Daniele Fusi

Application of neuronal recognition to digital editions

Digital editions and specialized applications

DIGITAL EDITION SYSTEM

- monographic corpora:
  - Sydney: digital corpus of Classical Greek Theatre (inscriptions, literary passages, archaeological data)
  - Padoa: heroes (inscriptions, literary passages)
- Greek and Latin, ancient and medieval inscriptions collections

SPECIALIZED APPLICATIONS

- edition as a base for specialized applications, which in turn add new data to it:
  - linguistical applications (language evolution, morphological inflection engine)
  - automated full prosodical and metrical analysis
  - paleographical applications

Epigraphical texts and images

- digital drawings
- text / image synchronized reading
- online virtual paleographical measurements

www.fusisoft.it

retouched digital photo

digital drawing

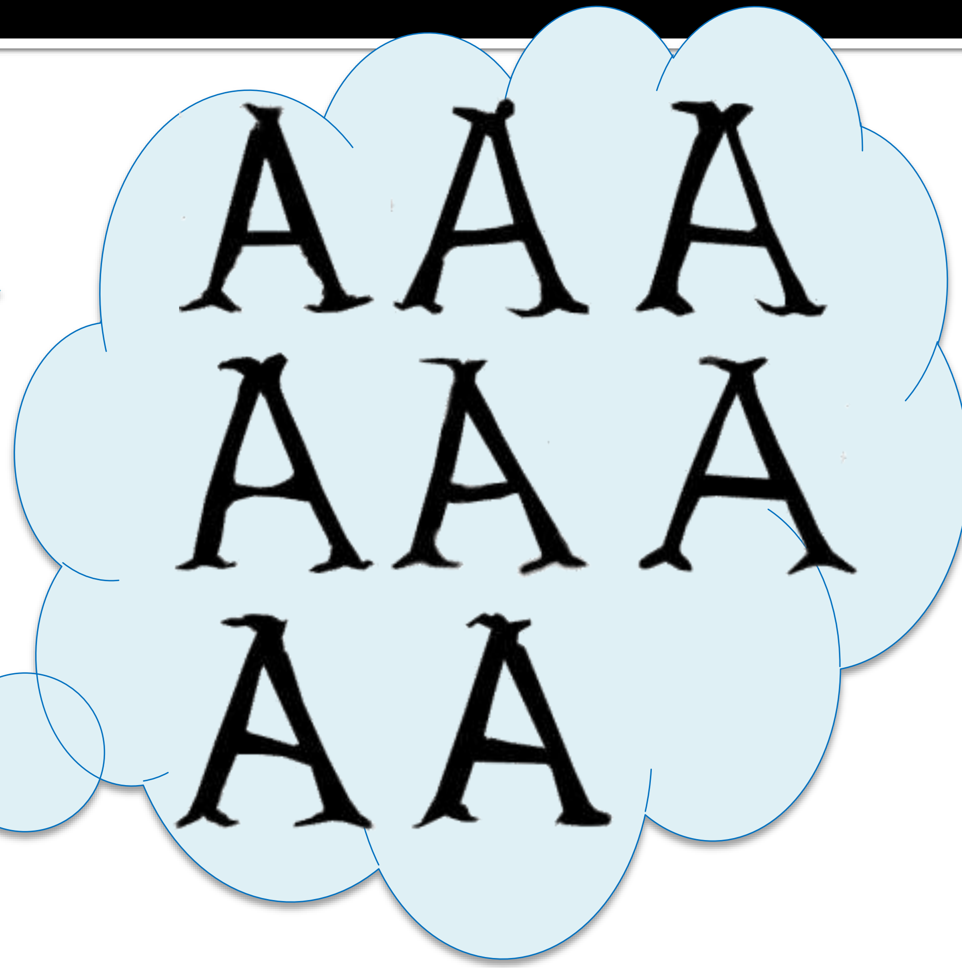
extracted drawing

Paleographical specializations

extract all the letter shapes from each drawing, getting a full digital paleographical catalog

Recognizing letter shapes

corpus, shapes
of letter "A" in III A.D.



shape never found
before by system

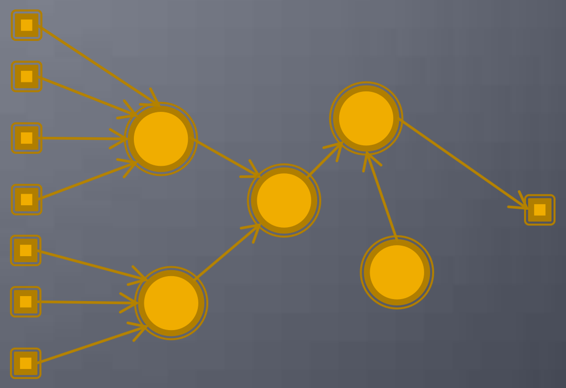
paleographical
indication: most
similar to III A.D.
"A" 's

A

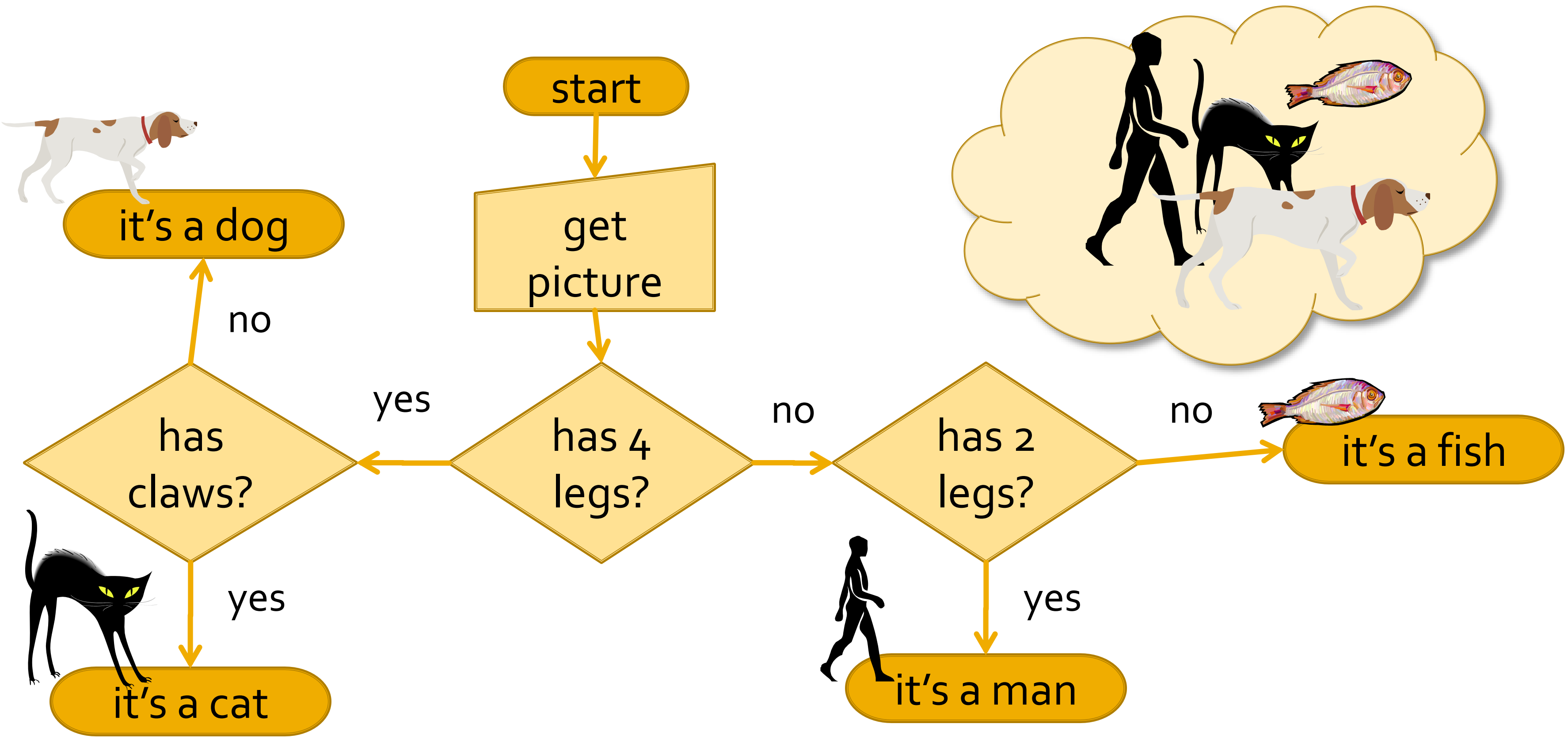
enable the system to automatically recognize letter shapes belonging to different regions or times: paleographical query

Artificial Neuronal Networks

Essential Overview



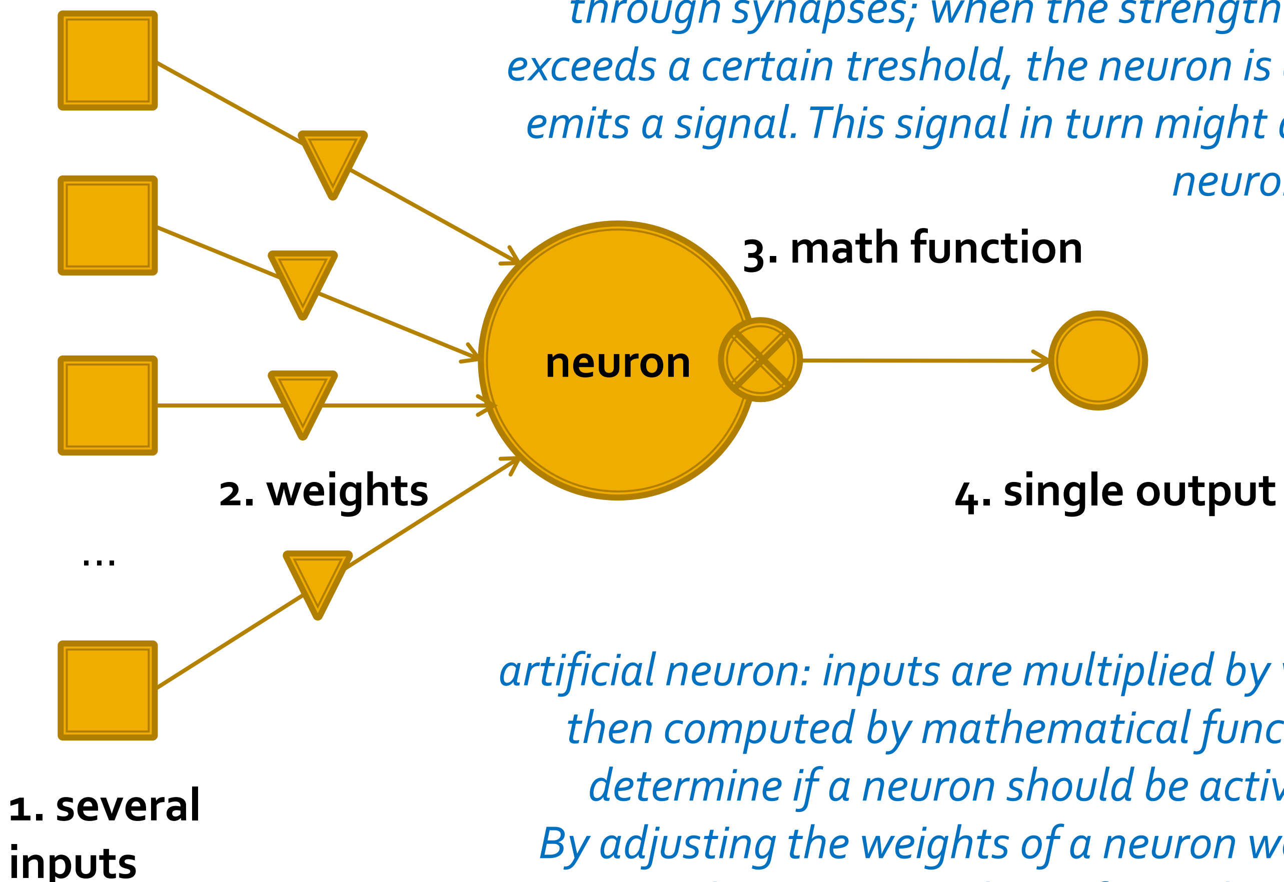
Algorithmic approach



algorithmic approach for solving problems: there is a sequence of instructions to follow, and this of course implies that we must know them in advance

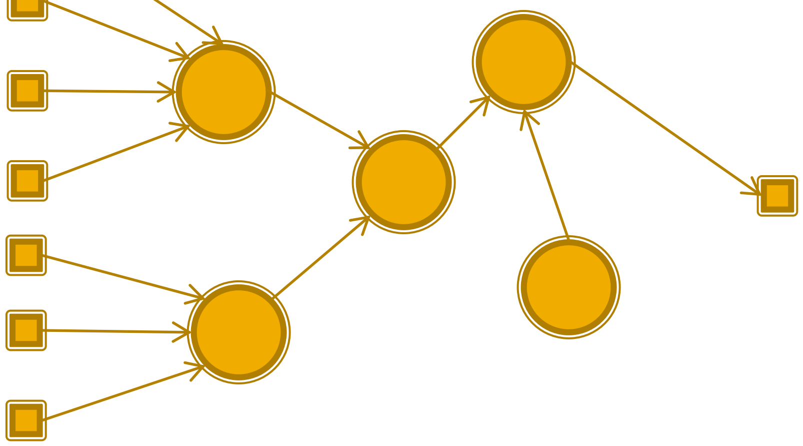
Artificial neuron

inspired by natural neurons, which receive signals through synapses; when the strength of the signals exceeds a certain treshold, the neuron is activated and emits a signal. This signal in turn might activate other neurons, and so on.



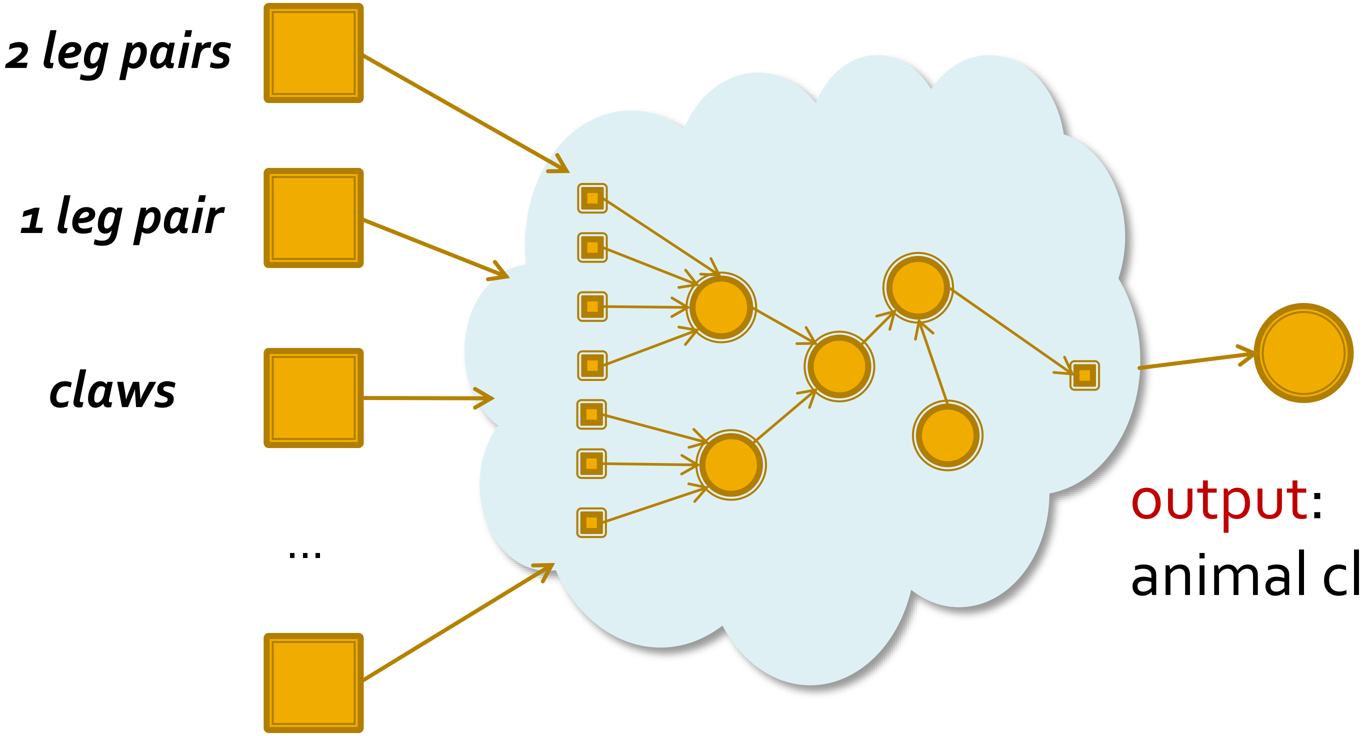
*artificial neuron: inputs are multiplied by weights and then computed by mathematical functions, which determine if a neuron should be activated or not. By adjusting the weights of a neuron we can obtain the output we desire for each specific input: the neuron **associates an input with an output**.*

Artificial neurons network



- set of several neurons, variously intercon-nected
- the interaction of neurons through their connections defines an emergent behaviour for the network
- network abilities supercede single neurons abilities

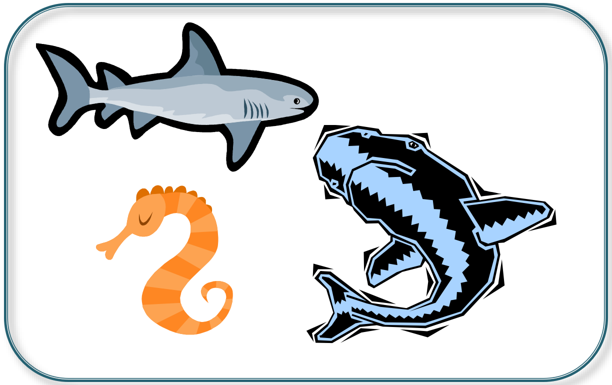
Artificial Neural Network: sample



output:
animal classification

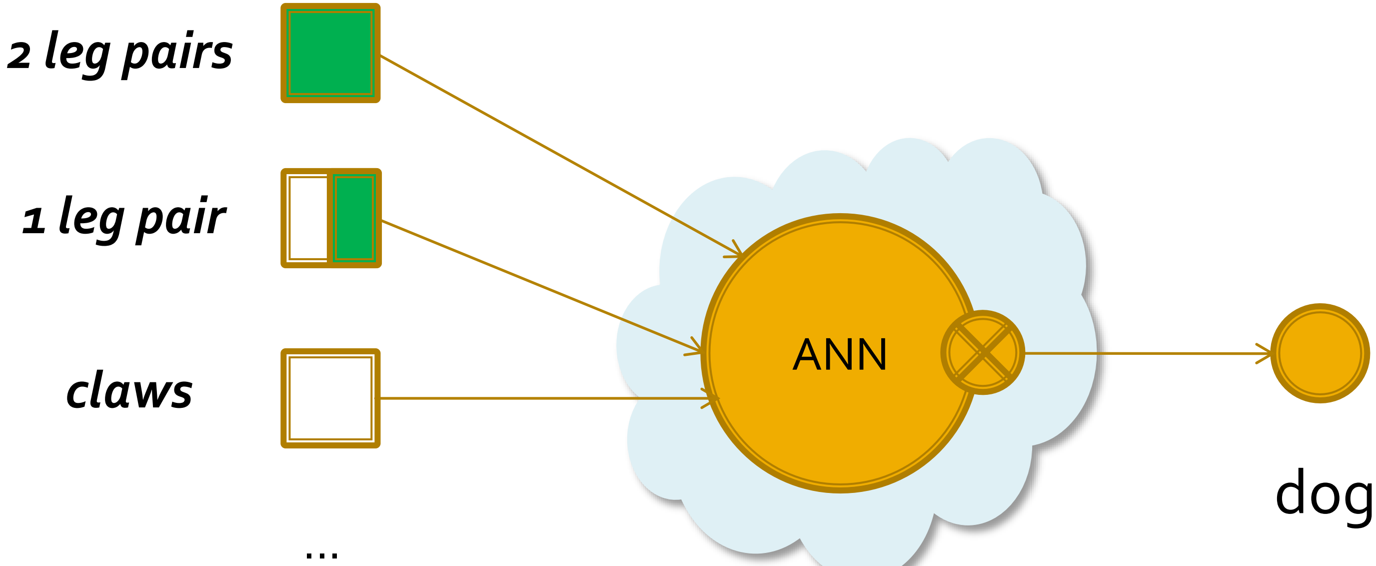
several inputs:
animals features

Learning by samples: training

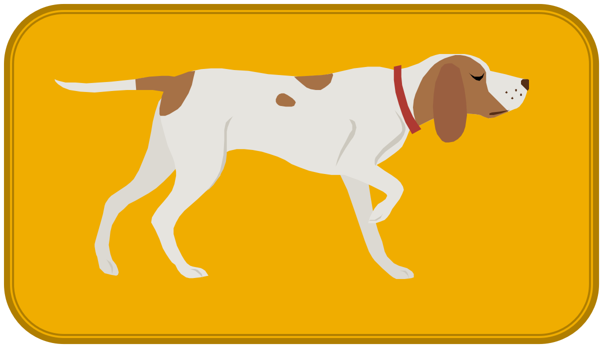


- ANN training:
  - present samples for each class
  - ANN learns to associate the features of each sample to its class

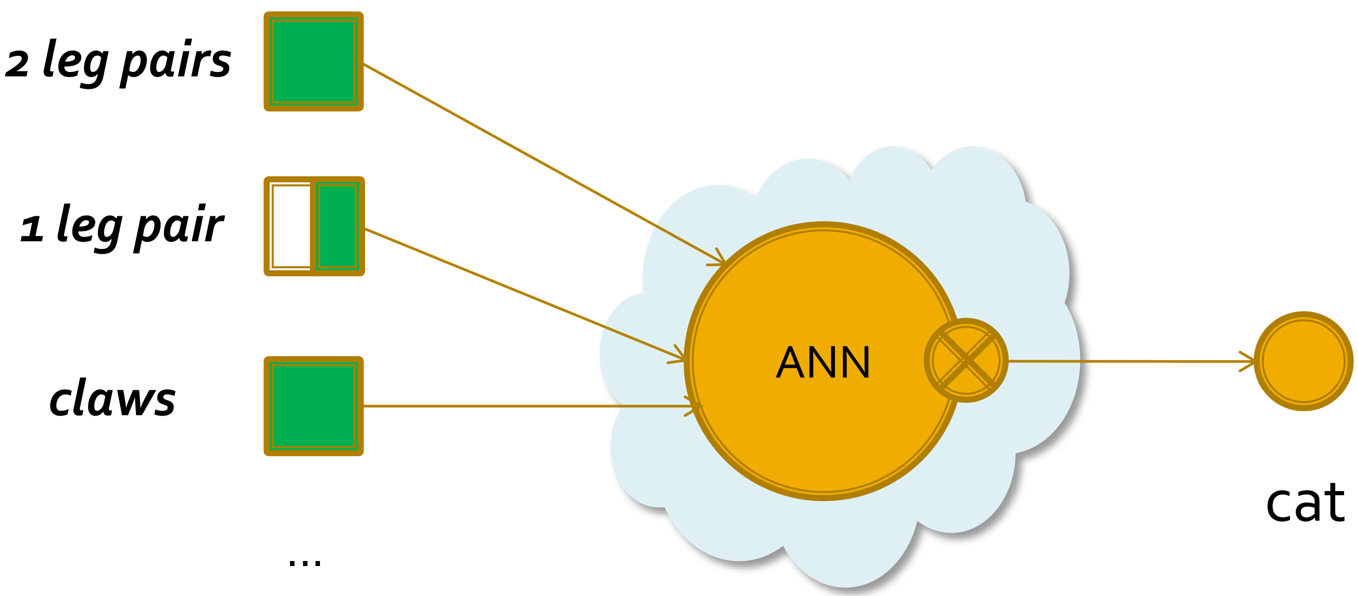
ANN: training mode



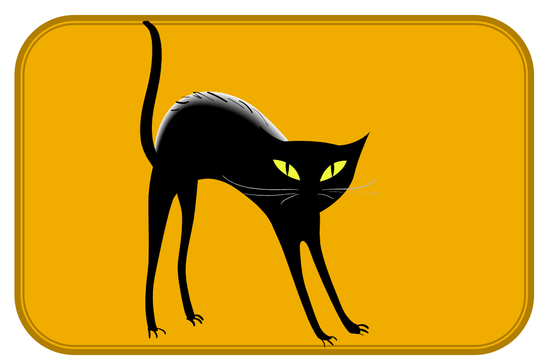
these are samples for the class 'dog'



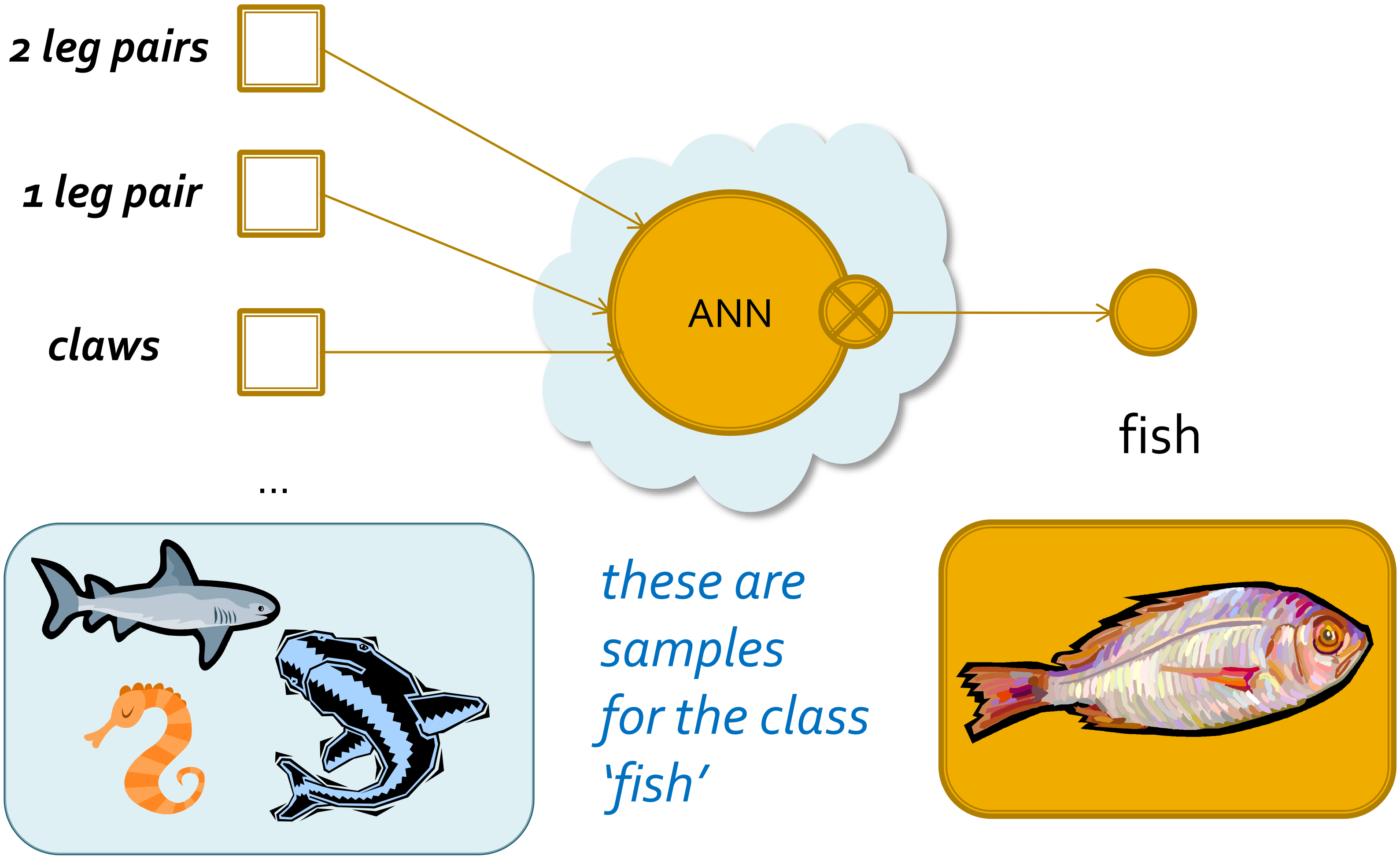
ANN: training mode



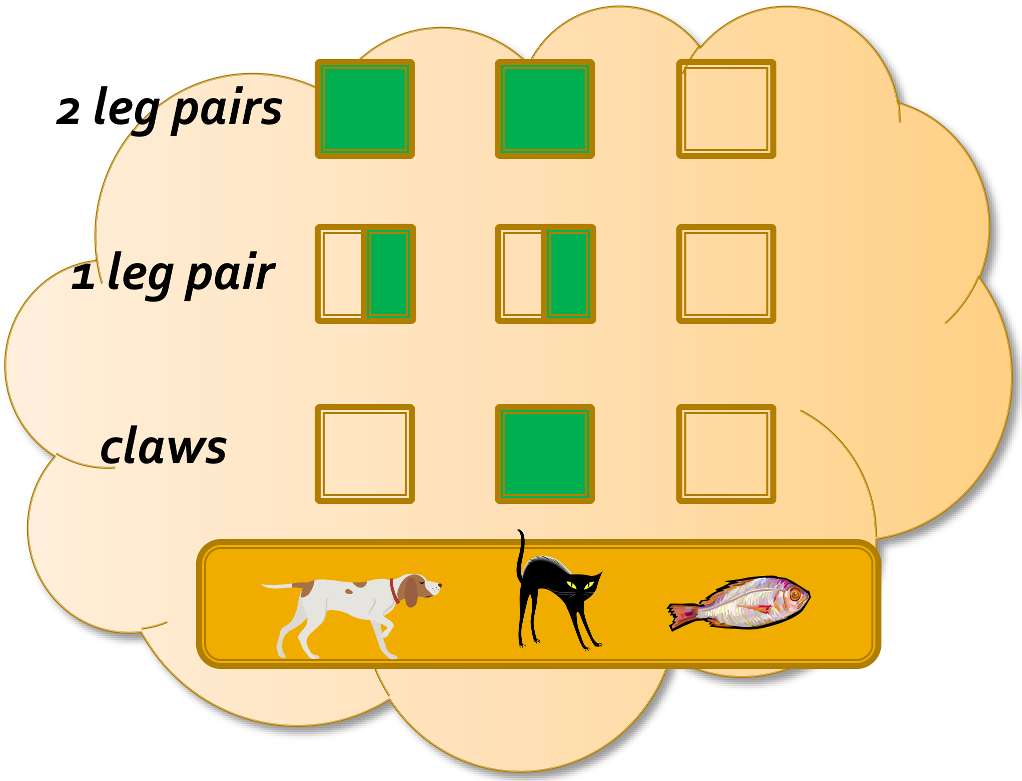
these are samples for the class 'cat'



ANN: training mode



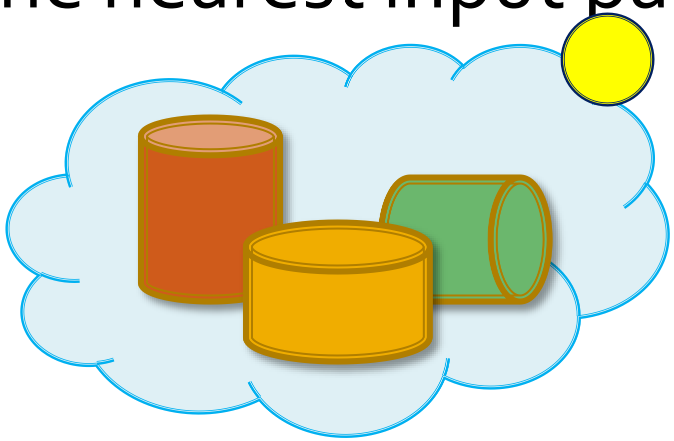
ANN: inference

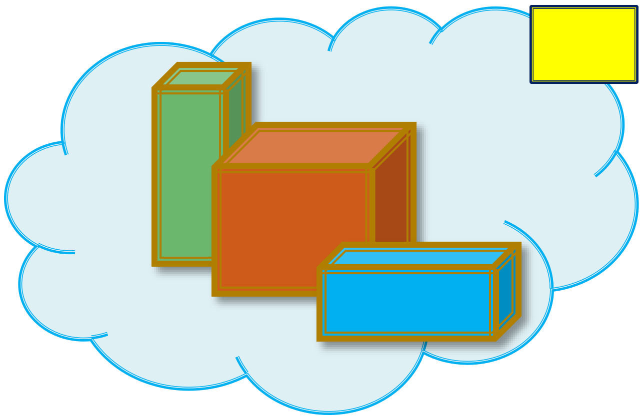


- we just feed the ANN with input samples for each class
- the ANN 'infers' the features which define each class from its samples
- samples never found before can be classified by similarity

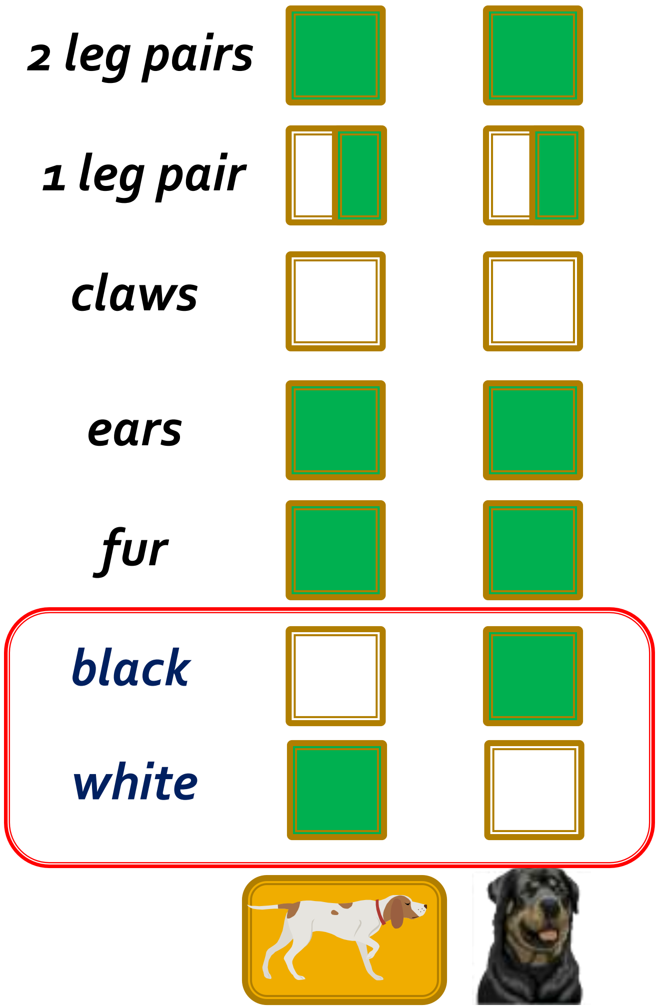
ANN: sense of similarity

- a neuron fires its signal from a **set** of input signals *(animal features)*: not all of them are required, but just enough for the neuron to fire
- the output is returned which is most similar to the nearest input pattern



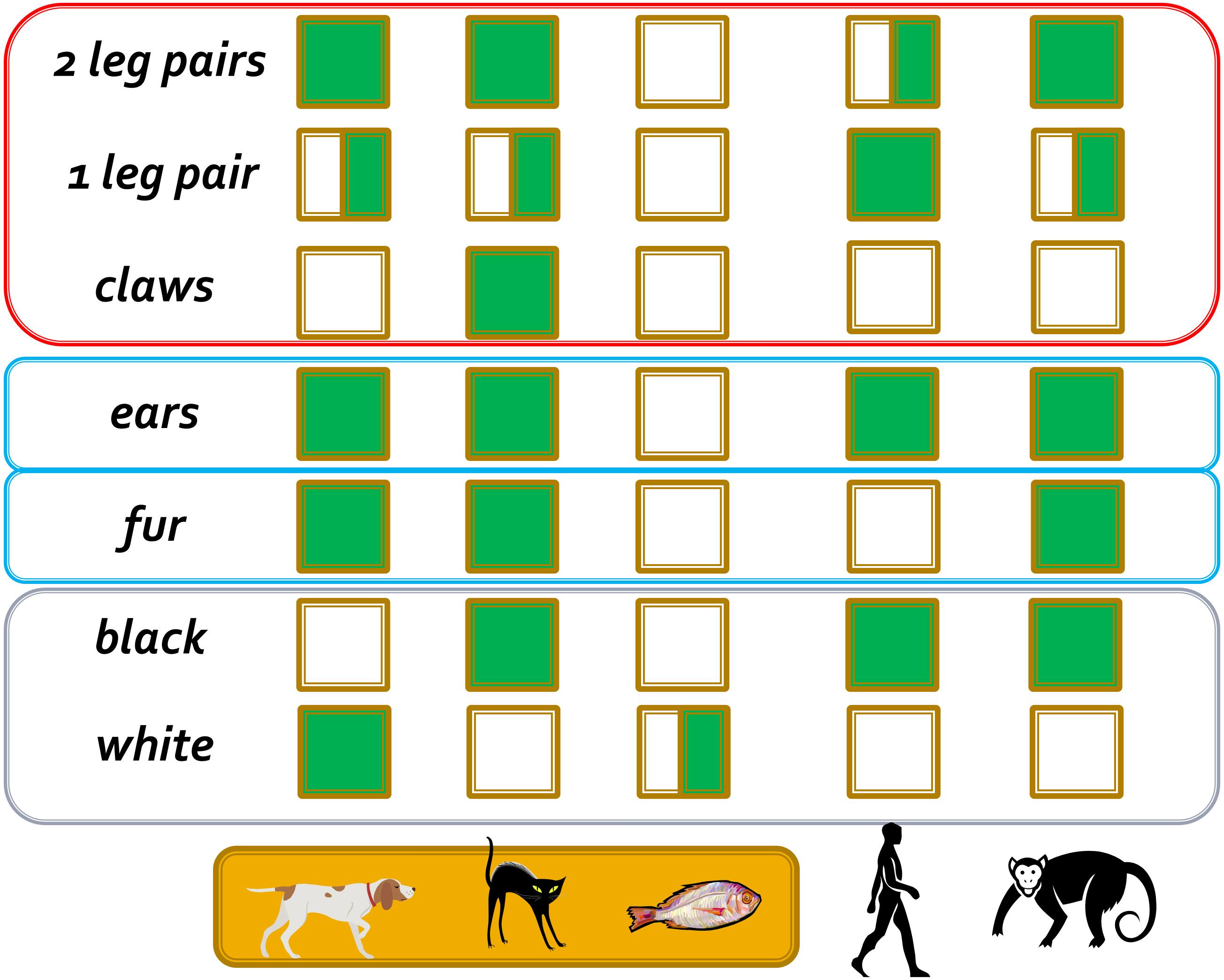


Similarity: sample



- *white dog* is the taught pattern
- *black dogs* too can be recognized as *dogs* (rather than *cats* or *fishes*), as their features set is most similar to the *dogs* class features set (except for color)

Input: weights



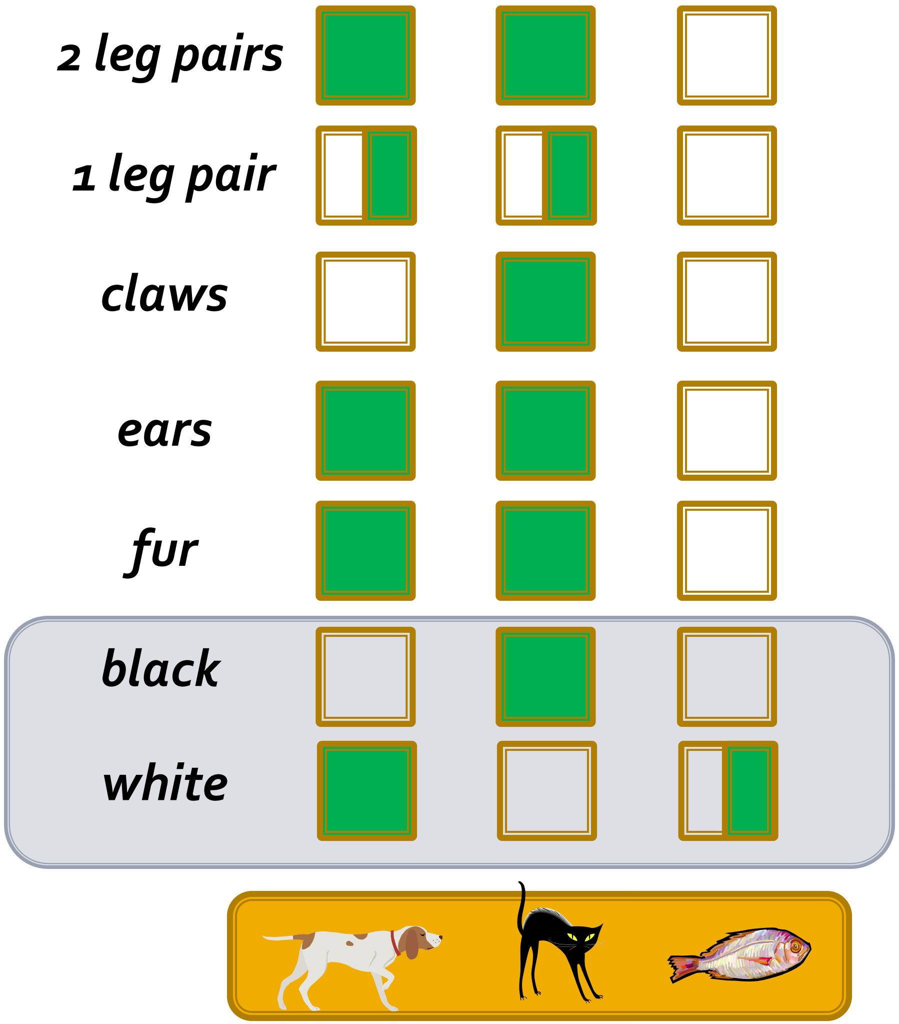
most weight

+ears if we want to distinguish just sea/non-sea animals

+fur if we want to distinguish just men from monkeys

least weight

Input: preprocessing



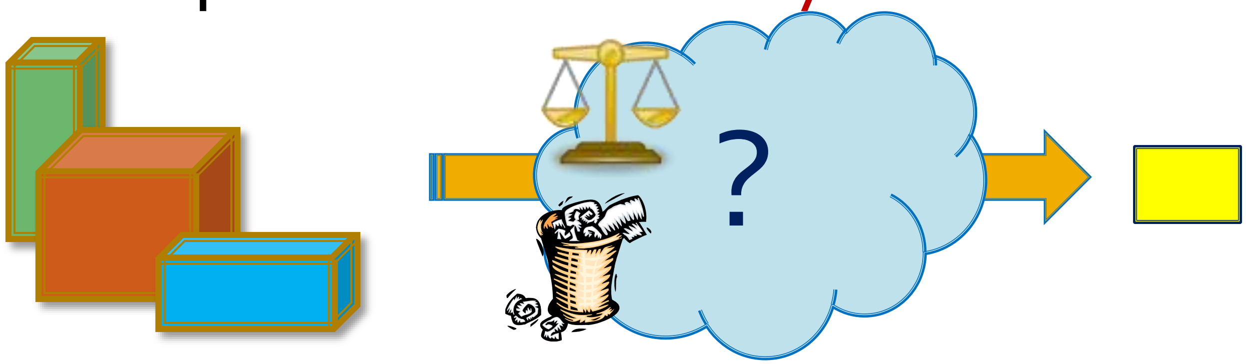
- several features in input data can be directly discarded as irrelevant ("noise", which might mislead ANN recognition): e.g. colors *(convert all animals pictures to B/W)*

ANN usages: pattern recognition

1. define a set of classes (e.g. dogs, cats, fishes)
2. each entity we want to classify is defined by a set of features
3. we *(preprocess and)* present samples of each class to the ANN, and it infers the typical features set for each class
4. once trained, any other sample presented can be classified as the most similar to any of the specified classes

Benefits of ANN in recognition

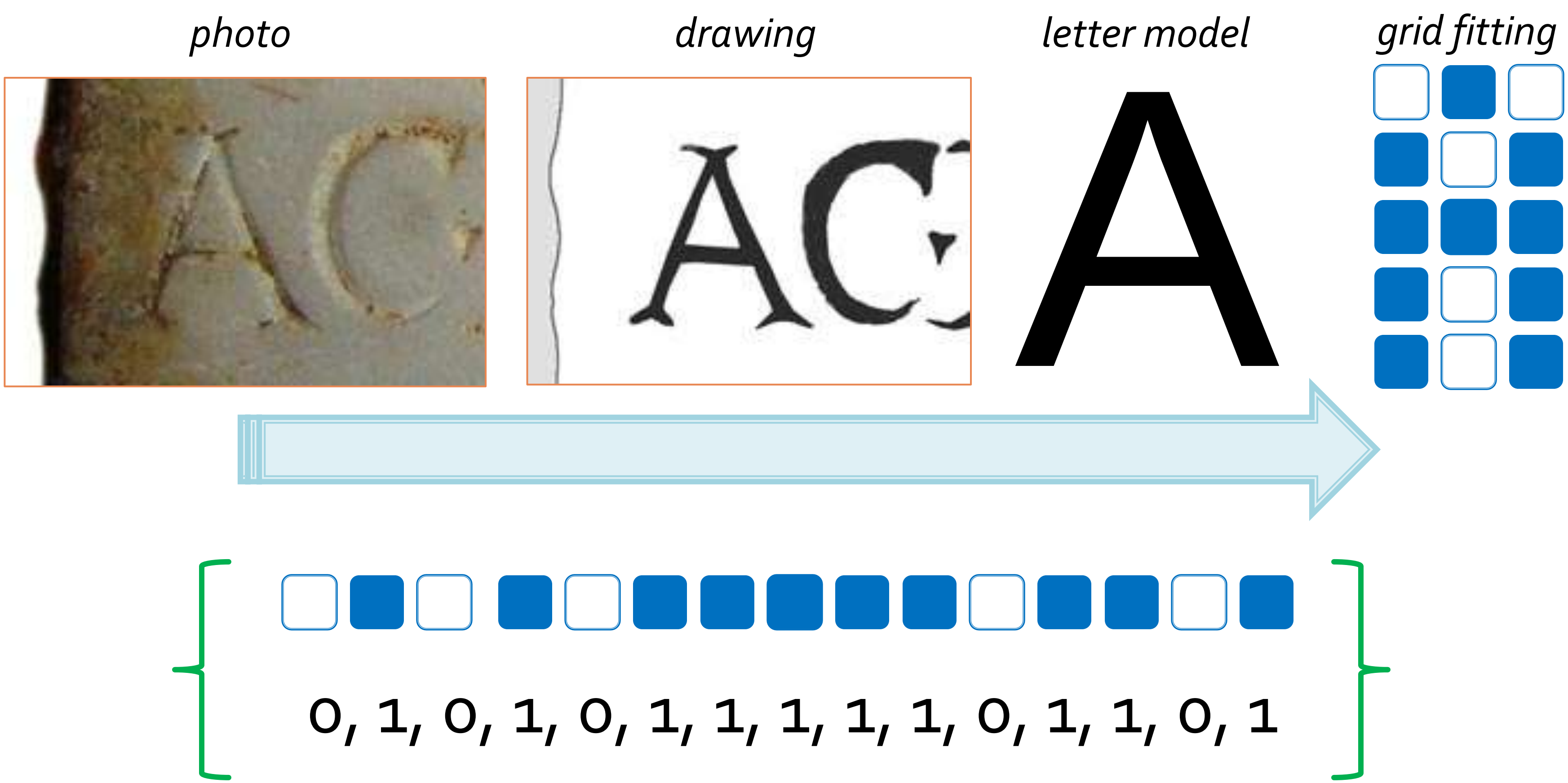
- an algorithmic approach, which might be impossible, is not required: it's the ANN which 'figures out' patterns from samples *(we tell it what we want, not how to do it)*
- ANN weights and preprocessing techniques rule out irrelevant features and allow to define patterns similarity



ANN in paleography

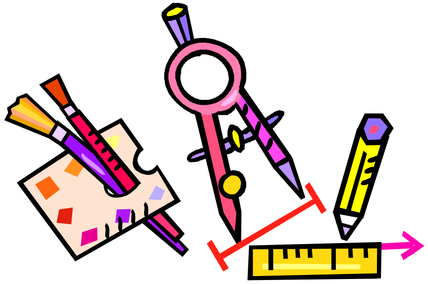
Practical issues

Defining patterns from shapes

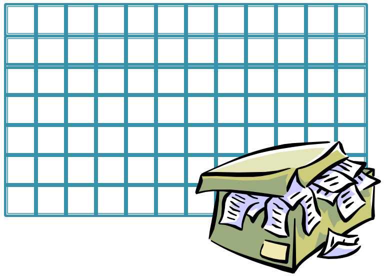


sample as a numeric vector

Practical issues



- often complex preprocessing required for letters taken from photos (colors, background, size, outlining...)

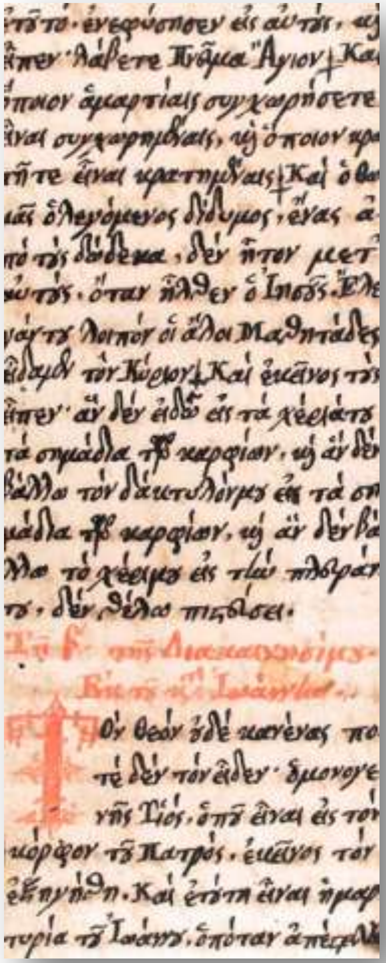


- large vectors (grids) to account for shapes details, whence memory and performance issues

- lack of very high number of samples required by this model

Paleography: additional issues

- manuscripts add complexity (ligatures) and classes (letters groups or words as single classes)
- high number of classes for finer nuances (different shapes of the same letter, not only different letters)
- weighting input: similarity treshold must be:
  - *low* to allow variant forms be classified as same letter
  - *high* to allow them to be classified as paleographically different classes



Practical solutions: preprocessing

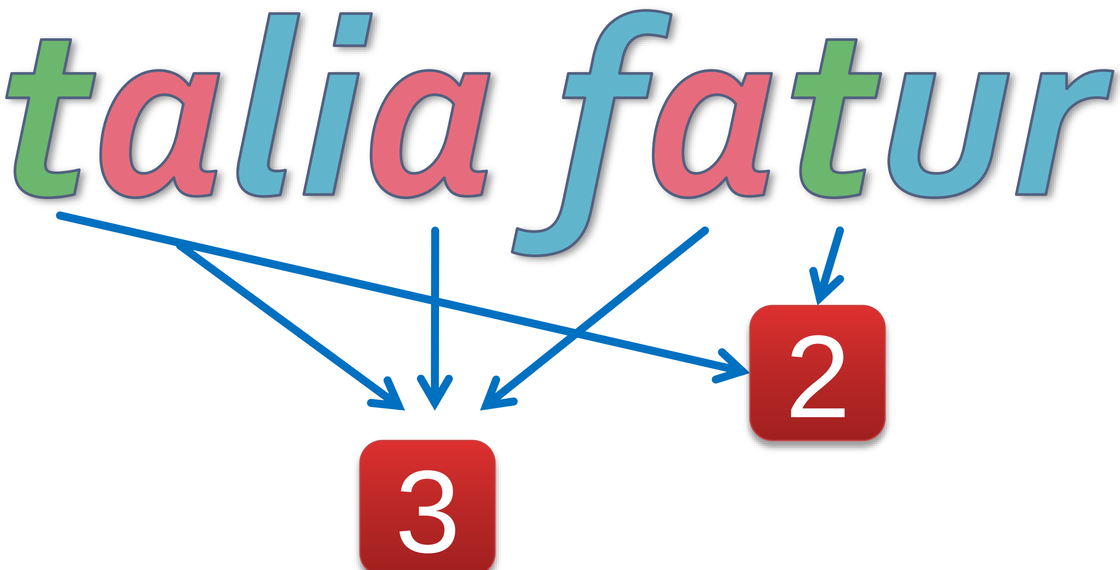
- graphical preprocessing, coupled with:
- theorical framework capable of defining relevant (most weighting) features for each shape: we cannot expect to just feed an ANN with a bunch of shapes and let it recognize letter variants. Data quality matters.
  ANN is no magic!

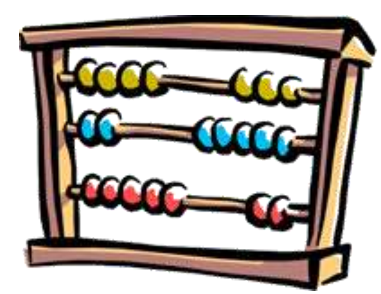
Sample: a textual analogy

Detecting alliteration in Classical texts

Case study: alliteration

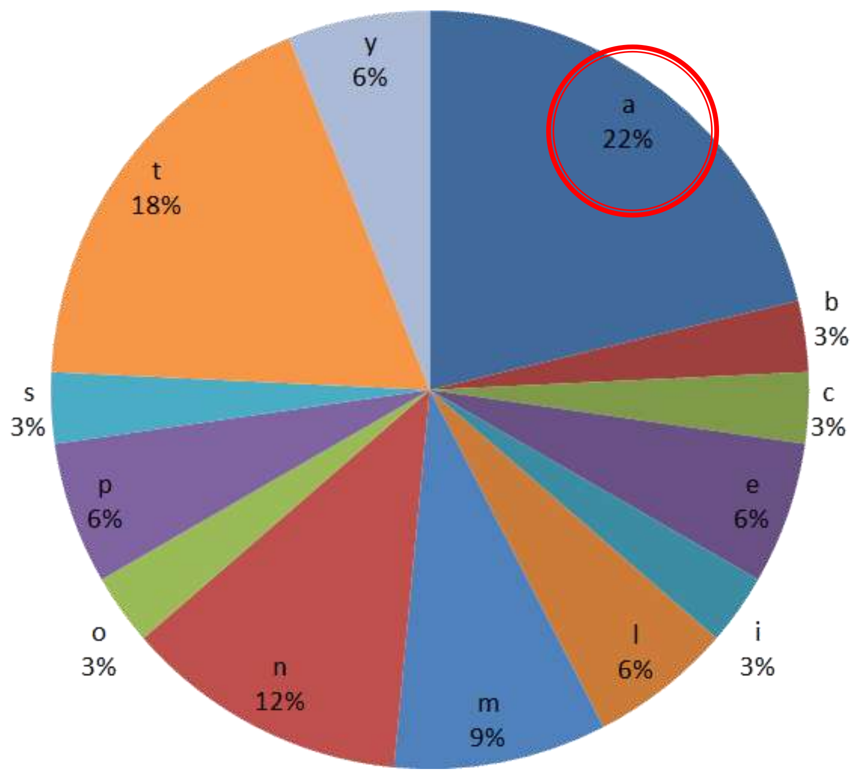
- “repetition of sounds”: naive solution: just counting letters (Evans)





Alliteration: counting letters

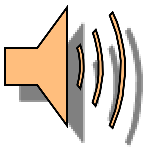
tympana tenta tonant palmis et cymbala / circum concava (Lucr.2,618-9)



- «t» does not clearly stand out
- vowels look overestimated
- all the letters have the same weight
- most slices are just noise

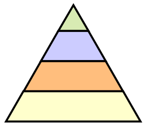
Theorical framework

- phonemic analysis (at least approximate): ears, not eyes (preprocessing)



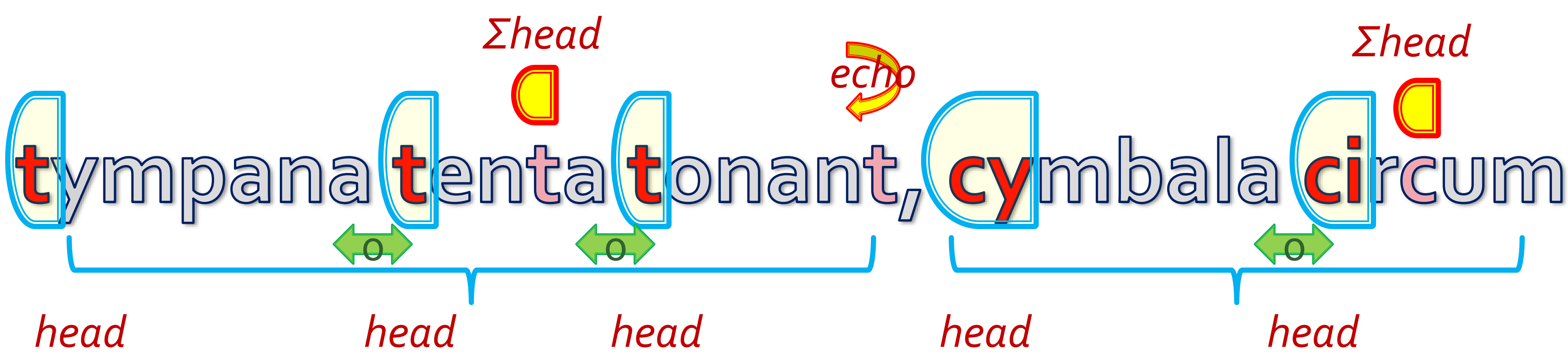
- mainly word-beginning sounds (IE inflectional languages endings, poetic tradition, diachronic phenomena…)

- hierarchy: word, syllable, phoneme, in sentence (or line) scope



Algorithm definition

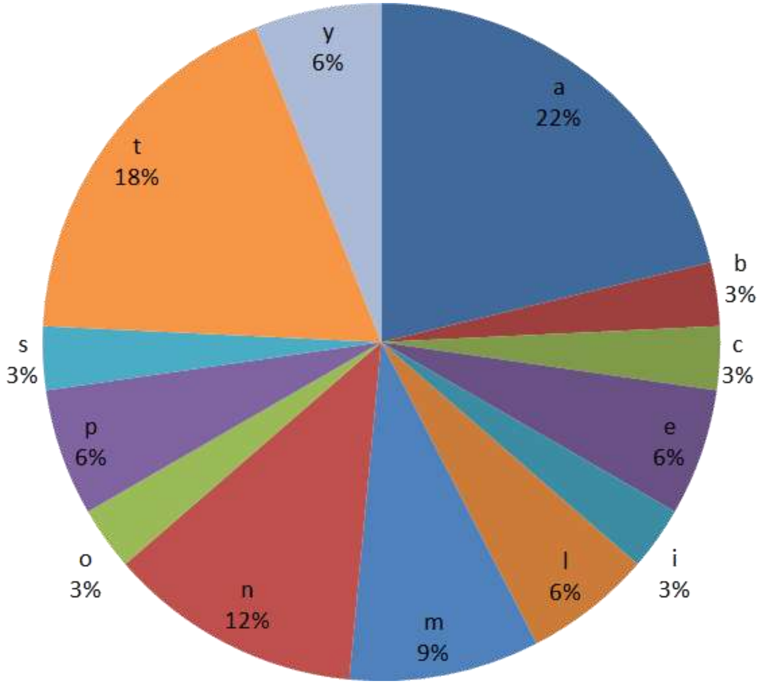
1. find all the words beginning with same sounds
2. sequence treshold: min.number of adjacent words
3. for each sequence, count the shared sounds and get their distance (word heads)
4. repeat for syllables in sequence words (syllabic heads)
5. also examine segments echoing the head sound in the remaining syllables portions (segmental echoes)



Processing for alliteration detection

JUST COUNTING LETTERS

- count letters in text



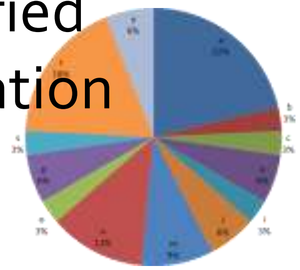
TEXT PROCESSING

- sounds, not letters
- scope: sentence / line
- levels hierarchy from word to syllable up to phoneme
- word beginning sounds are paramount
- relative distance and extent of equal word portions

Analogies with ANN approach

ALLITERATION IN TEXT

- naive approach: just take a bunch of characters and swallow it (letters counts) with no preliminary analysis or theorical grounds
- too much noise, generated from an oversimplified definition of alliteration



SHAPES RECOGNITION

- naive approach: just take letters images and feed an ANN with them
- too much noise: our **theory** must tell us how to adjust weighting and which preprocessing is required to rule out distracting features; or recognition quality will be poor, just as alliteration does not appear from a letters chart

Preprocessing and purposes

ALLITERATION IN TEXT

- different literary genres
- different poetical traditions
- different languages
- *adjust analysis parameters accordingly*

SHAPES RECOGNITION

- inscriptions or manuscripts
- different number of classes
- different level of graphical complexity
- different number of available samples
- *adjust ANN weights and preprocessing accordingly (paleography expertises required!)*

Practical scenarios

Solution hints

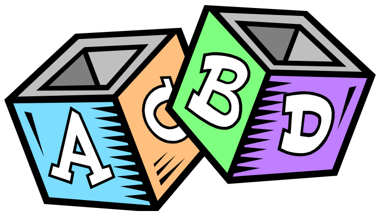
Issues: adjusting and partitioning

ANN ISSUES

- lack of samples
- very complex shapes and high number of classes
- weights must range from lowest (to recognize variants as same letter) to highest (to recognize nuances up to scribal variants)

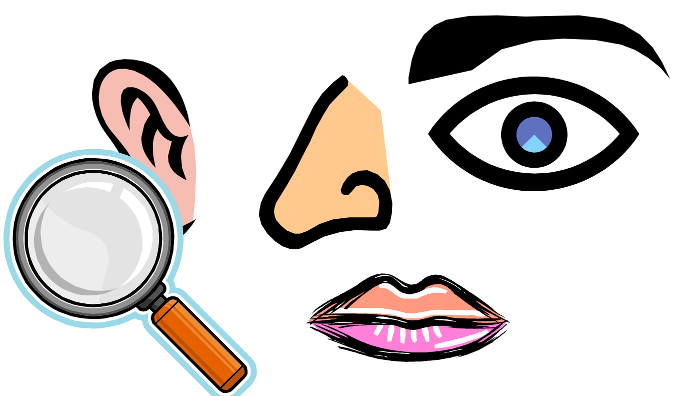
PARTITIONING PROBLEMS

- setup a theorical framework for defining the relevance of letters traits (whence weights and preprocessing)
- allow partitioning data into more abstract subsets

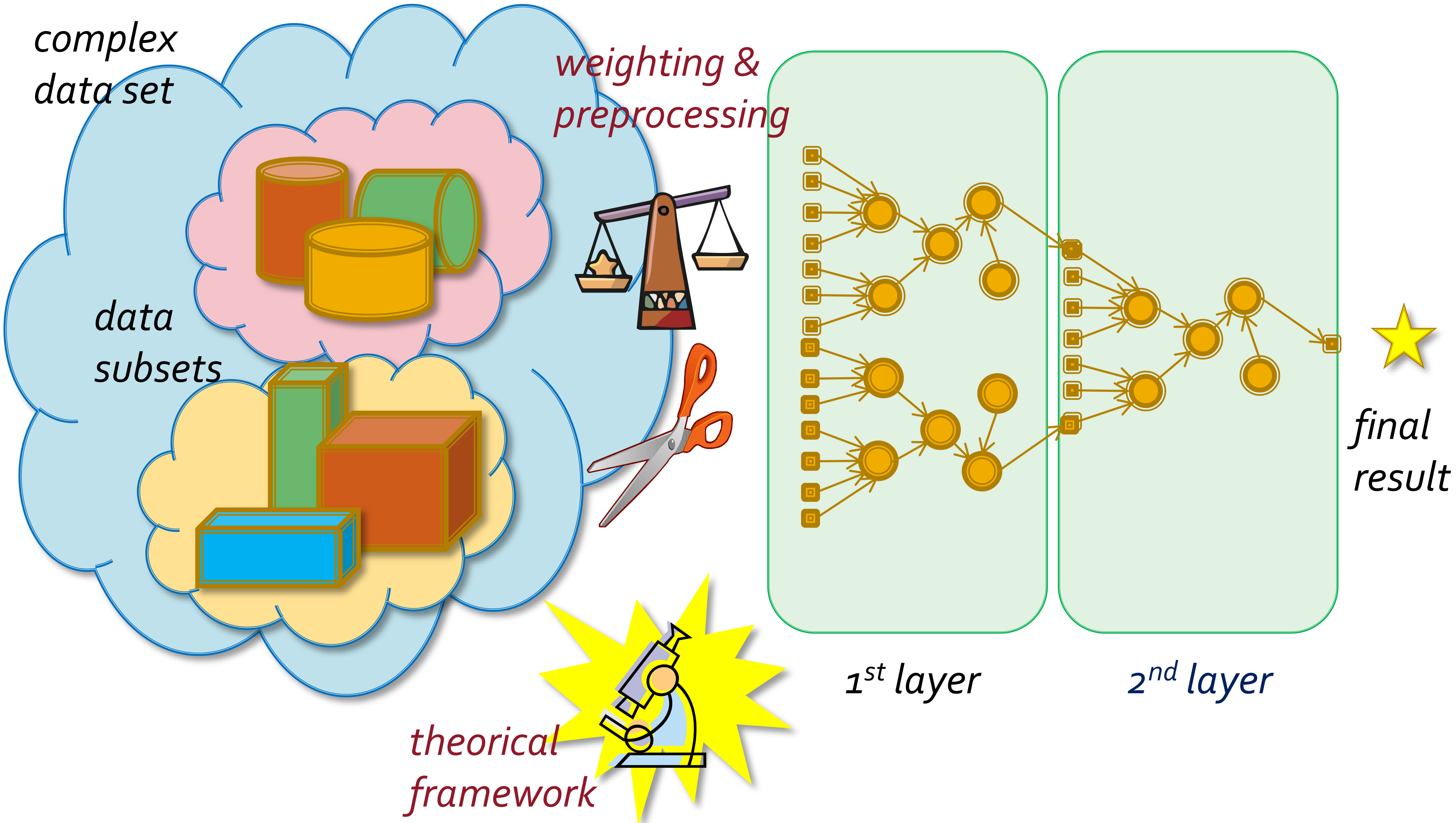


Partitioning data: analogy

IDENTIKIT	ANN RECOGNITION
an approximate face reconstruction is built of single essential blocks	input samples can be heavily preprocessed (transformed) into simpler shapes according to our theorical framework
each block is recognized (picked up) and then added to the face	several layers or networks can be trained for recognizing different traits of these shapes



Partitions and layers



Daniele Fusi

http://www.fusisoft.it

daniele.fusi@uniroma1.it