Intelligenza Artificiale

The History of Artificial Intelligence

Ivan Heibi
Dipartimento di Filologia Classica e Italianistica (FICLIT)
Ivan.heibi2@unibo.it

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

History of AI (summary)

1940 – 50: Early days

- 1943: McCulloch & Pitts: Boolean circuit model of brain
- 1950: Turing's “Computing Machinery and Intelligence”

1950—70: Excitement

- 1950s: Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
- 1965: Robinson's complete algorithm for logical reasoning

1970—90: Knowledge-based approaches

- 1969—79: Early development of knowledge-based systems
- 1980—88: Expert systems industry booms
- 1988—93: Expert systems industry busts: “AI Winter”

1990—: Statistical approaches:

- Resurgence of probability, focus on uncertainty
- General increase in technical depth
- Agents and learning systems… “AI Spring”?

History of AI (diagram)

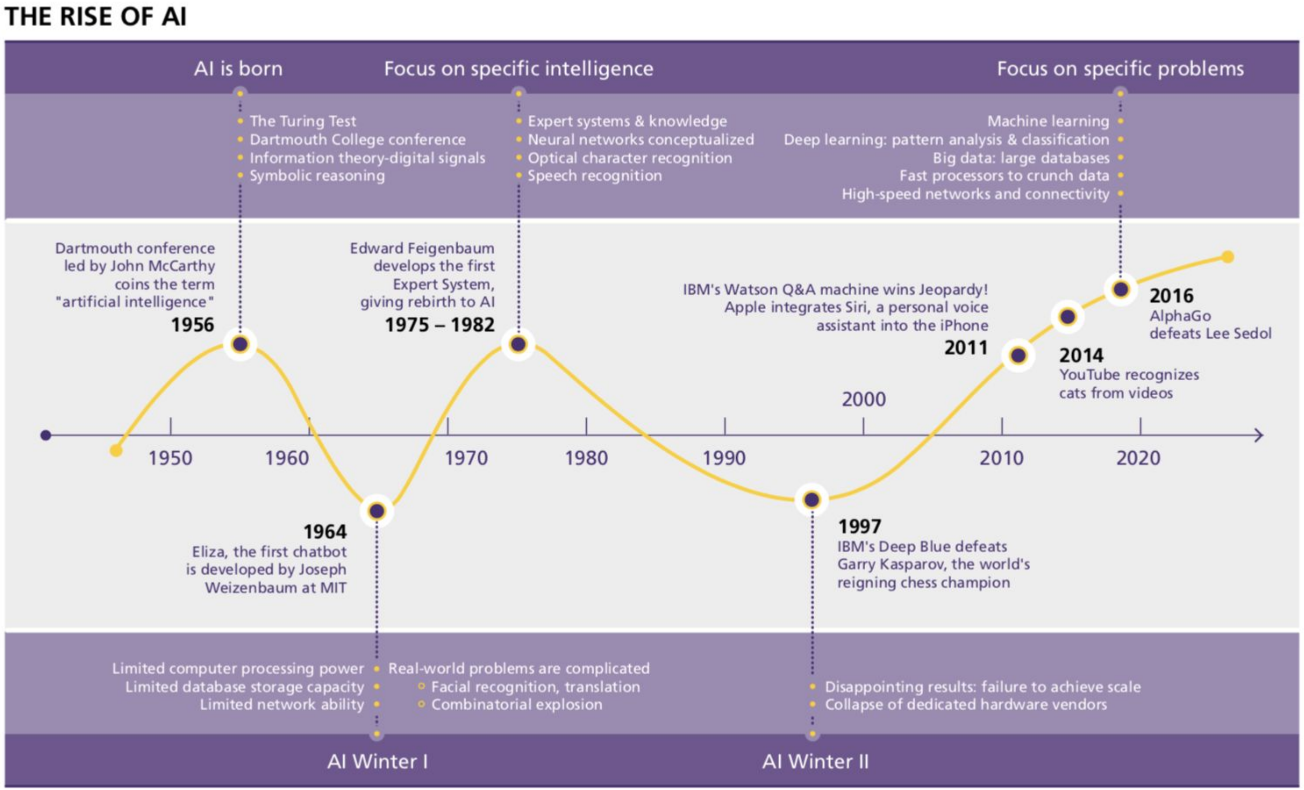


Figure 1: An AI timeline; **Source:** Lavenda, D./Marsden, P.

source dhl via @mikequindazzi

Eliza demo

https://www.masswerk.at/elizabot/

Text to image

https://stablediffusionweb.com/#ai-image-generator

> “a monster wearing glasses in a university class”

> “a flying professor over Rome”

> “Batman attending the university in the city of Bologna”

The gestation of Artificial Intelligence (1943-1955)

The first work that is now generally recognized as AI was done by **Warren McColluch and Walter Pitts (1943)**.

They drew on three sources:

- Knowledge of the basic physiology and **function of neurons** in the brain;
- A formal analysis of **propositional logic** due to Russel and Whitehead;
- Turing's theory of **computation**.

They proposed a model of artificial neurons in which each neuron is characterized as being "on" or "off". With a switch to "on" occurring in response to stimulation by a sufficient number of adjacent neurons.

The state of a neuron was conceived as "factually equivalent to a proposition which proposed its adequate stimulus." They show that any function can be computed by some network of connected neurons (including logical connectives)

The birth of machine learning

McCollough and Pitts also suggested that suitably defined networks could learn.

Donald Hebb (1949) demonstrated a simple updating rule for modifying the connection strengths between neurons.

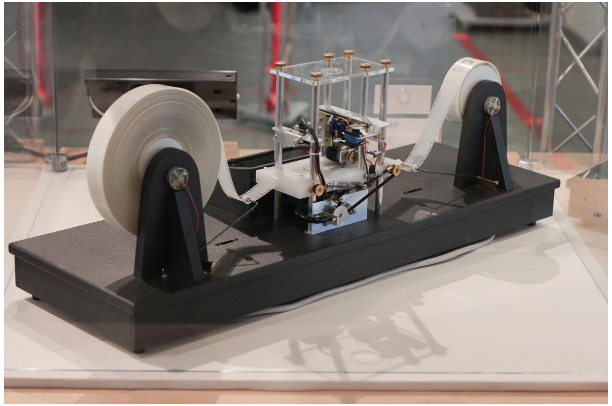
His rule, now called Hebbian learning, remains an influential model today.

Marvin Minsky and Dean Edmons built the first neural network computer in 1950 (which used 3000 vacuum tubes and a surplus automatic pilot mechanism to simulate a network of 40 neurons)

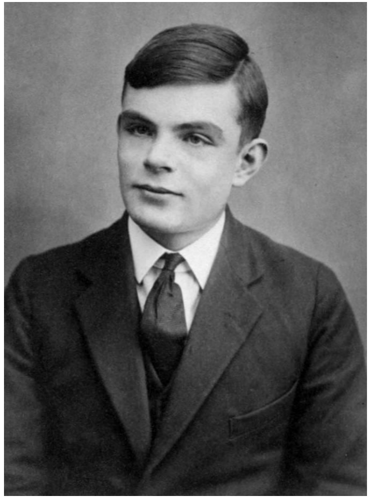
Alan Turing

Alan Turing vision was the most influential one.

He introduced the Turing Test, machine learning, genetic algorithms, and reinforcement learning.

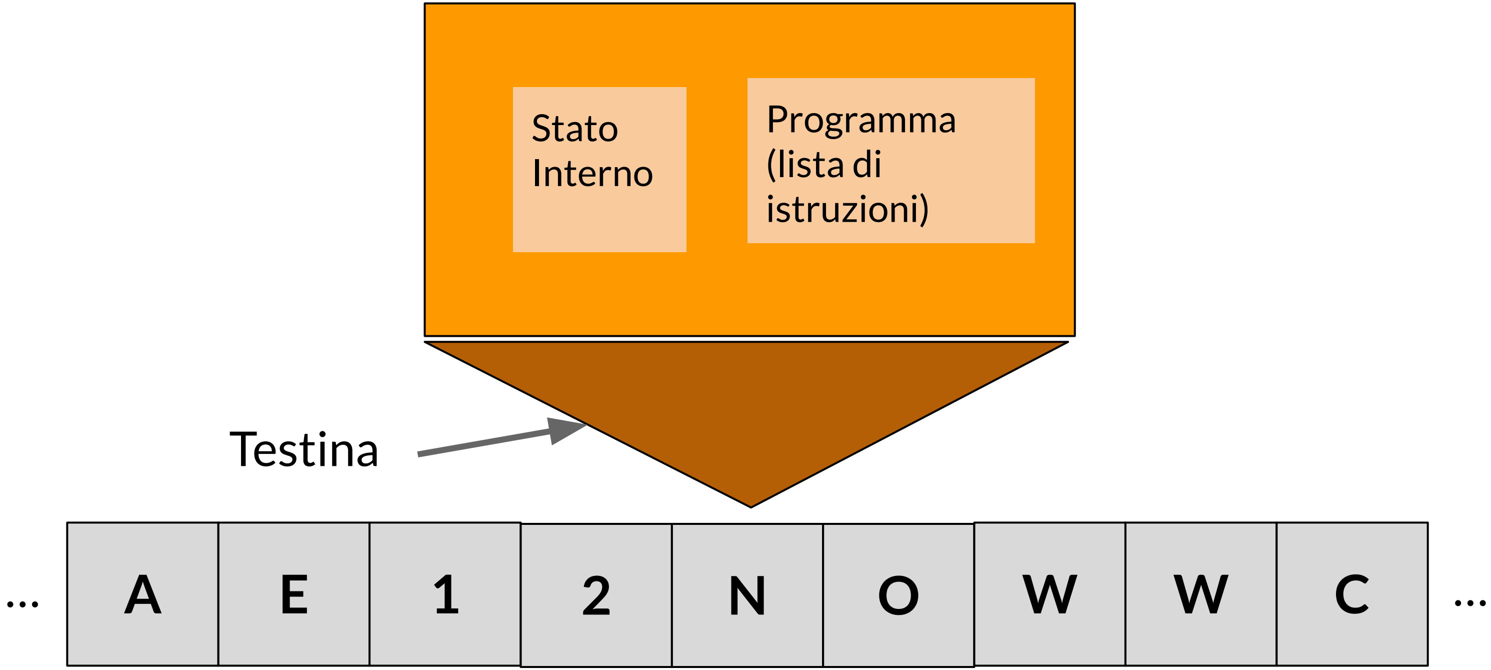


https://en.wikipedia.org/wiki/File:Turing_Machine_Model_Davey_2012.jpg



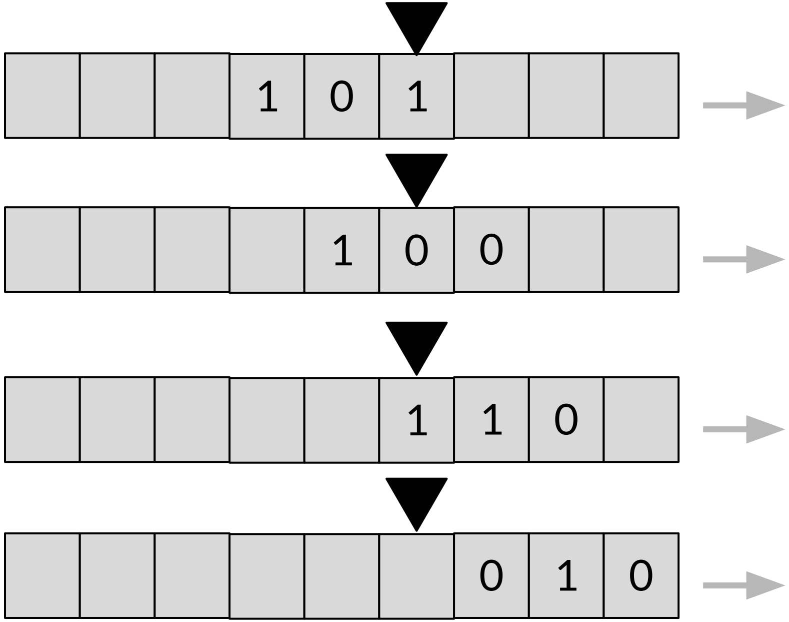
Turing, Alan M. "Computing machinery and intelligence" (1950)

Turing machine



Turing machine – example

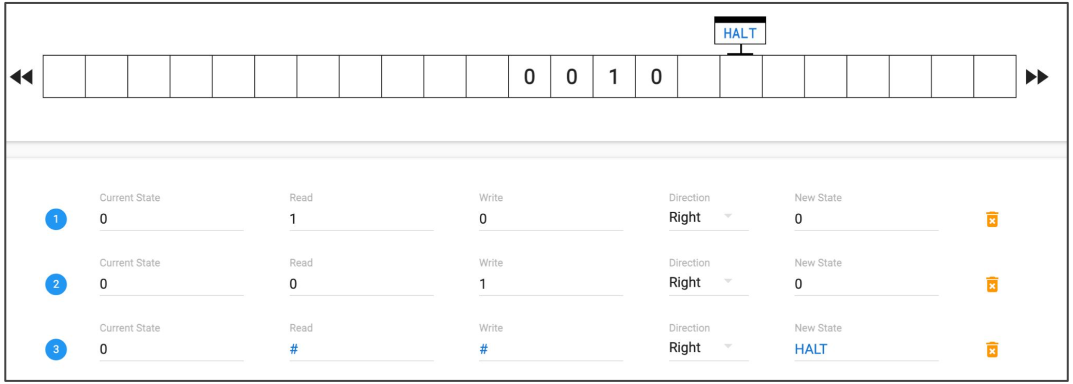
Symbol read	Write instruction	Move instruction
Blank	None	None
0	Write 1	Move tape to the right
1	Write 0	Move tape to the right



Turing machine – example

State	**Symbol read**	**Write instruction**	**Move instruction**	***Next state***
State 0	Blank	None	None	*Stop state*
	0	Write 1	Move the tape to the right	*State 0*
	1	Write 0	Move the tape to the right	*State 1*

Turing machine – demo



http://turingmachine.vassar.edu/pQ10hDdA

The birth of artificial intelligence (1956)

John McCarthy convinced Minsky, Shannon, Rochester to help him to bring together U.S. researchers interested in automata theory, neural nets, and the study of intelligence.

They organised a two-month workshop at **Dartmouth** in the summer of 1956.

Two researchers from Carnegie Tech stole the show: Allen Newell and Herbert Simon.

They had a reasoning program: *“We have invented a computer program capable of thinking non-numerically”*

logic-theorist - The sources of the first theorem prover.
https://github.com/theoremprover-museum/logic-theorist

Early enthusiasm, great expectations (1952-1969)

The intellectual establishment preferred to believe that “a machine can never do X”.

AI researchers naturally responded by demonstrating one X after another.

John McCarthy referred to this period as the “Look, Ma, no hands!” era.

Newell and Simon’s early success was followed up with the General Problem Solver.

Newell and Simon formulated the famous physical subsystem symbol hypothesis which states that “a physical symbol system has the necessary and sufficient means for general intelligent action”. It means that any system exhibiting intelligence must operate by manipulating data structures.

Herbert Gelernter (1959) constructed the Geometry Theorem Prover

Arthur Samuel wrote programs for playing checkers

Early enthusiasm, great expectations (1952-1969): John McCarthy

McCarthy defined high-level **programming language Lisp (that became dominant for the next 30 years)**.

In 1958, McCarthy published a paper entitled “Programs with Common Sense” in which **he describe the Advice Taker**, a hypothetical program that can be seen as the first complete AI system.

The program was designed to use knowledge to search for solutions to problems.

The Advice Taker embodied the central principles of knowledge representation and reasoning:

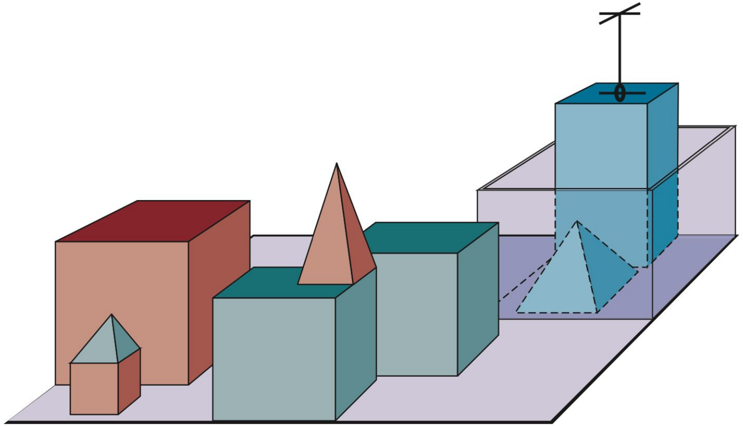
- McCarthy showed how some simple axioms would enable Advice Taker to generate a plan to drive to the airport.
- The Advice Taker was designed to accept new axioms in the normal course of operation (without being reprogrammed).
- The Advice Taker had both a formal explicit representation of the world and a set of general for manipulating the representation thus mimicking the deduction process.
- It separated the knowledge from the reasoning component.

Early enthusiasm, great expectations (1952-1969): Marvin Minsky

Minsky supervised a series of students who chose problems that appeared to require intelligence to solve. > **Divide-et-impera approach** that significantly characterizes AI today!

These limited domains became known as **microworlds**, such as:

- A closed form of calculus integration
- Geometric analogies (e.g., ANALOGY)
- Algebra story problems (e.g., STUDENT)
- Block worlds which consists of a set of solid blocks placed on a tabletop. A typical task in this world is to rearrange the blocks in a certain way, using a robot hand that can pick up one block at a time.



Self assessment

https://forms.gle/WnM4LNHKyKxVYffSA

A dose of reality (1966-1973) (1)

In 1957, Herbert Simon made concrete prediction:

> Within 10 years a computer would be chess champion

> A significant mathematical theorem would be proved by machine

These predictions came true in 40 years (rather than 10).

The early systems turned out to fail miserably when tried out on wider selections of problems and on more difficult problems.

The first kind of difficulty arose because most early programs succeeded by means of simple syntactic manipulations.

A dose of reality (1966-1973) (2)

The second kind of difficulty was the intractability of many of the problems that AI was attempting to solve.

Most of the early AI system implemented a **combinatorial approach to solve problems**.

Before theory of computational complexity was developed, it was widely thought that “scaling up” to larger problems was simply a matter of faster hardware and larger memory!

The fact that a program can find a solution in principle does not mean that the program contains any of the mechanisms needed to find it in practice.

Genetic algorithms were introduced but didn’t work until representations were improved.

Knowledge-based systems: The key to power? (1969-1979) (1)

Problem solving methods that had arisen during the first decade of AI was of **general purpose search mechanism** trying to string together elementary reasoning steps to find complete solutions.

> Such approaches have been called weak methods because, although general, they do not scale up to larger or difficult problem instances.

The alternative to weak methods is to use more powerful, **domain-specific knowledge that allows larger reasoning steps** and can more easily handle typically occurring cases in narrow area of expertise. (one might say that to solve a hard problem, you have to almost know the answer already!). These kinds of systems are also called expert systems.

Example:

- DENDRAL program (1969) which was able to infer molecular structure from the information provided by a mass spectrometer. Instead of trying all possible structures, it uses knowledge about recurrent embedded substructures
- Medical Diagnosis tools which incorporate a calculus of uncertainty.
- R1 program by DEC, to buy computers from a shop

Knowledge-based systems: The key to power? (1969-1979) (2)

The importance of incorporating domain knowledge motivates the effort in developing programs able to understand natural language.

Charniak suggested that robust language understanding would require general knowledge about the world and a general method for using the knowledge.

The emphasis was less on language per se and more on the problems of representing and reasoning with the knowledge, for example:

- Representing stereotypical situations
- Describing human memory organization
- Understanding plans and goals

Knowledge-based systems: The key to power? (1969-1979) (3)

The widespread growth of applications to real-world problems caused a concurrent increase in the demands for workable knowledge representation scheme

Some based on **Prolog,** others followed Minksy's idea of **frames** ("A framework for representing knowledge." (1974))

Expert systems industry booms and the busts (1980 - 1993)

AI industry boomed from a few million dollars in 1980 to billions of dollars in 1988, including hundreds of companies **building expert systems,** vision systems, robots, and software and hardware specialized for these purposes.

Soon after that came a period called the "**AI winter**," in which many companies fell by the wayside as they failed to deliver on extravagant promises (complex domains).

Reasons:

> the reasoning methods used by the systems broke down in the face of **uncertainty**

> the systems could **not learn from experience**.

Neural networks and machine learning (1986 - present)

The **back-propagation learning algorithm** (the connectionist models)

> these methods have the **capability to learn** from examples (experience)—they can compare their predicted output value to the true value on a problem and modify their parameters to decrease the difference, making them more likely to perform well on future examples.

The brittleness of expert systems led to a new, more scientific approach incorporating **probability** rather than Boolean logic, and **machine learning** rather than hand-coding.

Big data (2001 - present)

Throughout the 60-year history of computer science, the emphasis has been on the algorithm as the main subject of study.

Some recent work in AI suggests that for many problems, it makes more sense to worry about the data and be less picky about what algorithm to apply.

One influential paper in this line was **Yarowsky's work on Word Sense Disambiguation**: given the use of the word "plant" in a sentence, does that refer to flora or factory?

> Previous approaches relied on human-labelled examples combined with machine-learning algorithms.

> Yarosky showed that this task can be done with no labelled examples at all with accuracy of 96% by analysing a very large corpus with the dictionary definitions of the two senses (i.e. "flora, plant life" and "works, industrial plant").

Deep learning (2011–present)

Deep learning refers to machine learning using **multiple layers** of simple, adjustable computing elements.

Experiments were carried out with such networks as far back as the 1970s, and in the form of convolutional neural networks they found some success in handwritten digit recognition in the 1990s.

It was not until 2011, however, that deep learning methods really took off. This occurred first in speech recognition and then in visual object recognition. In 2012 in a competition to classify images into one of a thousand categories (armadillo, bookshelf, corkscrew, etc.), a deep learning system created in Geoffrey Hinton, demonstrated a dramatic improvement over previous systems, which were based largely on handcrafted features.

Deep learning depends on **powerful hardware** an the availability of large amounts of **training data**

The State of the Art

AI100 from the Stanford University convenes panels of experts to provide reports on the state of the art in AI, it also produces an **AI Index** at aiindex.org to help track progress

What can AI do today?

- Robotic vehicles
- Machine translation
- Speech recognition
- Recommendations
- Game playing
- Image understanding
- Medicine
- …

Risks and Benefits of AI

Benefits: the potential for AI and robotics to free humanity from menial repetitive work and to dramatically increase the production of goods and services could presage an era of peace and plenty. E.g., The capacity to accelerate scientific research could result in cures for disease and solutions for climate change and resource shortages

Risks:

- **Biased decision making**
- **Impact on employment**
- **Deep Fake images**
- **...**

Self assessment

https://forms.gle/oo9czUTLf8eQm6du6