

به نام خدا



دانشکده مهندسی کامپیوتر

هوش مصنوعی و سیستم های خبره

پروژه سوم (ژنتیک)

دکتر آرش عبدی

پاییز ۱۴۰۲

طراحان:

بهاره کاوسی نژاد

فرناز خوش دوست آزاد

مهدی اوشنی



- در صورت وجود هر گونه ابهام در سوالات تنها به طراح آن سوال پیام دهید.
 - با توجه به تنظیم شدن ددلاین تمارین توسط خود شما امکان تمدید وجود ندارد اما میتوانید از 10 روز تاخیر خود تا 5 روز پس از اتمام ددلاین استفاده نمایید.
 - خوانا و مرتب بنویسید.
 - برای این پروژه شما باید حداقل یک صفحه یه داک گزارش همراهش ارسال کنید اما قرار نیست که توی این داک کدتون رو توضیح بدید!
- بلکه قراره توش روندی که طی انجام پروژه طی کردید و چالش هایی که باهاش برخوردید (از لحظه ای که داک رو باز کردید، تا سرچ کردن داخل یوتوب، تا باگ خاصی که موقع ران برخوردید و ...) رو به صورت خلاصه شرح بدید تا ما با روند انجام کار شما و مدل فکریتون آشنا بشیم

آیدی تلگرام طراحان :

@HelenAzaad

@Bahareh_0281

@thegreatguilty



پروژه سوم هوش مصنوعی و سیستم های خبره: ژنتیک

هدف پروژه: پیاده سازی الگوریتم ژنتیک

شرح پروژه:

قسمت اول: خوشه بندی

مقدمه

در این تمرین هدف clustering دیتاست Iris با استفاده از genetic algorithm می باشد. در ادامه الگوریتم گام به گام توضیح داده شده است.

توضیح الگوریتم

1. Initialize a population

فرض کنید n دیتا پوینت داریم در این روش هر کروموزوم یک ارایه با سایز $n+1$ می باشد که به صورت زیر ساخته می شود. عدد k بیانگر حداکثر تعداد دسته ها برای دسته بندی دیتا می باشد که آن را به دلخواه براساس زمانی که الگوریتم طول میکشد و بهینه کار کردن آن انتخاب کنید. فرض کنید k دسته داریم در این صورت هر یک از دیتا پوینت ها به یکی از دسته ها ۱ تا k متعلق است خانه i ام در کروموزوم بیانگر این است که دیتا پوینت i به کدام دسته تعلق دارد همچنین خانه $n+1$ نیز بیانگر تعداد دسته ها در این کروموزوم می باشد.

پس درواقع هر کروموزوم یک دسته بندی از داده ها را با استفاده از حداکثر k دسته بیان میکند. مثال:

Genotype1: 22345123453321454552 5

در مثال بالا ۵ دیتا پوینت 1,2,7,13,20 به یک دسته تعلق دارند و لیبل آن دسته دو می باشد. هم چنین کلاستر یا دسته ای که لیبل آن ۱ می باشد شامل دیتا پوینت ها 6,14 می باشد.

2. Operators

برای انجام اعمال Crossover و Mutation میتوانید روش های مختلفی را اتخاذ کنید و براساس مساله روش جدید نیز بسازید.

مثال هایی برای Mutation :

- یکی از دیتا پوینت ها را انتخاب کرده و لیبل کلاسی که به آن تعلق دارد را به صورت رندوم با لیبل دیگری عوض کنید.



- یکی از دیتا پوینت ها را انتخاب کرده و لیبل کلاسی که به آن تعلق دارد را به لیبل نزدیک ترین کلاس به آن دیتا پوینت تعویض کنید.
 - یکی از کلاستر ها را انتخاب کنید و نزدیک ترین کلاستر به آن براساس centeroid ها را پیدا کرده و سپس لیبل همه ی اعضا این کلاستر را با لیبل نزدیک ترین کلاستر عوض کنید.
 - یک کلاستر را به دو کلاستر تقسیم کنید که این کار میتواند براساس فاصله از centeroid در آن کلاستر باشد.
- با استفاده از ترکیب روش های بالا و ایده خودتان روش جدید بسازید.

مثال هایی برای Crossover :

Genotype6: 2|2222|111113333344444

Genotype7: 4|4444|333335555511111

Child1: 2 4444 111113333344444

Child2: 4 2222 333335555511111

- دو کروموزوم A و B را در نظر بگیرید:

A – 1123245125432533424

B – 1212332124423221321

تعداد دلخواهی از کلاستر های A را در نظر بگیرید برای مثال کلاستر ۱ و ۲ دیتا پوینت ها این دو کلاستر در B در کلاسترهای ۱ و ۲ و ۳ قرار گرفته اند (برای مثال در چهار قرار نگرفته اند) حال کلاستر ۱ و ۲ را به ۳ کلاستر مانند B تقسیم کنید.

Objective function

میتوانید تابع را براساس معیار های مختلف و همچنین استفاده از معیار های خوب بودن کلاستر ها بسازید.

مثال:

هدف تابع میتواند minimize کردن فاصله اقلیدوسی بین نقاطی که در یک گروه هستند باشد و یا میتواند مانند الگوریتم k-means استفاده از centerio d ها هر کلاستر تابع را تعریف کرد.

Dataset

الگوریتم را با استفاده از دیتاست Iris که در این [لینک](#) در دسترس است اجرا کنید. (همچنین میتوانید از کتابخانه scikit-learn به آن دسترسی داشته باشید).

این دیتاست شامل ۴ ویژگی عددی می باشد. فاصله دو دیتاپوینت را با استفاده از فاصله اقلیدوسی این ویژگی ها محاسبه کنید.

معیار های ارزیابی

- درک درست از Evolutionary Algorithms و اجرا گام به گام این الگوریتم.
- درک نحوه حل مساله خوشه بندی با استفاده از این الگوریتم.
- پیاده سازی توابع برای هر مرحله از الگوریتم و توضیح آنها.
- مقایسه کوتاهی از الگوریتم k-means و خوشه بندی با استفاده از الگوریتم های تکاملی با روش بالا بر اساس نظر خودتان.
- از آنجایی که درصد درستی خوشه بندی به پیچیده تر و معنا دار بودن تعریف operatorها بستگی دارد، درصد درستی خوشه بندی هدف و معیار این تمرین نمی باشد.

قسمت دوم: گراف

مسئله Traveling Salesman Problem با الگوریتم ژنتیک

توضیح کلی:

مسئله Traveling Salesman Problem (TSP) یکی از مشهورترین و پرکاربردترین مسائل بهینه سازی در علوم کامپیوتر است. در این مسئله، فرض می شود که یک فروشنده دارای یک لیست از شهرها است و می خواهد به هر شهر دقیقاً یک بار سفر کند و سپس به شهر اصلی بازگردد. هدف این مسئله پیدا کردن کوتاه ترین مسیر ممکن است که فروشنده باید طی کند.

توضیح پروژه:

در این پروژه، شما باید الگوریتم ژنتیک را برای حل مسئله TSP پیاده سازی کنید.

مراحل اجرا:

برای پیاده سازی الگوریتم ژنتیک برای مسئله TSP، شما می توانید مراحل زیر را دنبال کنید:

1. تعریف فرد و جمعیت: یک فرد را به عنوان یک مسیر ممکن بین شهرها تعریف کنید. سپس جمعیت را به عنوان مجموعه ای از این افراد در نظر بگیرید.
2. تابع ارزیابی: یک تابع ارزیابی برای اندازه گیری کیفیت هر فرد (مسیر) نیاز است. در این مسئله، می توانید مسافت کل مسیر را به عنوان معیار کیفیت استفاده کنید. مسیرهای کوتاه تر باید امتیاز بیشتری دریافت کنند.
3. انتخاب والدین: از یک روش انتخاب والدین مانند روش رولت های انتخابی (roulette wheel selection) استفاده کنید تا فردی که برای تولید نسل بعدی استفاده می شود را انتخاب کنید. احتمال انتخاب هر فرد باید بر اساس ارزش فیتنس (fitness) آن فرد محاسبه شود.
4. تولید نسل بعدی: با استفاده از عملیات ترکیب (crossover) و جهش (mutation)، نسل بعدی را تولید کنید. برای عملیات ترکیب، می توانید از روش های مختلفی مانند ترکیب یک نقطه ای (single-point crossover)، ترکیب دو نقطه ای (two-point crossover) یا ترکیب یکنواخت (uniform crossover) استفاده کنید. برای عملیات جهش، می توانید به طور تصادفی نقاطی از مسیر را جابجا کنید.
5. تکامل نسل ها: با تکرار مراحل 3 و 4، نسل های جدید را تولید کنید تا به جواب بهینه نزدیک شوید. به عنوان معیار پایانی، می توانید تعداد نسل های تولید شده یا تغییرات اندازه ی بهترین فرد را در نظر بگیرید.
6. نمایش و نتیجه گیری: بهترین مسیری که در هر نسل به دست آمده است را نمایش دهید و نتیجه گیری کنید. همچنین، می توانید مسیر را روی نقشه رسم کنید تا بهبودهای مراحل مختلف الگوریتم را مشاهده کنید.

مثال ورودی:

فرض کنید 5 نود داریم و ماتریس فاصله بین آنها به صورت زیر است (ورودی به صورت یک ماتریس داده می شود که هر عنصر نشان دهنده فاصله بین دو نود گراف است):

16	21	20	29	0
28	19	15	0	29
25	13	0	15	20
17	0	13	19	21
0	17	25	28	16

خروجی مورد انتظار:

خروجی شامل مسیر بهینه ای است که ما باید طی کنید تا کمترین فاصله کل را داشته باشد. همچنین باید کل فاصله طی شده را نیز نمایش دهید. فرض کنید الگوریتم بهینه سازی مسیر زیر را یافته است:



مسیر: $1 < 3 < 4 < 5 < 2 < 1$

طول کل مسیر: 69

موارد قابل توجه:

- برای اجرای الگوریتم ژنتیک برای مسئله TSP، باید ماتریس فاصله بین هر دو شهر را تعریف کنید.
- برای افزایش کارایی الگوریتم، می توانید از روش های بهبودی مانند الگوریتم های محلی جستجو (local search algorithms) یا الگوریتم های ترکیبی استفاده کنید.
- می توانید پارامترهای الگوریتم ژنتیک مانند اندازه جمعیت، تعداد نسل ها، احتمال ترکیب و جهش را تغییر دهید تا بهترین نتیجه را بدست آورید.