

گزارش تمرین هوش مصنوعی

ریحانه درفشی ۶۱۰۳۹۶۰۹۸

توضیح کلی:

هدف در این تمرین حل پازل با استفاده از روش یادگیری تقویتی است. به این صورت که اگر حرکت درست انجام شد مدل پاداش میگیرد و به سمتی حرکت میکند که پاداش خود را ماکزیمم کند. در ده هزار اپیزود به انجام این کار میپردازیم و میانگین پاداش ها در هر هزار اپیزود و دیاگرام نهایی شامل استییت ها و اکشن ها به خروجی داده میشود.

توضیح کد:

```
def is_neighbor(i, j):
```

چک میکند آیا دو ورودی داده شده همسایه هستند یا خیر.

```
class Solver:
```

کلاس اصلی کد که در نهایت دیاگرام مورد نظر جهت حل مسئله را تشکیل میدهد.

```
self.q_table = np.array(np.zeros([9, 9]))
```

دیاگرام نهایی که تا متود ران آن را کامل میکند.

```
for step in range(self.max_steps_per_episodes):
    if random.uniform(0, 1) > self.exploration_rate:
        next_position = np.argmax(self.q_table[current_position])
    else:
        playable_actions = []
        for j in range(9):
            if is_neighbor(current_position, j) and not (len(route) >= 2 and
j == route[-2]):
                playable_actions.append(j)
            next_position = np.random.choice(playable_actions)

        state[current_position], state[next_position] = state[next_position],
state[current_position]
        done = tuple(state) == final_state
        reward = 1 if done else 0 # whether if we are done or not
```

یک قدم به جلو حرکت کرده و زنجیره تصمیم گیری را کامل میکند و تشخیص میدهد که حرکت درستی انجام شده است یا نه و با توجه به آن مقدار ریوارد را مشخص میکند.

```
self.q_table[current_position, next_position] =  
self.q_table[current_position, next_position] * (  
    1 - self.learning_rate) + self.learning_rate * (reward +  
self.discount_rate * np.max(  
    self.q_table[next_position]))
```

پایاده سازی فرمول محاسبه ترنزیشن های `q_table`

```
def result(self):
```

آماده سازی خروجی

خروجی:

```
(venv) rey@rey:~/artificial_inteligance$ python3 AI_HW1_qlearnin.py  
100% |  
0 : 0.053  
  
1000 : 0.02  
2000 : 0.031  
3000 : 0.035  
4000 : 0.033  
5000 : 0.035  
6000 : 0.031  
7000 : 0.022  
8000 : 0.053  
9000 : 0.058
```

میانگین پاداش ها در هر هزار اپیزود

```
0 :
0.0 ,0.7469 ,0.0 ,
0.7477 ,0.0 ,0.0 ,
0.0 ,0.0 ,0.0 ,

1 :
0.7487 ,0.0 ,0.7458 ,
0.0 ,0.7486 ,0.0 ,
0.0 ,0.0 ,0.0 ,

2 :
0.0195 ,0.2612 ,0.0 ,
0.0 ,0.0 ,0.7462 ,
0.0 ,0.0 ,0.0 ,

3 :
0.4798 ,0.0 ,0.0 ,
0.0 ,0.7433 ,0.0 ,
0.7465 ,0.0 ,0.0 ,

4 :
0.0 ,0.7457 ,0.0 ,
0.7439 ,0.0 ,0.7312 ,
0.0 ,0.7439 ,0.0 ,

5 :
0.0196 ,0.0 ,0.2928 ,
0.0 ,0.7445 ,0.0 ,
0.0 ,0.0 ,0.7476 ,

6 :
0.7474 ,0.0 ,0.0 ,
0.7481 ,0.0 ,0.0 ,
0.0 ,0.7451 ,0.0 ,

7 :
0.0312 ,0.0 ,0.0 ,
0.0 ,0.7451 ,0.0 ,
0.7435 ,0.0 ,0.7436 ,

8 :
0.7476 ,0.0 ,0.0 ,
0.0 ,0.0 ,0.7454 ,
0.0 ,0.7434 ,0.0 ,
(venv) rey@rey:~/artificial_intelligence$
```

دیگرام نهایی، بدین صورت برای خانه متناظر با هر عدد، مشخص شده که حرکت به کدام خانه احتمال موفقیت بیشتری را در پی دارد.