سامان اسلامی نظری
۸۱۰۱۹۹۳۷۵

به نام خدا

تمرین ژنتیک
درس هوش مصنوعی

۱. جمعیت اولیه بسیار کم یا بسیار زیاد چه مشکلاتی را به وجود می‌آورند؟

استفاده از جمعیت کوچک می‌توند موجب تنوع کم کروموزم‌ها و در نهایت ناموفق ماندن الگوریتم شود. استفاده از جمعیت زیاد نیز می‌توند بسیار هزینه‌بر بوده و نیازمند قدرت محاسباتی بیش‌تری می‌باشد. بنابراین باید اندازه مناسبی را برای جمعیت اولیه بدست آورد.

۲. اگر تعداد جمعیت در هر دوره افزایش یابد، چه تاثیری روی دقت و سرعت الگوریتم می‌گذارد؟

با افزایش جمعیت، دقت جواب نهایی نیز افزایش می‌یابد؛ اما همانطور که در بخش قبل نیز گفته شد، جمعیت زیاد می‌تواند محاسبات سنگینی را در پی داشته باشد. بنابراین نکته منفی این کار سنگینی محاسبات آن و در نتیجه طولانی‌تر شدن زمان پایان یافتن الگوریتم است.

۳. تاثیر هر یک از عملیات‌های crossover و mutation را بیان و مقایسه کنید. آیا می‌توان فقط یکی از آن‌ها را استفاده کرد؟ چرا؟

عملیات crossover با ترکیب کروموزم‌ها سعی در ایجاد نسل بعدی و یافتن کروموزوم جواب دارد. از آنجا در هر مرحله از ساخت نسل جدید، از بهترین کروموزم‌ها بهره می‌بریم، این عملیات در هر مرحله باعث پیشرفت کروموزم‌ها می‌شود. بنابراین می‌توان گفت crossover در ایجاد نسل بهتر قدم برمی‌دارد.

عملیات mutation سعی در شانسی‌تر کردن الگوریتم داشته و با این امید که با استفاده از شانسی تغییر دادن ژن‌ها ممکن است به یک کروموزم بهتر برسیم، کارش را انجام می‌دهد. علاوه بر آن mutation باعث می‌شود که الگوریتم سریعا به یک نسل خاص گرایش نیابد. گاهی ممکن است همه ویژگی‌ها در جمعیت اولیه ظاهر نشوند؛ در اینجا وظیفه mutation این است که این ویژگی‌هایی که در نظر گرفته نشدند را، با یک توزیع احتمالی، بعدا به جمعیت جدید اضافه کند.

استفاده تنها از crossover باعث می‌شود خیلی زود نسل کروموزوم‌ها به نقطه‌ای ثابت برسد. علاوه بر آن اگر در جمعیت اولیه ویژگی خاصی را در ژن‌ها در نظر نگرفته باشیم، با crossover خالی آن ویژگی هیچگاه ایجاد نخواهد شد.
استفاده تنها از mutation نیز باعث پیشرفت جمعیت و در نهایت رسیدن به جواب نخواهد شد. چرا که mutation هدفی را برای جمعیت دنبال نمی‌کند.

۴. به نظر شما چه راهکارهایی برای سریع‌تر به جواب رسیدن در این مسئله خاص وجود دارد؟

جواب اول و واضح مسئله بهینه‌سازی توابع داخلی مسئله است. تعیین پارامترهای مناسب برای حل مسئله (نظیر جمعیت اولیه، نرخ جهش و دیگر پارامترهای کلاس HyperParameters) تاثیر بسزایی در سرعت رسیدن به جواب خواهد داشت.

۵. با استفاده از این روش‌ها، باز هم ممکن است که کروموزم‌ها 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؟

گاها ممکن است الگوریتم در یک بیشینه محلی گیر کند. جهش (Mutation) برای فرار از همین موقعیت‌ها به درد بخور است. از آنجا که جهش شانسی است، برای آن که بتوان حالت‌های دیگری را نیز امتحان کرد، می‌توان الگوریتم را چند بار، در صورت گیر کردن، اجرا کرد تا با استفاده از عملکرد متفاوت تابع جهش، نتایج مختلف را بدست آورد.

۶. چه راه حلی برای تمان شدن برنامه در صورتی که مسئله جواب نداشته باشد پیشنهاد می‌دهید؟

می‌توان برای تعداد نسل‌ها و تعداد باری که الگوریتم اجرا کرد محدودیت قرار داد. از محدودیت زمانی نیز می‌توان استفاده کرد.