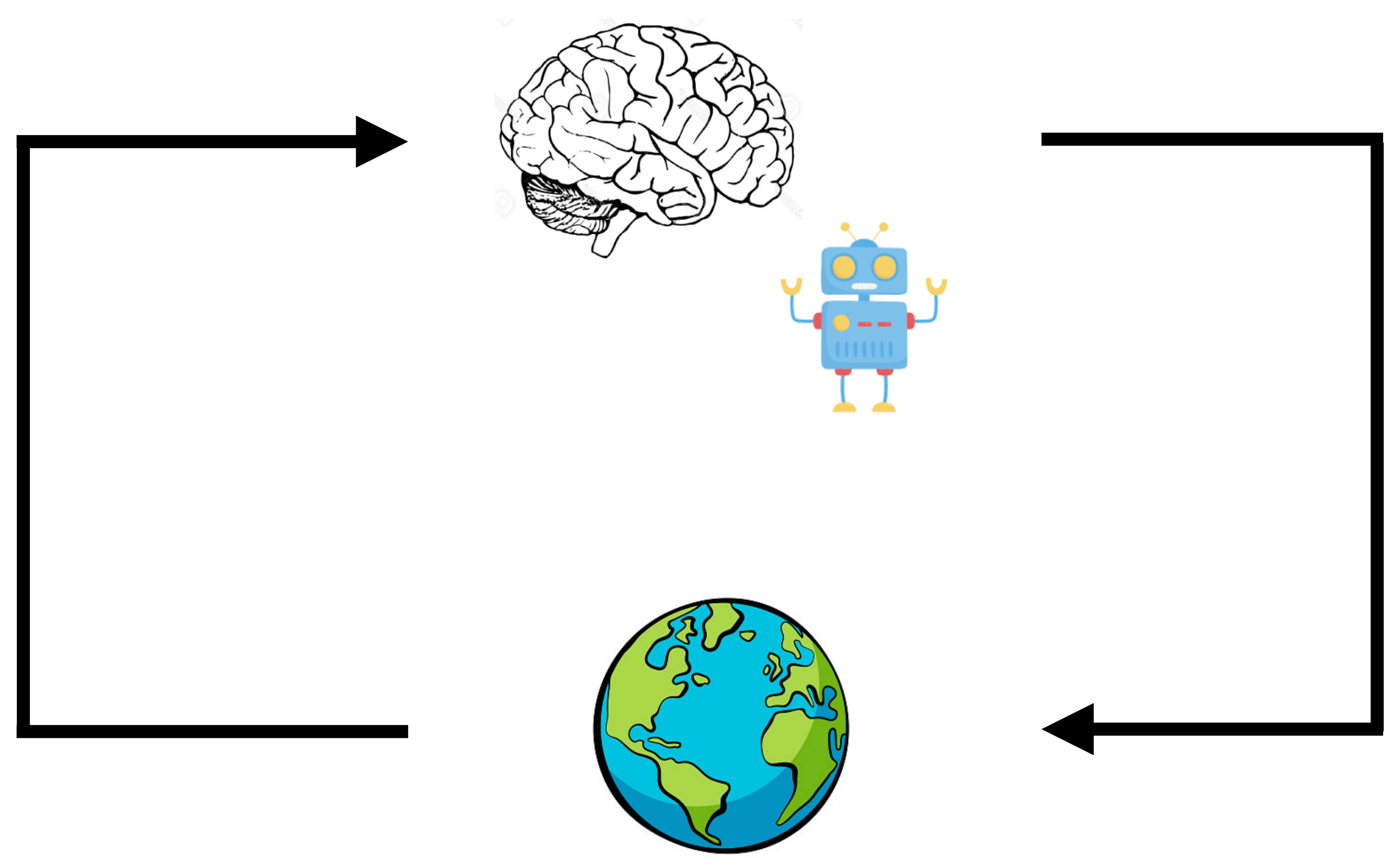
An introduction to Reinforcement Learning

31st of May 2022

Recap: Temporal Difference Learning

Based on a reward signal, agents learn **values of actions/states:**

$$V_\pi(s) = \mathbb{E}_\pi[R \mid s_0 = s]$$



TD Learning:

Prediction error

$$V(s_t) \leftarrow V(s_t) + \alpha \cdot (r + \gamma \cdot V(s_{t+1}) - V(s_t))$$

Learning rate Discount rate

Rescorla Wagner Learning:

Prediction error

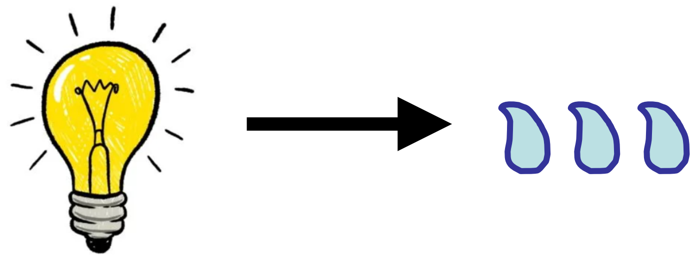
$$V(s_t) \leftarrow V(s_t) + \alpha \cdot (r - V(s_t))$$

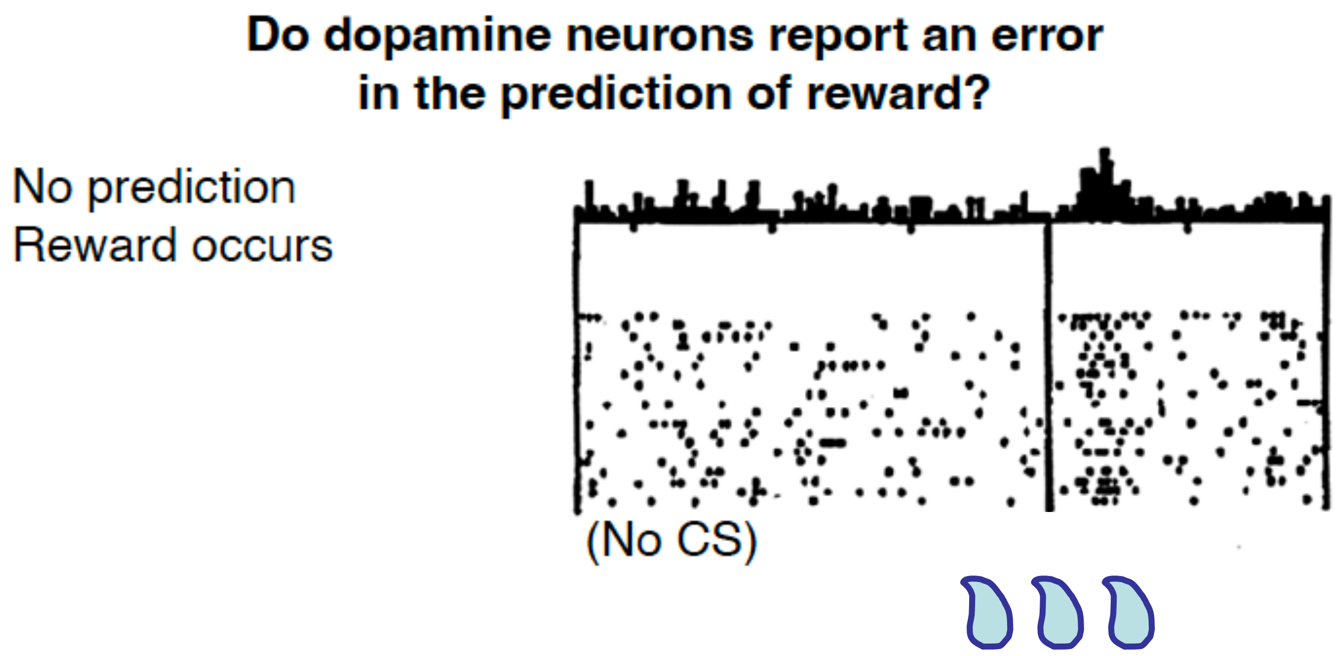
Learning rate

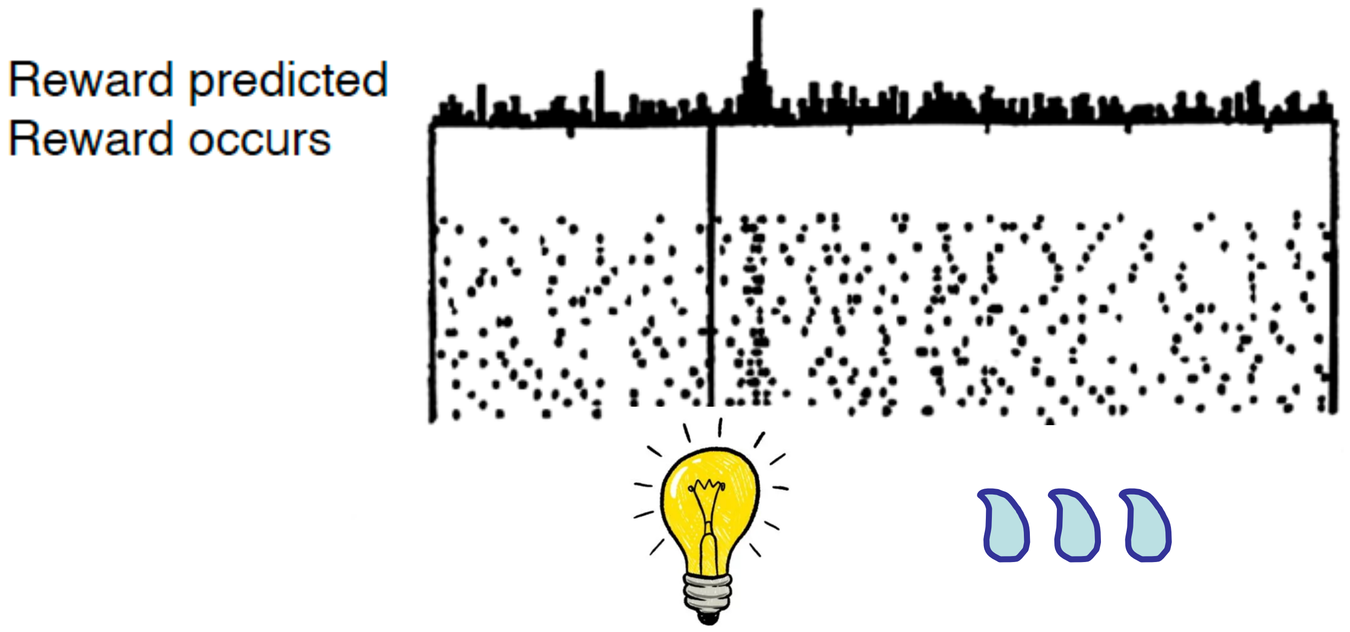
Recap: Can RL tell us anything about the brain?

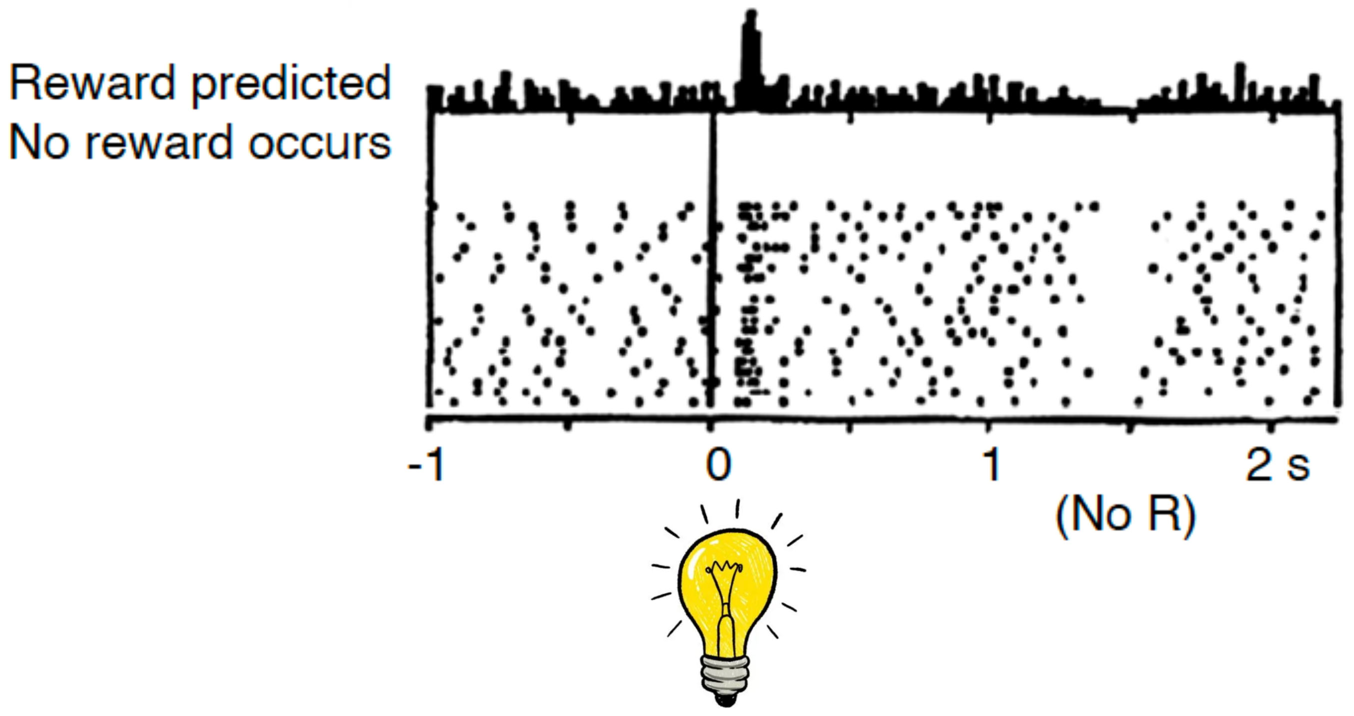
$$V(s_t) \leftarrow V(s_t) + \alpha \cdot (r + \gamma \cdot V(s_{t+1}) - V(s_t))$$

- It looks like DA signals the reward prediction error in TD learning (Schultz, Dayan & Montague Science, 1997)



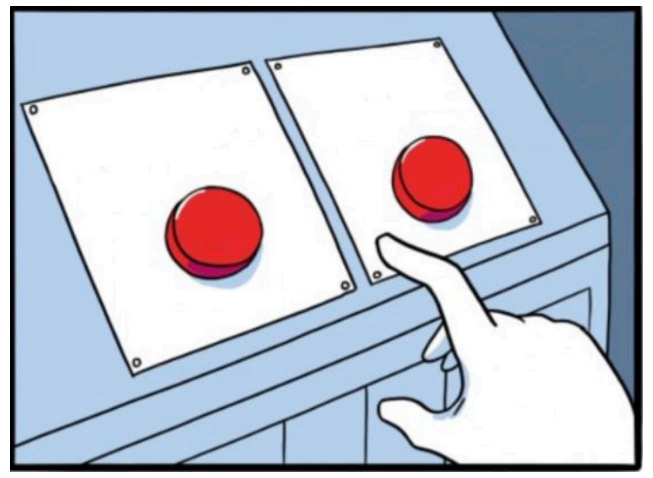




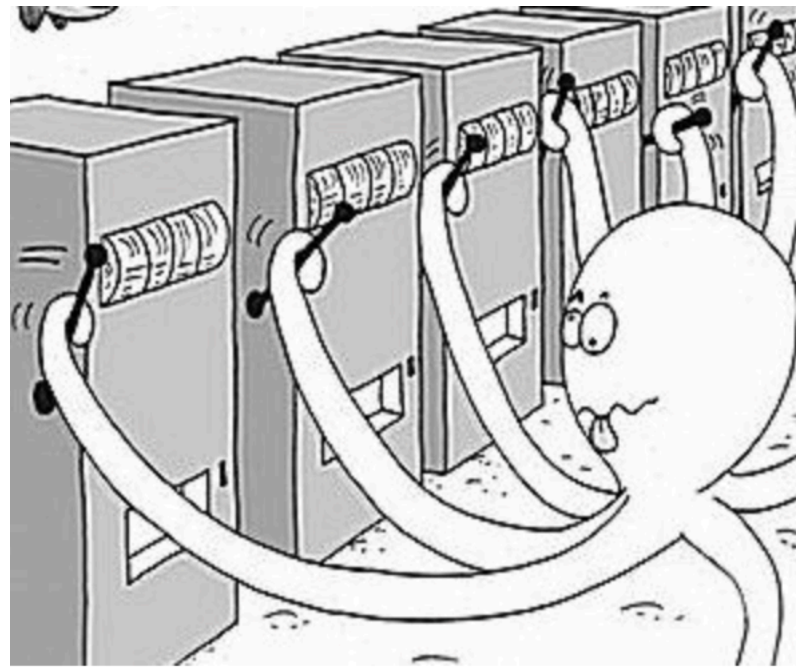


Coding: TD Learning

https://github.com/schwartenbeckph/RL-Course/tree/main/2022_05_24



Multi-armed bandits



- Problems where agents are faced with different options
  - Have to find out which of these are good or bad via trial-and-error
- Key problem: **exploitation vs. exploration**
  - **Random** vs. **goal-directed exploration**
- At the heart of many modern RL studies
  - Ideal testbed for different **models of action selection**
- Still in simplified RL setting
  - *Stationary* environment
  - Only consider *immediate reward* (for now)
  - Non-*contextual*
  - *Tabular*

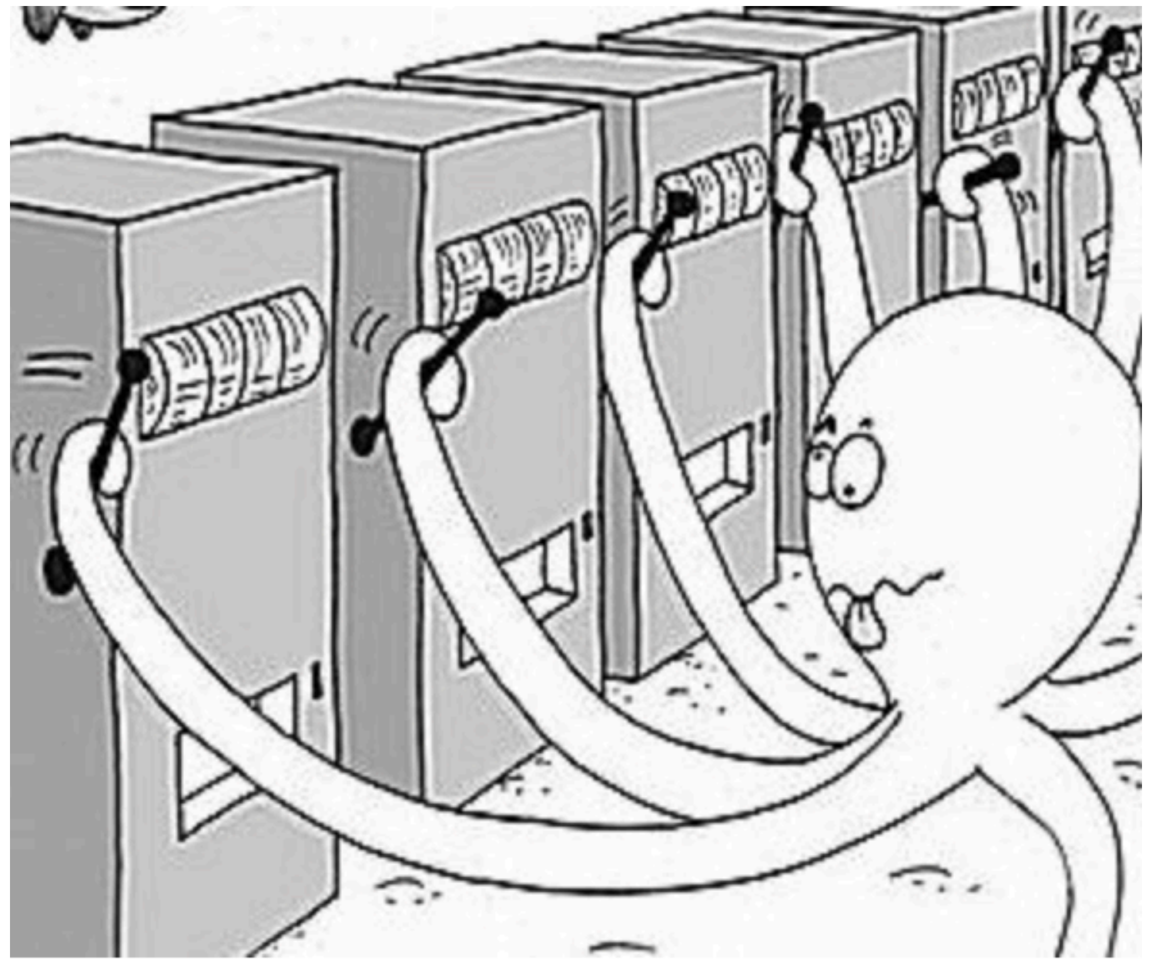
Multi-armed bandits

Greedy action selection:

$$P(a_t = a) = \begin{cases} 1 & \text{if } a_t = \text{argmax}_a V_t(a) \\ 0 & \text{otherwise} \end{cases}$$

Epsilon-greedy action selection:

$$P(a_t = a) = \begin{cases} 1 - \epsilon & \text{if } a_t = \text{argmax}_a V_t(a) \\ \epsilon / N & \text{otherwise} \end{cases}$$



Softmax action selection:

$$P(a_t = a) = \frac{e^{V_t(a) \cdot \beta}}{\sum_{i=1}^{N} e^{V_t(a_i) \cdot \beta}}$$

Action is governed by a **policy**:

$$\pi(a, s) = P(a_t = a \mid s_t = s)$$

Upper-confidence-bound (UCB) action selection:

$$P(a_t = a) = \text{argmax}_a [V_t(a) + c \cdot \sqrt{\frac{\ln t}{N_t(a)}}]$$

Coding: Multi-Armed Bandits

https://github.com/schwartenbeckph/RL-Course/tree/main/2022_05_31

Dates and topics

14.06.2022	• Q-learning, SARSA
21.06.2022	
	• Model-based RL
28.06.2022	
05.07.2022	• Applications
	• Model fitting, testing psych hypotheses
12.07.2022	• Deep RL
	• Current research
19.07.2022	
	• ‘Do your project session’?
26.07.2022	

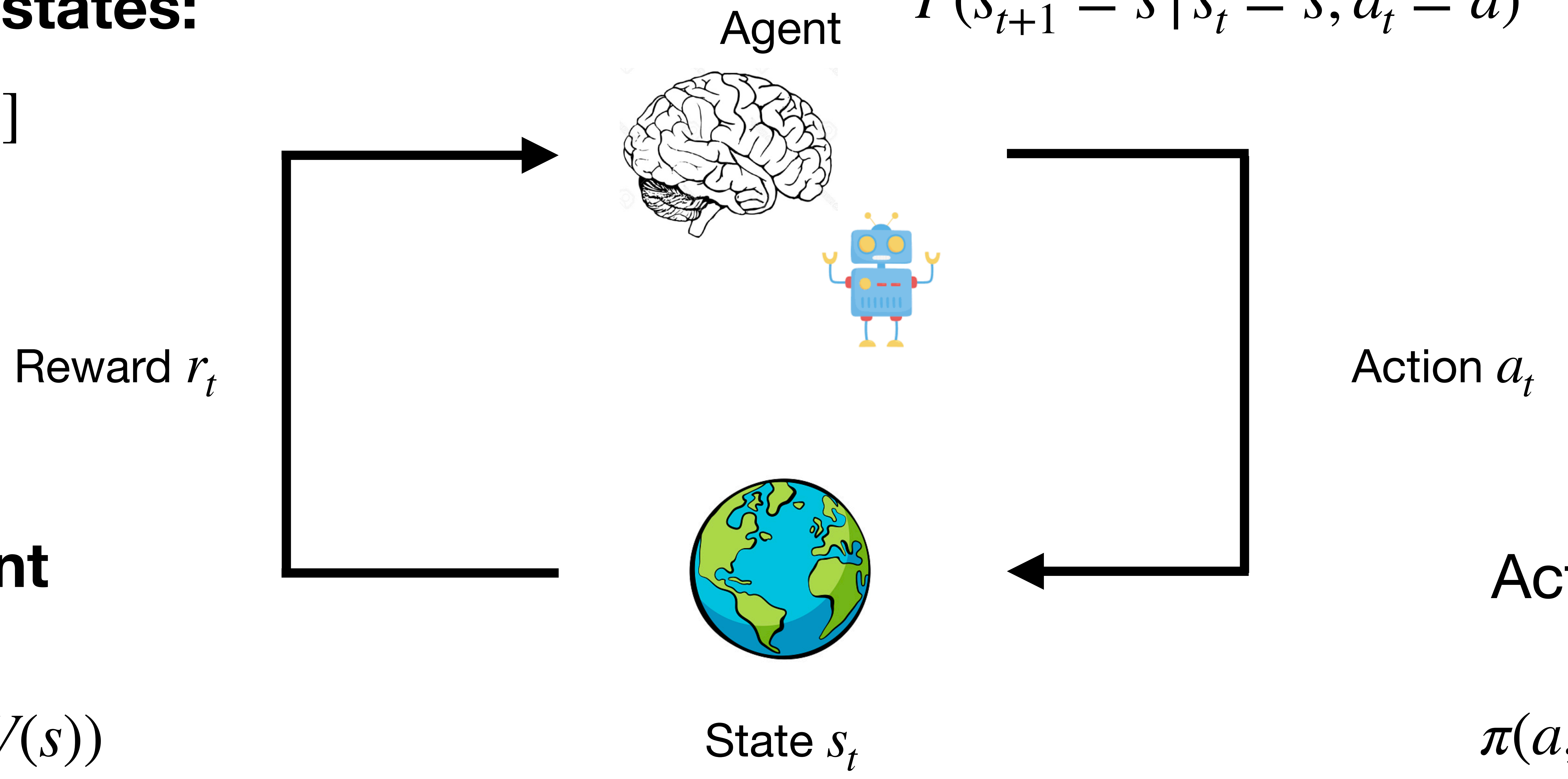
Recap: Basic setup: how to agents learn to act?

Based on a reward signal, agents learn **values of actions/states:**

$$V_\pi(s) = \mathbb{E}_\pi[R \mid s_0 = s]$$

Agents can learn a **model of the environment** to make smarter decisions, e.g.:

$$P(s_{t+1} = s \mid s_t = s, a_t = a)$$



Values can be **learnt** (simplified!!):

$$V(s) \leftarrow V(s) + \alpha \cdot (r - V(s))$$

Learning rate

Prediction error

Action is governed by a **policy**:

$$\pi(a, s) = P(a_t = a \mid s_t = s)$$