Symbolic vs. sub-symbolic computation

IA: Hard vs Soft Computing

Hard Computing

"Conscious"

Symbolic

High level

Computation as manipulation of symbols via structured rules

Rule-Based Systems (Optimiz. and search)

Logic and Constraint Programming

Ontologies

Probabilistic computation

…

Soft Computing

"Subconscious"

Sub-simbolic

Connectionism (Hebb 40's)

Computation (and mental processes) as the emergent process of interconnected networks of simple units.

Neural networks (deep learning)

Swarm intelligence

Topics and Techniques

Knowledge representation

Logic

Ontologies

Sub-symbolic

Problem solving

Search (in large spaces

Optimization

Adversarial search (games)

Constraint Satisfaction Problems

Reasoning

Inference

Planning

Logic programming PROLOG

Constraint Programming

Uncertain reasoning

- *Probabilistic reasoning*
- *Markov decision processes*
- *Bayesan networks*

Neural networks

- *Feed forward network*
- *Recurrent network*
- *Deep learning*

Learning and data analytics

- *Supervised*
- *Unsupervised*
- *Renforcement*
- *Inductive*

Topics and Techniques

Communication

Natural Language Processing

Human-machine communication

Automatic translation

Perception

Acoustic perception

Visual perception

Robotics

Sensors and attuators

Autonomous decision making

Expert systems

- *Medicine*
- *Finance*

Natural programming

- *Genetic algorithms*
- *Ant computing*

Cognitive modeling

- *Cognitive states*
- *Emotions*

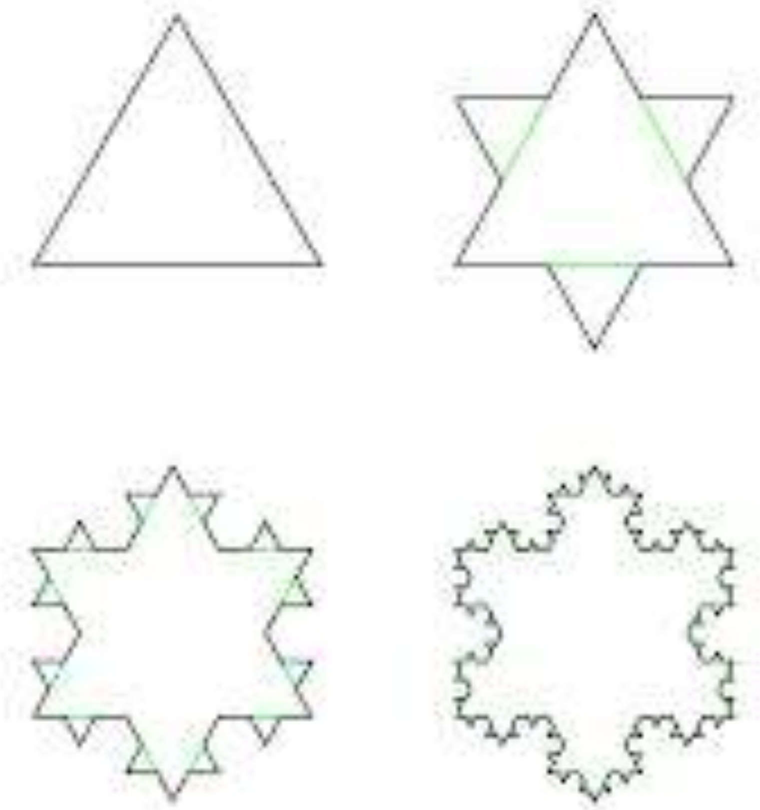
A closer look

1) Traditional Computer Science

2) Symbolic AI (expert systems, logic based reasoning)

3) Sub-symbolic AI (Machine *L*earning, *D*eep *L*earning etc.)

Traditional CS: A Fractal (Koch snowflake)



Traditional CS: (a Python program)

```
#Disegna la funzione di Koch partendo da un segmento
#di lunghezza x e fermandosi quando la lunghezza è
#minore o eguale a 10

import turtle
def koch(x) :

      if x<=10:
         turtle.forward(x)
      else:
         koch(x/3)
         turtle.left(60)
         koch(x/3)
         turtle.right(120)
         koch(x/3)
         turtle.left(60)
         koch(x/3)
```

Symbolic computation (expert systems)

I. Logic based reasoning
II. Can provide explanations for decisions
III. Not appropriate for problems such as image recognition (cannot be solved algorithmically)

More on this later

Example of rule based system: simple diagnostic problem

Goal: suggest a Drug which is appropriate for a patient on the basis of some clynical analysis
`prescribe(Drug).`

Knowledge base

Facts:

```
gram(neg).
not(allergic(antb)).
```

Rules:

`R1: gram (neg) → id (ecoli).`

If the result of the analysis is *gram-negativ* then the identity is *enterium-coli*

`R2: gram (pos) → id (strep).`

If the result of the analysis is *gram-positivo* then the identity is *streptococcus*

`R3: id(strep) OR id(bact) → ind(pen).`

If the identity is streptococcus or bacterium then it is appropriate to indicate penicillin

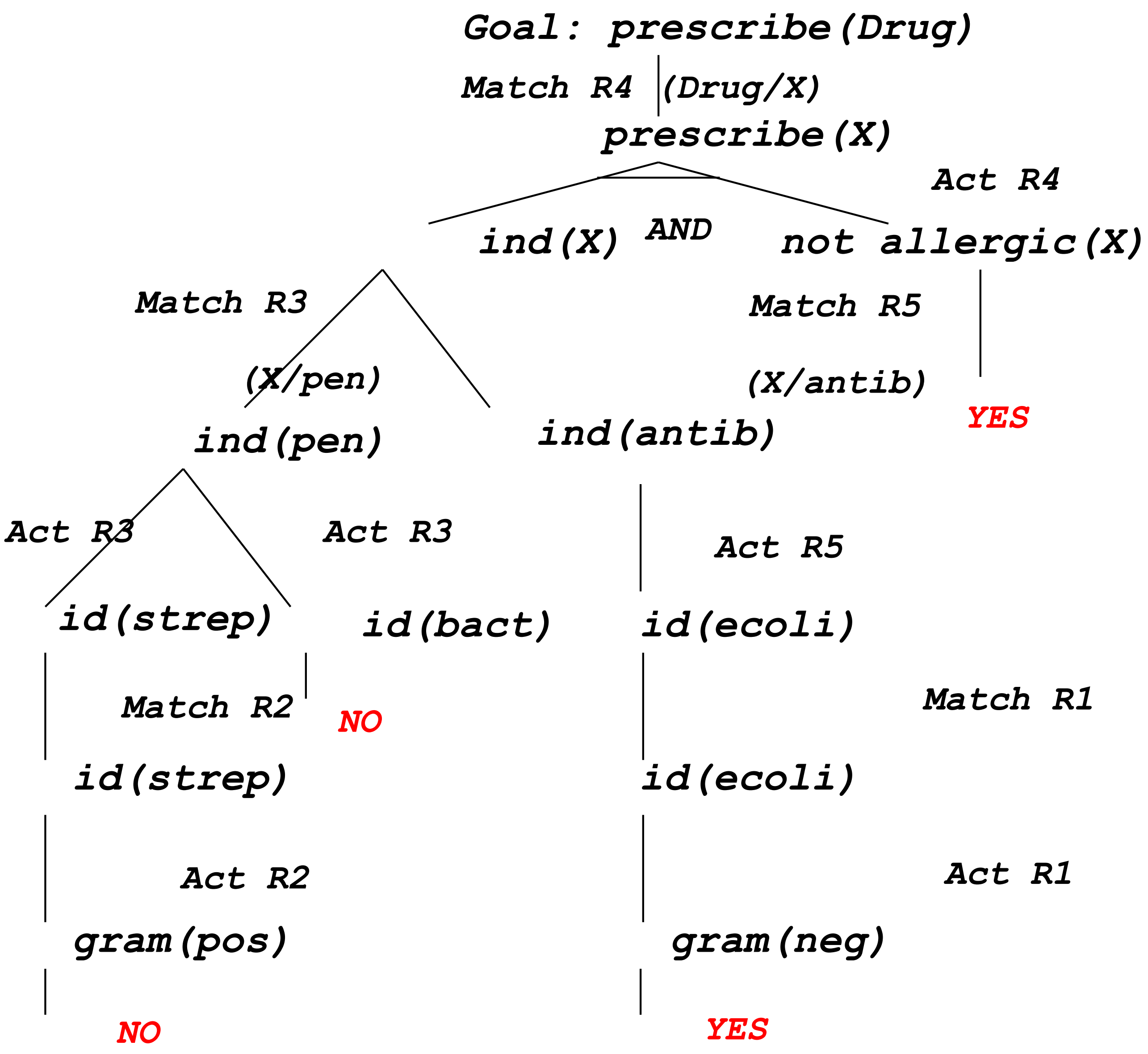
`R4: ind(X) AND not (allergic(X)) → prescribe(X)`

If it is appropriate to indicate the drug X and the patient is not allergic to X then we can prescribe X

`R5: id(ecoli) → ind(antb).`

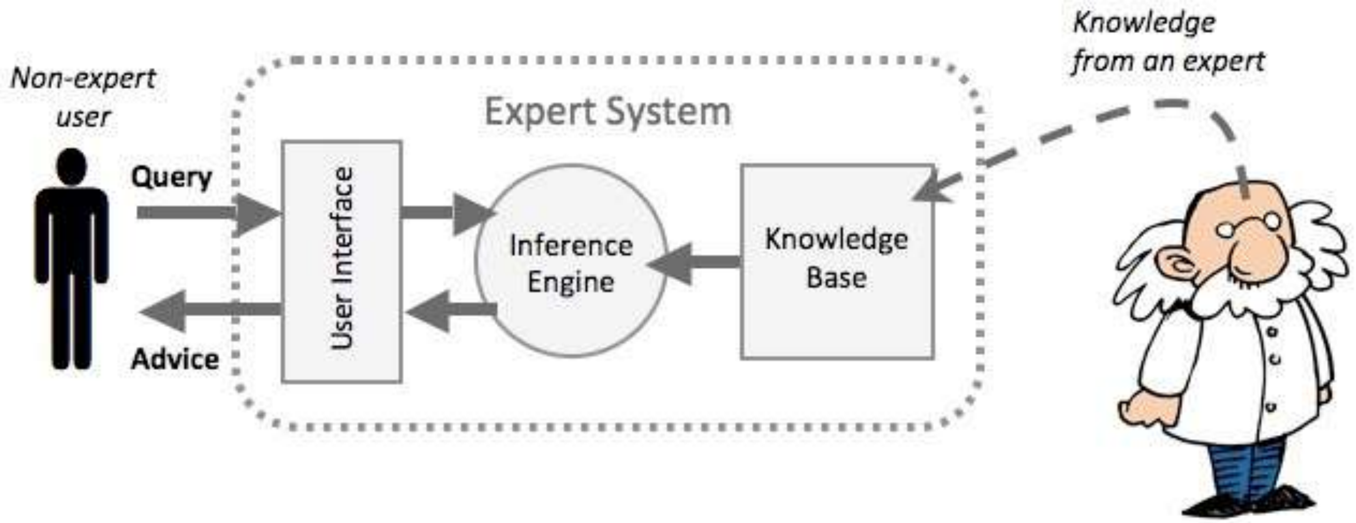
If the identity is ***enterium-coli*** then it isappropriate to indicate antibiotics.

Motore di Inferenza o Controllo (backward)



Expert systems and decision support systems (1970 -)

- Needed a domain knowledge provided by humans (typically rules expressed in logic)
- Solve problems in a limited domain
- Have performance similar to an expert of the domain
- Uses knowledge representation and search in large spaces
- Build dynamically the solution
- Search and solution generation by means of an inference engine which elaborates rules



Picture from Tan et al. The application of expert system: A revie

Logic and AI

Many different logics

Propositional, first-order, modal (epistemic, temporal, deontic) …

Many techniques for representing knowledge and reasoning

Deduction, Induction, Abduction …

Many languages and tools

PROLOG, Constraint programming

As the base of many Expert Systems

PROLOG (Kowalski 1973): logic programming

Example: sum of two integer numbers

1. sum(0,X,X). Fact
2. sum(X+1,Y,Z+1) ← sum(X,Y,Z). Rule

Meaning:

1. the sum of 0 and X is X
2. if the sum of X and Y is Z then the sum of X+1 and Y is Z+1

Many possible queries

sum(2,4,Z). Answer Z = 6
sum(2,X,5).
Answer X = 3

Deduction

Deduction allows to derive consequences of the assumed:

B can be derived from A if B is a logical consequence of A (A |= B)

Given the truth of the assumption A follows the truth of the conclusion B

In logic programming

T		B		E (deduce)
parent(X,Y) :- mother(X,Y). parent(X,Y) :- father(X,Y).	$\cup$	mother(mary,vinni). mother(mary,andre). father(carrey,vinni). father(carry,andre).	$\rightarrow$	parent(mary,vinni). parent(mary,andre). parent(carrey,vinni). parent(carrey,andre).

Induction

Inductive reasoning allows inferring B from A where B does not follow necessarily from A

The premises are viewed as supplying strong evidence for the truth of the conclusion

In some case derivation of general principles from observations: we have seen white swans we derive the rule "all swans are white"

In LP

E+

parent(mary,vinni).
parent(mary,andre).
parent(carrey,vinni).
parent(carrey,andre).

E-

parent(mary,carrey).

$\cup$

B

mother(mary,vinni).
mother(mary,andre).
father(carrey,vinni).
father(carry,andre).

$\rightarrow$

T (induce)

parent(X,Y) :- mother(X,Y).
parent(X,Y) :- father(X,Y).

$\forall e^+ \in E^+: \quad B \cup T \models e^+$ (T is complete)

$\forall e^- \in E^-: \quad B \cup T \not\models e^-$ (T is consistent)

Abduction

Abduction allows to infer A an explanation of B

In other words, the precondition A is inferred (abduced) from the consequence B.

Given a theory T and an observation O we infer (abduce) and explantion E Such that

$T \cup E \models O$

$T \cup E$ is consistent

Abduction in LP

T

parent(X,Y) :- mother(X,Y).

parent(X,Y) :- father(X,Y).

$\cup$

E

parent(mary,vinni).

parent(mary,andre).

parent(carrey,vinni).

parent(carrey,andre).

$\rightarrow$

B (abduce)

mother(mary,vinni) $\vee$ father(mary, vinni).
mother(mary,andre) $\vee$ father(mary,andre).
mother(carrey,vinni) $\vee$ father(carrey,vinni).
mother(carey,andre) $\vee$ father(carry,andre).

Spesso si usano anche “vincoli di integrità” per controllare la generazione di ipotesi.

$T \cup B \models E$

T

parent(X,Y) :- mother(X,Y).

parent(X,Y) :- father(X,Y).

mother(X,Y) $\rightarrow$ female(X).

father(X,Y) $\rightarrow$ male(X).

female(mary).

male(carrey).

IC

$\cup$

E

parent(mary,vinni).

parent(mary,andre).

parent(carrey,vinni).

parent(carrey,andre).

$\rightarrow$

B (abduce)

mother(mary,vinni).
mother(mary,andre).
father(carrey,vinni).
father(carey,andre).

$T \cup B \models E$

$T \cup B \models IC$

Summarizing: Symbolic AI systems

1. Use logic based reasoning
   a. Many techniques for representing knowledge and reasoning (deduction, abduction, induction …)
   b. Many languages and tools (Lisp, Prolog, MinZinc …)
   c. Many concrete problems can be solved

2. Can create new knoweldge: Inductive learning in logic.
   a. To find a hypothesis, expressed in logical terms, that classifies the examples well and generalizes well to new examples.
   b. Inductive Logic Programming : new knowledge about the functional genomics of yeast (2009)

3. Robust systems, can provide explanations for decisions

4. Very sensitive to syntax (of course)

5. Not appropriate for problems that cannot be solved algorithmically

Sub-symbolic computation

I. Based on (artificial) neural networks and ML

II. Very good for image recognition and many problems which cannot be solved algorithmically

III. Cannot provide explanations for decisions

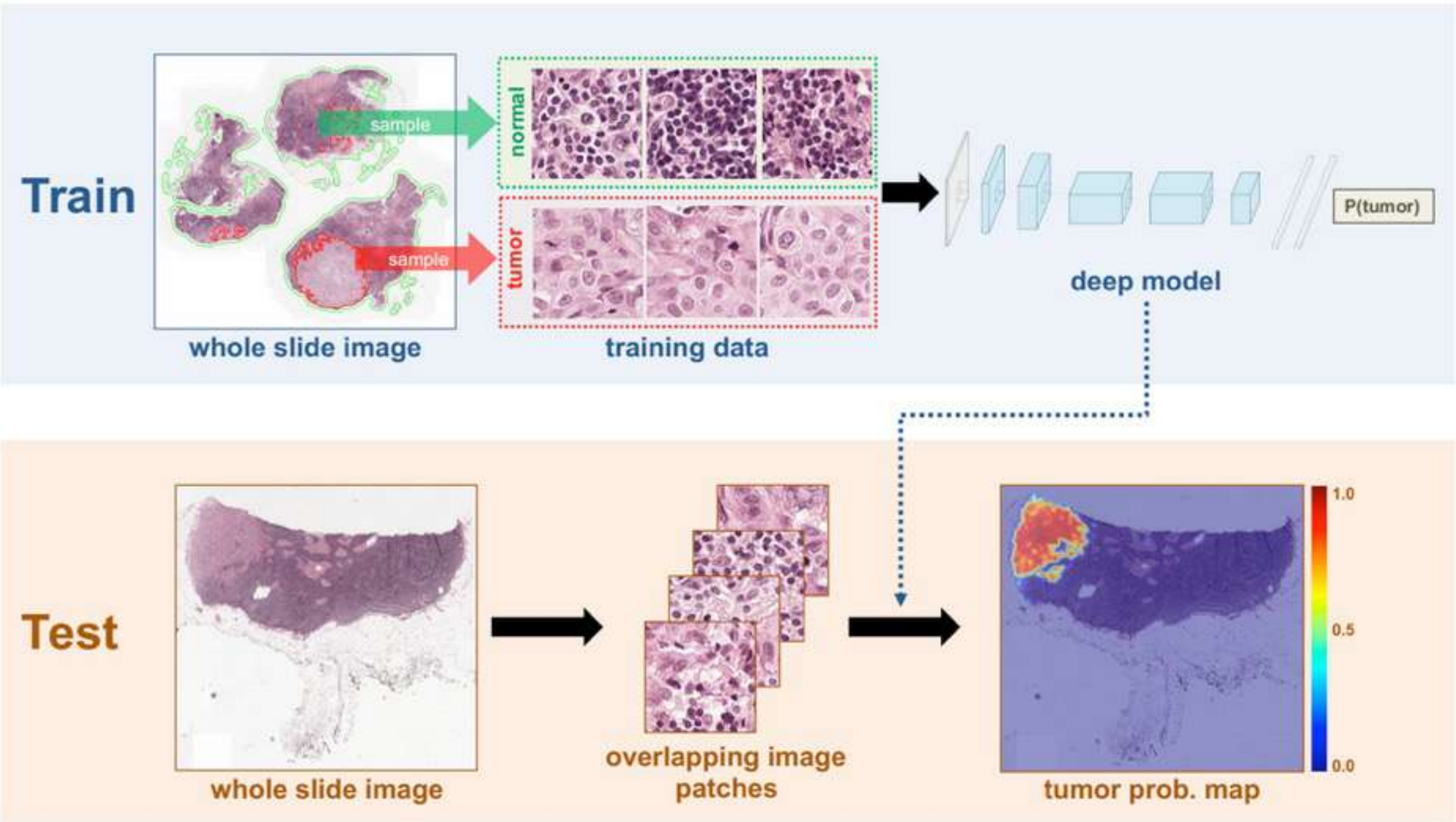
Two examples:

Google DeepMind's Deep Q-learning playing Atari Breakout (please see you tube)

Yolo v2please see you tube, e.g.

(https://www.youtube.com/watch?v=VOC3huqHrss

Sub-symbolic computation



Neural networks

Inspired by the brain model. A network with:

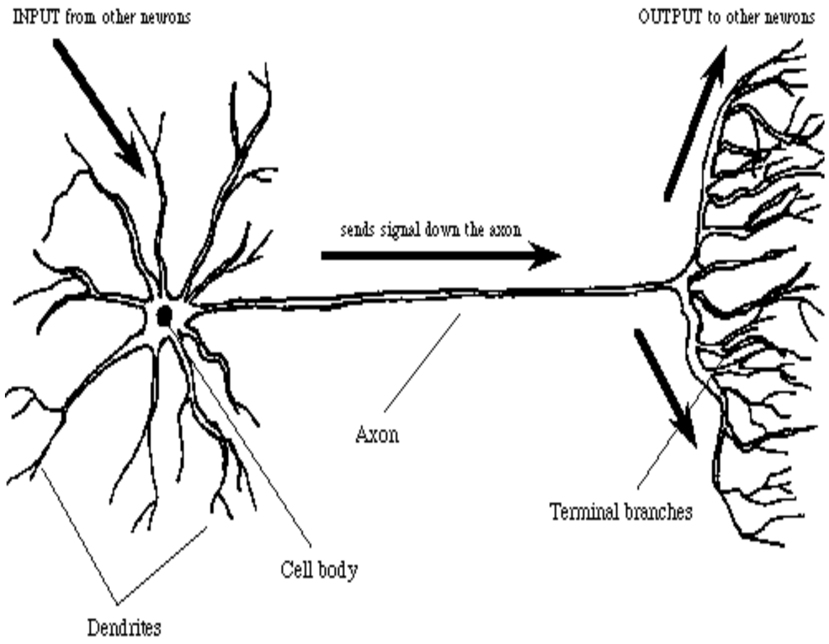
- Many computational units (neurons) with low computational power
- many (weighted) connections
- Distributed control highly parallel

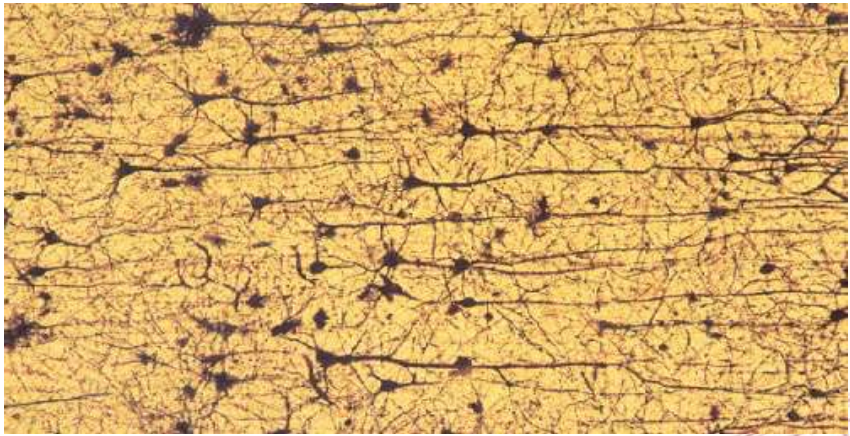
Approach completely different from the symbolic one:

- Not explicit knowledge
- Knowledge embedded int the structure of the network and the weights of the connections.

Learning ability

- Learning from differences between known target and observed facts (I/O)
- Learning mathematical functions associated to node computation





The artificial neuron

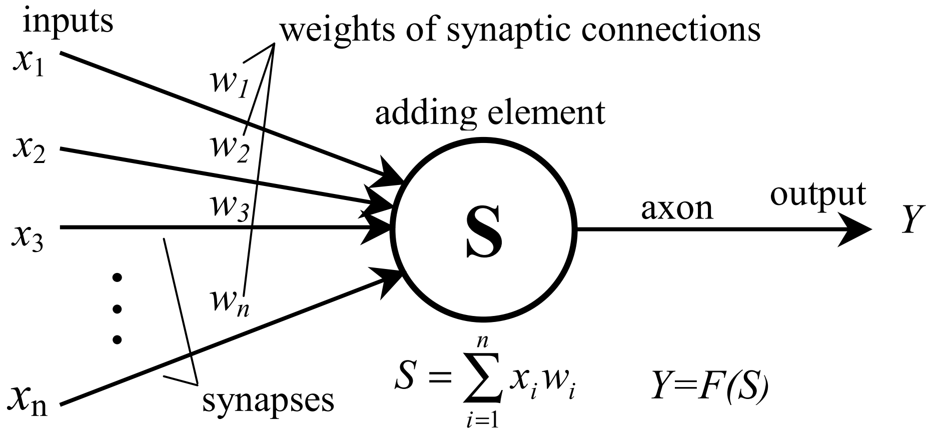


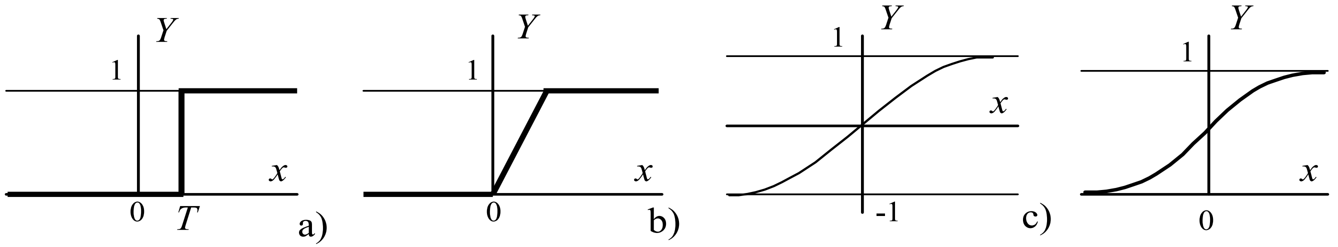
Figure 2. General structure of neuron

$$Y = F\left(\sum_{i=1}^{n} x_i \cdot w_i\right)$$

F - activation function,

the condition to activate a neuron is $\sum_{i=1}^{n} w_i x_i - \theta \geq 0$

where θ is threshold of transfer function.



Activation functions: a) jump, b) threshold, c) hyperbolic tangent, d) logistic function

The perceptron learning rule

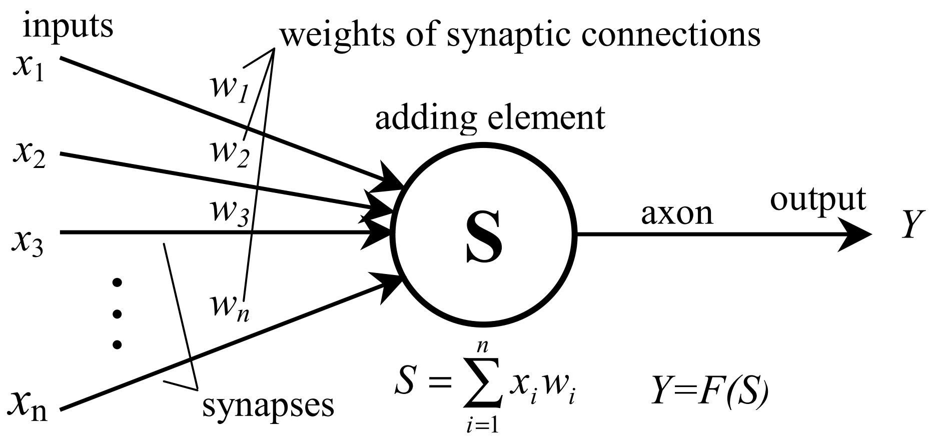


Figure 2. General structure of neuron

$$Y = F\left(\sum_{i=1}^{n} x_i \cdot w_i\right)$$

F - activation function,

the condition to activate a neuron is $\sum_{i=1}^{n} w_i x_i - \theta \geq 0$

The following rule is used for updating the weights

$$w_i \leftarrow w_i + \alpha\,(\,y - h_{\mathbf{w}}(\mathbf{x}))*x_i$$

This is essentially the update for linear regression.

Converges to a prefect linear separato if the data points are linear separable

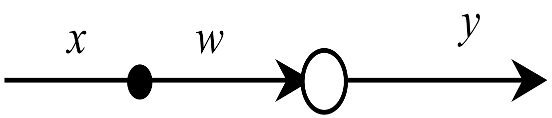
An example: representation of boolean functions

We consider elementary Boolean functions NOT, OR, AND, XOR. Table 1 shows these functions.

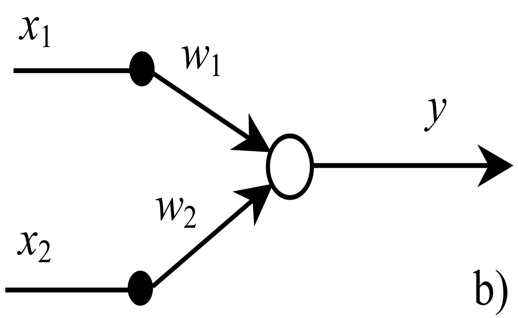
Table 1. Elementary Boolean functions

x_1	x_2	$\overline{x_1}$	$x_1 + x_2$	$x_1 \cdot x_2$	$x_1 \oplus x_2$
0	0	1	0	0	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	1	0

The basic question is, whether one neuron is enough to represent each of these Boolean functions. The neural networks to represent the first three functions are shown in figure 4.

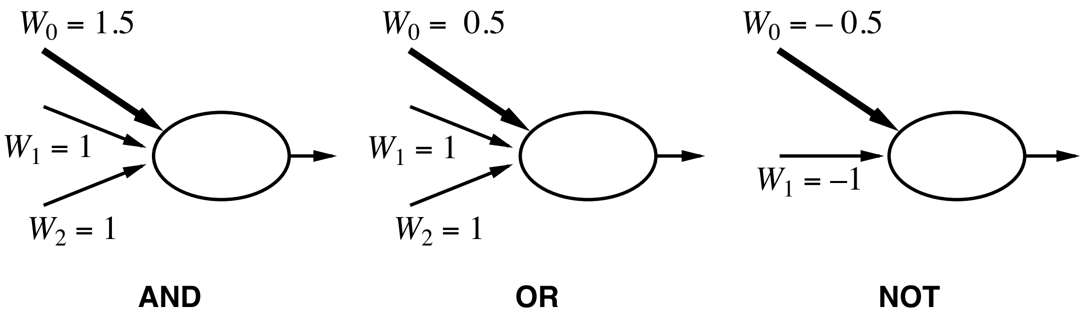


a)



b)

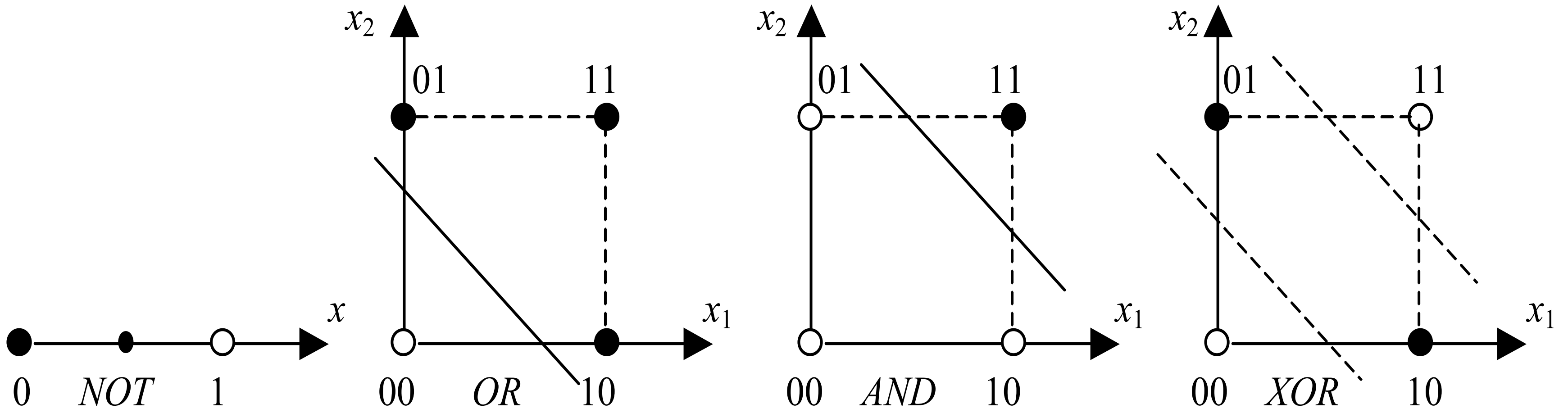
Implementing logical functions



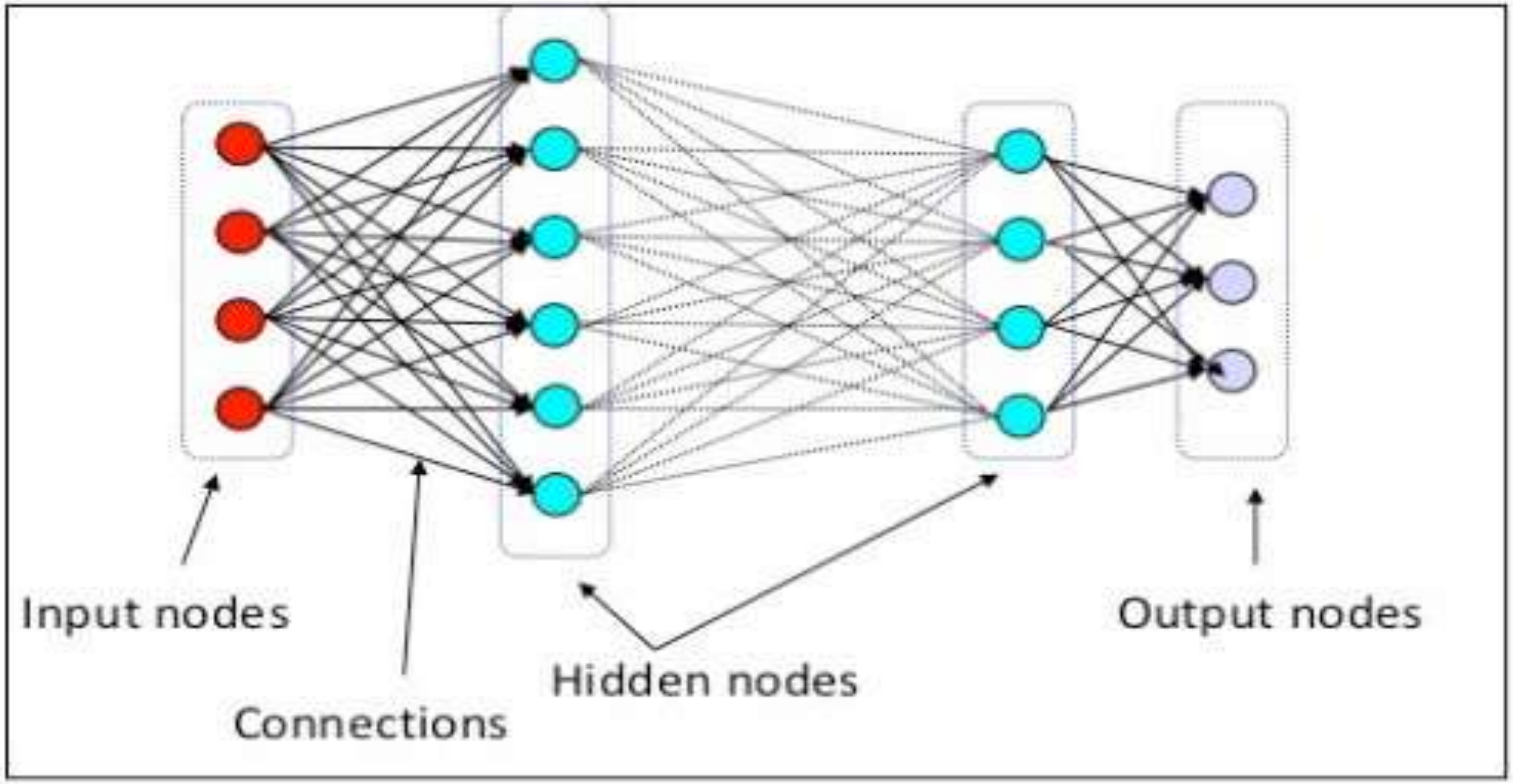
McCulloch and Pitts: every Boolean function can be implemented

The artificial neuron (McCulloch and Pitts 1943)

$$x_1 w_1 + x_2 w_2 = \theta \tag{4}$$



A neural network structure



Many layers with many neurons (order of Millions)

Weighted connections (parameters) among neurons (order of Billions)

Changing the values of the parameters we obtain different results

How can we select the right values ?

Applications of AI: categories

"Normal" human activities (very difficult for machines)

- Natural Language Processing (NLP)
- Vision
- Movement and robotics
- ...

better sub-symbolic

Formal activities

- Games
- Mathematics and Logic
- ....

Specialized activities

- Expert and decision support systems
- Reccomender systems
- Diagnosis
- Planning
- ...

Some examples

Chess
- Limited number of states
- Explicit, non ambiguous rules
- In a sense “easy” (but huge search space)

Natural language processing (Watson)
- Ambiguous, context depended
- Cognitive states

Robotics and autonomous systems
- In a physical ambient
- Dynamic, real-time
- In part non symbolic

Natural language understanding and QA

Watson (IBM) won at Jeopardy, 2011 !

Jeopardy: Given an answer must find a question

Language understanding + Question Answering

Uses 4 terabytes of data (Encylopedia etc.)

Analyses 200 M pages of content in 3 second

Practical applications of Watson

Financial domain: Bridgewater Associates (managing 160 $ bn) hired the chief developer of Watson to create a system for managing daily operations. Long term goal is to have in 5 years ¾ of the managing decision done by software

Health domain: Watson Oncology is a cognitive computing system deevloped at Memorial Sloan Kettering Cancer Center to interpret cancer patients' clinical information and identify individualized, evidence-based treatment options.

Personalized tutoring The Teaching Assistant of the 2016 Artificial Intelligence course at Georgia Tech was a program (based on IBM Watson). It was answering students questions on-line with a success rate of 97%.

Weather forecast Watson used to analyze data from over 200000 stations

John Searle: “Watson Doesn't Know It Won on 'Jeopardy!’ IBM invented an ingenious program—not a computer that can think.”

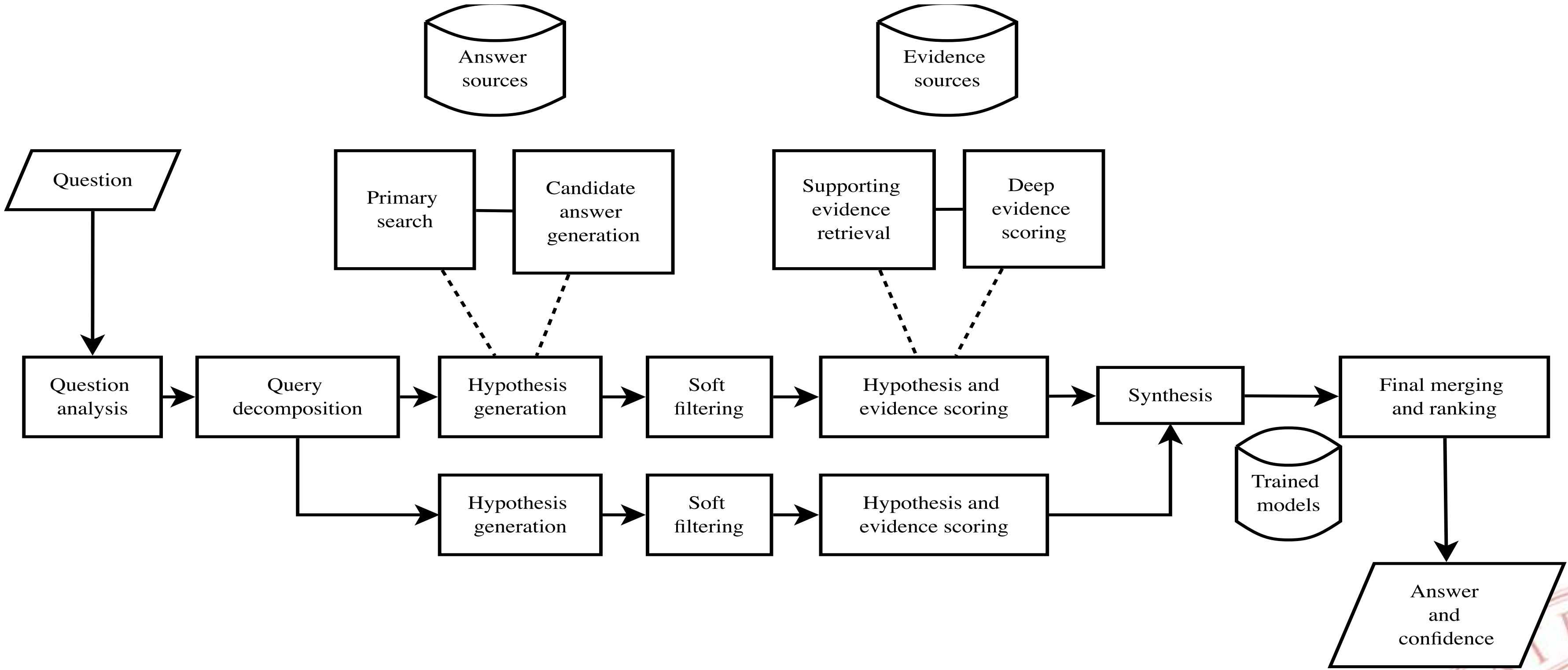
Noam Chomsky: “Watson understands nothing. It’s a bigger steamroller. Actually, I work in AI, and a lot of what is done impresses me, but not these devices to sell computers.”

Not creation of a new algorithm but ability to quickly execute hundreds of proven language analysis algorithms simultaneously to find the correct answer.[12][

Both hardware and software are needed for efficiency and results:

Element	Number of cores	Time to answer one Jeopardy! question
Single core	1	2 hours
Single IBM Power 750 server	32	<4 min
Single rack (10 servers)	320	<30 seconds
IBM Watson (90 servers)	2 880	<3 seconds

“The system we have built and are continuing to develop, called DeepQA, is a massively parallel probabilistic evidence-based architecture. For the Jeopardy Challenge, we use more than 100 different techniques for analyzing natural language, identifying sources, finding and generating hypotheses, finding and scoring evidence, and merging and ranking hypotheses”. [1]



[1] *Building Watson: An Overview of the DeepQA Project*. AI Magazine Fall, 2010. David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty