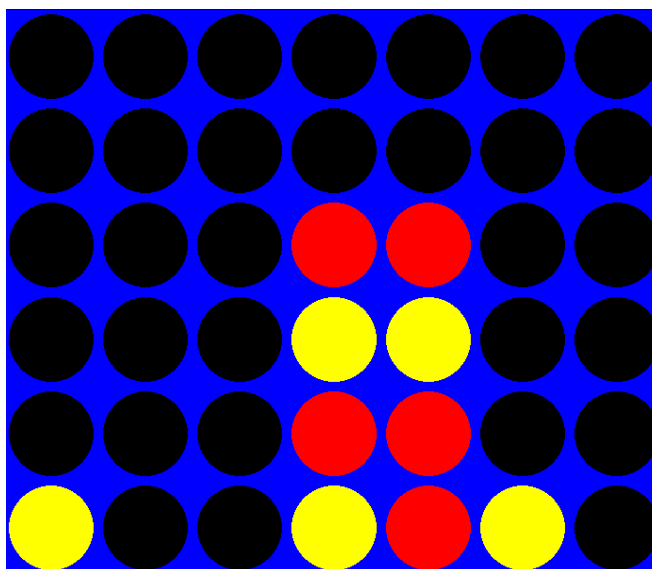


قسمت دوم : بازی Connect4

شما و دوستتان تصمیم گرفته‌اید که بازی Connect4 را بازی کنید ولی از آنجایی که شما به شدت تازه‌کار هستید به دنبال راه چاره می‌گردید که به یاد یکی از مباحث درس هوش مصنوعی می‌افتید. شما تصمیم می‌گیرید که الگوریتم minimax را پیاده‌سازی کنید تا شانس برد خود را افزایش بدهید. پس از پیاده‌سازی این الگوریتم، تصمیم می‌گیرید پیش از بازی با دوستانتان، الگوریتم را روی کامپیوتر امتحان کنید تا از درستی آن مطمئن شوید.

توضیح بازی

بازی Connect4 یک بازی دو نفره استراتژیک است که روی یک صفحه 6×7 انجام می‌شود. این صفحه شامل 6 سطر و 7 ستون است و بازیکنان به نوبت مهره‌های خود را در یکی از ستون‌های خالی صفحه می‌اندازند. مهره‌ها همیشه به پایین‌ترین جای خالی در ستون انتخاب‌شده می‌افتند و ستون‌ها به مرور با مهره‌ها پر می‌شوند.



هدف بازی این است که یک بازیکن بتواند چهار مهره‌ی خود را به صورت متوالی در یک ردیف افقی، عمودی یا مورب قرار دهد. همچنین در صورتی که تمام خانه‌های صفحه پر شوند و هیچ‌کدام از بازیکنان نتوانند چهار مهره متوالی در یک ردیف داشته باشند، بازی مساوی اعلام می‌شود. این بازی نیاز به برنامه‌ریزی و پیش‌بینی حرکات دارد، زیرا هر حرکتی که یک بازیکن انجام می‌دهد می‌تواند فرصت‌های جدیدی برای حریف ایجاد کند.

در این بازی شما با مهره های قرمز بازی می‌کنید و مهره های زرد برای حریف شما یعنی کامپیوتر در نظر گرفته شده و شروع کننده بازی به صورت شانسی انتخاب می‌شود. فرض کنید که کامپیوتر نیز مانند شما بهینه عمل می‌کند.

جهت آشنایی بیشتر با فرایند بازی می‌توانید به این [لینک](#) مراجعه کنید.

پیاده سازی

هدف شما پیاده‌سازی الگوریتم minimax برای شکست دادن دوستتان است. کد بازی به شما داده شده است اما این کد کامل نیست و شما باید بخش TODO را کامل کنید. شما باید تابع minimax را کامل کنید که در واقع پیاده‌سازی الگوریتم minimax برای این بازی است.

شما می‌توانید برای تمیزی کد خود، متد و توابع دیگری را به کد اضافه کنید اما بهتر است تغییری در بخش‌های دیگر کد ایجاد نکنید و این بخش‌ها ثابت بمانند. برای الگوریتم minimax خود، به یک تابع heuristic برای ارزشیابی هر یک از حالات نیاز دارید که کد آن نیز به شما داده شده است.

برای استفاده از کد کافیسیت یک نمونه از کلاس Connect4Game با توجه به آرگومان‌های مد نظر بسازید و پس از پیاده سازی تابع minimax، تابع play را در نمونه‌ی خود صدا زده و نتیجه‌ی بازی به صورت زیر برگردانده می‌شود:

1: بازیکن برنده شده است.

0: بازی مساوی شده است.

-1: کامپیوتر برنده شده است.

همچنین برای استفاده از رابط گرافیکی به منظور پیاده‌سازی راحت‌تر، می‌توانید پرچم ui را هنگام ساخت نمونه True کنید. می‌توانید از متد print_board نیز برای بررسی صفحه بازی در هر مرحله استفاده کنید.

بررسی نتایج

برای درک کامل الگوریتم کد را با عمق‌های مختلف و بدون هرس 50 الی 100 بار اجرا کنید و میانگین زمان و شانس پیروزی و میانگین تعداد نودهای دیده شده را برای هر عمق حساب کنید.

هرس آلفا و بتا: برای افزایش سرعت کد و کاهش نودهای دیده شده، هرس آلفا و بتا را به کد اضافه کنید و موارد ذکر شده در بخش قبل را مجدداً بررسی کنید.

در نهایت متد `check_results` را فراخوانی کنید و نتایج بدست آمده را در گزارش خود توضیح دهید.

سوالات

1. آیا میان عمق الگوریتم و پارامترهای حساب شده در بخش بالا روابطی می بینید؟ بررسی کنید که عمق الگوریتم چه تاثیراتی بر روی شانس پیروزی، زمان و گره های دیده شده می گذارد.
2. آیا می توان ترتیب دیدن فرزندان هر نود را به گونه ای انتخاب کنیم که بیشترین هرس را داشته باشیم؟ اگر جواب شما مثبت است روش خود را توضیح دهید و در غیر اینصورت توضیح دهید که چرا این عمل امکان پذیر نیست.
3. `Branching Factor` را توضیح دهید و بگویید که با پیشرفت این بازی چه تغییراتی می کند؟
4. توضیح دهید که چرا به هنگام هرس کردن الگوریتم بدون از دست دادن دقت خود سریع تر می شود.
5. چرا در حالاتی که حریف به صورت شانسی عمل می کند (مانند این پروژه)، استفاده از `minimax` بهینه ترین روش نیست؟ چه الگوریتمی می تواند جایگزین این الگوریتم باشد؟ توضیح دهید.

نکات پایانی

- دقت کنید که کد شما باید به نحوی زده شده باشد که نتایج قابلیت بازتولید داشته باشند.
- توضیحات مربوط به هر بخش از پروژه را بطور خلاصه و در عین حال مفید در گزارش خود ذکر کنید. از ابزارهای تحلیل داده مانند نمودارها استفاده کنید. حجم توضیحات گزارش شما هیچ گونه تاثیری در نمره نخواهد داشت و تحلیل و نمودارهای شما بیشترین ارزش را دارد.
- سعی کنید از پاسخ های روشن در گزارش خود استفاده کنید و اگر پیش فرضی در حل سوال در ذهن خود دارید، حتما در گزارش خود آن را ذکر نمایید.
- پس از مطالعه کامل و دقیق صورت پروژه، در صورت وجود هرگونه ابهام یا سوال با طراحان پروژه در ارتباط باشید.
- نتایج، گزارش و کدهای خود را در قالب یک فایل فشرده با فرمت `AI_CA2_[stdNumber].zip` در سامانه ایلرن بارگذاری کنید. به طور مثال `AI_CA2_810101999.zip`
- محتویات پوشه باید شامل فایل پاسخ های شما به سوالات کتبی، فایل `jupyter-notebook`، خروجی `html` و فایل های مورد نیاز برای اجرای آن باشد. از نمایش درست خروجی های مورد نیاز در فایل `html` مطمئن شوید.

- توجه کنید این تمرین باید به صورت تک نفره انجام شود و پاسخ های ارائه شده باید نتیجه فعالیت فرد نویسنده باشد. در صورت مشاهده تقلب به همه افراد مشارکت کننده، نمره تمرین 100- و به استاد نیز گزارش می گردد. همچنین نوشته نشدن کدها توسط هوش مصنوعی نیز بررسی می شود!

موفق باشید