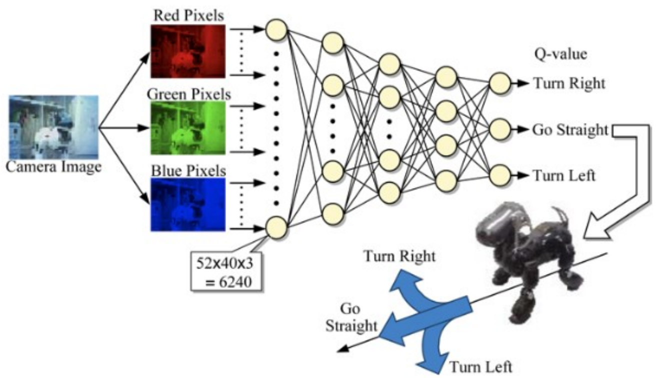
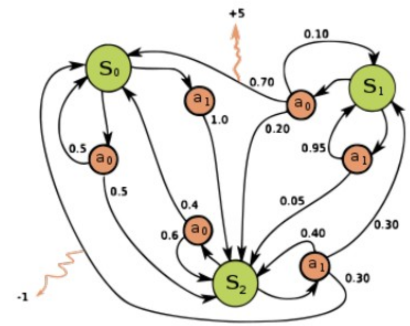
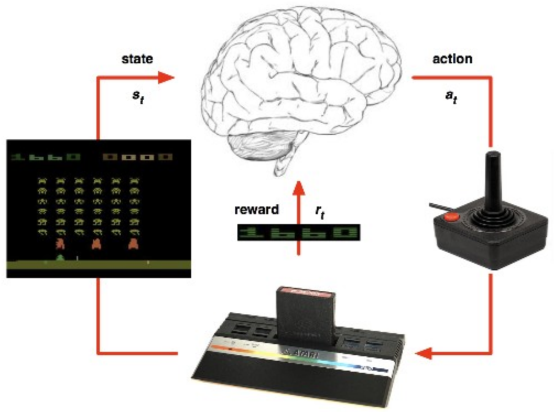
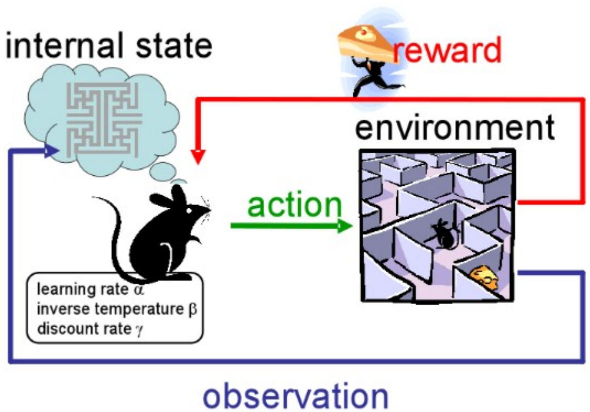
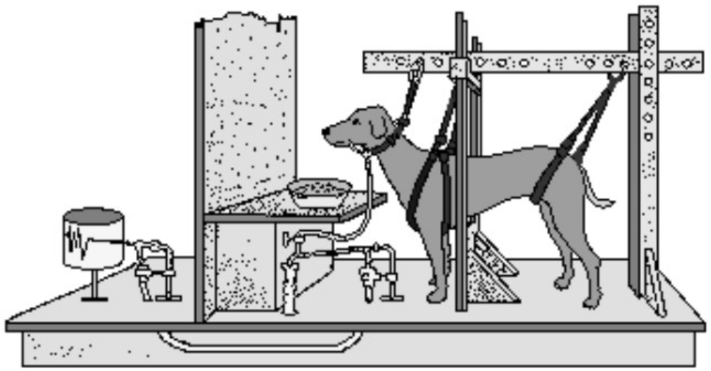
COMP 138: Reinforcement Learning











Instructor: Jivko Sinapov

Announcements

Reading Assignment

- Chapters 4 and 5 of Sutton and Barto

Homework 1

- Introduction
- Part 1: Programming Ex. as in the book: You can use subsections
- Part 2: Additional Question and Experiment
- Summary and Conclusion
- Extra Credit
- Submit: 1 PDF file of your report; 1 ZIP file containing all code + README.txt

Research Article Topics

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

Discussion:
What makes a good project?

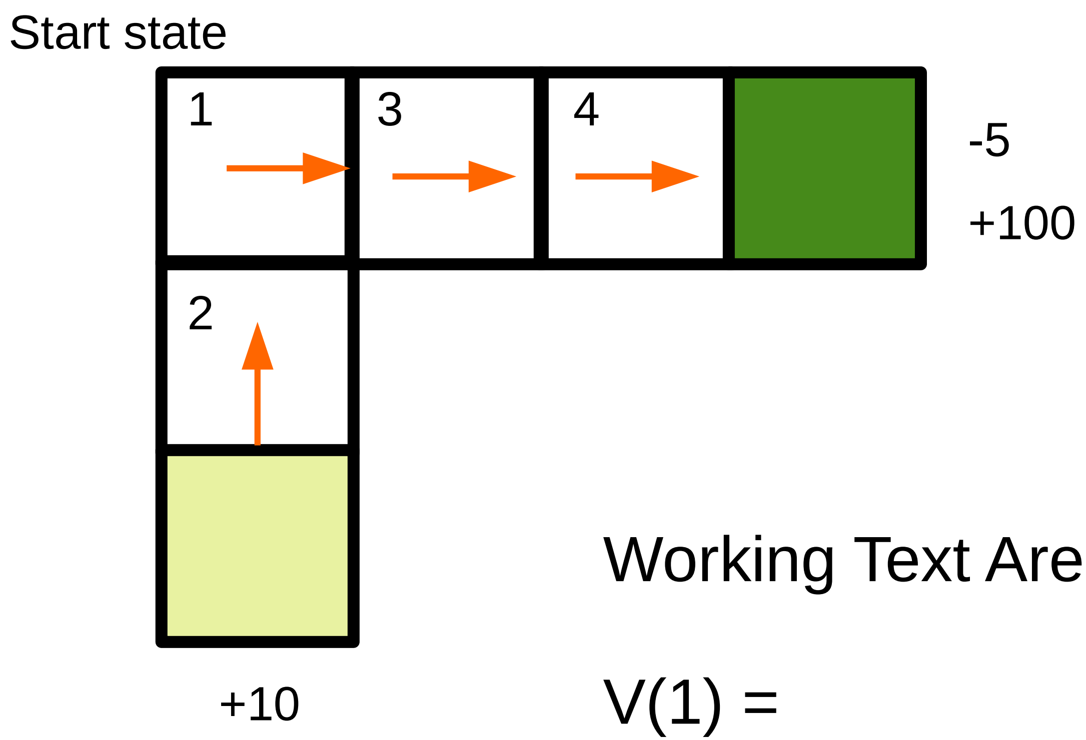
Example Project Video

https://www.youtube.com/watch?v=VMp6pq6_QjI

Policies and Value Functions

(exercise on board with L-shaped world)

Policies and Value Functions



Gamma = 0.05

Working Text Area:

V(1) =
V(2) =
V(3) =
V(4) =

Dynamic Programming

Dynamic Programming

“Dynamic Programming refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner.

While some decision problems cannot be taken apart this way, decisions that span several points in time do often break apart recursively.

Likewise, in computer science, if a problem can be solved optimally by breaking it into sub-problems and then recursively finding the optimal solutions to the sub-problems, then it is said to have optimal substructure.”

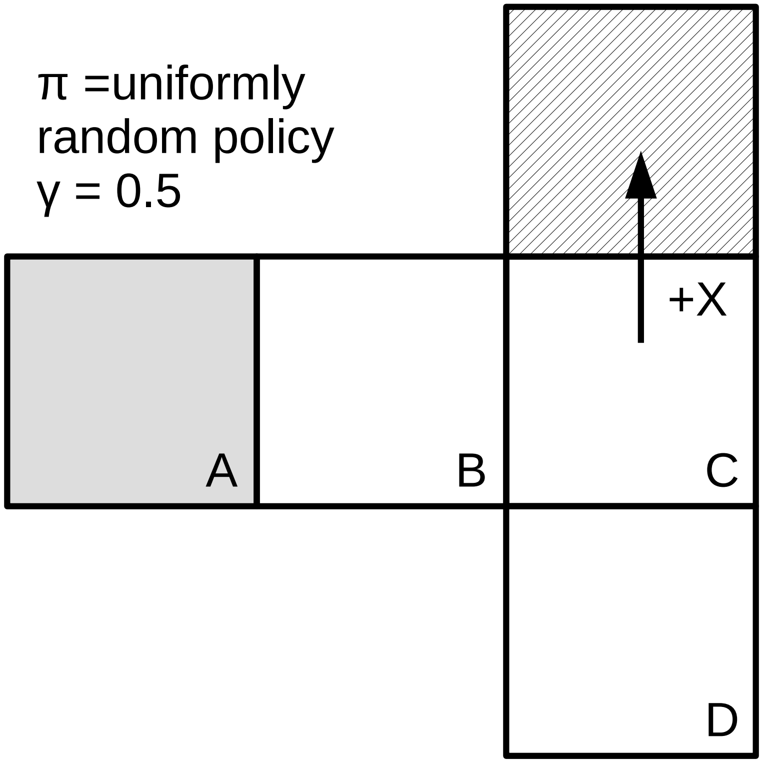
- wikipedia

Policy Evaluation

Iterative policy evaluation

Input π, the policy to be evaluated
Initialize an array $V(s) = 0$, for all $s \in \mathcal{S}^+$
Repeat
  $\Delta \leftarrow 0$
  For each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$ (a small positive number)
Output $V \approx v_\pi$

π =uniformly
random policy
$\gamma = 0.5$

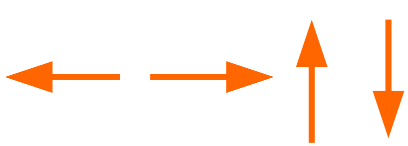


	V_0	V_1	V_2	V_3	V_4	V_5
A	0	0	x/24			
B	0	0	x/12			
C	0	x/3				
D	0	x/6				

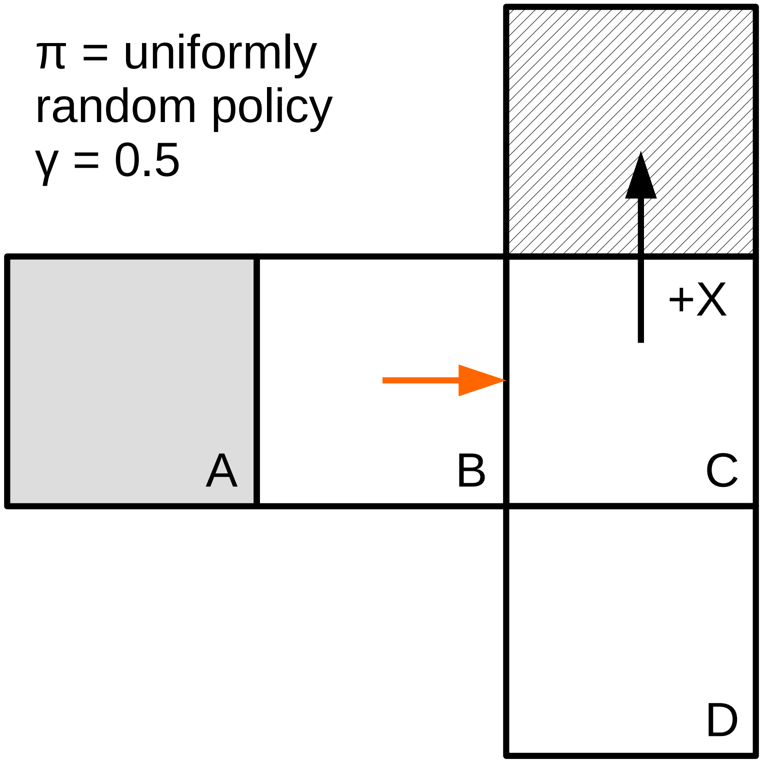
At each state, the agent has 1 or more actions allowing it to move to neighboring states. Moving in the direction of a wall is not allowed

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_k(s')\Big] \end{aligned}$$

WORKING TEXT AREA:
½ * (0) + ½ * (1/2* x/3)



π = uniformly random policy
$\gamma = 0.5$

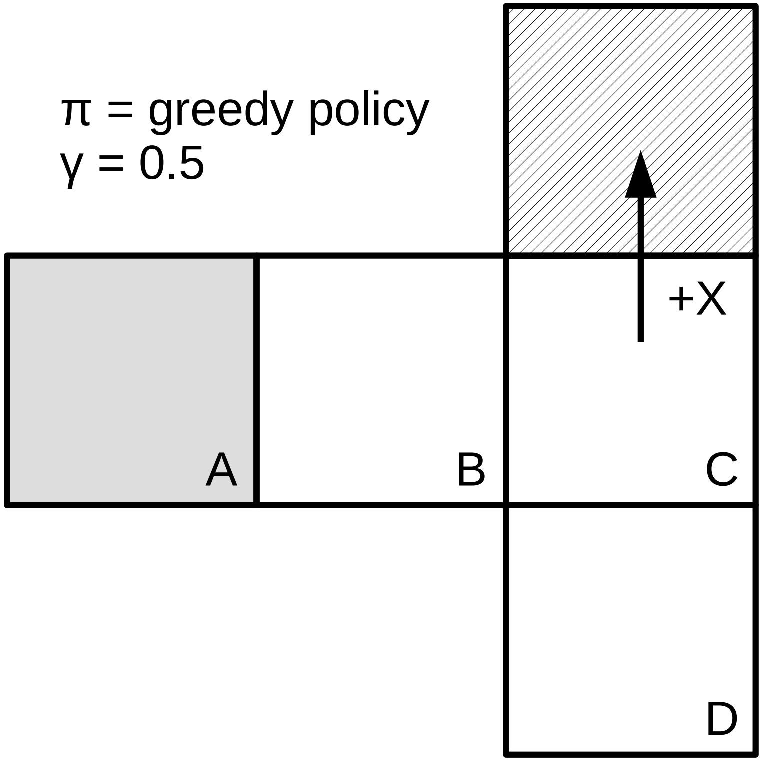


	V_0	V_1	V_2	V_3	V_4	V_5
A	0	0	0	X/24		
B	0	0	X/12			
C	0	X/3	3X/8			
D	0	X/6	3X/16			

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_k(s')\Big] \end{aligned}$$

Working area: 3x/16

π = greedy policy
γ = 0.5



	V_0	V_1	V_2	V_3	V_4	V_5
A	0					
B	0					
C	0					
D	0					

$$
\begin{aligned}
v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_k(s')\Big]
\end{aligned}
$$

Policy Improvement

- Main idea: if for a particular state *s*, we can do better than following the current policy by taking a different action, then the current policy is not optimal and changing it to follow the different action at state *s* improves it

Policy Iteration

- evaluate → improve → evaluate → improve → …..

Value Iteration

- Main idea:
  - Do one sweep of policy evaluation under the current greedy policy
  - Repeat until values stop changing (relative to some small Δ)

Policy iteration (using iterative policy evaluation)

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat

   $\quad \Delta \leftarrow 0$

   $\quad$ For each $s \in \mathcal{S}$:

   $\quad\quad v \leftarrow V(s)$

   $\quad\quad V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s))\big[r + \gamma V(s')\big]$

   $\quad\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$

   until $\Delta < \theta$ (a small positive number)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*

   For each $s \in \mathcal{S}$:

   $\quad$ *old-action* $\leftarrow \pi(s)$

   $\quad \pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

   $\quad$ If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*

   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Monte Carlo MDP Demo

https://colab.research.google.com/drive/1A9Ce6vJApuIZnHmrv98W6wTb8WFl5eWi?usp=sharing

THE END