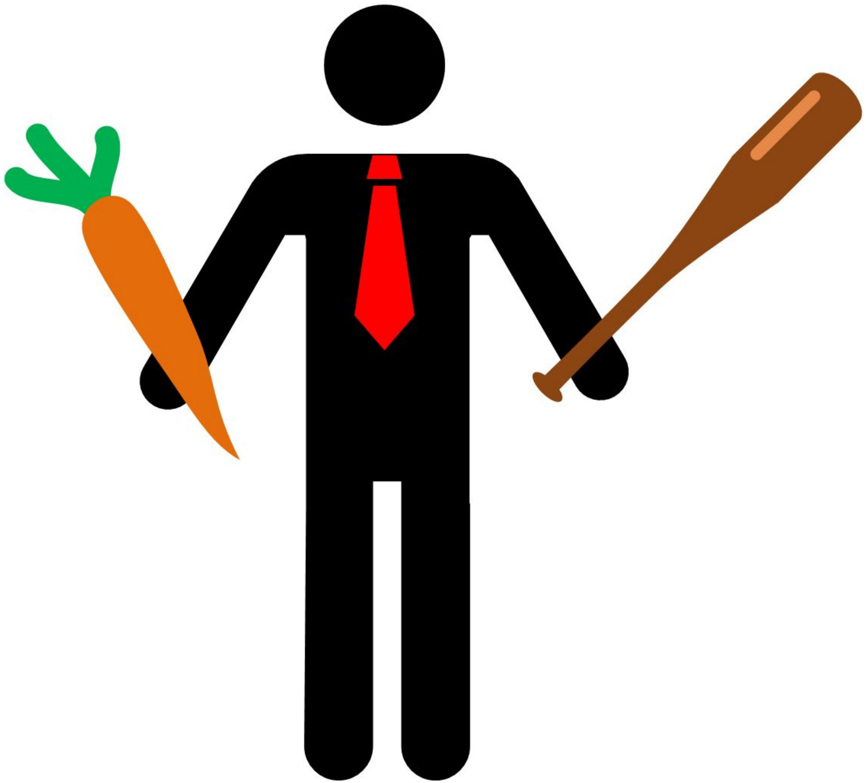
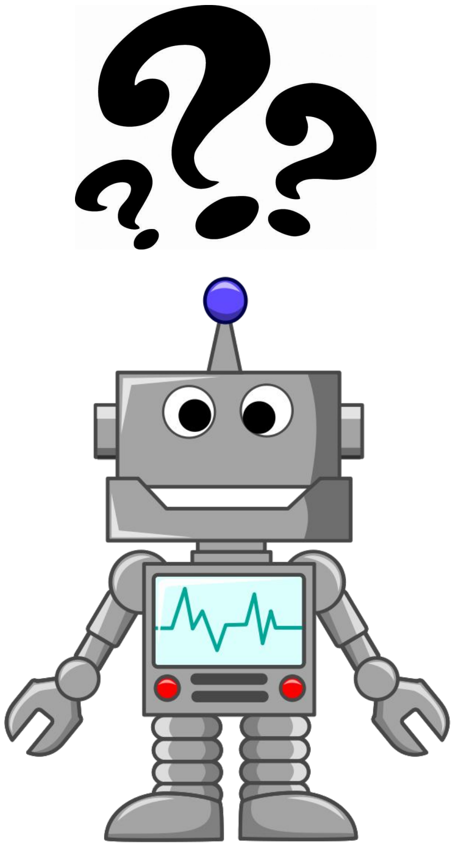
Seminar in Deep Reinforcement Learning

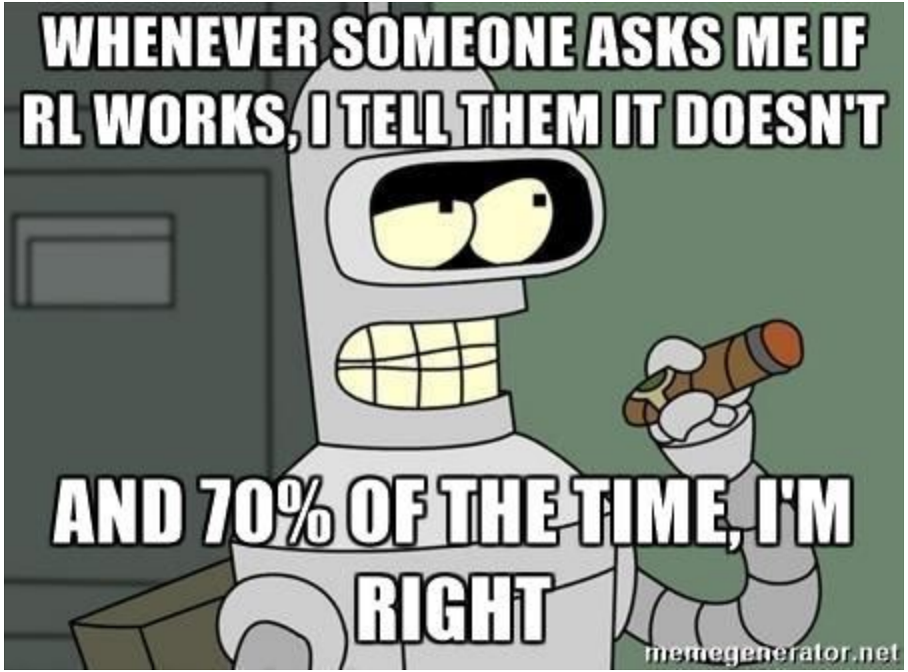
Introduction

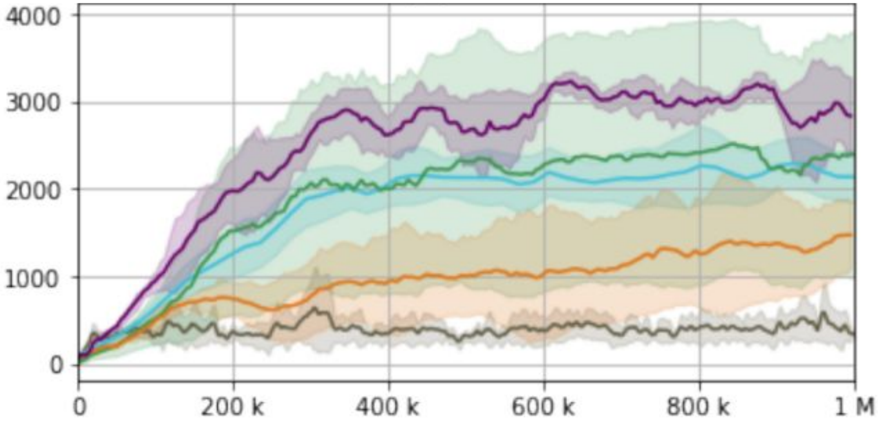
Reinforcement Learning...

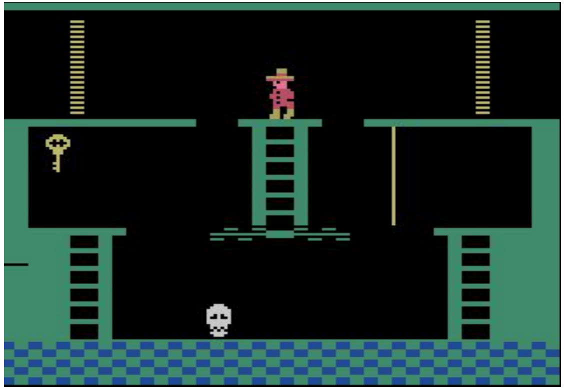




Why you should NOT use reinforcement learning...







Why you should NOT use reinforcement learning...

Reward Engineering can be hard - small auxiliary rewards might become the main focus of the agent

Why is reinforcement learning promising?

- Humans are reinforcement learners
- Agents can outperform teachers
- Can model any (easy verifiable) task

Disclaimer: This is a seminar...

(almost) no basics

participation required

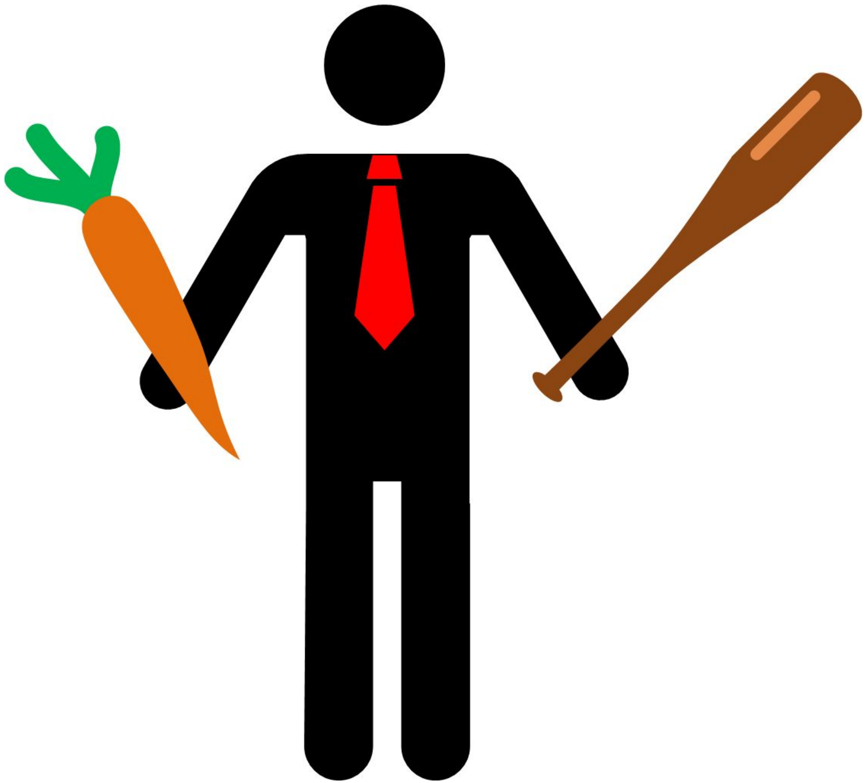
Format

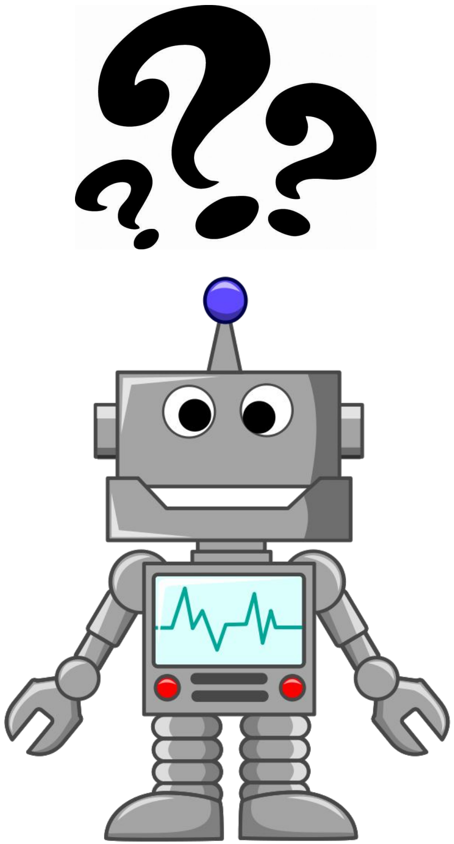
- Assigned topics
- 35 min presentation
- 10 min facilitated discussion
- Voluntary coding challenge

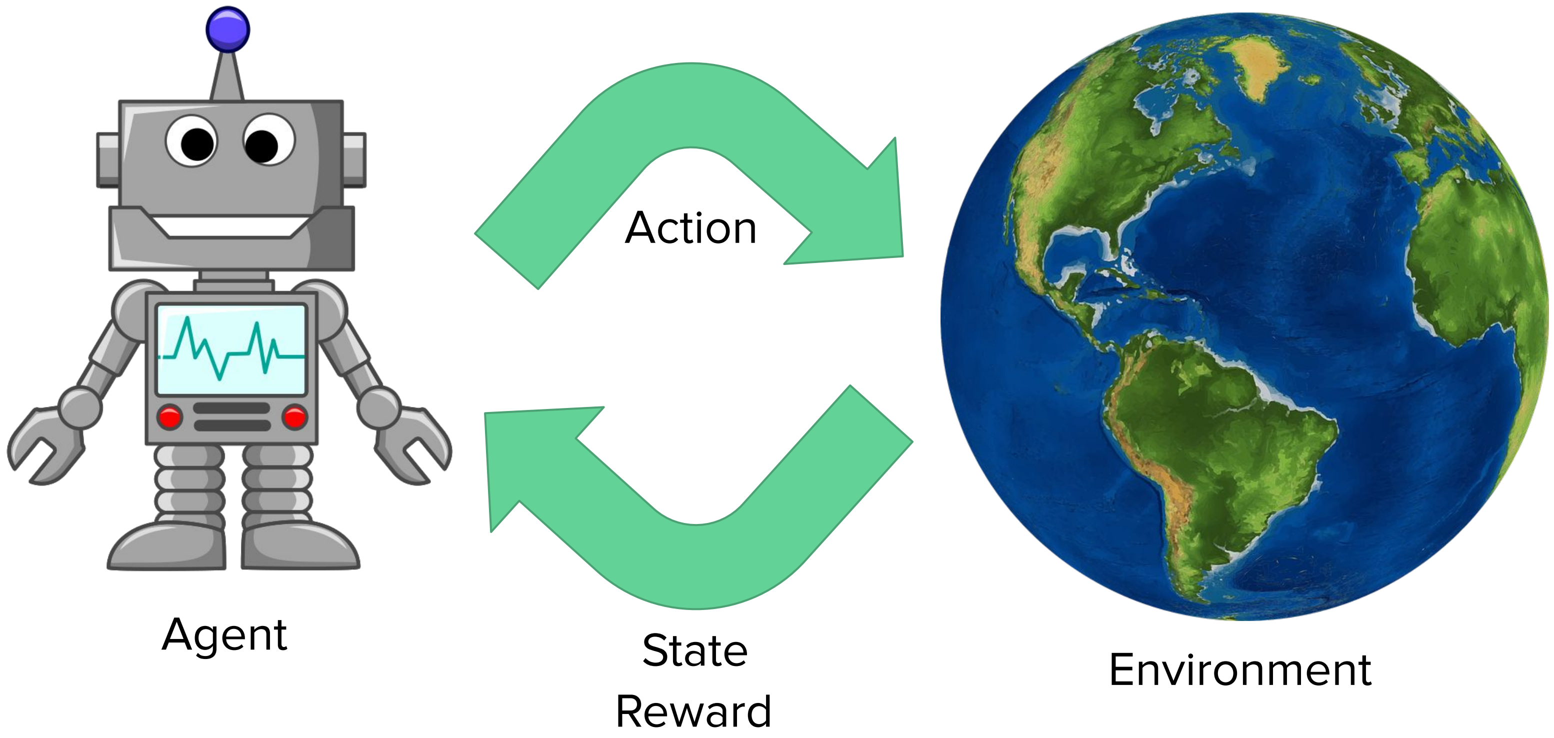
Grade =

presentation + active participation (+ challenge)

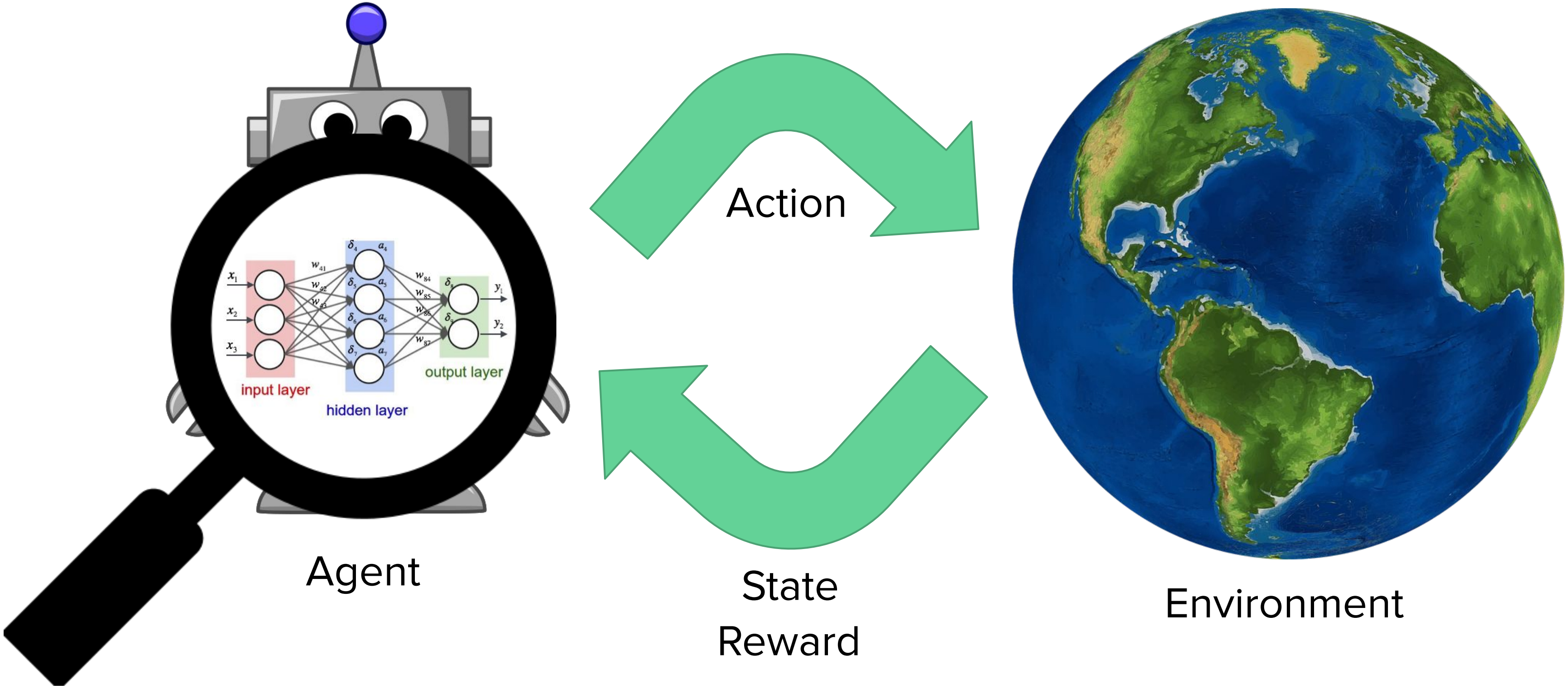
Reinforcement Learning…



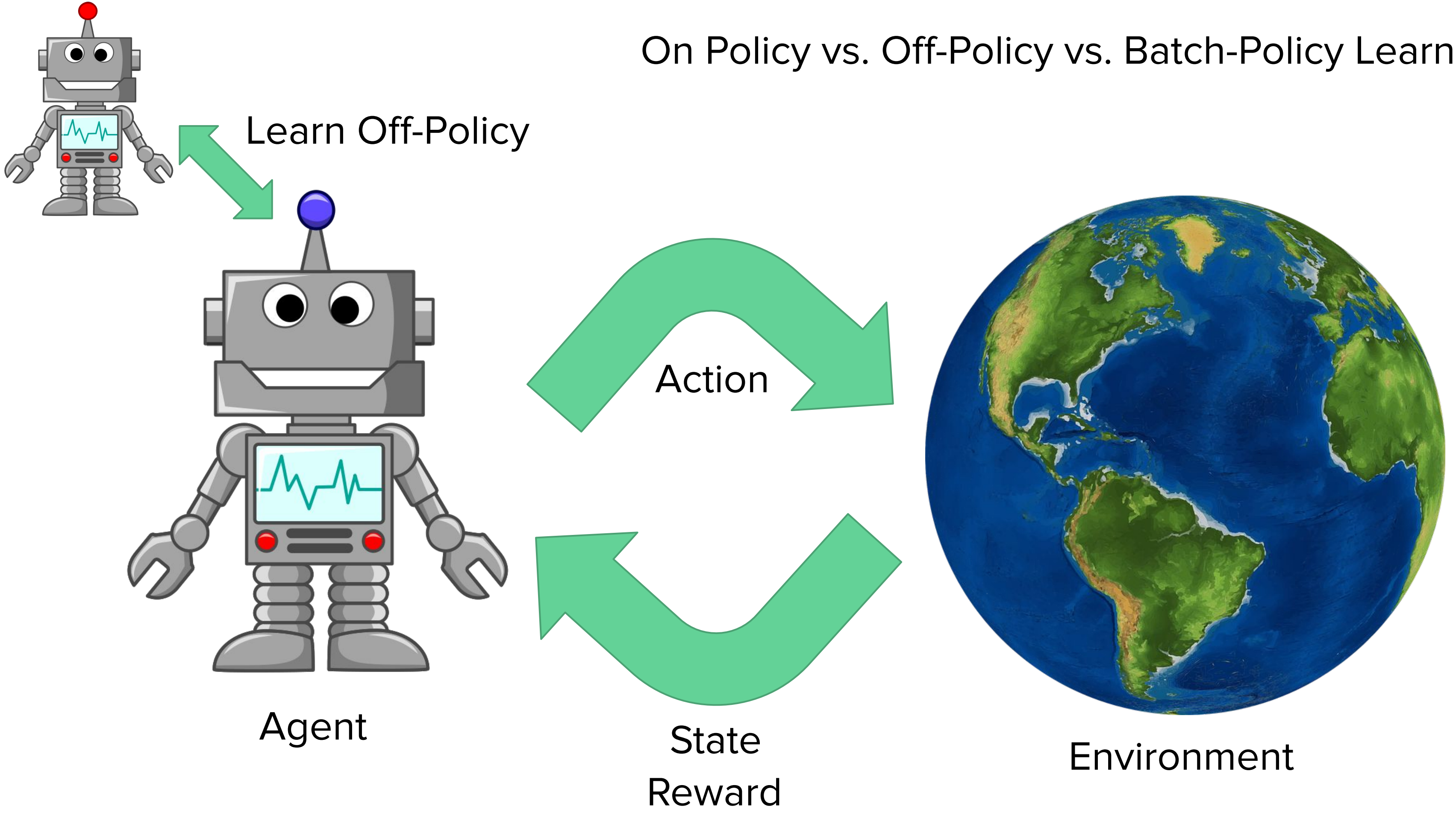



Action
Agent
State
Reward
Environment

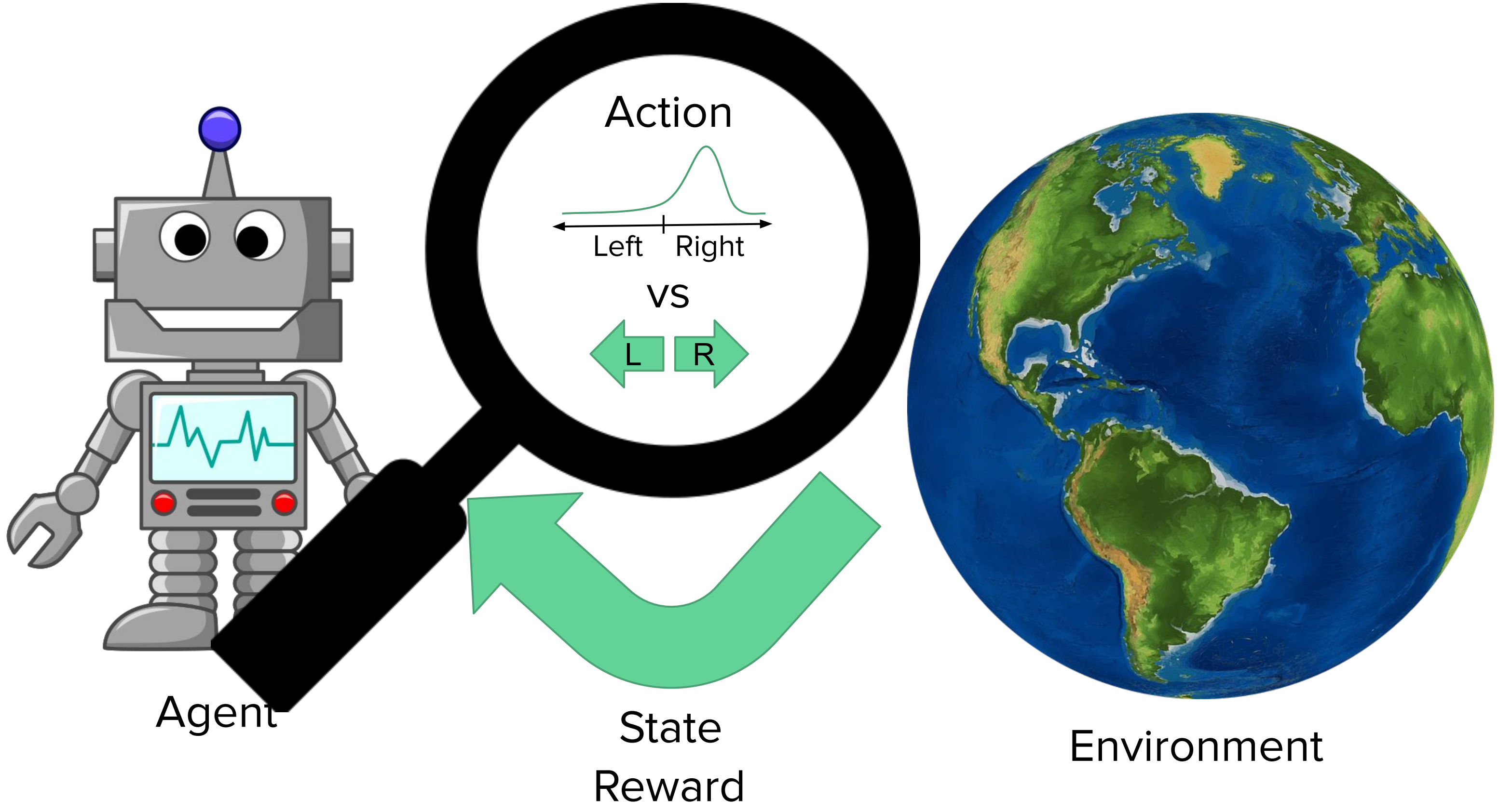
Deep Learning and Neural Architecture



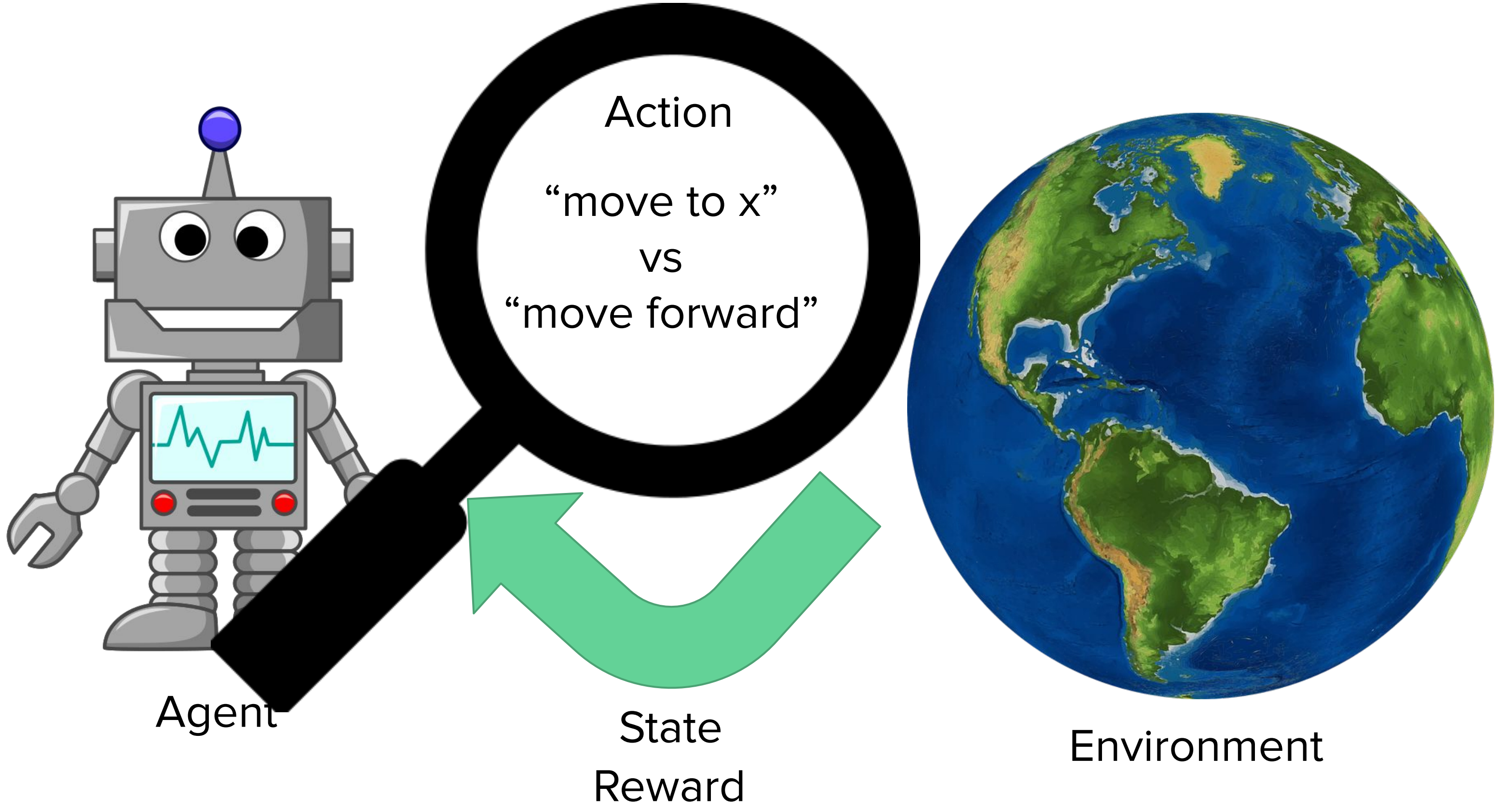
On Policy vs. Off-Policy vs. Batch-Policy Learning



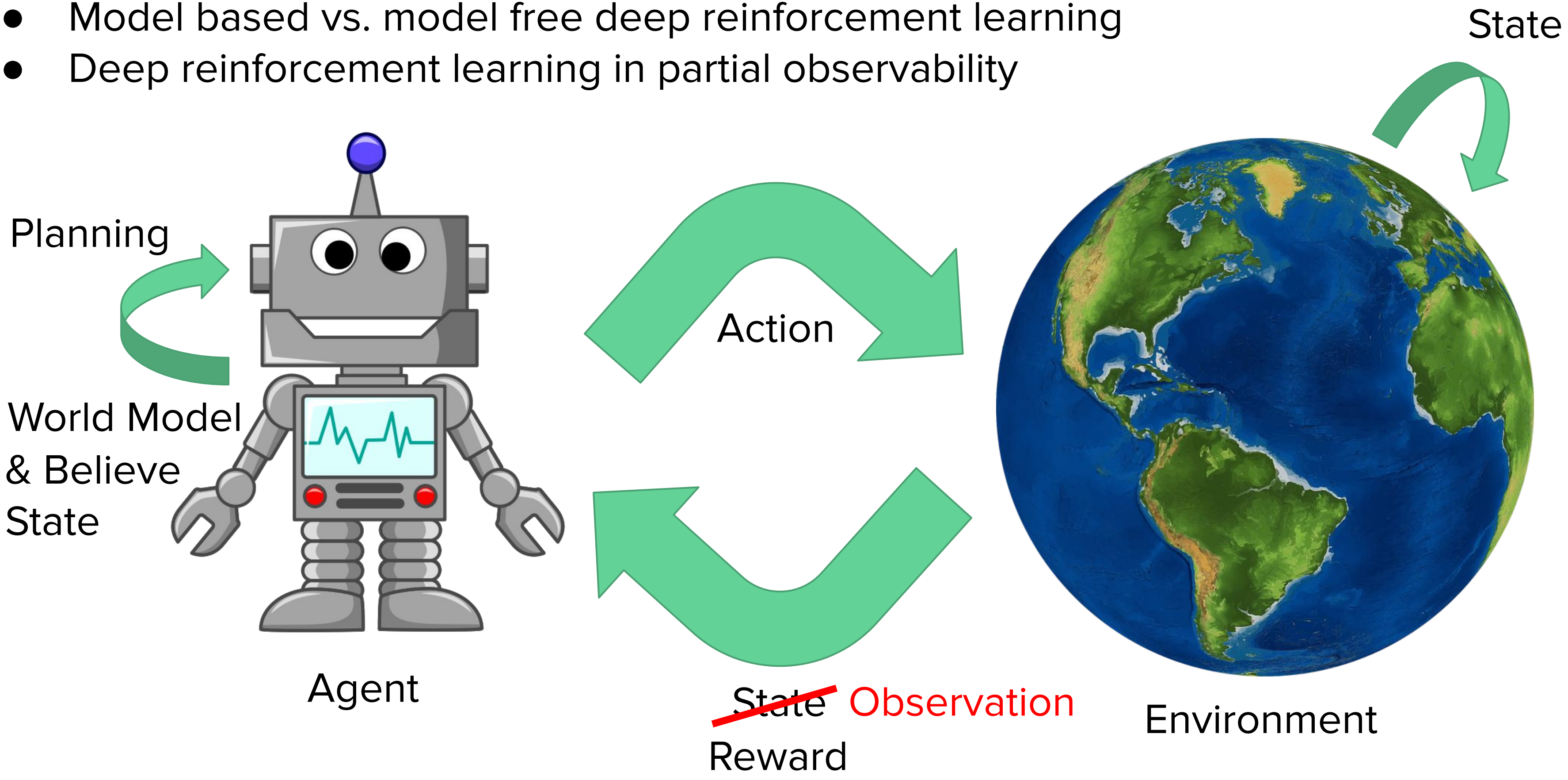
Deep reinforcement learning in continuous action spaces



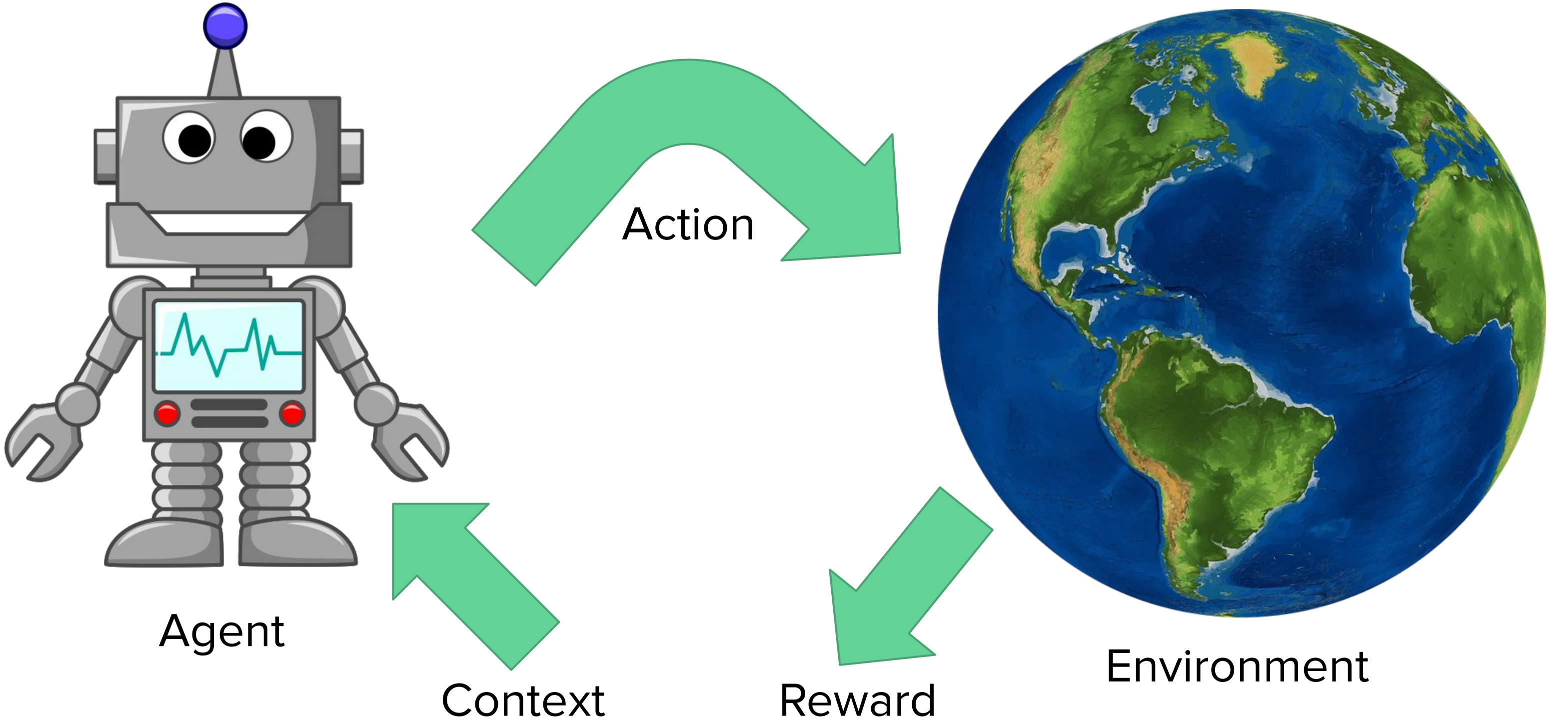
Hierarchical deep reinforcement learning



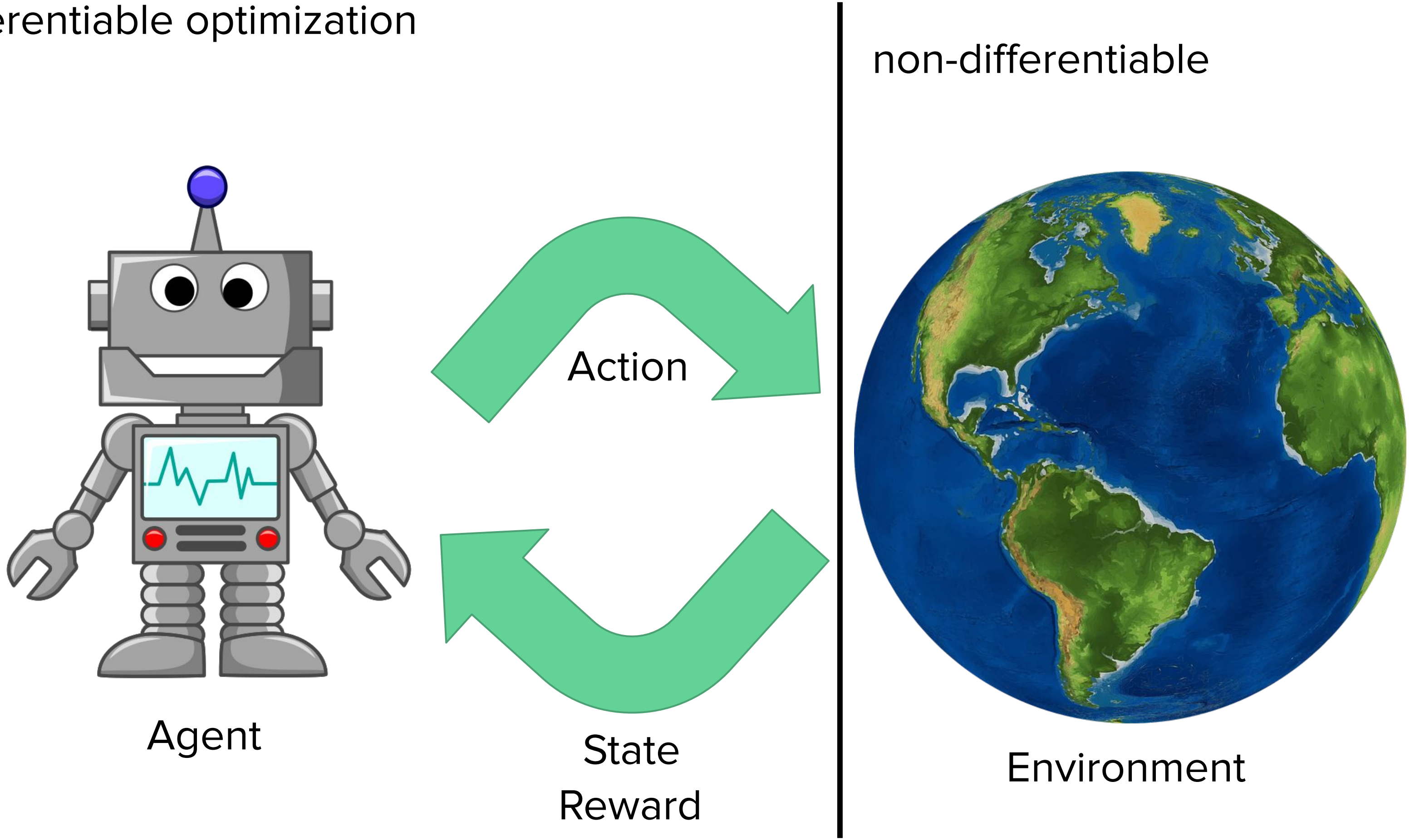
- Deep reinforcement learning and stochastic planning in games
- Model based vs. model free deep reinforcement learning
- Deep reinforcement learning in partial observability



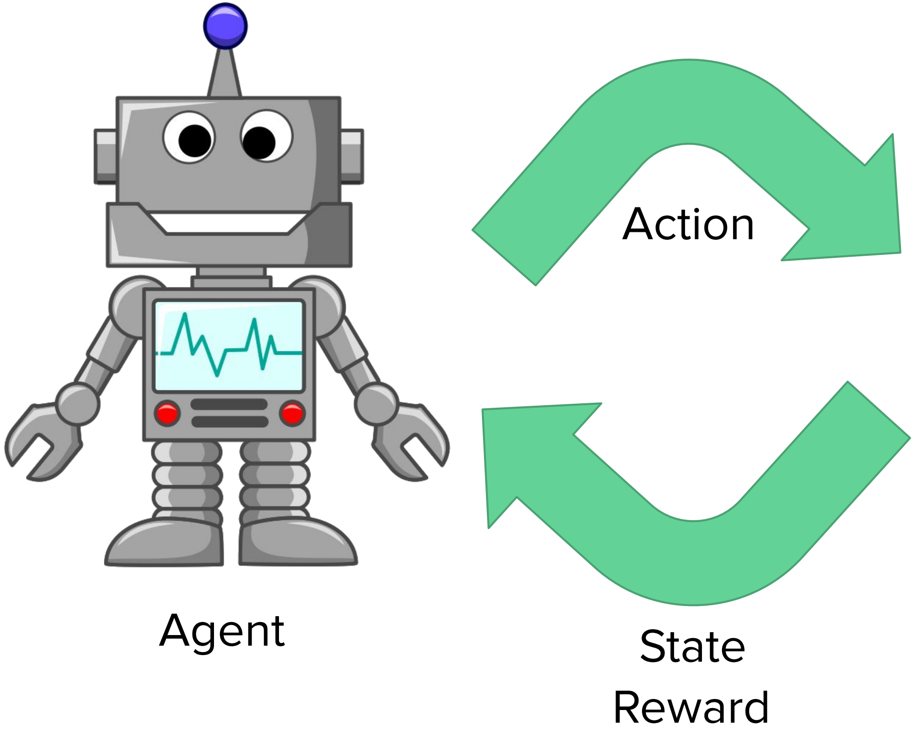
Multi-Armed Bandits



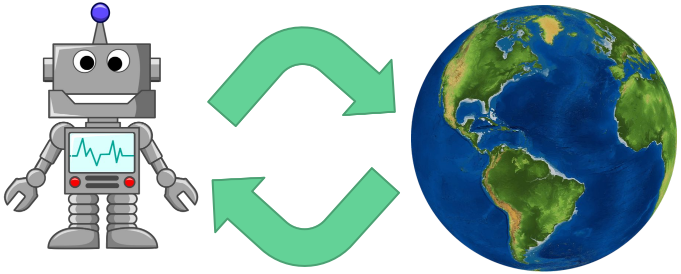
Non-differentiable optimization



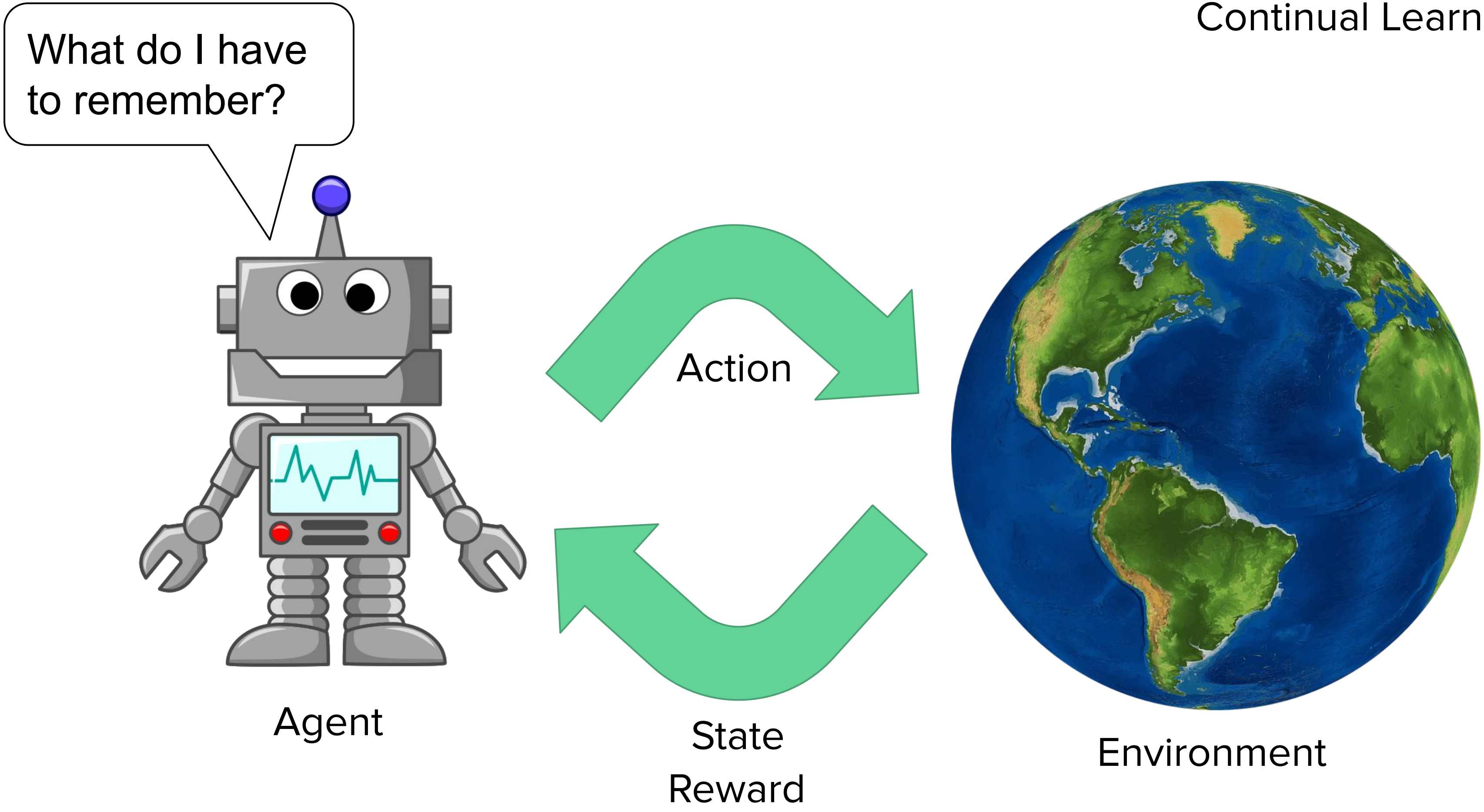
Meta-Learning



non-differentiable



Continual Learning



Task?

Maximize the discounted cumulative reward in each episode by finding a good policy

Task?

Maximize the discounted cumulative reward in each episode by finding a good policy

Task?

Maximize the **discounted cumulative reward** in each episode by finding a good policy

$$R = \sum_t \gamma^t r_t$$

$$\gamma \in (0, 1]$$

Task?

Maximize the discounted cumulative reward in each **episode** by finding a good policy

$$R = \sum_{t=0}^{T_e} \gamma^t r_t$$

$$\gamma \in (0, 1]$$

Task?

Maximize the discounted cumulative reward in each episode by finding a good **policy**

$$R^{\pi} = \sum_{t=0}^{T_e} \gamma^t r_t$$

$$\pi(a|s_t) = Pr(a|s_t)$$

How?

Estimate remainder of R^{π} in each state s_t

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[\sum_{t'=t}^{T_e} \gamma^{t'-t} r_{t'}]$$

$$Q^{\pi}(s_t, a_t) = r_t | a_t + \mathbb{E}_{\pi|a_t}[\sum_{t'=t+1}^{T_e} \gamma^{t'-t} r_{t'}]$$

$$\begin{aligned} V^\pi(\boldsymbol{s}_t) &= \mathbb{E}_\pi\left[\sum_{t'=t}^{T_e} \gamma^{t'-t} \boldsymbol{r}_{t'}\right] \\ &= \mathbb{E}_\pi[\boldsymbol{r}_t] + \gamma \mathbb{E}_\pi\left[\sum_{t'=t+1}^{T_e} \gamma^{t'-t-1} \boldsymbol{r}_{t'}\right] \\ &= \mathbb{E}_\pi[\boldsymbol{r}_t] + \gamma V^\pi(\boldsymbol{s}_{t+1}) \end{aligned}$$

$$Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \boldsymbol{r}_t | \boldsymbol{a}_t + \mathbb{E}_{\pi | a_t}\left[\sum_{t'=t+1}^{T_e} \gamma^{t'-t} \boldsymbol{r}_{t'}\right]$$

$$= \boldsymbol{r}_t | \boldsymbol{a}_t + \gamma V^{\pi}(\boldsymbol{s}_{t+1})$$

$$V^{\pi}(\boldsymbol{s}_t) = \mathbb{E}_{a \sim \pi}[Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a})]$$

Policy?

$$\pi^{greedy}(a|\boldsymbol{s}_t) = \mathbf{1}_{a=\max_{a'} Q^*(\boldsymbol{s}_t, a')}$$

Q-Learning

Watkins (1989)

$$Q^{greedy}(s_t, a_t) = r_t|a_t + \gamma \max_{a'} Q^{greedy}(s_{t+1}, a')$$

↑

iff $Q^{greedy} \equiv Q^*$

$$y(s_t, a_t) := r_t|a_t + \gamma \max_{a'} \tilde{Q}(s_{t+1}, a')$$

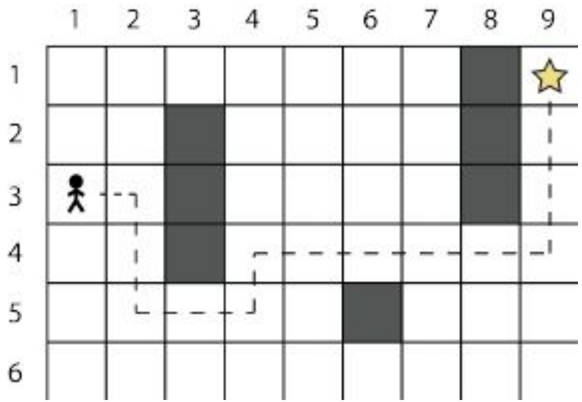
$$\delta_{TD} = y(s_t, a_t) - \tilde{Q}(s_t, a_t)$$

$$\rightarrow \text{minimize } \delta_{TD}^2$$

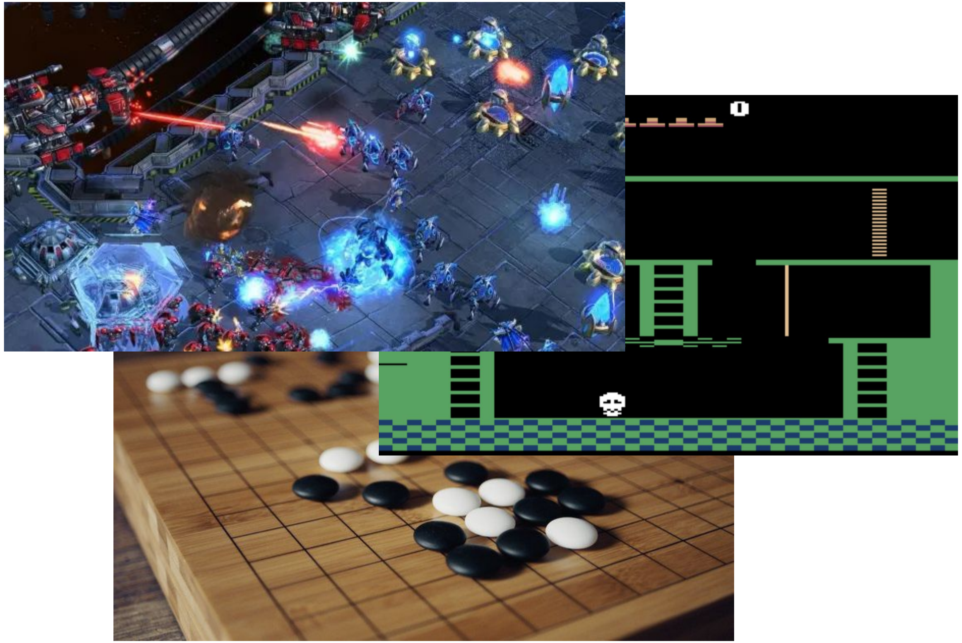
Classical RL vs Deep RL

estimate $\tilde{Q}(s, a)$

$\forall (s, a) \in \mathcal{S} \times \mathcal{A}$

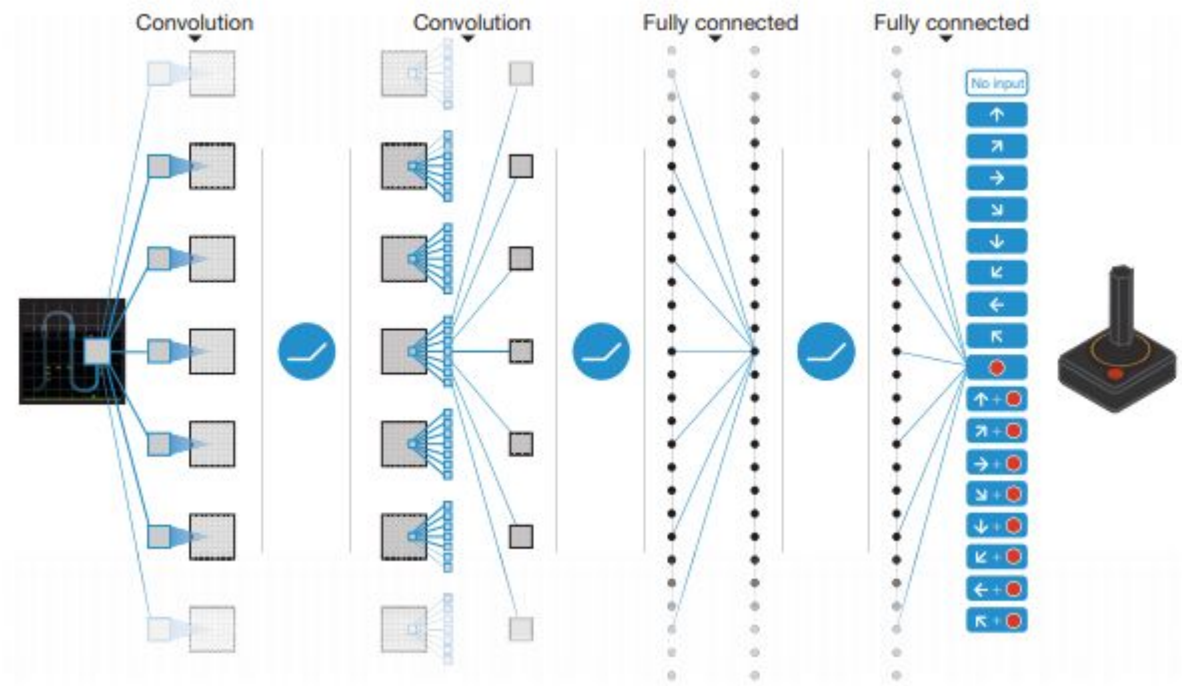


vs.



$\rightarrow$ approximate $\tilde{Q}(\cdot, \cdot)$

Human-level control through DRL (Mnih et al., 2015)



DQN

Deep Learning works for... RL?

- ...~~large data~~ sets...
  many interactions
- ...with ~~labeled~~ data points...
  self-labeled
- ...which are iid

➜ target network

➜ replay buffer

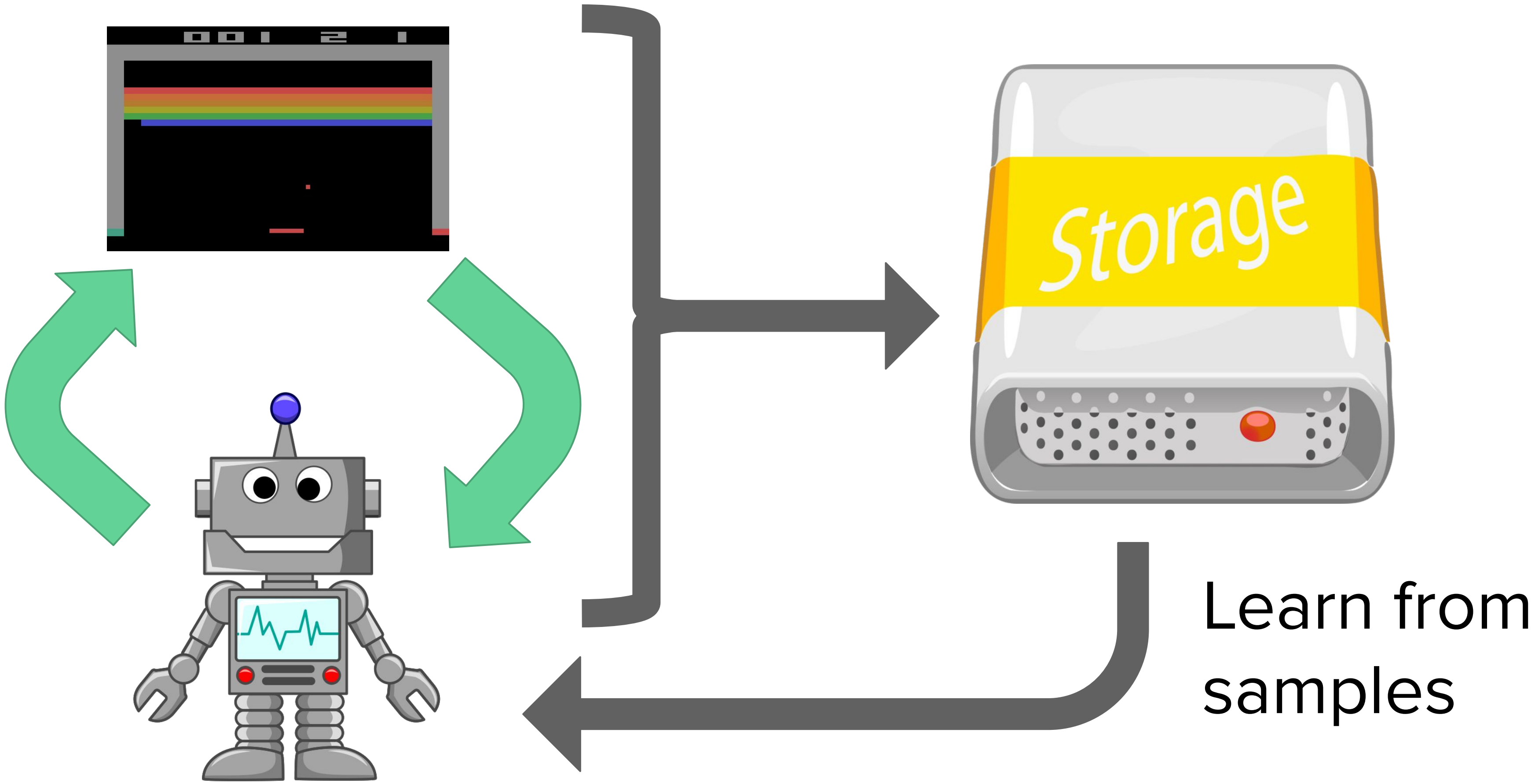
Target Network

$$y(s_t, a_t) := r_t | a_t + \gamma \max_{a'} \tilde{Q}_{\theta^-}(s_{t+1}, a')$$

$$\delta_{TD} = y(s_t, a_t) - \tilde{Q}_\theta(s_t, a_t)$$

$$\rightarrow \text{minimize } \delta_{TD}^2$$

Replay Buffer



Task?

Maximize the discounted cumulative reward in each episode by finding a good policy

$$R^{\pi} = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T_e} \gamma^t r_t\right]$$

$$\pi_\theta(a|s_t) = Pr(a|s_t;\theta)$$

$$\max_\theta R^{\pi_\theta}$$

$$\rightarrow \theta_{k+1} = \theta_k + \alpha \nabla_\theta R^{\pi_\theta}$$

$$\nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)] = ?$$

Policy Gradient Derivation

$$
\begin{aligned}
\nabla \mathbb{E}_{\pi}[R(\tau)] &= \nabla \int R(\tau)\pi(\tau)d\tau \\
&= \int R(\tau)\nabla\pi(\tau)d\tau \\
&= \int R(\tau)\pi(\tau)\frac{\nabla\pi(\tau)}{\pi(\tau)}d\tau \\
&= \int R(\tau)\pi(\tau)\nabla \log \pi(\tau)d\tau \\
&= \mathbb{E}_{\pi}[R(\tau)\nabla \log \pi(\tau)]
\end{aligned}
$$

Markov

$$\pi_\theta(\tau) = \mathcal{P}(s_0) \prod_{t=0}^{T_e} \pi_\theta(a_t | s_t) p(s_{t+1} | s_t, a_t)$$

$$\rightarrow \nabla_\theta \log \pi_\theta(\tau) = \sum_{t=0}^{T_e} \nabla_\theta \log \pi_\theta(a_t | s_t)$$

$$\mathbb{E}_\pi [R(\tau) \nabla \log \pi(\tau)]$$

$$= \mathbb{E}_\pi [(\sum_{t=0}^{T_e} \gamma^t r_t)(\sum_{t=0}^{T_e} \nabla \log \pi(a_t | s_t))]$$

$$\mathbb{E}_\pi[(\sum_{t=0}^{T_e} \gamma^t r_t)(\sum_{t=0}^{T_e} \nabla \log \pi(a_t|s_t))]$$

Causality

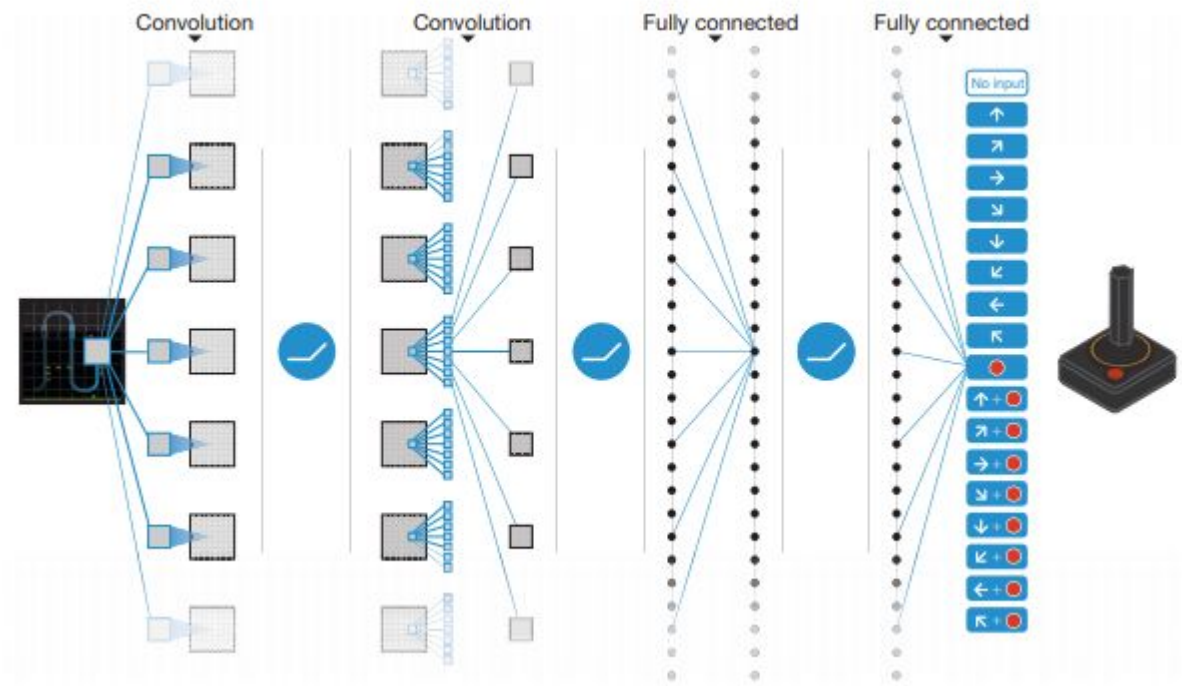
$$= \mathbb{E}_\pi[\sum_{t=0}^{T_e}(\sum_{t'=t}^{T_e} \gamma^{t'-t} r_{t'})\nabla \log \pi(a_t|s_t)]$$

$$= \mathbb{E}_\pi[\sum_{t=0}^{T_e} V^\pi(s_t)\nabla \log \pi(a_t|s_t)]$$

b independent of action

$$= \mathbb{E}_\pi[\sum_{t=0}^{T_e}(V^\pi(s_t) - b)\nabla \log \pi(a_t|s_t)]$$

Asynchronous Methods for DRL (Mnih et al., 2016)



A3C

Asynchronous Methods for DRL (Mnih et al., 2016)

$$\max_\theta \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T_e} (V^{\pi_\theta}(s_t) - b) \log \pi_\theta(a_t|s_t)\right]$$

$$b = \tilde{V}_\phi(s_t)$$

$$V^{\pi_\theta}(s_t) \approx \sum_{t'=t}^{t+n} \gamma^{t'-t} r_{t'} + \gamma^n \tilde{V}_\phi(s_{t+n})$$

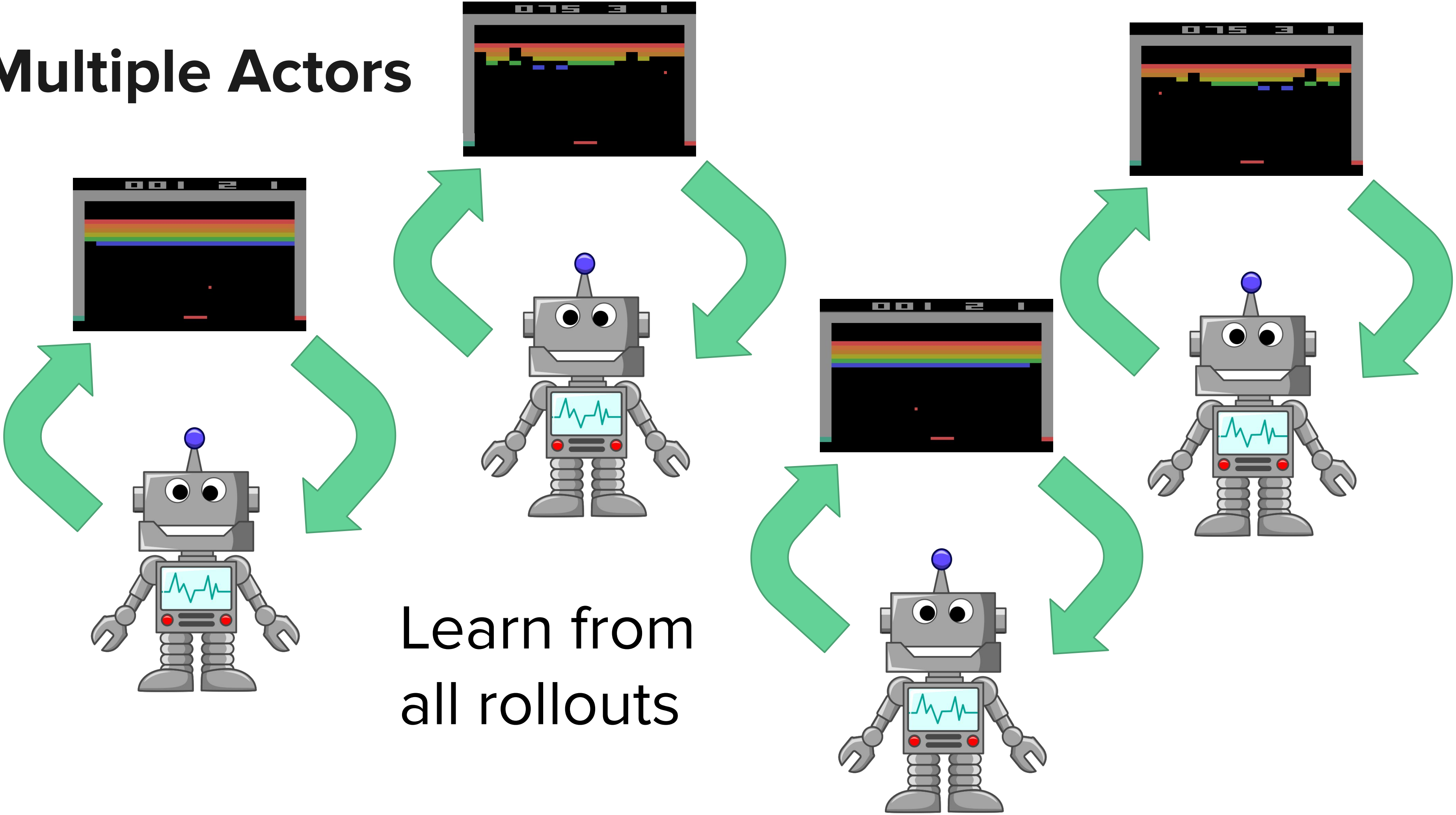
A3C

Deep Learning works for... RL?

- ...~~large data~~ sets...
  many interactions
- ...with ~~labeled~~ data points... → multi-step target
  self-labeled
- ...which are iid

→ multiple actors
→ entropy regularization

Multiple Actors



Learn from all rollouts

Entropy Regularization

... act as random as possible

$$\max_\theta \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T_e} (V^{\pi_\theta}(\boldsymbol{s}_t) - b) \log \pi_\theta(a_t | \boldsymbol{s}_t) \right.$$
$$\left. - \lambda \pi_\theta(a_t | \boldsymbol{s}_t) \log \pi_\theta(a_t | \boldsymbol{s}_t)\right]$$

DQN vs A3C

sample efficient	sample inefficient
slow to train	fast to train
(almost) deterministic	stochastic
only 1 network	2 (1.5) networks

Coding Challenge

https://github.com/OliverRichter/DRL_Seminar_BlackJack

References

Deep Reinforcement Learning Doesn't Work Yet, *Irpan Alex,* https://www.alexirpan.com/2018/02/14/rl-hard.html

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,* Nature 2015

Asynchronous Methods for Deep Reinforcement Learning, *Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, Koray Kavukcuoglu,* ICML 2016