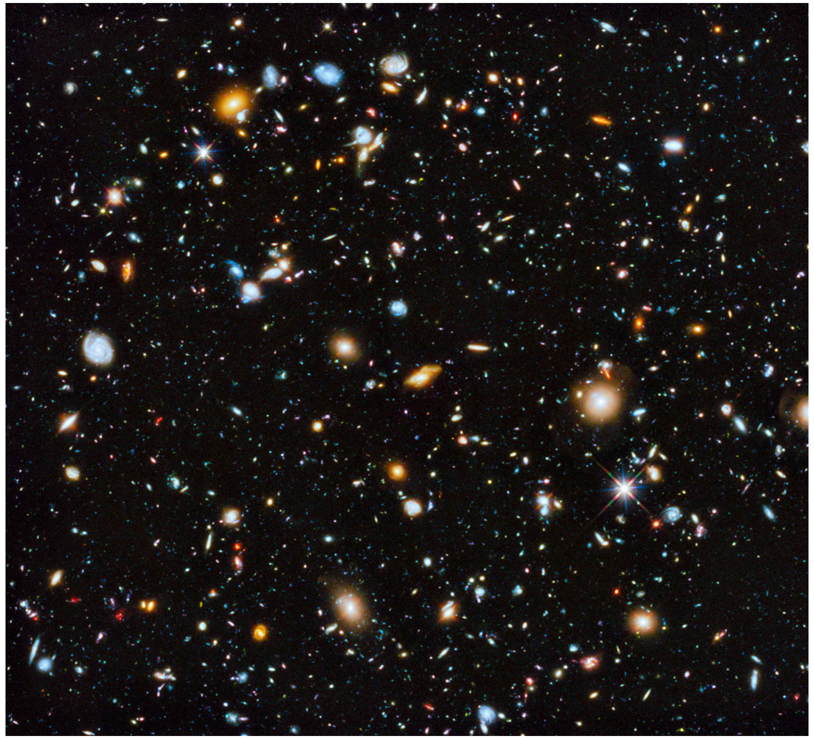
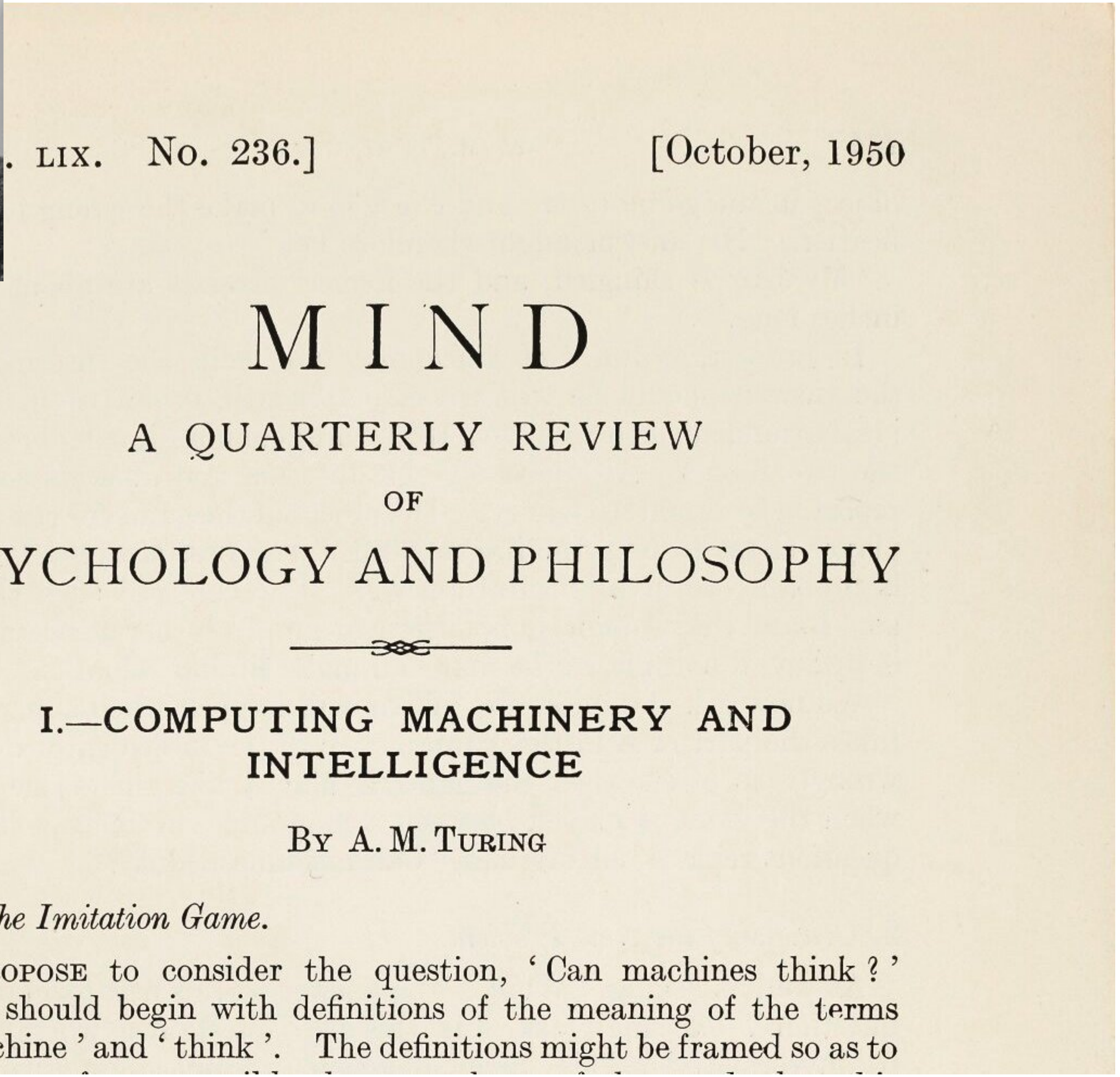
General: AI history



- I will present a short history of AI, which will necessarily be simplified and incomplete. But, I hope that it will give you a general appreciation of AI's multi-faceted history.



. LIX. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

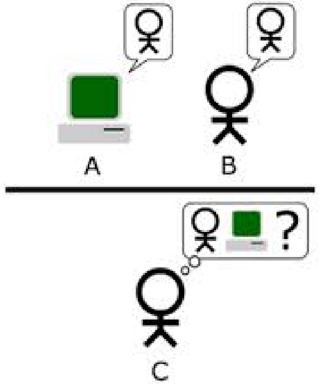
PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to



objective specification

Many people think that a very abstract activity, like the playing of chess, would be best. It can also be maintained that it is best to provide the machine with the best ***sense organs*** *that money can buy, and then teach it to understand and speak English. This process could follow the normal* ***teaching of a child****. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried.*

- A natural place to start talking about the history of AI is Alan Turing's landmark paper published in 1950 called Computing Machines and Intelligence.
- In this paper, Turing asked the question, "Can machines think?" and answered it with the Imitation Game, more commonly known as the Turing Test. As some of you might know, a machine is said to pass the Turing test if it can convince a human judge that it's actually a human through natural language dialogue.
- This paper is remarkable not because it built a system or proposed any methods, but because it framed the philosophical discussions for decades to come. You have to appreciate how difficult a notion like intelligence is to pin down. So this was really the first actionable, formal answer to the question, "Can machines think?"
- Whether passing the Turing test is something that should be directly worked on is questionable and controversial, but the philosophical implications are quite thought-provoking.
- For us, one important takeaway of the Turing test is the separation of the **objective** specification of what we want a system to do (the "what") from the methods that might get us there (the "how"). This decoupling is a major theme throughout this course.
- At the end of the paper, Turing discusses two possible approaches. The first is based on solving abstract problems like chess, which is the route taken by symbolic AI. The second is where you build a machine and teach like a child, which is the route taken by neural and statistical AI.
- I will now tell three stories of symbolic, neural, and statistical AI.

1956

- 1956 is the beginninig of our first story.

Birth of AI

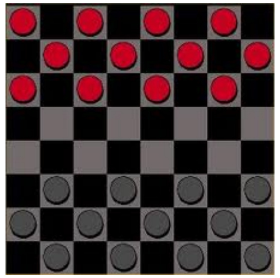
1956: John McCarthy organized workshop at Dartmouth College

Every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it.

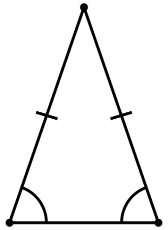
general principles

- It is the year that the name **artifical intelligence** was coined.
- John McCarthy, who later founded the Stanford AI lab, organized a workshop at Dartmouth College that summer.
- In addition to McCarthy, the workshop was attended by Marvin Minsky, Allen Newell, Herbert Simon, etc., all of whom went on to make seminal contributions in AI.
- The participants laid out a bold proposal: to build a system that could capture every aspect of intelligence. They were after **generality**.
- Indeed, during this post-war era, computers were just coming on the scene. It was a very exciting time and people were ambitious.

Birth of AI, early successes



Checkers (1952): Samuel's program learned weights and played at strong amateur level



Problem solving (1955): Newell & Simon's Logic Theorist: prove theorems in Principia Mathematica using search + heuristics; later, General Problem Solver (GPS)

- A few notable systems were created during this time.
- Arthur Samuel wrote a program that could play checkers at a strong amateur level.
- Alan Newell and Herbert Simon's Logic Theorist could prove theorems. For one theorem, it actually found a proof that was more elegant than the human-written proof. They tried to publish a paper on the result, but the paper got rejected because it was not a new theorem. Perhaps the reviewers failed to realize that the third author was actually a computer program.
- Later, they developed the General Problem Solver, which promised to solve any problem (which could be suitably encoded in logic), again carrying forward the ambitious "general intelligence" agenda.

Overwhelming optimism...

Machines will be capable, within twenty years, of doing any work a man can do. —Herbert Simon

Within 10 years the problems of artificial intelligence will be substantially solved. —Marvin Minsky

I visualize a time when we will be to robots what dogs are to humans, and I'm rooting for the machines. —Claude Shannon

- With these initial success, it was a time of high optimism, with all the leaders of the field, all impressive thinkers, predicting that AI would be "solved" in a matter of years.

...underwhelming results

Example: machine translation

The spirit is willing but the flesh is weak.

(Russian)

The vodka is good but the meat is rotten.

1966: ALPAC report cut off government funding for MT, first AI winter

- Despite the successes, certain tasks such as machine translation were complete failures.
- There is a folklore story of how the sentence "The spirit is willing but the flesh is weak" was translated into Russian and then back to English, leading to the amusing translation "The vodka is good but the meat is rotten".
- However, this translation was not so amusing to government agencies funding the research. In 1966, the ALPAC report resulted in funding being cut off for machine translation.
- This marked the beginning of the first AI winter.

Implications of early era

Problems:

- **Limited computation**: search space grew exponentially, outpacing hardware
- **Limited information**: complexity of AI problems (number of words, objects, concepts in the world)

Useful contributions (John McCarthy):

- Lisp
- Garbage collection
- Time-sharing

- What went wrong? Two things.
- The first was computation. Most of the approaches casted problems as logical reasoning, which required a search over an exponentially large search space. The hardware at the time was simply too limited.
- The second is information. Even if researchers had infinite computation, AI would not have been solved. There are simply too many concepts, words, and objects in the world, and this information has to somehow be put into the AI system.
- Though the grand ambitions were not realized, some generally useful technologies came out of the effort. Lisp was way ahead of its time in terms of having advanced language features. People programming in high-level languages like Python take garbage collection for granted. And the idea that a single computer could simultaneously be used by multiple people (time sharing) was prescient.

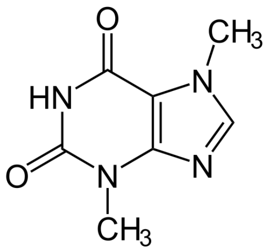
Knowledge-based systems (70-80s)

Expert systems: elicit specific domain knowledge from experts in form of rules:

IF: 1) The stain of the organism is gramneg, and
2) The morphology of the organism is rod, and
3) The aerobicity of the organism is aerobic
THEN: There is strongly suggestive evidence (.8) that
the class of the organism is enterobacteriaceae

- In the 1970s and 80s, AI researchers looked to knowledge as a way to combat both the computation and information limitations of the previous era.
- At this time, expert systems became fashionable, where a domain expert would encode their domain expertise in these systems, usually in the form of if-then rules.

Knowledge-based systems (70-80s)



DENDRAL: infer molecular structure from mass spectrometry

MYCIN: diagnose blood infections, recommend antibiotics

XCON: convert customer orders into parts specification

- There was also a noticeable shift in focus. Instead of the solve-it-all optimism from the 1950s and 60s, researchers focused on building narrow practical systems in targeted domains.
- Famous examples from this era included systems for chemistry, medical diagnosis, and business operations.

Knowledge-based systems

Wins:

- Knowledge helped both the **information** and **computation** gap
- First **real application** that impacted industry

Problems:

- Deterministic rules couldn't handle the **uncertainty** of the real world
- Rules quickly became too **complex** to create and maintain

A number of people have suggested to me that large programs like the SHRDLU program for understanding natural language represent a kind of dead end in AI programming. Complex interactions between its components give the program much of its power, but at the same time they present a formidable obstacle to understanding and extending it. In order to grasp any part, it is necessary to understand how it fits with other parts, presents a dense mass, with no easy footholds. Even having written the program, I find it near the limit of what I can keep in mind at once. — Terry Winograd

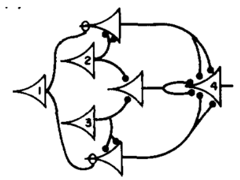
1987: Collapse of Lisp machines and second AI winter

- What knowledge (in addition to the restriction to narrow domains) did was not only providing information to the system, but it also helped alleviate the need for as much computation, by placing constraints on the space of possibilities.
- Also, this was the first time AI had a real impact on industry, rather than being just an academic's playground.
- However, knowledge engineering ran into major limitations. First, deterministic rules failed to capture the uncertainty in the real world, thought there were attempts to patch this heuristically as an afterthought.
- Second, these systems were just too much work to create and maintain, making it hard to scale up to more complex problems.
- Terry Winograd built a famous dialogue system called SHRDLU summed up well by the sentiment in this quote: the complex interactions between all the components made it too hard for mortals to even grasp. After that, he moved to Stanford and became an HCI professor.
- During the 80s, there was again a lot of overpromising and underdelivering, the field collapsed again. It seemed like history was repeating itself.
- We will now leave the story of symbolic AI, which dominated AI for multiple decades...

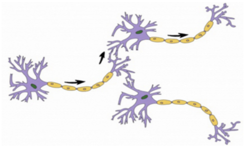
1943

- ...and go back in time to 1943 to tell the story of neural AI.

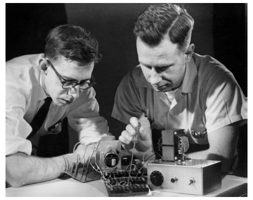
Artificial neural networks



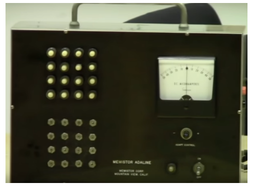
1943: artificial neural networks, relate neural circuitry and mathematical logic (McCulloch/Pitts)



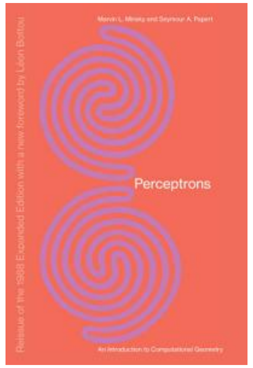
1949: "cells that fire together wire together" learning rule (Hebb)



1958: Perceptron algorithm for linear classifiers (Rosenblatt)



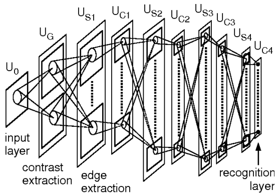
1959: ADALINE device for linear regression (Widrow/Hoff)



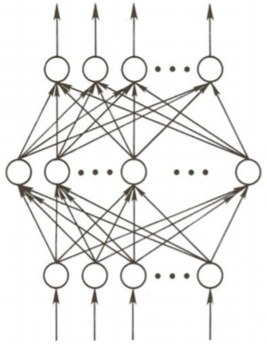
1969: Perceptrons book showed that linear models could not solve XOR, killed neural nets research (Minsky/Papert)

- In 1943, neurophysiologist Warren McCulloch and logician Walter Pitts devised a simple mathematical model of a neuron, giving birth to the field of (artificial) neural networks.
- They showed how this model could compute arbitrary logical functions (and, or, not, etc.), but did not suggest a method for learning this model.
- In 1949, neuropsychologist Donald Hebb introduced the first learning rule. It was based on the intuition that cells that fire together wire together. This rule was nice in that it was local, but it was unstable and so didn't really work.
- In 1958, Frank Rosenblatt developed the Perceptron algorithm for learning single-layer networks (a.k.a. linear classifiers), and built a device that could recognize simple images.
- In 1959, Bernard Widrow and Ted Hoff came up with ADALINE, a different learning rule corresponding to linear regression. A multi-layer generalization called MADALINE was used later to eliminate echo on phone lines, one of the first real-world applications of neural networks.
- 1969 was an important year. Marvin Minsky and Seymour Papert published a book that explored various mathematical properties of Perceptrons. One of the (trivial) results was that the single-layer version could not represent the XOR function. Even though this says nothing about the capabilities of deeper networks, the book is largely credited with the demise of neural networks research, and the continued rise of symbolic AI.

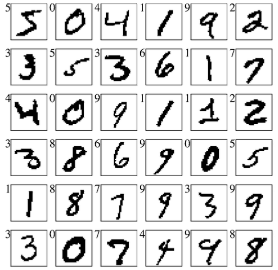
Revival of connectionism



1980: Neocognitron, a.k.a. convolutional neural networks for images (Fukushima)



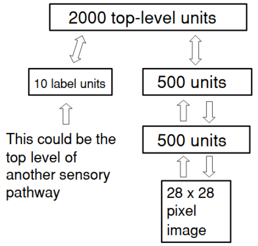
1986: popularization of backpropagation for training multi-layer networks (Rumelhardt, Hinton, Williams)



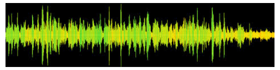
1989: applied convolutional neural networks to recognizing handwritten digits for USPS (LeCun)

- In the 1980s, there was a renewed interest in neural networks under the banner of connectionism, and there were many new links to psychology and cognitive science.
- The Neocognitron developed by Kunihiko Fukushima was the first convolutional neural network, with multiple layers and pooling. It was trained in a rather heuristic way.
- Donald Rumelhardt, Geoff Hinton, and Ronald Williams rediscovered (yet again) and popularized backpropagation as a way to train multi-layer neural networks, and showed that the hidden units could capture interesting representations.
- Yann LeCun built a system based on convolutional neural networks to recognize handwritten digits. This was deployed by the USPS to recognize zip codes, one of the early success stories of neural networks.

Deep learning



2006: unsupervised layerwise pre-training of deep networks (Hinton et al.)



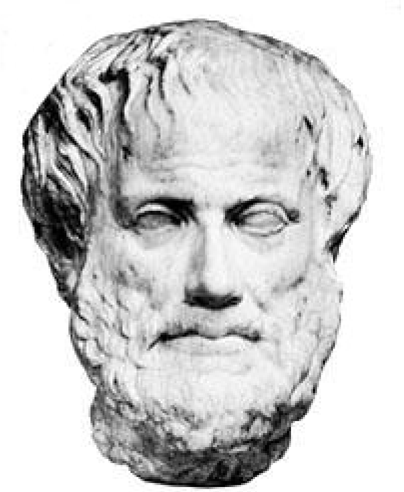
2009: neural networks outperform Hidden Markov Models in speech recognition, transformed speech community

2012: AlexNet obtains huge gains in object recognition; transformed computer vision community

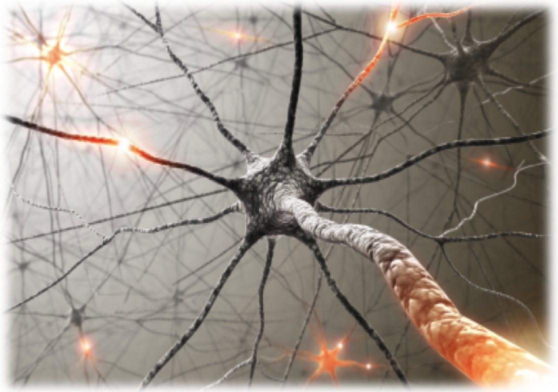
2016: AlphaGo uses deep reinforcement learning, defeat world champion Lee Sedol in Go

- But until the mid-2000s, neural network research was still quite niche, and they were still notoriously hard to train. In 2006, this started changing when Geoff Hinton and colleagues published a paper showing how deep networks could be trained in an unsupervised manner, and then fine-tuned on a small amount of labeled data. The term deep learning started around this time. This "pre-training" technique is ubiquitous today.
- The real break for neural networks came around the turn of the decade.
- From 2009, researchres at University of Toronto, Microsoft, Google, and IBM, developed deep learning approaches (e.g., deep belief networks) that significantly outperformed traditional Hidden Markov Models (HMMs) on speech recognition tasks (e.g., phone recognition on the TIMIT dataset). Soon thereafter, neural approaches became the dominant paradigm in industry.
- In 2012, Alex Krizhvesky, Ilya Sustkever, and Geoff Hinton trained a landmark convolutional neural network called AlexNet, which resulted in massive improvements on the ImageNet benchmark, turning the skeptical computer vision community into believers almost instantaneously.
- In 2016, DeepMind's AlphaGo was another turning point. By defeating humans at Go, a feat that many experts thought was still a few decades away, deep learning firmly established itself as the dominant paradigm in AI.

Two intellectual traditions



symbolic AI



neural AI

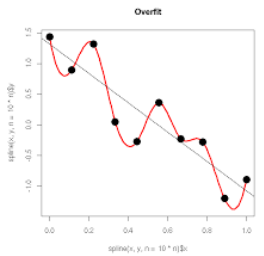
Food for thought: deep philosphical differences, but deeper connections (McCulloch/Pitts, AlphaGo)?

- So far, we've seen two intellectual traditions, symbolic AI, with roots in logic and neural AI, with roots in neuroscience.
- While the two have fought fiercely over deep philosophical differences, perhaps there are deeper connections.
- For example, McCulloch and Pitts' work from 1943 can be viewed as the root of deep learning, but that paper is mostly about how to implement logical operations.
- The game of Go can be perfectly characterized by a set of simple logic rules. But AlphaGo did not tackle the problem directly using logic and instead leveraged the pattern matching capabilities of artificial neural networks.

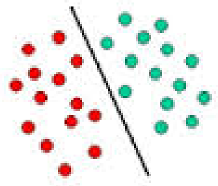
1801

- But there's a third and final story we must tell to complete the picture. This story is not really about AI per se, but rather the influx of certain other areas that have helped build a solid mathematical foundation for AI. This **statistical AI** (broadly construed) perspective is also how we will frame the topics in this course.

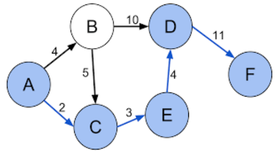
Early ideas from outside AI



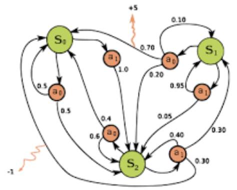
1801: linear regression (Gauss, Legendre)



1936: linear classification (Fisher)



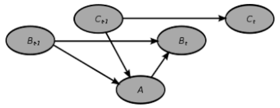
1956: Uniform cost search for shortest paths (Dijkstra)



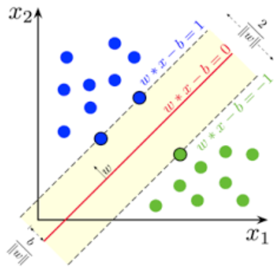
1957: Markov decision processes (Bellman)

- The idea of fitting models from data, which is at the heart of machine learning and modern AI, goes back to as far as Gauss and Legendre, who developed the principle of least squares for linear regression.
- Classification (linear discriminant analysis) was developed by Fisher in statistics.
- In general, machine learning has quite a bit of overlap with the statistics and data mining communities, who worked on solving concrete problems without the lofty goals of "intelligence".
- Besides machine learning, AI consists of sequential decision making problems. Along these lines, there's Dijkstra's algorithm for finding shortest paths for deterministic settings.
- Bellman developed Markov decision processes in the context of control theory, which handles uncertainty in the world.
- Note that these developments largely predated AI.

Statistical machine learning



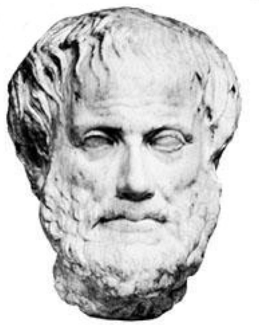
1985: Bayesian networks (Pearl)



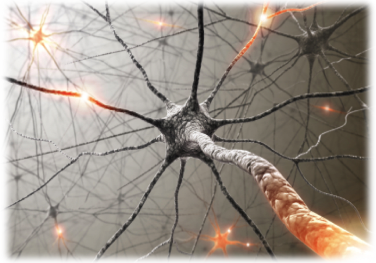
1995: Support vector machines (Cortes/Vapnik)

- You might have noticed that our story of symbolic AI ended at the end of the 1980s, but neural AI only became widespread in the 2010s.
- This is because for much of the 1990s and 2000s, the term AI wasn't actually used as much as it is today, partly to put distance between the most recent failed attempts in symbolic AI and partly because the goals were more down-to-earth.
- People talked about **machine learning** instead, and during that time period, machine learning was dominated by two paradigms.
- The first is Bayesian networks, developed by Judea Pearl, which provides an elegant framework for **reasoning under uncertainty**, something that symbolic AI didn't have a satisfying answer for.
- The second is Support Vector Machines (SVMs), which originated from statistical learning theory and optimization. SVMs were easier to tune than neural networks and became the favored tool in machine learning.

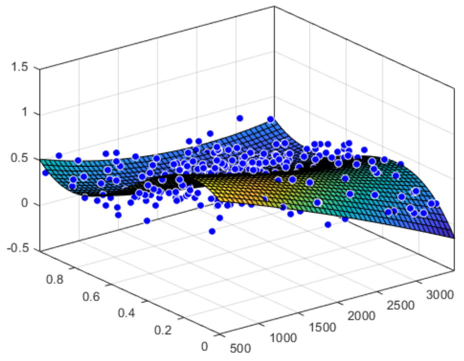
Three intellectual traditions



symbolic AI



neural AI



statistical AI

- This concludes our tour of the three stories that make up what AI is today.
- **Symbolic AI** took a top-down approach and failed to fulfill its original promise. But it offered a vision and did built impressive artifacts for ambitious problems like question answering and dialogue systems along the way.
- **Neural AI** took a completely different approach, proceeding bottom-up, starting with simple perceptual tasks, which the symbolic AI community wasn't interested in. It offered a class of models, deep neural networks, which with today's data and computing resources, has proven capable of conquering ambitious problems.
- Finally, **statistical AI** foremost offers mathematical rigor and clarity. For example, we define an objective function separate from the optimization algorithm, or have a language to talk about model complexity in learning. This course will be largely presented through the lens of statistical AI.
- Stepping back, the modern world of AI is like New York City—it is a melting pot that has drawn from many different fields ranging from statistics, algorithms, neuroscience, optimization, economics, etc. And it is the symbiosis between these fields and their application to important real-world problems that makes working in the field of AI so rewarding.

Further reading

Wikipedia article: https://en.wikipedia.org/wiki/History_of_artificial_intelligence

Encyclopedia of Philosophy article: https://plato.stanford.edu/entries/artificial-intelligence

Turing's Computing Machinery and Intelligence: https://www.csee.umbc.edu/courses/471/papers/turing.pdf

History and Philosophy of Neural Networks: https://research.gold.ac.uk/10846/1/Bishop-2014.pdf