دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر

«مبانی و کاربردهای هوش مصنوعی ترم پائیز ۱۴۰۴»

پروژه سوم

در انجام پروژه‌ها به نکات زیر توجه فرمائید:

۱- پیاده‌سازی پروژه‌ها را به زبان برنامه‌نویسی پایتون انجام دهید.

۲- مطابق قوانین دانشگاه هر نوع کپی‌برداری و اشتراک کار دانشجویان غیرمجاز است و پاسخ به پروژه‌ها باید به صورت انفرادی و بدون استفاده از ابزارهای هوش مصنوعی انجام شود، در صورت مشاهده چنین مواردی با طرفین شدیدا برخورد خواهد شد.

۳- فایل پروژه را با فرمت StudentID_AI_P03.zip تا ساعت ۲۳:۵۹ روز ۱۴۰۴/۰۹/۱۹ فقط در بخش مربوطه در سایت درس آپلود نمائید.

۴- توجه نمائید پاسخ پروژه‌ها تنها در صورت آپلود در سامانه کورسز پذیرفته خواهد شد و ارسال پاسخ از طریق ایمیل یا تلگرام بررسی نخواهد شد.

۵- در مجموع برای پروژه‌ها ۱۰ روز تاخیر مجاز دارید که می‌توانید در طول ترم بسته به شرایط از آن استفاده نمائید، در صورت اتمام تاخیر مجاز، هر روز تاخیر منجر به کسر نمره از پروژه خواهد شد.

فهرست مطالب

۱- مقدمه

یکی از بنیادی‌ترین مفاهیم در هوش مصنوعی، یادگیری از طریق تعامل با محیط و دریافت بازخورد است. در بسیاری از مسائل دنیای واقعی، عامل هوشمند نه از قبل نقشه‌ی کاملی از محیط دارد و نه می‌تواند همه‌ی پیامدهای اعمال خود را دقیق پیش‌بینی کند؛ بلکه باید با آزمون و خطا و بر اساس پاداش‌ها و جریمه‌هایی که دریافت می‌کند، کم‌کم یاد بگیرد چه رفتاری خوب و چه رفتاری بد است. این ایده‌ی کلی هسته‌ی اصلی یادگیری تقویتی است؛ جایی که یک عامل در چارچوب یک فرایند تصمیم‌گیری مارکوف تلاش می‌کند سیاستی بیاموزد که در بلندمدت بیشترین پاداش ممکن را به دست آورد. هدف از این پروژه، درک عمیق‌تر این مفاهیم از طریق پیاده‌سازی عملی الگوریتم‌های تقویتی و مشاهده‌ی رفتار آن‌ها در یک محیط تعاملی است.

در این پروژه، شما با محیط کلاسیک LunarLander کار خواهید کرد؛ محیطی که در آن یک فرودگر در میدان گرانشی سیاره‌ای خیالی قرار دارد و باید با استفاده از نیروی موتورهای جانبی و اصلی، به‌صورت کنترل‌شده روی سطح فرود آید. حالت‌های محیط شامل موقعیت و سرعت فرودگر، زاویه و سرعت زاویه‌ای آن و همچنین وضعیت تماس پایه‌ها با سطح هستند و اعمال، روشن و خاموش کردن موتورهای مختلف را در بر می‌گیرند. تابع پاداش به‌گونه‌ای طراحی شده که فرود نرم و دقیق را تشویق می‌کند، برای مصرف بی‌رویه‌ی سوخت و حرکات نامناسب جریمه می‌دهد و برای سقوط یا خروج از محدوده‌ی مجاز، پاداش منفی سنگینی در نظر می‌گیرد. این طراحی باعث می‌شود مسئله نه‌تنها از نظر دینامیک حرکت، بلکه از منظر شکل‌دهی پاداش و پایدار کردن یادگیری نیز چالش‌برانگیز شود.

این پروژه به‌گونه‌ای طراحی شده است که شما در نقش طراح یک عامل هوشمند قرار می‌گیرید که باید با تکیه بر الگوریتم‌های مختلف یادگیری تقویتی، رفتاری مناسب برای کنترل فرودگر بیاموزد. در گام‌های مختلف، ابتدا با Evaluation Policy Iterative و سپس با Iteration Policy کار خواهید کرد تا بفهمید چگونه می‌توان ارزش هر حالت را تحت یک سیاست مشخص محاسبه و سپس خود سیاست را بهبود داد. در ادامه، با پیاده‌سازی Learning-Q، با مفاهیمی مانند تعادل میان اکتشاف و بهره‌برداری، نقش نرخ یادگیری و ضریب تنزیل، و چالش‌های همگرایی در محیط‌های نویزی و پیچیده روبه‌رو می‌شوید. تفاوت در

پاداش‌ها، احتمال خطا در فرود، و امکان گیر افتادن در رفتارهای زیان‌بار باعث می‌شود تحلیل و مقایسه‌ی این روش‌ها معنای عمیق‌تری پیدا کند.

در نهایت، هدف این پروژه صرفا پیاده‌سازی چند الگوریتم نیست، بلکه کسب یک درک شهودی از رفتار و کارایی آن‌ها در یک محیط پویاست. با بررسی نمودارهای پاداش در طول اپیزودها، مشاهده‌ی ویدیوی رفتار عاملِ آموزش‌دیده، تحلیل سرعت همگرایی و پایداری سیاست نهایی، درمی‌یابید که چگونه انتخاب الگوریتم، تنظیم پارامترها و طراحی پاداش می‌تواند تفاوت چشمگیری در کیفیت، سرعت و بهینگی یادگیری ایجاد کند. این تجربه می‌تواند یکی از اولین گام‌های عملی شما در درک نحوه‌ی تصمیم‌گیری هوشمند در محیط‌های پیچیده و نامطمئن باشد.

۲- شرح دقیق محیط

در این پروژه از محیط LunarLander کتابخانه‌ی Gymnasium استفاده می‌کنیم. این محیط یک نسخه‌ی ساده‌شده از مسئله‌ی کنترل و بهینه‌سازی مسیر یک فرودگر است که باید در میدان گرانش، روی سکوی فرود در مختصات تقریبا $(0,0)$ به‌طور ایمن فرود بیاید. موتورهای فرودگر یا کاملاً خاموش‌اند یا با توان بالا کار می‌کنند و همین باعث می‌شود مسئله بیشتر شبیه یک مسئله‌ی تصمیم‌گیری گسسته شود تا کنترل پیوسته. سوخت محیط از دید عامل نامحدود است، بنابراین عامل می‌تواند چندین بار امتحان کند، در فضا مانور بدهد و در نهایت فرود خوبی یاد بگیرد.

۲-۱- فضای اعمال

در نسخه‌ای که در این پروژه استفاده می‌کنیم، فضای اعمال گسسته و شامل ۴ عمل ممکن است:

- ۰: عدم انجام هیچ کاری (تمام موتورها خاموش هستند.)
- ۱: روشن کردن موتور جانبی چپ (ایجاد نیروی رانش که فرودگر را به سمت راست می‌چرخاند.)
- ۲: روشن کردن موتور اصلی (ایجاد نیروی رانش به سمت بالا و کاهش سرعت سقوط.)

- ۳: روشن کردن موتور جانبی راست (ایجاد نیروی رانش که فرودگر را به سمت چپ می‌چرخاند.)

نسخه‌ی پیوسته‌ی این محیط نیز وجود دارد که در آن عمل یک بردار دو بعدی در بازه‌ی $[-1,1]$ است و شدت رانش موتور اصلی و موتورهای جانبی را مشخص می‌کند.

۲-۲- فضای مشاهدات

حالت محیط به‌صورت یک بردار ۸ بعدی واقعی مدل می‌شود که هر مولفه‌ی آن اطلاعاتی درباره‌ی وضعیت فعلی فرودگر می‌دهد:

- مختصات افقی فرودگر نسبت به مرکز صفحه
- مختصات عمودی فرودگر
- سرعت افقی
- سرعت عمودی
- زاویه‌ی چرخش نسبت به افق
- سرعت زاویه‌ای
- تماس پای چپ با زمین (۰ یا ۱)
- تماس پای راست با زمین (۰ یا ۱)

۲-۳-تابع پاداش

پس از هر گام، عامل یک پاداش عددی دریافت می‌کند و مجموع پاداش‌های یک اپیزود معیار اصلی عملکرد آن است. ساختار پاداش در این محیط به‌صورت تقریبی به شکل زیر است:

- هرچه فرودگر به سکوی فرود نزدیک‌تر باشد، پاداش افزایش می‌یابد و هرچه دورتر شود، پاداش کاهش پیدا می‌کند.

- هرچه سرعت‌های افقی و عمودی کم‌تر (فرود نرم‌تر) باشند، پاداش بیشتر است و سرعت‌های زیاد (حرکت خشن یا سقوط تند) جریمه می‌شوند.
- انحراف زیاد از حالت افقی (زاویه‌ی بزرگ) جریمه دارد و تلاش می‌شود فرودگر تقریباً صاف روی سطح بنشیند.
- برای هر پا که با سطح تماس پیدا می‌کند، حدود ۱۰ امتیاز مثبت به پاداش اضافه می‌شود.
- برای مصرف سوخت، جریمه‌ی کوچکی در هر فریم در نظر گرفته شده است:
- روشن بودن موتورهای جانبی در هر فریم حدود ۰.۰۳ امتیاز کم می‌کند.
- روشن بودن موتور اصلی در هر فریم حدود ۰.۳ امتیاز کم می‌کند.
- اگر فرودگر سالم و روی سکو فرود بیاید، در پایان اپیزود ۱۰۰ امتیاز اضافه می‌شود.
- اگر فرودگر سقوط کند یا به‌شدت برخورد کند، در پایان اپیزود ۱۰۰ امتیاز دریافت می‌کند.

به‌صورت متعارف گفته می‌شود یک اپیزود (یا سیاست) زمانی خوب در نظر گرفته می‌شود که امتیاز کلی آن حداقل حدود ۱۰۰ باشد.

۲-٤-وضعیت اولیه

در ابتدای هر اپیزود، فرودگر تقریبا در بالای مرکز صفحه قرار می‌گیرد و یک نیروی اولیه‌ی تصادفی کوچک به آن اعمال می‌شود تا جهت و سرعت شروع در هر اپیزود کمی متفاوت باشد. این تصادفی بودن باعث می‌شود عامل فقط برای یک وضعیت خاص حفظی یاد نگیرد، بلکه سعی کند سیاستی یاد بگیرد که در طیفی از وضعیت‌های اولیه نیز عملکرد خوبی داشته باشد.

ساختار کلی پروژه و وظیفه‌ی شما

در این پروژه باید ابتدا محیط LunarLander را به‌صورت یک MDP مدل کنید؛ یعنی مشخص کنید حالت‌ها چه هستند، چه اعمالی در هر لحظه در اختیار عامل قرار دارد، پاداش‌ها چگونه تعریف شده‌اند و با چه ضریب کاهش γ مجموع پاداش‌ها را در نظر می‌گیرید. بعد از این صورت‌بندی مفهومی، یک سیاست تصادفی پیاده‌سازی می‌کنید که در هر حالت، به‌طور یکنواخت یکی از اعمال ممکن را انتخاب می‌کند و با اجرای چندین اپیزود، میانگین پاداش و رفتار این سیاست ساده را به‌عنوان خط پایه اندازه می‌گیرید. سپس نوبت به پیاده‌سازی Evaluation Policy Iterative می‌رسد در این مرحله با استفاده از معادله‌های تکراری، مقدار ارزش حالت‌ها را برای یک سیاست مشخص)مثلاً همین سیاست تصادفی (تخمین می‌زنید. در ادامه با به‌کارگیری ایده‌ی Improvement Policy و تکرار چرخه‌ی ارزیابی سیاست و بهبود سیاست، الگوریتم Iteration Policy را پیاده‌سازی می‌کنید تا از یک سیاست اولیه‌ی ساده، به سیاستی به‌مراتب بهتر و پایدار برسید.

در گام پایانی، باید الگوریتم Learning-Q را برای همین محیط پیاده‌سازی کنید. چون فضای حالت LunarLander پیوسته است، لازم است ابتدا آن را به روشی مناسب گسسته‌سازی کنید تا بتوانید یک جدول $Q(s,a)$ رای حالت–عمل‌ها بسازید. سپس با استفاده از یک سیاست اپسیلون گریدی، در هر گام بین اکتشاف و بهره‌برداری تعادل برقرار می‌کنید، و با به‌روزرسانی تکراری Q طبق فرمول استاندارد Learning-Q، عامل را روی تعداد زیادی اپیزود آموزش می‌دهید. در انتها، با اجرای عاملِ آموزش‌دیده بدون اکتشاف و محاسبه‌ی میانگین پاداش و مشاهده‌ی رفتار فرودگر، عملکرد Q-Learning را با سیاست تصادفی و سیاست حاصل از Iteration Policy مقایسه و در گزارش خود تحلیل می‌کنید.

گزارش کار و تحلیل نتایج

در گزارشکار خود باید ابتدا در یک بخش، محیط را به‌صورت یک MDP توصیف کنید؛ یعنی توضیح دهید حالت‌ها، اعمال ممکن، پاداش‌ها و حالت‌های پایانی چه هستند و سپس مقدار تابع ارزش تحت سیاست تصادفی یکنواخت را نشان دهید و به‌صورت کیفی تحلیل کنید که چرا بعضی خانه‌ها ارزش بالاتری دارند. بعد از آن، تابع ارزش بهینه و سیاست بهینه‌ای را که با استفاده از Iteration Policy به دست آورده‌اید ارائه کنید و تفاوت آن را با سیاست تصادفی از نظر ساختار مسیرها و مقادیر ارزش را کامل توضیح دهید.

در بخش بعد، برای محیط LunarLander باید فضای مشاهدات و اعمال را معرفی کنید، دقیقا توضیح دهید که حالت پیوسته را چگونه گسسته‌سازی کرده‌اید و Learning-Q پیاده‌سازی‌شده را توصیف کنید. سپس باید نتایج را ارائه و تحلیل کنید: بازده سیاست تصادفی را با عامل Learning-Q مقایسه کرده، حداقل یک نمودار منحنی یادگیری بازده برحسب شماره‌ی اپیزود بیاورید. در بخشی جداگانه، بر اساس ویدیوهای ضبط‌شده از اجرای سیاست‌ها، به‌صورت کیفی تفاوت رفتار عامل تصادفی و عامل یادگرفته‌شده را توصیف کنید مثلا این‌که عامل تصادفی چطور اغلب سقوط می‌کند یا از محدوده خارج می‌شود، و عامل یادگرفته‌شده معمولا در چه شرایطی موفق به فرود نرم می‌شود و چه الگوهای شکستی هنوز در آن دیده می‌شود. در پایان گزارش، بخشی برای بحث و بهبودهای ممکن قرار دهید و در آن محدودیت‌های روش Learning-Q جدولی برای LunarLander را ذکر کنید و ایده‌هایی برای بهبود از جمله گسسته‌سازی بهتر، استراتژی‌های اکتشاف متفاوت و... پیشنهاد دهید.