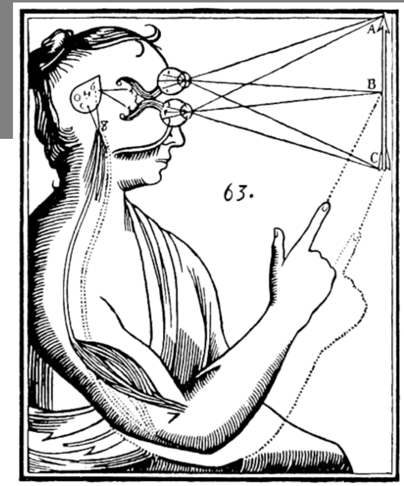
Introduction

Marco Piastra

Artificial Mind?



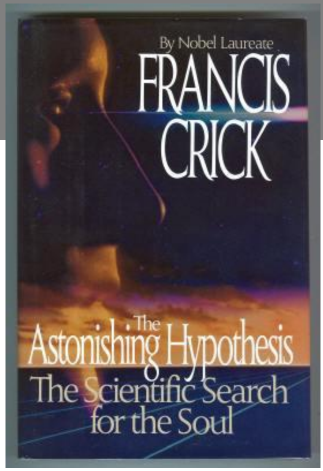
(from Wikipedia)

- [Descartes, R., *Discours de la méthode pour bien conduire sa raison, et chercher la vérité dans les sciences,* 1637]

    "I had after this described the **reasonable soul**, and shown that *it could by no means be educed from the power of matter*, as the other things of which I had spoken, but that it must be expressly created;

    and that it is not sufficient that it be lodged in the human body exactly like a pilot in a ship, unless perhaps to move its members,

    but that it is necessary for it to be joined and united more closely to the body, in order to have sensations and appetites similar to ours, and thus constitute a real man" [English version from Project Gutenberg]



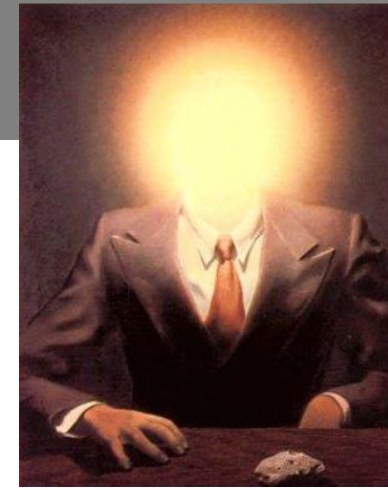
(from Wikipedia)

- [Crick, F., *The Astonishing Hypothesis,* 1994]

  "You, your joys and your sorrows, your memories and your ambitions, your sense of personal identity and free will,

  are in fact no more than the behavior of a vast assembly of nerve cell and their associated molecules."

Artificial Brain: can machines think?



(from Wikipedia)

- [Searle, J. R., *Minds, Brain and Science,* 1986]

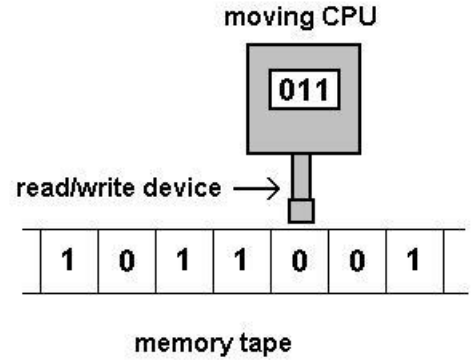
  "Because we do not understand the brain very well
  we are constantly tempted to use the latest technology
  as a model for trying to understand it.

  In my childhood we were always assured that the brain
  was a telephone switchboard ('*What else could it be?*').

  I was amused to see that Sherrington, the great British neuroscientist, thought that the brain worked like a telegraph system. Freud often compared the brain to hydraulic and electro-magnetic systems. Leibniz compared it to a mill, and I am told some of the ancient Greeks thought the brain functions like a catapult.

  At present, obviously, the metaphor is the digital computer."

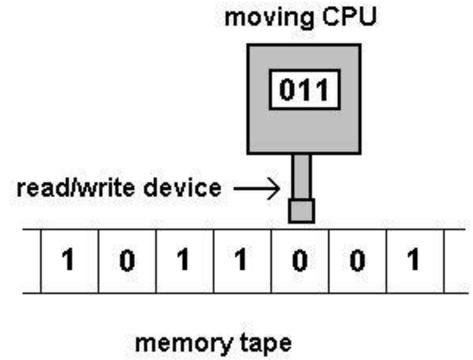
Turing Machine (A. Turing, 1937)



- An abstract model of effective computation
    - A **tape**, made up of individual **cells**
    - Each cell contains a **symbol**, from a finite **alphabet**
    - A **read/write head**, which can move in each direction - one cell at time
    - A **state register** that keeps the current **state**, from a finite set
    - A **transition table**, i.e. a set of *entries* like this:
        - { <*current_state*, *symbol_read*> → <*next_state*, *symbol_written*, *move*> }

- The **transition table** describes a *finite state machine*
    - Each *transition* is governed by the input symbol, the current state and the corresponding entry in the transition table
    - The next state is written into the state register
    - The output is written to the cell
    - Then the head moves (i.e. *left*, *right*, *none*)

Turing Machine (A. Turing, 1937)



- An abstract model of effective computation

  A **tape**, made up of individual **cells**

  Each cell contains a **symbol**, from a finite **alphabet**

  A **read/write head**, which can move in each direction - one cell at time

  A **state register** that keeps the current **state**, from a finite set

  A **transition table**, i.e. a set of *entries* like this:

  { <*current_state*, *symbol_read*> → <*next_state*, *symbol_written*, *move*> }

- What is the meaning of this?

  The Turing Machine is a mathematical model of a physical computing device

  Any given problem for which there is a Turing Machine that computes the solution is clearly computable by a physical machine

  Is the vice-versa also true?

  (*Whenever a problem is computable by a physical machine, does it exist a Turing Machine for it?*)

Church-Turing Thesis

Caution: there is no such a thesis in the original writings of either author. Its formulation could be extrapolated from both, hence the attribution (made by others)

- A possible formulation (from Wikipedia):

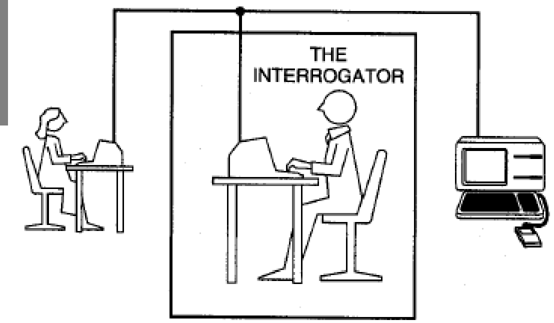
  "Every 'function which would naturally be regarded as computable' can be computed by a Turing machine."

  The vagueness in the above sentence gives raise to different interpretations. One of these (though not entirely equivalent) is (from Wikipedia):

  "Every 'function that could be physically computed' can be computed by a Turing machine."

  Searle: "... At present, obviously, the metaphor is the digital computer."

Can machines think? (the Turing Test)



(from Wikipedia)

- Turing, A., *Computing Machinery and Intelligence*, 1950

  "[The 'imitation game'] is played with three people,
  a man (A), a woman (B), and an interrogator (C) who may be of either sex.

  The interrogator stays in a room apart from the other two.

  The object of the game for the interrogator is to determine which of the other two is the man and which is the woman.

  He knows them by labels X and Y, and at the end of the game he says
  either 'X is A and Y is B' or 'X is B and Y is A'

  The interrogator is allowed to put **questions** to A and B. […]

  We now ask the question,
  'What will happen when a machine takes the part of A in this game?'

  Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman?

  These questions replace our original, 'Can machines think?' "

Artificial Brain:
can machines think?

Artificial Brain:
can machines think?

Do answers, however partial,
change the original question?

Deep Blue

(from Wikipedia)

In 1945 A. Turing mentions playing chess as an example of intelligent human activity that some days machines could perform

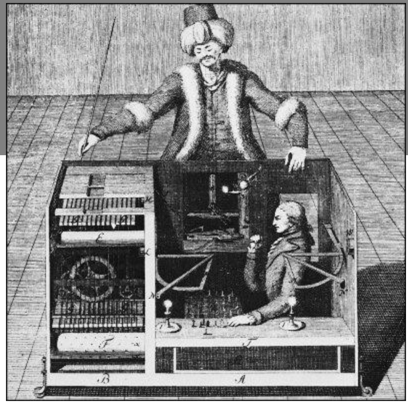
In 1946 A. Turing defines the first *algorithm* for playing chess

In 1948 C. Shannon wrote a famous article on the possible strategies for playing chess *automatically*

In1997 the *Deep Blue* system, made by IBM, beats the world chess champion Gary Kasparov

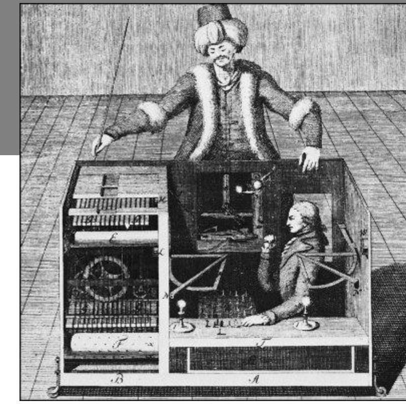
- Deep Blue, 1997 [Campbell, M., Hoane, A. J., Hsu, F., 2001]
  - 30 standard CPUs (120Mhz) + 480 special-purpose CPUs ('chess search engines', each evaluating 2.5M moves per second)
  - Three-layered hardware architecture, 30 GB of RAM
  - Software written in C
  - Wide usage of a large database of recorded games played by grand masters

Can machines play chess?


(from Wikipedia)

- Programming a Computer for Playing Chess [Shannon, 1948]
  - Chess game statistics
    - Typically, 30 legitimate moves at each stage
    - More than 10^{43} different legitimate chessboard configurations
    - More than 10^{120} possible games
  - Strategy A
    - It is based on an *evaluation function* $f(\mathrm{P})$ defined for all possible, **final** positions
    - The machine computes backwards the values of $f(\mathrm{P})$ of all possible, **non-final** positions starting from all possible **final** positions
    - The value assigned to each **non-final** position P is equal to the sum of f values of the **final** position which P may lead to
    - At each move, the computer chooses the move that leads to the position with the maximum value of f

Can machines play chess?



(from Wikipedia)

- Programming a Computer for Playing Chess [Shannon, 1948]

  Chess game statistics

  Typically, 30 legitimate moves at each stage

  More than 10^{43} different legitimate chessboard configurations

  More than 10^{120} possible games

Strategy A

It is based on an *evaluation function* $f(\mathrm{P})$ defined for all possible, **final** positions

The machine computes backwards the values of $f(\mathrm{P})$ of all possible, **non-final** positions starting from all possible **final** positions

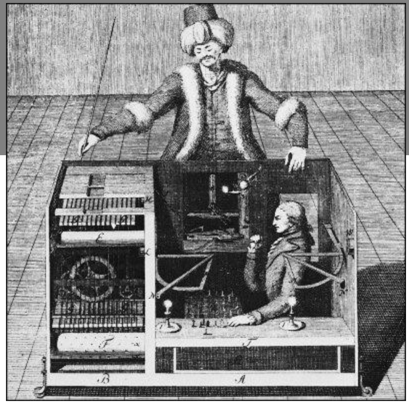
The value assigned to each **non-final** position P is equal to the sum of *f* values of the **final** position which P may lead to

At each move, the computer chooses the move that leads to the position with the maximum value of *f*

This strategy is unfeasible, even with fastest computers, as it entails exploring all possible games

Current supercomputer: ~10^{16} FLOPS,
i.e. ~10^{23} FLO *per year*
At one FLO *per game (irrealistic)*
this would require ~10^{97} *years*

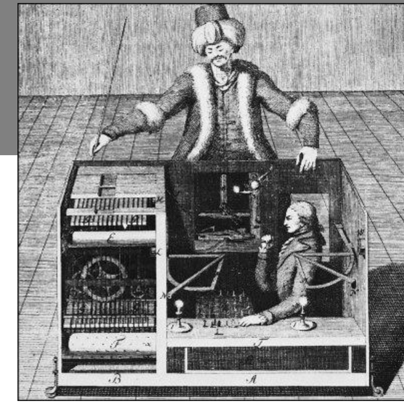
Can machines play chess?



(from Wikipedia)

- Programming a Computer for Playing Chess [Shannon, 1948]
  - Chess game statistics
    - More than 10^{43} different legitimate chessboard configurations
    - More than 10^{120} possible games
  - Strategy A (***revised***)
    - Use an *approximate evaluation function* $f^*(\mathrm{P})$ on all possible positions
    - Given the current position in the game, the machine *looks forward* by exploring all possible positions not farther away than k moves
    - The computer chooses the move with the MINIMAX method (*see next slide*)

Can machines play chess?



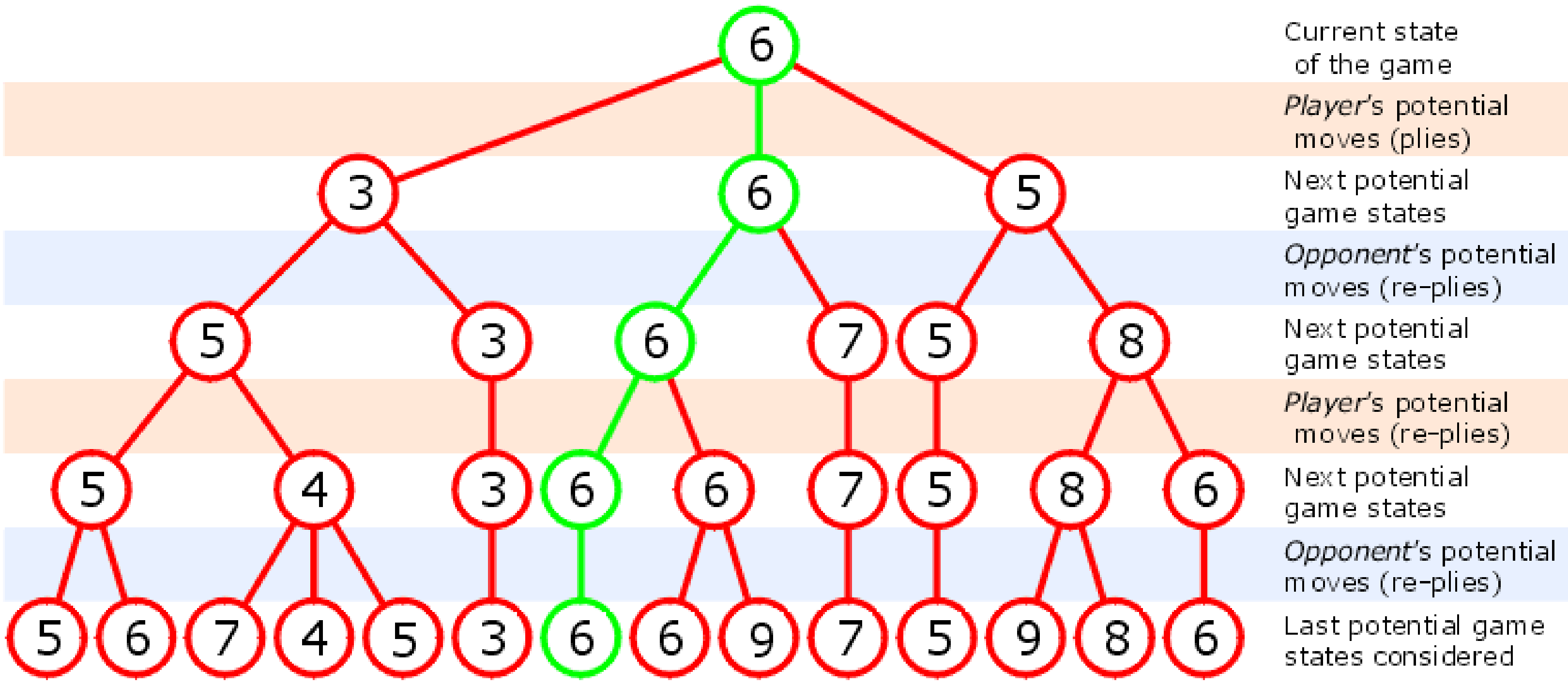
(from Wikipedia)

- Programming a Computer for Playing Chess [Shannon, 1948]

  Strategy A (***revised***)

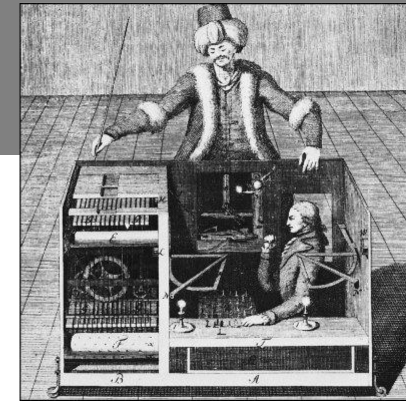
  The computer chooses the move with the MINIMAX method

Minimax on a two-person game tree of 4 plies



(from Wikipedia)

Can machines play chess?



(from Wikipedia)

- Programming a Computer for Playing Chess [Shannon, 1948]

  Strategy A (***revised***)

  In the game tree for chess, each node has an average *branching factor* of 30

  The number of nodes in the game tree is $O(b^d)$

  b is the average *branching factor*
  d is the *depth* (i.e. how far the exploration goes)

  The complete game tree for ply 2 contains 30^2 (i.e. around 10^3) nodes
  The complete game tree for ply 6 around 10^9 nodes

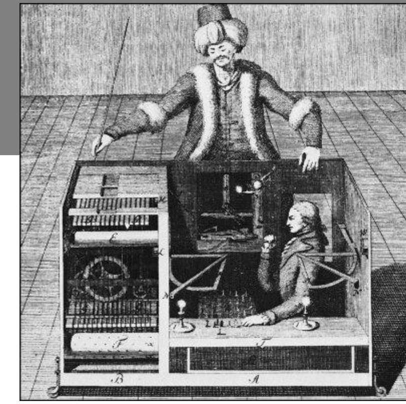
  A computer that can evaluate 10^6 positions per second would take more than 16 minutes

  A typical chess game has ply 80-90

  Human master players are believed to have an implicit *lookahead* of ply 30-40 and more (but without an explicit computation…)

  Note: the MINIMAX method can optimized (i.e. with *alpha-beta pruning*, see Wikipedia) so that it is possible to double the *depth* that can be explored in the same time

Can machines play chess?



(from Wikipedia)

- Programming a Computer for Playing Chess [Shannon, 1948]

  Chess game statistics

  More than 10^{43} different legitimate chessboard configurations

  More than 10^{120} possible games

  Strategy A (***revised***)

  Use an *approximate evaluation function* $f^{*}(\mathrm{P})$ on all possible positions

  Given the current position in the game, the machine *looks forward* by exploring all possible positions not farther away than k moves

  The computer chooses the move with the MINIMAX method (see after)

  Strategy B

  "A good human player examines only **a few selected variations** and carries these out to a reasonable stopping point"

  Use two functions that evaluate the stability of a position P and to what extent a move M in a position P *is worth being examined* at all

  In short: find *higher level* patterns

Strategy A or Strategy B?

[Shannon, 1948]

Due to the high computational complexity of Strategy A,
he foresees a progressive development of Strategy B

(i.e. something like "Computer can improve by emulating humans")

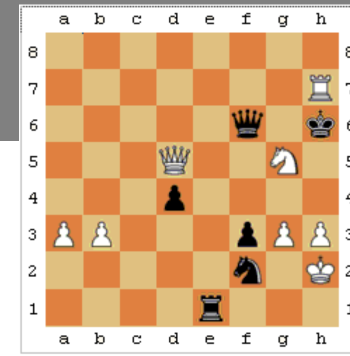
How did it go, in reality?

- At the early stages of computer chess technology, Strategy B was preferred
- During the period 1959-1962 a first 'credible' player was developed (Kotok-McCarthy) (at the *beginner* level)
- In 1973 the developers of the soon-to-be world champion in computer chess players abandoned Strategy B in favor of Strategy A
- Since then, Strategy A – with significant improvements – dominates the scene

  This includes *Deep Blue* and all current top-ranking computers

  Excellent computer chess players (i.e. *grandmaster level*) are now available for smartphones

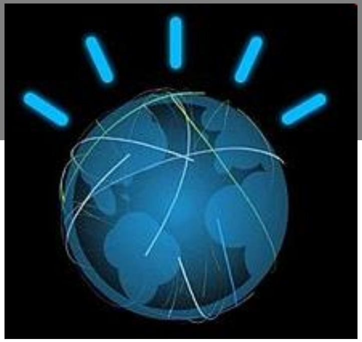
Deep Blue



- Deep Blue, 1997 [Campbell, M., Hoane, A. J., Hsu, F., 2001]
  - Great *lookahead* power
    - On the average, it could search ply 12.2 ply in three minutes
  - Dedicated hardware
    - Special evaluation primitives implemented in silicon
  - Hybrid dedicated machine: hardware + software
    - Software algorithms in C for standard CPUs, easily modified
    - Specialized processors for exploring the game tree
  - Massive parallelism
    - More than 500 processors for parallel exploration
  - Evaluation function based on a huge database of games by grand masters (humans)

(A *supercomputer* for those times - It was turned off at the end of the match)

DeepQA (a.k.a. "Watson")



- *Jeopardy!* : a quiz game

(from Wikipedia)

Category: General Science

Clue: When hit by electrons, a phosphor gives off electromagnetic energy in this form.

Answer: Light (or Photons)

Category: Diplomatic Relations

Clue: Of the four countries in the world that the United States does not have diplomatic relations with, the one that's farthest north.

Answer: North Korea

Category: Rhyme Time

Clue: It's where Pele stores his ball.

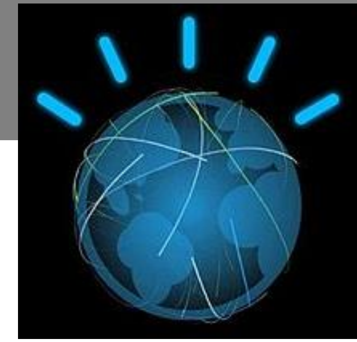
Answer: soccer locker

Category: Lincoln Blogs

Clue: Secretary Chase just submitted this to me for the third time; guess what, pal. This time I'm accepting it.

Answer: his resignation

DeepQA (a.k.a. "Watson")



- DeepQA, 2010 [Ferrucci, D., et al. 2010]

  The Event (14-18/02/2011)

  In a sequence of three "*Jeopardy!*" games, Watson beats in a very convincing way the all-times human champions
  - Brad Rutter, winner of the highest amount of money
  - Ken Jennings, winner of the longest string of games

(from Wikipedia)

  Jeopardy! : a quiz game

  In the real game, questions can also be about images, audio or video displays

  DeepQA can only accept spoken text as input

  Autonomous search, local memory

  The rules of the challenge forbid connecting to Internet during the game: DeepQA must use its local memory only

  It does use Internet during training

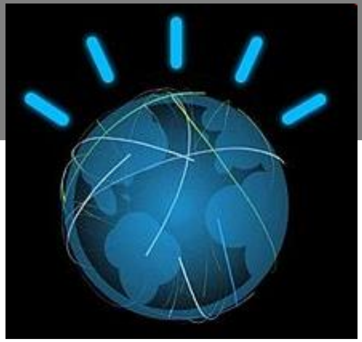
  Conventional hardware, massive parallelism

  High Performance system, with 2880 standard CPUs (no specialized hardware required)

  Linux SUSE ES 11, Software in Java and C++, with Apache Hadoop and Apache UIMA

  (IBM makes business on Watson, now)

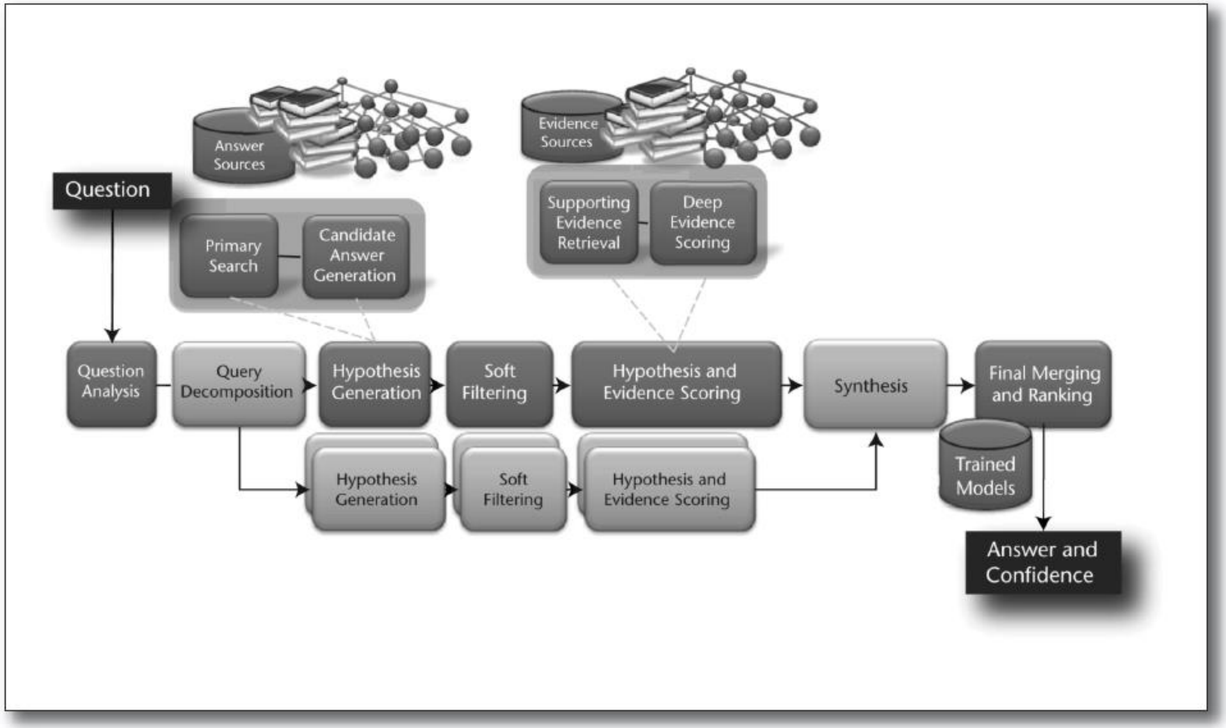
DeepQA (a.k.a. "Watson")



- How does it work?

Very little is known…

(from Wikipedia)



(from [Ferrucci, D, et al. 2010])

Processing stream

"They used nearly every trick in the book.." (from a video on YouTube)

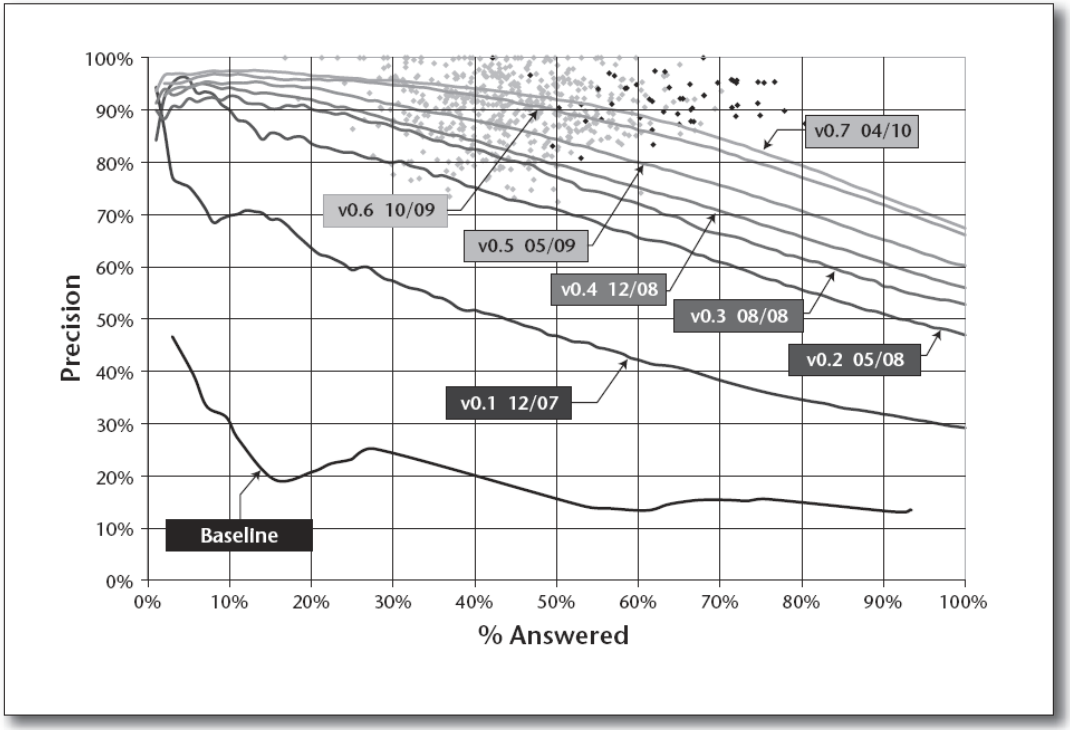
Several competing streams in parallel

Each stream scores a 'degree of confidence': the best answer is chosen, at the end

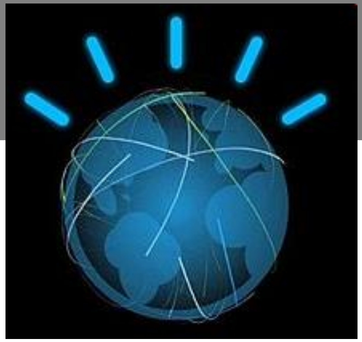
DeepQA (a.k.a. "Watson")

- How does it work?

Very little is known…



(from [Ferrucci, D, et al. 2010])



(from Wikipedia)

Progressive, incremental training

Vast usage of *machine learning* techniques

Is Watson intelligent?

- "Does Watson Think?"

  [D. Ferrucci, transcript from video
  http://www.ted.com/webcast/archive/event/ibmwatson]

  "Huh, hmm, what's my favorite response on that?
  (*Do submarines swim?*)
  [...]
  I'd like to look at it as a sort of task-based view:
  when you think of Watson playing Jeopardy!
  it is acting like an intelligent Jeopardy! player,
  if you deconstruct its intelligence
  you're gonna find lots of different algorithms
  no one of them you would look at and say
  "Wow! That's really intelligent! It really understands the question!"
  [...]
  You have this holistic effect,
  where it's solving a problem that <u>you</u> formally think
  that takes <u>you</u> *think*, to solve that problem, ...
  Watson is doing it in a perhaps different way.
  [...]
  And I think ultimately of it as a tool, that helps humans solving problems... "

(from Wikipedia)

Generalization: Deep Mind

- Playing Atari with Deep Reinforcement Learning
  [2013, V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, http://arxiv.org/abs/1312.5602, see also http://www.nature.com/nature/journal/v518/n7540/full/nature14236.html]

A software system only

Runs on virtually any Linux-based system, it contains optional provisions for GPU

It's open source

https://github.com/kuz/DeepMind-Atari-Deep-Q-Learner

Sophisticated machine-learning techniques

Uses reinforcement learning (RL - *see later in this course*)

in combination with convolutional neural networks (CNN)

Same configuration, multiple games

Same configuration applied to arcade games

It learned to play 7 (2013) or 49 (2015) different games

It is *autonomous*

It learns by itself, it receives no human expertise as input

In many cases, it outperforms human players

(from GitHub)

Beyond Chess: AlphaGo

By Google DeepMind - Google DeepMind AlphaGo Logo, Public Domain

- Mastering the game of Go with deep neural networks and tree search,
  [2016, D. Silver, et al. (22 authors), http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html]

  "The game of Go originated in China more than 2,500 years ago.
  The rules of the game are simple: Players take turns to place black or white stones on a board, trying to capture the opponent's stones or surround empty space to make points of territory.
  As simple as the rules are, Go is a game of profound complexity. There are more possible positions in Go than there are atoms in the universe." [https://deepmind.com/research/alphago/]

A software system only

Monte-Carlo tree search

Instead of a systematic exploration of the tree of possible moves in the game (e.g. MINIMAX) the method 'plays out' entirely a few games, from current position, selecting moves at random: the move corresponding to the best 'playout' is then selected (*plus some optimizations*)

Sophisticated machine-learning techniques

Deep neural networks (trained on human matches) for selecting the moves in the 'playouts'

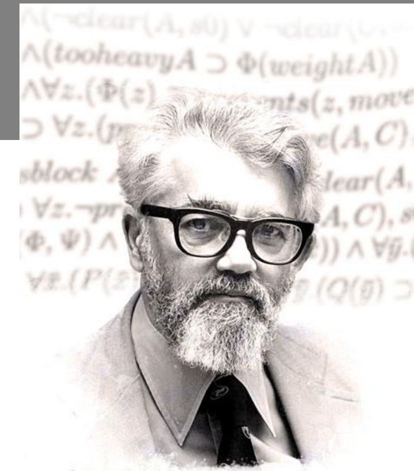
Subsequent autonomous self-training: playing again itself and improving via reinforcement learning (RL - *see later in this course*)

It is a very strong player

On March 2016, AlphaGo won 4-1 against the legendary Lee Sedol, the top Go player in the world over the past decade

Artificial Intelligence
(a few historical hints)

"Artificial Intelligence" (first appearance of the term)


(from Wikipedia)

- [John McCarthy et al., 1955]

  "We propose that a two-month, ten man study
  of **artificial intelligence** carried out during the summer of 1956 [...]

  The study is to proceed on the basis of the conjecture
  that every aspect of learning or any other feature of **intelligence**
  can in principle be ***so precisely described***
  that a machine can be made to ***simulate*** it. [...]

  It may be speculated that a large part of human thought
  consists of manipulating **words** according to **rules** of **reasoning**
  and **rules** of **conjecture**."

The Physical Symbol System Hypothesis (PSSH)

[Newell, A., Simon, H., *Computer Science as Empirical Inquiry Symbols and Search*, 1976]

"A **physical symbol system** consists of a set of entities, called **symbols**, which are physical patterns that can occur as components of another type of entity called an **expression** (or symbol structure).

Thus, a **symbol structure** is composed of a number of **instances** (or tokens) of symbols related in some physical way (such as one token being next to another).

At any instant of time the system will contain a collection of these symbol structures.

Besides these structures, the system also contains a collection of **processes** that operate on *expressions* to produce other *expressions*: processes of creation, modification, reproduction and destruction."

Do elephants play chess?

[Brooks, R., *Elephants Don't Play Chess*, 1990]

- Criticism of *intelligence* intended as the manipulation of *symbols*

  A unique and synchronous control system

  Studies on cerebral lesions suggest otherwise

  A unique, *general purpose* and neutral computational device

  Studies about human visual perception show clear preferences towards some interpretations over others

  A unique language for the internal representation of reality

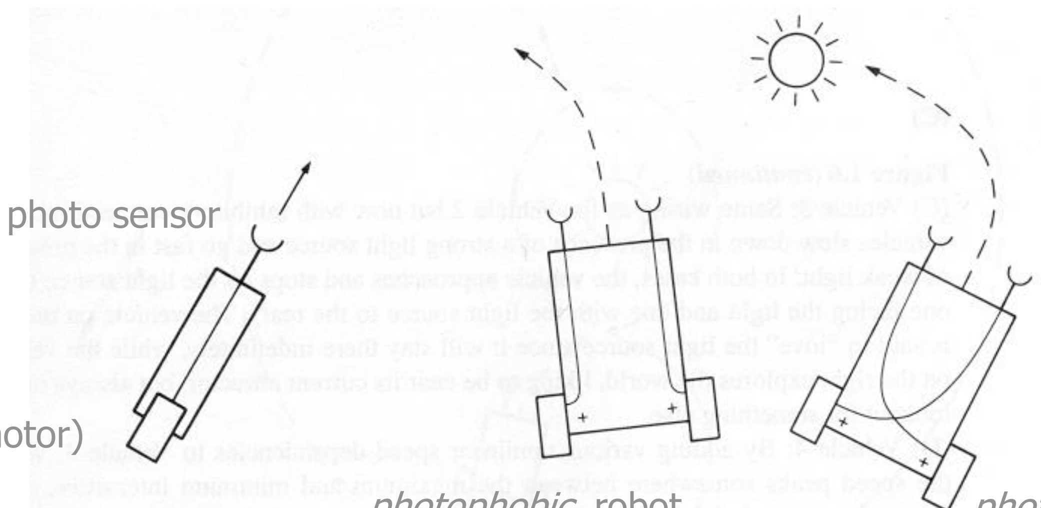
  Human beings do it differently – e.g. *change blindness* [O'Reagan, J. K., Rensink, R. A., Clark, J. J., 1999]

  Total separation between the thinker and its hardware (*disembodiment*)

  Hence excluding all forms of non-symbolical intelligence

  (Besides, how could it possibly *evolve* such a form of intelligence?)

Does this <u>behavior</u> *look* intelligent?



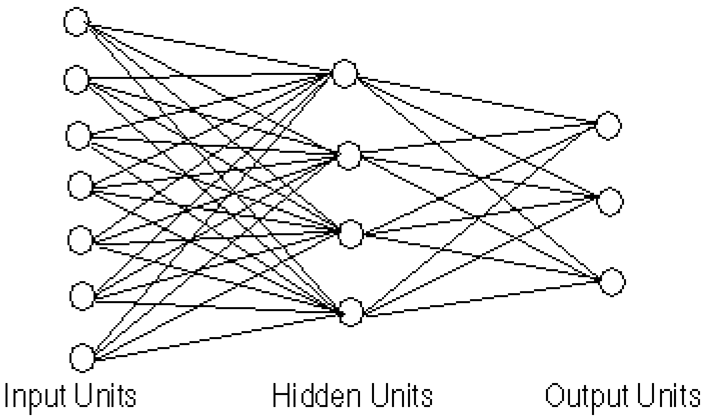
[Braitenberg, V., 1986]

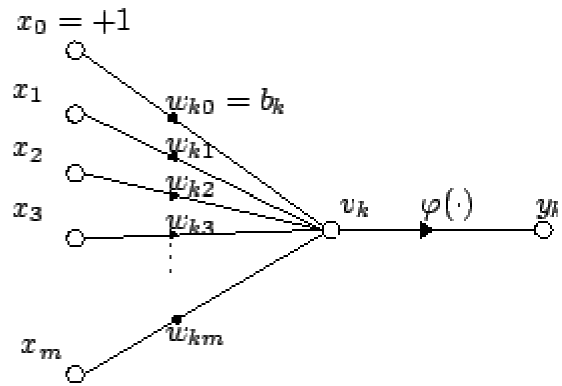
- Direct connection

  These robots by V. Braitenberg have just a *reactive* behavior, i.e. no ‘thought in between’: since sensors are directly connected to actuators

  The resulting behavior is remarkable anyway ... (“*intelligence is in the eye of the beholder*”)

Emulation or simulation? Connectionism






(from Wikipedia)

"In our view, people are smarter than today's computers because the brain employs a basic computational architecture that is more suited to deal with a central aspect of the natural information processing tasks that people are so good at."
[Rumelhart, D.E., J.L. McClelland and the PDP Research Group (1986)
Parallel Distributed Processing: Explorations in the Microstructure of Cognition]

- Basic assumption

  Mental phenomena can be described by interconnected networks of simple and often uniform units

Artificial Intelligence:
short plan of this course

Artificial Intelligence: *part 1*

- *Reasoning with symbols*

  Propositional logic then first-order logic, logic programming (hints)

  Representation: language and semantics

  Inference: entailment

  Automation: machines that computing entailment

  Plausible reasoning: beyond the scope of entailment

Artificial Intelligence: *part 2*

- *Reasoning with numbers*

  Probability and machine learning

  Representation: probability, graphical models

  Inference: answers from joint probabilities

  Supervised learning: learning from complete and well-formed datasets

  Unsupervised learning: when some of the data are either missing or *hidden*

  Reinforcement learning: learning while experiencing (even *online*)

  Self-organization:

  the system changes its configuration in reaction to inputs (even *online*)