هوش مصنوعی **تمرین شماره‌ی 6**

مدرسین: **دکتر فدایی و دکتر یعقوب‌زاده**

طراحان: امیرعلی رحیمی، شهریار عطار، فربد عظیم محسنی

مهلت تحویل: پنجشنبه ۲۴ خرداد ۱۴۰۳، ساعت ۲۳:۵۹

بخش کتبی

MDP

سوال اول

یک جدول داریم که هر خانه در آن با شماره ردیف و ستون شناخته می‌شوند (ابتدا ردیف). Agent همیشه از حالت (1,1) که با حرف S مشخص شده شروع می‌کند. دو حالت هدف نهایی وجود دارد، (2,3) با پاداش 5+ و (1,3) با پاداش 5-. پاداش‌ها در حالت‌های غیر نهایی صفر می‌باشد. تابع Transition به گونه‌ای است که حرکت مورد نظر Agent (شمال، جنوب، غرب یا شرق) با احتمال 0.8 اتفاق می‌افتد. با احتمال 0.1 به هر یک از حالت‌های عمود بر جهت مورد نظر می‌رسد. اگر برخوردی با دیوار رخ دهد، Agent در همان حالت باقی می‌ماند.

		+5
S		-5

1) نتایج دو دور اول Value Iteration را با مقدار تخفیف[1] 0.9 محاسبه کنید. توجه کنید که خانه‌های (3, 1) و (3, 2) دارای Value ثابت می‌باشند. راهنمایی:

$$V_{i+1}(s) = \max_a \left(\sum_{s'} T(s,a,s')(R(s,a,s') + \gamma V_i(s')) \right)$$

[1] Discount Factor

S	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
V_0	0	0	5	0	0	-5
V_1	0	3.6	5	0	0	-5
V_2	2.592	3.924	5	0	2.142	-5

2) Policy را با توجه به جدول ارزش‌های بالا محاسبه کنید (برای خانه‌هایی که دو یا چند Action با مقدار برابر وجود دارد می‌توانید هر کدام را به دلخواه انتخاب کنید).

S	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
π*(S)	right	right	-	right	up	-

3) حال فرض کنید که تابع Transition را ندارید، اکنون برای اینکه بتوانید Policy بهینه را بدست آورید باید از روش‌هایی مانند Q-Learning و یا Monte Carlo استفاده کنید. Monte Carlo یک روش Model-Free می‌باشد، درباره این روش جستجو کنید. حال فرض کنید agent ما Policy-ای که همیشه به سمت راست برود را انتخاب می‌کند و سه آزمایش زیر را اجرا می‌کند، تخمین‌های Monte Carlo (کاربرد مستقیم) برای خانه‌های (1,1) و (2,2) با توجه به این مسیرها چیست؟ (reward-ها را بر اساس جدول اولیه در نظر بگیرید)

I) (1,1)–(1,2)–(1,3)
II) (1,1)–(1,2)–(2,2)–(2,3)
III) (1,1)–(2,1)–(2,2)–(2,3)

4) اگر فرض کنیم Agent بر اساس TD-Learning[2] یاد می‌گیرد، با استفاده از نرخ یادگیری 0.1 و با فرض مقادیر اولیه صفر (به جز خانه‌های نهایی)، بعد از 2 مرحله Iteration، برای هر خانه چه Value-ای داریم؟ (امتیازی)

DQN

سوال اول

درباره الگوریتم و کاربردهای DQN[3] تحقیق کنید و مطالبی که متوجه شدید را به طور خلاصه توضیح دهید.

[2] Temporal Difference Learning
[3] Deep Q-Network

بخش عملی

مقدمه

یادگیری تقویتی یکی از شاخه‌های یادگیری ماشین است و با توجه به گستردگی آن، در زمینه‌های گوناگونی مانند نظریه بازی‌ها[4]، نظریه کنترل[5]، سامانه‌های چند عامله[6]، نظریه اطلاعات[7] و غیره استفاده می‌شود. در یادگیری تقویتی، عامل هوشمند با جستجو و اکتشاف در محیط قابل تعامل مورد نظر، در ابتدا باید دانش و تجربه‌ای را از محیط خود جمع آوری کند. سپس، بر اساس دانش کسب شده، باید عمل‌ها و رفتارهایی را در آن محیط انجام دهد تا مجموع پاداشی که از آن محیط می‌گیرد بیشینه شود.

توضیح مسئله

مسئله اول: Snake (نوستالژی)

شرک برای نجات فیونا راهی سفر شد تا پرنسس را از دست اژدها نجات دهد! خر شرک که پس از جدایی از اژدها دوست نداشت دوباره او را ببیند، ترجیح داد در باتلاق منتظر شرک بماند تا او برگردد و با هم دوش گِل بگیرند. او برای اینکه حوصله اش سر نرود خواست در گوشی خود بازی کند اما از آن‌جا که همیشه با اژدها با هم بازی می‌کردند، تنها بازی‌های دو نفره در گوشی خود داشت؛ او از شما کمک خواسته که یک حریف مصنوعی برای او طراحی کنید که جای خالی اژدها را برای او پر کند.

بازی‌ای که خر شرک دوست دارد بازی کند بازی قدیمی Snake می‌باشد. در نسخه دو نفره آن هدف این است که با خوردن سیب‌ها بزرگ شوید و حریف را شکست دهید. بازی در حالتی تمام می‌شود که کله یکی از مارها به بدن مار دیگر برخورد کند (در این حالت ماری که کله‌اش به بدن مار دیگر برخورد کرده می‌بازد، در این حالت توجه کنید که اگر سر مار به بدن خودش نیز برخورد کند نیز مار می‌بازد) و یا کله دو مار با یکدیگر برخورد کند که در این حالت ماری که طول بیشتری داشته باشد می‌برد، در حالتی که کله دو مار به یکدیگر برخورد کند و طول برابر داشته باشند نیز هیچکدام از مارها برنده نمی‌شود. صفحه بازی یک جدول 20 در 20 می‌باشد و حالت ابتدایی بازی به این شکل می‌باشد که هر دو مار به صورت تصادفی در نقطه ای از جدول ظاهر می‌شوند.

قوانین بازی کمی متفاوت از قوانین بازی اصلی می‌باشد، مثلا مار نمی‌تواند از صفحه خارج شده و در صورتی که تلاش کند خارج شود بازنده می‌شود. مانند بازی اصلی به طور تصادفی سیب‌هایی در زمین بازی ظاهر می‌شوند و باعث افزایش طول مار می‌شوند. اگر در شرایطی برای بازی به حالت‌های خاص که در صورت گفته نشده بود برخوردید، به دلخواه می‌توانید در مورد آن‌ها تصمیم بگیرید (توجه کنید که هدف این تمرین تمرکز بر روی نوشتن بازی نیست).

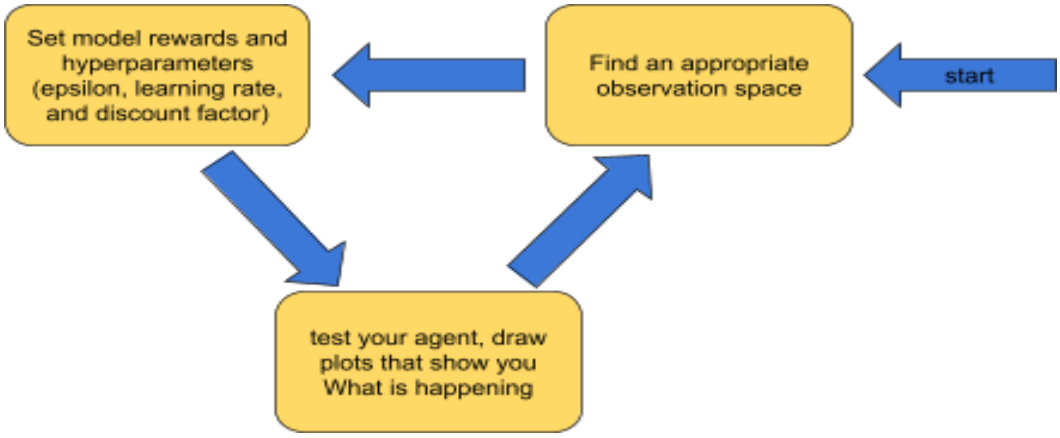
---

[4] Game Theory
[5] Control Theory
[6] Multi-Agent Systems
[7] Information Theory

شما باید با استفاده از روش Qlearning که در درس یاد گرفته اید agent خود را آموزش دهید و با توجه به شرایط گفته شده باید تلاش کنید که حریف قوی برای خر شرک آموزش دهید که جای خالی اژدها را احساس نکند و حوصله‌اش تا زمانی که شرک برگردد سر نرود. توجه کنید که با توجه به اینکه Observation Space بزرگ می‌باشد باید سعی کنید با روش‌هایی این مشکل را رفع کنید. برای راهنمایی می‌توانید به این لینک مراجعه کنید.

به طور کلی یکی از مهم‌ترین روش‌ها در این حالت تعریف یک سری ویژگی است که محیط را توصیف کند به طور مثال فرض کنید وظیفه شما برنامه‌ریزی N کار و یافتن بهترین آن‌هاست، اگر N بزرگ‌تر از 10^3 باشد Observation Space شما دارای حدود 10^3 گره[8] و 10^6 یال[9] است و Agent شما باید با محاسبه تابع هزینه[10] پس از پیمایش[11] کامل درخت راه‌حل بهینه را پیدا کند که در نتیجه استفاده از روش‌های معمول و یا حتی با استفاده از بهینه‌سازی‌هایی مانند ACO[12] زمان بسیار زیادی طول خواهد کشید. به عنوان راه‌حل شما می‌توانید یک مجموعه X تعریف کنید و تعدادی از ویژگی‌های محیط را استخراج کنید و با کمک آن به آموزش مدل بپردازید.

برای این سوال روش‌های مختلفی وجود دارد که می‌توانید این کار را انجام دهید، به طور مثال می‌توانید Observation Space را فقط شامل مختصات سیب و کله مار حریف در نظر بگیرید و یا از فاصله اقلیدسی[13] یا فاصله منهتن[14] یا دیگر روش‌ها استفاده کنید. یک راه‌حل ساده‌تر هم می‌تواند این باشد که فضای اطراف را محدود کنید، به طور مثال خانه‌هایی که در دو حرکت بعد ممکن است کله مار در آن باشد (که ده خانه می‌شود) یا حالت‌های دیگر در نظر بگیرید و Observation Space را کاهش دهید. نمای کلی کاری که در این بخش باید انجام دهید به شکل زیر است:



همچنین پیشنهاد می شود کار های زیر را برای بهتر train شدن agent انجام دهید:

- استفاده از Epsilon Decay

[8] Node
[9] Edge
[10] Loss Function
[11] Traverse
[12] Ant Colony Optimization
[13] Euclidean Distance
[14] Manhattan Distance

- کوچک تر کردن Observation Space تا جای ممکن

مار خود را با تعداد مختلفی Iteration و Hyperparameter-های مختلف امتحان کنید و مدل‌های آموزش داده شده را در فایلی ذخیره کنید (سه مدل مختلف کافی است). همچنین نمودار Reward کسب شده توسط مدل را به ازای هر Episode نشان دهید (نمودار تنها یکی از این 3 مدل کافی است).

مسابقه Snake (امتیازی)

حال می‌خواهیم عملکرد مدل‌های مار آموزش داده شده را با همدیگر مقایسه کنیم. برای این مرحله شما کد `snake.py` را به همراه `qtable.npy` در محل آپلود به صورت AI_CA6_Contest_SID.zip آپلود می‌کنید دقت کنید که پس از extract کردن باید این دو فایل در همانجا باشد و پوشه خارجی را zip نکنید. همچنین دقت داشته باشید که نام‌گذاری فایل‌ها را به درستی انجام دهید و برای توابعی که در `main.py` تعریف شده‌اند، پارامترهای ورودی و خروجی را تغییر ندهید، در صورت تغییر در مسابقات شرکت داده نمی‌شوید. بقیه کد را می‌توانید به دلخواه خود تغییر دهید، البته توجه داشته باشید که برای کاهش پیچیدگی کدی که در اختیارتان قرار گرفته نتیجه مسابقه داخل کلاس مار مشخص می‌شود، ولی در مسابقه نتیجه در خارج این کلاس محاسبه می‌شود و در صورتی که نتیجه گزارش شده توسط کلاس شما با نتیجه اصلی تطابق نداشته باشد (و یا هر کار دیگری مانند تغییر ناگهانی مختصات مار به جایی که مار حریف به بدن شما برخورد کند و ...) کار شما به عنوان تقلب حساب شده و از مسابقات حذف می‌شوید. مسابقات به صورت لیگ حذفی برگذار می‌شود و جدول به صورت تصادفی ساخته می‌شود. هر مسابقه به صورت 51 از 101 برگزار می‌شود یعنی کسی که حداقل 51 بازی از 101 بازی (در صورت تساوی بازی تکرار می‌شود) را ببرد مسابقه را می‌برد. به نفرات اول تا سوم به ترتیب 0.3، 0.2 و 0.1 نمره امتیازی تعلق می‌گیرد. همچنین دستیاران آموزشی یک مدل متوسط آموزش می‌دهند که اگر بتوانید آن را شکست دهید 0.1 نمره امتیازی به شما (جدا از نمره امتیازی رتبه و رتبه کسب شده در مسابقات) تعلق می‌گیرد.

نکات پایانی

- حجم توضیحات گزارش شما هیچ گونه تاثیری در نمره نخواهد داشت و تحلیل و نمودارهای شما بیشترین ارزش را دارد.
- سعی کنید از پاسخ‌های روشن در گزارش خود استفاده کنید و اگر پیش‌فرضی در حل سوال در ذهن خود دارید، حتما در گزارش خود آن را ذکر نمایید.
- توجه کنید این تمرین باید به صورت تک‌نفره انجام شود و پاسخ‌های ارائه شده باید نتیجه فعالیت فرد نویسنده باشد. در صورت مشاهده تقلب به همه افراد مشارکت‌کننده، نمره تمرین 100- و به استاد نیز گزارش می‌گردد. همچنین نوشته نشدن کدها توسط هوش مصنوعی نیز بررسی می‌شود!
- لطفاً گزارش، فایل کدها و سایر ضمائم مورد نیاز را با فرمت زیر در سامانه ایلرن بارگذاری نمایید.

AI_CA6_[Std number].zip