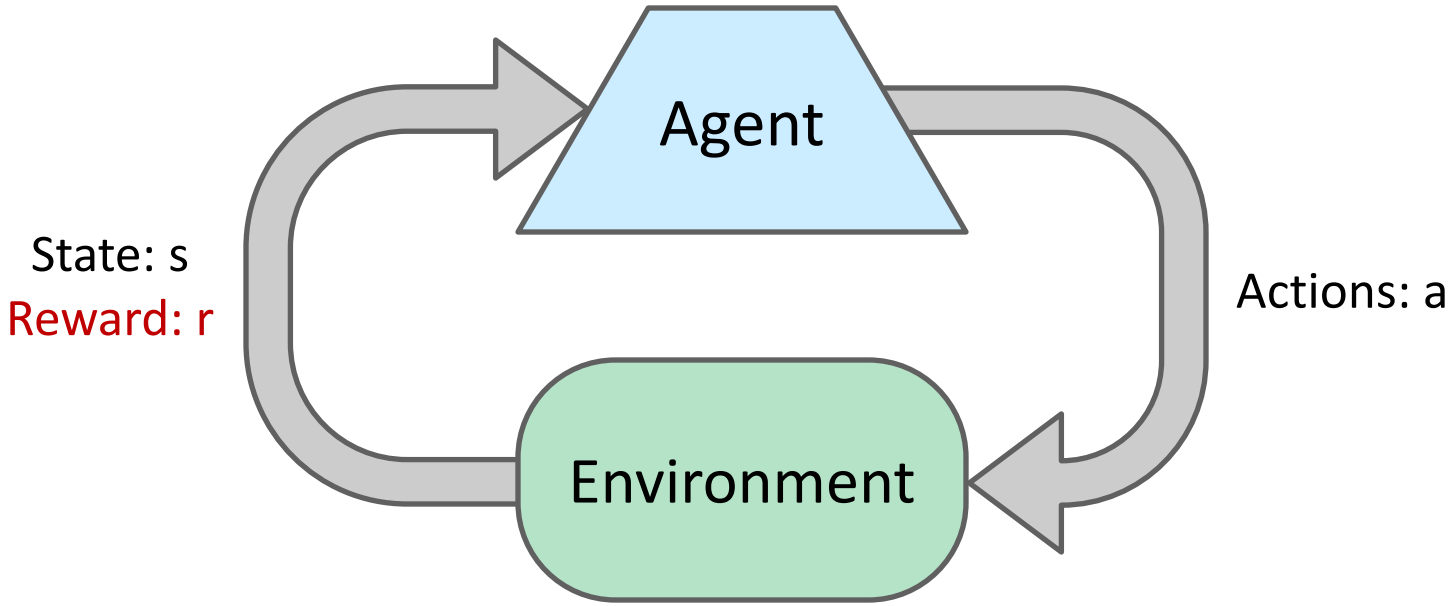
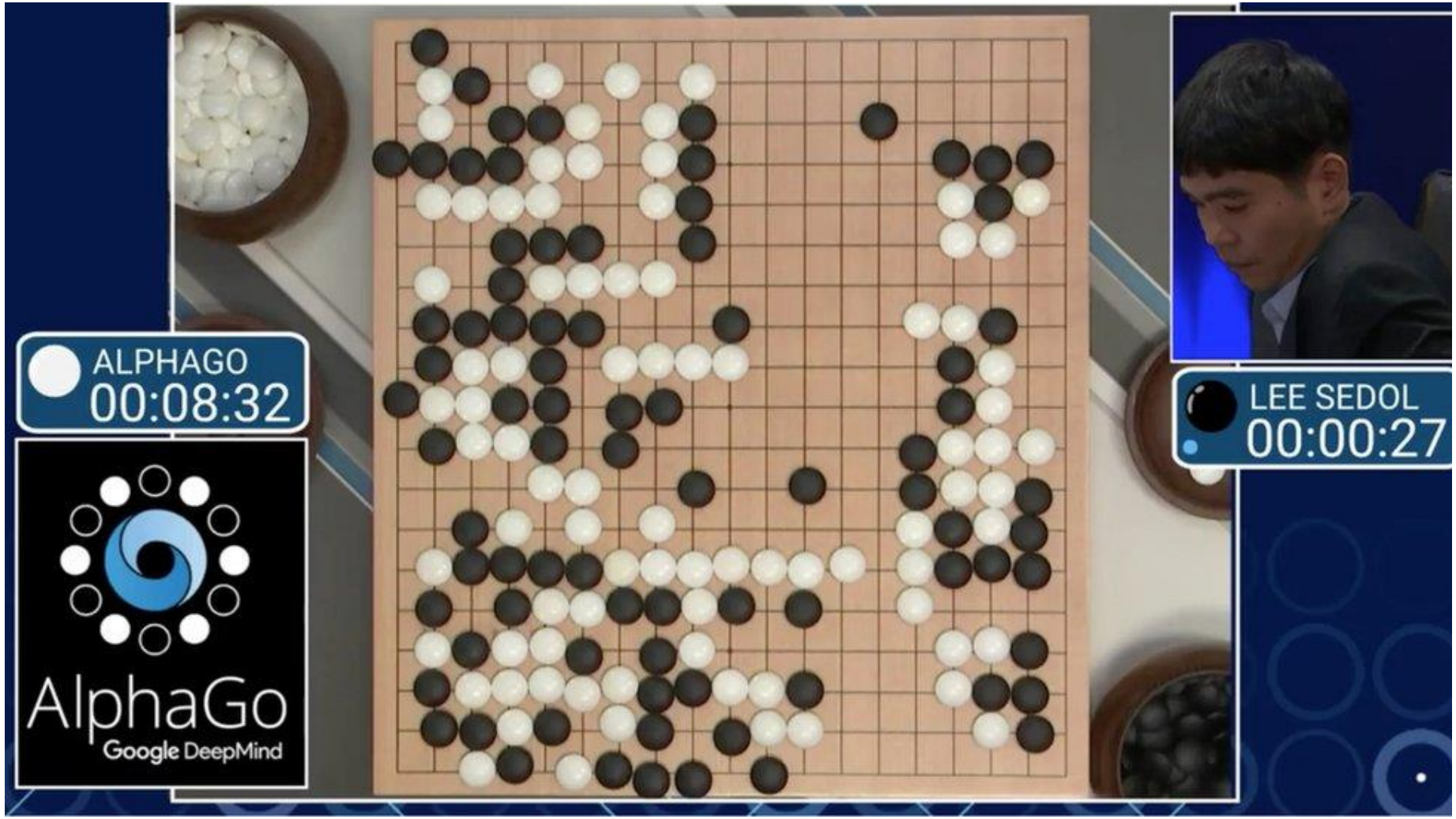
Reinforcement Learning



- Basic idea:
  - Receive feedback in the form of rewards
  - Agent's utility is defined by the reward function
  - Must (learn to) act so as to maximize expected rewards
  - All learning is based on observed samples of outcomes!

The Arcade Learning Environment


ALPHAGO
00:08:32
AlphaGo
Google DeepMind
LEE SEDOL
00:00:27


021 3 1

ChatGPT

Gemini

Claude

deepseek

Why Reinforcement Learning?

- Takes inspiration from nature
- Often easier to encode a task as a sparse reward (e.g. recognize if goal is achieved) but hard to hand-code how to act so reward is maximized (e.g. Go)
- General purpose AI framework

When might RL be a good tool for your problem?

When might RL be a good tool for your problem?

- Is your problem a sequential decision making problem?
- Are there “actions” that effect the next “state”?
- Do you know the rules of these effects?
- Can you write down a clear objective/score/reward/cost?
- Do you have a simulator?
- Lots of examples of sequences of decisions and their long-term consequences?
- Is it unclear what to do in each state? Exploration required?
- Are you looking for unique/creative/super-human solutions?

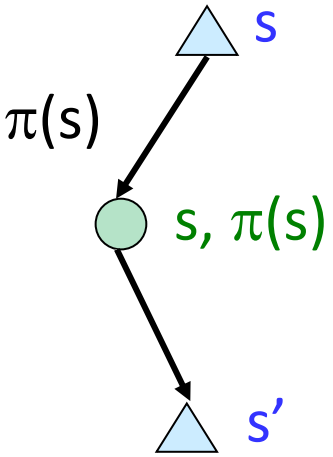
When might RL not be a good tool?

When might RL not be a good tool?

- Single step or static problem
- No clear reward signal.
- Reward signal is unavailable or very hard to write down.
- Well-known model of the environment.
- Deterministic environment
- Low-tolerance for exploration and trial and error
- No need for adaptive or novel solutions. The goal is to perform the task in a very predictable way.

Temporal Difference Learning

- Big idea: learn from every experience!
  - Update V(s) each time we experience a transition (s, a, s', r)
  - Likely outcomes s' will contribute updates more often



- Temporal difference learning of values
  - Policy still fixed, still doing evaluation!
  - Move values toward value of whatever successor occurs: running average

Sample of V(s): $sample = R(s, \pi(s), s') + \gamma V^{\pi}(s')$

Update to V(s): $V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + (\alpha)sample$

Same update: $V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(sample - V^{\pi}(s))$

Q-Learning

- Q-Learning: sample-based Q-value iteration

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

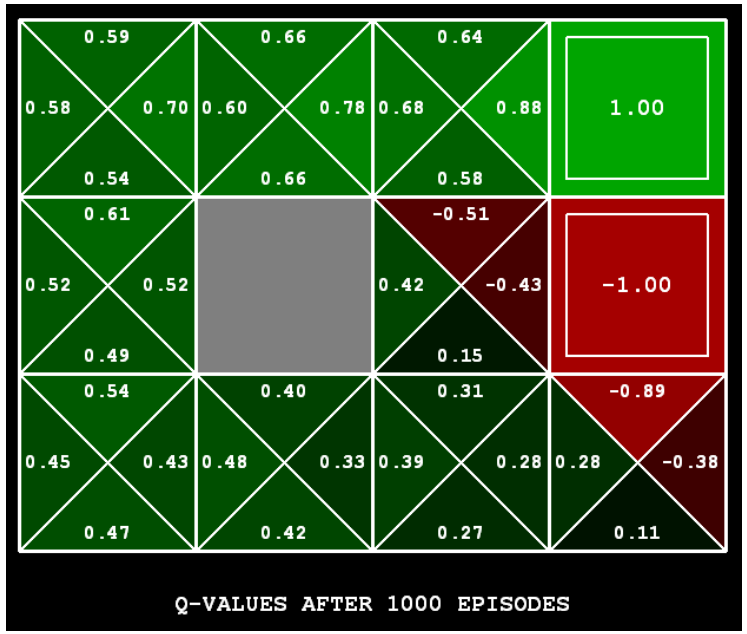
- Learn Q(s,a) values as you go
  - Receive a sample (s,a,s',r)
  - Consider your old estimate: $Q(s,a)$
  - Consider your new sample estimate:

$$sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$$

  - Incorporate the new estimate into a running average:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\,[sample]$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha(sample - Q(s,a))$$



Deep RL Makes a Big Splash!

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature › letters › article

Published: 25 February 2015

Human-level control through deep reinforcement learning

Volodymyr Mnih, Koray Kavukcuoglu ✉, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg & Demis Hassabis ✉

Startups

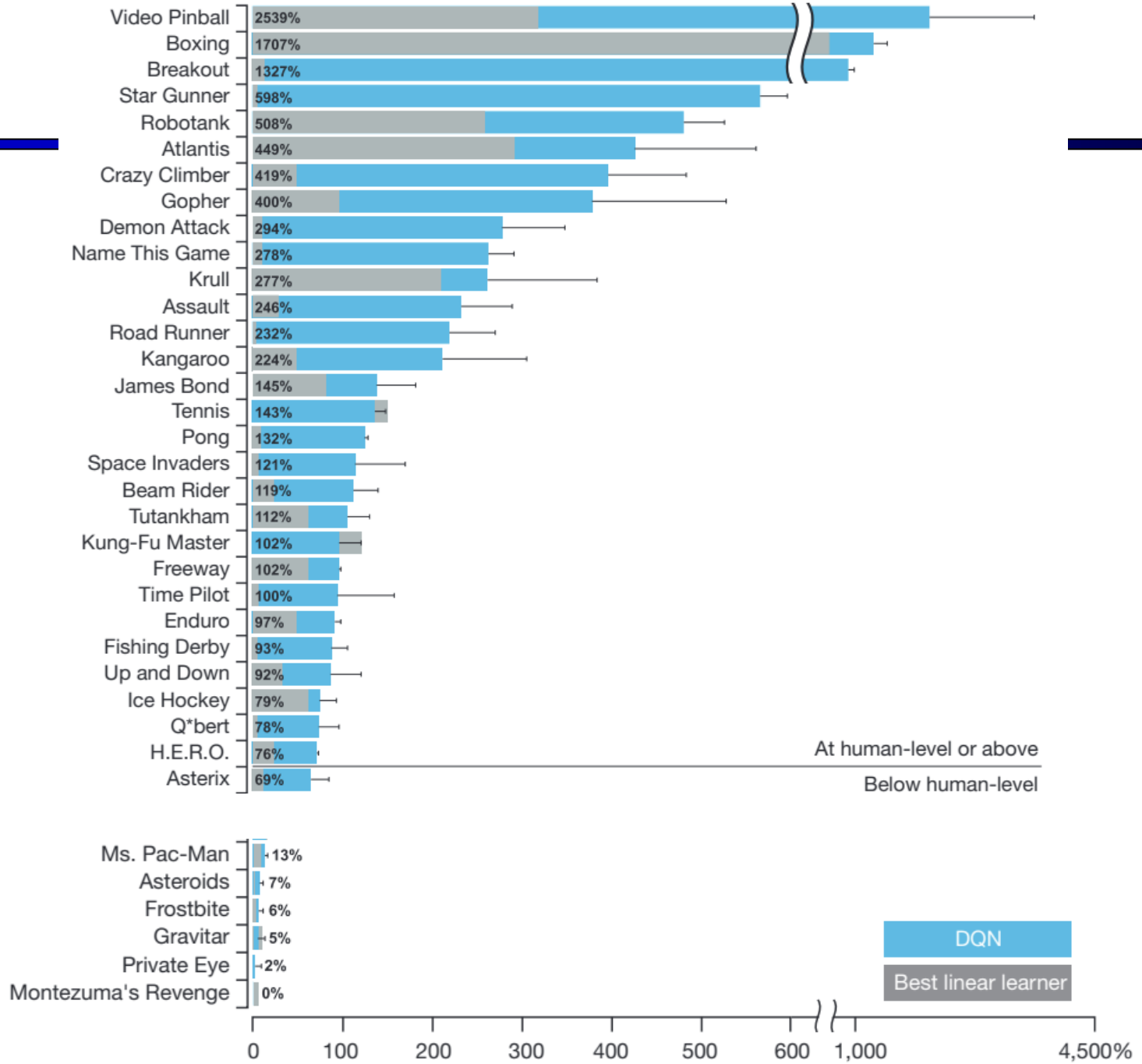
Google Acquires Artificial Intelligence Startup DeepMind For More Than $500M

Catherine Shu @catherineshu / 6:20 PM MST • January 26, 2014

Comment

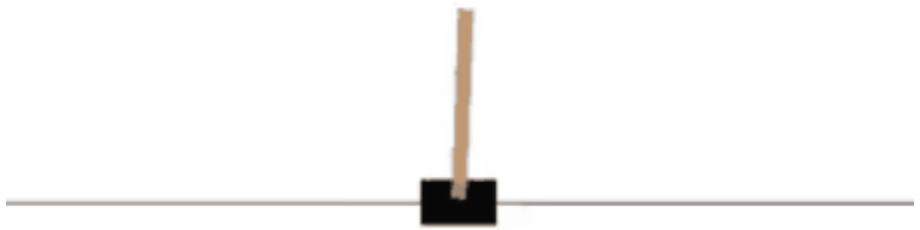
The Arcade Learning Environment


021 3 1


Video Pinball
Boxing
Breakout
Star Gunner
Robotank
Atlantis
Crazy Climber
Gopher
Demon Attack
Name This Game
Krull
Assault
Road Runner
Kangaroo
James Bond
Tennis
Pong
Space Invaders
Beam Rider
Tutankham
Kung-Fu Master
Freeway
Time Pilot
Enduro
Fishing Derby
Up and Down
Ice Hockey
Q*bert
H.E.R.O.
Asterix
2539%
1707%
1327%
598%
508%
449%
419%
400%
294%
278%
277%
246%
232%
224%
145%
143%
132%
121%
119%
112%
102%
102%
100%
97%
93%
92%
79%
78%
76%
69%
At human-level or above
Below human-level
Ms. Pac-Man
Asteroids
Frostbite
Gravitar
Private Eye
Montezuma's Revenge
13%
7%
6%
5%
2%
0%
DQN
Best linear learner
0
100
200
300
400
500
600
1,000
4,500%

Homework 4

- Q-Learning!
- DQN!



Tutorials > Reinforcement Learning (DQN) Tutorial

Shortcuts

Run in Google Colab Download Notebook View on GitHub

Reinforcement Learning (DQN) Tutorial
Replay Memory
+ DQN algorithm
+ Training

Reinforcement Learning (DQN) Tutorial

Created On: Mar 24, 2017 | Last Updated: Jun 18, 2024 | Last Verified: Nov 05, 2024

Author: Adam Paszke

Mark Towers

This tutorial shows how to use PyTorch to train a Deep Q Learning (DQN) agent on the CartPole-v1 task from Gymnasium.

You might find it helpful to read the original Deep Q Learning (DQN) paper

Task

The agent has to decide between two actions - moving the cart left or right - so that the pole attached to it stays upright. You can find more information about the environment and other more challenging

9:13 PM
2/11/2025

Lots of Advanced Exploration Strategies

Unifying Count-Based Exploration and Intrinsic Motivation

Marc G. Bellemare bellemare@google.com
Sriram Srinivasan srsrinivasan@google.com
Georg Ostrovski ostrovski@google.com
Tom Schaul schaul@google.com
David Saxton saxton@google.com
Rémi Munos munos@google.com
Google DeepMind
London, United Kingdom

INCENTIVIZING EXPLORATION IN REINFORCEMENT LEARNING WITH DEEP PREDICTIVE MODELS

Bradly C. Stadie
Department of Statistics
University of California, Berkeley
Berkeley, CA 94720
bstadie@berkeley.edu

Sergey Levine **Pieter Abbeel**
EECS Department
University of California, Berkeley
Berkeley, CA 94720
{svlevine,pabbeel}@cs.berkeley.edu

EXPLORATION BY RANDOM NETWORK DISTILLATION

Yuri Burda* OpenAI
Harrison Edwards* OpenAI
Amos Storkey Univ. of Edinburgh
Oleg Klimov OpenAI

Great blog article: https://lilianweng.github.io/posts/2020-06-07-exploration-drl/

DQN only works for discrete action spaces

- Next: How to deal with continuous action spaces