

باسمه تعالی

موضوع :

PSO و حل مسئله ی کوله پشتی

پاییز ۱۴۰۳ - ترم مهر ماه
هوش مصنوعی - دکتر مهدی مصلح
حسین کوشکیان

فهرست

مقدمه بر الگوریتم ژنتیک.....	3
Error! Bookmark not defined.	Crossover
Error! Bookmark not defined.	Mutation (جهش)
Error! Bookmark not defined.	نکات مهم در ارتباط با جهش
Error! Bookmark not defined.	انواع انتخاب
1. انتخاب بر اساس مناسب بودن (Fitness-Based Selection)	
Error! Bookmark not defined.	
Error! Bookmark not defined.	انتخاب تصادفی (Random Selection)
Error! Bookmark not defined. .	انتخاب بر اساس رنک (Rank Selection)
Error! Bookmark not defined.	انتخاب تعادلی (Elite Selection) (نخبه گرایی)
5. انتخاب تصادفی با جانشینی (Stochastic Universal Sampling)	
Error! Bookmark not defined.	
Error! Bookmark not defined.	ویژگی موازی سازی الگوریتم ژنتیک
Error! Bookmark not defined.	موازی سازی روال (Sequential Parallelism)
Error! Bookmark not defined.	روش جزیره (Island Model)
Error! Bookmark not defined.	منابع

مقدمه بر PSO

الگوریتم بهینه‌سازی ازدحام ذرات (Particle Swarm Optimization) یا به اختصار PSO، یک روش بهینه‌سازی مبتنی بر جمعیت در هوش مصنوعی است که از رفتار اجتماعی موجوداتی مثل پرندگان یا ماهی‌ها الهام گرفته شده است. این الگوریتم اولین بار توسط کندی و ابرهارت در سال 1995 معرفی شد و هدف آن یافتن بهینه‌ی سراسری یک تابع یا مسئله‌ی بهینه‌سازی پیچیده است.

نحوه کار الگوریتم PSO

در PSO، گروهی از ذرات به عنوان نمونه‌ها یا جواب‌های ممکن مسئله، در فضای جستجو حرکت می‌کنند. هر ذره دارای موقعیت و سرعت خاص خود است و بر اساس اطلاعاتی که از بهترین موقعیت خود و بهترین موقعیت دیگر ذرات دریافت می‌کند، به سمت جواب بهینه حرکت می‌کند.

الگوریتم PSO از دو مفهوم اساسی استفاده می‌کند:

1. **بهترین موقعیت فردی (pBest):** بهترین موقعیتی که هر ذره تا کنون پیدا کرده است.
2. **بهترین موقعیت جمعی (gBest):** بهترین موقعیتی که کل جمعیت ذرات به آن دست پیدا کرده است.

موقعیت و سرعت هر ذره در هر مرحله با استفاده از این دو مقدار و همچنین موقعیت فعلی آن ذره به‌روزرسانی می‌شود.

به عنوان یک روش جستجوی فراابتکاری، رویکرد مؤثری برای مسائل PSO الگوریتم پیچیده‌ی بهینه‌سازی است که دارای فضای جستجوی بزرگی هستند و به‌طور گسترده در حوزه‌های مختلف علوم داده و هوش مصنوعی مورد استفاده قرار می‌گیرد.

ویژگی های هر ذره

- موقعیت
- سرعت
- کیفیت
- بهترین تجربه ذره

بهترین تجربه ذره

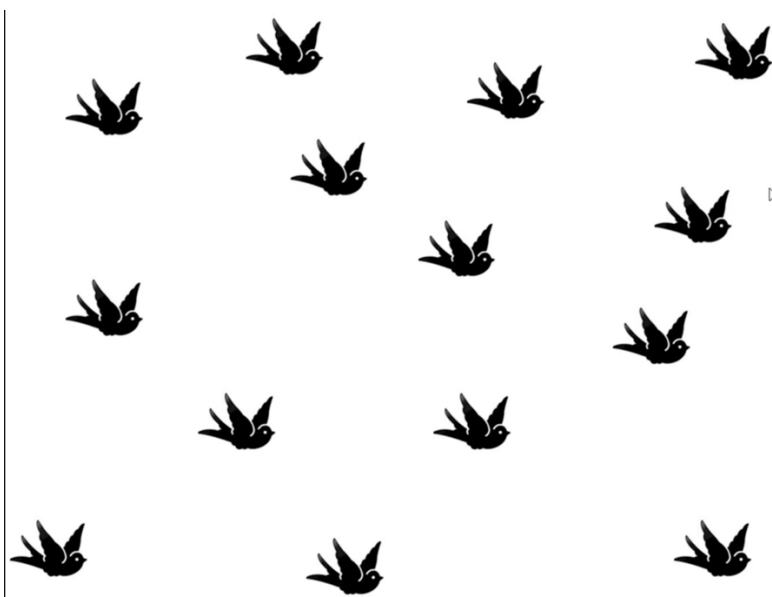
	sol1	sol2
iter1	۱۰۰	۹۰
iter2	۹۰	۳۵
iter3	۹۵	۲۵
iter4	۹۲	۲۵
iter5	۸۹	۲۵
iter6	۸۷	۲۵
iter7	۸۰	۲۵
iter8	۸۵	۳۵
iter9	۸۴	۳۰
iter10	۸۴	۲۹

برای مثال در عکس بالا بهترین تجربه ی ذره ی یک در شمارش هفتم است
مقدار ۸۰ و بهترین تجربه ی ذره ی دو ۲۵ است
ولی بهترین تجربه ی گروه مقدار ۲۵ است که در کل مقدار ها بررسی میشود

جمعیت اولیه به صورت تصادفی

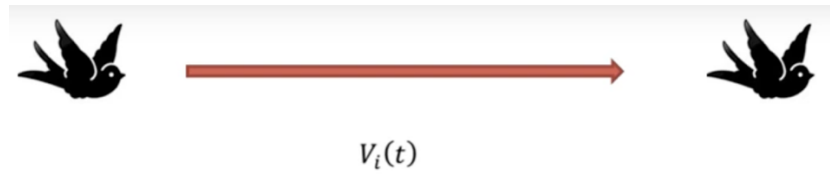
- موقعیت : بین حد پایین و بالا
- سرعت : صفر در نظر گرفته میشود
- کیفیت : محاسبه میشود
-

بهترین تجربه ی ذره = ذره



برای مثال در عکس بالا جمعیت اولیه ما ساخته و به صورت رندم پخش شده اند

هر ذره تمایل دارد به سمت هدفی حرکت کند

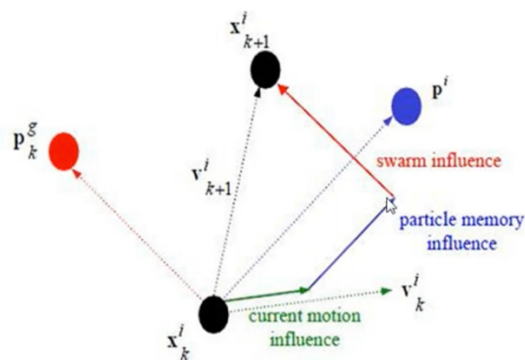


$$X_i = X_i(t-1) + V_i(t)$$

برای مثال ذره بالا وقتی میخواهد حرکت کند با یک سرعتی این فرایند را انجام میدهد
یعنی موقعیت هر ذره برابر است با جمع موقعیت قدیم با سرعت آن

چه عامل هایی بر این سرعت تاثیر دارد ؟

- سرعت قدیم
- فاصله اش تا بهترین تجربه شخصیش
- فاصله اش تا بهترین تجربه کل



$$V_i(t) = w * V_i(t - 1) + c_1 * rand_1 * (P_{i.best} - X_i(t - 1)) + c_2 * rand_2 * (P_{g.best} - X_i(t - 1))$$

تابع رندم در بالا کاری میکند که یک مسیر مستقیم را هر ذره نرود

حل مسئله کوله پشتی با PSO

مقداردهی اولیه

هر ذره (یا راه حل ممکن) به صورت یک بردار باینری تعریف می شود که نشان دهندهی مجموعه ای از آیتم هاست. اگر مقدار یک عنصر ۱ باشد، به این معنی است که آیتم مربوطه در کوله پشتی قرار گرفته است؛ در غیر این صورت، مقدار آن ۰ خواهد بود. ابتدا تعداد مشخصی از ذرات به صورت تصادفی مقداردهی می شوند.

تابع ارزیابی (Fitness)

برای هر ذره، مقدار تناسب (fitness) محاسبه می شود. تابع تناسب برای مسئله کوله پشتی، مجموع ارزش آیتم های انتخاب شده توسط ذره است، به شرطی که وزن کل آیتم ها از ظرفیت کوله پشتی تجاوز نکند. اگر وزن از ظرفیت بیشتر باشد، تناسب آن ذره به عنوان یک راه حل نامعتبر تعیین می شود (مثلاً به صفر می رسد)

به روزرسانی بهترین موقعیت‌ها

در این مرحله، برای هر ذره، بهترین موقعیت فردی و بهترین موقعیت جمعی به روزرسانی می‌شوند:

- اگر تناسب موقعیت فعلی ذره از بهترین موقعیت قبلی آن بیشتر باشد،
BestpBestpBestpبه روزرسانی می‌شود.
- اگر BestpBestpBestpجدید از (BestgBestgBestgبهترین موقعیت جمعی) بهتر باشد،
BestgBestgBestgنیز به روزرسانی می‌شود.

به روزرسانی موقعیت و سرعت

در PSO معمولی، هر ذره موقعیت و سرعت خود را با استفاده از فرمول‌هایی به روزرسانی می‌کند، اما در مسئله کوله‌پشتی که یک مسئله گسسته است، به روزرسانی موقعیت به صورت احتمالاتی انجام می‌شود.

برای هر عنصر از بردار باینری یک احتمال محاسبه می‌شود که به آن عنصر مقدار ۰ یا ۱ بدهد:

- اگر احتمال از یک حد مشخص بیشتر باشد، عنصر به ۱ تبدیل می‌شود.
- در غیر این صورت، عنصر به ۰ تبدیل می‌شود.

تکرار فرآیند

مراحل فوق تا زمانی که به تعداد تکرار مشخصی برسیم یا تغییر خاصی در جواب‌ها مشاهده نشود، ادامه می‌یابد.

نتایج و تحلیل

در این بخش، می‌توانید نتایج حل مسئله کوله‌پشتی با PSO را بررسی کنید. برای نمونه، می‌توانید زمان و کیفیت راه‌حل‌ها را با روش‌های دیگر مقایسه کنید و نشان دهید که

PSO در چه مواقعی کارآمدتر است. همچنین می‌توانید توضیح دهید که چطور PSO در پیدا کردن بهینه سراسری یا نزدیک به بهینه مؤثر است.

منابع

Open ai GPT

https://en.wikipedia.org/wiki/Particle_swarm_optimization

<https://www.karlancer.com/blog/particle-swarm-optimization-algorithm>