

به نام خدا



دانشکده مهندسی کامپیوتر

هوش مصنوعی و سیستم های خبره

پروژه سوم _ ژنتیک

دکتر آرش عبدی

پاییز 1403

طراحان :

آیدا نجفی

فاطمه باقری



- 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 امكان تمدید وجود ندارد.
- داک پروژه را خوانا و مرتب بنویسید.
- زبان برنامه نویسی دلخواه (پیشنهاد پایتون)
- کل محتوای ارسالی را زیپ کرده و نام آن را شماره دانشجویی خود قرار دهید.
- انجام تمرین تک نفره است. لطفا به تنها یی انجام شود، در غیر این صورت نمره منفی در نظر گرفته خواهد شد.

آیدی تلگرام طراحان :

@aida_NJM
@FatemehBagherii



تقریب تابع با کمک برنامه نویسی ژنتیک (GP)

یکی از مسائل مطرح در علوم ریاضی و مهندسی تقریب تابع است. تقریب تابع یعنی با داشتن توانایی محاسبه مقدار یک تابع نامعلوم در نقاط دلخواه، ضابطه ریاضی تابع مذکور تقریب زده شود. در این پژوهه برای تقریب تابع از برنامه نویسی ژنتیک استفاده می شود.

هدف پژوهه: ارزیابی میزان تسلط به مفاهیم پایه الگوریتم های تکاملی (در اینجا برنامه نویسی ژنتیک)

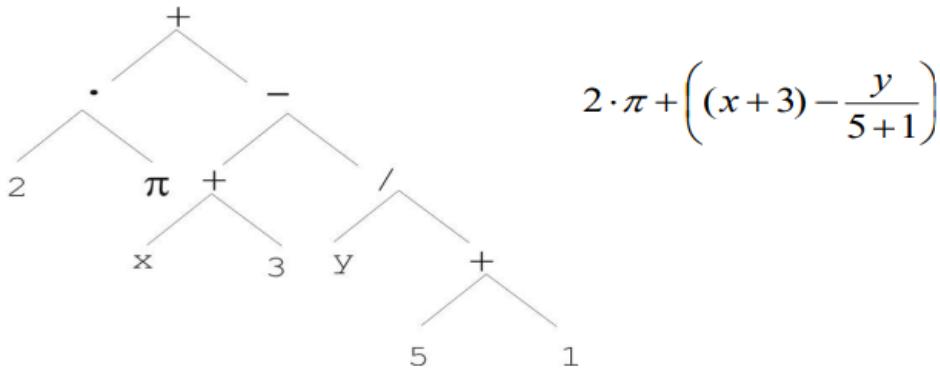
یکی از کاربرد های Genetic Programming ، تقریب توابع است. در این کاربرد، هر فرد از جمعیت بیانگر یک عبارت ریاضی است که به شکل یک درخت نمایش داده میشود. عناصر میانی این درخت عملگرهای ریاضی، منطقی و یا هر گونه عملگر تعریف شده کاربرد و برگ های درخت حاوی مقادیر ثابت و متغیرهای ورودی هستند. هدف از این کاربرد این است که فرمول ریاضیاتی یک تابع هدف را تخمین بزنیم. (تابع هدف یک Black Box است و تنها چیزی که از آن میدانیم یا قادریم بدست آوریم، بدست آوردن مقدار تابع در نقاط دلخواه است). بدین ترتیب در تعدادی نقطه دلخواه مقدار تابع را یافته و از آنها به عنوان مجموعه آموزشی یاد میکنیم. برای هر درخت بر حسب مقادیری که برای نقاط آموزشی بدست می آورد و میزان اختلافش با مقادیر واقعی میتوان یک مقدار شایستگی نسبت داد و بر اساس آن الگوریتم تکاملی را دنبال کرد.

عملگرهایی که در درختان به کار برده میشود حداقل شامل موارد زیر باشد (موارد بیشتری نیز میتوانند اضافه کنید):

عملگرهای دو عملوندی شامل: ${}^8(\text{توان})$ ، $/$ ، \times ، $-$ ، $+$ ، $/%$ (باقي مانده)

عملگرهای تک عملوندی شامل : $\text{Sin}(x)$ ، $\text{Cos}(x)$ ، \sqrt{a}

یک نمونه از درخت:



ورودی برنامه:

تعدادی نقطه آموزشی و مقدار خروجی تابع هدف در آن نقاط (برای سادگی، تابع یک بعدی در نظر گرفته می‌شود اما می‌توان از توابع با بعد بیشتر نیز استفاده کرد). می‌توانید در توابع خود از اعداد ثابت استفاده کنید برای مثال x^2 , $\cos(3x)$ و (علاوه بر اعداد گسسته (مثلاً بین 1 تا 10)، می‌توانید از اعداد اعشاری و بزرگتر نیز استفاده کنید)

تبصره 1 :

در عمل برای تولید ورودی و ارزیابی عملکرد الگوریتم می‌توان توابعی دلخواه (شامل هر عملگری که در الگوریتم پیش‌بینی شده است و حتی عملگرهایی که در الگوریتم پیش‌بینی نشده است) در نظر گرفت و در تعدادی نقطه دلخواه مقدار تابع را محاسبه کرد و این نقاط را به عنوان ورودی الگوریتم در نظر گرفت. حتماً درآزمایشها حالتی را نیز آزمایش کنید که در آن ورودی از روی تابعی تولید شده باشد که دارای عملگری است که در فهرست عملگرهای استفاده شده در GP وجود ندارد. مثال در ضابطه تابع استفاده شده برای تولید نقاط آزمایشی از عملگر لگاریتم یا تائزانت (یا ...) استفاده کنید ولی در GP این عملگرها مجاز نباشند. یا عملگر توان را در GP حذف کنید و نقاط آموزشی را با تابعی که این عملگرها را دارند تولید کنید.

تبصره 2:

در یکی از آزمایشها در تابع نقطه یا نقاط گسستگی ایجاد کنید و تفاوت رفتار GP را بررسی کنید.
مثال متوانید برای نقاط کوچکتر از عدد $c1$ از ضابطه $x^2 = y$ و در نقاط بین $c1$ تا $c2$ از
ضابطه $-x - 7 = y$ و در نقاط بزرگتر از $c2$ از تابع دیگری برای تولید نقاط آموزشی استفاده کنید.

تبصره 3:

علاوه بر آزمایش روی توابعی با ورودی یک بعدی (ورودی فقط x)، توابعی با ورودی بیش از یک
بعد (حداقل 2 بعد) را نیز آزمایش کنید. در مورد توابع دو بعدی (یا با ورودی بیش از 2 بعد) صرفا
توابع ساده را آزمایش کنید و نیازی به انجام نکات مذکور در تبصره 1 و 2 نیست.

تبصره 4:

از روند پیشرفت الگوریتم خود (برای مثال اختلاف بهترین تابع تولید شده در هر نسل و همچنین
MSE کل توابع ساخته شده در آن نسل) با استفاده از کتابخانه Plot نمودار تهیه کنید.

تبصره 5:

سعی کنید در پایان هر نسل و در پایان اجرای الگوریتم، بهترین تابع تشخیص داده را نشان دهید
و در صورت امکان فرمول ریاضی آن را ساده کنید و سعی کنید به تابع اولیه نزدیک تر کنید)
(اینجا بستگی به خلاقیت خودتان دارد)

تبصره 6:

نتایج به دست آمده را با نتایج پیشビینی شده مقایسه کنید (نمودار هر دو را با استفاده از کتابخانه
Plot به دست آورید) و MSE را محاسبه کرده و درصد درستی را به دست آورید.

$$\text{Accuracy Percentage} = \left(1 - \frac{MSE}{\text{Variance of True Outputs}} \right) \times 100$$

آنچه تحويل داده میشود:

1. کد اجرایی برنامه به همراه تست کردن چند ورودی متفاوت توسط دانشجو و نیز امکان تست کردن ورودی های دیگر در زمان تحويل برنامه .
2. گزارش از جزئیات اجرای پروژه که حداقل موارد زیر را در برمیگیرد (بدون ترتیب):
 - (a) نحوه دقیق نگاشت مسئله به درخت و محدودیت های احتمالی (عملگرها، اعداد ثابت، و...)
 - (b) تابع شایستگی
 - (c) تولید جمعیت اولیه
 - (d) نحوه انتخاب والدین
 - (e) نحوه تولید نسل بعد (نحوه ترکیب متقطع و جهش)
 - (f) شرط خاتمه الگوریتم
 - (g) چالش های مواجه شده و روش حل آنها
 - (h) گزارشی از آزمایش های مختلف انجام شده (حداقل 3 آزمایش برای 3 تابع مختلف ورودی) و ارزیابی و تحلیل عملکرد الگوریتم در آزمایش ها
 - (i) جمع بندی و نتایج عملی که درباره GP در این پروژه به دست آوردید.

نکته نهایی:

در تمام طول پروژه از خلاقیت و ایده های جدید شما استقبال می شود و نمره امتیازی به آنها تعلق می گیرد پس نگران نباشید و تلاش کنید از دانش خود برای نوآوری استفاده کنید 😊