3. Q-learning Algorithm

Extension Note: Project 5 due date has been extended by 1 **more** day to **September 6 23:59UTC** .

In this section, you will implement the Q-learning algorithm, which is a model-free algorithm used to learn an optimal Q-function. In the tabular setting, the algorithm maintains the Q-value for all possible state-action pairs. Starting from a random Q-function, the agent continuously collects experiences $(s, c, R(s, c), s')$ and updates its Q-function.

From now on, we will refer to $c = (a, b)$ as “an action" although it really is an action with an object.

Q-learning Algorithm

- The agent plays an action c at state s, getting a reward $R(s, c)$ and observing the next state s'.
- Update the single Q-value corresponding to each such transition:

$$Q(s,c) \leftarrow (1-\alpha)\, Q(s,c) + \alpha\left[R(s,c) + \gamma \max_{c' \in C} Q(s',c')\right]$$

Tip: We recommend you implement all functions from this tab and the next one before submitting your code online. Make sure you achieve reasonable performance on the *Home World* game

---

Single step update

1.0/1 point (graded)

Write a function `tabular_q_learning` that updates the single Q-value, aiven the transition date $(s, c, R(s,c), s')$.

Reminder: You should implement this function locally first. You can read through the next tab to understand the context in which this function is called

Available Functions: You have access to the NumPy python library as `np`. You should also use constants `ALPHA` and `GAMMA` in your code

```
def tabular_q_learning(q_func, current_state_1, current_state_2, action_index,
                       object_index, reward, next_state_1, next_state_2,
                       terminal):
    """Update q_func for a given transition

    Args:
        q_func (np.ndarray): current Q-function
        current_state_1, current_state_2 (int, int): two indices describing the current state
        action_index (int): index of the current action
        object_index (int): index of the current object
        reward (float): the immediate reward the agent recieves from playing current command
        next_state_1, next_state_2 (int, int): two indices describing the next state
```

```
13        terminal (bool): True if this epsiode is over
14
15
```

Press ESC then TAB or click outside of the code editor to exit

Correct

```
def tabular_q_learning(q_func, current_state_1, current_state_2, action_index,
                       object_index, reward, next_state_1, next_state_2,
                       terminal):
    """Update q_func for a given transition

    Args:
        q_func (np.ndarray): current Q-function
        current_state_1, current_state_2 (int, int): two indices describing the current state
        action_index (int): index of the current action
        object_index (int): index of the current object
        reward (float): the immediate reward the agent recieves from playing current command
        next_state_1, next_state_2 (int, int): two indices describing the next state
        terminal (bool): True if this epsiode is over

    Returns:
        None
    """
    if terminal:
        maxq_next_state = 0
    else:
        q_values_next_state = q_func[next_state_1, next_state_2, :, :]
        maxq_next_state = np.max(q_values_next_state)

    q_value = q_func[current_state_1, current_state_2, action_index,
                     object_index]
    q_func[current_state_1, current_state_2, action_index, object_index] = (
        1 - ALPHA) * q_value + ALPHA * (reward + GAMMA * maxq_next_state)
```

Test results

Hide output

CORRECT

Test: chooses correct nonterminal

Testing performs correct update when terminal = False

Output:

```
[1 4 1 0]
0.777989
[1 3 2 4]
0.010239
[1 4 2 0]
0.678928
[1 1 1 6]
0.671948
[1 0 1 6]
0.337290
[0 3 1 1]
0.251706
[1 4 1 0]
0.887469
[1 3 1 4]
0.066530
[0 2 1 2]
0.217068
[0 0 0 4]
0.402047
```

Test: chooses correct terminal

Testing performs correct update when terminal = True

Output:

```
[1 2 2 3]
-0.004490
[0 0 2 1]
0.459124
[0 2 2 3]
0.404280
[0 3 2 4]
0.354726
[0 2 1 4]
0.468082
[1 0 0 5]
0.488427
[0 3 0 1]
0.769790
[1 0 0 0]
0.268233
[1 0 2 0]
0.367245
[1 4 1 1]
0.544434
```

Hide output

Submit

You have used 3 of 25 attempts

Answers are displayed within the problem

Epsilon-greedy exploration

1.0/1 point (graded)

Note that the Q-learning algorithm does not specify how we should interact in the world so as to learn quickly. It merely updates the values based on the experience collected. If we explore randomly, i.e., always select actions at random, we would most likely not get anywhere. A better option is to exploit what we have already learned, as summarized by current Q-values. We can always act greedily with respect to the current estimates, i.e., take an action $\pi(s) = \arg\max_{c \in C} Q(s, c)$. Of course, early on, these are not necessarily very good actions. For this reason, a typical exploration strategy is to follow a so-called ε-greedy policy: with probability ε take a random action out of C with probability $1 - \varepsilon$ follow $\pi(s) = \arg\max_{c \in C} Q(s, c)$. The value of ε here balances exploration vs exploitation. A large value of ε means exploring more (randomly), not using much of what we have learned. A small ε, on the other hand, will generate experience consistent with the current estimates of Q-values.

Now you will write a function `epsilon_greedy` that implements the ε-greedy exploration policy using the current Q-function.

Reminder: You should implement this function locally first. You can read through the next tab to understand the context in which this function is called

Available Functions: You have access to the NumPy python library as `np`. Your code should also use constants `NUM_ACTIONS` and `NUM_OBJECTS`.

```
def epsilon_greedy(state_1, state_2, q_func, epsilon):
    """Returns an action selected by an epsilon-Greedy exploration policy

    Args:
        state_1, state_2 (int, int): two indices describing the current state
        q_func (np.ndarray): current Q-function
```

```
        epsilon (float): the probability of choosing a random command

    Returns:
        (int, int): the indices describing the action/object to take
    """
    # TODO Your code here
    explore = np.random.random() <= epsilon
    if explore:
       action_index, object_index = np.random.choice(NUM_ACTIONS, 1)[0], np.random.choice(NUM_OBJECTS, 1)[0]
```

Press ESC then TAB or click outside of the code editor to exit

Correct

```
def epsilon_greedy(state_1, state_2, q_func, epsilon):
    """Returns an action selected by an epsilon-Greedy exploration policy

    Args:
        state_1, state_2 (int, int): two indices describing the current state
        q_func (np.ndarray): current Q-function
        epsilon (float): the probability of choosing a random command

    Returns:
        (int, int): the indices describing the action/object to take
    """
    coin = np.random.random_sample()
    if coin < epsilon:
        action_index = np.random.randint(NUM_ACTIONS)
        object_index = np.random.randint(NUM_OBJECTS)
    else:
        q_values = q_func[state_1, state_2, :, :]
        (action_index,
         object_index) = np.unravel_index(np.argmax(q_values, axis=None),
                                          q_values.shape)

    return (action_index, object_index)
```

Test results

Hide output

CORRECT

```
        Test: must choose best
```

Testing returns correct action for epsilon = 0

Output:

```
(2, 1)
(0, 3)
(0, 2)
(0, 1)
(2, 1)
(0, 6)
(1, 4)
(1, 6)
(0, 4)
(2, 3)
```

Test: must choose random

Testing returns random action for epsilon = 1

Output:

```
Test passed
```

Hide output

Submit You have used 2 of 25 attempts

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Unit 5 Reinforcement Learning (2 weeks) :Project 5: Text-Based Game / 3. Q-learning Algorithm

Add a Post

Show all posts ▼ by recent activity ▼

?	epsilon greedy -- After thought I know where in exam time. I have this final question that has nothing to do with final exam. However, I understand the code in the *epsilon-gree...	1
?	[Staff]Can you plz check my code for the single step update?Tks. I thought I got the answer based on your previous insights(thanks very much) but now my output is very weird with right answer but extra eleme...	2
?	Q-learning Algorithm - Updating all the Q's Do I have to recompute all the Q's before selecting the max(Q(s',c'))?	7
?	[HELP] Could Any STAFF help to check my code about "Epsilon-greedy exploration"? Could any staff help to check my code, please? I have already applied the concept as what others' posts metioned and passed the Epsilon==1 cas...	6
?	Single step update [Staff] initializing q_func to zero is useless, could you please correct it?	6
?	Last attempt: Epsilon-greedy: not sure how to move from here Test: must choose best Testing returns correct action for epsilon = 0 Your output: (0, 6) (1, 0) (2, 0) (1, 0) (1, 6) (0, 5) (0, 0) (2, 5) (1, 1) (3, 5) Correct o...	4
?	[Staff]Can you plz check my code for the single step update?Tks. Keep getting the message that the np.array is not callable.	2

☑	Can anyone help point out what is wrong with my code for the single step update? I assume that I can post it here since it is wrong... if terminal is True: fMaxQ = 0 else: fMaxQ=np.max(q_func[next_state_1, next_state_2, :, :]) q_fun...	3
?	my code for q learning and e-greedy can you check and guide why the submitted code is wrong? what am I missing? what should I think about?	5
💬	[STAFF] An extension of this project's deadline please?? Project is too challenging with too little theoretical background and coding practice I'm devoting as much time as I can to this project but this is really challenging. I spend a lot of time trying to just understand the setup of the ga...	6
?	Why do we need to use the alpha/exponential average for Q-learning if the reward and the transitions are deterministic? Dear Staff, IF the rewards and the transitions are all deterministic, then why are we updating Q-func through alpha/exponential average. One sa...	9
💬	Epsilon-greedy exploration Hint Take a look at np.unravel_index and np.argmax. I've solved it with other functions first and answer was marked correct, but then it was failed in ...	1
?	Why is my IDE complaining about my slice indexing?	2