

بخش تحلیلی

سؤال اول

Grid World زیر را در نظر بگیرید، عامل می‌تواند به چپ، راست، بالا یا پایین حرکت کند. حرکت روی یک مربع سبز، پاداش ۱- را به همراه دارد، در حالی که حرکت روی یک مربع قرمز، پاداش ۱۰۰۰- را به دنبال دارد و اپیزود را پایان می‌دهد و عامل به حالت شروع باز می‌گردد. اپیزود همچنین در صورتی که عامل به حالت ۷، حالت هدف، برسد، خاتمه می‌یابد. عامل در حالت ۱ (حالت هدف) شروع می‌کند. فرض کنید ما دو الگوریتم، Q-learning و SARSA را با ϵ ثابت برابر ۰.۱ و $\gamma = 1$ آموزش می‌دهیم و دو سیاست را یاد می‌گیریم: سیاست A و سیاست B.

سیاست A مسیر،

۷-۸-۲۱-۲۰-۱۹-۱۸-۱۷-۱۶-۱۵-۱۴-۱

را در طی آموزش یاد می‌گیرد، همچنین سیاست B مسیر،

۷-۸-۲۱-۲۲-۳۵-۳۴-۳۳-۳۲-۳۱-۳۰-۲۹-۲۸-۱۵-۱۴-۱

را در طی آموزش یاد می‌گیرد. کدام سیاست توسط Q-learning و کدام سیاست توسط SARSA یاد گرفته می‌شود، همچنین آیا می‌توانید بگویید کدام الگوریتم در این مسئله بهتر عمل می‌کند و چرا؟ پاسخ خود را به صورت تحلیلی بیان کنید نیازی به حل دستی مسئله نیست.

۲۹	۳۰	۳۱	۳۲	۳۳	۳۴	۳۵
۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲
۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱
۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸
۱	۲	۳	۴	۵	۶	۷

سؤال دوم

محیط شبکه‌ای زیر را در نظر بگیرید. عامل می‌تواند از هر خانه سفید شروع کند، می‌تواند به بالا، پایین، چپ یا راست حرکت کند. اقدامات قطعی هستند (مثلاً رفتن به چپ از حالت ۱۶ به حالت ۱۵ می‌رود مگر

اینکه عامل به دیوار برخورد کند). لبه‌های ضخیم‌تر دیوارها را نشان می‌دهند، و عامل در برخورد با دیوارها در همان خانه باقی خواهد ماند. عامل با انجام هر عمل در خانه هدف، سبز رنگ، (شماره ۱۲) پاداشی به ارزش r_g به‌دست می‌آورد و اپیزود را به پایان می‌رساند. همچنین انجام هر عمل در خانه قرمز (شماره ۵) پاداشی به ارزش r_r به‌دست می‌آورد و اپیزود را به پایان می‌رساند. انجام هر عملی در سایر خانه‌ها پاداشی معادل $r_s \in \{-1, 0, +1\}$ به همراه دارد (حتی اگر عمل باعث شود عامل در همان خانه باقی بماند). فرض کنید که $\gamma = 1, r_g = +5, r_r = -5$ به هر بخش از این سؤال به‌صورت کوتاه پاسخ دهید.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

الف) مقدار r_s همانطور که گفته شد می‌تواند از مجموعه مقابل باشد $r_s \in \{-1, 0, +1\}$ حال r_s را به‌گونه‌ای تعریف کنید که باعث شود سیاست بهینه، کوتاه‌ترین مسیر به مربع هدف که سبز می‌باشد (شماره ۱۲) را برگرداند. با استفاده از این r_s ، برای هر مربع مقدار بهینه ارزش را بیابید.

ب) فرض کنید حالا ما یک محیط شبکه‌ای دوم را با اضافه کردن یک ثابت c به همه پاداش‌ها به‌گونه‌ای که $r_s = +2$ درست کرده‌ایم (مقدار قبلی r_s را برابر با مقدار پیشنهادی خود در بخش الف در نظر بگیرید). سیاست بهینه چگونه تغییر می‌کند، مقادیر ارزش مربع‌های سفید برابر چند می‌شوند؟ (امتیازی)

ج) حالا محیط شبکه‌ای دوم را از بخش (ب) در نظر بگیرید و γ را طوری تغییر دهید که $0 < \gamma < 1$ باشد. آیا سیاست بهینه تغییر می‌کند؟ آیا این بستگی به انتخاب شما از گاما دارد؟ (راهنمایی: گاما را مقادیر خیلی نزدیک به یک و یا خیلی نزدیک به صفر در نظر بگیرید).

د) یک MDP عمومی را در نظر بگیرید. برای این حالت فرض کنید که horizon نامحدود است. یک سیاست π در این MDP یک تابع ارزش V_π ایجاد می‌کند (به‌عنوان V_{old}^π در نظر بگیرید). حال فرض کنید ما یک MDP جدید داریم که تنها تفاوت آن این است که به همه پاداش‌ها یک ثابت c اضافه شده‌است. آیا می‌توانید یک عبارت برای تابع ارزش جدید V_{new}^π که توسط π در این MDP دوم ایجاد شده‌است، بر حسب c, V_{old}^π و γ ارائه دهید؟ (امتیازی)

سؤال سوم

الگوریتم‌های Sarsa و Expected-Sarsa در یک محیط گسسته از منظر میزان پشیمانی،

- در یافتن سیاست e-optimal چه تفاوتی دارند؟
- در یافتن سیاست بهینه چه تفاوتی دارند؟

بخش پیاده‌سازی

توضیح مسئله

شرکتی در مزایده‌ای برنده شده که در آن قرار است [drone‌هایی](#)¹ را طراحی کند که در مواقع آتش‌سوزی در مناطق جنگلی، از یک موقعیت شروع به پرواز کرده و به یک موقعیت هدف که در آن درختان آتش گرفته‌اند برسد، و به اطفاء حریق بپردازد. شما به‌عنوان یک متخصص هوش مصنوعی در این شرکت وظیفه دارید که قسمت نرم‌افزاری مربوط به مسیریابی این drone‌ها را در یک محیط Grid World شبیه‌سازی کرده و از الگوریتم‌های یادگیری تعاملی برای هدایت drone‌ها برای رسیدن به موقعیت هدف بهره‌مند شوید.

نحوه پیاده‌سازی محیط

محیط آزمایشی را یک جدول ۶ در ۶ در نظر بگیرید. در این محیط، drone از یک موقعیت تصادفی شروع کرده و سعی می‌کند با عبور از میان درختان به موقعیت هدف برسد. drone نباید به درختان برخورد کند، و اگر برخوردی هم صورت گرفت، drone در مکان فعلی خود باقی می‌ماند. هر حالت در این محیط یک‌خانه (یا سلول) از جدول می‌باشد که با یک پوزیشن دکارتی $((x, y))$ مشخص می‌شود. انتخاب و تحلیل این‌که موارد دیگری به مجموعه حالت‌های عامل اضافه شود بر عهده شماست. در تصویر ۱ موقعیت هر کدام از خانه‌ها قابل مشاهده است.

¹ هواپیمای بدون خلبان

(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)
(5, 0)	(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)

تصویر ۱. موقعیت دکارتی خانه‌های محیط

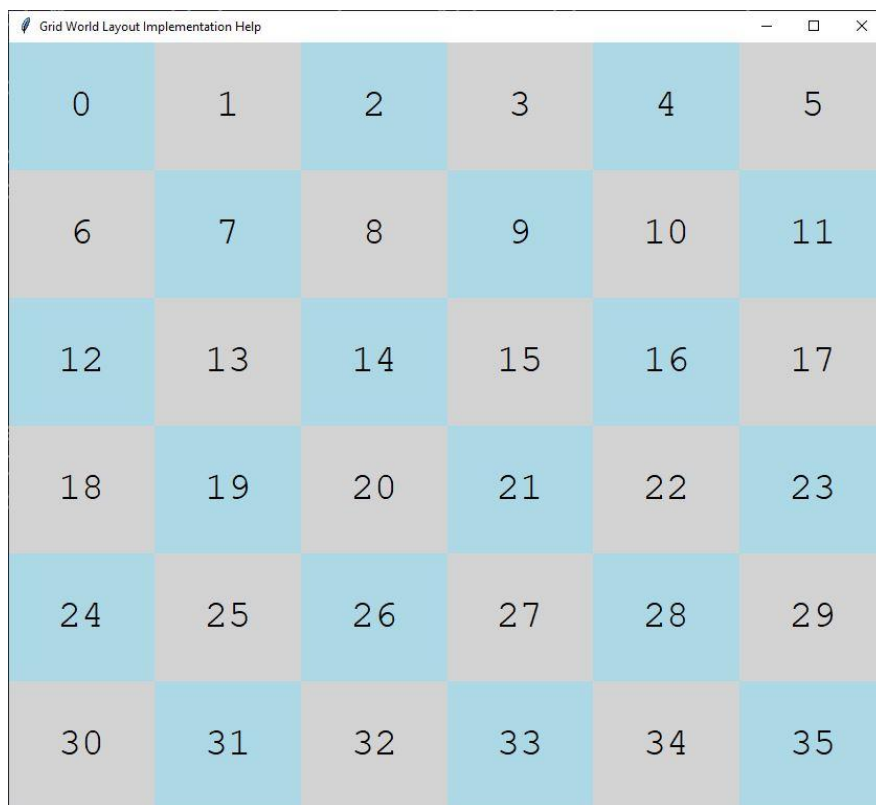
drone (عامل) در هر زمان می‌تواند از بین چهار حرکت پایین (۰)، بالا (۱)، راست (۲) و چپ (۳) یکی را انتخاب کند. در حالت‌های مرزی، در صورت انتخاب حرکت غیرمجاز، عامل در جای خود باقی می‌ماند. به‌علت وزش باد شدید، عمل انتخاب شده با یک احتمالی انجام می‌شود. به‌عبارت دیگر، در ۸۰٪ مواقع، عملی که عامل انتخاب می‌کند انجام می‌شود، در ۱۰٪ مواقع، عامل در جای خود باقی می‌ماند و در ۱۰٪ مواقع نیز عامل در جهت عکس حرکت می‌کند.

علاوه‌بر عامل و هدف، ۸ مانع (درختان) وجود دارد. این سه مفهوم به‌صورت تصادفی در محیط قرار می‌گیرند و نباید از منظر پوزیشن با یکدیگر overlap داشته باشند. تنها دو مانع می‌توانند در مرزها قرار بگیرند؛ دو مانع بیشتر نباید در کنار هم قرار بگیرند؛ مانع‌ها به‌نحوی روی جدول چیده شوند که بن‌بست ایجاد نشود و عامل بتواند مسیری مناسب برای رسیدن به هدف داشته باشد. شماره خانه‌ها در تصویر ۲ نشان داده شده است.

برای موقعیت‌های عامل و هدف از مجموعه‌های زیر استفاده کنید. برای یکی از عامل و هدف شماره‌ای را به‌صورت تصادفی از یک مجموعه به‌دلخواه انتخاب کرده، و برای دیگری از مجموعه دیگر به‌صورت تصادفی انتخاب کنید. موقعیت‌های عامل، هدف و مانع‌ها برای همه‌ی سوالات مربوط به پیاده‌سازی الگوریتم‌ها یکسان خواهند بود.

مجموعه خانه‌های کاندید اول برای هدف و یا عامل: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35\}$

مجموعه خانه‌های کاندید اول برای هدف و یا عامل: $\{23, 28, 29, 33, 34, 35\}$



0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

تصویر ۲. شماره خانه‌ها در محیط

در تابع پاداشی که تعریف خواهید کرد، به‌ازای هر مورد، پاداش موردنظر بایستی با میانگین و واریانس مناسب از توزیع نرمال نمونه‌برداری شود. این موارد به‌شرح زیر می‌باشد:

- پاداش منفی به‌ازای برخورد با مانع‌ها: میانگین 1- و واریانس 0.5
- پاداش منفی به‌ازای هر انتقال از یک‌خانه به خانه‌ی دیگر: میانگین 0.5- و واریانس 0.25
- پاداش مثبت به‌ازای رسیدن به خانه هدف: میانگین 25 و واریانس 5

علاوه‌بر مسائل فوق، عامل حاوی دو خصیصه درونی نیز می‌باشد: میزان شارژ باتری آن، و میزان سلامت فیزیکی آن.

عامل در ابتدای هر اپیزود از شارژ باتری ۱۰۰٪ شروع کرده و به‌ازای هر انتقال، مقداری از شارژ باتری‌اش کاسته می‌شود که توزیع نرمال با میانگین 0.35 و واریانس 0.15 دارد. اگر قبل از رسیدن به هدف، میزان شارژ باتری عامل کمتر از ۵٪ شود عامل باید به پایگاه خود جهت شارژ باتری مراجعه نماید. در مورد میزان سلامت فیزیکی، عامل از مقدار ۱۰۰٪ سلامتی کامل شروع کرده و به‌ازای هر برخورد با مانع، مقداری از سلامتی‌اش کم می‌شود که توزیع نرمال با میانگین 0.2 و واریانس 0.1 دارد. اگر میزان سلامتی عامل کمتر از ۱۵٪ شود عامل باید به پایگاه خود برگردد.

نحوه پیاده‌سازی محیط و الگوریتم‌ها همانند API استفاده شده در محیط‌های [پکیج Gymnasium](#) می‌باشد. می‌توانید برای آشنایی در مورد نحوه استفاده از محیط‌های این پکیج به این [لینک](#)، و برای آشنایی با نحوه ساخت محیط به کمک پکیج Gymnasium به این [لینک](#) مراجعه بفرمایید.

نکات پیاده‌سازی

برای پیاده‌سازی محیط و سؤالات آینده به این نکات توجه بفرمایید:

- برای پیاده‌سازی باید و حتماً از آخرین نسخه پکیج Gymnasium بهره ببرید.
- برای این‌که تنظیمات طراحی و عملکردی محیط، تولید مقادیر تصادفی و نتایج به‌دست آمده reproducible باشد، بایستی متغیر SEED به‌صورت جهانی و به‌مقدار سه رقم آخر شماره دانشجویی‌تان تعریف کنید. سعی کنید مقادیر را برای همه‌ی این موارد به‌صورت تصادفی با کمک متغیر SEED تولید کنید و از مقادیر hardcoded بپرهیزید.
- سیاست مورد استفاده برای عامل را epsilon-greedy در نظر بگیرید.
- در تمامی سؤالات به‌جز ذکر صریح در صورت سؤال مقدار اپسیلون را به‌صورت کاهشی مناسب و مقدار discount factor را 0.9 در نظر بگیرید. همچنین مقدار نرخ یادگیری را برابر با 0.1 در نظر بگیرید.
- برای تمامی روش‌های زیر مسئله را حداقل ۱۰ بار تکرار به‌اندازه‌ی حداقل ۷۵۰ اپیزود انجام دهید. پاداش دریافتی در طول یادگیری و سرعت همگرایی به سیاست بهینه (تعداد گام‌های برداشته شده و یا به‌قولی regret) را با رسم نمودارهای مناسب و همچنین گزارش مقدارشان، تحلیل و بررسی کنید.



- در پایان هر یک از سؤال‌های این بخش یکی از عامل‌ها (بهترین عامل و یا میانگین عامل‌ها) را پس از آموزش به‌اندازه‌ی ۱۰ اپیزود تست و مقادیر خواسته شده در بالا را گزارش کنید. همچنین، با استفاده از دستور `render`، رفتار عامل را در محیط نمایش دهید.
- در [لینکی](#) که جهت راهنمایی شما برای پیاده‌سازی محیط ارائه شده بود، نیازی پیاده‌سازی و استفاده از قسمت‌های “Creating Env” ، “Creating a Package” ، “Creating” “Environment Instances” ، “Using Wrappers” نیست. بعد از اتمام پیاده‌سازی کلاس محیط‌تان، صرفاً یک آبجکتی از آن کلاس ایجاد کنید.

تذکر: دقت شود که پارامترهای داده شده در کدهای مان برای الگوریتم‌ها و مدل‌ها صرفاً به‌عنوان یک گزینه‌ی اولیه بوده و ممکن است پارامترها را بتوان به‌طوری تنظیم کرد که یادگیری بهتر شود. در صورتی که در صورت سؤال به‌صورت صریح قید نشده باشد، می‌توانید این پارامترها را تغییر دهید.

سؤالات پیاده‌سازی

۱. الگوریتم `q-learning` را یک‌بار به‌ازای نرخ یادگیری 0.1 و بار دیگر به‌ازای نرخ یادگیری کاهشی پیاده‌سازی نمایید و نتایج به‌دست آمده را از منظر میزان حسرت (سرعت همگرایی و مقدار همگرا شده) در طول یادگیری با یک‌دیگر مقایسه کنید. روش انتخابی خود را برای کاهش مقدار اپسیلون در طول فرآیند یادگیری شرح دهید.

۲. الگوریتم‌های `Sarsa`، `Tree Backup n-Step` را به‌ازای سه مقدار `n` پیاده‌سازی کنید و نتایج به‌دست آمده را از منظر میزان حسرت (سرعت همگرایی و مقدار همگرا شده) با یک‌دیگر مقایسه کنید و در تحلیل نتایج، علت عملکرد بهتر به‌ازای یک مقدار `n` مشخص را تحلیل نمایید.

۳. با توجه به شماره‌ی دانشجویی خود به سؤال‌های زیر پاسخ دهید.

اگر رقم آخر شماره دانشجویی شما زوج است:

۳.۱. با استفاده از روش `on-Policy MC` مسئله را حل کنید و موارد خواسته شده را یک‌بار برای اپسیلون کاهشی و همچنین برای اپسیلون 0.1 انجام دهید و نتایج به‌دست آمده را از منظر میزان حسرت (سرعت همگرایی و مقدار همگرا شده) با یک‌دیگر مقایسه کنید.

۳.۲. الگوریتم Policy Iteration را با توجه به مواردی که از هندز آن ۲ آموخته‌اید برای محیط موردنظر پیاده‌سازی کنید. در مورد نحوه مقداردهی به ابرپارامترها^۱ بحث کنید و نتایج به‌دست آمده را تحلیل نمایید.

۳.۳. با توجه به اینکه محیط مسئله تصادفی می‌باشد، سعی کنید که با استفاده از الگوریتم MC محیط مسئله را یاد بگیرید و سپس بر روی محیط یادگیری شده الگوریتم PI را اجرا نمایید و نتایج این بخش را با بخش قبل مقایسه کنید و تحلیل خود را بیان نمایید.

اگر رقم آخر شماره دانشجویی شما فرد است:

۳.۴. با استفاده از روش off-Policy MC خواسته‌های مسئله را پاسخ دهید و موارد خواسته شده را یک‌بار برای اپسیلون کاهشی و همچنین برای اپسیلون 0.1 انجام دهید و نتایج به‌دست آمده را از منظر میزان حسرت (سرعت همگرایی و مقدار همگرا شده) با یک‌دیگر مقایسه کنید. توجه داشته‌باشید که سیاست رفتاری را یک سیاست epsilon-greedy در نظر گرفته و در هر مرحله آن را بر اساس آخرین مقدار Q-value ها به‌روزرسانی کنید.

۳.۵. الگوریتم Value Iteration را با توجه به مواردی که از هندز آن ۲ آموخته‌اید برای محیط موردنظر پیاده‌سازی کنید. در مورد نحوه مقداردهی به ابرپارامترها بحث کنید و نتایج به‌دست آمده را تحلیل نمایید.

۳.۶. با توجه به اینکه محیط مسئله تصادفی می‌باشد، سعی کنید که با استفاده از الگوریتم MC محیط مسئله را یاد بگیرید و سپس بر روی محیط یادگیری شده الگوریتم VI را اجرا نمایید و نتایج این بخش را با بخش قبل مقایسه کنید و تحلیل خود را بیان نمایید.

۴. الگوریتم‌های پیاده‌سازی شده در سؤالات قبل را از نظر سرعت یادگیری، میزان حسرت و پایدار بودن در طول یادگیری (بررسی نوسان در طول یادگیری) با هم مقایسه نمایید و تحلیل خود را بیان نمایید.

۵. به‌نظر شما این که محیط طراحی شده نسبت به عمل انتخاب شده توسط عامل به‌صورت قطعی اقدام نمی‌کند و حاوی stochasticity می‌باشد، چه تأثیری می‌تواند روی همگرایی الگوریتم‌ها و مدل‌هایی که پیاده‌سازی کرده‌اید داشته باشد؟

^۱ Hyperparameters



نکات تمرین

- استفاده از LLM ها در این تمرین مشکلی ندارد. اما در صورت استفاده لطفاً منبع و prompt خود را ذکر نمایید تا تقلب محسوب نشود.
- مهلت ارسال این تمرین تا پایان روز یکشنبه ۳ دی ماه خواهد بود.
- انجام این تمرین به صورت یک نفره می باشد. اما بحث و گفت و گو در دیسکورد مانعی ندارد.
- در رسم نمودارها حتماً باید axis label، title و grid داشته باشد و مقادیر به صورت گویا نمایش داده شود.
- سعی کنید از پاسخ های روشن در گزارش خود استفاده کنید و اگر پیش فرضی در حل سؤال در ذهن خود دارید، حتماً در گزارش خود آن را ذکر نمایید.
- لطفاً گزارش و کد تمرین را در فایل هایی که از طریق google Doc و google colab با شما به اشتراک گذاشته شده است، وارد نمایید.
- در صورت وجود سؤال و یا ابهام می توانید در channel مربوط به این تمرین با دانشجویان دیگر مطرح نمایید و یا برای ارتباط با دستیاران آموزشی از طریق ایجاد یک thread در همان channel دیسکورد، سؤال خود را مطرح نمایید.