دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

هوش مصنوعی
بهار ۱۴۰۱
استاد: محمدحسین رهبان
گردآورندگان: امیرحسین جوادی ، آرمان بابایی ، ابوالفضل رحیمی

تمرین هفتم | **یادگیری تقویتی** | مهلت ارسال: ۲۷ تیر

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص‌شده است.
- در این تمرین مجاز به استفاده از تاخیر مجاز نیستید و ددلاین مشخص شده آخرین زمان تحویل تمرین می‌باشد.
- هم‌کاری و هم‌فکری شما در انجام تمرین مانعی ندارد اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد.
- در صورت هم‌فکری و یا استفاده از هر منابع خارج درسی، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفا تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.

سوالات نظری (۷۰ نمره)

۱. (۲۰ نمره)

(آ) سه مسئله نام ببرید که در چارچوب یادگیری تقویتی قرار می‌گیرند و برای هر کدام حالت‌ها، اکشن‌ها و پاداش‌های مربوطه را مشخص کنید.

(ب) آیا یادگیری تقویتی، مدل کارایی برای تمام مسائل یادگیری goal-directed ارائه می‌دهد؟ توضیح دهید. اگر جوابتان نه است مثال نقض بیاورید.

(ج) رباتی قصد دارد از یک هزارتو خارج شود. برای این هدف شما تصمیم میگیرید که زمانی که ربات از هزارتو خارج شد پاداش $+1$ و در بقیه مواقع پاداش 0 بدهیم. بعد از مدتی بعد از شروع الگوریتم یادگیری، مشاهده می‌کنیم که هیچ پیشرفتی در ربات برای فرار از هزارتو مشاهده نمی‌شود. مشکل کجاست؟ چگونه میتوان این مشکل را حل کرد؟

۲. (۵۰ نمره)

امیرحسین به دلیل فعالیت‌های غیرقانونی به یک جزیره ناشناخته تبعید شده‌است. تنها راه زنده ماندن او شکار حیوانات داخل جزیره است. میزان مطلوبیت هر حالت از زندگی را برای امیرحسین در شکل ۱ مشخص کردیم. برای مدل سازی تغییر حالات او نسبت به افعالش دست به دامان transition functionیی مانند شکل ۲ شدیم.

State	Reward
Hungry-Tired	0
Hungry-Rested	1
Full-Tired	1
Full-Rested	2

Figure 1: States and Rewards

From State	Action	Probability	Result
Hungry-Tired	Rest	1.0	Hungry-Rested
Hungry-Tired	Hunt	0.8	Hungry-Tired
Hungry-Tired	Hunt	0.2	Full-Tired
Hungry-Rested	Rest	1.0	Hungry-Rested
Hungry-Rested	Hunt	0.8	Full-Rested
Hungry-Rested	Hunt	0.2	Hungry-Tired
Full-Tired	Rest	1.0	Hungry-Rested
Full-Tired	Hunt	0.8	Hungry-Tired
Full-Tired	Hunt	0.2	Full-Tired
Full-Rested	Rest	1.0	Hungry,Rested
Full-Rested	Hunt	0.8	Full-Tired
Full-Rested	Hunt	0.2	Hungry-Tired

Figure 2: Actions and Transitions

(آ) در صورتی که سیاست همواره استراحت باشد، V^{π}(Hungry-Tired) را بر حسب γ به دست بیاورید.

(ب) با فروض زیر مقدار Q(Hungry-Rested,Hunt) را مشخص کنید. پاسخ شما باید تابعی از موارد زیر باشد.

i. V*(Full-Rested) = k
ii. Q(Hungry-Tired,Rest) = A
iii. Q(Hungry-Tired,Hunt) = B
iv. $B > A$
v. γ is between 0 and 1

(ج) فرض کنید که $\gamma = ۱$ الگوریتم Value Iteration را برای سه مرحله اجرا کنید.

iteration	V*(Hungry-Tired)	V*(Hungry-Rested)	V*(Full-Tired)	V*(Full-Rested)
0	0	0	0	0
1				
2				
3				

Figure 3: Value Iteration

(د) سیاست در مرحله سوم چیست؟

(ه) فرض کنید که دنباله‌ای از استیت‌ها و اکشن‌ها را به شکل زیر تجربه کرده‌ایم.

Act	S	A	S′	Reward
1	Hungry-Tired	Rest	Hungry-Rested	1
2	Hungry-Tired	Hunt	Hungry-Tired	0
3	Hungry-Rested	Rest	Hungry-Rested	1
4	Hungry-Rested	Hunt	Full-Rested	2
5	Full-Rested	Rest	Hungry-Rested	1
6	Hungry-Rested	Hunt	Hungry-Tired	0

Figure 4: Sequence of Actions and Results

با فروض زیر جدول ۵ را کامل کنید.

i. $Q(s, a) = 0$ for all (s, a)
ii. $\alpha = 0.5$
iii. $\gamma = 1$

Act	Q (Hungry-Tired, Hunt)	Q (Hungry-Tired, Rest)	Q (Hungry-Rested, Hunt)	Q (Full-Rested, Rest)	Q (Hungry-Rested, Rest)
0	0	0	0	0	0
1					
2					
3					
4					
5					
6					

Figure 5: Q Value Table

(و) با داشتن دنباله‌ی داده‌شده در قسمت قبل موارد زیر را مشخص کنید.

i. T(Hungry-Tired,Hunt,Hungry-Tired)
ii. T(Hungry-Rested,Hunt,Hungry-Tired)

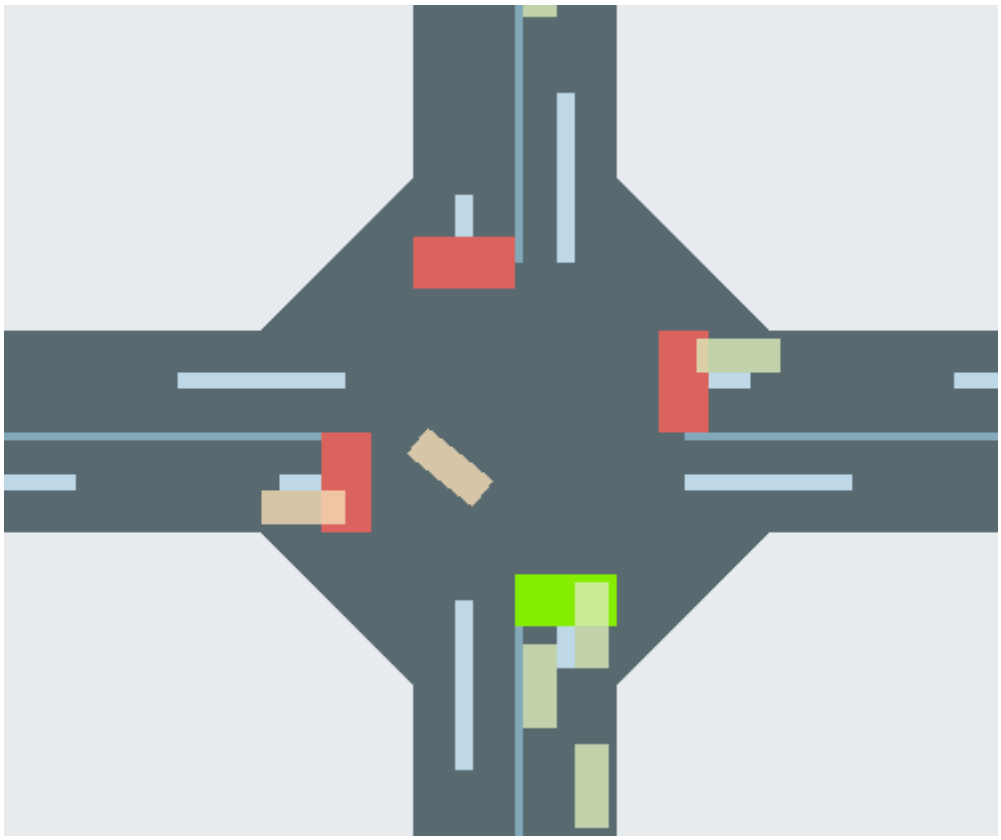
سوالات عملی (۷۰+۵۰ نمره)

۱. (۷۰ نمره) در این تمرین می‌خواهیم یک چراغ قرمز یک چهارراه را به کمک RL مدیریت کنیم. به این منظور یک شبیه‌ساز چراغ قرمز در اختیار شما قرار می‌گیرد. [۱]

چهارراهی که در این تمرین داریم یک از چهار خیابان چهاربانده (دو لاین رفت و دو لاین برگشت) تشکیل شده است. خودرویی که در لاین سمت راست مسیر قرار دارد حتما به طور مستقیم حرکت می‌کند و خودرویی که در لاین سمت چپ مسیر رفت قرار دارد حتما به سمت چپ گردش می‌کند.

چراغ قرمز در هر لحظه می‌تواند کارهای زیر را انجام دهد (چراغ‌هایی که وضعیت آن‌ها مشخص نشده قرمز می‌شوند):

(آ) تمام چراغ‌ها را به مدت ۵ ثانیه قرمز کن.
(ب) مسیرهای شرق‌ـمستقیم و غرب‌ـمستقیم را به مدت ۳۰ ثانیه سبز کن.
(ج) مسیرهای شمال‌ـمستقیم و جنوب‌ـمستقیم را به مدت ۳۰ ثانیه سبز کن.
(د) مسیرهای شرق‌ـچپ و غرب‌ـچپ را به مدت ۳۰ ثانیه سبز کن.
(ه) مسیرهای شمال‌ـچپ و جنوب‌ـچپ را به مدت ۳۰ ثانیه سبز کن.
(و) مسیرهای غرب‌ـچپ و غرب‌ـمستقیم را به مدت ۳۰ ثانیه سبز کن.
(ز) مسیرهای شرق‌ـچپ و شرق‌ـمستقیم را به مدت ۳۰ ثانیه سبز کن.
(ح) مسیرهای جنوب‌ـچپ و جنوب‌ـمستقیم را به مدت ۳۰ ثانیه سبز کن.
(ط) مسیرهای شمال‌ـچپ و شمال‌ـمستقیم را به مدت ۳۰ ثانیه سبز کن.



شکل ۶: خودرویی که در میان چهارراه قرار دارد در مسیر جنوب_چپ یا همان سومین مسیر قرار داشته و در حالی که چراغ قرمز مسیرهای جنوب_چپ و جنوب_مستقیم را سبز نگه داشته در حال حرکت است.

مثلا در شکل ۶ یک مثال از حالتی که مسیر جنوب_چپ و جنوب_مستقیم سبز هستند را نشان می‌دهد.

شبیه‌سازی که در اختیار شما قرار گرفته، بر اساس رابط کاربری Gym OpenAI طراحی شده است. توابع اصلی‌ای که برای کار با این شبیه‌ساز نیاز دارید عبارتند از:

- `reset` که شبیه‌سازی را به حالت اولیه در لحظه‌ی اول برمی‌گرداند و آرایه‌ای با هشت درایه برمی‌گرداند که هر یک، تعداد خودروی موجود در هر یک از **لاین**ها را برمی‌گرداند. تعداد خودروها در لحظه‌ی اولیه صفر است. این تعداد از مسیر سمت چپ غرب شروع و به طور پادساعت‌گرد می‌چرخد (پس درایه‌ی چهارم مربوط به مسیر جنوب_مستقیم است.)
- `step` این تابع action انتخابی شما (یک عدد در بازه‌ی $[۰, ۸]$ به ترتیب در تناظر با نه حالت ذکرشده در بالا) را ورودی می‌گیرد و پس از انجام دستور شما، نتیجه را در قالب یک آرایه‌ی هشت‌تایی مشابه `reset` برمی‌گرداند. علاوه بر این قرینه‌ی مجموع زمان توقف خودروها در چهارراه به عنوان پاداش، به اتمام رسیدن شبیه‌سازی (یک مقدار boolean که با اتمام شبیه‌سازی `True` می‌شود و نیاز است پس از آن تابع `reset` را برای برگشت به حالت اولیه صدا کنید.) و یک دیکشنری خالی برای انطباق با قواعد Gym OpenAI برمی‌گرداند.

پرونده‌ی `test.py` یک پاسخ تصادفی برای این سوال تولید می‌کند.

برای نصب موارد موردنیاز پیشنهاد می‌شود اقدامات زیر را انجام دهید:

(آ) با دستور `python3.8 -m venv ./penv/` یک محیط مجازی برای این تمرین ایجاد کنید.

[۱] این شبیه‌ساز مطابق این پروژه است.

(ب) با دستور source ./penv/bin/activate محیط مجازی خود را فعال کنید.

(ج) با دستور git clone https://github.com/cityflow-project/CityFlow.git کتابخانه‌ی CityFlow را که برای کار با ترافیک شهری طراحی شده‌است را دانلود کنید.

(د) با ورود به پوشه‌ی CityFlow دستور . pip install را اجرا کنید تا پروژه‌ی CityFlow نصب شود. (توجه کنید که محیط مجازی فعال باشد.)

(ه) با ورود به پوشه‌ی محتوایی که در اختیارتان قرار گرفته و هم‌سطح با پرونده‌ی setup.py از دستور pip install . استفاده کنید تا شبیه‌ساز نصب شود.

(و) به پوشه‌ی gym_cityflow/envs/1x1_config بروید و از این پوشه می‌توانید پرونده‌ی راه‌حل خود را اجرا کنید.

برای مشاهده‌ی عملکرد راه‌حل خود، پس از اجرای آن، پرونده‌ی CityFlow/frontend/index.html را در مرورگر خود باز کنید. در ادامه پرونده‌های roadnetlog.json و replay.txt را به ترتیب به عنوان Roadnet File و Replay File در پنل سمت چپ ورودی دهید و کلید Start را فشار دهید.

برای این تمرین **یک پرونده‌ی** راه‌حل خود را در کنار نموداری از عملکرد آن در episode های مختلف زیپ کرده و در کوئرا بارگذاری کنید.

۰. (سوال امتیازی ۵۰ نمره)
برای پاسخ به این سوال به پرونده‌ی ژوپیتر $DDQN.ipynb$ مراجعه کنید. در نهایت پس از تکمیل این پرونده آن را روی کوئرا بارگذاری کنید.