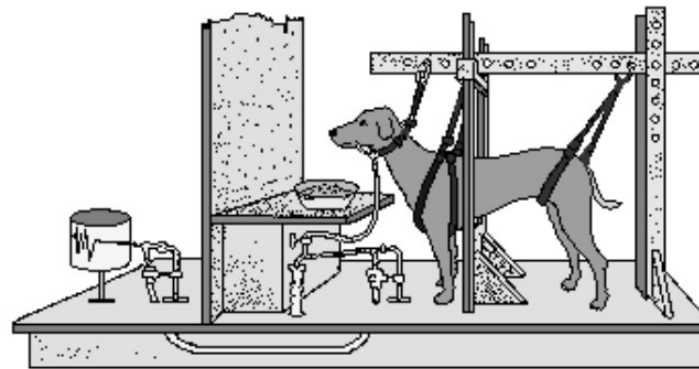
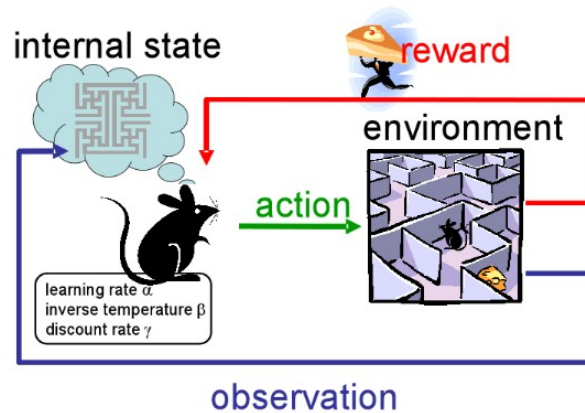
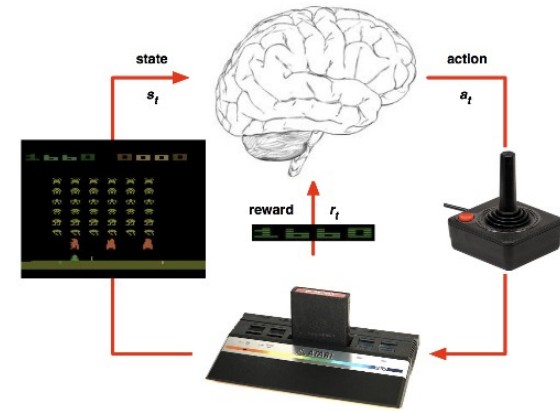
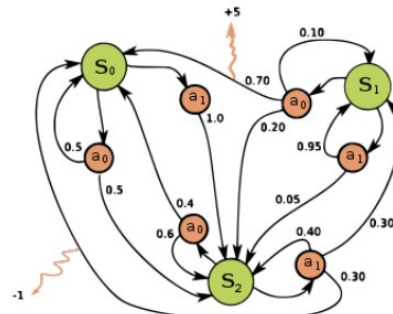
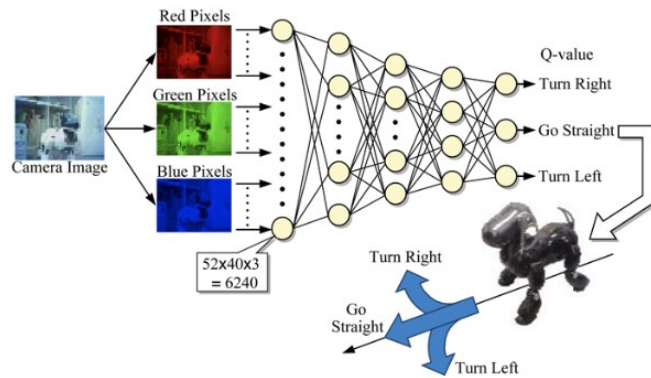


# COMP 138: Reinforcement Learning



Instructor: Jivko Sinapov

# Announcements

# Reading Assignment

- Chapter 7 of Sutton and Barto

# Research Article Topics

- Transfer learning
- Learning with human demonstrations and/or advice
- Approximating q-functions with neural networks

# Research Paper

- Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., & Thomaz, A. L. (2013). **Policy shaping: Integrating human feedback with reinforcement learning**. In *Advances in neural information processing systems* (pp. 2625-2633).
- Responses should discuss both readings
- You get extra credit for answering others' questions!

# Monte Carlo Methods

# Overview of Monte Carlo ES

Monte Carlo ES (Exploring Starts), for estimating  $\pi \approx \pi_*$

Initialize, for all  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}(s)$ :

$Q(s, a) \leftarrow$  arbitrary

$\pi(s) \leftarrow$  arbitrary

$Returns(s, a) \leftarrow$  empty list

Repeat forever:

Choose  $S_0 \in \mathcal{S}$  and  $A_0 \in \mathcal{A}(S_0)$  s.t. all pairs have probability  $> 0$

Generate an episode starting from  $S_0, A_0$ , following  $\pi$

For each pair  $s, a$  appearing in the episode:

$G \leftarrow$  the return that follows the first occurrence of  $s, a$

Append  $G$  to  $Returns(s, a)$

$Q(s, a) \leftarrow$  average( $Returns(s, a)$ )

For each  $s$  in the episode:

$\pi(s) \leftarrow \operatorname{argmax}_a Q(s, a)$

# On- vs. Off-policy Methods

- On-policy methods attempt to improve a policy that is used for gathering data
- Off-policy methods attempt to improve a different policy from the one used for gathering data



# Off-policy exploration in humans

<https://www.youtube.com/watch?v=8vNxjw2AqY>

first-visit MC control (for  $\varepsilon$ -soft policies), estimates  $\pi \approx \pi_*$

Initialize, for all  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}(s)$ :

$Q(s, a) \leftarrow$  arbitrary

$Returns(s, a) \leftarrow$  empty list

$\pi(a|s) \leftarrow$  an arbitrary  $\varepsilon$ -soft policy

Repeat forever:

(a) Generate an episode using  $\pi$

(b) For each pair  $s, a$  appearing in the episode:

$G \leftarrow$  the return that follows the first occurrence of  $s, a$

Append  $G$  to  $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

(c) For each  $s$  in the episode:

$A^* \leftarrow \arg \max_a Q(s, a)$

(with ties broken arbitrarily)

For all  $a \in \mathcal{A}(s)$ :

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$$

## first-visit MC control (

Initialize, for all  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}(s)$ :

$Q(s, a) \leftarrow$  arbitrary

$Returns(s, a) \leftarrow$  empty list

$\pi(a|s) \leftarrow$  an arbitrary  $\varepsilon$ -soft policy

Repeat forever:

(a) Generate an episode using  $\pi$

(b) For each pair  $s, a$  appearing in the episode:

$G \leftarrow$  the return that follows the first occurrence of  $s, a$

Append  $G$  to  $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

(c) For each  $s$  in the episode:

$A^* \leftarrow \arg \max_a Q(s, a)$

(with ties broken arbitrarily)

For all  $a \in \mathcal{A}(s)$ :

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$$

Is this on- or off-policy learning?

Does this algorithm learn the optimal policy?

Does it estimate the true Q function?

# Programming Assignment #2

# On-policy MC

## On-policy first-visit MC control (

Initialize, for all  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}(s)$ :

$Q(s, a) \leftarrow$  arbitrary

$Returns(s, a) \leftarrow$  empty list

$\pi(a|s) \leftarrow$  an arbitrary  $\varepsilon$ -soft policy

Repeat forever:

(a) Generate an episode using  $\pi$

(b) For each pair  $s, a$  appearing in the episode:

$G \leftarrow$  the return that follows the first occurrence of  $s, a$

Append  $G$  to  $Returns(s, a)$

$Q(s, a) \leftarrow \text{average}(Returns(s, a))$

(c) For each  $s$  in the episode:

$A^* \leftarrow \arg \max_a Q(s, a)$

(with ties broken arbitrarily)

For all  $a \in \mathcal{A}(s)$ :

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$$

How can we implement this algorithm efficiently?

# Off-policy learning and importance sampling

- The prediction problem: given data generated using policy  $b$ , what is the value function for policy  $\pi$ ?
- Off-policy prediction and control

# Temporal Difference Learning

- Overview of Section 6.1

# Learning in a Grid World

<https://www.youtube.com/watch?v=tovrpoUkzYU>



# Temporal Difference Learning

- Example 6.4

# Q-Learning and Sarsa

## Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Initialize  $Q(s, a)$ , for all  $s \in \mathcal{S}, a \in \mathcal{A}(s)$ , arbitrarily, and  $Q(\text{terminal-state}, \cdot) = 0$

Repeat (for each episode):

Initialize  $S$

Repeat (for each step of episode):

Choose  $A$  from  $\mathcal{A}(S)$  using policy derived from  $Q$  (e.g.,  $\epsilon$ -greedy)

Take action  $A$ , observe  $R, S'$

$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma \max_a Q(S', a) - Q(S, A)]$

$S \leftarrow S'$

until  $S$  is terminal

## Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Initialize  $Q(s, a)$ , for all  $s \in \mathcal{S}, a \in \mathcal{A}(s)$ , arbitrarily, and  $Q(\text{terminal-state}, \cdot) = 0$

Repeat (for each episode):

Initialize  $S$

Choose  $A$  from  $\mathcal{A}(S)$  using policy derived from  $Q$  (e.g.,  $\epsilon$ -greedy)

Repeat (for each step of episode):

Take action  $A$ , observe  $R, S'$

Choose  $A'$  from  $\mathcal{A}(S')$  using policy derived from  $Q$  (e.g.,  $\epsilon$ -greedy)

$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma Q(S', A') - Q(S, A)]$

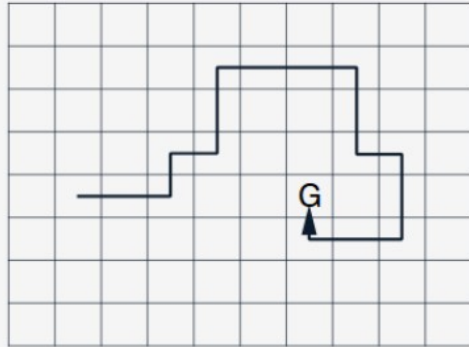
$S \leftarrow S'; A \leftarrow A'$

until  $S$  is terminal

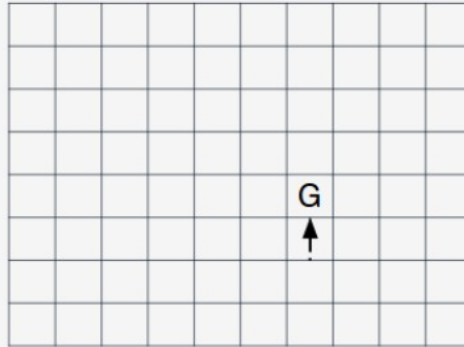
# Cliff-walking example

# Beyond 1-step updates

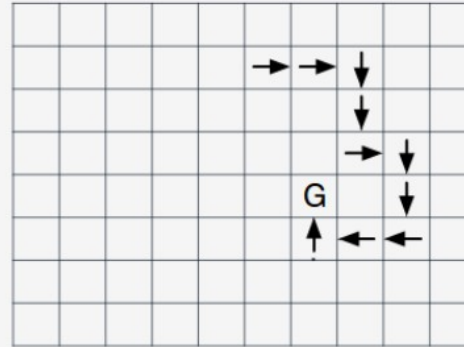
Path taken



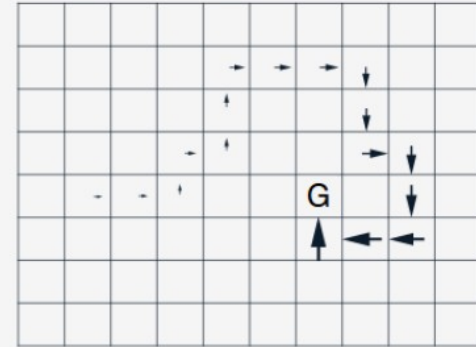
Action values increased by one-step Sarsa



Action values increased by 10-step Sarsa



Action values increased by Sarsa( $\lambda$ ) with  $\lambda=0.9$



# Moderated Discussion

THE END

