	→ Can be 1 or 0 … Unless you use all X as TR set.

In other words, in order to learn the target concept, one would have to present every single instance in X as a training example (lookup table)

Inductive Systems and Equivalent Deductive Systems

Dip. Informatica
University of Pisa



Language or search bias?

Dip. Informatica
University of Pisa

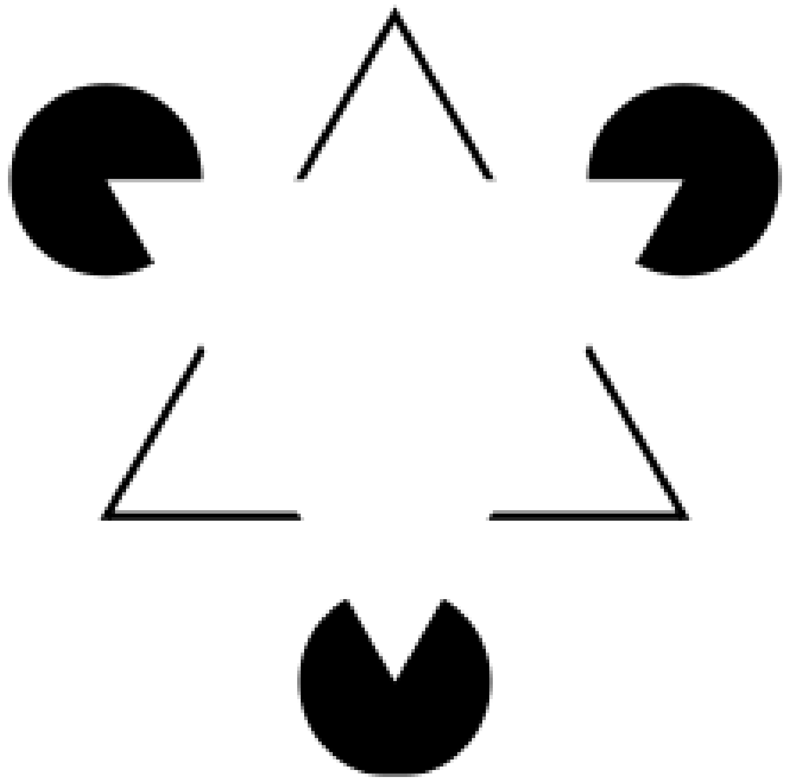
Why the *search bias* can be preferred over the *language bias*?

- In ML typically use ***flexible*** *approaches* (expressive hypothesis spaces, universal capability of the models, e.g. Neural Networks, DT)
- avoiding the *language bias*, hence <u>without excluding a priori the unknown target function</u>,
- retaining an inductive bias but focusing on the *search bias* (which is ruled by the learning algorithm).
  - In practice using an *incomplete* search strategy.

Conclusions:

- Learning without bias cannot extract any regularities from data (lookup-table: no generalization capabilities)
- Every state-of-the-art ML approach shows an *inductive bias*
- Issue: characterize the bias for different models/learning approaches

The Kanizsa triangle
Example of perception bias of our visual system



2. Tasks & Loss

Dip. Informatica
University of Pisa

We said … A "***good***" approximation to f from examples.

How to measure the quality of the approximation?

- Recall that we produce $h(\boldsymbol{x})$ value (output of the model for input $\boldsymbol{x}$)
- We want to measure the "distance" between $h(\boldsymbol{x})$ and d
(objective function for minimization of errors in training, check of errors in test)

We use a ("inner") *loss function/measure*: $L(h_w(\boldsymbol{x}),d)$ (for a pattern $\boldsymbol{x}$)
e.g. high value → poor approximation

The *Error* (or *Risk or Loss*) is an expected value of this L
e.g. a "sum" or mean of the inner loss L over the set of samples

$$Loss(h_{\boldsymbol{w}}) = E(\boldsymbol{w}) = \frac{1}{l}\sum_{p=1}^{l} L(h_{\boldsymbol{w}}(\boldsymbol{x}_p), d_p)$$

Note:
index p is used for the samples $p=1..l$

We will change L for different tasks

Note: at moment Error, Risk and Loss are considered equivalent, we will specify differences later through the course

Tasks: Common Tasks review

Dip. Informatica
University of Pisa

I will show a short survey of common learning *tasks* by specifying the (changing of the) nature

- of the output and hypothesis space
- of the *loss function (in particular of* L*)*,

i.e. Examples of loss functions: use it for future reference

Regression

Dip. Informatica
University of Pisa

- Regression: *predicting a numerical value*

- **Output**: $d_p = f(\mathbf{x}_p) + e$ *(real value function + random error)*
- **H**: a set of real-valued functions

- **Loss function** L : measures the approximation accuracy/error
- A *common* loss function for regression: the squared error

$$\boxed{L(h_{\boldsymbol{w}}(\boldsymbol{x}_p), d_p) = (d_p - h_{\boldsymbol{w}}(\boldsymbol{x}_p))^2}$$

- The mean over the data set provide the *Mean Square Error (MSE)*

MSE example

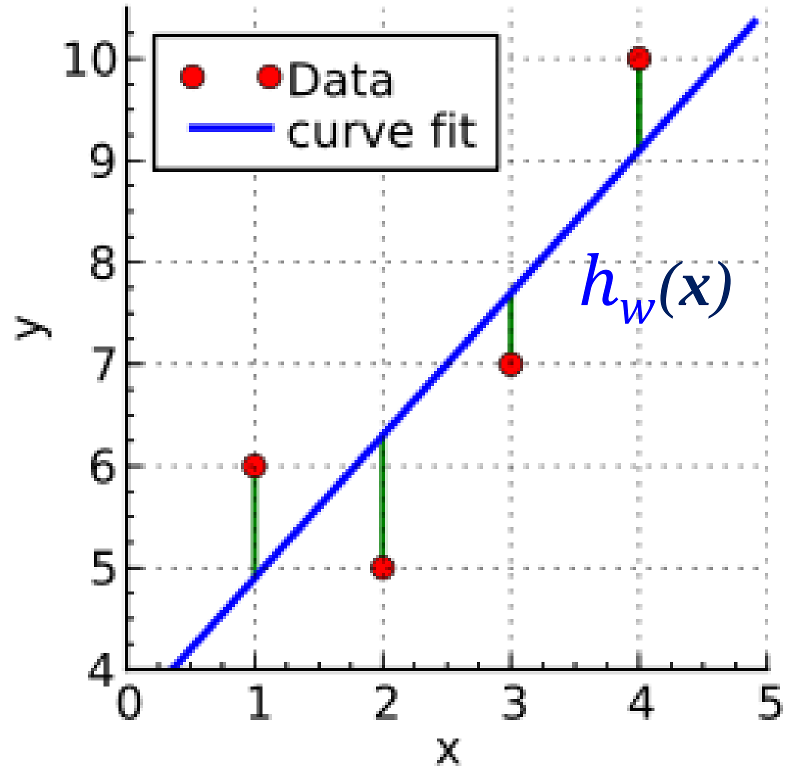
Dip. Informatica
University of Pisa

In the example we have
$h(x)=w_1x+w_0$ as the blue line
and in green the errors at the data points (x_i, y_i) (in red), where the target d_i for x_i is denoted y_i in the example

The Mean Square Error (MSE)
is the mean of the square of the green errors:

$$E(w) = \frac{1}{l}\sum_{p=1}^{l}\left(y_p - h_w(\boldsymbol{x}_p)\right)^2$$

$\boldsymbol{w}$ are the free parameters of the linear model



Note: this plot is taken elsewhere, I used different colors before: here the line is in blue. Also, the y are therein the desidered (target d) values

Classification

Dip. Informatica
University of Pisa

- Classification of *data into discrete classes*

- **Output**: e.g. {0,1}
- **H**: a set of indicator functions

- **Loss function** L : measures the classification error

Def

$$L(h_{\boldsymbol{w}}(\boldsymbol{x}_p), d_p) = \begin{cases} 0 & if \quad h_{\boldsymbol{w}}(\boldsymbol{x}_p) = d_p \\ 1 & \quad otherwise \end{cases}$$

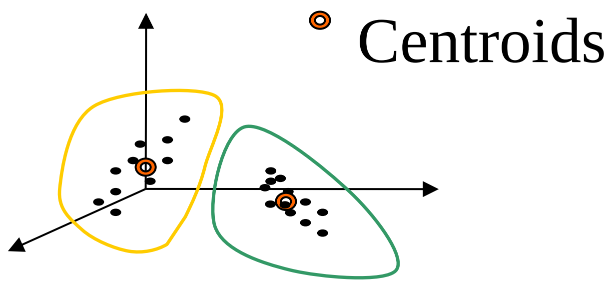
0/1 Loss

- The mean over the data set provide the *number/percentage of misclassified patterns*
- *E.g. 20 out of 100 are misclassified → 20% errors, i.e. 80% of* ***accuracy***

Clustering and Vector Quantization* *preview*

Dip. Informatica
University of Pisa

- Goal: optimal partitioning of unknown distribution in **x**-space into regions (clusters) approximated by a cluster center or *prototype*.



- **H**: a set of vector quantizers $x \rightarrow c(x)$

  continuos space → discrete space

- Loss function L : measures the vector quantizer optimality
- A *common* **loss function** *would be* the *squared error distortion*:

$$L(h(\mathbf{x}_p)) = (\mathbf{x}_p - h(\mathbf{x}_p)) \bullet (\mathbf{x}_p - h(\mathbf{x}_p))$$

$$\bullet = inner\_product$$

We'll see later

Proximity of the pattern to the centroid of its cluster

Density estimation* *preview*

Dip. Informatica
University of Pisa

- Density estimation (generative, "parametric methods")
  from an assumed class of density

- **Output**: a density e.g. normal distribution with mean m and variance $sigma^2$: $p(x \mid m, sigma^2)$
- **H**: a set of densities (e.g. m and $sigma^2$ are the two unknown *parameters*)

- A *common* **loss function** L for density estimation:

$$\boxed{L(h(\mathbf{x}_p)) = -\ln(h(\mathbf{x}_p))} \longrightarrow$$

We'll see later

- Related to "maximizing the (log) likelihood function". [not hear]
- E.g. $P(x_1,x_2,x_3,\ldots \mid m, sigma^2)$

3. Machine Learning & generalization

Dip. Informatica
University of Pisa

This is a fundamental concept of the course

- *Learning*: search for a *good function* in a function space from known data *(typically minimizing an Error/Loss)*

- Good w.r.t. generalization error: it measures how accurately the model predicts over novel samples of data
(*Error/Loss measured over new data)*

Generalization: crucial point of ML!!!

Easy to **use** ML tools *versus* **correct/good use** of ML

Generalization

Dip. Informatica
University of Pisa

- **Learning** phase **(training, fitting)**: build the model from know data – *training data* (and bias)
- **Predictive** or **Test** phase (deployment/ *Inference* use of the ML built model): apply the model to new examples:
  - we take the new inputs ***x'*** and we compute the response by the model
  - we compare with its target *d'* that the model has never seen
  - i.e. we make evaluation of the generalization capability of our predictive hypothesis

Note: *performance* in ML = generalization accuracy/ *predictive accuracy*

estimated by the error computed on the (hold out) Test Set

- **Theory**: E.g. Statistical Learning Theory [Vapnik] :
  - *under what (mathematical) conditions is a model able to generalize?* → see next lecture (just basic notions)

Validation

Dip. Informatica
University of Pisa

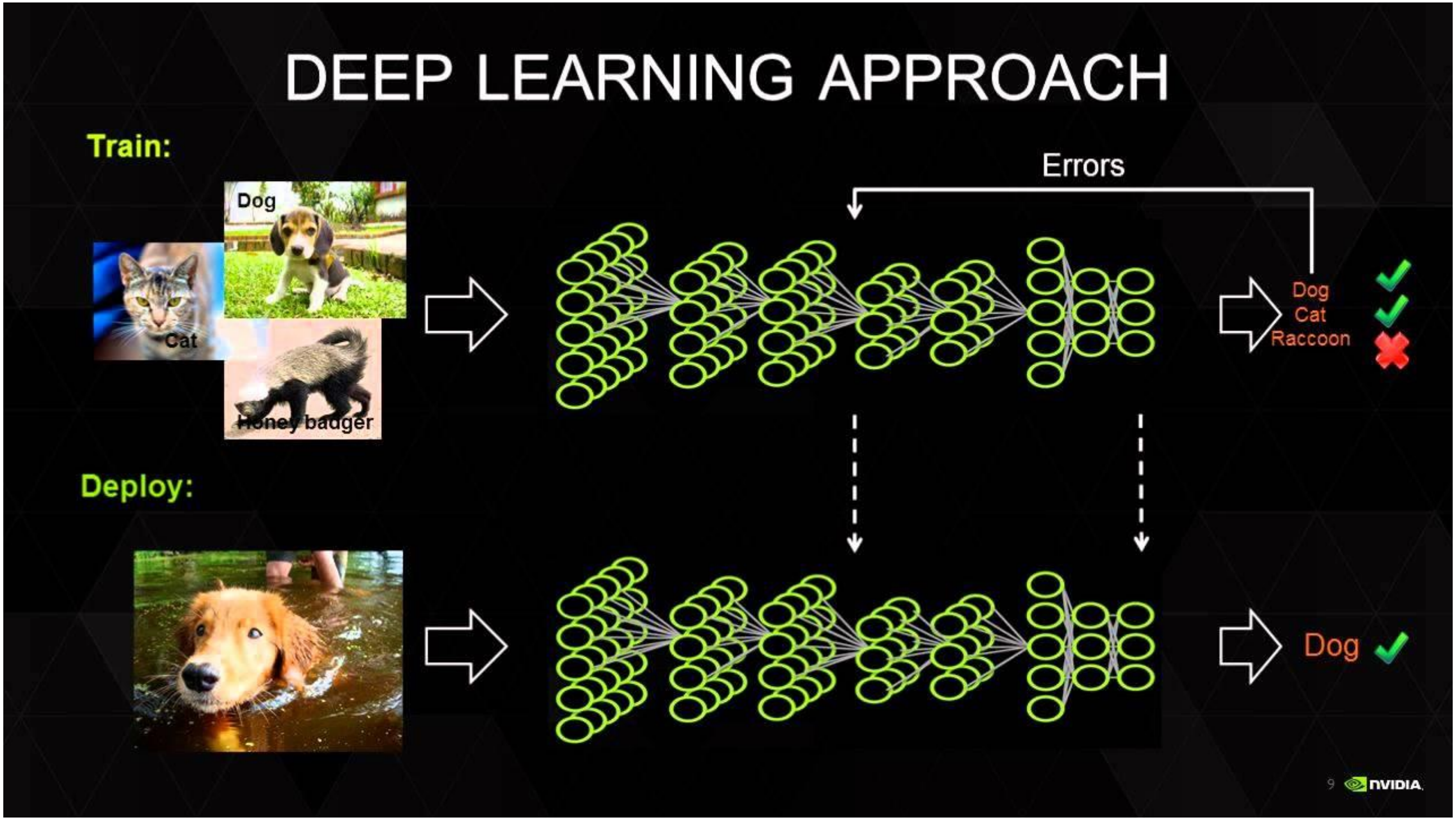
- Evaluation of performances for ML systems =
  Generalization/Predictive accuracy evaluation, i.e.:

- Validation !
- Validation !!
- Validation !!!

- In the following (next lecture) we will discuss some **validation techniques**
  - to evaluate (model assessment) and
  - to manage the generalization capability (model selection).

Exemplification of the Deployment/ *Inference* use

Dip. Informatica
University of Pisa



Exemplification of the Deployment/ *Inference* use

Dip. Informatica
University of Pisa

Even the inference part can be costly if you have millions of requests (e.g. at google)

A Google server rack containing multiple **Tensor Processing Units**, a special-purpose chip designed specifically for machine learning
The original TPU was designed specifically to work best with Google's TensorFlow.

Just for inference (mapping) !!!!

Summary of the Intro to ML

Dip. Informatica
University of Pisa

- Part I (now)
  - Motivations, contextualization in CS
  - Course info
- Part II (in Lect.s 2 and 3)
  - Utility of ML
  - Learning as function approximation (pilot example)
  - Design components of a ML system, including
    - Learning tasks
    - Hypothesis space (and first overview)
    - Inductive bias (examples in discrete hypothesis spaces)
    - Loss and learning tasks
    - Generalization (first part)
- Part III (in Lect. 4)
  - Generalization and Validation

Aim: *overview and terminology*
before starting to study models and learning algorithms

For information

Alessio Micheli

micheli@di.unipi.it

http://ciml.di.unipi.it

Dipartimento di Informatica
Università di Pisa - Italy

Computational Intelligence & Machine Learning Group