

به نام خدا

دانشکده‌ی مهندسی برق و کامپیوتر

درس هوش مصنوعی

تمرین کامپیوتری شماره ۲: ژنتیک

استاد: دکتر فدایی

علی پاکدل صمدی

۸۱۰۱۹۸۳۶۸

هدف پروژه:

هدف از انجام این پروژه، پیاده‌سازی الگوریتم ژنتیک می‌باشد، همچنین آشنایی با ایده‌هایی برای مدل‌سازی جفت‌گیری، جهش و انتخاب طبیعی می‌باشد. با استفاده از الگوریتم ژنتیک و مدل‌سازی‌های مختلف، می‌خواهیم کلید یک متن رمزنگاری شده را پیدا کرده و متن را به حالت اصلی برگردانیم.

توضیح کلی پروژه:

یک فایل `global_text` داده شده است تا با استفاده از تمیز کردن این فایل، بتوانیم یک لغت‌نامه استخراج کنیم و سپس در ادامه برای محاسبه امتیاز تناسب کروموزوم‌ها از این لغت‌نامه استفاده می‌کنیم. همچنین یک فایل `encoded_text` در اختیار داریم که یک متن رمزنگاری شده است که قرار است در ادامه این متن را رمزگشایی بکنیم. ابتدا ما تعدادی کلید به صورت تصادفی می‌سازیم. حال با لغت‌نامه خود به این کلیدها امتیاز می‌دهیم. سپس با `mutation` و `crossover` امتیاز تناسب این کروموزوم‌ها را افزایش داده و در نهایت کلید اصلی پیدا شود.

بخش صفر: تمیز کردن داده‌ها و ایجاد لغت‌نامه

روند تولید لغت‌نامه به این صورت است که کاراکتر به کاراکتر از فایل خوانده و هر جا یک حرف انگلیسی دیدیم، به یک استرینگ اضافه می‌کنیم. هر جا به کاراکتری به غیر از حروف رسیدیم، ابتدا تمام حروف این استرینگ ساخته شده را به `Upper Case` تبدیل کرده و به یک `set` اضافه می‌کنیم. دوباره این استرینگ را خالی کرده و از ادامه متن ادامه می‌دهیم. این `set` ساخته شده لغت‌نامه ما می‌باشد.

بخش یک: مشخص کردن مفاهیم اولیه

ژن‌های ما در این مثال حروف انگلیسی می‌باشد و کروموزوم‌ها از کنار هم قرار گرفتن این ژن‌ها تشکیل می‌شوند. کروموزوم‌ها در این مثال کلیدها می‌باشند.

بخش ششم: سوالات

۱. اگر جمعیت اولیه بسیار کم باشد، باعث می‌شود تنوع خوبی در کروموزوم‌ها نداشته باشیم و اگر تابع mutation ما نرخ تغییر کمی داشته باشد، باعث می‌شود رسیدن به جواب سخت و طولانی شود. بنابراین باید نرخ تابع mutation را تغییر داد که این نیز مشکلات خودش را دارد. حال اگر جمعیت اولیه ما بسیار زیاد باشد، باعث می‌شود عملیات‌های mutation و crossover انجام شده روی کروموزوم‌ها طولانی شود و در نتیجه زمان زیادی برای رسیدن به جواب نیاز خواهیم داشت.

۲. اگر در هر دوره جمعیت ما افزایش یابد، در قسمتی از برنامه جمعیت آنقدر زیاد خواهد شد که زمان زیادی برای عملیات‌های برنامه خواهد گرفته شد و عملاً الگوریتم بهینه نخواهد بود. همچنین شاید با توجه به زیادی جمعیت تنوع زیادی نیز بین کروموزوم‌ها ایجاد شود ولی لزوماً تنوع کافی نخواهد بود. همچنین افزایش جمعیت باعث می‌شود حافظه بیشتری نیاز داشته باشیم که این نیز مشکل‌ساز می‌باشد.

۳. اگر در الگوریتم‌های ژنتیک از mutation استفاده نشود، احتمال بالایی وجود دارد که جمعیت زیادی از کروموزوم‌ها بسیار شبیه به یکدیگر و حتی یکسان شوند و به نوعی تنوع از بین خواهد رفت، به همین دلیل mutation لازم است در این الگوریتم‌ها وجود داشته باشد. حال mutation به تنهایی کروموزوم‌ها را بسیار کم تغییر می‌دهد ولی crossover می‌تواند تغییر بیشتری ایجاد کند به همین دلیل crossover سرعت اجرای الگوریتم ژنتیک را افزایش می‌دهد. Mutation باعث ایجاد یک ترکیب جدید در ژن‌ها می‌شود ولی crossover بهترین ترکیب‌هایی که موجود است را تغییر می‌دهد و تغییر جدیدی در ژن‌ها ایجاد نمی‌شود و جای ژن‌ها عوض می‌شود. اگر از mutation استفاده نشود، تنها به بهترین ترکیب ممکن از ژن‌های موجود خواهیم رسید و شاید به جواب ممکن نرسیم.

۴. عملیات mutation باعث افزایش دقت می‌شود و تنوع خوبی به کروموزوم‌ها می‌بخشد. اما از آنجایی که معمولاً تغییرات ژن در crossover بیشتر است، باعث افزایش سرعت در الگوریتم ژنتیک می‌شود.

۵. می‌توان نرخ تغییر mutation یا همان احتمال تغییر ژن در تابع mutation را افزایش داد تا تغییرات در نوع ژن‌ها بیشتر شود.

۶. اگر تنها از crossover استفاده شود، ما تنها به یک سری کروموزوم که از یک نژاد هستند خواهیم داشت و این باعث می‌شود در اکثر مواقع به جواب نرسیم. زیرا تغییر جدیدی در کروموزوم‌ها ایجاد نمی‌شود و تنها جفت‌گیری بین دو کروموزوم صورت می‌گیرد و بچه این کروموزوم‌ها ویژگی‌های والد خود را دارد. حال اگر از mutation استفاده شود، همواره در حال ایجاد کروموزوم‌های متنوع می‌باشیم و به احتمال زیاد به جواب خواهیم رسید و تنها مشکل این روش، زمان زیادی است که باید الگوریتم اجرا شود. بنابراین اگر تنها از crossover استفاده شود به احتمال بالا به جواب نخواهیم رسید و اگر از mutation استفاده شود، زمان زیادی باید صرف کنیم ولی احتمالاً به جواب خواهیم رسید. بنابراین mutation موثرتر می‌باشد. (البته بستگی به الگوریتم‌هایی که در crossover و mutation پیاده‌سازی شده است دارد، در بعضی مواقع crossover به تنهایی می‌تواند موثر باشد)

۷. می‌توان هنگامی که تعداد خاصی از کروموزوم‌ها به بالاتر از امتیاز تناسب خاصی رسیدند، باقی کروموزوم‌های ضعیف را از مجموعه جمعیت کم کنیم. همچنین هنگام ایجاد تغییرات در mutation و یا crossover، چک کنیم که با این تغییر امتیاز بالاتر رفته یا خیر، و اگر بالاتر نرفته بود، این تغییرات در کروموزوم‌ها ثبت نشود و تنها کروموزوم‌هایی که بهبود یافته‌اند تغییر کنند.

نتیجه‌گیری کلی:

در این پروژه پیاده‌سازی الگوریتم ژنتیک را به همراه عملیات‌های crossover و mutation فرا گرفتیم و همچنین فهمیدیم برای الگوریتم‌های ژنتیک باید جمعیت اولیه‌ای که می‌سازیم تعداد مناسبی داشته باشید و کم و زیاد بودن آن مشکلاتی به همراه دارد. همچنین برای رسیدن به جواب و همچنین بهینه‌تر کردن از نظر زمانی، باید هر دو عملیات crossover و mutation اجرا شوند. الگوریتم‌های ژنتیک برای یافتن راه‌حل‌های با کیفیت برای مسائل بهینه‌سازی و جستجو با تکیه بر بیولوژیک و الهام از عملیات‌های جهش، جفت‌گیری و انتخاب طبیعی می‌باشد و معمولاً مفهوم نسبتاً ساده‌ای دارد. همچنین فهمیدیم که عملیات mutation لازم است ولی عملیات crossover بیشتر به سرعت الگوریتم کمک می‌کند.

منابع استفاده شده:

<https://stackoverflow.com>

<https://www.geeksforgeeks.org>

<https://www.w3schools.com>