به نام خدا

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

مبانی و کاربردهای هوش مصنوعی (پاییز ۱۴۰۱)

گزارش پروژه - فاز سوم

استاد درس:

دکتر جوانمردی

نام دانشجو:

محمدجواد رضوانیان

مهلت ارسال تمرین: ۳۰ دی ماه ۱۴۰۱

فاز سوم، یادگیری تقویتی

۱. پاسخ سوال ۱، تکرار ارزش

در این مرحله قصد داریم که با استفاده از مقادیر ارزش بدست آمده در گام‌ها و حالات قبلی، مقادیر حالات بعدی را در آینده بروزرسانی کنیم. برای انجام این کار از تابع getStates() استفاده میکنیم؛ این تابع تمام حالات محیط را بر می‌گرداند. تابع getPossibleAction(state) با دریافت یک حالت می‌تواند تمام اقدامات ممکن را برای حالت مورد نظر برگرداند. تابع getQValue(state, action) یک مقدار QValue بر می‌گرداند. در حقیقت سیاست بدست آمده در قدم k ام، مقدار جایزه مورد انتظار برای قدم k + ۱ را به ما خواهد گفت.

۱.۱ بررسی کد

ما در حقیقت باید توابع runValueIteration()، computeQValueFromValues() و computeActionsFromValues() را تغییر دهیم. توابع مذکور به صورت زیر تغییر می‌کنند:

```
    def runValueIteration(self):
        # Write value iteration code here
        "*** YOUR CODE HERE ***"
        for state in self.mdp.getStates(): #get all the states
            self.values[state] = 0.0

        for i in range(self.iterations):   #run for these many iterations
            next_values = self.values.copy() #copy back the old values
            for state in self.mdp.getStates():
                state_values = util.Counter() #values for actions of this state
                for action in self.mdp.getPossibleActions(state):
                    state_values[action] = self.getQValue(state, action)
                next_values[state] = state_values[state_values.argMax()]  #update for each state
            self.values = next_values.copy() #copy back the new values
    def computeQValueFromValues(self, state, action):
        """
          Compute the Q-value of action in state from the
          value function stored in self.values.
        """
        "*** YOUR CODE HERE ***"
        transition_probs = self.mdp.getTransitionStatesAndProbs(state, action)
        QValue = 0.0
        for transition in transition_probs:
            Tstate, prob = transition
            QValue += prob * (self.mdp.getReward(state, action, Tstate) + self.discount *
self.getValue(Tstate))

        return QValue

        util.raiseNotDefined()
    def computeActionFromValues(self, state):
```

```
        """
          The policy is the best action in the given state
          according to the values currently stored in self.values.

          You may break ties any way you see fit.  Note that if
          there are no legal actions, which is the case at the
          terminal state, you should return None.
        """
        "*** YOUR CODE HERE ***"
        if (self.mdp.isTerminal(state)):
            return None
        else:
            QValues = util.Counter()
            actions = self.mdp.getPossibleActions(state)
            for action in actions:
                QValues[action] = self.computeQValueFromValues(state, action)

            return QValues.argMax()

        util.raiseNotDefined()
```

همانطور که مشاهده می‌کنید، در توابع بالا قبل از رسیدن به util.raiseNotDefined() تابع به return رسیده و امید داریم که برنامه با این خطا مواجه نشود.

در تابع computeActionFromValues() برای برگرداندن مقدار QValues از تابع argmax() استفاده شده است.

۲.۱ خروجی‌های آزمایش

خروجی کد مربوطه با استفاده از دستور داده شده به صورت زیر خواهد بود:

```
python autograder.py -q q1
```

```
P3 python autograder.py -q q1
D:\Projects\Python\Pacman World\P3\autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see th
e module's documentation for alternative uses
  import imp
Starting on 1-20 at 19:25:22

Question q1
===========

*** PASS: test_cases\q1\1-tinygrid.test
*** PASS: test_cases\q1\2-tinygrid-noisy.test
*** PASS: test_cases\q1\3-bridge.test
*** PASS: test_cases\q1\4-discountgrid.test

### Question q1: 4/4 ###


Finished at 19:25:22

Provisional grades
==================
Question q1: 4/4
------------------
Total: 4/4

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

همانطور که مشاهده می‌کنید، عامل ما نمره ۴ از ۴ را برای این سوال کسب کرده است.

۲. بررسی گذر از پل

برای بهینه سازی این مسئله، با انجام آزمایشات متعدد (به این صورت که مقدار answerNoise را هر دفعه تقسیم بر ۲ می‌کنیم تا کم شود و به مقدار مطلوب ما برسد) می‌توانیم مقدار تقریبی ۰.۰۰۲۵ یا حتی برای اطمینان بیشتر مقدار ۰.۰۰۱۲۵ را در آن قرار دهیم. (راستش نمی‌خواستم که مقدار تخفیف را دستکاری کنم و با یک سرچ در اینترنت دیدم که می‌شود با این راه سریع تر به مقدار بهینه نزدیک شد)

علت اصلی هم این است که عامل ما باید یک مقدار نزدیک صفر برای noise داشته باشد تا بتواند به سیاست مناسب خود برسد.

۱.۲ بررسی کد

همانطور که گفته شد، مقدار answerNoise را به مقدار ۰.۰۰۱۲۵ کاهش می‌دهیم تا به پاسخ مورد نظر نزدیک شویم:

```
def question2():
    answerDiscount = 0.9
    answerNoise = 0.00125
    return answerDiscount, answerNoise
```

۲.۲ خروجی

خروجی قطعه کد ما بعد از اجرای دستور آمده در دستور کار به صورت زیر خواهد بود:

```
python autograder.py -q q۲
```

```
P3 python autograder.py -q q2
D:\Projects\Python\Pacman World\P3\autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see th
e module's documentation for alternative uses
  import imp
Starting on 1-20 at 19:36:59

Question q2
===========

*** PASS: test_cases\q2\1-bridge-grid.test

### Question q2: 1/1 ###


Finished at 19:36:59

Provisional grades
==================
Question q2: 1/1
------------------
Total: 1/1

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

۳. سیاست ها

در اینجا باید مقادیر سه‌گانه‌ی تخفیف، نویز و جایزه زندگی را به گونه‌ای تغییر دهیم که:

- خروجی نزدیک را ترجیح دهد و رسیک صخره را بپذیرد.
- خروجی نزدیک را ترجیح دهد اما از صخره اجتناب کند.
- خروجی دور را ترجیح دهد و ریسک صخره را بپذیرد.
- خروجی دور را ترجیح دهد اما از صخره اجتناب کند.
- از همه چیز اجتناب کند.

۱.۳ بررسی کد

باید ۵ تابع از فایل analysis.py را تغییر داد تا به خروجی مورد نظر رسید:

```
def question3a():
    answerDiscount = 0.2
    answerNoise = 0.0
    answerLivingReward = 0.0
    return answerDiscount, answerNoise, answerLivingReward

def question3b():
    answerDiscount = 0.25
    answerNoise = 0.2
    answerLivingReward = 0.0
    return answerDiscount, answerNoise, answerLivingReward

def question3c():
    answerDiscount = 0.9
    answerNoise = 0.0
    answerLivingReward = 0.0
    return answerDiscount, answerNoise, answerLivingReward

def question3d():
    answerDiscount = 0.9
    answerNoise = 0.2
    answerLivingReward = 0.0
    return answerDiscount, answerNoise, answerLivingReward

def question3e():
    answerDiscount = 0.9
    answerNoise = 0.2
    answerLivingReward = 1.0
    return answerDiscount, answerNoise, answerLivingReward
```

همانطور که مشاهده میکنید، برای اینکه عامل ما از صخره اجتناب کند، مقدار answerNoise را باید زیاد کرد (۰ یا ۰.۲) و برای اینکه عامل ما نزدیک ترین خروجی را ترجیح دهد، باید مقدار تخفیف در نظر گرفته شده را کم کرد (فاصله آن را از ۱ زیاد

کرد تا اینکه عامل ما بتواند مقدار نزدیک را انتخاب کند) اگر مقدار تخفیف عددی بین ۰ تا ۱ باشد، هر چه این مقدار بزرگ تر باشد، آن وقت عامل ما به خروجی دورتر را انتخاب می‌کند.

۲.۳ خروجی

خروجی پروژه ما به صورت زیر خواهد بود:

```
python autograder.py -q q۳
```

```
P3 python autograder.py -q q3
D:\Projects\Python\Pacman World\P3\autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see th
e module's documentation for alternative uses
  import imp
Starting on 1-20 at 20:25:51

Question q3
===========

*** PASS: test_cases\q3\1-question-3.1.test
*** PASS: test_cases\q3\2-question-3.2.test
*** PASS: test_cases\q3\3-question-3.3.test
*** PASS: test_cases\q3\4-question-3.4.test
*** PASS: test_cases\q3\5-question-3.5.test

### Question q3: 5/5 ###


Finished at 20:25:51

Provisional grades
==================
Question q3: 5/5
------------------
Total: 5/5

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

همانطور که مشاهده می‌کنید عامل ما توانسته است که تمام ۵ تست را پاس کند.

۴. تکرار ارزش ناهمزمان

در این روش، به جای اینکه به صورت دسته‌ای بروزرسانی ها انجام شود، فقط یک حالت را بروزرسانی می‌کنیم.

۱.۴ بررسی کد

برای پیاده سازی این قسمت، باید تنها متد درون آن را پیاده سازی کرد چون به صورت بازگشتی دوباره به آن ارجاع داده میشود:

```
    def runValueIteration(self):
        "*** YOUR CODE HERE ***"
        self.values = collections.defaultdict(float)
        states = self.mdp.getStates()
        for state in states:
            self.values[state] = 0

        "*** YOUR CODE HERE ***"
        statesSize = len(states)

        for i in range(self.iterations):
            state = states[i % statesSize]

            if not self.mdp.isTerminal(state):
                A = self.computeActionFromValues(state)
                Q = self.computeQValueFromValues(state, A)
                self.values[state] = Q
```

در متد بالا در ابتدا یک ValueIterationAgent گرفته میشود و سپس به صورت بازگشتی سازنده آن صدا زده میشود. سپس مقادیر آن از مجموعه کلاس دیکشنری مقادیر اولیه خوانده میشود و سپس از آن به عنوان یک آرایه برای بروزرسانی استفاده میشود. حالا باید تمام حالت های ممکن را نیز به همین روش نگهداری کرد و با این حالات و مقادیر، یک آرایه با مقادیر صفر ایجاد کرد.

حالا به تعداد عملیات iteration که میتوانیم انجام دهیم، مقدار حالت را بروزرسانی میکنیم. حالا اگه حالت ما ترمینال نبود، اقدام مناسب را انتخاب کرده و سپس مقدار QValue را برای آن قرار میدهیم.

۲.۴ خروجی

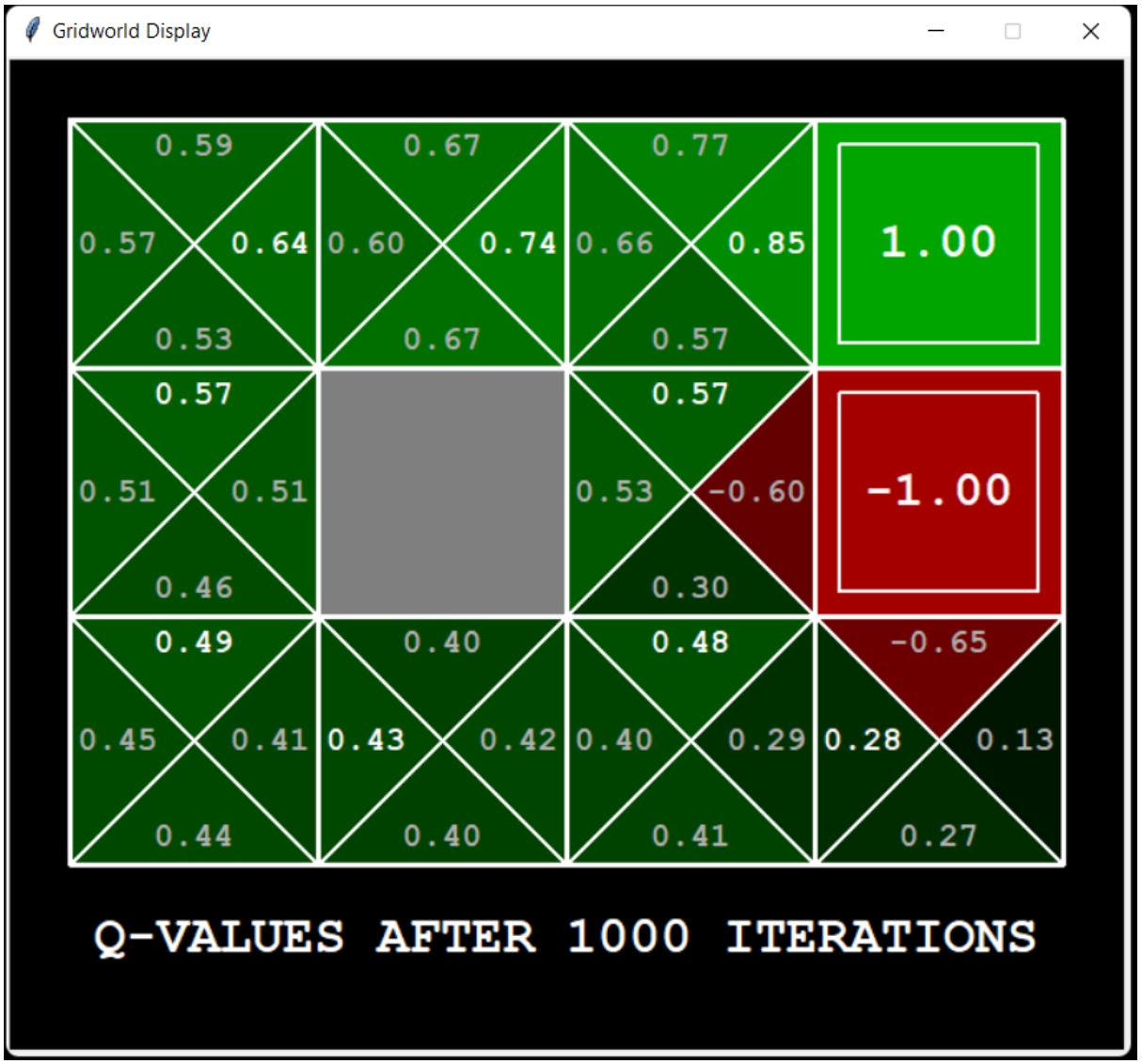
خروجی ما بعد از اجرای دستور آمده در دستورکار، به صورت زیر است:

```
python autograder.py -q q4
```

```
P3 python autograder.py -q q4
D:\Projects\Python\Pacman World\P3\autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see th
e module's documentation for alternative uses
  import imp
Starting on 1-21 at 21:42:15

Question q4
===========

*** PASS: test_cases\q4\1-tinygrid.test
*** PASS: test_cases\q4\2-tinygrid-noisy.test
*** PASS: test_cases\q4\3-bridge.test
*** PASS: test_cases\q4\4-discountgrid.test

### Question q4: 1/1 ###


Finished at 21:42:15

Provisional grades
==================
Question q4: 1/1
------------------
Total: 1/1

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

حالا اگر محیط شبیه‌سازی را اجرا کنیم، می‌بینیم که خروجی به صورت زیر است:

```
python gridworld.py -a asynchvalue -i ۱۰۰۰ -k ۱۰
```



گزارش خروجی به صورت زیر است:

```
EPISODE 10 COMPLETE: RETURN WAS 0.43046721000000016


AVERAGE RETURNS FROM START STATE: 0.4954329198000001
```

۵. تکرار ارزش اولویت بندی شده

۱.۵ بررسی کد

کد پیاده‌سازی تابع مورد نظر به صورت زیر است:

```
    def runValueIteration(self):
        "*** YOUR CODE HERE ***"

        mdp = self.mdp
        values = self.values
        discount = self.discount
        iterations = self.iterations
        theta = self.theta
        states = mdp.getStates()


        predecessors = {} # dict
        for state in states:
          predecessors[state] = set()

        pq = util.PriorityQueue()

        # computes predecessors and puts initial stuff into pq
        for state in states:
          Q_s = util.Counter()

          for action in mdp.getPossibleActions(state):
            # assigning predecessors
            T = mdp.getTransitionStatesAndProbs(state, action)
            for (nextState, prob) in T:
              if prob != 0:
                predecessors[nextState].add(state)

            # computing Q values for determining diff's for the pq
            Q_s[action] = self.computeQValueFromValues(state, action)

          if not mdp.isTerminal(state): # means: if non terminal state
            maxQ_s = Q_s[Q_s.argMax()]
            diff = abs(values[state] - maxQ_s)
            pq.update(state, -diff)


        # now for the actual iterations
        for i in range(iterations):
          if pq.isEmpty():
            return

          state = pq.pop()
```

```
        if not mdp.isTerminal(state):
          Q_s = util.Counter()
          for action in mdp.getPossibleActions(state):
            Q_s[action] = self.computeQValueFromValues(state, action)

          values[state] = Q_s[Q_s.argMax()]

        for p in predecessors[state]:
          Q_p = util.Counter()
          for action in mdp.getPossibleActions(p):
            # computing Q values for determining diff's for the pq
            Q_p[action] = self.computeQValueFromValues(p, action)

          #if not mdp.isTerminal(state): # means: if non terminal state
          maxQ_p = Q_p[Q_p.argMax()]
          diff = abs(values[p] - maxQ_p)

          if diff > theta:
            pq.update(p, -diff)
```

یبس

۲.۵ خروجی

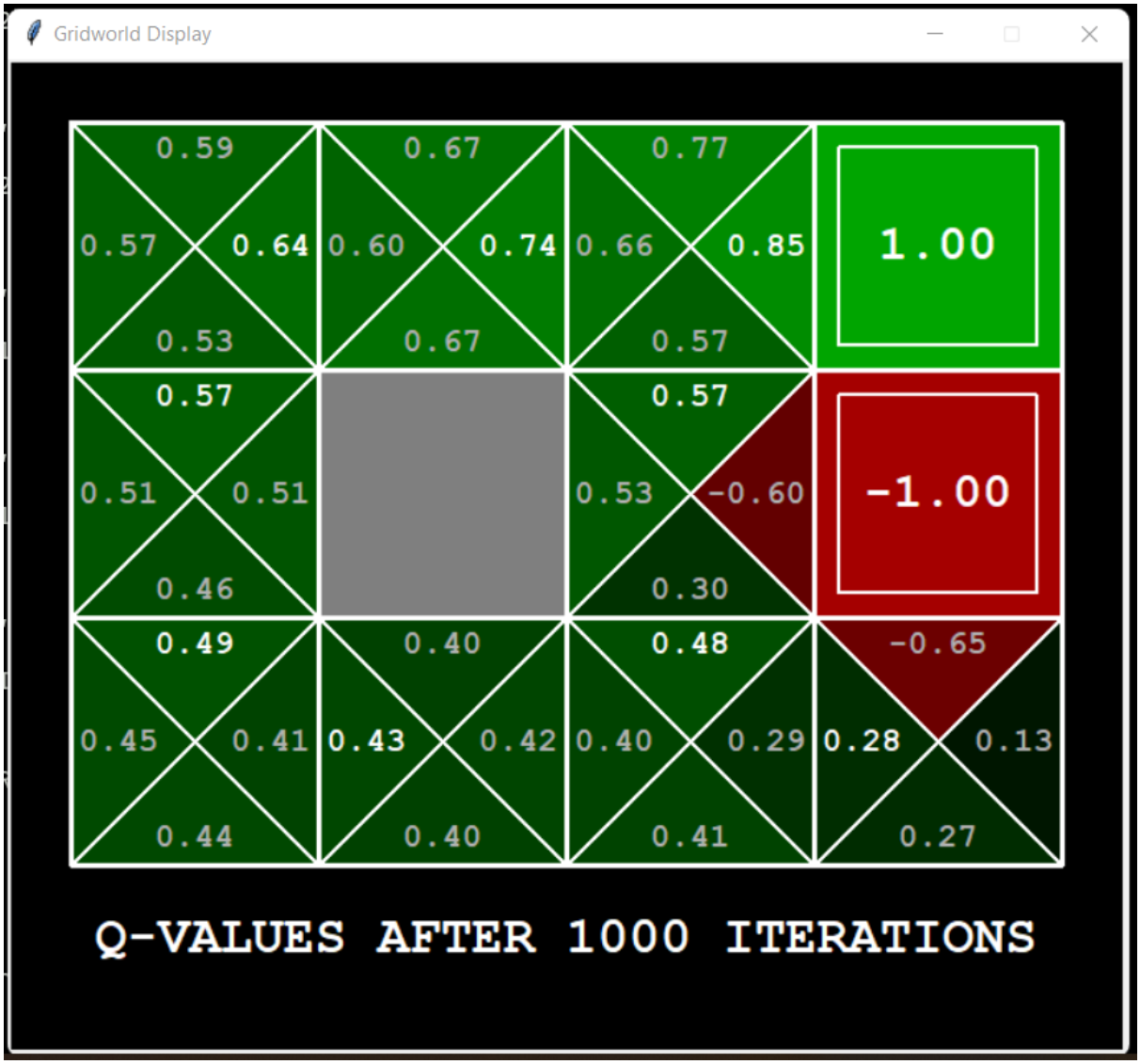
خروجی کد ما با استفاده از دستور مذکور در دستور کار به صورت زیر است:

```
python autograder.py -q q۵
```

```
P3 python autograder.py -q q5
D:\Projects\Python\Pacman World\P3\autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see th
e module's documentation for alternative uses
  import imp
Starting on 1-21 at 21:56:58

Question q5
===========

*** PASS: test_cases\q5\1-tinygrid.test
*** PASS: test_cases\q5\2-tinygrid-noisy.test
*** PASS: test_cases\q5\3-bridge.test
*** PASS: test_cases\q5\4-discountgrid.test

### Question q5: 3/3 ###


Finished at 21:56:58

Provisional grades
==================
Question q5: 3/3
----------------
Total: 3/3

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

وضعیت خروجی های ما:


Gridworld Display
0.64
0.74
0.85
1.00
0.57
0.57
-1.00
0.49
0.43
0.48
0.28
VALUES AFTER 1000 ITERATIONS


Gridworld Display
0.59
0.57
0.64
0.53
0.67
0.60
0.74
0.67
0.77
0.66
0.85
0.57
1.00
0.57
0.51
0.51
0.46
0.57
0.53
-0.60
0.30
-1.00
0.49
0.45
0.41
0.44
0.40
0.43
0.42
0.40
0.48
0.40
0.29
0.41
-0.65
0.28
0.13
0.27
Q-VALUES AFTER 1000 ITERATIONS

۶ یادگیری Q

برای حل این قسمت فقط باید متد getAction() را پیاده سازی کرد. برای اینکار باید از util.flipCoin() استفاده کرد و مقدار epsilon را به آن پاس داد. (راستش فقط از همین نکات و راهنمایی هایی که در ابتدای متد نوشته شده استفاده کردم و هیچ ویژگی یا راه حل خاص دیگری را استفاده نکردم)

۱.۶ بررسی کد

```
    def getAction(self, state):
        """
          Compute the action to take in the current state.  With
          probability self.epsilon, we should take a random action and
          take the best policy action otherwise.  Note that if there are
          no legal actions, which is the case at the terminal state, you
          should choose None as the action.

          HINT: You might want to use util.flipCoin(prob)
          HINT: To pick randomly from a list, use random.choice(list)
        """
        # Pick Action
        legalActions = self.getLegalActions(state)
        action = None
        "*** YOUR CODE HERE ***"
        if len(legalActions) == 0:
          return action
        else:
          boolean = util.flipCoin(self.epsilon)
          if boolean:
            action = random.choice(legalActions)
          else:
            action = self.computeActionFromQValues(state)
        return action
```

۲.۶ خروجی

```
P3 python autograder.py -q q7
D:\Projects\Python\Pacman World\P3\autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see th
e module's documentation for alternative uses
  import imp
Starting on 1-21 at 22:16:13

Question q7
===========

*** PASS: test_cases\q7\1-tinygrid.test
*** PASS: test_cases\q7\2-tinygrid-noisy.test
*** PASS: test_cases\q7\3-bridge.test
*** PASS: test_cases\q7\4-discountgrid.test

### Question q7: 2/2 ###


Finished at 22:16:14

Provisional grades
==================
Question q7: 2/2
------------------
Total: 2/2

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

۸. بررسی دوباره عبور از پل

۱.۸ بررسی کد

راستش خسته شدم و تو اینترنت سرچ کردم دیدم که وقتی که ۵۰ بار عملیات عامل را تکرار کنیم متوجه میشویم که باید مقادیر ما بسیار زیاد و بزرگ باشند. **پس بهترین راه حل برای این صورت مسئله، پاک کردن صورت مسئله است!** 😅

```python
def question8():
    return 'NOT POSSIBLE'
    # If not possible, return 'NOT POSSIBLE'
```

۲.۸ خروجی

```
(P3) python autograder.py -q q8
D:\Projects\Python\Pacman World\P3\autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see th
e module's documentation for alternative uses
  import imp
Starting on 1-21 at 22:20:37

Question q8
===========

*** PASS: test_cases\q8\grade-agent.test

### Question q8: 1/1 ###


Finished at 22:20:37

Provisional grades
==================
Question q8: 1/1
------------------
Total: 1/1

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

۹. پک من و یادگیری

۱.۹ بررسی کد

هیچ تغییری در کد ایجاد نکردم!!

۲.۹ خروجی

```
Average Score: 479.76
Scores:        495.0, 503.0, 503.0, 503.0, 499.0, 499.0, 503.0, 503.0, 503.0, 495.0, 503.0, 495.0, 503.0, 503.0, 503.0, 503.0, 499.0, 503.0, 495.0, 495.0, 499.0, 499.0, 4
95.0, 499.0, 495.0, 495.0, 499.0, 499.0, 503.0, 499.0, -514.0, 503.0, 503.0, 495.0, 503.0, 495.0, 503.0, 503.0, 495.0, 495.0, 495.0, 503.0, 495.0, 499.0, 499.0, 499.0, 50
3.0, 499.0, 499.0, 503.0, 503.0, 503.0, 503.0, 503.0, 499.0, 495.0, 503.0, 503.0, 503.0, 503.0, 503.0, 503.0, 495.0, 503.0, 503.0, 495.0, 503.0, 499.0, 503.0, 503.0, 495.
0, 499.0, 495.0, 503.0, 503.0, -512.0, 503.0, 503.0, 499.0, 499.0, 503.0, 503.0, 499.0, 499.0, 503.0, 495.0, 503.0, 503.0, 495.0, 499.0, 503.0, 503.0, 503.0, 499.0, 499.0
, 495.0, 495.0, 499.0, 503.0, 495.0
Win Rate:      98/100 (0.98)
Record:        Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Loss,
 Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win,
 Win, Win, Win, Win, Win, Win, Win, Win, Win, Loss, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
, Win
*** PASS: test_cases\q9\grade-agent.test (1 of 1 points)
***     Grading agent using command:  python pacman.py -p PacmanQAgent -x 2000 -n 2100 -l smallGrid -q -f --fixRandomSeed
***     98 wins (1 of 1 points)
***         Grading scheme:
***          < 70:  0 points
***         ≥ 70:  1 points

### Question q9: 1/1 ###


Finished at 22:32:06

Provisional grades
==================
Question q9: 1/1
------------------
Total: 1/1

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```

۱۰. یادگیری تقریبی

۱.۱۰ بررسی کد

برای حل این مسئله، کافی است که دو متد getValue() و update() را کامل کنیم. متد اول وظیفه این را دارد که همان مقدار وزن‌ها در ویژگی را برگرداند و متد دوم وظیفه بروزرسانی مقدار difference در فرمول مطرح شده در کلاس را دارد.

```
    def getQValue(self, state, action):
        "*** YOUR CODE HERE ***"
        features = self.featExtractor.getFeatures(state,action)
        QValue = 0.0
        for feature in features:
            QValue += self.weights[feature] * features[feature]
        return QValue
    def update(self, state, action, nextState, reward):
        "*** YOUR CODE HERE ***"
        QValue = 0
        difference = reward + (self.discount * self.getValue(nextState) - self.getQValue(state,
action))
        features = self.featExtractor.getFeatures(state, action)
        for feature in features:
          self.weights[feature] += self.alpha * features[feature] * difference
```

۲.۱۰ خروجی

خروجی کد ما به صورت زیر است.

```
P3 python autograder.py -q q10
D:\Projects\Python\Pacman World\P3\autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib and slated for removal in Python 3.12; see th
e module's documentation for alternative uses
  import imp
Starting on 1-21 at 22:35:30

Question q10
============

*** PASS: test_cases\q10\1-tinygrid.test
*** PASS: test_cases\q10\2-tinygrid-noisy.test
*** PASS: test_cases\q10\3-bridge.test
*** PASS: test_cases\q10\4-discountgrid.test
*** PASS: test_cases\q10\5-coord-extractor.test

### Question q10: 3/3 ###


Finished at 22:35:30

Provisional grades
==================
Question q10: 3/3
------------------
Total: 3/3

Your grades are NOT yet registered.  To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```