

CSC 665: Artificial Intelligence

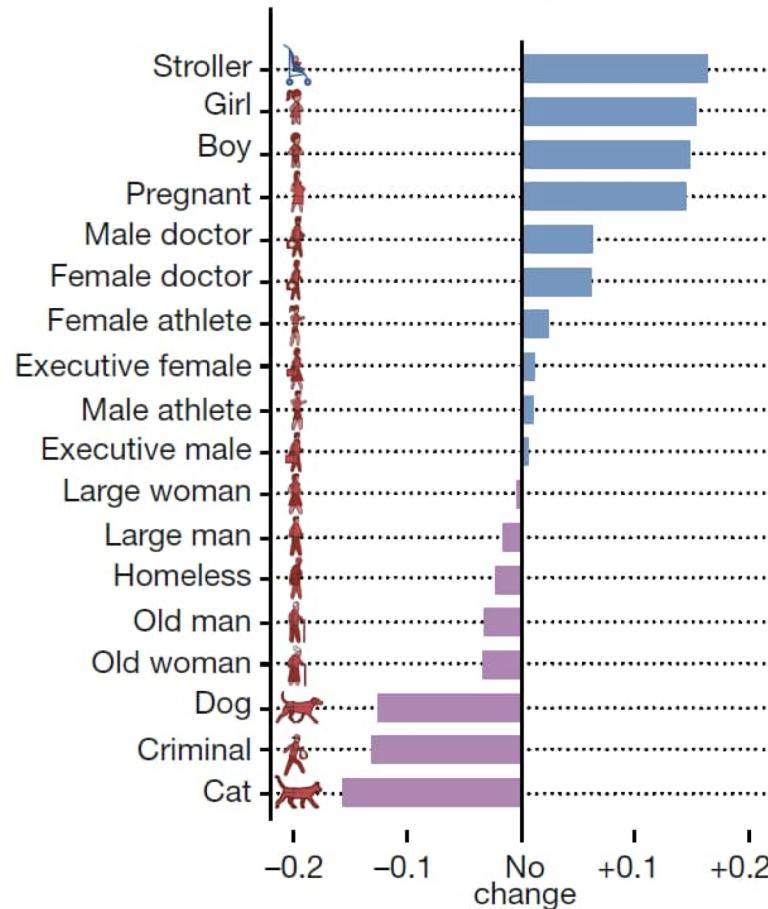
Reinforcement Learning

Instructor: Pooyan Fazli
San Francisco State University

AI in the News

b

Preference in favour of sparing characters



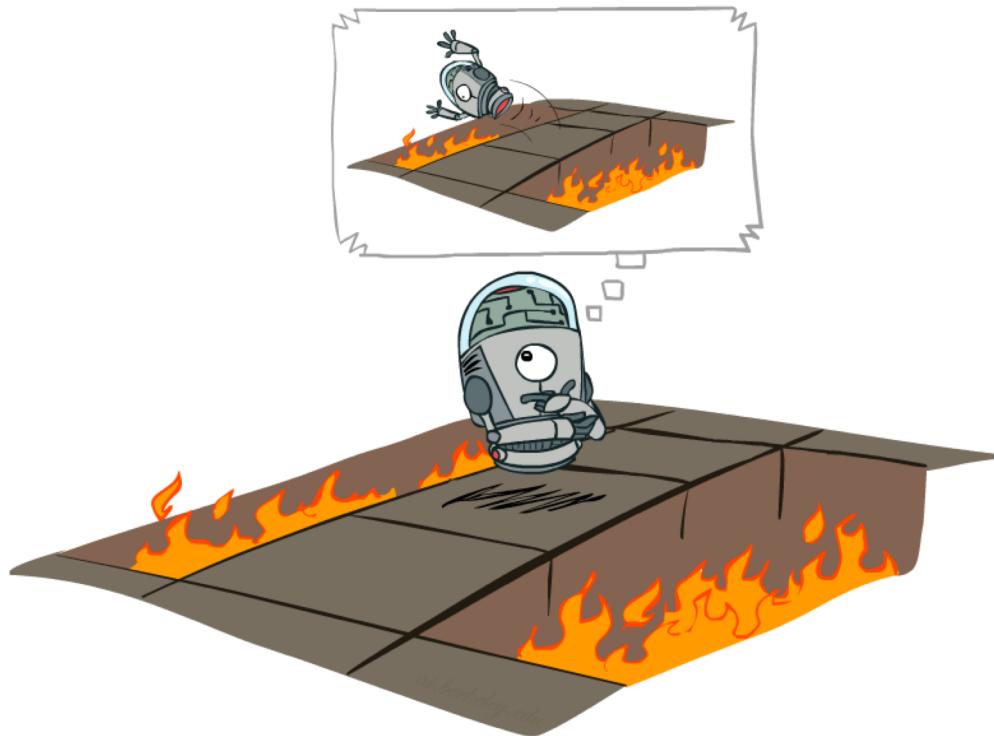
New study of millions of people's preferences re who self-driving cars should kill and spare when all lives can't be saved: <https://go.nature.com/2O4eoEk>

Reinforcement Learning

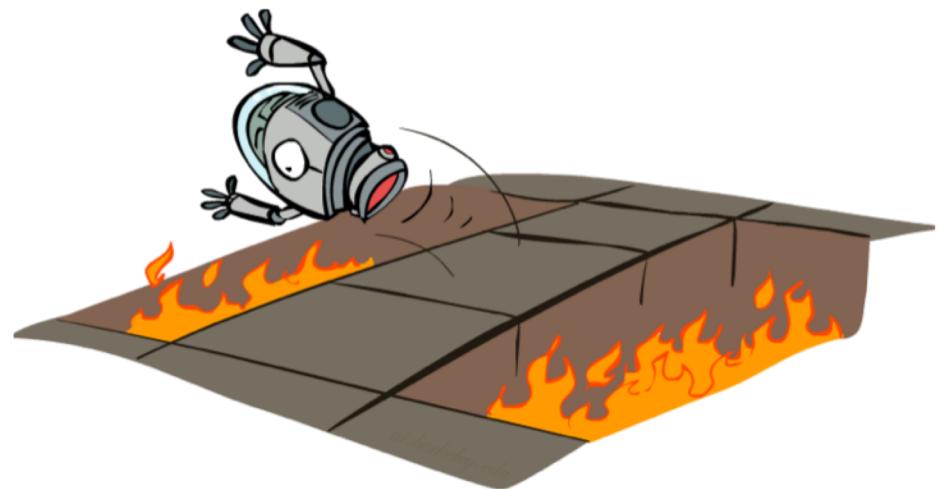
- Still assume a Markov decision process (MDP):
 - A set of states $s \in S$
 - A set of actions (per state) A
 - A transition function $T(s,a,s')$
 - A reward function $R(s,a,s')$
- Still looking for an optimal policy $\pi^*(s)$
- New twist: don't know T or R (model of MDP)
 - I.e. we don't know which states are good or what the actions do

Reinforcement Learning

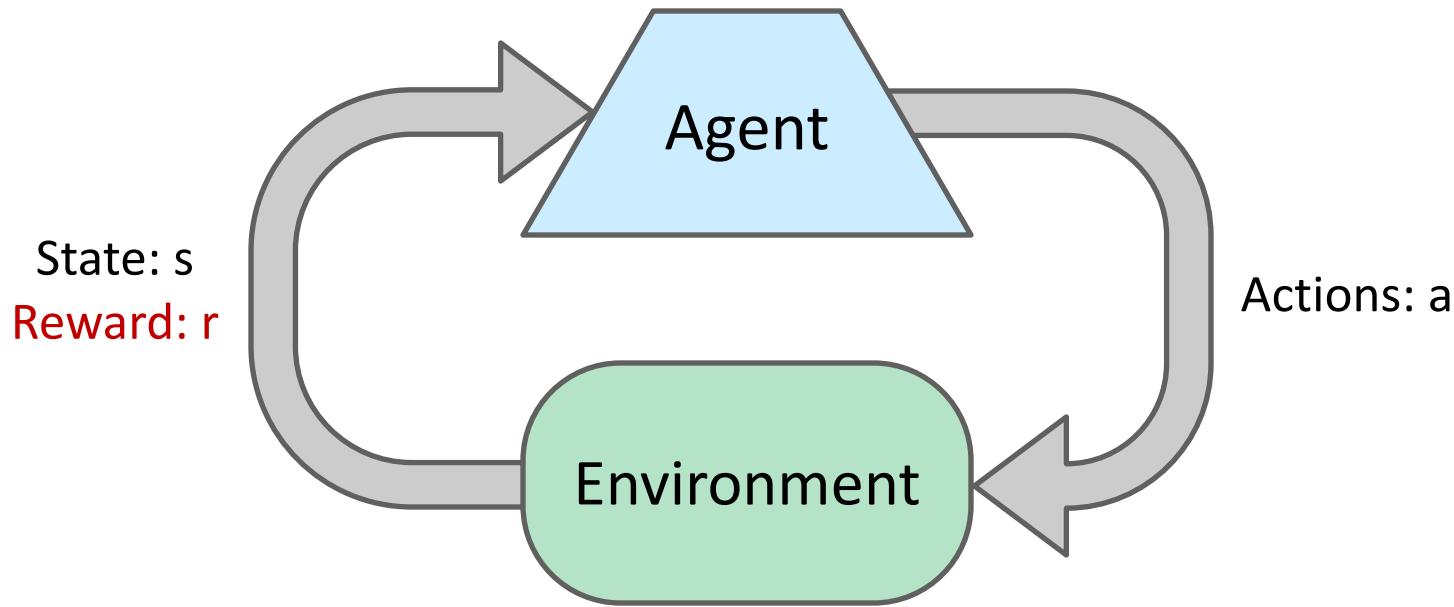
MDP computes an optimal policy (offline)



RL learns an optimal policy (online)



Reinforcement Learning



- Basic idea:
 - Receive feedback in the form of **rewards**
 - Must (learn to) act so as to **maximize expected rewards** based on observed samples of outcomes!

Example: Learning to Walk



Initial



After Learning [1K Trials]

Example: Learning to Walk



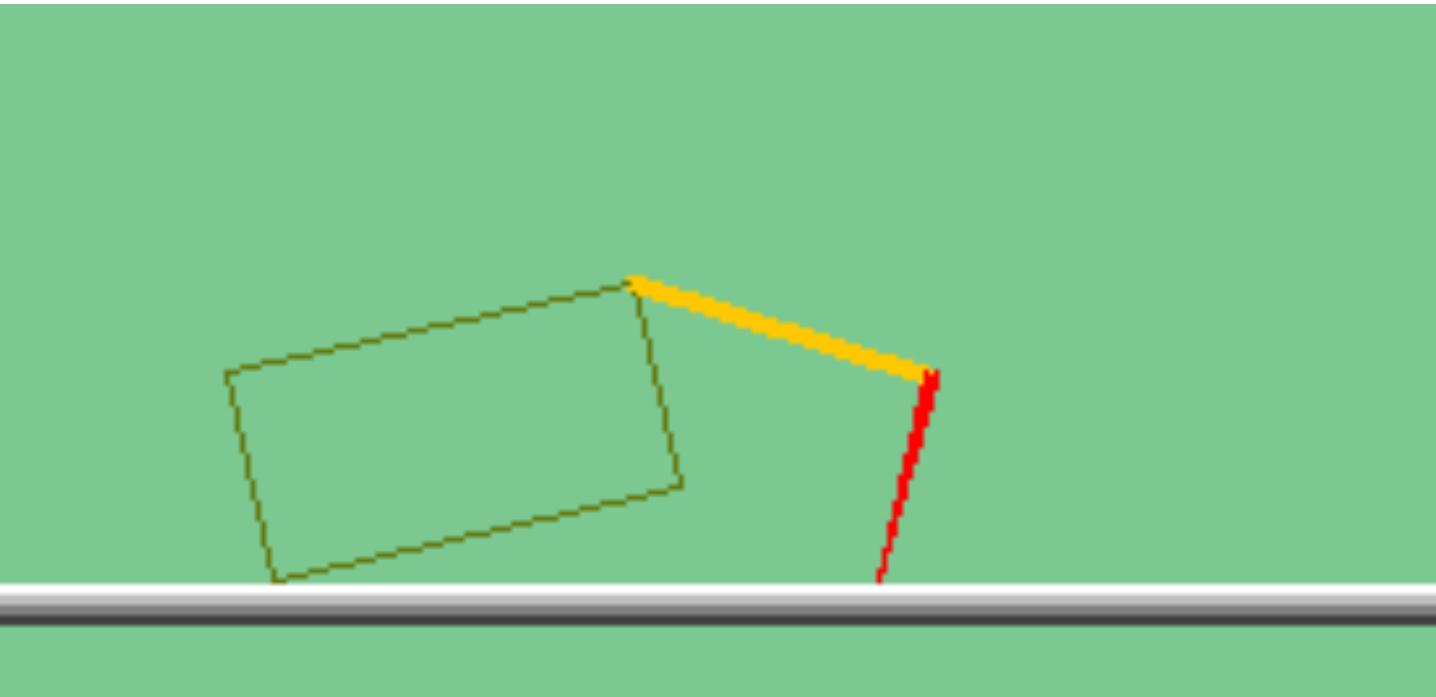
Initial

Example: Learning to Walk

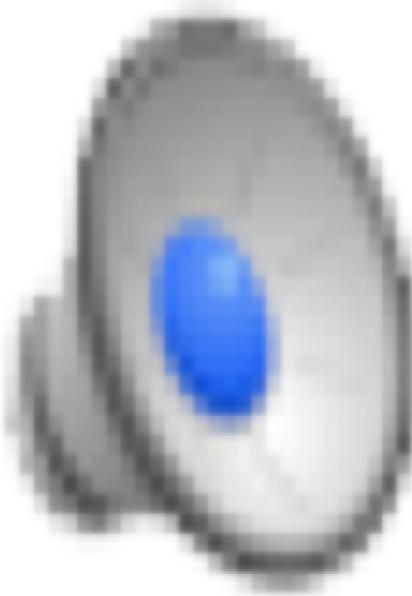


Finished

The Crawler!



Video of Demo Crawler Bot



Passive vs Active Reinforcement Learning

- **Passive RL Learning**
 - The agent acts based on a fixed policy π and tries to learn how good the policy is by observing the world go by

- **Active RL Learning**
 - The agent attempts to find an optimal policy by exploring different actions in the world

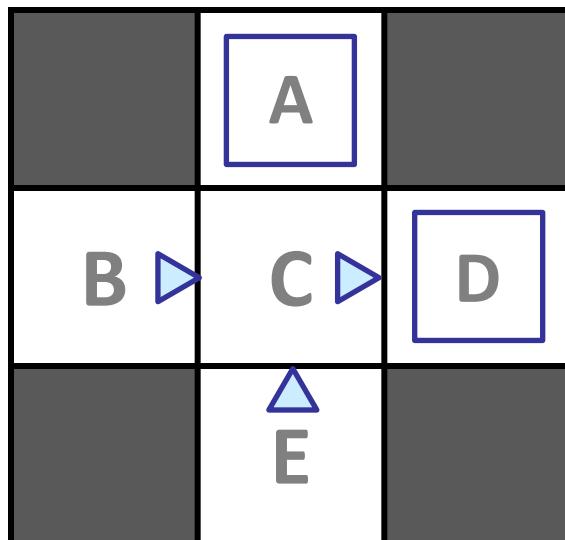
Model-Based vs Model-Free Learning

- Model-Based approach to RL:
 - Learn the MDP model (T and R), or an approximation of it
 - Use it to evaluate the fixed policy (Passive RL Learning)
 - Use it to find the optimal policy (Active RL Learning)

 - Model-Free approach to RL:
 - Without explicitly learning the model (T and R)
 - Evaluate the fixed policy (Passive RL Learning)
 - Find the optimal policy (Active RL Learning)
- Directly learning Values of each state
Don't learn T and R**

Example: Model-Based Learning

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Learned Model

$$\hat{T}(s, a, s')$$

$T(B, \text{east}, C) = 1.00$
 $T(C, \text{east}, D) = 0.75$
 $T(C, \text{east}, A) = 0.25$
...

$$\hat{R}(s, a, s')$$

$R(B, \text{east}, C) = -1$
 $R(C, \text{east}, D) = -1$
 $R(D, \text{exit}, x) = +10$
...

Model-Based Learning

- Step 1: Learn the MDP model (T and R)
 - Count outcomes s' for each s, a
 - Normalize to give an estimate of $\hat{T}(s, a, s')$
 - Discover each $\hat{R}(s, a, s')$ when we experience (s, a, s')
- Step 2: Solve the learned MDP
 - Use Policy Evaluation to evaluate the fixed policy (Passive RL Learning)
 - Use Value Iteration to find the optimal policy (Active RL Learning)

Model-Free Learning

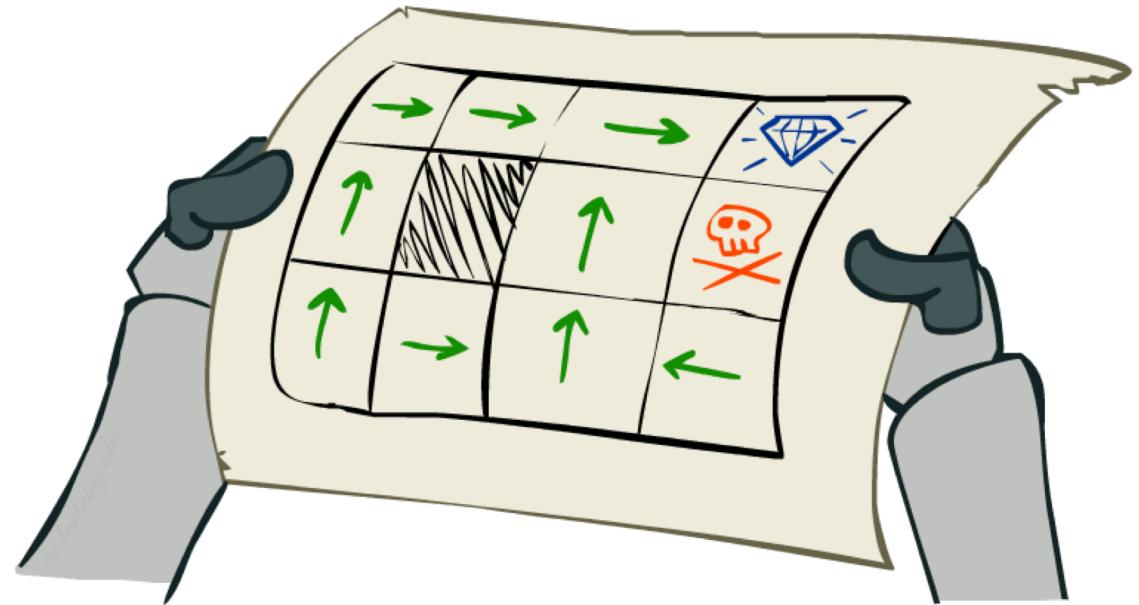
Passive Reinforcement Learning

- **Task: Policy Evaluation**

- Input: a fixed policy $\pi(s)$
- You don't know the transitions $T(s,a,s')$
- You don't know the rewards $R(s,a,s')$
- **Goal: learn the state values**

- In this case:

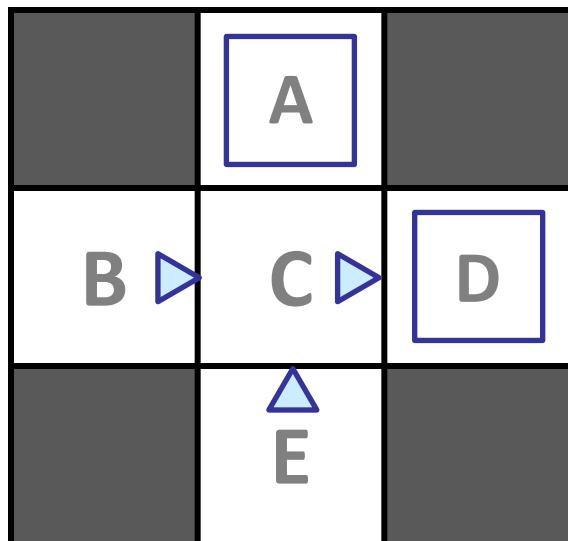
- No choice about what actions to take
- Just execute the policy and learn from experience
- This is NOT offline planning! You actually take actions in the world.



Direct Evaluation

Example: Direct Evaluation

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Output Values

	-10	
A	+4	+10
B	+8	
C		D
	-2	
E		

Direct Evaluation

- Goal: Compute values for each state under π
- Idea:
 - Act according to π
 - Use the **actual** sum of discounted rewards from s
 - Average over multiple trials and visits to s
- This is called Direct Evaluation

Problems with Direct Evaluation

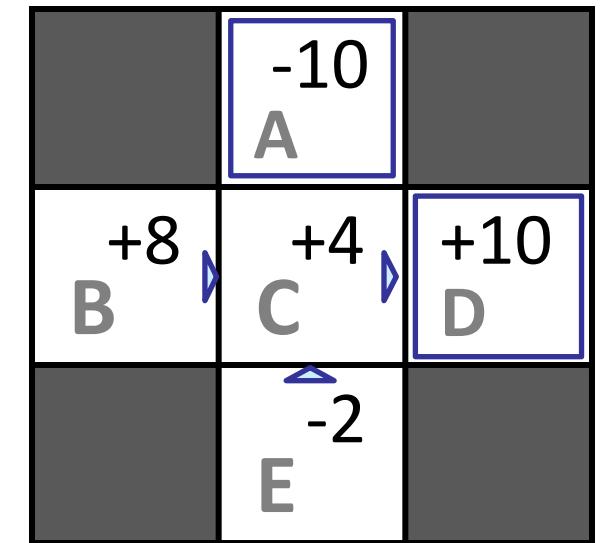
- What's good about **Direct Evaluation**?

- It's easy to understand
- It doesn't require any knowledge of T, R
- It eventually computes the correct average values, using just sample transitions

- What bad about it?

- It **ignores information about state connections**
- So, it takes a long time to learn

Output Values



If B and E both go to C under this policy, how can their values be different?

Question?

Direct Evaluation is a Approach

Model-Free ✓

Model-Based

Temporal Difference Learning

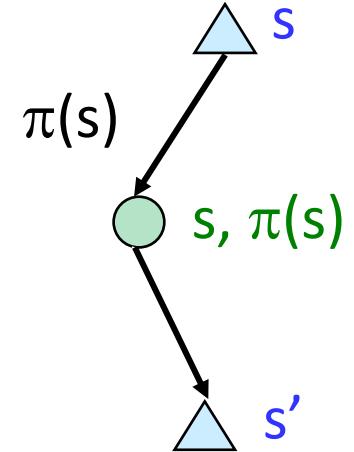
Temporal Difference Learning

- Idea: learn from every experience!
 - Update $V(s)$ each time we experience a transition (s, a, s', r)

- Temporal Difference Learning of values
 - Policy still fixed, still doing evaluation!

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

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



- **α is the learning rate:** determines to what extent the newly acquired information will override the old information.

Example: Temporal Difference Learning

States

	A	
B	C	D
	E	

Observed Transitions

B, east, C, -2

	0	
0	0	8
	0	

C, east, D, -2

	0	
-1	0	8
	0	

	0	
-1	3	8
	0	

Assume: $\gamma = 1, \alpha = 1/2$

$$V^\pi(s) \leftarrow (1 - \alpha)V^\pi(s) + \alpha [R(s, \pi(s), s') + \gamma V^\pi(s')]$$

Question?

TD Learning is a Approach.

Model-Free ✓

Model-Based

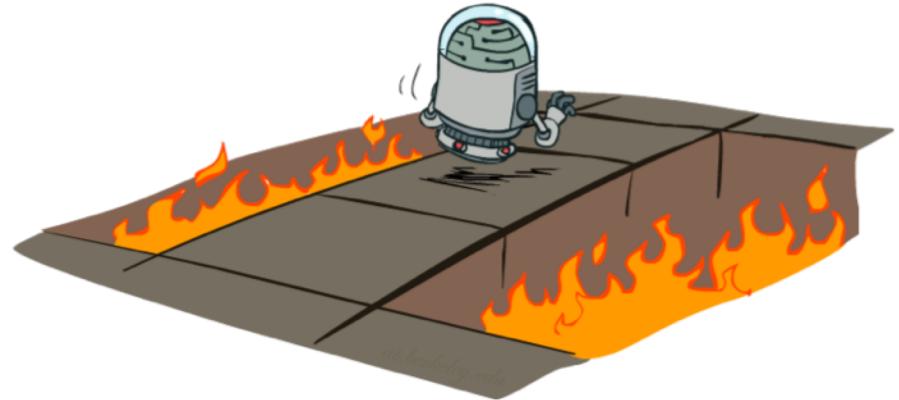
Active Reinforcement Learning

- Full reinforcement learning: optimal policies

- You don't know the transitions $T(s,a,s')$
- You don't know the rewards $R(s,a,s')$
- You choose the actions now
- Goal: learn the optimal policy / values

- In this case:

- Learner makes choices!
- This is NOT offline planning! You actually take actions in the world and find out what happens...



Q-Learning

Q-Learning

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

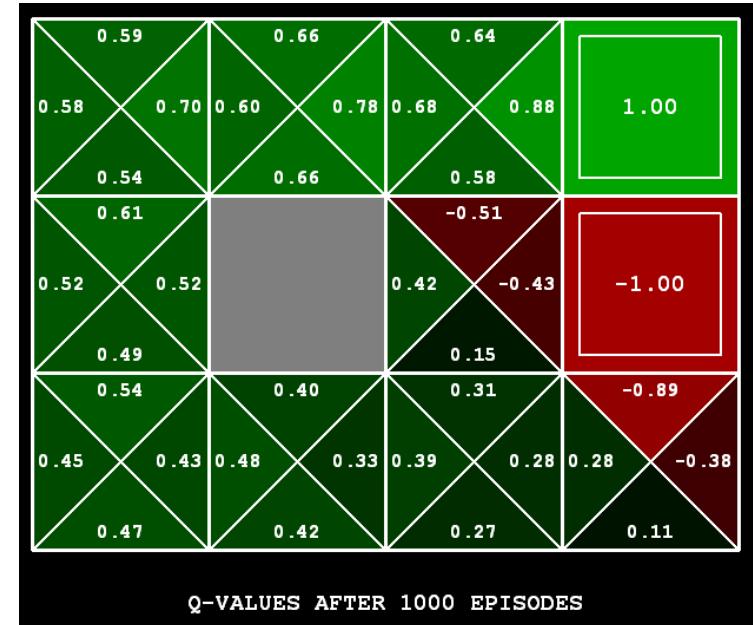
$$\text{sample} = R(s, a, s') + \gamma \max_{a'} Q(s', a')$$

- Incorporate the new estimate:

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

or

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + (\alpha) \left[r + \gamma \max_{a'} Q(s', a') \right]$$



Q-Learning – Manual Grid



Q-Learning – Auto Grid

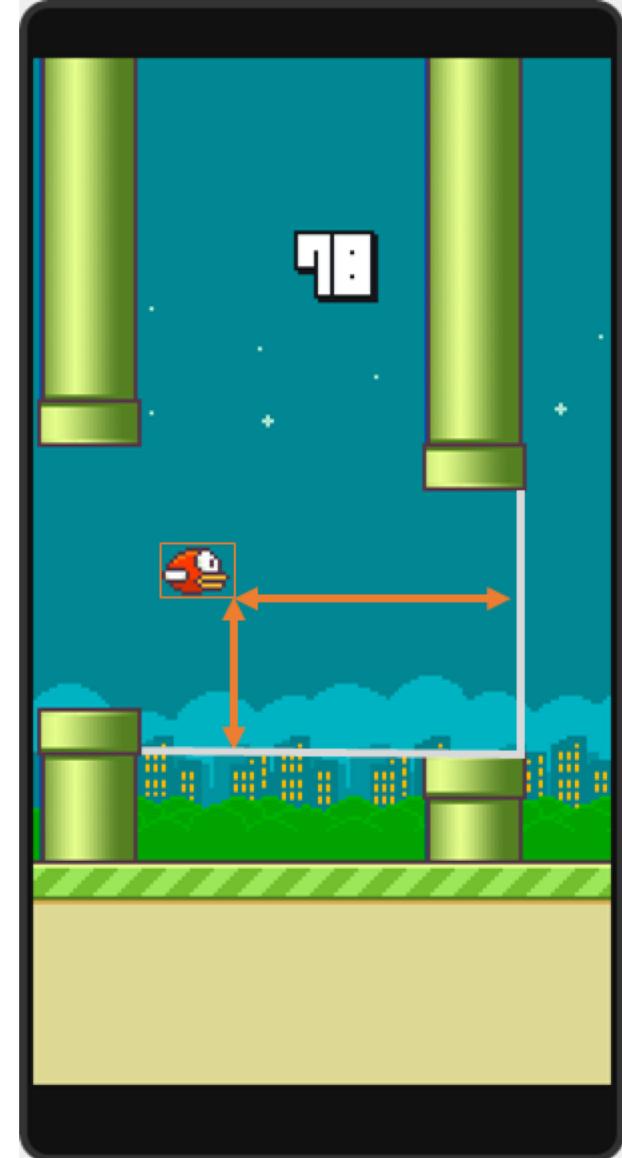


Q-Learning Properties

- Amazing result: Q-learning converges to optimal policy!
- This is called **off-policy learning**
- Caveats:
 - You have to explore enough
 - You have to eventually make the learning rate small enough
 - ... but not decrease it too quickly

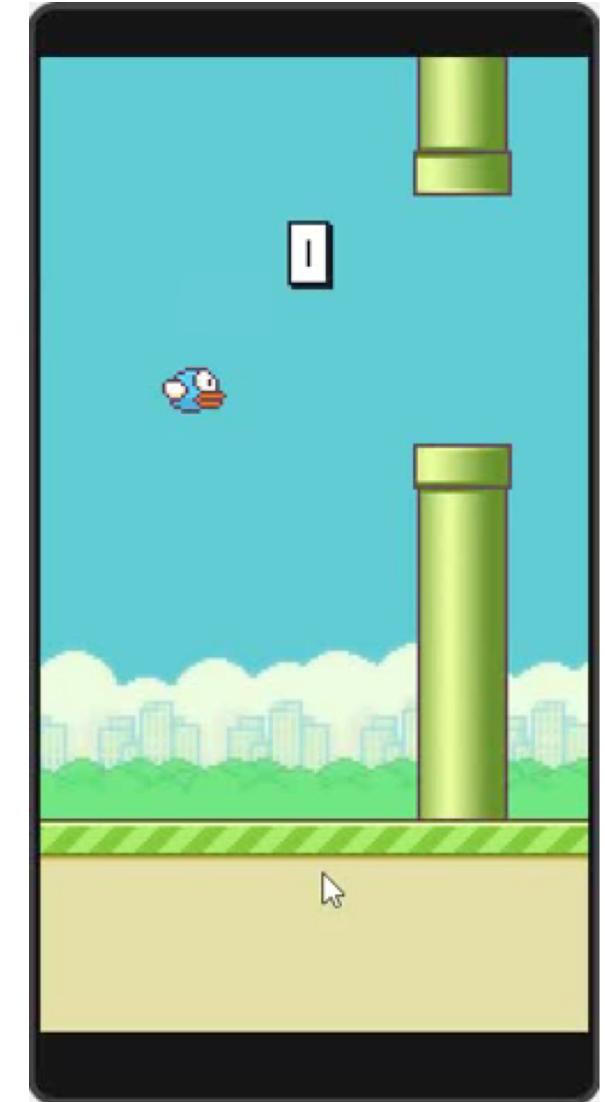
Flappy Bird RL

- State space
 - Discretized vertical distance from lower pipe
 - Discretized horizontal distance from next pair of pipes
 - Life: Dead or Living
- Actions
 - Click
 - Do nothing
- Rewards
 - +1 if Flappy Bird still alive
 - -1000 if Flappy Bird is dead
- 6-7 hours of Q-learning



Flappy Bird RL

- State space
 - Discretized vertical distance from lower pipe
 - Discretized horizontal distance from next pair of pipes
 - Life: Dead or Living
- Actions
 - Click
 - Do nothing
- Rewards
 - +1 if Flappy Bird still alive
 - -1000 if Flappy Bird is dead
- 6-7 hours of Q-learning



Summary: Passive vs Active Reinforcement Learning

- **Passive RL Learning**
 - Policy Evaluation on Approx. MDP (Model-Based)
 - Direct Evaluation (Model-Free)
 - Temporal Difference (TD) Learning (Model-Free)
- **Active RL Learning**
 - Value Iteration on Approx. MDP (Model-Based)
 - Q-learning (Model-Free)

The Story So Far: MDPs and RL

Known MDP: Offline Solution

Goal

Compute V^* , Q^* , π^*

Evaluate a fixed policy π

Technique



Unknown MDP: Model-Based

Goal

Compute V^* , Q^* , π^*

Evaluate a fixed policy π

Technique



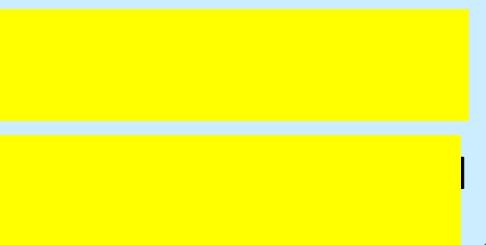
Unknown MDP: Model-Free

Goal

Compute V^* , Q^* , π^*

Evaluate a fixed policy π

Technique



The Story So Far: MDPs and RL

Known MDP: Offline Solution

Goal

Compute V^* , Q^* , π^*

Evaluate a fixed policy π

Technique

Value Iteration

Policy Evaluation

Unknown MDP: Model-Based

Goal

Compute V^* , Q^* , π^*

Evaluate a fixed policy π

Technique

VI on approx. MDP

PE on approx. MDP

Unknown MDP: Model-Free

Goal

Compute V^* , Q^* , π^*

Evaluate a fixed policy π

Technique

Q-learning

TD Learning/Direct Eval

Reading

- Read Sections 21.1, 21.2, and 21.3 in the AIMA textbook