هوش مصنوعی
تمرین کامپیوتری شماره‌ی ۲

مدرس: **دکتر فدایی**
طراحان: محمد سعادتی، مجید فریدفر، پویا صادقی

مهلت تحویل: سه‌شنبه 16 آبان ۱۴۰۲، ساعت ۲۳:۵۹

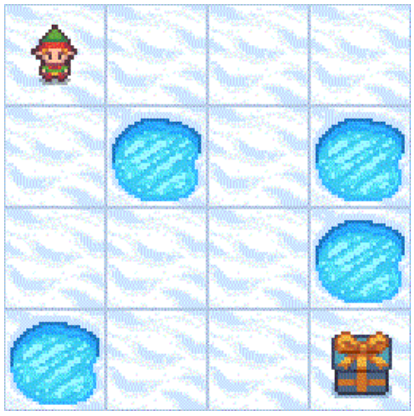
مقدمه

یادگیری تقویتی یکی از شاخه‌های یادگیری ماشین است و با توجه به گستردگی آن، در زمینه‌های گوناگونی مانند نظریه بازی‌ها[1]، نظریه کنترل[2]، سامانه‌های چند عامله[3]، نظریه اطلاعات[4] 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.
در این پروژه قصد داریم با استفاده از برخی الگوریتم‌های یادگیری تقویتی، مسائل مشابه دنیای واقعی را حل کنیم. در این تمرین سه الگوریتم Value Iteration ، Policy Iteration و Q-Learning را در شرایط مختلف پیاده‌سازی و تحلیل می‌کنید.

توضیح مسائل

مسئله اول: Frozen Lake

مسئله دریاچه یخزده شامل عبور از یک دریاچه یخزده از خانه شروع[5] تا خانه هدف[6] بدون افتادن در هیچ سوراخی[7] با حرکت بر روی دریاچه[8] است. محیط دریاچه یک جدول 4 در 4 است. برای آشنایی دقیق‌تر با این محیط می‌توانید از این لینک استفاده کنید.

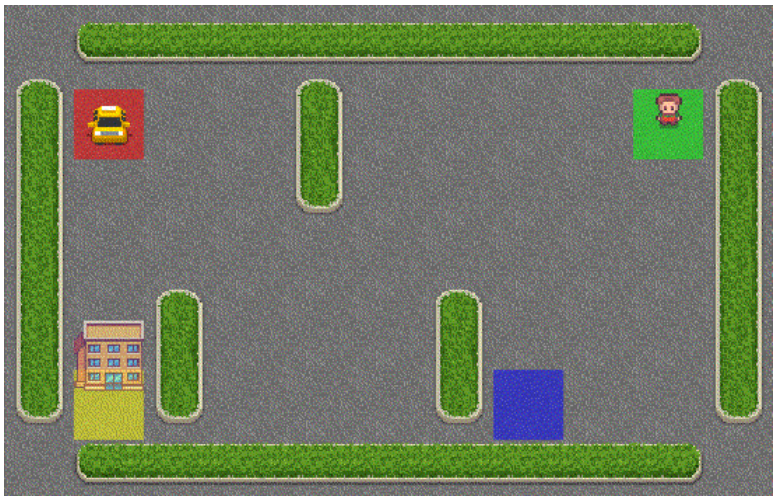


[1] Game Theory
[2] Control Theory
[3] Multi-Agent Systems
[4] Information Theory
[5] Start(S)
[6] Goal(G)
[7] Hole(H)
[8] Frozen(F)

عامل در هر زمان می‌تواند از بین چهار حرکت چپ(0)، پایین(1)، راست(2) و بالا(3) یکی را انتخاب کند. البته توجه کنید که در حالت‌های مرزی در صورت انتخاب حرکت غیرمجاز عامل در سر جای خود باقی می‌ماند. عامل به ازای رسیدن به خانه هدف پاداش 1+ و به ازای افتادن در سوراخ یا حرکت در دریاچه پاداش 0 دریافت می‌کند.

مسئله دوم: Taxi

غضنفر که برخلاف شما درس هوش مصنوعی را ندارد، قصد دارد تاکسی اینترنتی‌ای راه‌اندازی کند که نیازی به راننده نداشته باشد و عاملی هوشمند به جای راننده مسافرین را به مقصدشان برساند. بدین منظور ابتدا می‌خواهد که امکان این مسئله را در یک محیط فرضی بسنجد. محیط شهر یک جدول ۵ در ۵ است که دور تا دور و بخش‌هایی از درون آن دیوارهایی وجود دارد. برای آشنایی دقیق‌تر با این محیط می‌توانید از این لینک استفاده کنید.



عامل در هر زمان می‌تواند از بین شش حرکت پایین(0)، بالا(1)، راست(2)، چپ(3)، سوار کردن مسافر(4) و پیاده کردن مسافر(۵) یکی را انتخاب کند. در حالت‌های مرزی، در صورت انتخاب حرکت غیرمجاز عامل در سر جای خود باقی می‌ماند. از طرفی، بسته به شماره‌ی دانشجویی شما، مبدا مسافر یکی از خانه‌های رنگی و مقصد وی خانه‌ی دیگر می‌باشد. عامل به ازای رساندن مسافر پاداش 20+، به ازای سوار کردن یا پیاده کردن غیر مجاز پاداش 10-، و در غیر این صورت به دلیل زمان از دست رفته پاداش 1- دریافت می‌کند.

نحوه‌ی استفاده از محیط

برای این تمرین قصد داریم با Gym که یک رابط کاربردی برای یادگیری تقویتی و مجموعه‌ای از محیط‌های مختلف است، آشنا شویم. در این لینک توضیح ساده‌ای از نحوه‌ی استفاده از محیط‌های آن داده شده است. همچنین کد این محیط، در فایل ضمیمه آورده شده است.

توجه داشته باشید که برای محیط Taxi، شما حتماً باید در زمان reset محیط، seed محیط را برابر با سه رقم آخر شماره دانشجویی خود کنید. برای مثال اگر شماره‌ی دانشجویی شما 810100123 باشد، قطعه کد به صورت زیر خواهد بود.

```
import gym
env = gym.make('Taxi-v3')
env.seed(seed=123)
Initial_state = env.reset()
```

هر حالت در این محیط با یک عدد نمایش داده می‌شود. میتوان از دستور زیر برای پیدا کردن حالت مسئله استفاده کرد. مثلا اگر در حالت 139 باشیم، اطلاعات حالت مذکور را به شکل زیر میتوان پیدا کرد.

```
taxi_row, taxi_col, pass_idx, dest_idx = env.decode(139)
```

نکات پیاده‌سازی

برای پیاده‌سازی سوالات آینده به این نکات توجه فرمایید:

- سیاست مورد استفاده برای عامل را epsilon-greedy در نظر بگیرید.
- در تمامی سوالات به جز ذکر صریح در صورت سوال مقدار اپسیلون را با روشی مناسب کاهش داده و مقدار ضریب تخفیف[9] را 0.9 در نظر بگیرید. همچنین مقدار نرخ یادگیری[10] را برابر 0.1 در نظر بگیرید.
- برای روش‌های Policy Iteration و Value Iteration، مسئله را به اندازه‌ی حداقل 100 اپیزود انجام دهید و همگرایی به سیاست بهینه را بررسی نمایید.
- برای روش Q-Learning، مسئله را حداقل 20 بار تکرار و به اندازه‌ی حداقل 2000 اپیزود انجام دهید، متوسط مجموع پاداش دریافتی در پایان هر اپیزود در طول یادگیری را رسم نمایید و همگرایی به سیاست بهینه را بررسی نمایید.
- همچنین در پایان هر یک از مسائل بخش پیاده سازی، یکی از عامل‌ها (بهترین یا میانگین عامل‌ها) را پس از آموزش، به اندازه‌ی 40 اپیزود تست کنید و همچنین با استفاده از دستور render رفتار آن را نمایش دهید. (نمایش رفتار عامل به صورت گرافیکی و رنگی نمره امتیازی خواهد داشت.)

تذکر: دقت شود که پارامترهای داده شده صرفا به عنوان یک گزینه‌ی اولیه بوده و ممکن است پارامترها را بتوان طوری تنظیم کرد که یادگیری بهتر شود. در صورتی که در صورت سوال صریحاً قید نشده باشد، شما می‌توانید این پارامترها را تغییر دهید.

پیاده‌سازی مسائل

بخش یک: حل مسئله Frozen Lake

در این بخش به پیاده سازی دو الگوریتم Value Iteration ، Policy Iteration و حل مسئله Frozen Lake با استفاده از این دو الگوریتم خواهید پرداخت.

1. در مورد الگوریتم Value Iteration و نحوه بدست آمدن سیاست بهینه توسط این الگوریتم تحقیق کنید و توضیح مختصری درباره آن در گزارش خود ذکر کنید. جهت کسب اطلاعات بیشتر در مورد این الگوریتم می‌توانید به بخش 4.4 کتاب Reinforcement Learning: An Introduction مراجعه کنید.
2. کد مربوط به الگوریتم Value Iteration را در فایل نوت‌بوک بنویسید. توجه کنید که نیاز نیست این الگوریتم را از ابتدا پیاده سازی کنید و تنها قسمت های مشخص شده در فایل نوت‌بوک را کامل کنید.
3. در مورد الگوریتم Policy Iteration و نحوه بدست آمدن سیاست بهینه توسط این الگوریتم تحقیق کنید و توضیح مختصری درباره آن در گزارش خود ذکر کنید.. جهت کسب اطلاعات بیشتر در مورد این الگوریتم می‌توانید به بخش 4.3 کتاب Reinforcement Learning: An Introduction مراجعه کنید.
4. کد مربوط به الگوریتم Policy Iteration را در فایل نوت‌بوک بنویسید. توجه کنید که نیاز نیست این الگوریتم را از ابتدا پیاده سازی کنید و تنها قسمت‌های مشخص شده در فایل نوت‌بوک را کامل کنید.

---

[9] discount factor
[10] learning rate

5. حال، با استفاده از این دو الگوریتم به حل مسئله پرداخته و مقدار ارزش استیت‌ها[11]، ارزش استیت–اکشن‌ها[12] و سیاست بهینه[13] را بدست آورید (مقدار ارزش استیت‌ها و سیاست بهینه را با رسم یک جدول 4 در 4 بر روی نقشه نشان دهید). همچنین رفتار عامل با پیروی از سیاست بهینه را برای هر یک از این دو الگوریتم بر روی نقشه نشان دهید.
6. مقادیر ارزش‌ها و سیاست بهینه‌ی بدست آمده توسط این دو الگوریتم را با یکدیگر مقایسه کرده و تحلیل کنید. همچنین، سرعت همگرایی الگوریتم‌ها را با هم مقایسه کنید.

بخش دو: حل مسئله Taxi

در این بخش با الگوریتم Q-Learning آشنا خواهید شد و مسئله Taxi را در شرایط مختلف با این الگوریتم حل و بررسی خواهید کرد.

7. در مورد الگوریتم Q-Learning تحقیق کنید و توضیح مختصری درباره آن در گزارش خود بنویسید. جهت کسب اطلاعات بیشتر در مورد این الگوریتم می‌توانید به بخش 6.5 کتاب Reinforcement Learning: An Introduction مراجعه کنید.
8. کد مربوط به الگوریتم Q-Learning را در فایل نوت‌بوک کامل کنید. توجه کنید که نیاز نیست این الگوریتم را از ابتدا پیاده سازی کنید و تنها قسمت های مشخص شده در فایل نوت‌بوک را کامل کنید.
9. الگوریتم Q-Learning را یکبار به ازای نرخ یادگیری ثابت 0.1 و بار دیگر به ازای نرخ یادگیری کاهشی پیاده‌سازی نمایید و نتایج بدست آمده را از حیث میزان پاداش (سرعت همگرایی و مقدار همگرا شده) با یکدیگر مقایسه کنید. روش انتخابی خود برای کاهش مقدار اپسیلون در طی فرآیند یادگیری را توضیح دهید.
10. رفتار عامل با پیروی از سیاست بهینه را برای این الگوریتم بر روی نقشه نشان دهید.

نکات پیاده‌سازی و تحویل

- نمودارها حتماً باید label axis، title و grid داشته باشند و مقادیر به‌صورت گویا نمایش داده شود.
- 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_CA2_[Std number].zip

[11] V(s) = state value
[12] Q(s,a) = state-action value
[13] π^* = optimal policy