Welcome to CS63!

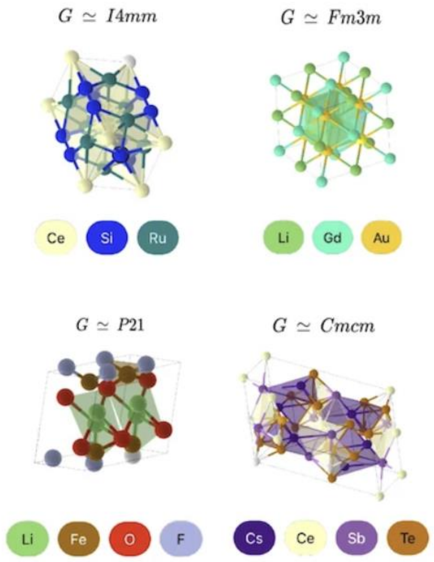
Prof. Gabe Hope

About me!

- 2^{nd} Year at Swarthmore (1^{st} time teaching AI)
- Before Swat I lived in California for 8 Years
  - 2 Years at Harvey Mudd
  - 6 Years at UC Irvine for my PhD
  - But originally from New York!
- A selection of things I love:
  - Cooking Italian food
  - Social deduction games
  - Indie folk music
  - Travelling to new places
  - Table (and regular) tennis
  - Crossword puzzles
  - Science Youtube
  - Working with Swat students!

What do I work on?

- Probabilistic Inference
    - How can we *reconstruct* corrupted data such as images or text?
- Gen AI for Scientific Applications
    - Can we *generate* new materials with desired properties?
    - Can we *learn* to model the dynamics of the climate?
- Semi-supervised learning
    - Can we learn to characterize insect behavior with only a few examples?



Course overview

Schedule

Course Information

Schedule

Class:	Monday, Wednesday, 9:00-10:15am, Science Center 204
Lab A (Hope):	Tuesday 1:05-2:35pm, Martin 113
Lab B (Hope):	Tuesday 2:45-4:15pm, Martin 113
Lab C (Soni):	Tuesday 2:45-4:15pm, Martin 313

Welcome back to teaching Prof. Soni!

Weekly rhythm

Schedule

WEEK	DAY	TOPICS	READINGS	LAB	EXTRAS
1	Aug 31	Intro to AI **[S26]**	• Roots of Artificial Intelligence **Mitchell Ch. 1** • **Reading questions**	Lab 1: **Robot Maze** **Due Sep. 9 at 11:59 p.m.**	Optional Lab 0: **Python Refresher/Warmup** Optional: **Quick, Draw! AI game**
	Sep 02	Agents, Environments, State-based Search **[S26]**	• Problem solving **R&N 3.1-3.3**		Optional: **The Fastest Maze-Solving Competition On Earth**
2	Sep 07	No class on Labor Day		Lab 2: **Informed search** **Due Sep. 14 at 11:59 p.m.**	
	Sep 09	Uninformed Search **[S26]**, Informed Search **[S26]**	• Uninformed search **R&N 3.4** • Informed search **R&N 3.5-3.7** • **Reading questions**		Optional: **Google Maps is unreasonably fast. Let me explain** **A* pathfinding visualizer**
3	Sep 14	Local Search **[S26]**	• Hill climbing, Simulated annealing, and Beam search **R&N 4.1** • **Reading questions**	Lab 3: **Local search** **Due Sep. 21 at 11:59 p.m.**	Optional: **Simulated annealing demo**
	Sep 16	Game Search **[S26]**	• **Minimax**		

- Complete readings ***before*** class
- Labs:
  - Assigned on Tuesdays at **noon** (work in lab sections)
  - Due on **Mondays at midnight**
  - More on labs tomorrow!
- Extras:
  - Optional, but fun!

Assessment

Course grade breakdown

Weight	Coursework
10%	Class Participation
45%	Exams
25%	Labs
20%	Final Project

Class participation

How participation is calculated

50%	**Attendance and punctuality** Regular attendance in lectures and labs, with on-time arrival.
50%	**In-class quizzes** Responses to reading quizzes and other in-class questions.
+10%	**Ed bonus** Earn up to 10% bonus credit by answering questions on Ed. (Up to 100% of the participation grade)

Attendance and participation will be managed using iClickers!

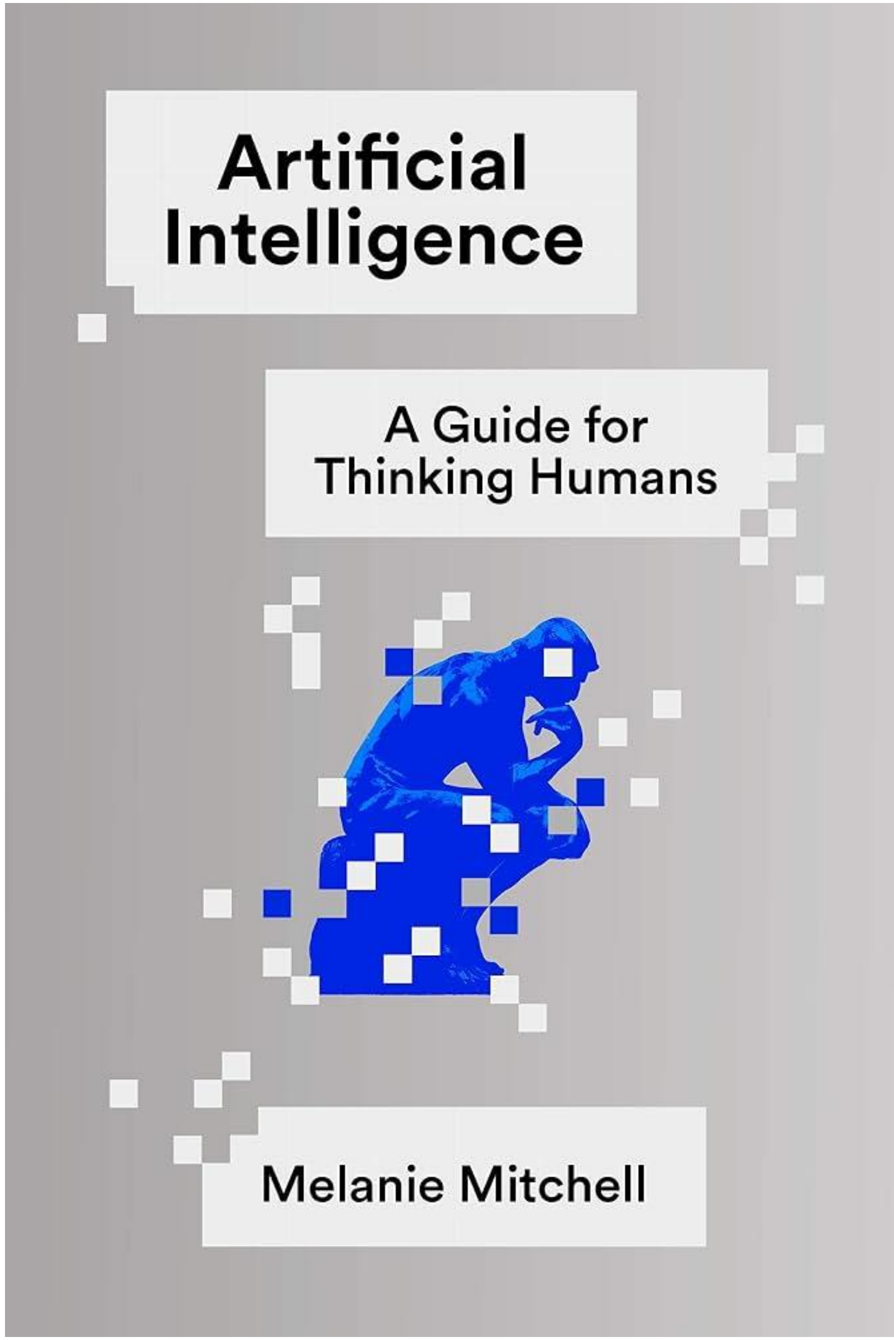
- Starting next week (September 9th), you **must obtain an iClicker by then**
  - Instructions are on the course website
  - Please speak to me if you have trouble obtaining one
- Quizzes will also use iClickers
  - Will consist of 2-4 multiple choice questions
    - Full credit will be given for at most 1 wrong answer
  - Should ***not*** require preparation beyond doing the readings (and paying attention in class).

iClicker+

Available at the College bookstore

Readings

- Most readings are provided on the calendar
- Some will require the Mitchell book
  - Available at the bookstore!
- Recommend following the reading advice on the course website!
- **Reading questions**
  - Many readings will have associated questions
  - You do **not** need to submit answers, but **should** be prepared to discuss them in class



Exams

- 2 midterm exams – equally weighted
  - October 26th
  - November 24th
- **No final exam**
  - Finals week will focus on the course project

Final project

- Open ended, your opportunity to explore a subject of your choice within AI
- More information closer to release date
  - But keep in mind what you might want to do as we go!

Please read through the entire course syllabus carefully!

This week

- **Today:** Introduction to the field of AI
- **Lab 1**
  - Introduction to Jupyter and the AI programming toolkit
- **Wednesday:** Introduction to state-based search
  - **Before class:**
    - Remember to do the assigned readings!
    - Submit the course survey!
- **Thursday or Friday:** Optional Python, NumPy and MatPlotLib review

So, what is AI?

Discuss with 2-3 neighbors

- Name 3-4 different tools or applications that you think fall under the category of *Artificial Intelligence*
  - What do you expect to learn in this course?
  - Try to think beyond LLMs!

- Given your answers to first question, try to come up with a definition of artificial intelligence
  - How would you describe AI to someone from 100 years ago?

- Now take a step back, how you you define *intelligence*?
  - Is it how we think, how we behave?
  - What could a computer or a robot do that would make you think it was intelligent?

Let’s work backwards!

What makes something intelligent?

Where does human intelligence come from?

- Nature vs. Nurture debate
  - Nature: Evolution
  - Nurture: Learning and Culture
- Both contribute to human intelligence

If humans are intelligent, what does it mean to build *artificial intelligence*

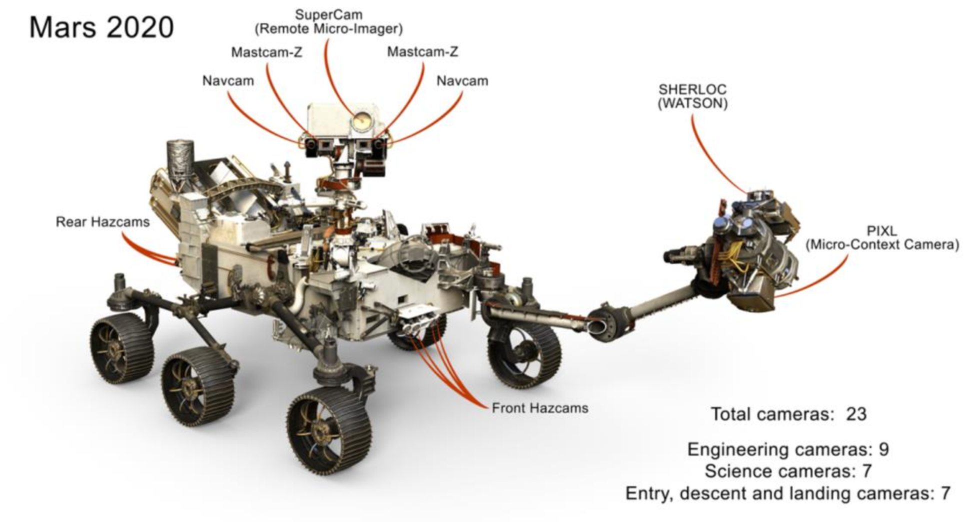
- Some possible approaches:

Think like a human	**Think rationally**
• Cognitive science / neuroscience • Does intelligence require biology?	• Logic and automated reasoning • But, not all problems can be solved just by these techniques
Act like a human • Turing test • ELISA, Loebner prize • “What is 1228 x 5873”? … “I don’t know, I’m just a human	**Act rationally** • Basis for intelligent agents framework • Unclear if this captures the current scope of AI research

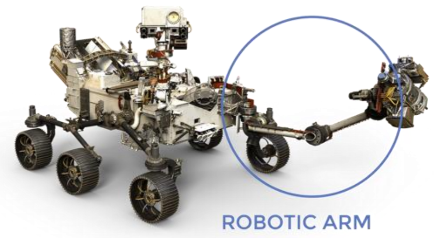
A working definition for us

The creation and study of computational agents that <u>act</u> intelligently

- An ***agent*** is an entity that perceives and acts in an environment
- Examples include: bacteria, spiders, cats, thermostats, robots and people
- Not agents: rocks, the wind



Sensors to perceive environment



Wheels and arms to act in environment

Features of intelligent action

- Appropriate for circumstances and goals
- Flexible to changing environments
- Able to learn from experience
- Makes reasonable choices given limited information and time to decide
  - *Anytime* algorithm

Note: there isn't really a hard line between the study of "AI" and the study of "algorithms", but these features will inform our focus!

Goal of AI

- To understand the principles that make intelligent behavior possible
- Work toward this goal by designing, building and testing computational agents that perform tasks we view as requiring intelligence

Why this goal?

- Notice that the focus is on external behavior – how an agent acts – not on internal thought
- External behavior is more easily measured and analyzed

Act

Think

Two main approaches to AI

Cognitive science

- Try to better understand natural intelligence by building computational agents
- *Act like a human*

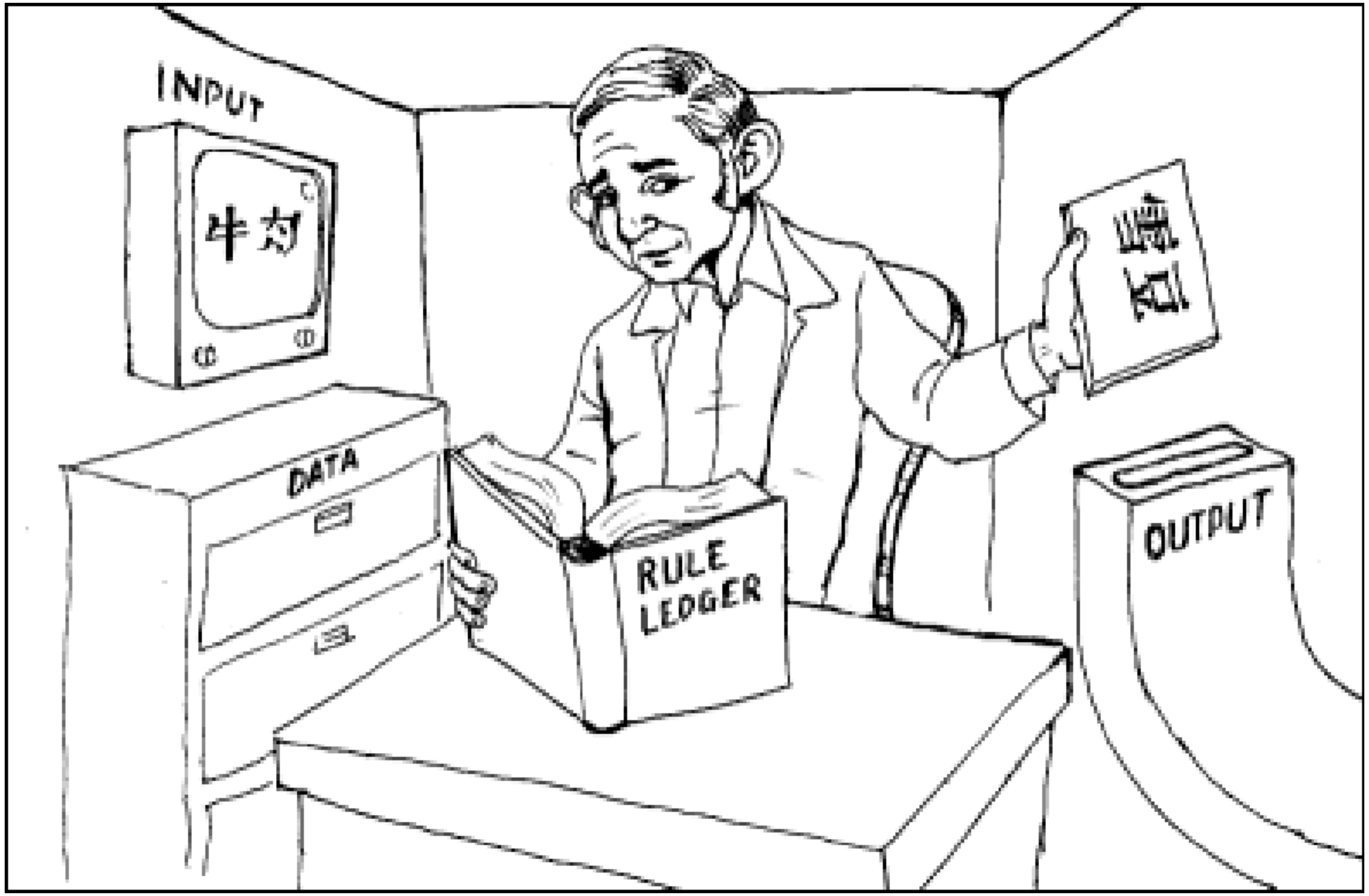
CS/Engineering

- Build the best possible machine intelligence without worrying about whether it mimics natural intelligence
- *Act rationally*

Can intelligence be faked?

A famous thought experiment: *The "Chinese room" argument*

- Person who knows English but not Chinese sits in a room
- Receives notes in Chinese
- Has a systematic rule book for how to write Chinese characters in response to input Chinese characters, uses this to return notes from the room



Is this intelligence?

External Chinese speaker

- *Yes! I'm getting perfectly reasonable responses from the room*

- Poole and Mackworth, authors of an AI textbook claim that if an agent behaves intelligently, it *is* intelligent
- AI creates real intelligence through artificial means

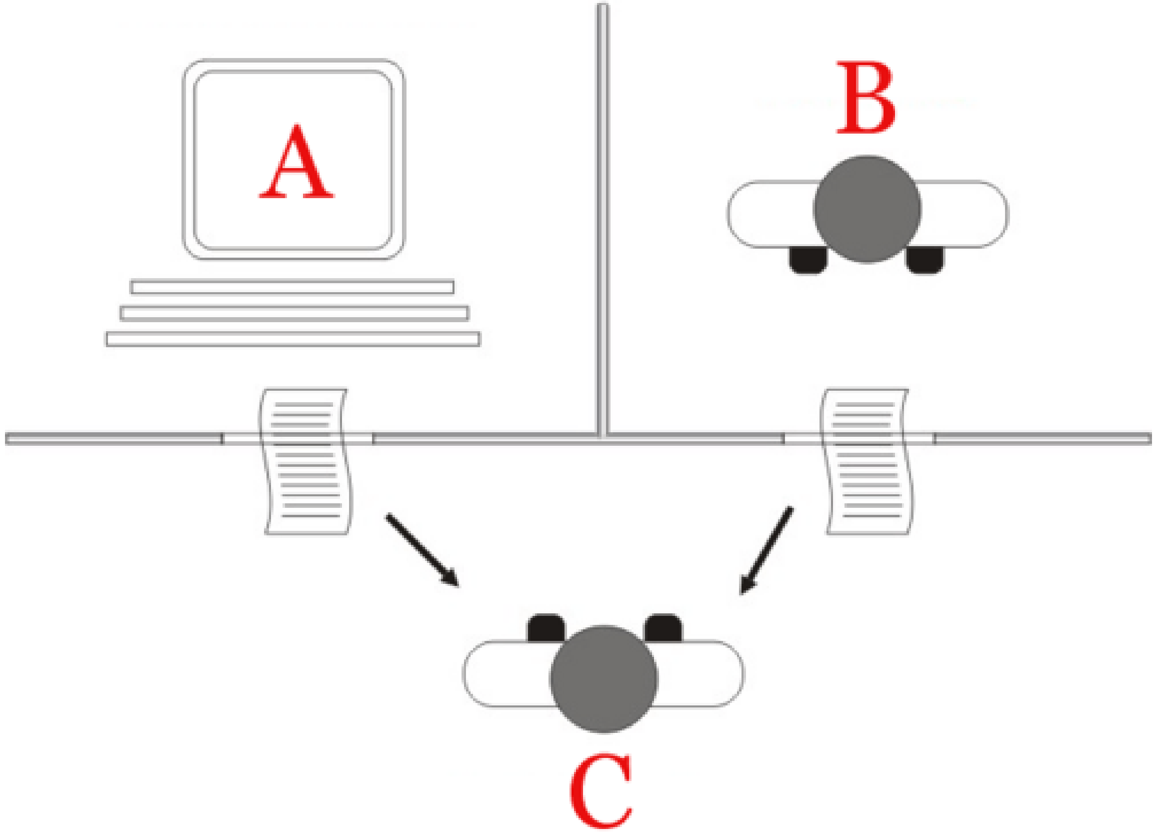
Internal English speaker

- *I have absolutely no idea what's going on…*

- Searle, who introduced this thought experiment claims it is not

Alan Turing's test: *The imitation game*

- In 1950 Alan Turing wrote a paper entitled *Computing Machinery and Intelligence* which argued that intelligence is defined by external behavior
- He invented a game to test whether an unseen agent was intelligent
- Helps us judge whether we have an intelligent agent, but doesn't tell us how to create one!



The observer (C) must guess which of the two agent's they're conversing with is a computer. The computer "wins" if C guesses incorrectly

Defining the field

AI is a relatively young field

First use of the term, was in a 1956 proposal:

“We propose that a two-month, ten man study of artificial intelligence be carried out in the summer of 1956 at Dartmouth College...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.”

Attendees at this workshop were from IBM, Stanford, MIT, and CMU, all still important centers of AI today.

Early coalescence into 2 philosophies

Top-down approach:

Symbolic AI

- Replicates conscious thought
- Data is *symbolic,* abstracts away irrelevant complexities of the world
- Algorithms use logic to find solutions

Bottom-up approach:

Sub-symbolic (neural) AI

- Replicates subconscious brain computation
- Data is numeric, contains signal and noise
- Uses learned algorithms that train from data

Introduction by example

Symbolic AI

Symbolic Example: General Problem Solver

- Invented by Cognitive scientists Herbert Simon and Allen Newell at CMU
- Had people talk out loud while they they solved logic puzzles
- Built an AI program to mimic the way humans solved these problems
- Example: the water pitcher problem

Water Pitcher Problem

You have two water pitchers:

- One holds 3 liters the other holds 5 liters
- The pitchers have no markings on them
- You can fill the pitchers to the top from a faucet
- You can pour the contents down the drain
- You can pour from one pitcher to the other
    - Until the one poured into is full
    - Or until the one being poured from is empty

- **Goal:** get a specific amount of water into each pitcher
- **Example:** get exactly 2 liters in the 3 liter pitcher

Start by coming up with a clear representation of the problem

- State: represents all key information about the problem
- Operators (or Moves): How one state can be legally transformed into another state

One state representation for the Water Pitcher problem

`(amountIn3L, amountIn5L)`

`Start state: (0, 0)`

`Goal state: (2, 0)`

Operators for Water Pitcher problem

Assume current state is: `(x, y)`

- `fill(pitcher)`
  - `fill(p3)` **new state is** `(3, y)`
  - `fill(p5)` **new state is** `(x, 5)`
- `drain(pitcher)`
  - `drain(p3)` **new state is** `(0, y)`
  - `drain(p5)` **new state is** `(x, 0)`
- `pour(pitcher1, pitcher2)`
  - `pour(p3,p5)`**, space in p5 is** `5-y`, `if x<=5-y`**, new state is** `(0,y+x)`**, else new state is** `(x-(5-y),y+x-(5-y))`
  - `pour(p5,p3)`**, space in p3 is** `3-x`**, similar to above**

If current state is `(3,4)`**, what does** `pour(p3, p5)` **do?**

If current state is `(1,5)`**, what does** `pour(p5, p3)` **do?**

Process of finding a solution

Start state: `(0,0)`

Goal state: `(2,0)`

- What are all the productive moves that we can make from the start state?
- A *productive* move is one that leads to a new state
- Then what are all the productive moves that we can make from those states?
- Continue this process until the goal is reached, or no new states exist

How the process begins

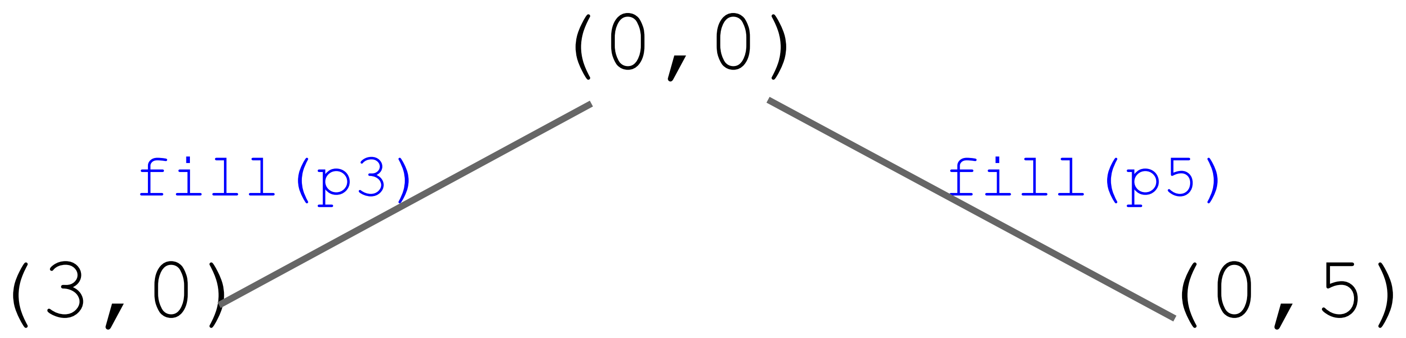
Start state: `(0,0)`

Possible moves:

- `fill(p3)`
- `fill(p5)`
- `drain(p3)`
- `drain(p5)`
- `pour(p3,p5)`
- `pour(p5,p3)`

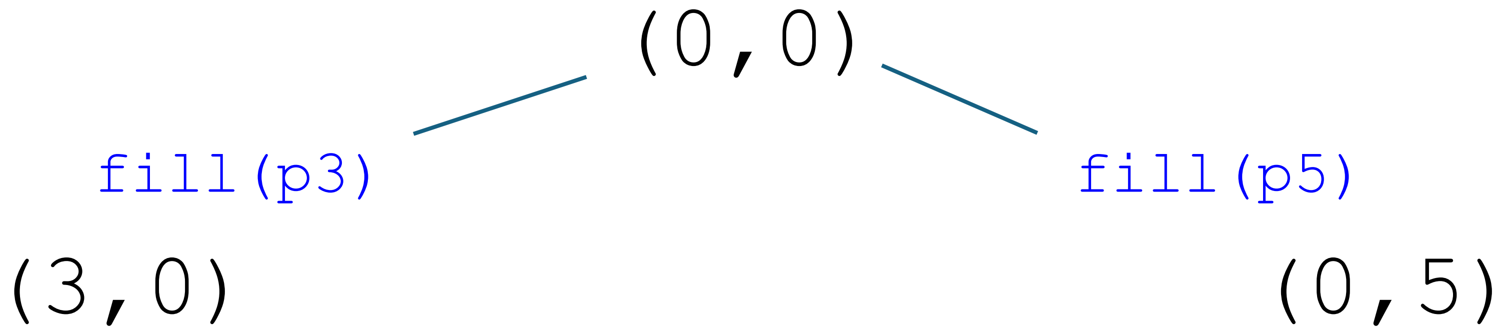
Which moves are productive?

How the process works



- We can represent this process as a search tree where the root is the start state
- Each node is a unique state
- Each branch represents the action we took to move to that state
- Once the goal state is reached, the actions along the path from the root to the goal state are the solution
- If all unique states have been visited and the goal state is not found, then no solution is possible

Is it possible to get exactly 2 liters in the 3 liter pitcher?



Is it possible to get exactly 1 liter in the 5 liter pitcher? Is so, what are the steps?

```
                    (0,0)
           /                  \
    fill(p3)                   fill(p5)
  (3,0)                              (0,5)
```

Introduction by example

Sub-Symbolic (Neural) AI

Subsymbolic example: Perceptron

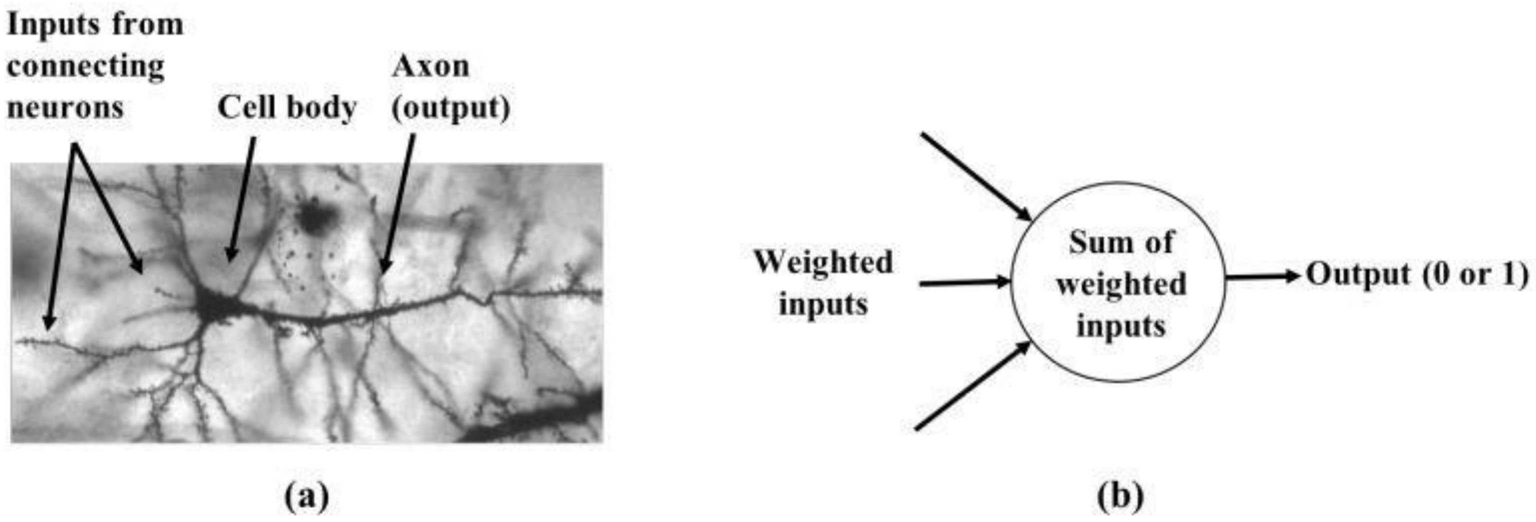


Figure 1: (a) A neuron in the brain.[16] (b) A simple perceptron.

Can we replicate a **useful** neuron?

Task: Recognize handwritten 8's

Figure 2: Examples of handwritten digits.[18]

Represent handwritten digits as a 1-D array of pixel values (0-255)

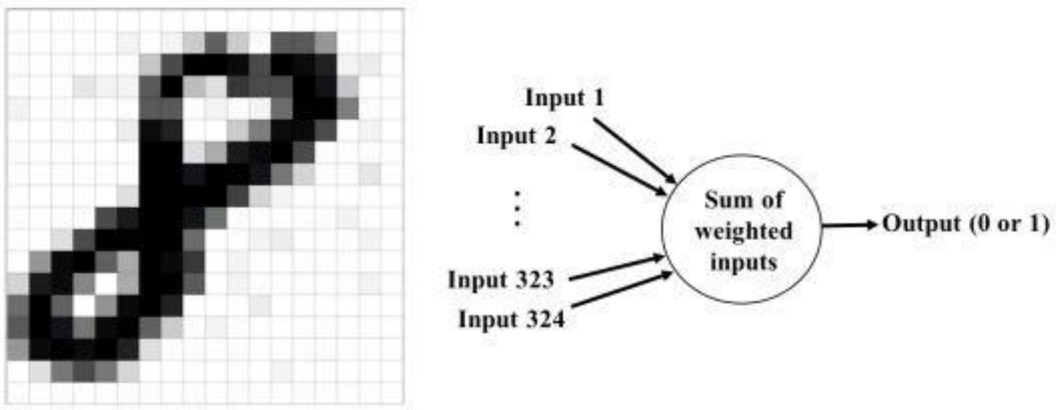
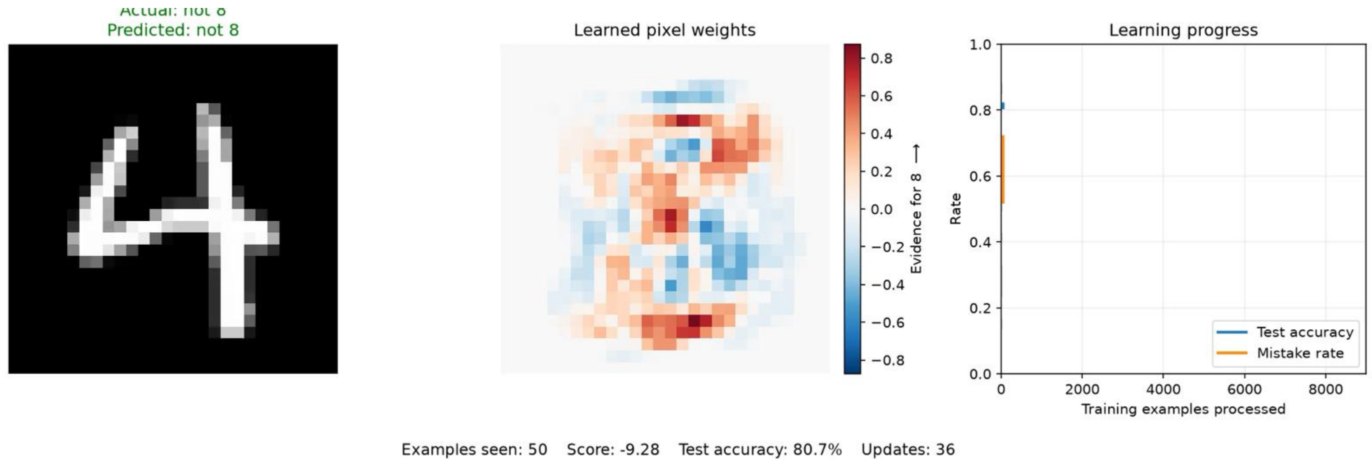


Figure 3: An illustration of a perception that recognizes handwritten ‘8’s.[19] Each pixel in the 18×18-pixel image corresponds to an input to the perceptron, yielding 324 (= 18 × 18) inputs.

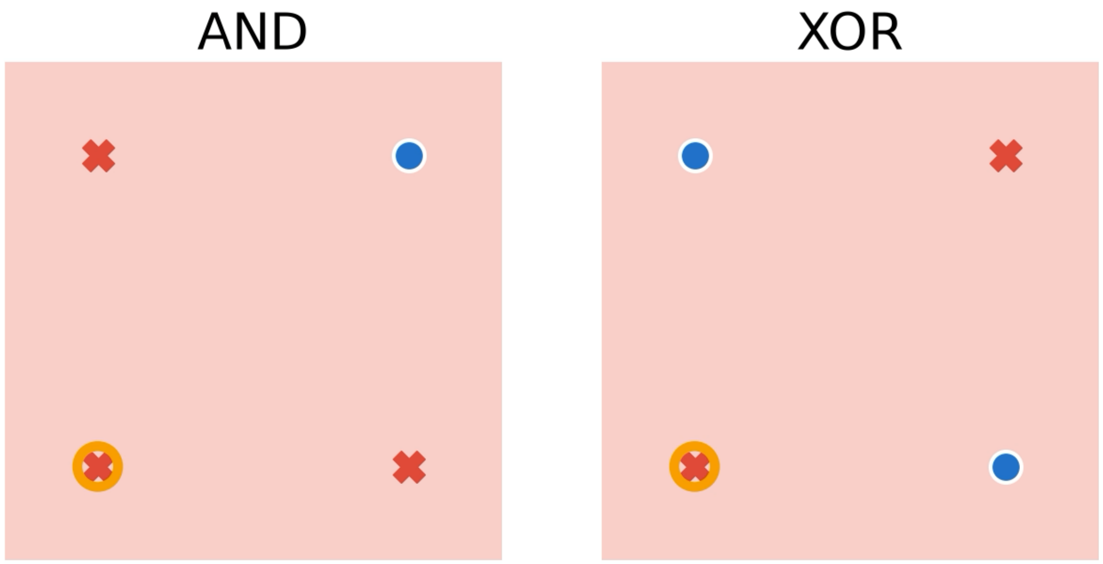
Perceptrons

- Start out with random weights
- Weights are adjusted during training
- For this problem they encode how likely it is that a particular pixel within an image part of a handwritten 8



Limitations of perceptrons

- Minsky and Papert proved that perceptrons could only solve linearly separable problems
- Perceptrons can solve logic problems like AND, OR, but not XOR

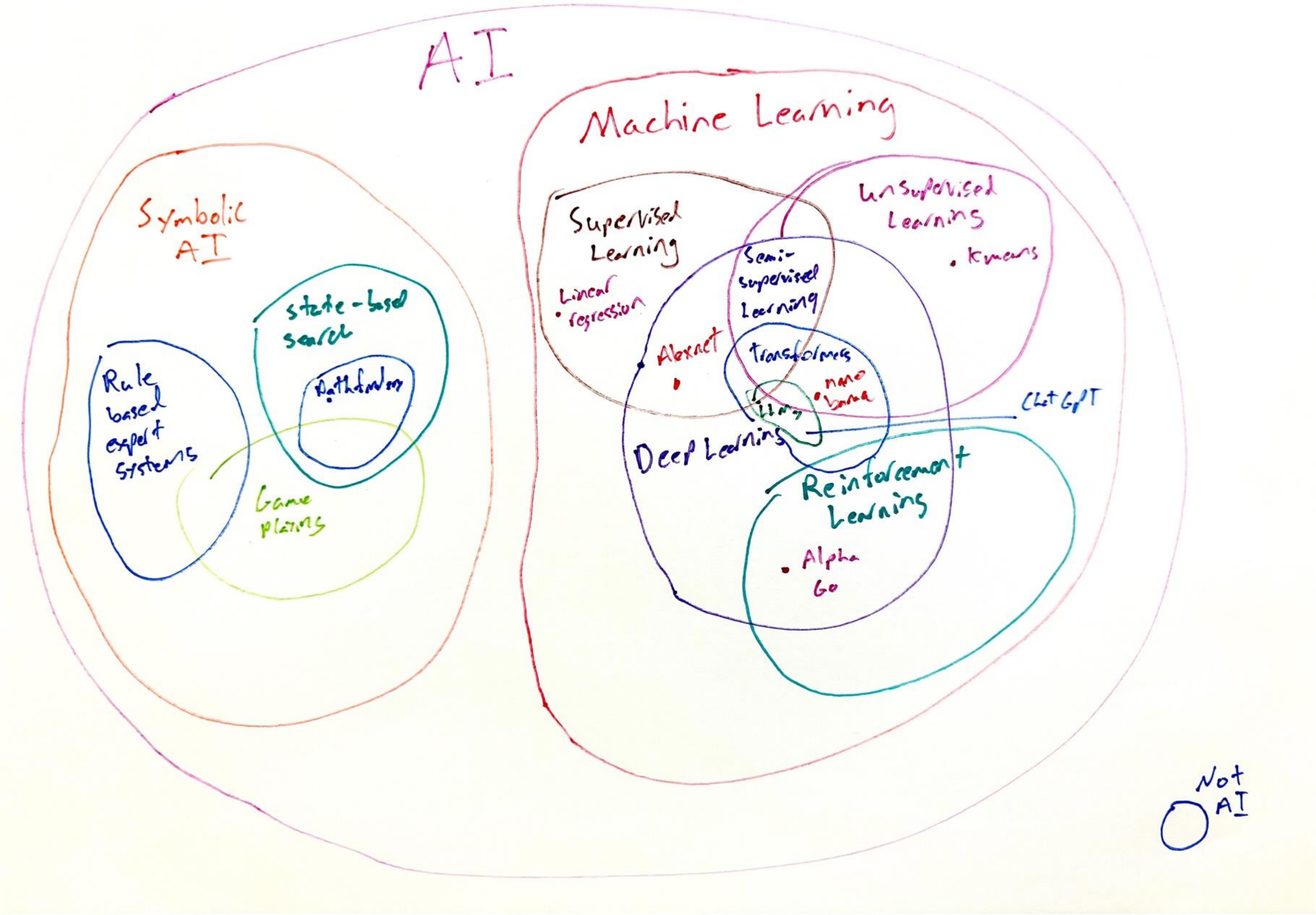


Limitations of perceptrons

- Minsky and Papert argued that though multiple layers of perceptrons would be more powerful, but no learning algorithm existed to train them
- This negative prognosis led to a decrease in subsymbolic AI work until the back propagation algorithm was discovered in the mid 1980s

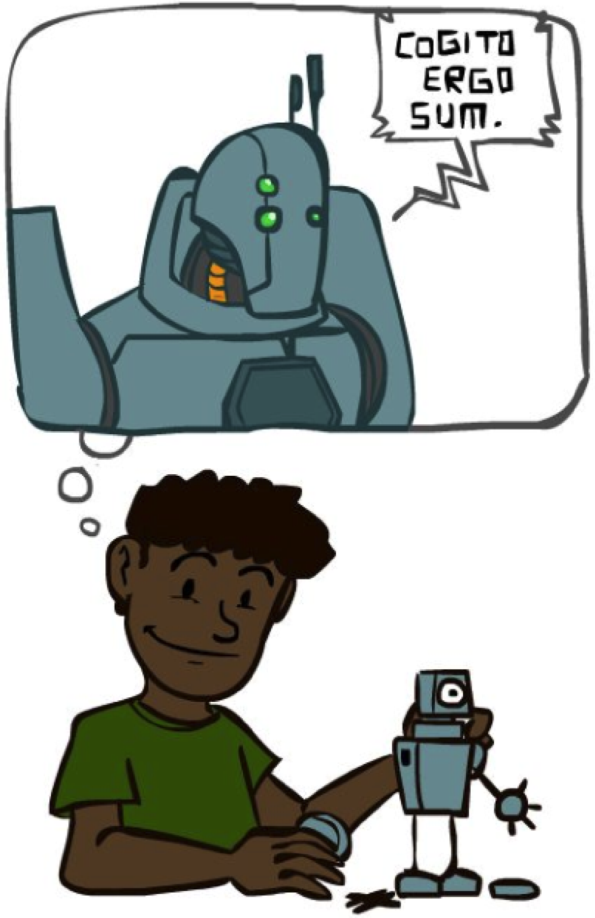
So, what is AI?

A very rough map of the field



AI post-Dartmouth

- 1956—70: Excitement: Look, Ma, no hands!
  - 1950s: Early AI programs: chess, checkers (RL), theorem proving
  - 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
  - 1985—90: Revival of neural nets
  - 1988—93: Expert systems industry busts: "AI Winter"
- 1990—2012: Statistical approaches + subfield expertise
  - Resurgence of probability, focus on uncertainty
  - General increase in technical depth
  - Agents and learning systems… "AI Spring"?
- 2012—2022: Big data, deep learning, generative models
  - Image recognition, speech recognition, machine translation
  - Language models, transformers, image generators
- 2022—present: LLMs…..
  - ChatGPT, scaling laws, LLM industry booms
  - Large reasoning models



Credit: Dan Klein and Stuart Russell

A lesson: General-purpose AI is much harder to achieve than anticipated

- AI research uncovered a paradox: “Easy things are hard” (Minsky)
- In other words, tasks that are easy for even children to do have proven hard to get computers to solve
  - Understanding natural language
  - Recognizing objects from images
  - Learning a new concept from a few examples
- While tasks we consider challenging, such as being a chess champion, have been easier for AI to achieve