

## قسمت اول: ژنتیک

### مقدمه

در این قسمت، با روش‌هایی که برگرفته از طبیعت و انتخاب طبیعی هستند، آشنا می‌شویم. در این روش‌ها که به طور کلی الگوریتم‌های ژنتیک نامیده می‌شوند، ایده‌هایی برای مدل‌سازی جفت‌گیری، جهش و انتخاب طبیعی به کار گرفته می‌شود. در این گونه الگوریتم‌ها، ممکن است با انتخاب معیارهای ساده‌ی انتخاب طبیعی، نتایج مطلوب به دست نیاید و باید معیاری در نظر بگیریم که علاوه بر عملکرد فردی، به گوناگونی جمعیت نیز اهمیت دهد.

الگوریتم‌های ژنتیک عموماً در مسئله‌هایی با فضای حالت بزرگ کاربرد دارند؛ این الگوریتم‌ها نمونه گرفتن از جمعیت و ترکیب و تغییر افراد و ارزیابی آن‌ها انجام می‌دهند و سعی می‌کنند که نسل به نسل جواب‌ها را بهبود دهند تا به جواب مورد نظر برسند.

در این پروژه قصد داریم با استفاده از الگوریتم‌های ژنتیک، یکی از مسائل مربوط به بهینه‌سازی را پیاده‌سازی کنیم. هدف یک مسئله بهینه‌سازی، یافتن بهترین راه‌حل از مجموعه بزرگی از راه‌حل‌های ممکن است، جایی که «بهترین» راه‌حل با مجموعه‌ای از معیارها یا اهداف تعریف می‌شود. حل مسائل بهینه‌سازی می‌تواند به مزایای قابل توجهی مانند افزایش کارایی، کاهش هزینه‌ها، بهبود عملکرد و موارد دیگر منجر شود.

### توضیح مسئله

در این قسمت شما به یکی از مسائل مربوط به مدیریت پورتفولیو<sup>۱</sup> می‌پردازید. در این مسئله شما می‌خواهید در ابتدای سال ۲۰۲۳ در بازار سهام سرمایه‌گذاری کنید. به این منظور باید سرمایه‌ای که در اختیار دارید را به گونه‌ای روی هر کدام از دارایی‌های سهامی<sup>۲</sup> تقسیم کنید، که بیشترین بازده<sup>۳</sup> و کمترین ریسک<sup>۴</sup> را داشته باشید. به عبارت دیگر، لازم است ضرایبی به هر کدام از سهم‌هایی که در اختیار دارید اختصاص دهید تا بازدهی را بیشینه و ریسک را کمینه کند. این ضرایب نشان‌دهنده نسبت سرمایه شما است که بر روی هر دارایی سرمایه‌گذاری می‌شود، بنابراین جمع این ضرایب همواره باید یک باشد. برای پیدا کردن این ضرایب از داده‌های تاریخی بازار استفاده خواهید کرد.

<sup>۱</sup> Portfolio Management

<sup>۲</sup> Stock Assets

<sup>۳</sup> Return

<sup>۴</sup> Risk

برای این منظور یک دیتاست در اختیار شما گذاشته شده است که نشان‌دهنده میزان بازده و ریسک تعدادی از دارایی‌های سهامی از ابتدای سال ۲۰۲۰ تا انتهای سال ۲۰۲۲ می‌باشد. به طور مثال به اطلاعات سهام شرکت‌های زیر توجه کنید.

Ticker	Risk	Return
AAPL	0.2586	0.9482
GOOGL	0.4285	0.466
AMZN	0.5779	0.1
TSLA	0.7	5.12
MSFT	0.3	0.56
NVDA	0.62	2.27

در زمینه دارایی‌های سهام، Return به سود یا زیان سرمایه‌گذار در سرمایه‌گذاری خود اشاره دارد که به صورت درصدی از سرمایه‌گذاری اولیه بیان می‌شود. به عنوان مثال اگر شما سهام شرکت Tesla (با نماد TSLA) را در ابتدای سال ۲۰۲۰ خریداری می‌کردید، در انتهای ۲۰۲۲ سرمایه شما ۵۱۲ درصد (۵.۱۲ برابر) افزایش می‌یافت! همچنین Risk، بزرگترین درصد افت از اوج ارزش سهام به پایین‌ترین نقطه آن است. به طور مثال در بین سال‌های ۲۰۲۰ تا ۲۰۲۳، سهام شرکت Tesla، هفتاد درصد افت قیمت را تجربه کرده است.

در اینجا یک انتخاب از ضرایب می‌تواند به صورت زیر باشد:

AAPL: 0.1, GOOGL: 0, AMZN: 0, TSLA: 0.7, MSFT: 0, NVDA: 0.2

یا با توجه به اینکه TSLA عملکرد خیلی خوبی داشته است، یک حالت دیگر می‌تواند به صورت زیر باشد:

AAPL: 0, GOOGL: 0, AMZN: 0, TSLA: 1, MSFT: 0, NVDA: 0

اما سرمایه‌گذاری تمام پول خود در یک سهم، به دلایل مختلف، یک استراتژی سرمایه‌گذاری خوب در نظر گرفته نمی‌شود. به طور خلاصه این دلایل عبارتند از: عدم تنوع، ریسک بالاتر، نقد شوندگی پایین‌تر، پتانسیل محدود رشد و ... بنابراین در کنار بازدهی و ریسک، یک معیار دیگر می‌تواند **تعداد سهامی** باشد که شما در سبد خود قرار می‌دهید (در مثال بالا، سهم‌هایی که ضرایب مخالف صفر دارند).

توجه داشته باشید که مسائل بهینه‌سازی اغلب به عنوان یافتن حداکثر یا حداقل مقدار یک تابع معین، مشروط به محدودیت‌های خاصی تعریف می‌شوند. در برخی موارد، می‌توان یک راه‌حل صریح برای مسئله بهینه‌سازی پیدا کرد که تمام محدودیت‌ها را برآورده کند و مقدار بهینه تابع را ارائه دهد. با این حال، در بسیاری از موارد، یافتن چنین راه‌حلی ممکن نیست.

در مسئله تعریف شده پیدا کردن بهترین پاسخ (بهترین ضرایب برای سرمایه‌گذاری) ممکن است در یک مجموعه ضریب خاص خلاصه نشود و جواب واحدی برای این مسئله وجود نداشته باشد. به همین خاطر سعی می‌کنیم تا با بررسی بازده و ریسک نهایی و همچنین تعداد سهامی که در سبد قرار دارند، این حد را مشخص کنیم.

## پیاده‌سازی مسئله

### بخش یک: مشخص کردن مفاهیم اولیه

در الگوریتم‌های ژنتیک ابتدا باید یک تعریف برای ژن ارائه دهید و سپس با استفاده از آن، یک کروموزوم بسازید. هر کروموزوم مجموعه‌ای از ژن‌ها است و این مجموعه یا همان کروموزوم، یک راه پیشنهادی برای حل مسئله مورد نظر می‌باشد. توجه داشته باشید که در الگوریتم‌های ژنتیک باید اکثر کارها را با استفاده از تصادفی کردن وقایع انجام دهید، چرا که اگر فضای حالت بزرگ باشد پیدا کردن شرطی که همه‌ی محدودیت‌ها را برقرار سازد بسیار دشوار است. به همین دلیل، تعریف کروموزوم‌ها اهمیت ویژه‌ای دارد و باید به گونه‌ای باشد که امکان اعمال تابع تناسب و توابع دیگر بر روی آن فراهم باشد.

### بخش دو: تولید جمعیت اولیه

پس از تعریف و پیاده‌سازی کروموزوم‌ها، باید جمعیت اولیه‌ای از کروموزوم‌ها به صورت تصادفی بسازید. تعداد این جمعیت می‌تواند به عنوان یک پارامتر حل مسئله باشد و به انتخاب‌های شما بستگی دارد.

### بخش سه: پیاده‌سازی و مشخص کردن تابع معیار سازگاری<sup>5</sup>

بعد از تولید جمعیت اولیه، نیاز داریم تا تابع معیاری تعریف کنیم که بتواند برای شناسایی کروموزوم‌های برتر که شرایط و محدودیت‌های مسئله را بهتر مدل می‌کنند استفاده شود. ابتدا یک تعریف مناسب برای این تابع معیار ارائه دهید، و سپس آن را برای این مسئله پیاده‌سازی کرده، و میزان سازگاری جمعیت خود را بدست آورید.

### بخش چهار: پیاده‌سازی crossover و mutation و تولید نسل بعدی

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 روش‌های crossover و mutation استفاده گردد. تابع crossover بر روی دو کروموزوم اعمال می‌شود، و آن‌ها را ترکیب می‌کند تا به کروموزوم‌هایی از ترکیب آن دو که در حالت ایده آل بهترین ویژگی‌های دو ژن اولیه را دارند برسد. این ترکیب و نرخ ایجاد آن باید به عنوان پارامترهای مسئله در نظر گرفته شوند.

---

<sup>5</sup> Fitness Function

تابع `mutation` بر روی یک کروموزوم اعمال می‌شود، و آن را جهش و یا تغییر می‌دهد؛ به این امید که بتواند به کروموزوم بهتری جهش پیدا کند. می‌توانید درصد معقولی از ژن‌های برتر<sup>۶</sup> را نیز برای انتقال مستقیم به نسل‌های آینده در نظر بگیرید. توجه داشته باشید که در این مسئله `mutation` می‌تواند تاثیر زیادی بر روی نتایج داشته باشد، بنابراین به میزان و درجه جهشی که اعمال می‌کنید توجه کنید.

### بخش پنج: ایجاد الگوریتم ژنتیک روی مسئله

پس از انجام بخش‌های بالا، باید این توابع پیاده‌سازی شده را در یک الگوریتم استفاده کنید. توجه کنید که نیاز است هاپیر پارامترهایی برای میزان `randomness` و نحوه نزدیک شدن به پاسخ نهایی خود داشته باشید که با تغییر آن‌ها به جواب بهتری برسید.

### بخش شش: ارزیابی نتایج

در ادامه، برای ارزیابی الگوریتمی که پیاده‌سازی کرده‌اید، یک نمونه واقعی را مورد بررسی قرار دهید. این نمونه، در فایل `sample.csv` قرار داده شده است که شامل ۴۰۰ سهم می‌باشد. با اجرای الگوریتم خود روی این نمونه، ضرایب را به دست آورید. توجه داشته باشید که ضرایبی که به دست می‌آورید، باید شرایط زیر را داشته باشند:

(۱) بازده حداقل ۱۰۰۰ درصد باشد.

(۲) ریسک حداکثر ۶۰ درصد باشد.

(۳) تعداد سهامی که در سبد خود قرار می‌دهید، حداقل ۳۰ تا باشد.

یک مثال از ضرایب به دست آمده در فایل `sample_coeffs.csv` موجود می‌باشد.

### بخش هفت: سوالات

۱. جمعیت اولیه بسیار کم یا بسیار زیاد چه مشکلاتی را به وجود می‌آورند؟
۲. اگر تعداد جمعیت در هر دوره افزایش یابد، چه تاثیری روی دقت و سرعت الگوریتم می‌گذارد؟
۳. تاثیر هر یک از عملیات `crossover` و `mutation` را بیان و مقایسه کنید. آیا می‌توان فقط یکی از آنها را استفاده کرد؟ چرا؟
۴. به نظر شما چه راهکارهایی برای سریع‌تر به جواب رسیدن در این مسئله‌ی خاص وجود دارد؟
۵. با وجود استفاده از این روش‌ها، باز هم ممکن است که کروموزوم‌ها پس از چند مرحله دیگر تغییر نکنند. دلیل این اتفاق و مشکلاتی که به وجود می‌آورد را شرح دهید. برای حل آن چه پیشنهادی می‌دهید؟
۶. چه راه حلی برای تمام شدن برنامه در صورتی که مسئله جواب نداشته باشد پیشنهاد می‌دهید؟

\* جواب این سوالات را به صورت کامل در گزارش خود بنویسید.

---

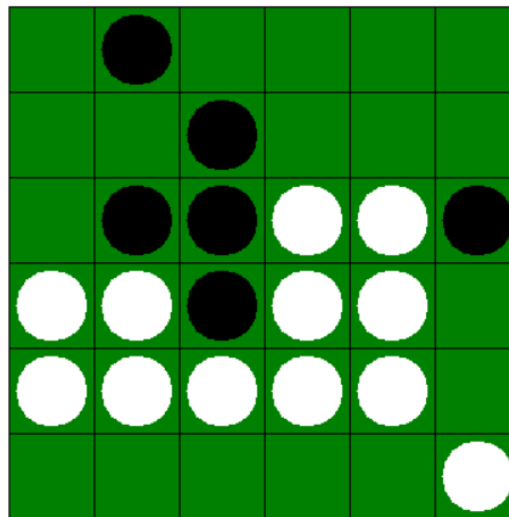
<sup>۶</sup> Elites

## قسمت دوم: بازی Othello

### مقدمه

شما و دوستان تصمیم گرفته‌اید که بازی Othello را بازی کنید ولی از آنجایی که شما به شدت تازه کار 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



برای شروع، چهار مهره در وسط صفحه به صورت ضربدری، قرار می‌گیرند. هر یک از دو بازیکن به نوبت یک حرکت انجام می‌دهند. مهره را جایی قرار دهید که یک یا چند مهره حریف را محاصره کند. انجام حرکت به معنی گذاشتن یک مهره (از طرف رنگ خود) در صفحه و محصور کردن یک یا چند مهره حریف در یک یا چند راستا است. در نتیجه مهره‌های محاصره شده را برگردانده و به رنگ مهره خود درآورید. این به معنی محاصره و تصاحب است. در واقع، مهره‌های حریف که در تصاحب مهره‌های شما (بین دو مهره شما در راستای ۸ گانه) قرار دارند، به رنگ مهره‌های شما در می‌آیند. هدف داشتن بیشترین مهره رنگ خود روی صفحه در پایان بازی است. لازم به ذکر است که هیچ یک از ۲ بازیکن نمی‌تواند حرکتی انجام دهد که هیچ کدام از مهره‌های حریف را برنگرداند. در صورتی که همچین حرکتی وجود نداشت، نوبت به حریف داده می‌شود.

در این بازی شما همیشه مهره‌های سفید را بازی می‌کنید و شروع کننده بازی به صورت شانسی انتخاب می‌شود و اندازه صفحه بازی نیز  $6 \times 6$  است. می‌توانید فرض کنید که کامپیوتر نیز مانند شما بهینه عمل می‌کند. جهت آشنایی بیشتر با فرایند بازی می‌توانید به این [لینک](#) مراجعه کنید.

## پیاده‌سازی

هدف شما پیاده‌سازی الگوریتم minimax برای شکست دادن دوستان است. کد بازی در فایل main.py به شما داده شده است اما این کد کامل نیست و شما باید بخش‌های 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 (اضافه کردن مواردی مثل getter و setter مانعی ندارد اما باید در گزارش کار ذکر کنید).

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

برای استفاده از کد کافیسیت یک نمونه از کلاس Othello با توجه به آرگومان‌های مد نظر بسازید و پس از پیاده‌سازی تابع get\_human\_move 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 True کنید.

## بررسی نتایج

برای درک کامل الگوریتم کد را با عمق‌های ۱، ۳ و ۵ بدون هرس ۱۰۰ الی ۲۰۰ 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 خود، هرس آلفا و بتا را به کد اضافه کنید و موارد ذکر شده در بخش قبل را برای عمق‌های یاد شده مجدداً بررسی کنید. همچنین عمق ۷ را به عمق‌های مورد بررسی خود اضافه کنید. \* توجه کنید که برای گرفتن نمره‌ی کامل الگوریتم شما باید در عمق ۵، ۹۰ درصد و در عمق ۷، ۹۵ درصد شانس برد داشته باشد.

## سوالات

۱. چه مواردی را به هنگام محاسبه‌ی هیوریستیک خود در نظر گرفته‌اید و چرا؟
۲. آیا میان عمق الگوریتم و پارامترهای حساب شده روابطی می‌بینید؟ به طور کامل بررسی کنید که عمق الگوریتم چه تاثیری بر روی شانس پیروزی، زمان و گره‌های دیده شده می‌گذارد.
۳. 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.
۴. Branching Factor را توضیح دهید و بگویید که با پیشرفت این بازی چه تغییری می‌کند؟
۵. توضیح دهید که چرا به هنگام هرس کردن الگوریتم بدون از دست دادن دقت خود سریع‌تر می‌شود.
۶. 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؟ به طور کامل توضیح دهید.

\* جواب این سوالات را به صورت کامل در گزارش خود بنویسید.

## نکات پایانی

- تمامی نتایج باید در یک فایل فشرده با عنوان zip <#SID>-AI-CA2 تحویل داده شود. این فایل باید شامل موارد زیر باشد:
  - یک پوشه به نام code شامل کدهای تمام قسمت‌هایی از تمرین که پیاده‌سازی کرده‌اید.
  - گزارش پروژه با فرمت PDF و شامل شرح تمامی کارهای انجام شده، نتایج به دست آمده و تحلیل‌ها و بررسی‌های خواسته شده در صورت پروژه.
  - در صورتی که از Notebook Jupyter استفاده می‌کنید نیازی به ارسال جداگانه کدها و گزارش نیست و هر دو را می‌توانید در یک فایل Notebook قرار دهید. حتما خروجی html فایل Notebook خود را نیز همراه فایل Notebook ارسال کنید.
- توجه داشته باشید علاوه بر ارسال فایل‌های پروژه، این پروژه تحویل نیز گرفته خواهد شد. بنابراین لازم است بر تمامی قسمت‌های کدتان تسلط کافی را داشته باشید و تمام بخش‌های پروژه باید قابلیت اجرای مجدد در زمان تحویل را داشته باشند. همچنین در صورت عدم حضور در زمان تحویل، نمره‌ای دریافت نخواهید کرد.
- هیچگونه شباهتی در انجام این پروژه بین افراد مختلف پذیرفته نمی‌شود. در صورت کشف هرگونه تقلب برای همه افراد متقلب نمره ۱۰۰- در نظر گرفته می‌شود.
- استفاده از مراجع با ارجاع به آنها بلامانع است. اما در صورتی که گزارش شما ترجمه عینی از آن‌ها باشد یا از گزارش افراد دیگر استفاده کرده باشید کار شما تقلب محسوب می‌شود.
- در صورتی که سوالی در مورد پروژه داشتید بهتر است در فروم درس مطرح کنید تا بقیه از آن استفاده کنند، در غیر این صورت به طراحان پروژه ایمیل بزنید و از یکی از آنها بپرسید.

موفق باشید.