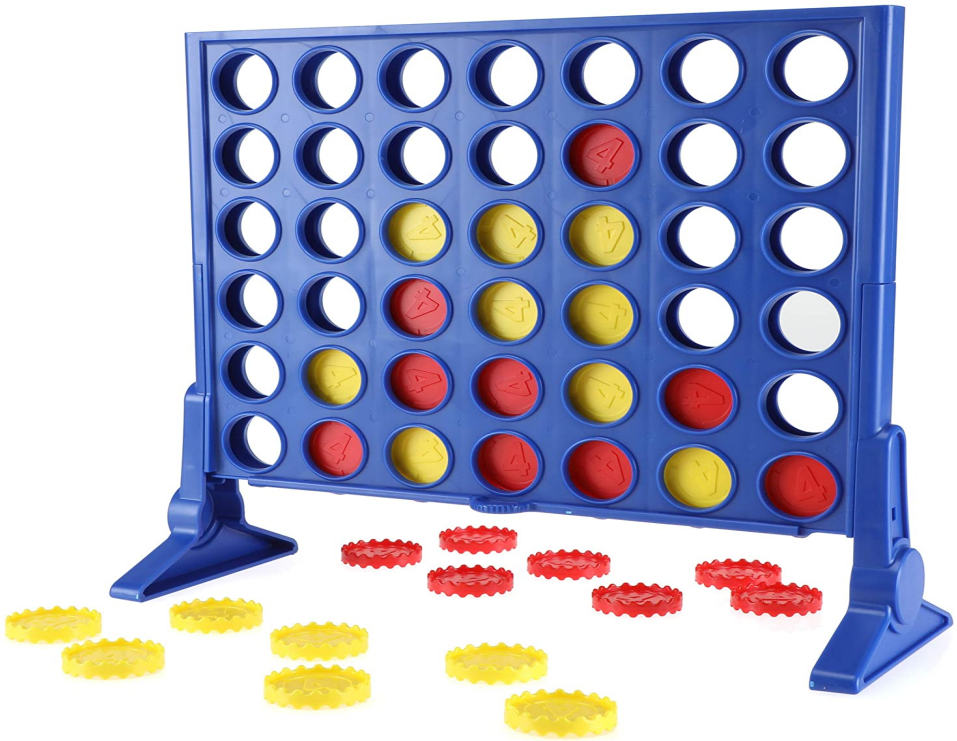
هوش مصنوعی
تمرین شماره‌ی ۲: بازی

مدرس: **دکتر فدایی و دکتر یعقوب‌زاده**

طراحان: سینا کمالی، امیرحسین عباسکوهی

مهلت تحویل: یکشنبه 21 فروردین ۱۴۰۱، ساعت ۲۳:۵۵

مقدمه



در یک روز آفتابی شما و دوست عزیزتان هاشم در حال قدم زنی در پارک هستید. هاشم که از بازی کردن Elden Ring خسته شده بود، تصمیم گرفت که اوقات فراغت‌تان را صرف یک بازی حضوری و به دور از مانیتور کنید. او بازی Connect4 را پیشنهاد داد. شما تمام روز را سرگرم بازی کردن بودید و در نهایت متوجه شدید که تعداد برد ها نصف‌نصف بوده است. این نتیجه هیچکدام از شما را راضی نمی کرد. از آنجا که هاشم هم مثل شما با مباحث هوش مصنوعی آشنا بود، پیشنهاد داد که هر دو بروید و بر روی agent ای برای بازی کردن Connect4 کار کنید و پس از مدتی آن دو agent را مجبور به رقابت با هم بکنید که برنده نهایی را از آنجا شناسایی کنید. شما که به هیچ وجه نمی خواستید به هاشم ببازید، تصمیم گرفتید که با استفاده از الگوریتم min-max یک agent ای بنویسید که یک بار برای همیشه هاشم را به اتاقش و به پیش Elden Ring برگرداند.

توضیح بازی

همانطور که در مقدمه بازی شرح داده شد، در بازی Connect4 هدف، رسیدن به 4 مهره هم‌رنگ در یک خط افقی، عمودی و یا مورب است. اولین بازیکنی که به شرط بالا برسد بازی را پیروز می شود. در صورتی که تمام خانه ها پر شوند و شرط 4 خانه با مهره های هم‌رنگ اتفاق نیفتاده باشد، آن‌گاه بازی مساوی شده است.
صفحه بازی Connect4، یک صفحه با 6 ردیف و 7 ستون می باشد. هر بازیکن در هر مرحله، باید یکی از ستون ها را انتخاب کند. مهره بازیکن در ستون انتخاب شده تا جایی که به مهره دیگری برسد و یا به پایین ترین ردیف برسد پایین می‌رود و در آنجا قرار می‌گیرد.

پیاده سازی

هدف شما پیاده سازی الگوریتم min-max برای شکست دادن هاشم است. کد بازی به شما داده شده است اما این کد کامل نیست و شما باید بخش های ToDo را کامل کنید. شما باید تابع get_your_input را کامل کنید که فقط یک مقدار، که انتخاب شما برای ستونی که در آن مهره می‌اندازید است، را برمی‌گرداند. برای محاسبه این مقدار از الگوریتم min-max استفاده کنید و پس از پیاده سازی، آن در این تابع صدا کنید.
شما می توانید برای تمیزی کد خود، متد و توابع دیگری را به کد اضافه کنید اما حق هیچ گونه تغییری در بخش های دیگر کد را ندارید و این بخش ها باید ثابت باقی بمانند (اضافه کردن مواردی مثل getter و یا setter مانعی ندارد اما باید در گزارش کار ذکر کنید).

دقت کنید که باید برای الگوریتم min-max خود یک تابع heuristic برای ارزشیابی هر یک از حالات تعریف کنید. تابع heuristic خود را در گزارش شرح دهید.

بررسی نتایج

برای درک کامل و آزمایش کد خود، با 3 بار ران کردن کد خود برای عمق های 1، 3 و 5، بررسی کنید که شانس پیروزی شما چه مقدار است. همینطور زمان اجرای هر عمق را ثبت کنید.

هرس آلفا و بتا: برای افزایش سرعت کد خود و کاهش نود های خود، هرس آلفا و بتا را به کد اضافه کنید و سرعت اجرای کد، تعداد نود های مورد بررسی و شانس پیروزی را برای عمق های یاد شده مجدداً بررسی کنید. همچنین عمق 7 را به عمق های مورد بررسی خود اضافه کنید.

نکات تکمیلی:

- در هر بخش، بازی را با سایز زمین های 7*6 (سایز معروف و همیشگی بازی)، 8*7 و همینطور 10*7 اجرا کنید و نتایج تمامی بخش ها (جدولی از ترکیب تمامی سایز ها و عمق ها) را به صورت کامل در گزارش خود بیاورید.
- روش محاسبه شانس پیروزی: بازی را 200 مرتبه در حالت مد نظر اجرا کنید و با بدست آوردن تعداد برد ها، شانس پیروزی را حساب کنید.

سوالات

- **سوال 1**: یک heuristic خوب چه ویژگی هایی دارد؟ علت انتخاب heuristic شما و دلیل برتری آن نسبت به تعدادی از روش های دیگر را بیان کنید.
- **سوال 2**: آیا میان عمق الگوریتم و پارامتر های حساب شده روابطی می‌بینید؟ به طور کامل بررسی کنید که عمق الگوریتم چه تاثیراتی بر روی شانس پیروزی، زمان و گره های دیده شده می‌گذارد.
- **سوال 3**: وقتی از روش هرس کردن استفاده می کنید، برای هر گره درخت، فرزندانش به چه ترتیبی اضافه می شوند؟ آیا این ترتیب اهمیت دارد؟ چرا این ترتیب را انتخاب کردید؟

نکات پایانی

- نتایج و گزارش های خود را در یک فایل فشرده با عنوان AI-CA2-Game-SID.zip آپلود کنید.
- در صورت هرگونه سوال بهتر است در فروم درس مطرح کنید تا بقیه هم از آن استفاده کنند، در غیر این‌صورت با طراحان در ارتباط باشید.
- هدف از تمرین یادگیری شماست. لطفاً خودتان انجام دهید.