History of AI

Early excitement

1940s	McCulloch & Pitts neurons; Hebb's learning rule
1950	Turing's "Computing Machinery and Intelligence"
1954	Georgetown-IBM machine translation experiment
1956	**Dartmouth meeting: "Artificial Intelligence" adopted**
1950s-1960s	**"Look, Ma, no hands!" period:**
	Samuel's checkers program, Newell & Simon's
	Logic Theorist, Gelernter's Geometry Engine

Herbert Simon, 1957

- *"It is not my aim to surprise or shock you--- but … there are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until--- in a visible future---the range of problems they can handle will be coextensive with the range to which human mind has been applied.* ***More precisely: within 10 years a computer would be chess champion, and an important new mathematical theorem would be proved by a computer.****"*

- Simon's prediction came true --- but 40 years later instead of 10
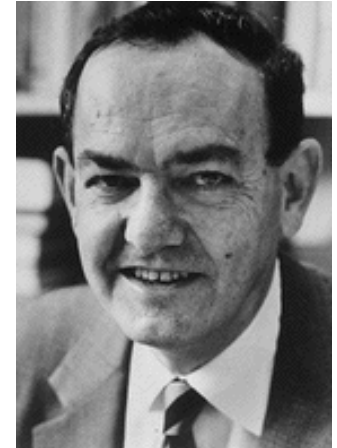
A dose of reality

1940s   McCulloch & Pitts neurons; Hebb's learning rule

1950   Turing's "Computing Machinery and Intelligence"

1954   Georgetown-IBM machine translation experiment

1956   **Dartmouth meeting: "Artificial Intelligence" adopted**

1950s-1960s   **"Look, Ma, no hands!" period:**

               Samuel's checkers program, Newell & Simon's

               Logic Theorist, Gelernter's Geometry Engine

1966—73   Setbacks in machine translation

               Neural network research almost disappears

               Intractability hits home

Harder than originally thought

- ## 1966: Weizenbaum's <u>Eliza</u>:
  - " … mother …" → "Tell me more about your family"
  - "I wanted to adopt a puppy, but it's too young to be separated from its mother."

- ## 1950s: during the Cold War, automatic Russian-English translation attempted
  - 1954: Georgetown-IBM experiment
    - Completely automatic translation of more than sixty Russian sentences into English
    - Only six grammar rules, 250 vocabulary words, restricted to organic chemistry
  - 1966: ALPAC report: machine translation has failed to live up to its promise
    - "The spirit is willing but the flesh is weak." → "The vodka is strong but the meat is rotten."
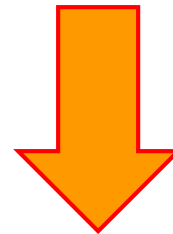
Blocks world (1960s – 1970s)
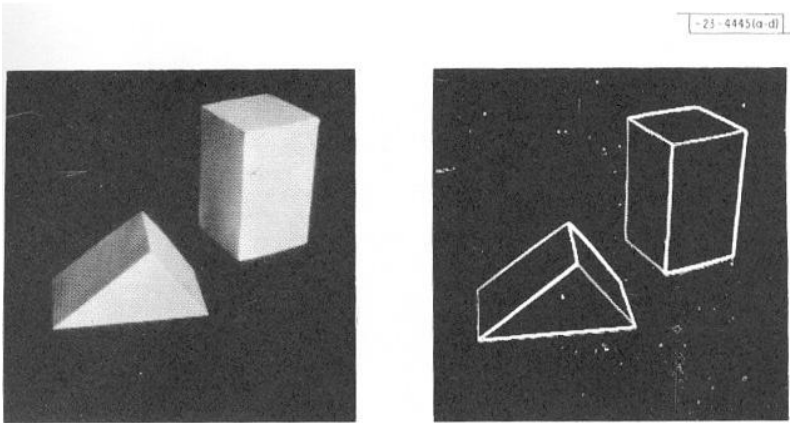
(a) Original picture.

(b) Differentiated picture.
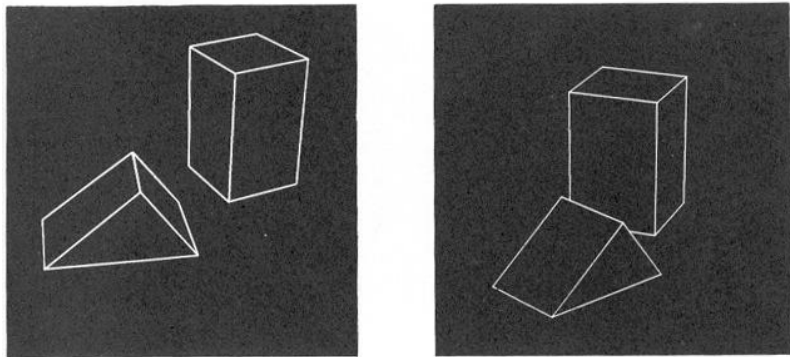
(c) Line drawing.

(d) Rotated view.

Roberts, 1963

???

"Moravec's Paradox"

- Hans Moravec (1988): *"It is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility."*

- Possible explanations
  - Early AI researchers concentrated on the tasks that "white male scientists" found the most challenging, abilities of animals and two-year-olds were overlooked
  - We are least conscious of what our brain does best
  - Sensorimotor skills took millions of years to evolve
  - Our brains were not designed for abstract thinking

The rest of the story

1974-1980 The first "AI winter"

1970s Knowledge-based approaches

1980-88 Expert systems boom

1988-93 Expert system bust; the second "AI winter"

1986 Neural networks return to popularity

1988 Pearl's *Probabilistic Reasoning in Intelligent Systems*

1990 Backlash against symbolic systems; Brooks' "nouvelle AI"

1995-present Increasing specialization of the field

    Agent-based systems

    Machine learning everywhere

    Tackling general intelligence again?

History of AI on Wikipedia
AAAI Timeline
Building Smarter Machines: NY Times Timeline

Some patterns from history

- ## Boom and bust cycles
  - Periods of (unjustified) optimism followed by periods of disillusionment and reduced funding

- ## "High-level" vs. "low-level" approaches
  - **High-level:** start by developing a general engine for abstract reasoning
    - Hand-code a knowledge base and application-specific rules
  - **Low-level:** start by designing simple units of cognition (e.g., neurons) and assemble them into pattern recognition machines
    - Have them learn everything from data

- ## "Neats" vs. "scruffies"
  - Today: triumph of the "neats" or triumph of the "scruffies"?

What accounts for recent successes in AI?

- Faster computers
  - The IBM 704 vacuum tube machine that played chess in 1958 could do about **50,000 calculations per second**
  - Deep Blue could do **50 billion calculations per second** – a million times faster!

- Lots of storage, lots of data

- Dominance of statistical approaches, machine learning

AI gets no respect?

- Ray Kurzweil: *"Many observers still think that the AI winter was the end of the story and that nothing since come of the AI field, yet today many thousands of AI applications are deeply embedded in the infrastructure of every industry."*

- Nick Bostrom*: "A lot of cutting edge AI has filtered into general applications, often without being called AI because once something becomes useful enough and common enough it's not labeled AI anymore."*

- Rodney Brooks: *"There's this stupid myth out there that AI has failed, but AI is around you every second of the day."*

AI gets no respect?

- [AI effect](): As soon as a machine gets good at performing some task, the task is no longer considered to require much intelligence

- Calculating ability used to be prized – not anymore

- Chess was thought to require high intelligence
  - Now, massively parallel computers essentially use brute force search to beat grand masters

- Learning once thought uniquely human
  - Ada Lovelace (1842): "The Analytical Engine has no pretensions to *originate* anything. It can do *whatever we know how to order it* to perform."
  - Now machine learning is a well-developed discipline

- Similar picture with animal intelligence

- Does this mean that there is no clearcut criterion for what constitutes intelligence?

Take-away message for this class

- Our goal is to use machines to solve hard problems that traditionally would have been thought to require human intelligence

- We will try to follow a sound scientific/engineering methodology

  - Consider relatively limited application domains

  - Use well-defined input/output specifications

  - Define operational criteria amenable to objective validation

  - Use *abstraction* to zero in on essential problem features

  - Focus on general-purpose tools with well understood properties