



بسمه تعالی
دانشکده مهندسی مکانیک
پردیس دانشکده های فنی
دانشگاه تهران



هوش مصنوعی

پروژه 2

بازی و ژنتیک

دانشجو:

شهریار نامداری

810098043

استاد: دکتر فدایی

فروردین 1401

ژنتیک:

بخش صفر:

فایل txt داده شده خوانده شد و فقط حروف الفبا + فاصله و کاراکتر " ' " نگه داشته شدند. سپس کلمات stop word از این مجموعه حذف شده و سپس در یک set کلمات با جدا کننده فاصله ذخیره شدند. همه حروف نیز به uppercase تبدیل شدند.

بخش یک:

کروموزوم تولید شده توسط ما همان کلید میباشد. یک کلید 14 حرفی که باید به اندازه طول حروف موجود در فایل رمزگذاری شده پشت سر هم قرار داده شود. قاعدتا ژن های ما نیز حروف خواهند بود که تمام حروف بزرگ الفبای انگلیسی را میتوانند شامل شوند. برای کروموزوم های اولیه به صورت رندوم این حروف انتخاب میشوند.

بخش دو:

برای شروع تعدادی ازین کروموزوم ها به عنوان جمعیت اولیه میسازیم. این مقدار را تغییر میدهیم و نتایج را ثبت میکنیم تا جمعیت اولیه بهینه را بیابیم.

بخش سه:

تابع معیار سازگاری ساخته شد بدین صورت که می شمارد چند حرف decode شده در دیکشنری موجود است.

بخش چهار:

توابع crossover و mutation ساخته شدند بدین صورت که در crossover یک جدا کننده داریم و در mutation با احتمال کمی حروف رندوم با حروف الفبا جایگذاری میشوند. مقدار Pc برابر با 0.85 و مقدار Pm برابر با 0.05 است.

بخش پنجم:

تابع ژنتیک در قالب یک کلاس پیاده سازی شد و جزئیات در فایل notebook قابل مشاهده است.

بخش ششم:

1. جمعیت پایین باعث میشود در هر مرحله تغییر کمی رخ دهد و در نتیجه زمان زیادی طول بکشد تا به نتیجه برسیم. جمعیت بالا نیز باعث میشود که حجم محاسبات به قدری بالا باشد که باز از نظر محاسبات سنگین شود و زمان زیادی صرف شود.

همچنین جمعیت خیلی پایین موجب میشود که انعطاف پذیری جمعیت پایین آید و نتواند از یک جایی به بعد بهتر شود و ممکن است پسرفت نیز کند!

نتیجه جمعیت اولیه 10 عدد:

روی لوپ بینهایت میفتد و گاهی درصد کاهش نیز میابد!

نتیجه جمعیت اولیه 20 عدد:

Duration: 173.27326083183289 s

Answer: ALBERTEINSTEIN

نتیجه جمعیت اولیه 30 عدد:

Duration: 182.97021317481995 s

Answer: ALBERTEINSTEIN

نتیجه جمعیت اولیه 40 عدد:

Duration: 94.4152729511261 s

Answer: ALBERTEINSTEIN

نتیجه جمعیت اولیه 50 عدد:

Duration: 280.9479146003723 s

Answer: ALBERTEINSTEIN

نتیجه جمعیت اولیه 60 عدد:

Duration: 199.32779335975647 s

Answer: ALBERTEINSTEIN

انگار برای این مساله جمعیت اولیه 40 بهینه می باشد.

2. اگر در هر محله جمعیت مورد بررسی افزایش یابد محاسبات کند و کند تر میشود و نتیجه آنچنان بهینه ای نیز نخواهد داشت. همانند سوال اول که دیدیم با افزایش جمعیت اولیه مدت زمان حل مساله افزایش یافت.

3. اگر از mutation استفاده نکنیم باعث میشود که از قابلیت هایی که در جمعیت موجود ما وجود ندارد استفاده نکنیم و در نتیجه تنها از قابلیت جمعیت اولیه بهره بگیریم که ممکن است ناقص باشد! اگر از crossover استفاده نکنیم باعث میشود که از تغییرات بزرگ و ظرفیت های موجود در والدین برتر استفاده نکنیم و فرزندی تولید کنیم که تقریباً به شکل رندوم تولید میشوند که سرعت رشد بسیار پایینی خواهیم داشت.

4. Crossover – زیرا با هر بار crossover ژن هایی که برتر بوده اند ترکیب شده و تغییر زیادی حاصل میشود که بخش بزرگی از نتیجه حاصل شده از والدین گرفته شده. اما در mutation تغییرات رندوم و کوچک بوده و به همین دلیل فرزند حاصل شده پیشرفت چندانی زیادی نمیکند.

5. گاهی در شرایطی مثل جمعیت اولیه 50 که بالاتر امتحان شد، ممکن است چندین کروموزوم گیر کنند و تغییری نکنند که در هم میتوان زمان زیادی را منتظر ماند تا به طور شانس تغییر حاصل

شود و هم میتوان یک یا دو کروموزوم برتر را حذف کرد و آموزش را ادامه داد. مانند شخصی که از همه نظر عالی است و فقط یک نقص دارد اما به دلیل عالی بودن از همه جهت به سختی میشود آن تک نقصش را اصلاح کرد 😊

6. Crossover – همانطور که در سوال 4 هم بیان شد در تغییرات به کمک crossover تغییرات شدید و خوبی حاصل میشود که از والدین برتر گرفته شده‌اند. در صورتی که در mutation شانس حرف اول را میزند و تغییرات اتفاقی هستند.

7. میتوان تابع fitness را قوی تر نوشت که میتواند بسیار موثر باشد. برای مثال به جای اینکه ببینیم چند کلمه درست موجود است ببینیم چند بخش از کلمات درست موجود است که نتیجه بهتری حاصل میشود. البته نوشتن چنین تابعی مقداری دشوار تر خواهد بود.

کلید یافت شده نهایی: ALBERTEINSTEIN می‌باشد.