

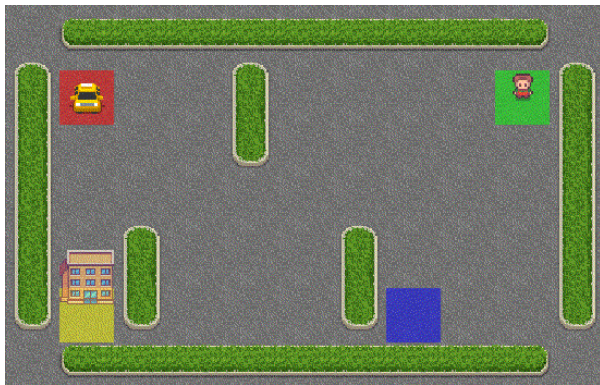


فاز دوم پروژه مبانی و کاربردهای هوش مصنوعی

۱- مقدمه

هدف از این مرحله از پروژه طراحی و پیاده‌سازی یک عامل یادگیر برای تعامل با محیط و دستیابی به اهداف است. به این منظور عامل باید با بکارگیری تکنیک‌های یادگیری تقویتی با استفاده از تجربیاتی که حین تعامل با محیط بدست می‌آورد کنش‌های مناسب را یاد بگیرد. تأکید اصلی در این مرحله از پروژه استفاده از روش‌های بدون مدل (model free) است.

علاوه بر محیط معرفی شده در مرحله قبلی پروژه (Frozen Lake)، در این مرحله از پروژه محیط دیگری بنام تاکسی (Taxi) هم باید در نظر گرفته شده و یک عامل یادگیر برای هدایت خودران تاکسی توسعه داده شود. برای اینکار همانند مرحله گذشته باید قسمت TODO از کد داده شده را تکمیل کنید. برای شبیه‌سازی محیط تاکسی از کتابخانه [OpenAI Gym](https://gymnasium.farama.org/) استفاده شده است. نکته بسیار مهم این است که برای استفاده از این محیط باید حتما نسخه Gym شما برابر ۰.۲۱.۰ باشد.



۲- معرفی محیط تاکسی

یکی از مهم‌ترین وظایف یک تاکسی رساندن هرچه سریع‌تر مسافران به مقاصد خود می‌باشد. در این جا می‌توانیم سرعت بیشتر را متناظر با پیمایش مسافت کمتر برای جابه‌جایی مسافران از یک نقطه به نقطه دیگر در نظر بگیریم. از آنجا که با توجه به پیشرفت تکنولوژی تاکسی‌های سنتی جای خود را به تاکسی‌های بدون راننده می‌دهند، هدف ما در محیط طراحی یک تاکسی خودران می‌باشد که بتواند مسافران را با کمترین حرکات ممکن به مقصد برساند. محیط استفاده شده در این مرحله از پروژه نسخه سوم محیط تاکسی است که در یک صفحه ۵×۵ تعریف شده است.

موجودیت‌های محیط:

در این محیط با چند نوع موجودیت روبه‌رو هستیم. اول خانه‌های خالی که عامل هوشمند (تاکسی) می‌تواند در این خانه‌ها قرار گیرد. دوم، دیوارها هستند که امکان عبور از آن توسط عامل وجود ندارد. موجودیت سوم خانه‌های رنگی هستند که مسافر و مقصد او در این خانه‌ها قرار می‌گیرند. تاکسی، مسافر و مقصد او (که با علامت ساختمان مشخص شده است) نیز دیگر موجودیت‌ها محیط هستند. مسافر و مقصد او در یکی از خانه‌های رنگی بازی (یکی از چهار خانه سبز، قرمز، زرد و آبی) قرار می‌گیرند. نقشه محیط همیشه ثابت است (محل خانه‌های رنگی و دیوارها تغییر نمی‌کند) اما محل قرارگیری اولیه مسافر، مقصد او و تاکسی در هر بازی به صورت تصادفی انتخاب می‌گردد.

کنش‌های ممکن در محیط:

تاکسی خودران باید مسافر را از خانه قرارگیری او سوار کرده و با طی کوتاه‌ترین مسافت ممکن مسافر را به مقصد رسانده و پیاده کند. کنش‌های قابل انجام برای عامل در این محیط عبارتند از:

(۱) کنش شماره صفر: حرکت به سمت جنوب (پایین)

(۲) کنش شماره یک: حرکت به سمت شمال (بالا)

(۳) کنش شماره دو: حرکت به سمت شرق (راست)

(۴) کنش شماره سه: حرکت به سمت غرب (چپ)

(۵) کنش شماره چهار: سوار کردن مسافر

(۶) کنش شماره پنج: پیاده کردن مسافر

همه کنش‌ها در این محیط قطعی هستند.

فضای مشاهده در محیط:

این محیط دارای ۵۰۰ مختلف است که از ترکیب ۲۵ مکان مختلف قرارگیری تاکسی، ۵ موقعیت قرارگیری مسافر و ۴ موقعیت برای مقصد مسافر بدست می‌آید. هر حالت توسط چندتایی \langle سطر مکان تاکسی، ستون مکان تاکسی، موقعیت مسافر، موقعیت مقصد مسافر \rangle مشخص می‌شود که در یک عدد صحیح (observation) به صورت زیر کدگذاری می‌شود.

$$observation = ((taxi_{row} \times 5 + taxi_{col}) \times 5 + passenger) \times 4 + destination$$

موقعیت‌هایی که مسافر می‌تواند در آن‌ها حضور داشته باشد عبارتند از:

(۱) عدد صفر: خانه قرمز

(۲) عدد یک: خانه سبز

(۳) عدد دو: خانه زرد

(۴) عدد سه: خانه آبی

(۵) عدد چهار: مسافر در تاکسی باشد

موقعیت مقصد مسافر یکی از موارد زیر می‌تواند باشد:

(۱) عدد صفر: خانه قرمز

(۲) عدد یک: خانه سبز

(۳) عدد دو: خانه زرد

(۴) عدد سه: خانه آبی

برای بدست آوردن مختصات تاکسی، موقعیت مسافر و همین‌طور مقصد او می‌توان از متد `env.decode()` استفاده کرد که خروجی آن یک `list_reverseiterator` می‌باشد که با پیمایش بر روی آن (مثلاً به کمک `for`) می‌توانید به این چهار مقدار (به ترتیب سطر تاکسی، ستون تاکسی، موقعیت مسافر و موقعیت مقصد مسافر) دست یابید.

حالات اتمام بازی در محیط:

بازی در دو حالت ممکن است به اتمام برسد:

۱- در صورتی که عامل بتواند مسافر را به مقصد رسانده و پیاده کند.

۲- در صورتی که حداکثر تعداد دورهای ممکن برای محیط (در این محیط ۲۰۰ دور) به پایان برسد.

امتیازدهی در محیط:

امتیازدهی در این محیط به صورت زیر است:

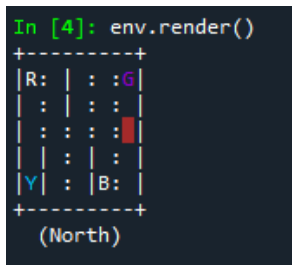
(۱) ۲۰+ به ازای پیاده کردن مسافر در مقصد

(۲) ۱۰- به ازای انجام اشتباه کنش سوار یا پیاده کردن

(۳) ۱- به ازای انجام هر کنشی که جزء موارد بالا نباشد

۳- نکات پیاده‌سازی

(۱) برای تعامل با محیط و اعمال کنش انتخاب شده باید از تابع `step` استفاده شود. در محیط تاکسی خروجی تابع `step` چهار پارامتر است: مورد اول همان `observation` می‌باشد که به کمک تابع `decode` می‌توانیم آن را به چهار مقدار مشخص کننده حالت محیط تبدیل کنیم؛ پارامتر بعدی پاداش عامل از انجام کنش است؛ پارامتر سوم `done flag` است که در صورتی که بازی به درستی تمام شده باشد برابر `True` و در غیر این صورت برابر `False` خواهد بود. خروجی چهارم نیز برابر با پارامتر `info` است که در این محیط احتمال انجام کنش را نشان می‌دهد که برابر یک است (کنش‌ها قطعی هستند).



(۲) در صورتی که بخواهید تصویر نقشه بازی و موقعیت موجودیت‌های مختلف را مشاهده کنید باید از تابع `render` استفاده کنید. در واقع خروجی تابع `render` به شکل روبه‌رو است که مستطیل قرمز رنگ نشان دهنده تاکسی (عامل هوشمند)، حرف بنفش موقعیت مسافر و حرف آبی موقعیت مقصد مسافر را نشان می‌دهد.

(۳) در صورتی که کتابخانه `OpenAI Gym` را نصب ندارید می‌توانید از دستور زیر استفاده نمایید:

```
pip install gym==۰.۲۱.۰
```

و در صورتی که از محصولات اناکوندا نیز استفاده می‌کنید می‌توانید از دستور زیر استفاده نمایید:

```
conda install -c conda-forge gym== ۰.۲۱.۰
```

راهنمایی: در این مرحله برای آموزش عامل یادگیر باید تعامل عامل با محیط چندین بار تکرار شود (هر بار شامل چندین دور) و در هر تکرار دانش یادگرفته شده (مثلاً `Q table`) ذخیره شود تا در تکرار بعدی مورد استفاده قرار گیرد. پس از آموزش، عامل هوشمند باید بتواند با بهره‌گیری از دانش یادگرفته شده مسیر بهینه برای جابجایی مسافر از مبدأ به مقصد را پیدا کند.

۴- نکات تحویل پروژه

(۱) با توجه به اهمیت یادگیری بخش‌های مقدماتی درس حتما باید برنامه خود را از ابتدا پیاده‌سازی نمایید. هر تشابه با کدهای موجود در اینترنت موجب گرفتن نمره صفر خواهد شد. در صورتی که از مقاله، سایت یا کد گیت‌هایی الهام گرفته‌اید حتما باید در بخش منابع مستندات خود آن‌ها را وارد نمایید.

(۲) در صورتی که به مشکلی در پیاده‌سازی برنامه برخوردید حتما با دستیار پروژه درس در میان گذاشته و یا در سایت کوئرا مشکل خود را اعلام نمایید. پروژه خود را حتما تا زمان مقرر در سایت کوئرا آپلود کنید.

(۳) گروه‌بندی دانشجویان در انجام این مرحله از پروژه همانند مرحله قبلی است و امکان تغییر گروه‌ها وجود ندارد.

(۴) فایل ارسالی شما باید شامل فایل پیاده‌سازی و مستندات پروژه باشد. در مستندات پروژه باید راجع به استراتژی‌های پیاده‌سازی و همین‌طور الگوریتم‌ها و روش‌هایی مورد استفاده، صحبت شده باشد. **در مستند ارسالی خود حتما به الگوریتم انتخابی خود و دلیل استفاده از آن اشاره نمایید.** در مسیر انجام پروژه هر ابتکار یا فعالیتی که منجر به عملکرد بهتر عامل شده است را توضیح دهید. در نهایت منابعی که برای پیاده‌سازی پروژه استفاده کرده‌اید را ذکر نمایید. بهتر است که برای اطمینان هرچه بیشتر فایل Q table ذخیره شده (یا هر دانش ذخیره شده دیگر) را به صورت جداگانه ارسال کنید.

(۵) راه‌های ارتباط با دستیار پروژه به صورت زیر می‌باشد:

- Telegram ID: @arshia_hemmat

- Email: arshiahemmat93@gmail.com

در صورتی که در باز کردن از هر کدام از هایپرلینک‌ها مشکل دارید، می‌توانید از این قسمت لینک‌ها را باز کنید.

https://www.gymlibrary.dev/environments/toy_text/taxi

موفق باشید