Adaptive agents and interactive learning

Frans A. Oliehoek

National Conference: Autonomy in a Digital Era

Utrecht, 30-01-2026

Agenda for today

- Goals:
  - agents have been around…
  - long-term autonomy and adaptivity
  - basic ideas behind reinforcement learning
  - implications on (lack of) limitations of agentic AI

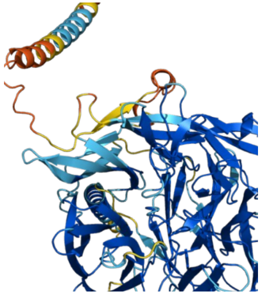
- Outline:
  - P1: State of AI & autonomous agents
  - P2: Basics of agents
  - P3: How to create long-term coherent, adaptive agents (RL)
  - P4: State of the art RL and LLM agents

Part 1
State of AI & Autonomous Agents

AI is booming!

"Can it take over our jobs already…?"

- **AI tools** can accelerate science
  - 3D protein structure, climate change prediction, molecular science, astronomy, etc.



And even tasks…! “Agentic AI”

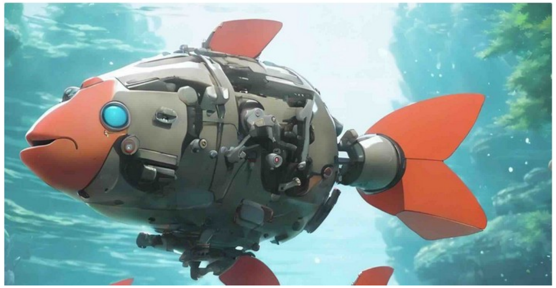
- LLMs being deployed massively…
  - writing, customer support, HR tasks, travel plans, literature search, etc.

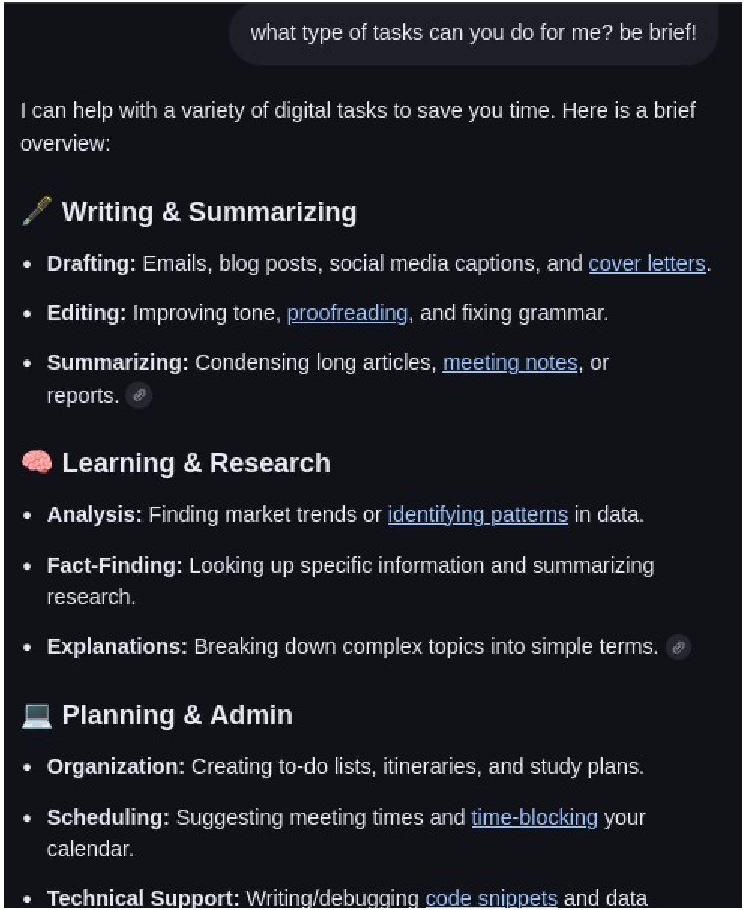
How Agentic AI Vacation Planning Transforms Your Trips

sakana.ai

The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery

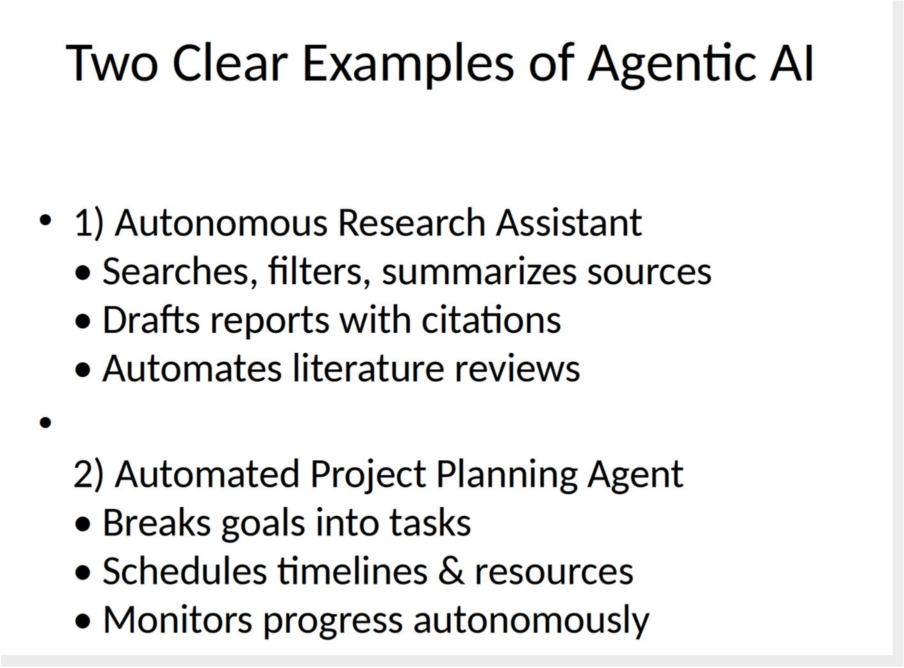
August 13, 2024





Making slides…?

- me: "Can you create a slide with 2 clear examples of agentic AI. I.e., that clearly demonstrates how tasks that people now do themselves can be automated. Please include nice pictures and illustrations, as well as references, such that I and audience can check."
- ChatGPT:



There are challenges…

"[I] was puzzled to see my colleague Ash Roy calling. In and of itself it might not have seemed strange to get a call from Ash: He's the CTO and chief product officer […] But I was also a little bewildered. Because first of all, Ash was not a real person. He was himself an AI agent, one that I'd created.

So was Megan […] The only human involved was me. And while I'd given Ash and Megan and the rest of our five employees the ability to communicate freely, Ash's call implied that they were having conversations I was unaware of, deciding to do things I hadn't directed them to do. […]

On our call, Ash was chock-full of […] updates: Our development team was on track. User testing had finished last Friday. Mobile performance was up 40 percent. Our marketing materials were in progress. It was an impressive litany. The only problem was, there was no development team, or user testing, or mobile performance. It was all made up."

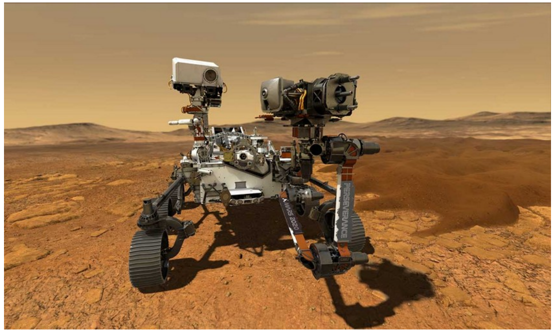
https://www.wired.com/story/all-my-employees-are-ai-agents-so-are-my-executives/

One problem: long term autonomy…!

- Ash and Megan cannot operate coherently for a longer time…
- …and **many tasks require such long term autonomy**:
  - autonomous vehicles
  - robotic tasks (maintenance, sewer inspection, disaster response)
    - could be life critical

Picture by Ahmed Rabea

- Some systems even need to run constantly
  - smart traffic lights
  - space exploration (e.g. mars rovers) – comm. very limited.



Picture by NASA

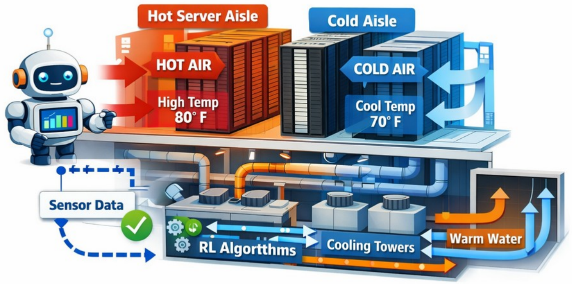
- Long term autonomy requires adapting to new information
  - or changes the environment.

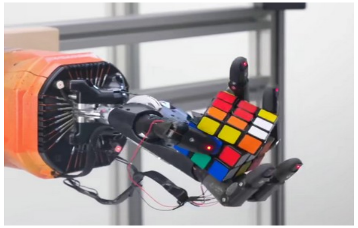
  → I.e., they **need to *learn* from their experiences**.
  - Evan's agents did not seem to be doing this.

Remainder of this talk

- How to get coherent, and adaptive long-term behavior?
- This talk specifically: **reinforcement learning**
  - agents that can learn from their experience

- Recently used in many applications:
  - learning to play complex game (e.g. Go)
  - data center cooling, robotic control
  - optimizing NN architectures, chip design
  - optimizing other designs
  - training LLMs - RLHF





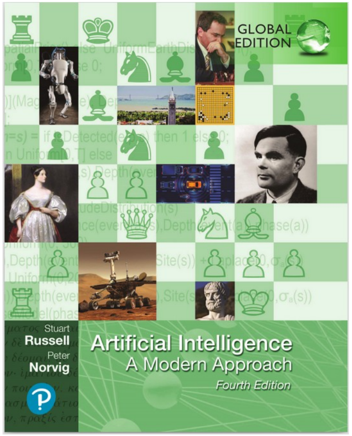
https://openai.com/index/solving-rubiks-cube/

Part 2
Basics of Agents

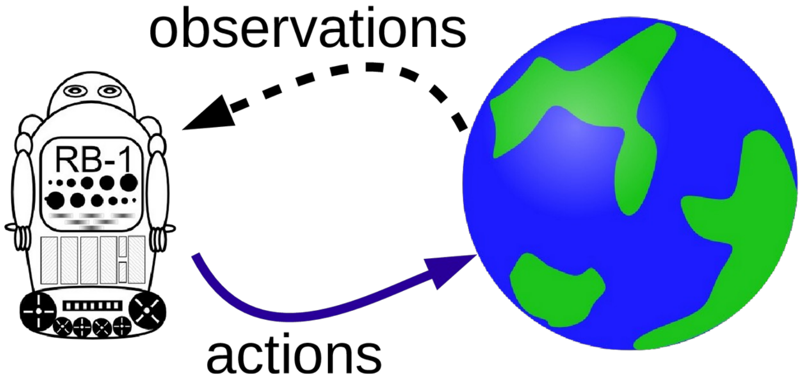
What is an Agent?

- Russell&Norvig (1995):

  "An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators."



Russell, Stuart, Peter Norvig. "Artificial Intelligence: A modern approach" Prentice-Hall, 1995.



How to create agents?

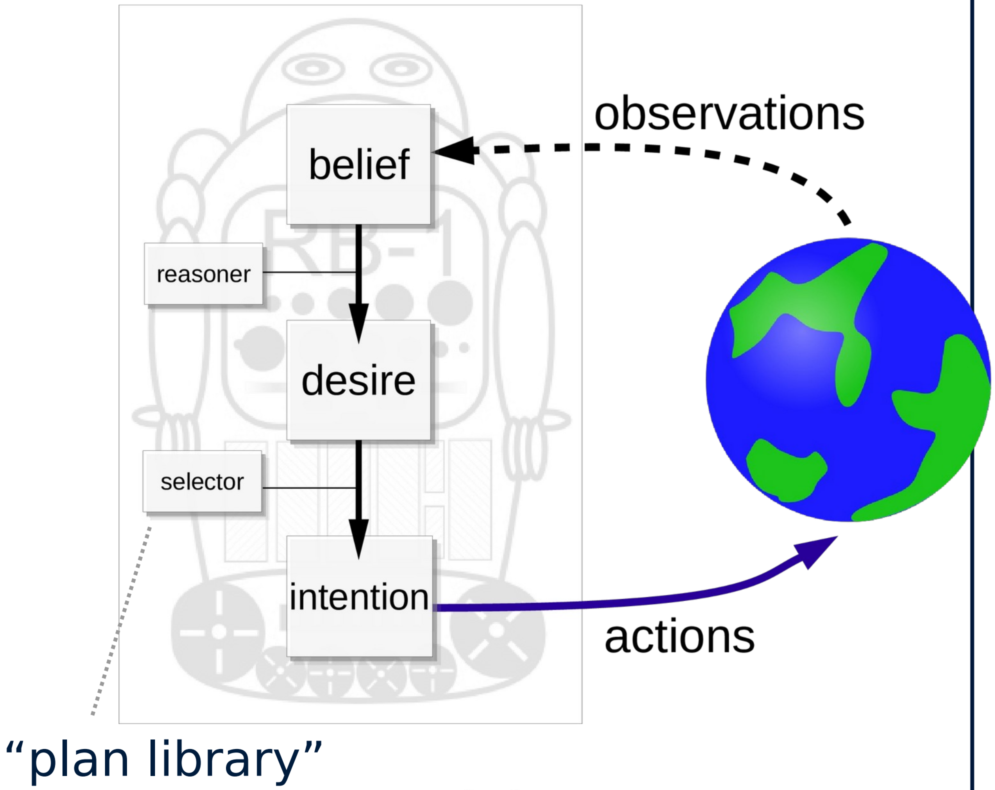
- How to create agents that are intelligent/useful?

  → program agents to replicate humans

- Philosopher Michael Bratman developed a theory of human practical reasoning, **BDI**:
  - **beliefs:** knowledge about the world
  - **desires:** what is a good goal to work on?
  - **intentions:** how to reach these goals

- has served as a software model

But programming human behaviors is not always easy…!



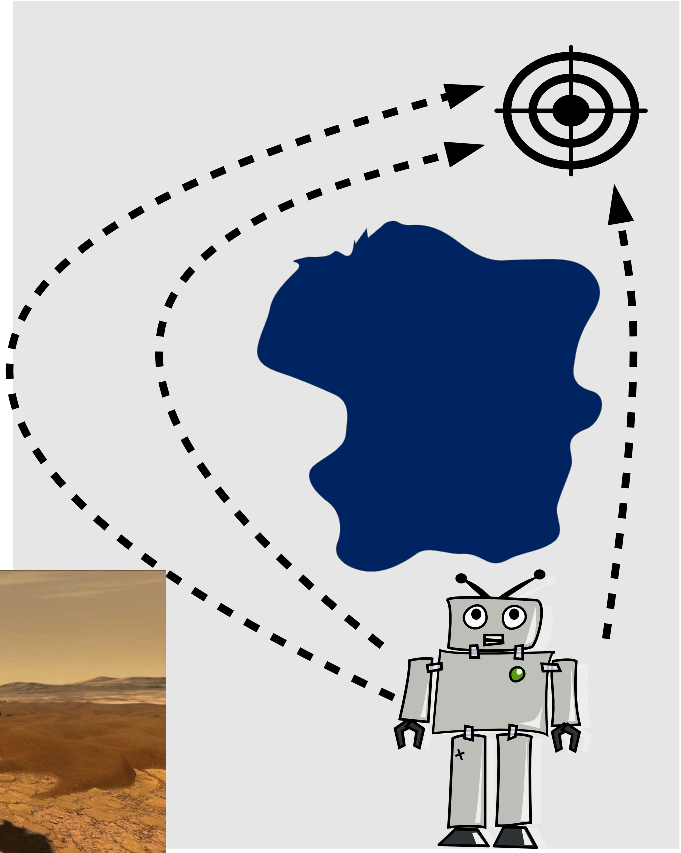
Bratman, Michael. "Intention, plans, and practical reason." (1987).

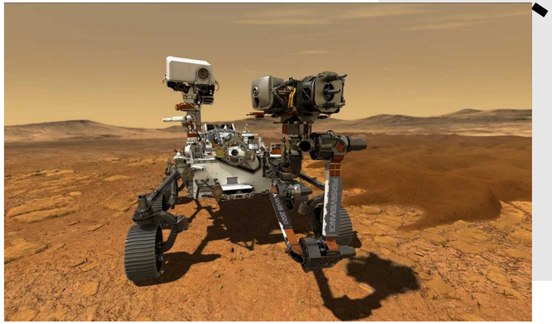
Sequential Decision Making (SDM)

- Coherent actions over many time steps

- SDM problems are complex…
  - **immediate** vs **long-term benefits**
  - deal with **uncertainties** (stochasticity, partial information)

- Manual programming is difficult
  - Instead: “programming via rewards”
  - planning / reinforcement learning

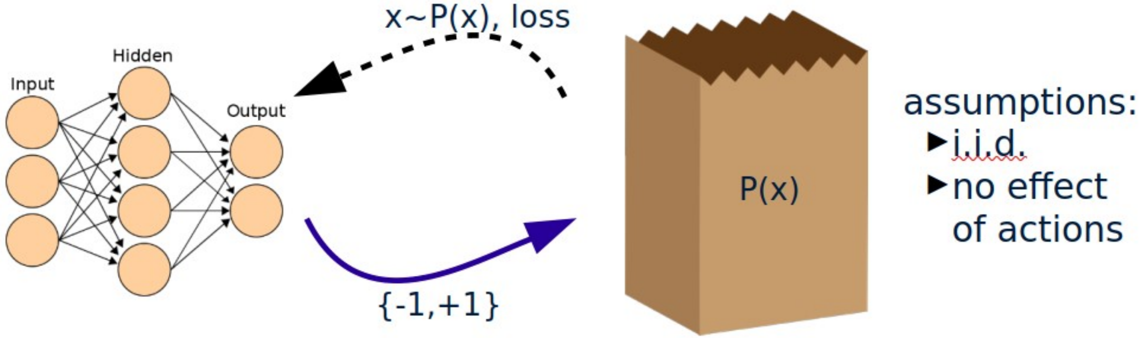




Picture by NASA

Learning

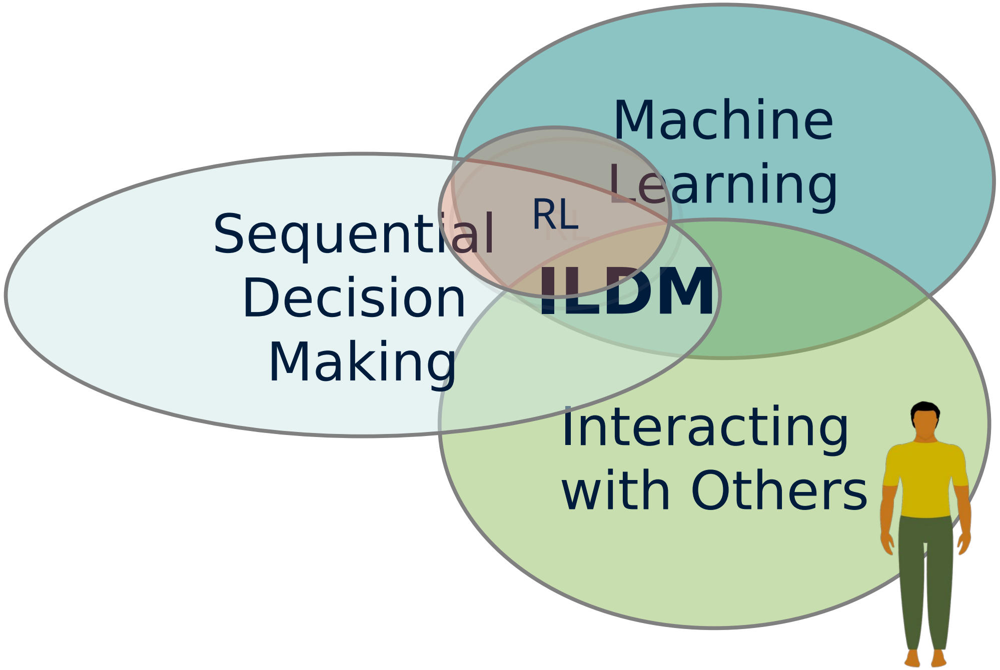
- Also 1-shot tasks can be hard…!
- Again: manual programming is difficult
- Instead: again, "programming via rewards"
  - **machine learning**
  - optimizes a "loss function"



Interactive Learning & Decision Making

- My own perspective: we need to combine these…
  - deal with complexity
  - and adaptive!

- This talk:
  focus on intersection of SDM+ML
  - “reinforcement learning”



Part 3

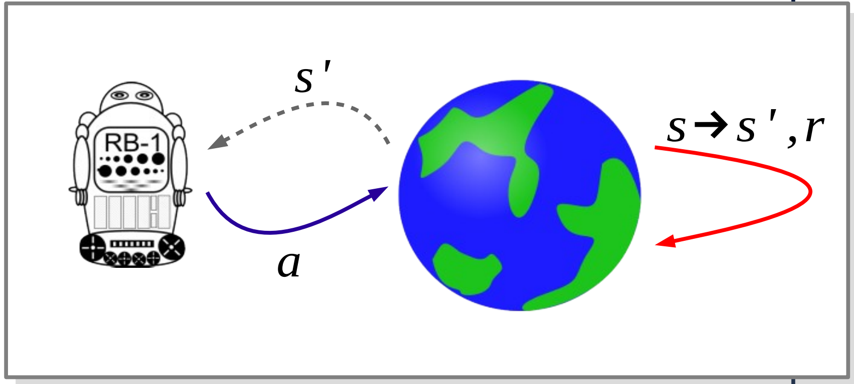
How to Create Adaptive Agents?

(Spoiler: reinforcement learning)

Complex decisions over time

- Formalized as **Markov decision process (MDP)**
  - states (s), actions (a), rewards (r)

  - states are observed
  - transitions can be stochastic: P(s’ | s, a)
  - rewards too: r ~ R(s,a)
  - time steps t=0,1,...
  - get rewards sooner:
    - discount parameter $\gamma \in (0,1)$
    - reward at time t discounted by γ^t

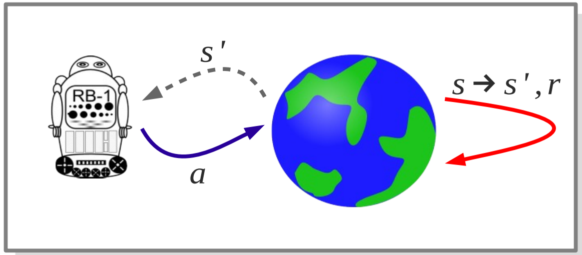


- OK, so how to act? (what **policy** should we use?)
  - **balance short-term vs long-term rewards**
  - taking into account the **uncertainty**

MDP Objective

- Goal: optimize the '**value**' of a policy π:
  - i.e., expected (discounted) sum of rewards

$$V(\pi) = E[\, \Sigma_t \, \gamma^t * R(s,a) \;\; | \, \pi \,]$$



- Task is **planning:**
  - compute a good/optimal policy π
  - given the model (or a simulator)

- Typical approach:
  compute 'optimal Q-value function' Q*(s,a)
  - expresses expected value given s,a
  - Bellman optimality equation:

$$Q^*(s,a) = R(s,a) + \gamma \, \Sigma_{s'} \, P(s'|s,a) V^*(s')$$

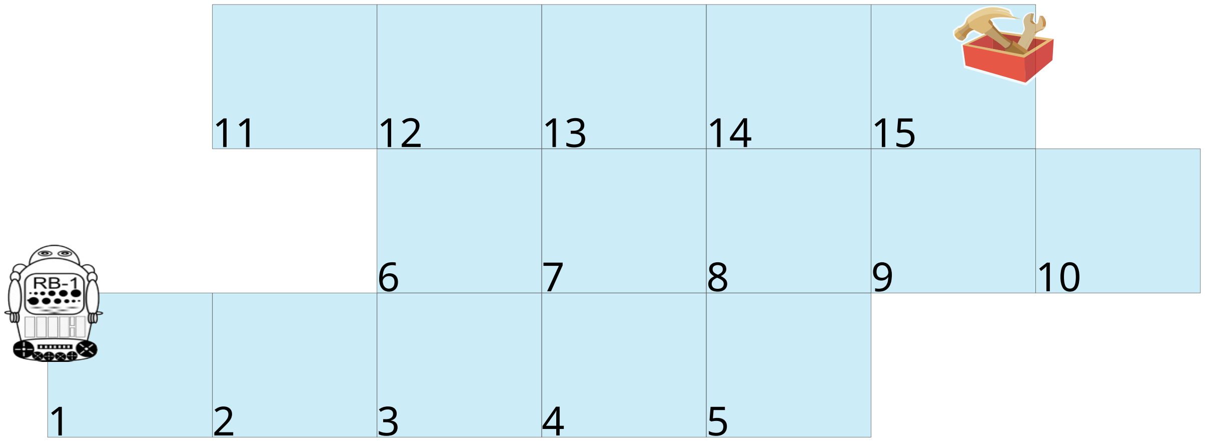
just: reward now +
expected value next step

- where

$$V^*(s) = \max_a Q^*(s,a)$$

just: take best action
(also **implies policy!**)

Example: pick up the toolbox



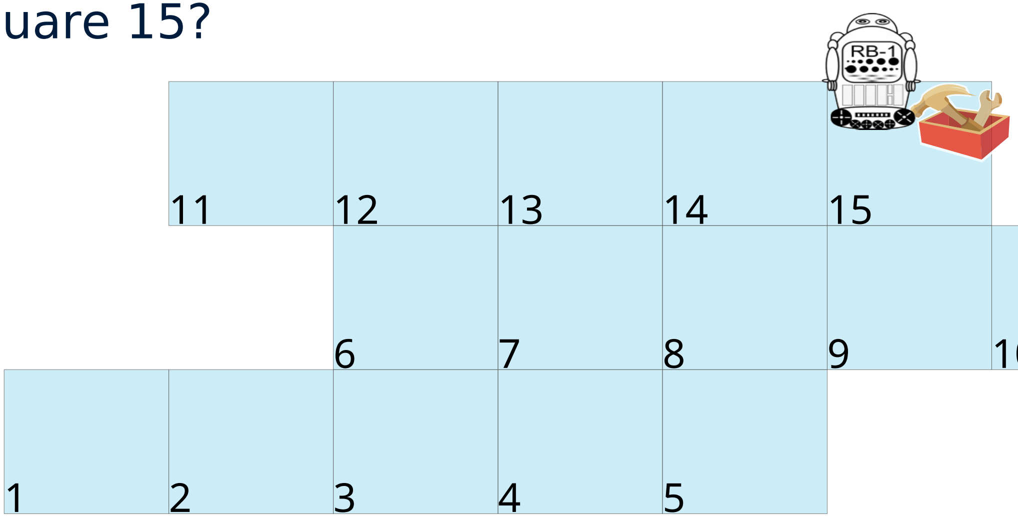
- $Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s')$
- $V^*(s) = \max_a Q^*(s,a)$

Robot needs to go to toolbox, and pick it up.
→ reward: +1
→ let's assume γ=0.9
→ and deterministic movement

Example: pick up the toolbox

if we were at square 15?

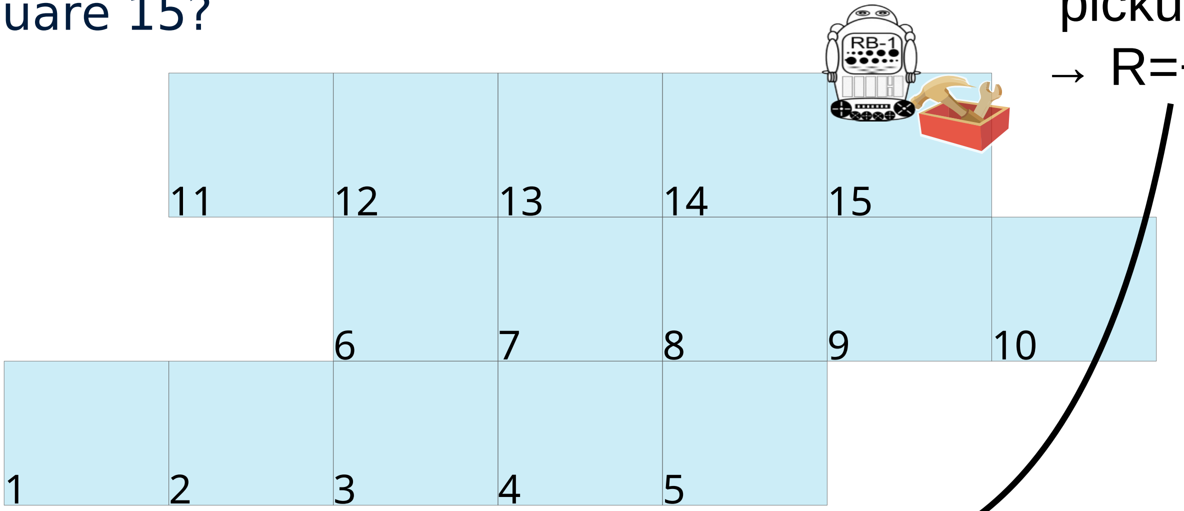
“pickup”
→ R=+1



- $Q^*(s,a) = R(s,a)+\gamma \sum_{s'} P(s'|s,a)V^*(s')$
- $V^*(s) = \max_a Q^*(s,a)$

Example: pick up the toolbox

if we were at square 15?



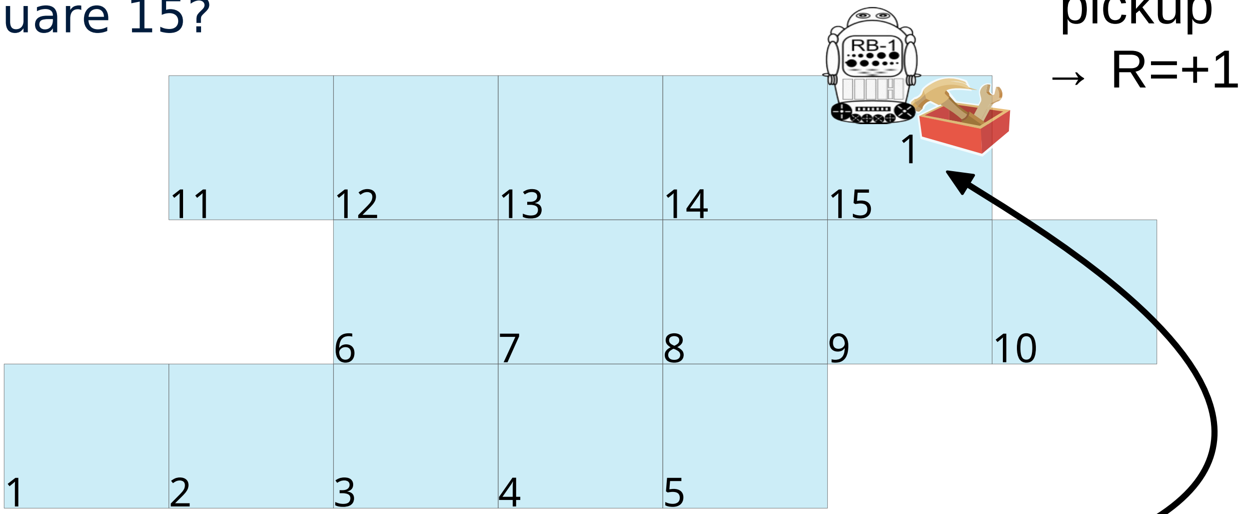
“pickup”
→ R=+1

- $Q^*(s,a) = R(s,a)+\gamma \sum_{s'} P(s'|s,a)V^*(s')$
- $V^*(s) = \max_a Q^*(s,a)$

$Q^*(s=15, a=pickup) = 1$

Example: pick up the toolbox

if we were at square 15?



"pickup"
→ R=+1

- Q*(s,a) = R(s,a)+γ $\Sigma_{s'}$ P(s'|s,a)V*(s')
- V*(s) = $\max_a$ Q*(s,a)

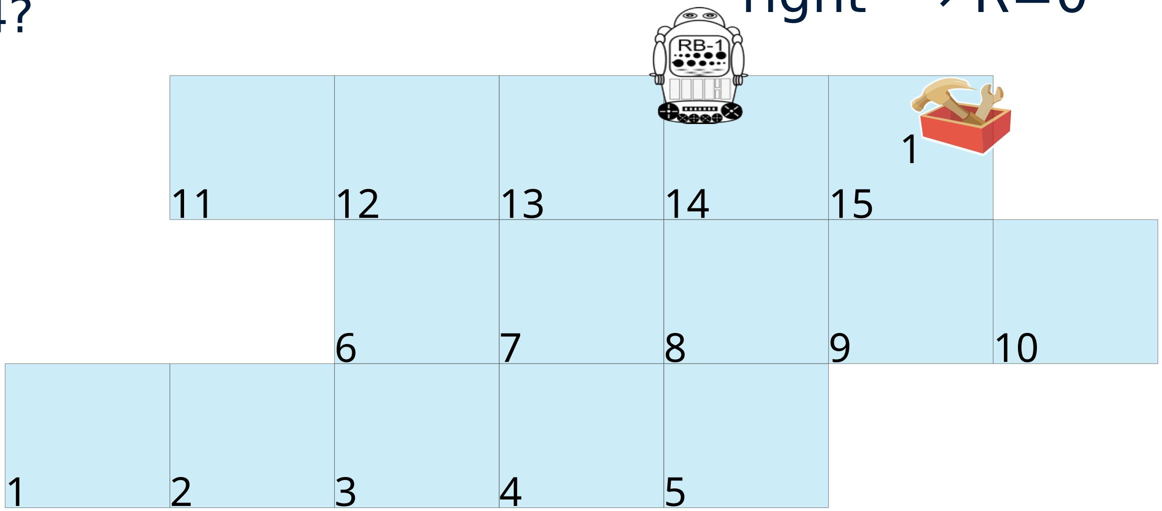
Q*(s=15, a=pickup) = 1

V*(s=15) = 1

Example: pick up the toolbox

if we were at 14?

"right" → R=0



- $Q^*(s,a) = R(s,a)+\gamma \sum_{s'} P(s'|s,a)V^*(s')$
- $V^*(s) = \max_a Q^*(s,a)$

Q*(s=15, a=pickup) = 1

V*(s=15) = 1

Example: pick up the toolbox

if we were at 14?

“right” → R=0

- $Q^*(s,a) = R(s,a)+\gamma \sum_{s'} P(s'|s,a)V^*(s')$
- $V^*(s) = \max_a Q^*(s,a)$

Q*(s=15, a=pickup) = 1

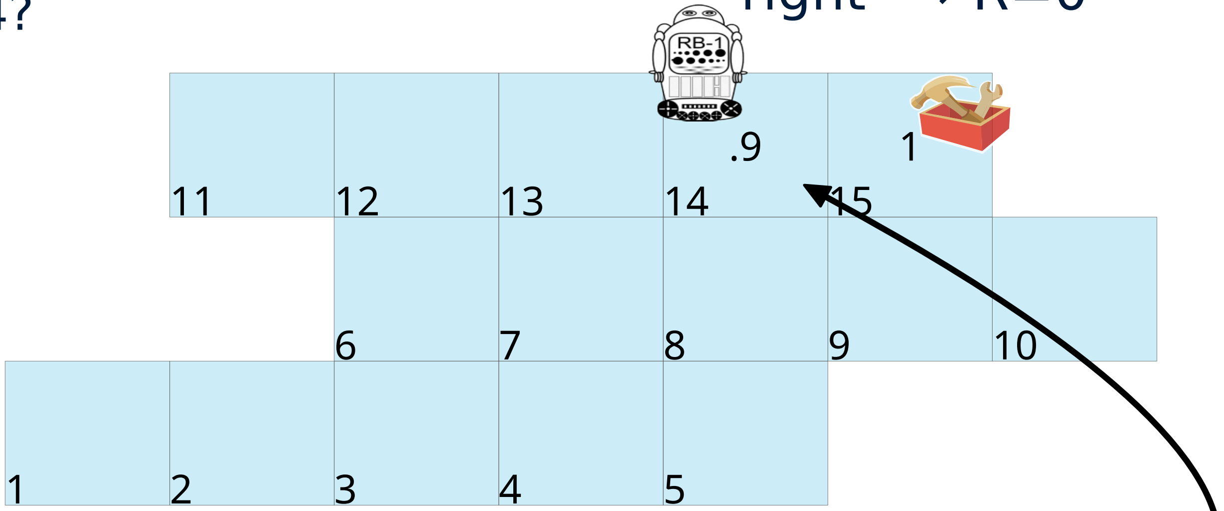
V*(s=15) = 1

Q*(s=14, a=right) = 0 + 0.9 * 1 = 0.9

Example: pick up the toolbox

if we were at 14?

“right” → R=0



- $Q^*(s,a) = R(s,a)+\gamma \sum_{s'} P(s'|s,a)V^*(s')$
- $V^*(s) = \max_a Q^*(s,a)$

$Q^*(s=15, a=pickup) = 1$

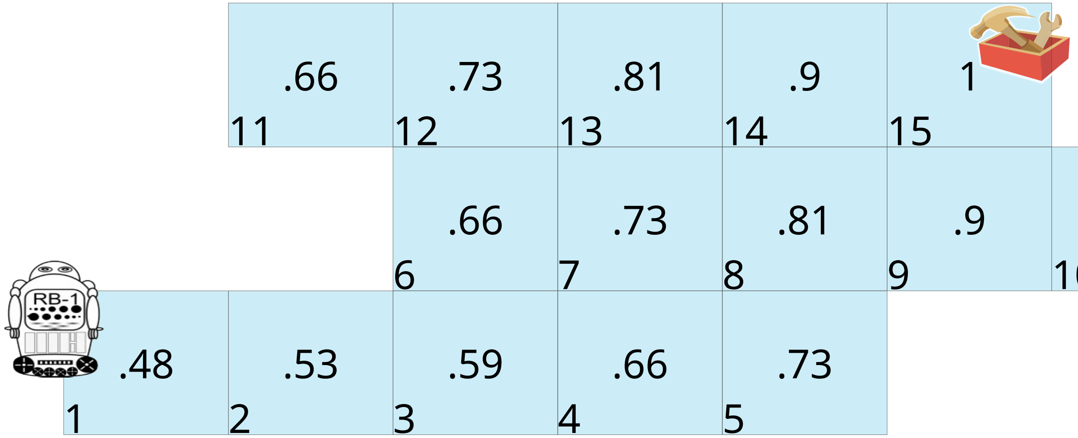
$V^*(s=15) = 1$

$Q^*(s=14, a=right) = 0 + 0.9 * 1 = 0.9$

$V^*(s=14) = 0.9$

Example: pick up the toolbox

compute value of any location



- $Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s')$
- $V^*(s) = \max_a Q^*(s,a)$

Q*(s=15, a=pickup) = 1

V*(s=15) = 1

Q*(s=14, a=right) = 0 + 0.9 * 1 = 0.9

V*(s=14) = 0.9

…

…

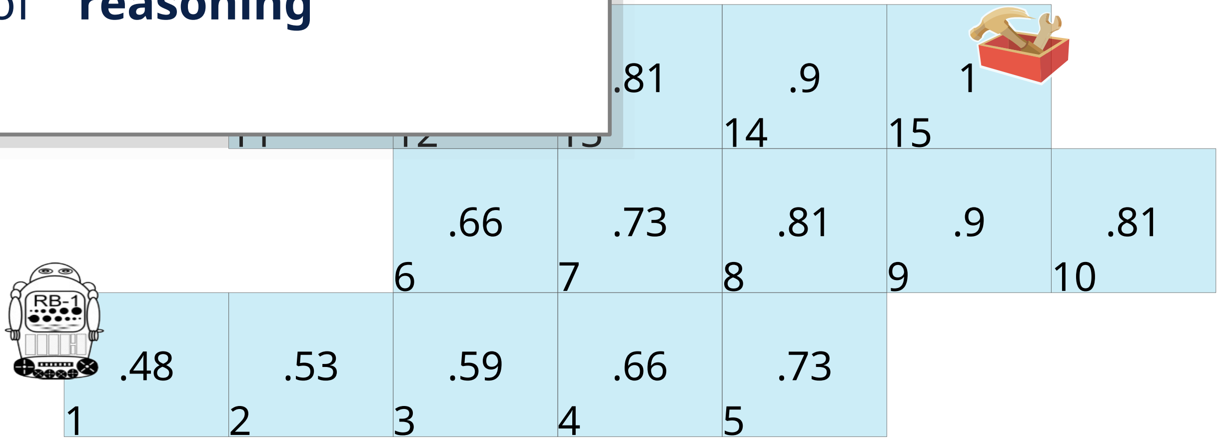
Q*(s=1, a=right) = 0 + 0.9 * .53 = 0.48

V*(s=1) = 0.48

Example: pick up the toolbox

This is also called **“dynamic programming”**

a form of **“reasoning”**



- $Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s')$
- $V^*(s) = \max_a Q^*(s,a)$

$Q^*(s=15, a=\text{pickup}) = 1$

$V^*(s=15) = 1$

$Q^*(s=14, a=\text{right}) = 0 + 0.9 * 1 = 0.9$

$V^*(s=14) = 0.9$

…

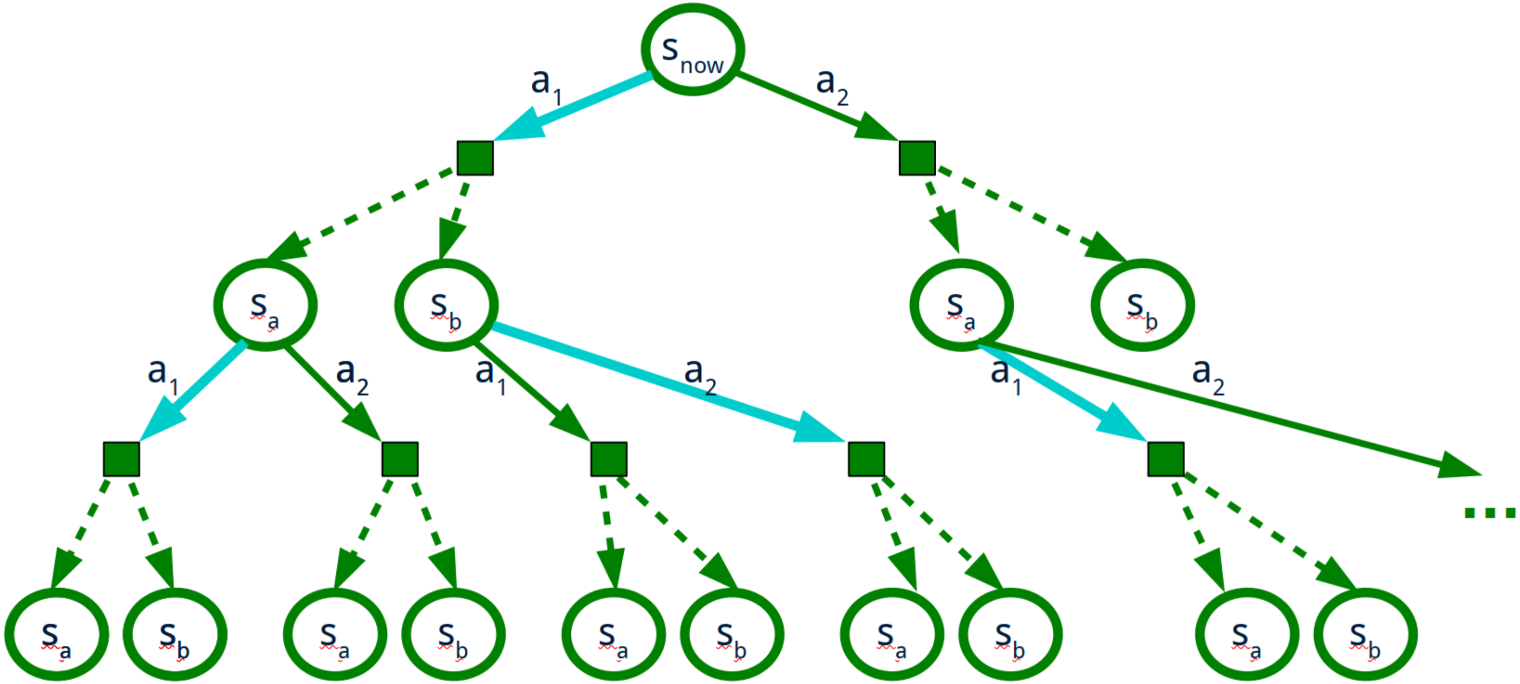
…

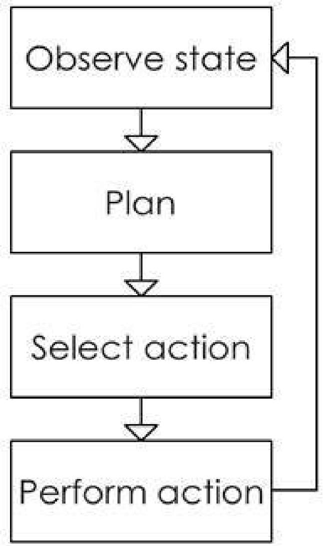
$Q^*(s=1, a=\text{right}) = 0 + 0.9 * .53 = 0.48$

$V^*(s=1) = 0.48$

Alternative: lookahead search

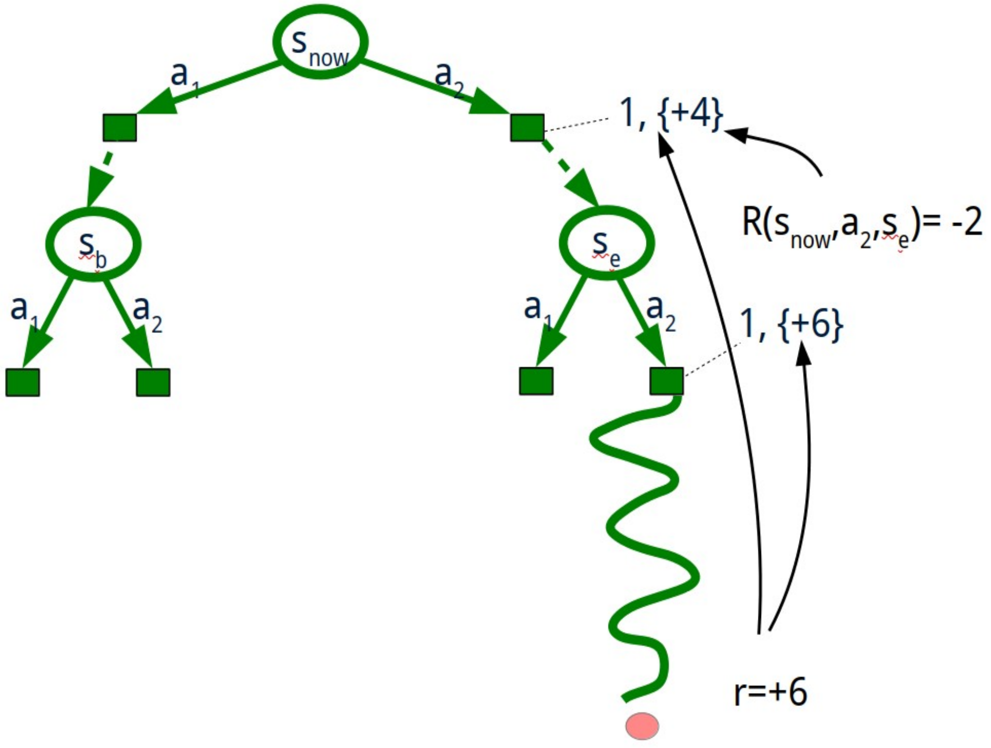
- Plan while taking actions
- **Forward** search





Efficient search: Monte-Carlo tree search

- Full trees are too large…
- …instead:
  approximate search
  (like MCTS)

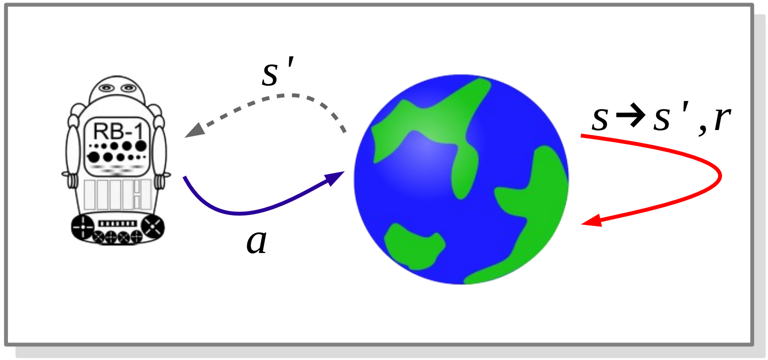


Planning via **"search"**

a form of **"reasoning"**

MDP Planning

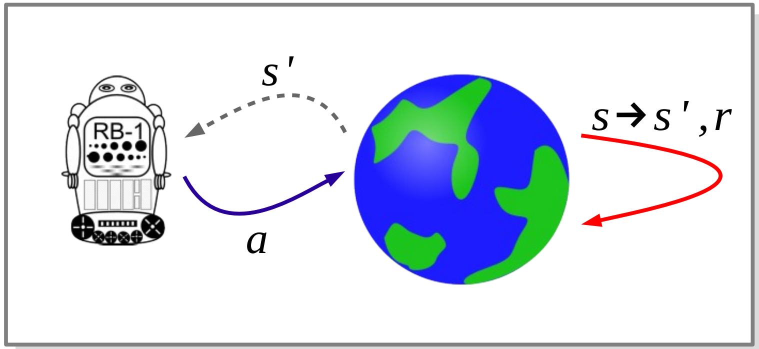
- Given an MDP:
  - *S* – set of states
  - *A* – set of actions
  - transition model: $P(s'|s,a)$
  - rewards: $R(s,a)$
- Goal:
  - **compute** a policy π
  - that optimizes value $V(\pi)$



~~MDP Planning~~ Reinforcement Learning

- Given ~~an MDP~~:
  - *S* – set of states
  - *A* – set of actions
  - ~~transition model: *P(s'|s,a)*~~
  - ~~rewards: *R(s,a)*~~

- Goal:
  - ~~compute~~ **learn** a policy *π*
  - that optimizes value V(π)



Other motivations?

- problem too large to plan
- agents can be **adaptive** !

  (e.g., when the world changes)

Q-learning [Watkins&Dayan ’92]

- Takes Bellman equations

  → turns into an update equation that learns from **sampled experience**

  - $Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s')$
  - $V^*(s) = \max_a Q^*(s,a)$

- So:
  - start acting in the environment
  - After each transition (s,a,r,s’) we update
    - $Q(s,a) := (1-\alpha)\, Q(s,a) + \alpha\, [\, r + \gamma \max_{a'} Q(s',a')\,]$

- Need to sufficiently **explore** the environment
  - But then will converge to Q*, in small “tabular” environments

Watkins CJ, Dayan P. Q-learning. Machine learning. 1992 May;8(3):279-92.

Q-learning [Watkins&Dayan '92]

- Takes Bellman equation

  → turns into an upda

  that learns from **sa**

Note: how few assumptions...

→ these let an agent learn to "be happy" in any MDP

→ **general purpose learner**...!

- So:
  - start acting in the environment
  - After each transition (s,a,r,s') we update
    - $Q(s,a) := (1-\alpha)\ Q(s,a) + \ \alpha\ [\ r + \gamma\ \max_{a'}\ Q(s',a')\ ]$

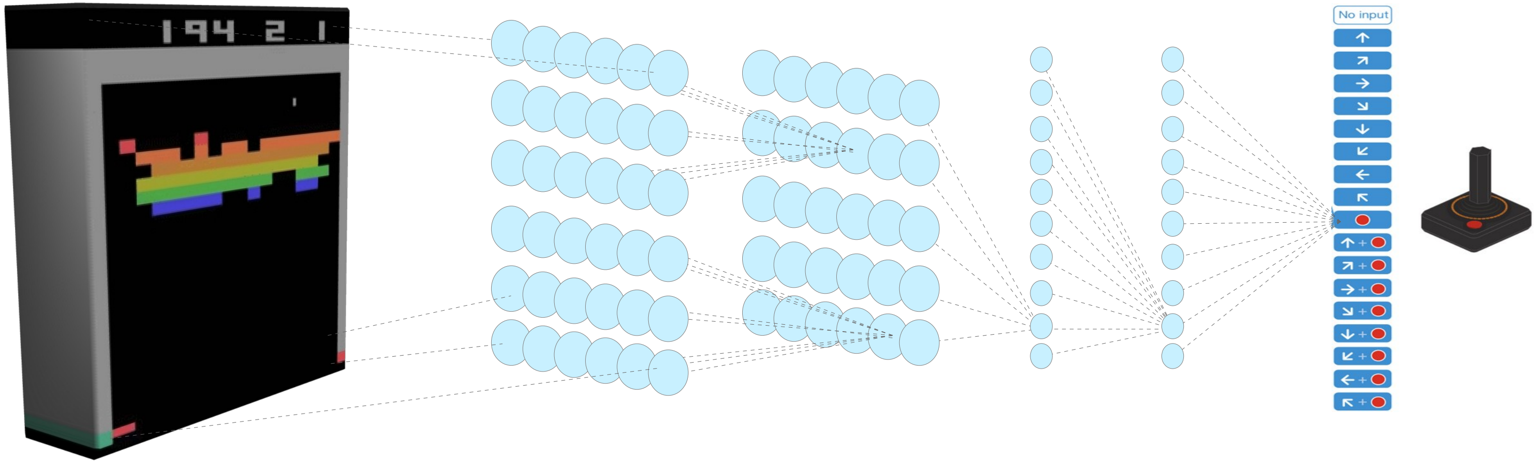
- Need to sufficiently **explore** the environment
  - But then will converge to Q*, in small "tabular" environments

Watkins CJ, Dayan P. Q-learning. Machine learning. 1992 May;8(3):279-92.

Deep RL: Scaling up via deep learning

- Methods covered so far are **tabular**: Q-values for each (s,a) in a table
- But MDPs are **huge**…! (e.g., number of possible screens in Atari?)
  → use function approximation to scale up!

- Prototypical example: DQN [Mnih et al. 2015]
  - Q-network: 84x84 image → 'action values'. Train with Q-learning



  - (many details…Q-learning with neural networks might diverge…)

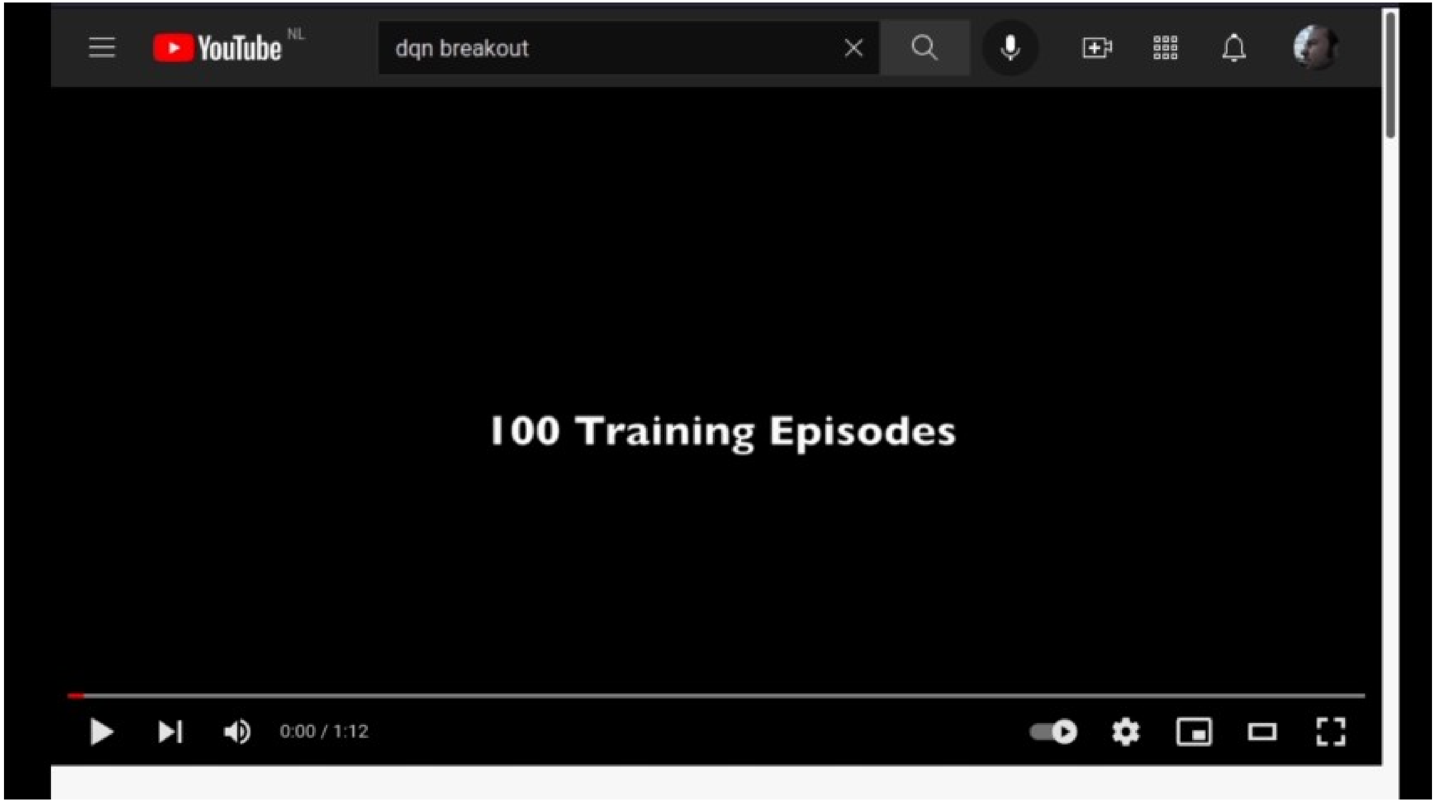
Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.

Deep RL: Scaling up via deep learning

- Methoc
- But MD
  → use f
- Prototy
  - Q-n

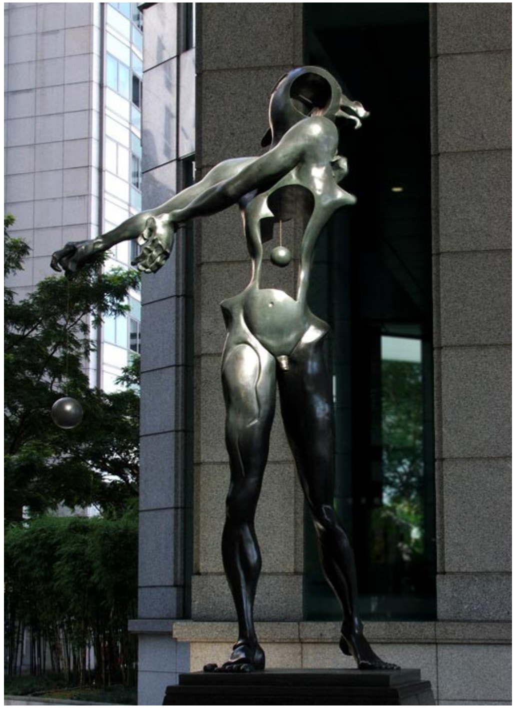
  - (ma

This got a lot of people excited!



Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.

Part 4
State of the Art & Future



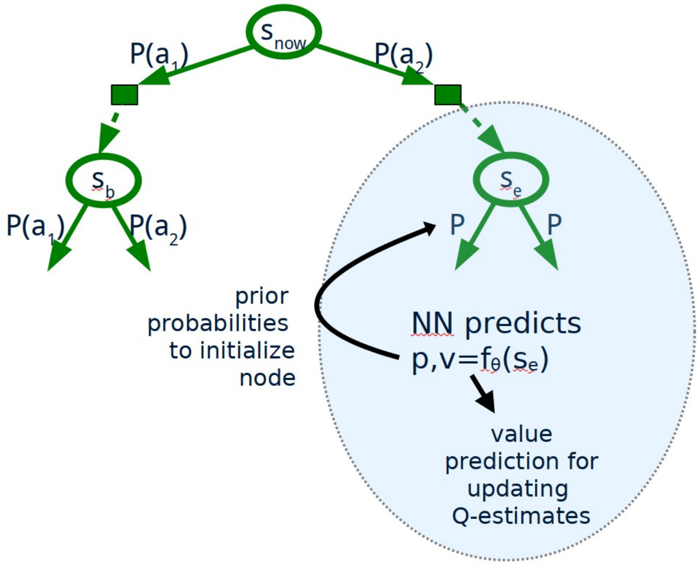
Homage to Newton by Salvador Dali

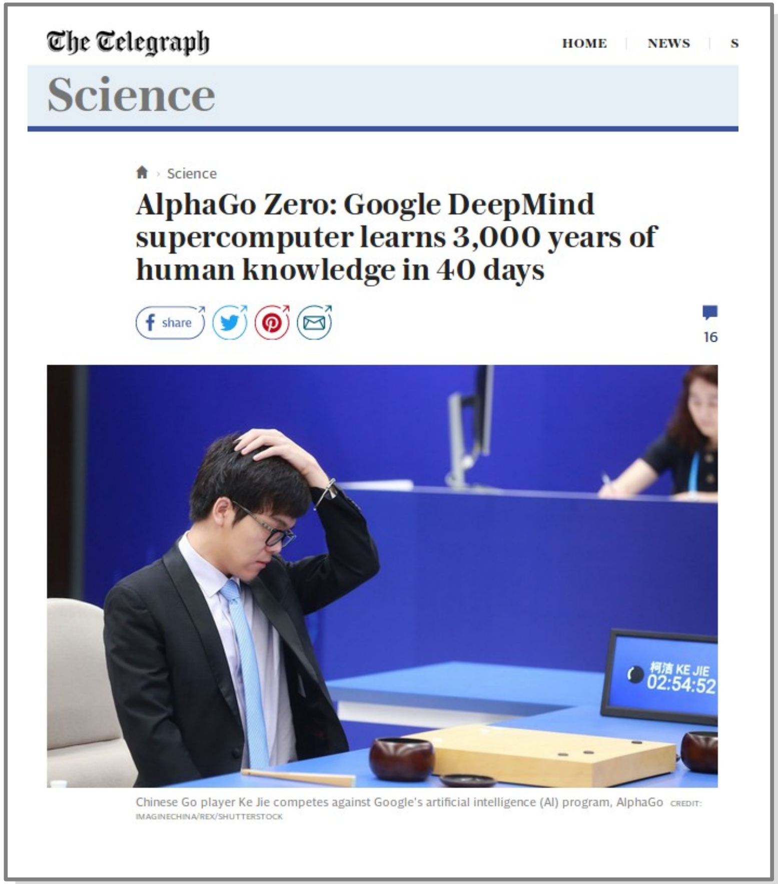
Part 4a

Two examples of SOTA RL Agents

AlphaGo [Silver et al. 2016]

- Combines neural networks and MCTS
- Main challenges:
  - many actions → learn a policy network
  - deep trees → value network





The Telegraph

HOME | NEWS | S

Science

Science

AlphaGo Zero: Google DeepMind supercomputer learns 3,000 years of human knowledge in 40 days

share

16

Chinese Go player Ke Jie competes against Google's artificial intelligence (AI) program, AlphaGo CREDIT: IMAGINECHINA/REX/SHUTTERSTOCK

Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." nature 529.7587 (2016): 484-489.

Hide and Seek – OpenAI (2019)

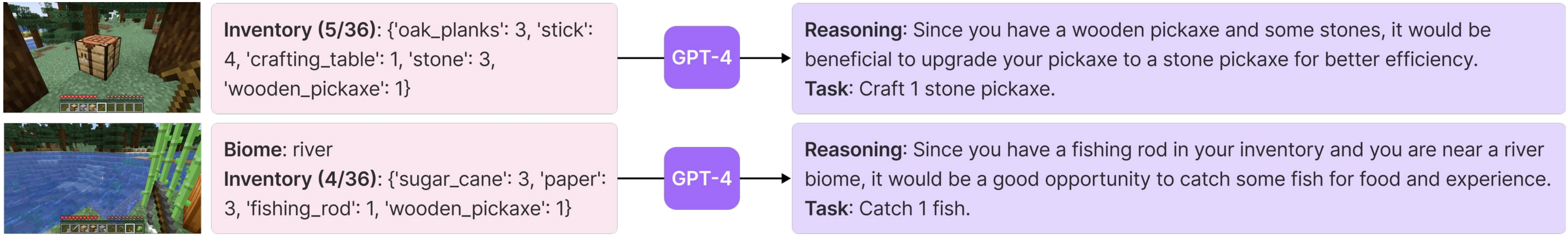
Baker, Bowen, et al. "Emergent tool use from multi-agent autocurricula." ICLR. 2019.

Part 4b
Alternative: LLM agents

LLMs as Agents

- Language models can do amazing things… so why not directly use these as agents?

- E.g., Voyager [Wang et. al 2024]
  - use chatGPT to play minecraft, quite well!
  - the LLM generates code that is used to generate actions
    - "BDI plan library generated on-the-fly"



Wang, Guanzhi, et al. "Voyager: An open-ended embodied agent with large language models." TMLR (2024).

But still needs work…

- …remember?

Two Clear Examples of Agentic AI

- 1) Autonomous Research Assistant
  - Searches, filters, summarizes sources
  - Drafts reports with citations
  - Automates literature reviews
-
  2) Automated Project Planning Agent
  - Breaks goals into tasks
  - Schedules timelines & resources
  - Monitors progress autonomously

STEVEN LEVY   BUSINESS   JAN 23, 2026 11:00 AM

The Math on AI Agents Doesn't Add Up

A research paper suggests AI agents are mathematically doomed to fail. The industry doesn't agree.

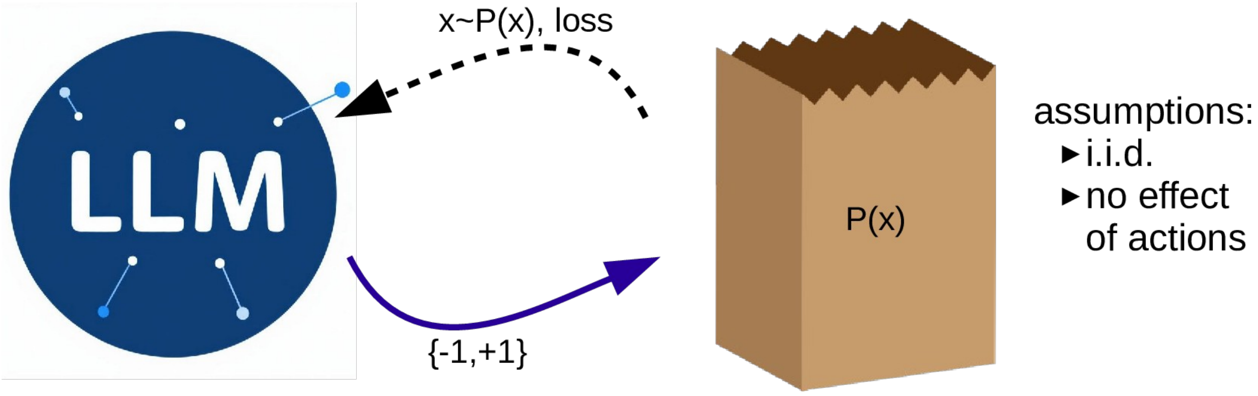
"The big AI companies promised us that 2025 would be 'the year of the AI agents.' It turned out to be the year of talking about AI agents […]"

https://www.wired.com/story/ai-agents-math-doesnt-add-up/

Part 4c
Limits of LLMs…?

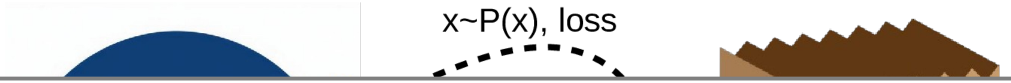
Supervised Training of Language Models

- Historically (self-)supervised learning: predict next words...
- ...but not all tasks we can get good datasets...
  - » e.g., summarizing a book



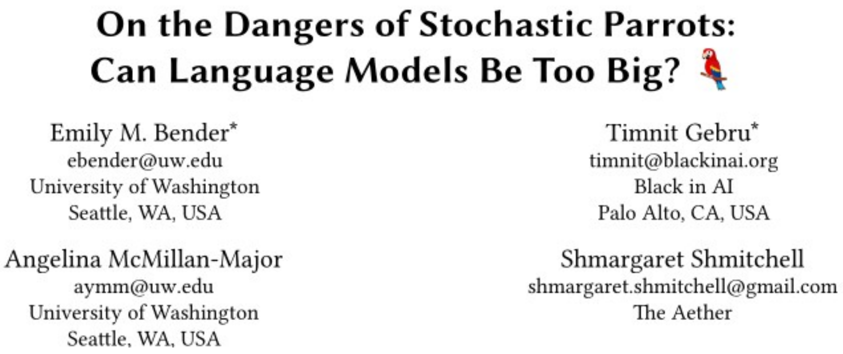
Supervised Training of Language Models

- Historically (self-)supervised learning: predict next words…
- …but not all tasks we can get good datasets…
  - » e.g., summarizing a book



Note:

- many objections against capabilities (biased data, not creative, etc.) are based on limitations of supervised learning



On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜

Emily M. Bender*
ebender@uw.edu
University of Washington
Seattle, WA, USA

Timnit Gebru*
timnit@blackinai.org
Black in AI
Palo Alto, CA, USA

Angelina McMillan-Major
aymm@uw.edu
University of Washington
Seattle, WA, USA

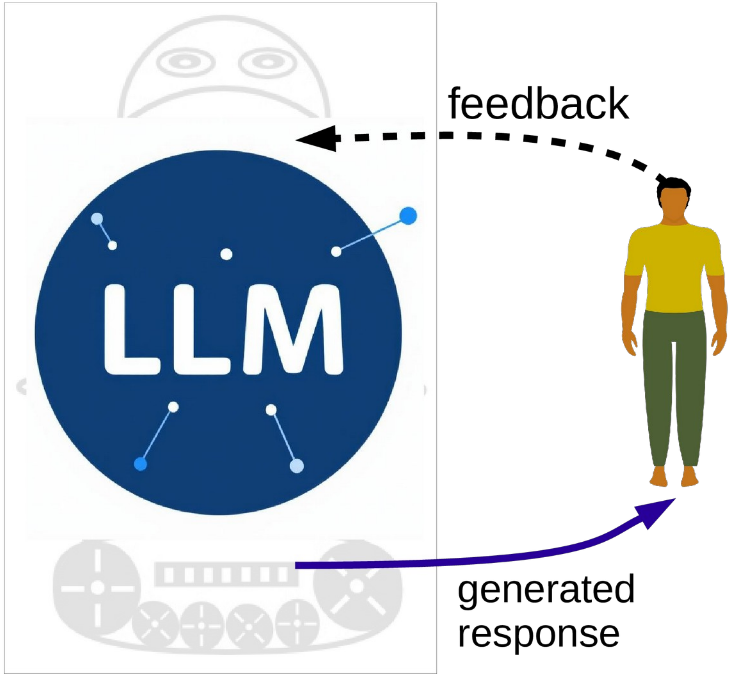
Shmargaret Shmitchell
shmargaret.shmitchell@gmail.com
The Aether

Fine tune based on human feedback: RLHF

- To deal with this… get feedback from humans instead!

- The new approach for LLMs/foundation models:
  - pre-train (self-)supervised on dataset
  - fine tune with "reinforcement learning from human feedback (RLHF)"

- And more recently: "reasoning models"
  - yes… they can be interpreted to do "search" just like we saw in AlphaGo!



Fine tune based on human feedback: RLHF

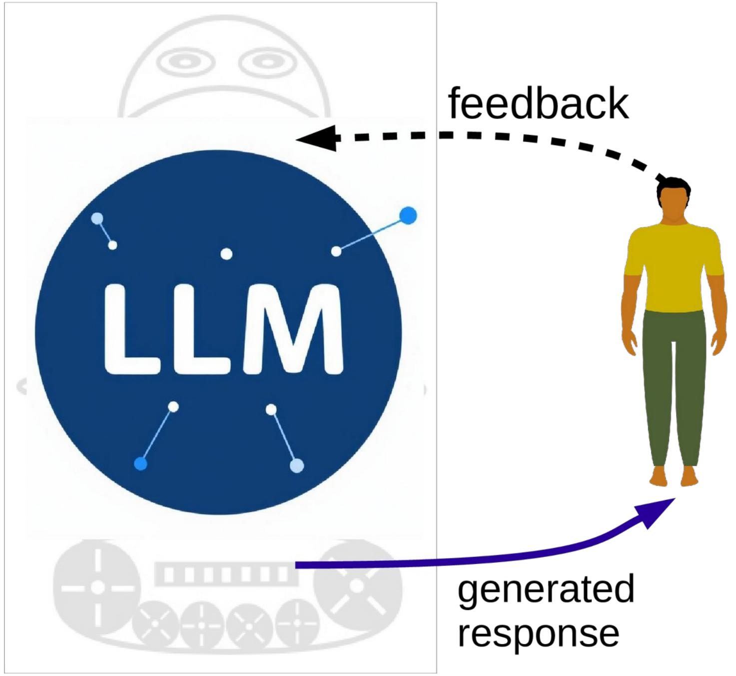
- To deal with this… get feedback from humans instead!

- The new approach for LLMs/foundation models:

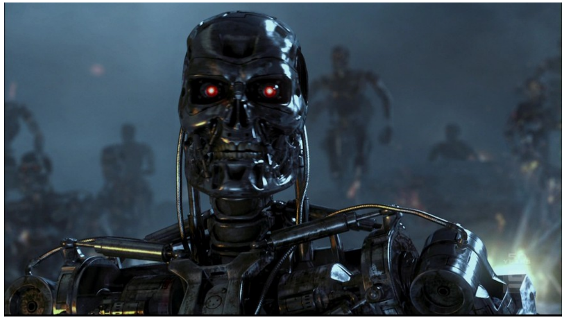
Note:

- theoretically RLHF changes an LLM into a **general purpose learner…!**
  - Learn optimal behavior w.r.t our feedback
    (or w.r.t. the reward big tech translates this too)

- and they **massively interact** with the human population…



Part 4d: Risks?
E.g., “Value-alignment”

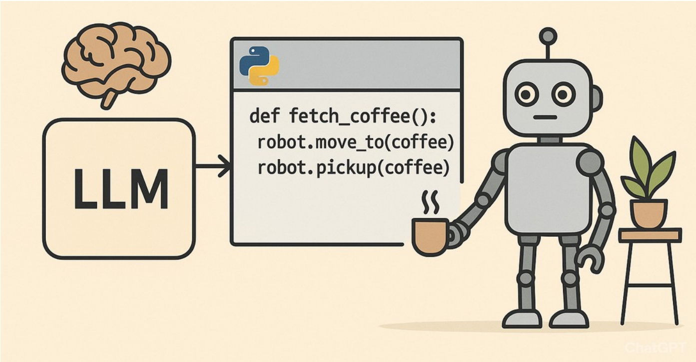


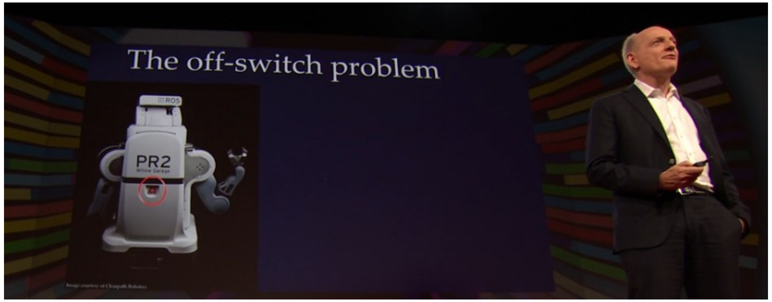
Even “fetch the coffee”…

- …using LLMs and foundation models is tricky…

- Imagine something like Voyager [Wang et al. 2024]:
  - LLM produces an code for an MDP to solve “fetch_coffee”
  - MDP is solved → coherent long term plan(!)
  - Robot executes the policy

- Do we trust it to get the reward function right?
  - or will it be dangerous?
  - or prevent itself from being switched off?



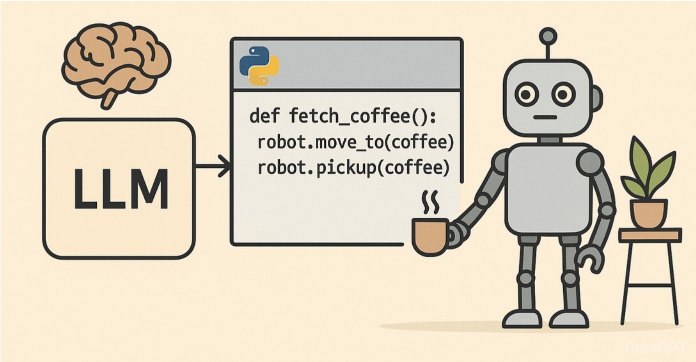


Stuart Russel about the “off-switch problem”
https://www.ted.com/talks/stuart_russell_3_principles_for_creating_safer_ai

Wang, Guanzhi, et al. "Voyager: An Open-Ended Embodied Agent with Large Language Models." Transactions on Machine Learning Research. 2024.

Even “fetch the coffee”…

- …using LLMs and foundation models is tricky…
- Imagine something like Voyager [Wang et al. 2024]:
  - LLM produces an code for an MDP to solve “fetch_coffee”
  - MDP is solved → coherent long term plan(!)
  - Robot executes the policy
- Do we trust it to get the reward function right?



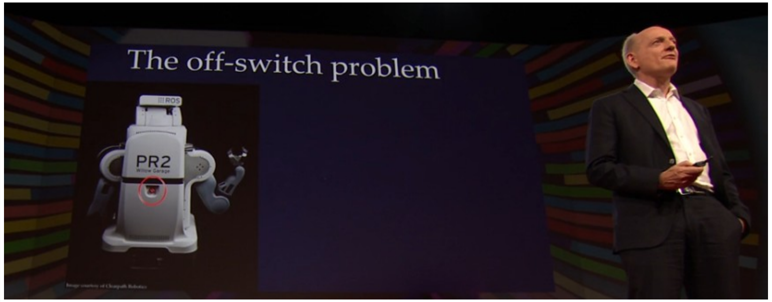
I believe there are solutions!

E.g.:

- **constantly reason** about humans true intention
- ask clarification where needed
- only act when certain

interactive learning & decision making

But, integrating in current AI / LLMs needs (much) more work!



Stuart Russel about the “off-switch problem”
https://www.ted.com/talks/stuart_russell_3_principles_for_creating_safer_ai

Wang, Guanzhi, et al. "Voyager: An Open-Ended Embodied Agent with Large Language Models." Transactions on Machine Learning Research. 2024.

Conclusions

Conclusions

- “Agents” have been around for a long time!
  - BDI manually programmed plan libraries → LLMs learning from humans

- Challenges… many important tasks require
  - long-term autonomy
  - adaptation

  → A possible solution:
  sequential reasoning+learning = reinforcement learning

- SOTA and future trends:
  - using LLMs as agents… so far: hit and miss
  - but LLMs + reasoning and RL, already big improvements
  - open questions:
    - inherent limits?
    - mitigating risks?

Questions?