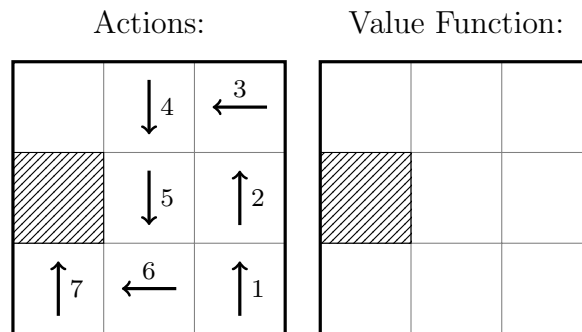


Reinforcement Learning Exercise

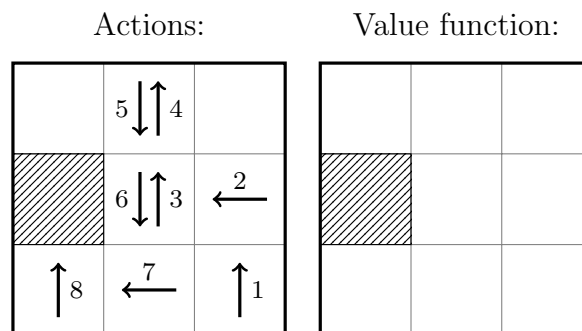
Question 1: Monte Carlo Estimation

Given is a gridworld with possible actions in each direction and a reward of -1 for each action. The marked grid is the terminal state. The actions for two episodes have been calculated and are displayed below. The numbers indicate the order of the steps.

- (a) The first episode uses the following actions. Calculate the value function after the episode.



- (b) (8 points) Here are the actions for the second episode. What is the updated value function?



Solution:

Actions(1. Episode):

Value Function:

-	-4	-5
	-3	-6
-1	-2	-7

Value from Ep 2:

-	-5	-
	-6	-7
-1	-2	-8

Averaged value function:

-	-4.5	-5
	-4.5	-6.5
-1	-2	-7.5

Question 2: Time Difference Learning

We would now like to use TD(0) prediction for the same problem. The actions are the same as above. The value function is initially -1 for every state except the terminal state. A constant step size $\alpha = 0.2$ is used. Run the algorithm for 2 episodes.

Initiale Value Funktion:

-1.0	-1.0	-1.0
	-1.0	-1.0
-1.0	-1.0	-1.0

- (a) The following actions are used for the first episode. Specify the value function after the first episode.

Actions:

	↓ ₄	← ₃
	↓ ₅	↑ ₂
↑ ₇	← ₆	↑ ₁

Value function:

- (b) These are the actions for the second episode. What is the updated value function?


Actions:

	5 ↓ ↑ ₄	
	6 ↓ ↑ ₃	← ₂
↑ ₈	← ₇	↑ ₁


Value Function:

Solution:


Actions:

	\downarrow_4	\leftarrow_3
	\downarrow_5	\uparrow_2
\uparrow_7	\leftarrow_6	\uparrow_1


Value after episode:

-1.0	-1.2	-1.2
	-1.2	-1.2
-1.0	-1.2	-1.2

Actions (2. Episode):

	$\begin{matrix} 5 \downarrow \uparrow 4 \end{matrix}$	
	$\begin{matrix} 6 \downarrow \uparrow 3 \end{matrix}$	\leftarrow_2
\uparrow_8	\leftarrow_7	\uparrow_1

Value after 2 episodes:

-1.0	-1.6	-1.2
	-1.56	-1.4
-1.0	-1.36	-1.4

Question 3: Q-Learning

Instead of prediction, we now want to use Q-learning as the control algorithm. To do this, we want to calculate the action-value function and then a better policy in each case. You have already performed iterations, and the current action values for each state in the direction of the actions N, E, S, and W are as shown below.

Initial action-value function:

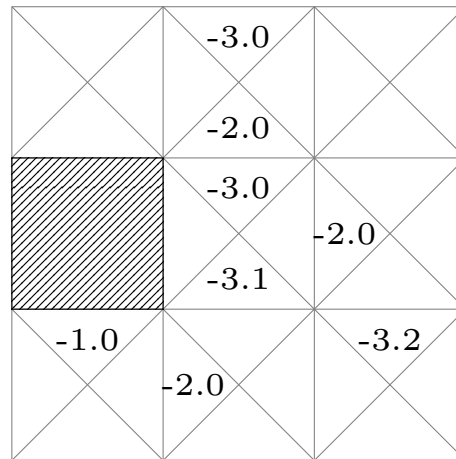
	-2.3	-3.2	-3.6
-2.4	-2.7	-2.0	-3.4
-1.0	-2.1	-2.8	
	-3.1	-3.2	
	-1.0	-3.2	-2.2
	-2.6	-3.4	
-1.0	-2.2	-2.7	
-2.2	-3.1	-2.1	-2.6
-2.1	-2.5	-3.7	

- (a) The actions drawn from the policy for each state are shown below. Calculate the action-value functions for the episode shown. This time, use a constant step size $\alpha = 1$ for the update. Only enter the values that have changed.

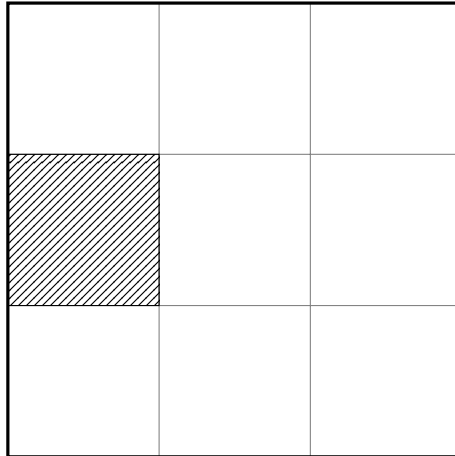
Actions:

	5 ↓ ↑ 4	
	6 ↓ ↑ 3 ← 2	
↑ 8 ← 7		↑ 1

Update the action-value function Q :

Solution:

- (b) Now calculate the greedy policy for all states from the values after the above update and plot them.



Solution:

