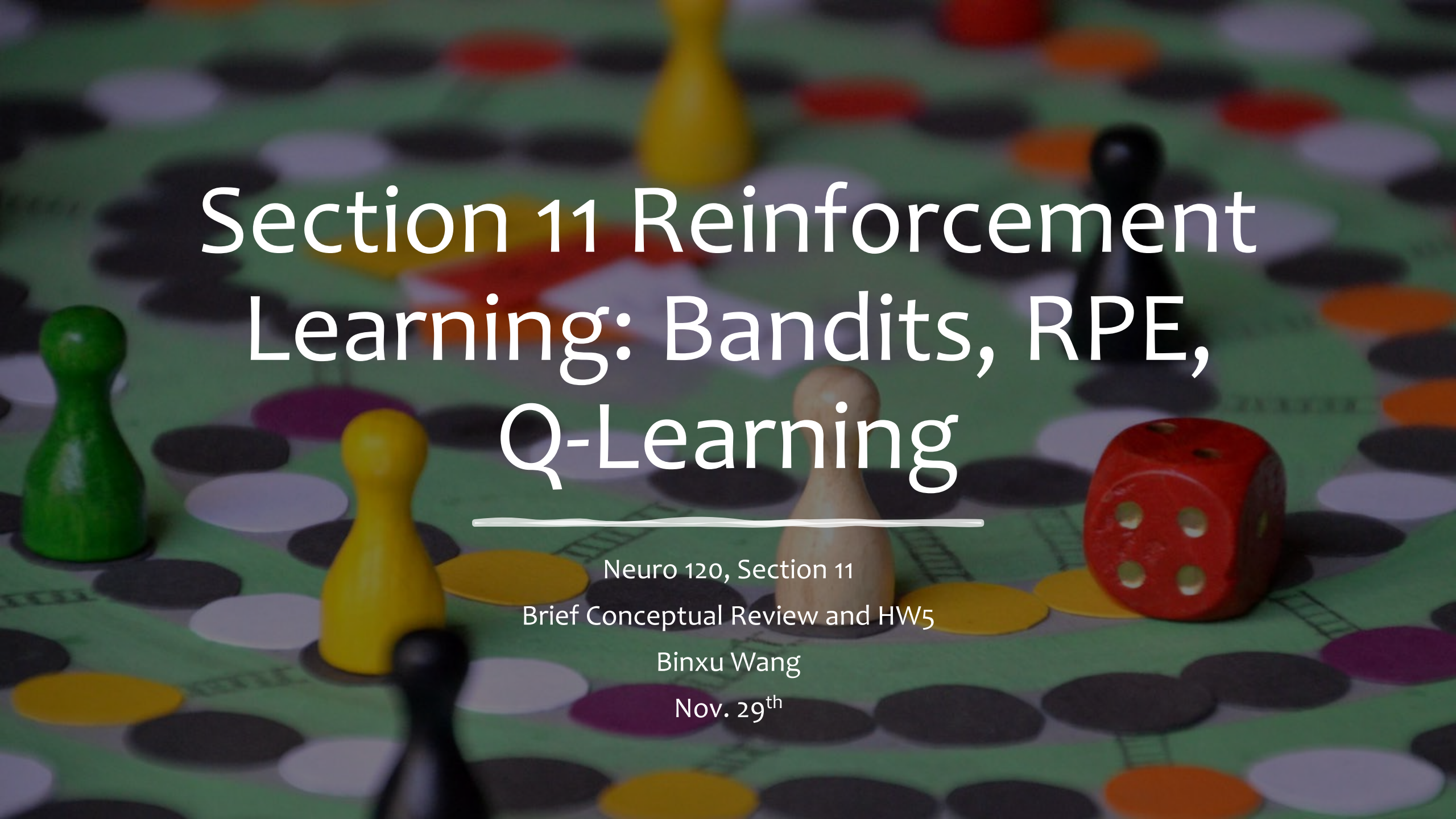


Section 11 Reinforcement Learning: Bandits, RPE, Q-Learning

Neuro 120, Section 11

Brief Conceptual Review and HW5

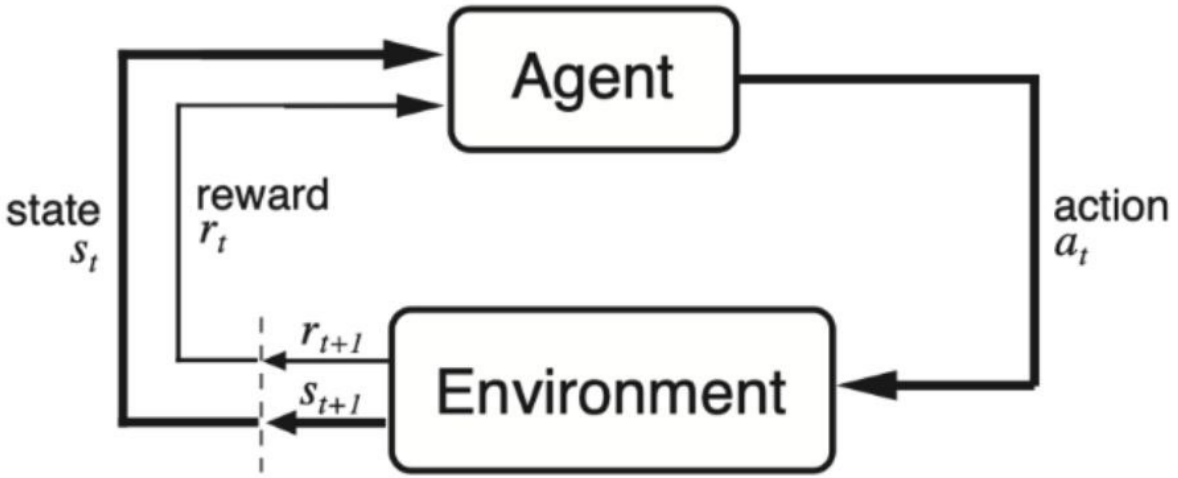
Binxu Wang

Nov. 29th

Key Concepts

Components of a RL Problem

- Agent
  - Act on environment
- Environment
  - Change state depending on action
  - Provide rewards as feedback.

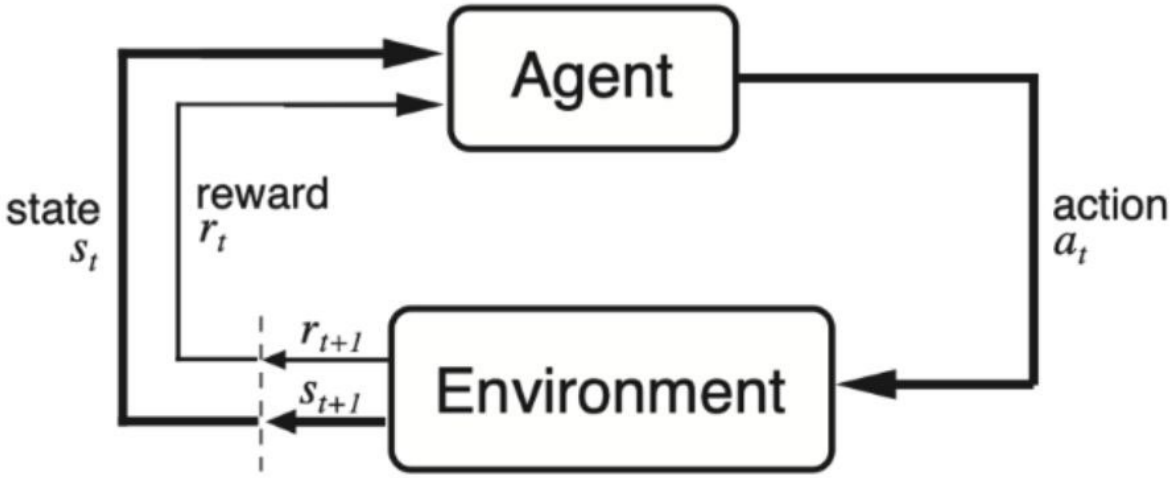


- Usually, agents are limited in their ability to control state…
  - Cannot set the state arbitrarily.

How to formulate agent, environment interaction?

State, Action and Reward

- State transition $p(s_{t+1}|s_t, a_t)$
  - *Laws of Physics*
  - How state change if you act on it.
  - If deterministic, we can write
    $$s_{t+1} = T(s_t, a_t)$$
- Reward function $\mathrm{r_t} = r(s_t, a_t)$
  - What you get by doing sth. at a state.
- Policy: $\pi(a_t|s_t)$
  - *Decision making / Action selection*
  - How you act based on your state.



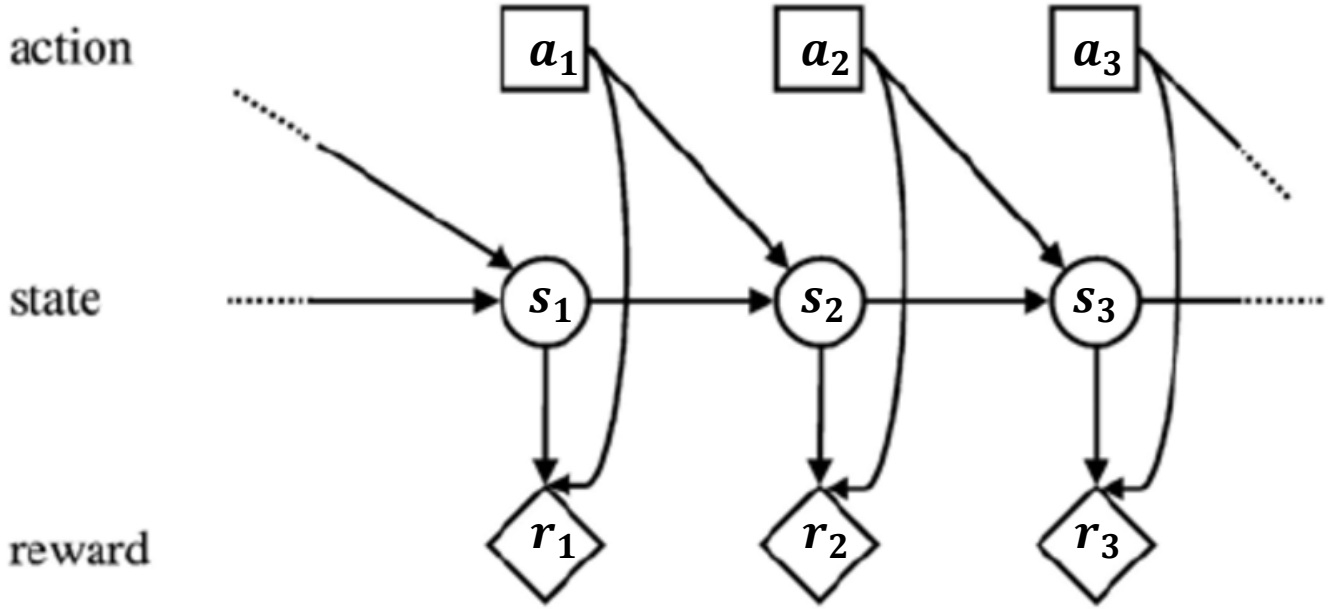
Markovian property

- Reward and future is fully determined by the state and future action. Not your past.
  - $p(s_{t+1}|s_t, a_t)$
  - $r(s_t, a_t)$

State-Action Chain

- A **trajectory** is a chain of state and action
  - $s_0, a_0, s_1, a_1, s_2 \ldots s_t, a_t, s_{t+1}$
- In the meantime, you get a chain of reward
  - $r_0, r_1, r_2, r_3 \ldots.$
- You'd like to maximize the total reward in an episode!
  - $R = \sum_{t=0}^{T} r_t$

- Episode and horizon T
  - How far are you planning and terminating the reward calculation?
  - E.g. maximize "reward" in a course, in the college, in your whole life…
  - Different horizon may prefer different strategy (~ Exploration, Exploitation)



Comparing Feedback with Supervised Learning

Supervised Learning

- Given a stimuli s, you get the "correct" / best response y
  - Target is known.
  - *Instructive*

Reinforcement Learning

- Given a state s, you get some reward r for your action a
  - Best, optimal action is not known beforehand.
  - *Evaluative*
  - *Open-ended, potentially creative.*

Copying the behavior of master player is a way to solve RL – imitation learning (e.g. AlphaStar)

Comparing Evaluation with Supervised Learning

Supervised Learning

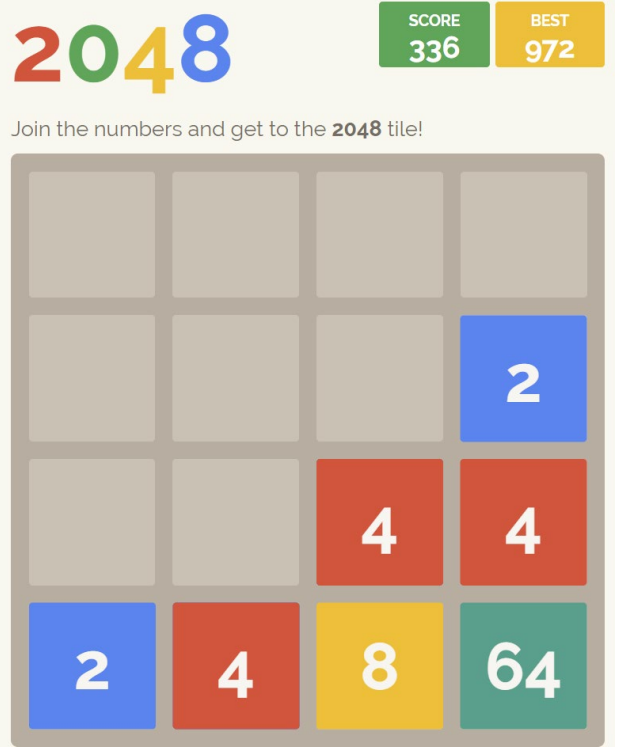
- We evaluate our model by its prediction performance on a fixed **test dataset**.

Reinforcement Learning

- To evaluate a policy, we need to interact with environment *repeatedly*.

Copying the behavior of master player is a way to solve RL – imitation learning (e.g. AlphaStar)

What are the states and actions?



Conceptually, what is state and action?

- What is state s
  - A representation / model of the environment.
  - An abstraction of relevant information, not everything.
- What is action a
  - **(Abstract) interface** between agent and environment.
  - May not contain all the details of action execution. (e.g. how a player put pawn on the board)
- The choices are arbitrary but important, when building an environment!

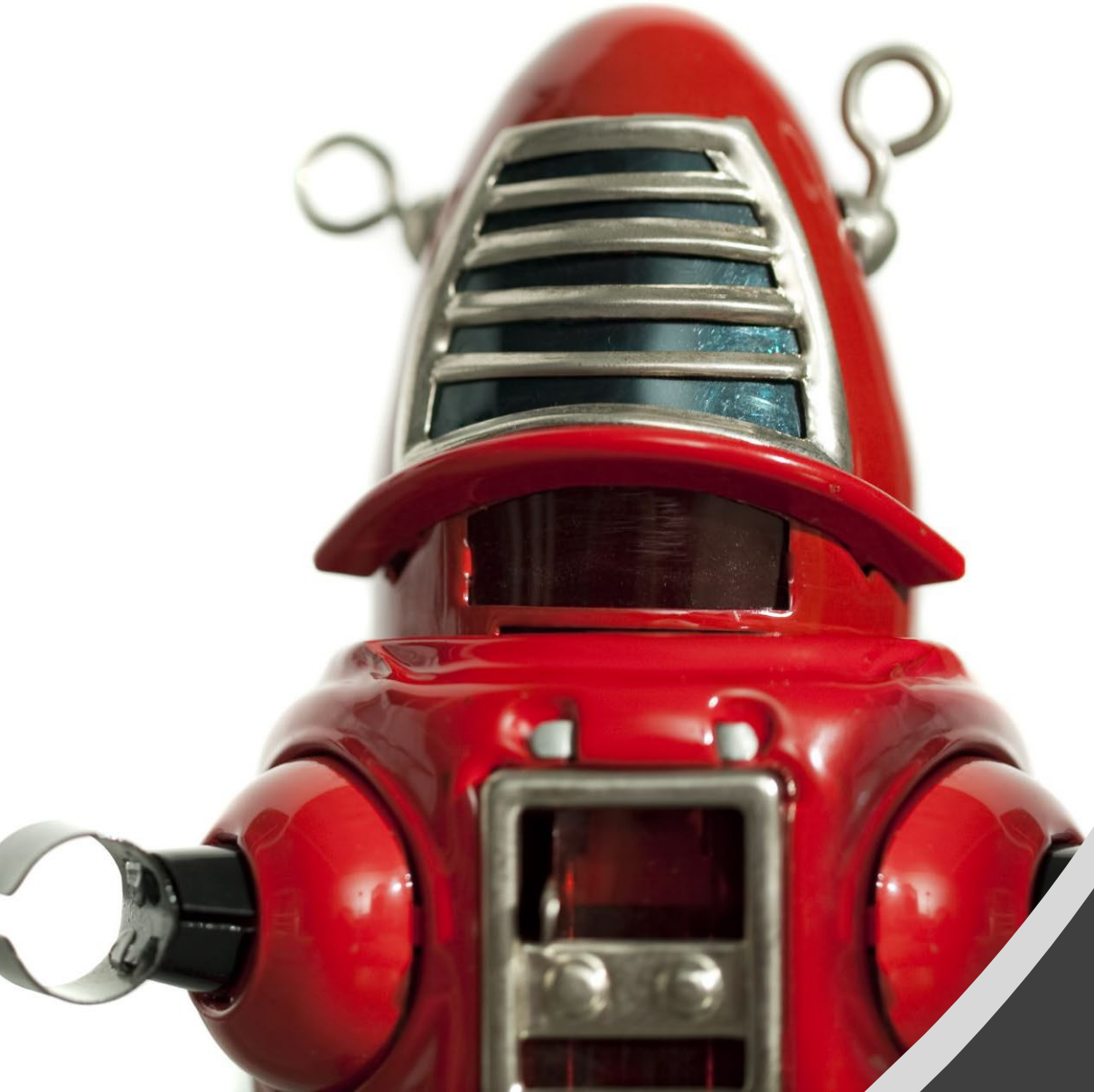
Reward, Value, Q Value

- Value function $V_\pi(s)$
  - Expected future reward at s, if we act using π
  - $V_\pi(s) = \sum_t r_t(s_t, a_t)$
    - $s_0 = s, a_t \sim \pi(s_t), s_{t+1} = T(s_t, a_t)$
- Q function $Q_\pi(s, a)$
  - Expected future reward at s with action a, if we act using π
  - $Q_\pi(s, a) = \sum_t r_t(s_t, a_t)$
    - $s_0 = s, a_0 = a; a_t \sim \pi(s_t), s_{t+1} = T(s_t, a_t)$
- Optimal Q, V: $Q_*(s, a), V_*(a)$
  - Expected reward from s, a when using the optimal strategy.

- Note
  - Policy π affects value V and Q value
    - E.g. s = getting into Harvard; s does not fully determine the future rewards, $s + \pi$ do.
  - $V_\pi(s)$, $Q_\pi(s, a)$ is not accessible to agent. Agent usually learn approximated model $\hat{V}, \hat{Q}$.

Why Value V, Q

- What if you know the ideal $Q_*(s, a)$ function? What should you do (best policy)?
  - Choose the highest value action at each s! $a = \arg\max_a Q_*(s, a)$
- What if you know the ideal $V_*(s)$ function?
  - Try to navigate to the high value state
- Why we need value? Not just reward?
  - Value is a model of all future reward. Help make long term decision.
  - *E.g. moving to Harvard has zero or negative reward, but it's a high value state…*



Multi-arm bandit

Multi-arm bandit problem

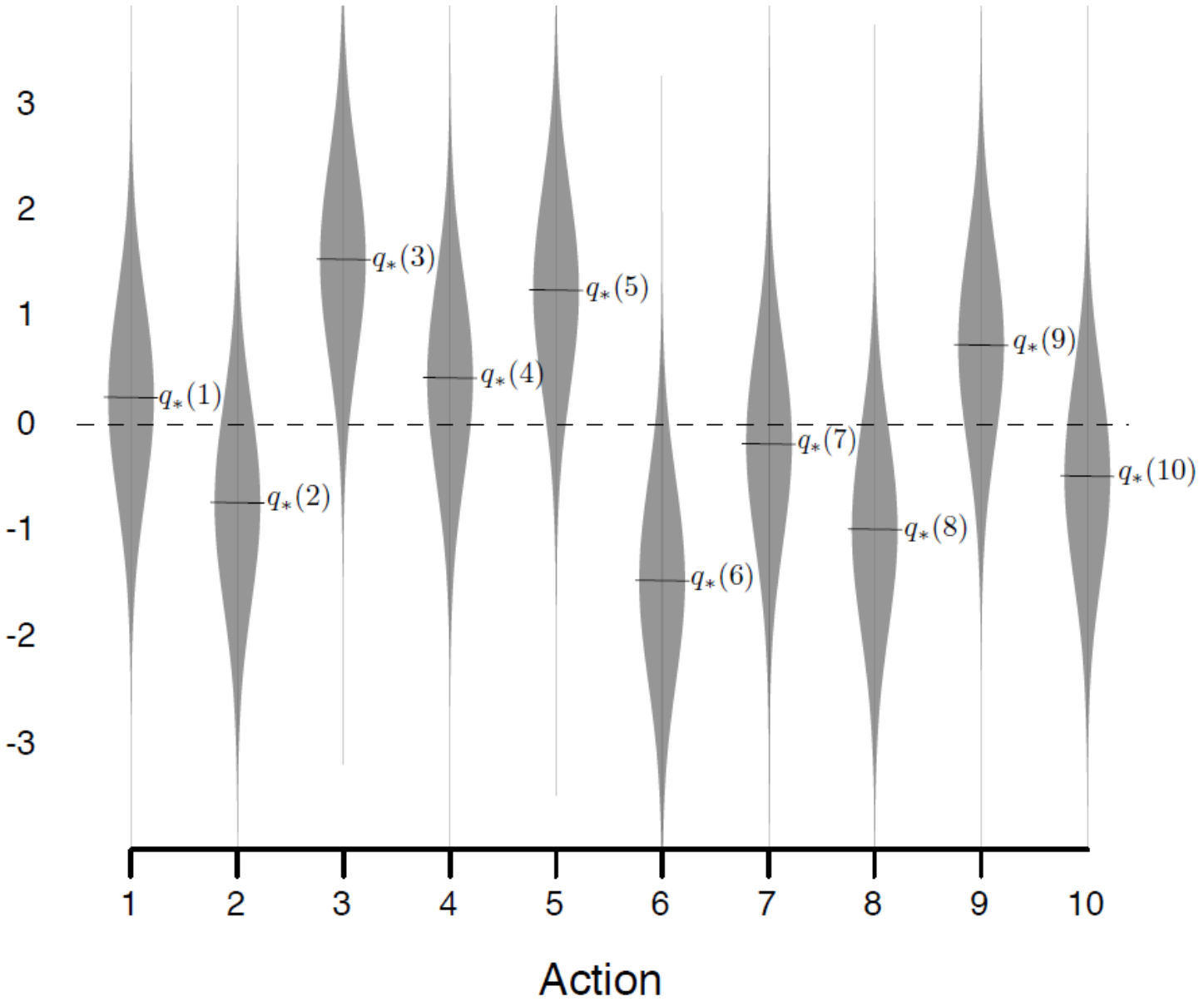
- Single state s_0, no change!
- Multiple actions $a \in \{a_1, a_2, a_3, \dots\}$ $(arms)$
- Each action a_i triggers reward from a distribution $r \sim P_i(r)$
  - Each action has an exact expected value
    $$q_*(a_i) = \mathbb{E}(r \mid a_i)$$
  - Q is only a function of action.

- Problem:
  - How to maximize reward with a fixed number of actions ?

Example 10-arm bandit

- What's the best strategy at this bandit machine?
- Bandit problem is *trivial* if you know $P_i(r)$ and true $q_*(a_i)$
- This knowledge needs to be **acquired through interaction!**

Reward Distribution $P_i(r)$ for Actions



Exploration Exploitation Tradeoff

- In decision making, there are roughly two stages
  - Learn about the actions; estimate their value $\hat{q}(a_i)$
  - Choose action based on their value $\hat{q}(a_i)$
- Correspond to two "types" of action
  - **Explore:** try new / less-explored actions
  - **Exploit:** gain reward from explored actions

Estimate Value by Collecting Data

- For bandit problem, we just can average the reward for this action

$$\hat{q}(a_i) = \frac{\sum_{t \text{ where } a_t = a_i} r_t}{\sum_{t \text{ where } a_t = a_i} 1}$$

  - Summarize the table

$$\hat{q}(a_1) = \frac{10 + 7 + 8}{3}$$

time	State	Action	reward
0	***1***	$\boldsymbol{a_1}$	**10**
1	*1*	a_2	5
2	***1***	$\boldsymbol{a_1}$	**7**
3	*1*	a_3	3
4	*1*	a_2	6
5	*1*	a_2	7
6	*1*	a_3	6
7	***1***	$\boldsymbol{a_1}$	**8**
…			

Collection of action and rewards

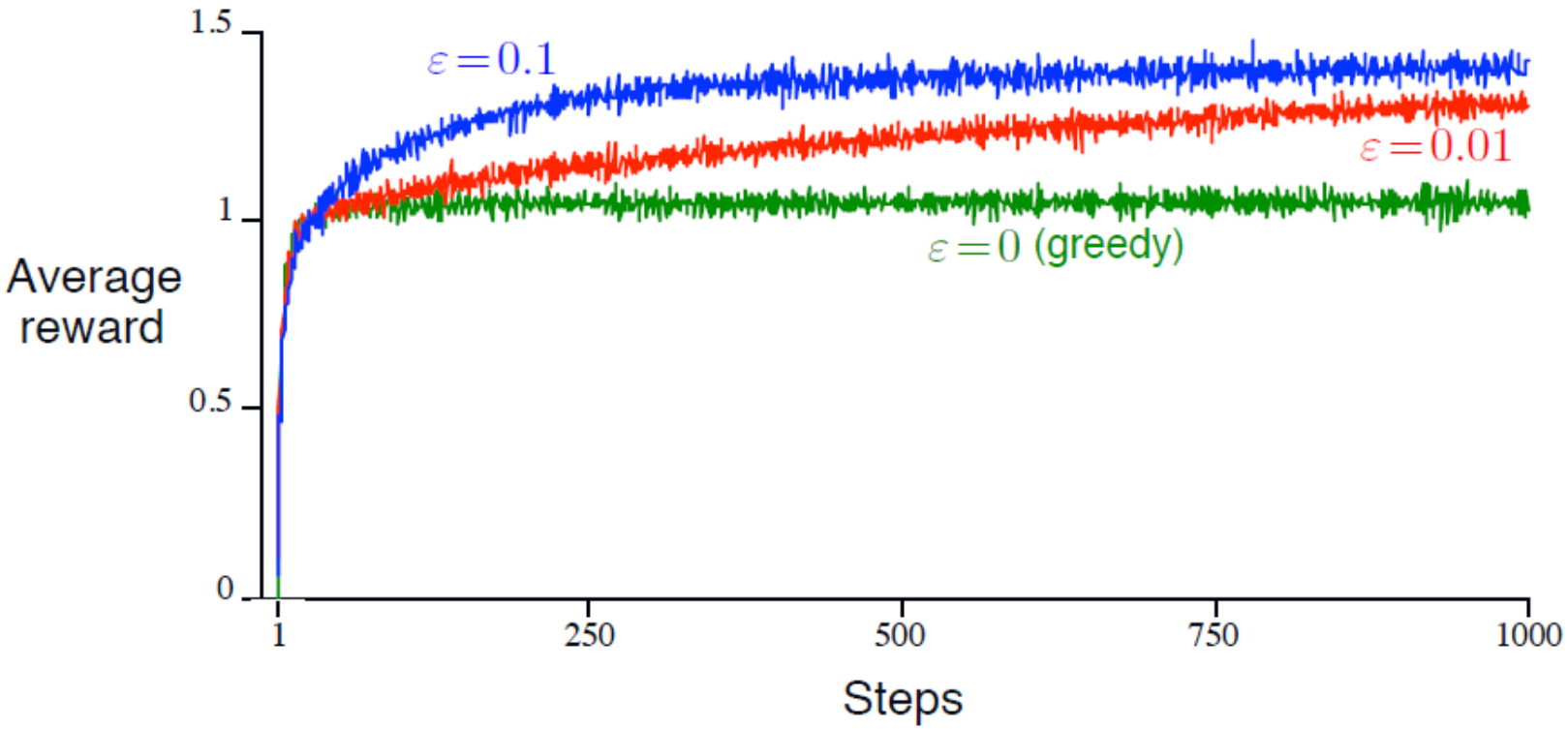
Simple Epsilon Greedy Exploration

- Choose best known action most of the time. $(1-\epsilon)$
- Do random things at rest of the time. (ϵ)

$$p(a_i) = \begin{cases} 1-\epsilon & \text{if } a_i = \arg\max_a \hat{q}(a) \\ \frac{\epsilon}{N} & \text{otherwise} \end{cases}$$

Results for ϵ greedy strategy

- Greedy is good in the short term.
- ϵ greedy Exploration is better for long term.



More sophisticated ways to explore?

- **Epsilon greedy** : randomly choose non-optimal choices.
- **Soft Max distribution** : choose based on expected value
  - $p(a_i) \propto \exp \hat{q}(a_i)$
- **Uncertainty based exploration** : explore actions that are valuable and with larger uncertainty in $\hat{q}(a)$

Learning Value V and Q

We showed how to act when we know $q(a)$, but how to learn these values?

Estimate Value by Collecting Data

- For bandit problem, we just can average the reward for this action

$$\hat{q}(a_i) = \frac{\sum_{t \text{ where } a_t = a_i} r_t}{\sum_{t \text{ where } a_t = a_i} 1}$$

  - Summarize the table

$$\hat{q}(a_1) = \frac{10 + 7 + 8}{3}$$

time	State	Action	reward
0	***1***	$\boldsymbol{a_1}$	**10**
1	*1*	a_2	5
2	***1***	$\boldsymbol{a_1}$	**7**
3	*1*	a_3	3
4	*1*	a_2	6
5	*1*	a_2	7
6	*1*	a_3	6
7	***1***	$\boldsymbol{a_1}$	**8**
…			

Collection of action and rewards

What can we do for general RL problems with states?

Can we do better and utilize the state transition?

Properties of Ideal Q,V

- V and Q values are related .
  - $Q_*(s_t, a_t) = V_*(s_{t+1}) + r(s_t, a_t),\ s_{t+1} = T(s_t, a)$
  - $V_*(s_t) = \max_a Q_*(s_t, a)$
- Interpretation
  - Value of a state is as high as the best action it can make.
  - Value of an action is as high as the value of the next state + the reward

γ temporal decay can be added

- E.g. One-million dollar today is better than one million dollar in 10 yrs.
- γ could be interpreted as the possibility of death …

Properties of Q,V

Simply combining the two equations

$$Q_*(s_t, a_t) = V_*(s_{t+1}) + r(s_t, a_t),\ s_{t+1} = T(s_t, a)$$

$$V_*(s_t) = \max_a Q_*(s_t, a)$$

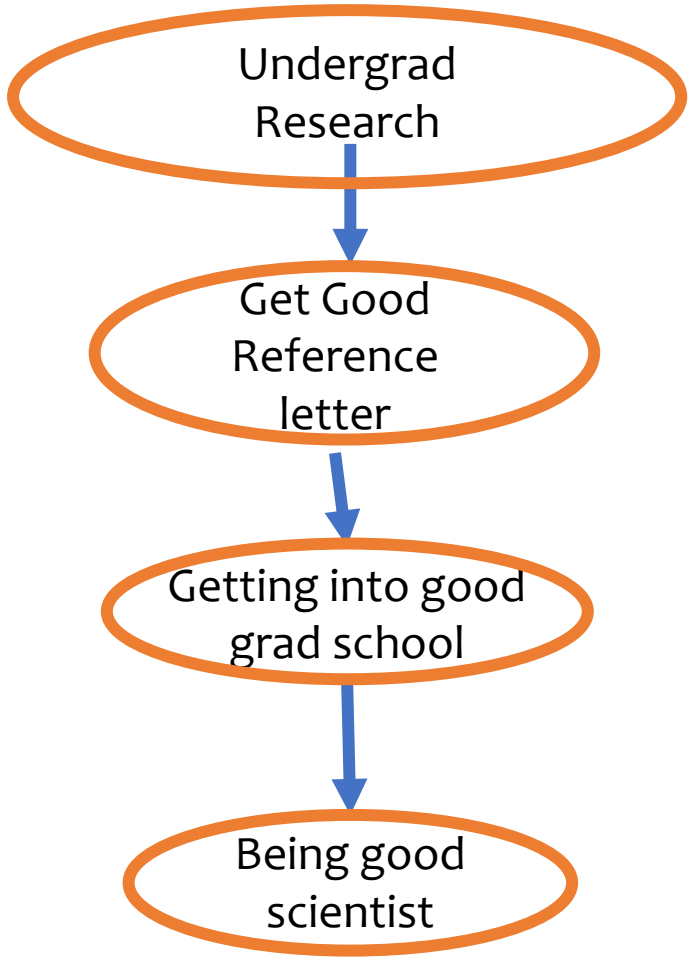
- Bellman equation of V

$$V_*(s_t) = \max_a [V_*(s_{t+1}) + r(s_t, a)]\,, \qquad s_{t+1} = T(s_t, a)$$

- Intuition
  - Say state s_j has high value, if s_i can transfer to s_j, then s_i also has high value.
  - s_i shall have value as high as the states it can transfer to.
  - E.g.
    - s_1 = Sending kids to good school. $V(s_1)$ is high
    - s_2 = Owning a house around good schools.
    - If s_2 is connected to s_1 by a low-cost path, then $V(s_2)$ is also high

A *cynical* academia example…

- If we know such state transition diagram …
- If being a good scientist is of high value to you …
- The high value flows backward…



Q-Learning

Still simply combining the two equations

$$Q_*(s_t, a_t) = V_*(s_{t+1}) + r(s_t, a_t),\ \ s_{t+1} = T(s_t, a)$$

$$V_*(s_t) = \max_a Q_*(s_t, a)$$

- Bellman equation of Q

$$Q_*(s_t, a_t) = r(s_t, a_t) + \max_a Q_*(s_{t+1}, a)\,, \qquad s_{t+1} = T(s_t, a)$$

  - This equation is satisfied by the ideal Q function.
  - *Off policy*: It can learn from experience of any policy.

- Q learning objective is to minimize the error

$$\Delta = r(s_t, a_t) + \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$$

- Update equation

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Delta$$

  - $\alpha < 1$ learning rate

Why we learn Q function not V function?

SARSA

- Bellman equation of Q

$$Q_\pi(s_t, a_t) = r(s_t, a_t) + Q_\pi(s_{t+1}, \hat{a}),$$
$$\hat{a} = \pi(s_{t+1}), s_{t+1} = T(s_t, \hat{a})$$

  - Choose not the best action at s_{t+1} but the action according to certain policy.
  - *On policy*: learn from experience of its own policy.

- Temporal difference

$$\Delta = r(s_t, a_t) + Q(s_{t+1}, \hat{a}) - Q(s_t, a_t)$$

- Update equation

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Delta$$

Comparing to Q learning:

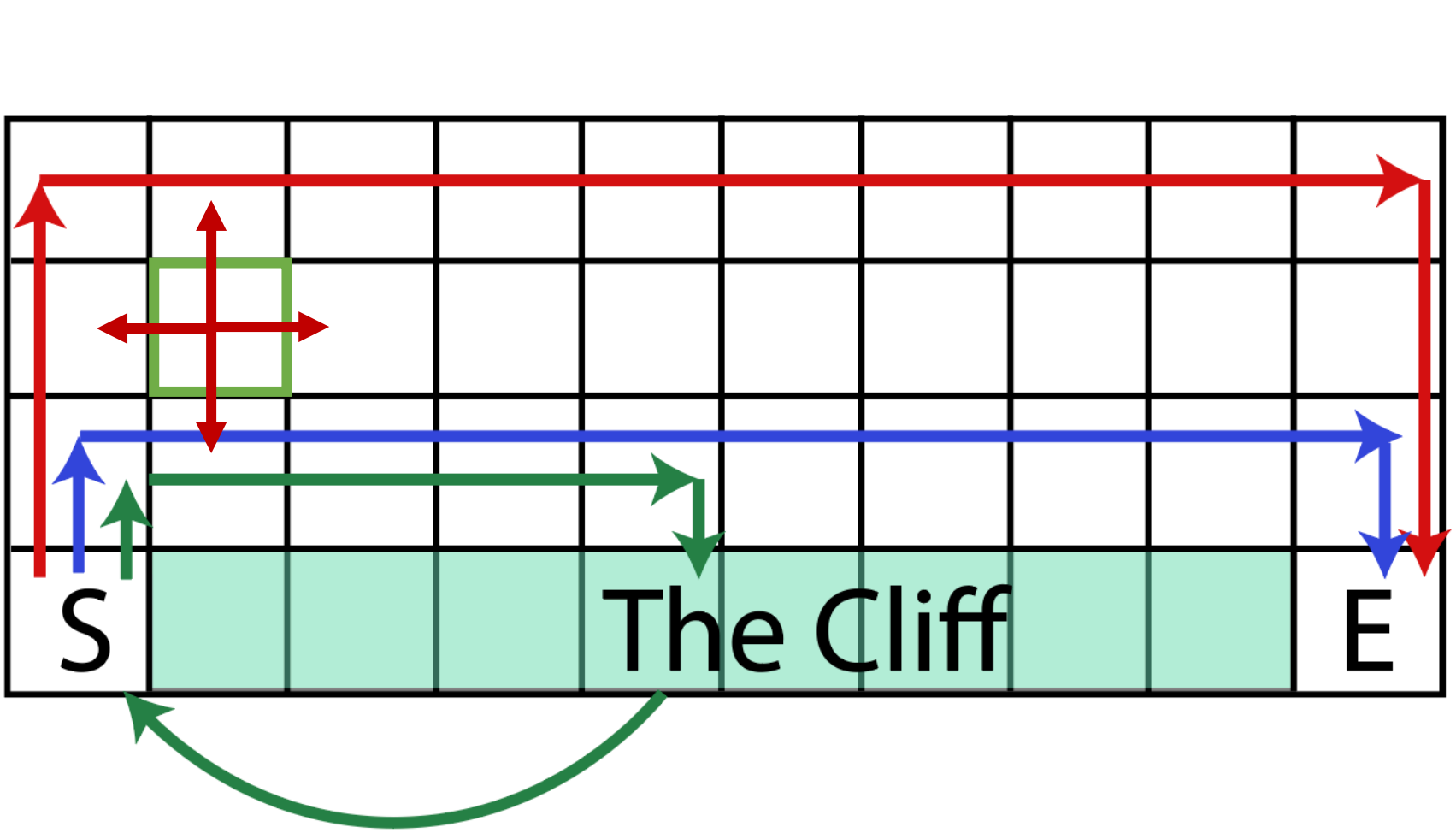
$$Q_*(s_t, a_t) = r(s_t, a_t) + \max_a Q_*(s_{t+1}, a),$$
$$s_{t+1} = T(s_t, a)$$

Homework 5 Environment

CliffWorld

Life is a punishment… Sometimes harder than other times.

You need to escape this hell as soon as possible.



States s_i

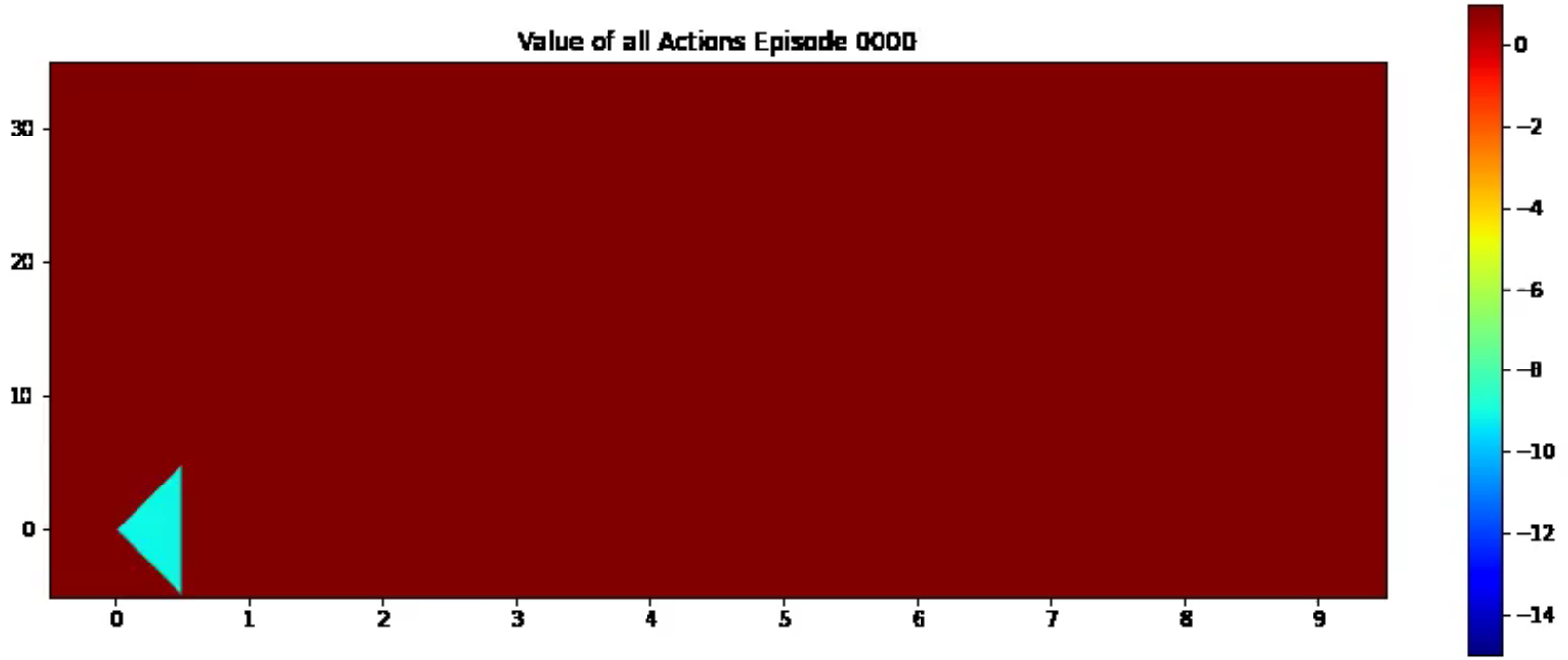
Actions $\{a_i\}$

Reward:

-1 normally

-100 falling off cliff

Dynamics of Q value



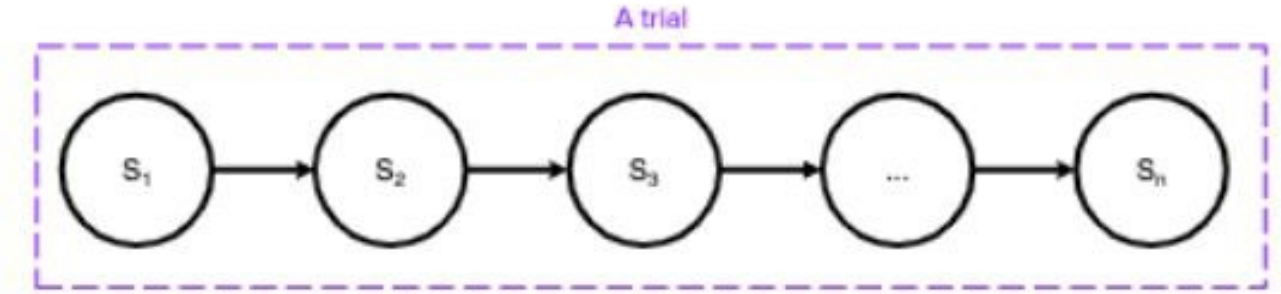
Q values for the states and the 4 actions through out learning

- Negative values flow back from the target
- Falling action is learned.
- Optimal action is picked, so only highest values flow into neighboring cells.

Conditioning and Reward Prediction Error

Classic Conditioning

- Action set is simple: a
  - Progress in time

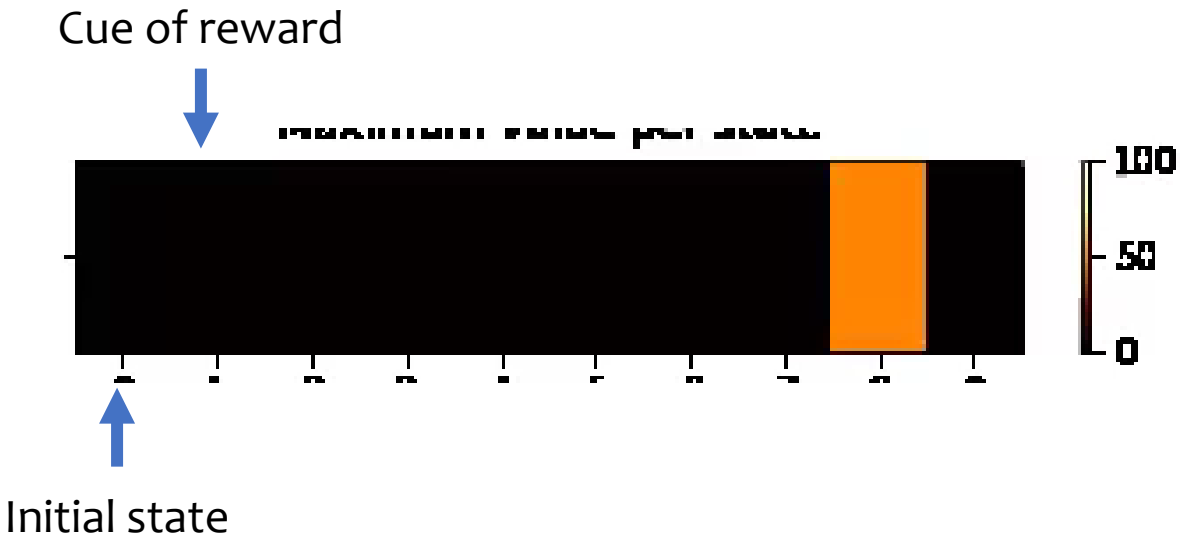


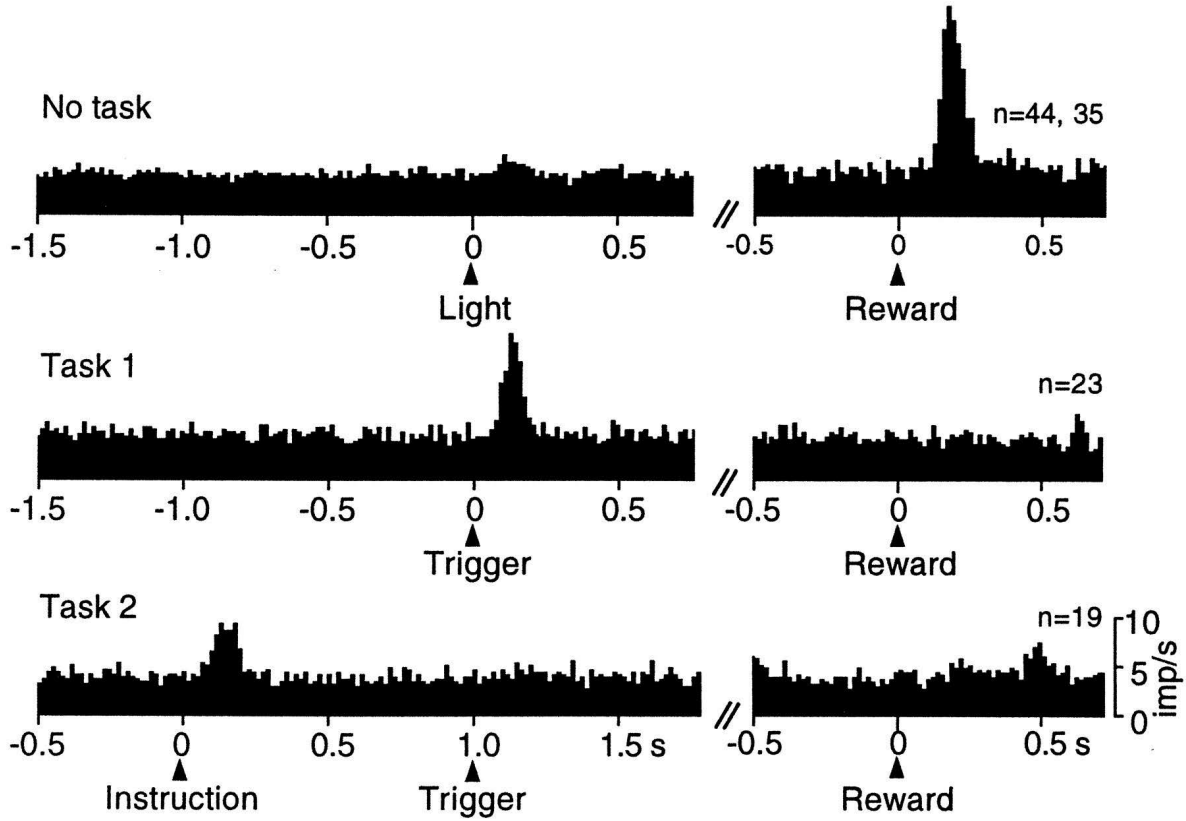
- State set s_i
  - Chain of states, not depending on a

- Value $V(s)$
  - expected reward following the state s_i

TD Learning and Pavlovian Conditioning.

- Through learning, dopamine activity move from the reward itself to the ***earliest reward predicting state***.
- Same for dopamine neuron firing



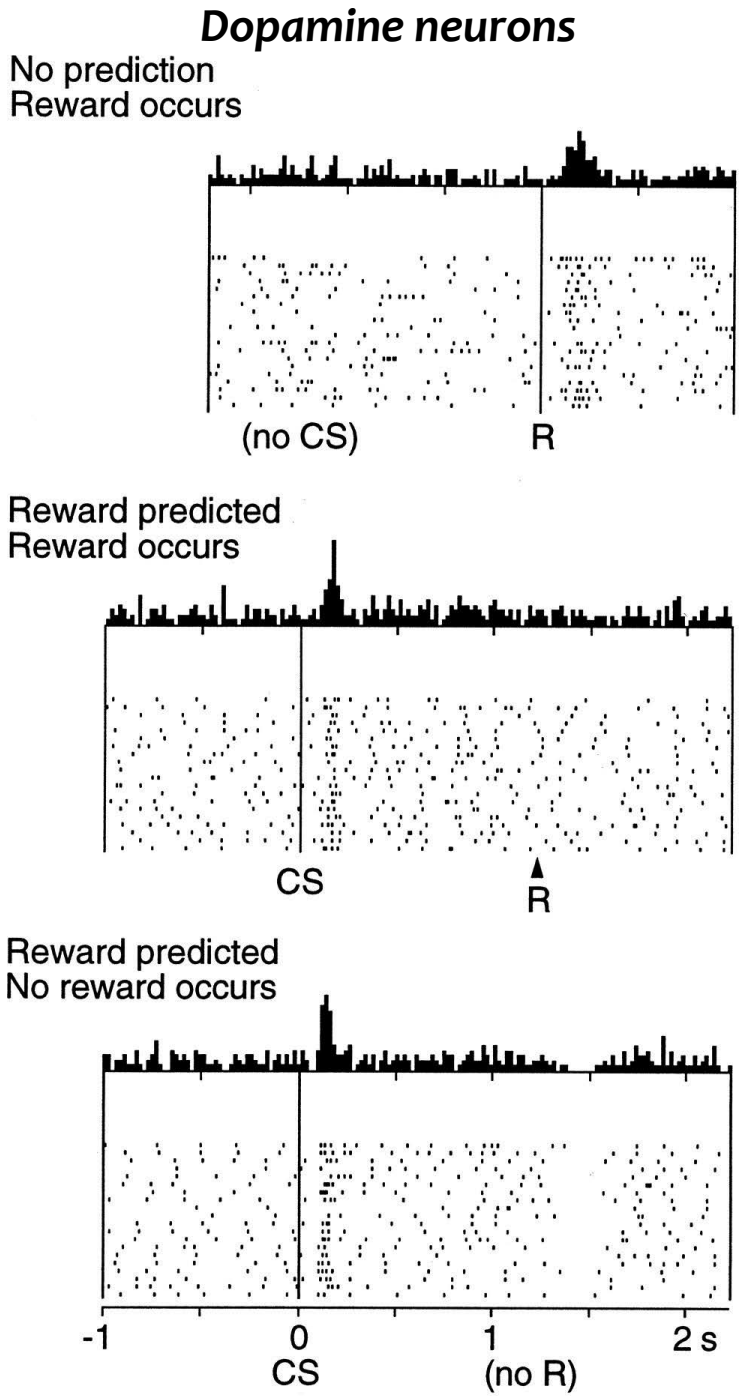


Dopamine neuron and Reward Prediction Error

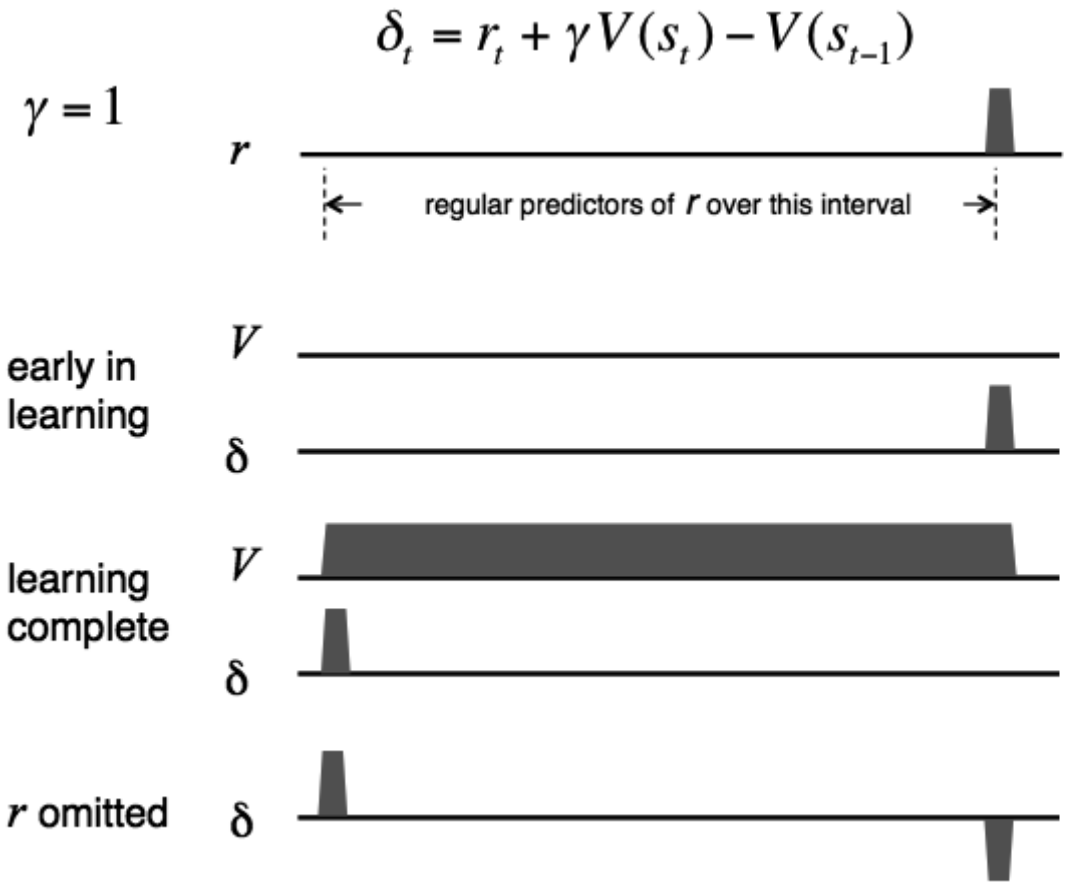
- r reward in current time step
- $V(s_t)$: expected future reward from time t
- $V(s_{t+1})$: expected future reward from time t+1

- No prediction
  - $\Delta = r + V(s_{t+1}) - V(s_t) = r$

- Predicted
  - At reward,
    $\Delta = r + V(s_{t+1}) - V(s_t) = r + 0 - r = 0$
  - At cue,
    $\Delta = r + V(s_{t+1}) - V(s_t) = 0 + r - 0 = r$
  - At missing reward,
    $\Delta = r + V(s_{t+1}) - V(s_t) = 0 + 0 - r = -r$



RPE through out learning



Food for Thought

- What is the earliest reward predicting state?
  - Unpredicted predictive state…
- Why don’t dopamine neurons fire from the start of the trial?
  - Cue is more strongly associative with reward, than start of trial.
- Should RPE fully disappear after learning?
  - Yep, overtrain will decrease RPE even at cue.