

مهلت تحویل: جمعه ۱۸ آبان ۱۴۰۳، ساعت ۲۳:۵۹

قسمت اول : الگوریتم ژنتیک

بهنام از بچگی آرزو داشت که یک هنرمند شود ولی در نهایت سر از مهندسی کامپیوتر در آورد. او که هنوز امید خود را از دست نداده است قصد دارد از الگوریتم ژنتیک یک هنرمند بسازد و در این راه از شما کمک خواسته است.



او تصمیم می گیرد که از مثلث ها استفاده کند و با کنار هم گذاشتن آن ها یک نقاشی زیبا خلق کند. مسئله به این صورت است که یک عکس به شما داده می شود و شما با استفاده از الگوریتم ژنتیک و با کنار هم گذاشتن مثلث ها باید کاری کنید که عکس ساخته شده تا حد امکان مشابه تصویر اصلی باشد.

پیاده سازی مسئله

بخش یک : تعریف مفاهیم

در ابتدای مسئله لازم است تعریف صحیح ژن در مسئله را پیدا کنید و با استفاده از آن ها کروموزوم را بسازید. هر کروموزوم یک پاسخ برای مسئله است و با انتخاب کردن کروموزوم ها به صورت تصادفی سعی می شود که فضای حالت بزرگ مسئله پوشش داده شود.

بخش دو: ساخت جمعیت اولیه

در ابتدا لازم است جمعیتی از کروموزوم ها را بسازید. برای این مسئله پیشنهاد می شود که حداکثر 100 کروموزوم داشته باشید چرا که پردازش ها و محاسبه فیتنس ها بر روی عکس انجام می شود و این موضوع می تواند باعث کند شدن کد شما شود. جمعیت شما باعث ایجاد تنوع در حالات شما می شود و به شما کمک می کند راه حل های مختلفی برای رسیدن به پاسخ پیدا کنید و از این رو اهمیت زیادی دارد. دقت کنید که انتخاب پارامتر جمعیت به خود شما بستگی دارد و در گزارش خود باید نمودار مقایسه روند افزایش فیتنس را در سه حالت با جمعیت های متفاوت بررسی کنید و نتایج را تحلیل کنید.

بخش سه: معیار سنجش سازگاری¹

برای اینکه بفهمید الگوریتم شما چه قدر خوب عمل می کند باید معیاری برای سنجش سازگاری پیدا کنید. در مسئله داده شده، معیار خوب بودن هر کروموزوم این است که به چه اندازه به عکس هدف نزدیک است. برای سنجش تفاوت دو عکس معیار های متفاوتی وجود دارد. چند تا از آن ها را در گزارش خود بیاورید و در نهایت از یکی از آنها به عنوان معیار سازگاری در الگوریتم خود استفاده کنید همچنین دلیل انتخاب خود را ذکر کنید.

بخش چهارم: پیاده سازی mutation، crossover و استراتژی انتخاب نسل بعد

حال برای اینکه به یک پاسخ از مسئله داده شده نزدیک شویم، نیاز است در هر نسل، جمعیت جدیدی با استفاده از جمعیت نسل قبل آن تولید گردد. برای این کار، باید از روش های crossover و mutation استفاده گردد. تابع crossover بر روی دو کروموزوم اعمال می شود، و آن ها را ترکیب می کند تا به کروموزوم هایی از ترکیب آن دو که در حالت ایده آل بهترین ویژگی های دو ژن اولیه را دارند برسد. این ترکیب و نرخ ایجاد آن باید به عنوان پارامترهای مسئله در نظر گرفته شوند. تابع mutation بر روی یک کروموزوم اعمال می شود، و آن را جهش و یا تغییر می دهد؛ به این امید که بتواند به کروموزوم بهتری جهش پیدا کند. می توانید درصد معقولی از ژن های برتر را نیز برای انتقال مستقیم به نسل های آینده در نظر بگیرید.

توجه داشته باشید که در این مسئله mutation می تواند تاثیر زیادی بر روی نتایج داشته باشد، بنابراین به میزان و درجه جهشی که اعمال می کنید توجه کنید.

¹ Fitness Evaluation

بخش پنجم: تحلیل نتایج

برای سنجش درستی الگوریتم می توانید از هر تصویر ورودی که دوست دارید استفاده کنید. در فایل های پروژه سه تصویر آماده برای شما آورده شده است که می توانید از آن ها استفاده کنید. اگر الگوریتم را به درستی پیاده سازی کرده باشید، بازسازی تصویر را با کیفیت خیلی بالا با توجه به پارامتر های انتخاب شده می تواند تا بیست دقیقه طول بکشد. از این رو برای اینکه از درستی کد خود مطمئن شوید پیشنهاد می شود که ابتدا با استفاده از تابع `resize` که در اختیار شما قرار داده شده است، سایز عکس ها را کوچک تر کنید تا بتوانید نتایج کار را سریعتر مشاهده کنید. همچنین رسم نمودار فیتنس در نسل های متفاوت و تحلیل آن می تواند کمک بسیاری به شما بکند.

نکات کمکی

- هر مثلث دارای یک رنگ است که هر رنگ چهار کانال دارند. سه کانال اول برای RGB و کانال چهارم نیز مربوط به `alpha` است که مقدار `transparency` را نشان می دهد. به این صورت که وقتی دو مثلث روی هم قرار بگیرند ترکیب رنگ RGB آن ها، رنگ آن قسمت مشترک را تعیین می کند و هر چه مقدار `alpha` بیشتر باشد وزن آن رنگ بیشتر خواهد بود.
- برای رنگ `background` تصویر همیشه مقدار `(0,0,0,255)` قرار میگیرد که سیاه است.
- هندسه هر مثلث با سه نقطه راس آن مشخص می شود.
- کد های پایه پروژه به شما داده شده است و شما باید قسمت های مربوط به تعریف `mutation`، `fitness`، `crossover` و سیاست مربوط به انتخاب نسل بعد را بنویسید.

سوالات

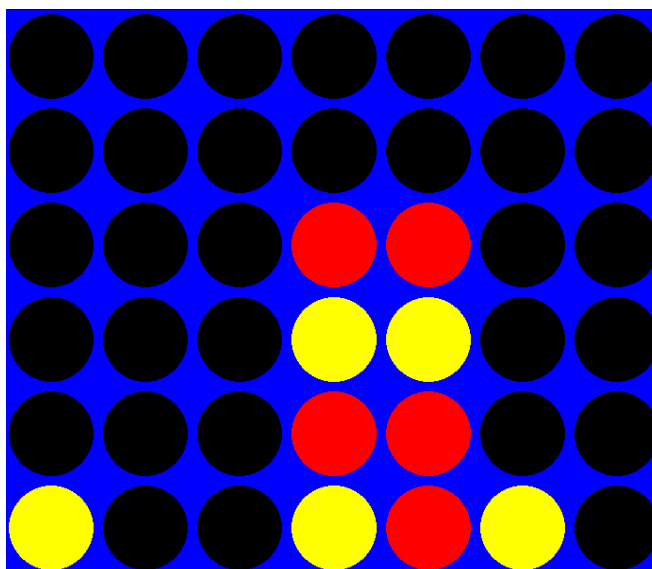
1. برای کروموزومی که در نظر گرفته اید فضای حالت آن را محاسبه کنید.
2. دو تا از ایده هایی که از نظر شما می تواند باعث سریع تر همگرا شدن این مسئله شود را توضیح دهید.
3. استراتژی های متفاوتی برای انتخاب نسل بعد در الگوریتم های ژنتیک وجود دارد. درباره دو مورد از آن ها توضیح دهید.

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 می‌بینید؟ بررسی کنید که عمق الگوریتم چه تاثیراتی بر روی شانس پیروزی، زمان و گره‌های دیده شده می‌گذارد.
2. آیا می‌توان ترتیب دیدن فرزندان هر نود را به گونه‌ای 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 می‌کند؟
4. توضیح دهید که چرا به هنگام هرس کردن الگوریتم بدون از دست دادن دقت خود سریع‌تر می‌شود.
5. چرا در حالتی که حریف به صورت شانسی عمل می‌کند (مانند این پروژه)، استفاده از minimax بهینه‌ترین روش نیست؟ چه الگوریتمی می‌تواند 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 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.
- 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 پاسخ‌های شما به سوالات کتبی، فایل `jupyter-notebook`، خروجی `html` و فایل‌های مورد نیاز برای اجرای آن باشد. از نمایش درست خروجی‌های 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