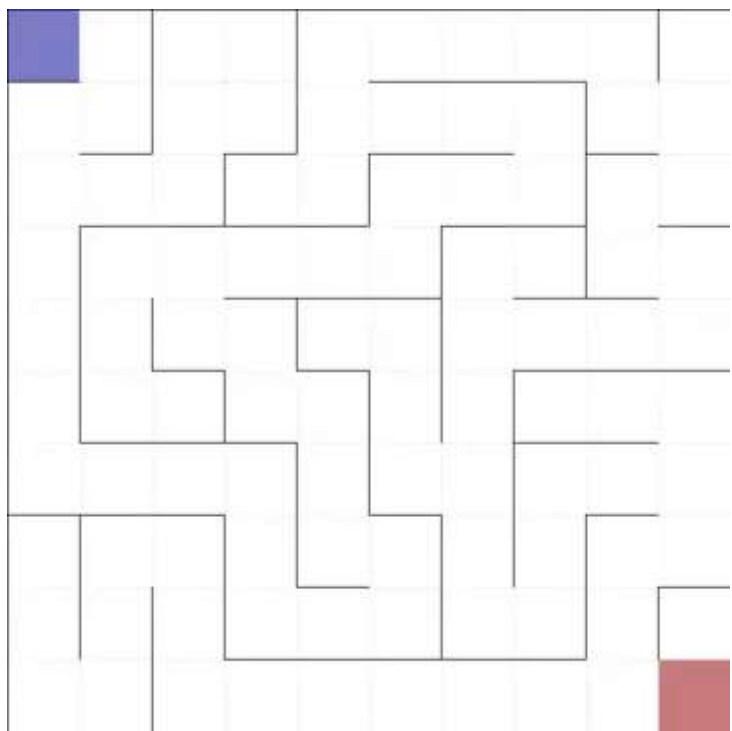


هزارتو



توضیح کلی

هدف این پروژه پیاده‌سازی یک عامل هوشمند است که بتواند در یک محیط غیرقطعی به فعالیت پرداخته و اهداف محیط را برآورده کند. برای درک هرچه بهتر کنش‌های عامل از محیط‌های ارث‌برده شده از فریمورک gym استفاده می‌کنیم. هدف این فاز از پروژه پیاده‌سازی تابعی برای انتخاب مناسب کنش‌ها با کمک الگوریتم QLearning یا SARSA است.

برای استفاده از این محیط، پس از دانلود بسته از [این لینک](#) در داخل پوشه دانلود شده ترمینال را باز کرده و دستور زیر را وارد کنید.

```
1 | python setup.py install
```

محیط

در این فاز از پروژه از محیط یک هزارتو استفاده می‌کنیم که متناسب با نیاز پروژه ما شخصی‌سازی شده است. این محیط شامل خانه شروع ($[0, 0]$)، خانه هدف ($[n, n]$)، دیوارهای ماز (به صورت تصادفی در هر اجرا)، خانه‌های پورتال (در صورتی که عامل وارد خانه پورتال شود به خانه هم‌رنگ همان خانه پورتال انتقال می‌یابد.) و عامل است. وظیفه شما هدایت عامل از خانه شروع به خانه هدف است. در واقع عامل ارائه شده باید بتواند در تعداد مشخصی اپیزود مسئله را یاد گرفته و بعد از آن بر اساس سیاست یادگرفته شده عمل کند به عبارت دیگر بعد از تعدادی اپیزود آموزشی یک یا چند اپیزود آزمایشی (حلقه for اصلی در کد) وجود خواهد داشت که عملکرد عامل‌ها بر اساس آن سنجیده می‌شود.

کنش‌های ممکن (action space)

عامل می‌تواند کنش‌های بالا (N)، راست (E)، پایین (S) و چپ (W) را در صورتی که حرکت آن مجاز باشد (با دیوار برخورد نداشته باشد) انتخاب نماید. نکته بسیار مهم اینکه کنش‌های عامل غیرقطعی هستند و به صورت زیر عمل می‌کنند: در صورتی که عامل قصد داشته باشد به هر سمتی حرکت کند تنها به احتمال ۰.۶ می‌تواند به آن سمت برود و به احتمال ۰.۲ به هر یک از کنش‌های همسایه‌اش می‌رود. در واقع اگر عامل حرکت به سمت بالا را انتخاب کند، تنها به احتمال ۰.۶ به سمت بالا رفته و به احتمال ۰.۲ به چپ و ۰.۲ به راست می‌رود. با توجه به اینکه کنش پایین در همسایگی کنش بالا قرار ندارد و دقیقاً روبه روی آن می‌باشد، احتمال انتخاب آن کنش صفر خواهد بود.

مشاهدات عامل در محیط (observation space)

موقعیت عامل یک زوج دوتایی (x,y) است که سطر و ستون کنونی عامل را نشان می‌شود. در صورتی که عامل در گوشه‌ها یا کنار دیوارها باشد و حرکتی انجام دهد که از زمین بخواهد خارجش کند در جای خود مانده و آن کنش اعمال نمی‌شود.

اتمام بازی (episode end)

بازی در حالات زیر پایان می‌یابد:

- در صورتی که عامل بتواند به هدف نهایی برسد
- در صورتی که تعداد کنش‌های مجاز عامل در محیط به پایان برسد.

پاداش در محیط (reward)

در این محیط به ازای هر کنش امتیاز $(-0.1)/n$ (n برابر تعداد سطر یا ستون محیط است.) و در صورتی که عامل در خانه هدف قرار گیرد امتیاز 1 لحاظ می‌گردد.

- توجه: در صورت نیاز می‌توانید پاداش‌های محیط را تغییر دهید (مثلاً در قالب تعریف یک تابع پاداش جدید) تا عامل بتواند سریع‌تر سیاست بهینه برای محیط را پیدا کند.

توابع استفاده شده در کد

تابع step

پس از انتخاب کنش مناسب با توجه به الگوریتمی که پیاده‌سازی کرده‌اید، لازم است آن را روی محیط اعمال کنید. بدین منظور از تابع step استفاده می‌شود که به عنوان ورودی یک کنش دریافت کرده و خروجی آن نتیجه اعمال آن کنش روی محیط به صورت یک tuple به شکل زیر است.


```
1 | (next_state: int, reward: int, done: bool, truncated: bool)
```

نکته: در صورتی که بازی به مشکلات سرور و یا زمان اجرا بر بخورد پارامتر truncated برابر مقدار True می‌شود و در صورتی که بازی تمام شود پارامتر done برابر مقدار True قرار می‌گیرد.

تابع `reset`

از این تابع برای برگرداندن عامل به نقطه شروع استفاده می‌شود و خروجی اول آن `next_state` نقطه شروع می‌باشد.

- توجه: برای انجام این بخش از پروژه لازم است کد خود را برای انتخاب کنش مناسب در قسمت `TODO` پیاده‌سازی کنید.

 `maze.py`

```
1 import gym
2 import gym_maze
3
4 # Create an environment
5 env = gym.make("maze-random-10x10-plus-v0")
6 observation = env.reset()
7
8 # Define the maximum number of iterations
9 NUM_EPISODES = 1000
10
11 for episode in range(NUM_EPISODES):
12
13     # TODO: Implement the agent policy here
14     # Note: .sample() is used to sample random action from the environment's act
15
16     # Choose an action (Replace this random action with your agent's policy)
17     action = env.action_space.sample()
18
19     # Perform the action and receive feedback from the environment
20     next_state, reward, done, truncated = env.step(action)
21
22     if done or truncated:
23         observation = env.reset()
24
25 # Close the environment
26 env.close()
```