

به نام خدا



دانشکده علوم کامپیوتر

هوش مصنوعی و سیستم های خبره

پروژه سوم

برنامه نویسی ژنتیک برای تقریب تابع

دکتر آرش عبدی

بهار ۱۴۰۳

طراحان:

ماکان حاجی سیدجوادى

الهام غلامى

## پروژه سوم درس هوش مصنوعی و سیستم های خبره

- در صورت وجود هرگونه ابهام در پروژه به طراح آن پیام دهید.
- علاوه بر آپلود داخل کوئرا، پروژه ها به صورت شفاهی ارائه گرفته میشوند؛ بنابراین تسلط شما هنگام ارائه شفاهی نسبت به گزارش پروژه اهمیت بیشتری دارد. برای توضیح بیشتر، اگر پروژه ۱۰۰ نمره داشته باشد، نمره ارائه شفاهی ۹۰ و نمره گزارش، ۱۰ خواهد بود؛ بنابراین سعی کنید قبل از ارائه شفاهی نسبت به کد خود و مبحث پروژه آمادگی لازم را داشته باشید.
- با توجه به تنظیم شدن ددلاین توسط خود شما امکان تمدید وجود ندارد.
- زبان برنامه نویسی پیشنهادی پایتون است ولی مجاز به استفاده از هر زبان دلخواهی هستید. نمره هر زبان با هم برابر است.
- پروژه تک نفره است لذا از هرگونه تقلب در انجام پروژه خودداری کنید.
- برای این پروژه شما باید حداقل یک صفحه توضیحات ارائه دهید. این توضیحات در زمان ارائه شفاهی به شما کمک می کنند. نیاز به توضیح کد نیست؛ روند پیاده سازی پروژه، چالش هایی که به آن خوردید و خلاقیت های شما ارزش بیشتری دارند.

آیدی تلگرام طراح:

- @abvkhia
- @elham\_gholamim

## هدف پروژه:

ارزیابی میزان تسلط به مفاهیم پایه الگوریتم های تکاملی (به طور خاص در این پروژه؛ الگوریتم های ژنتیک)

## شرح پروژه:

برنامه نویسی ژنتیک (GP) روشی ابتکاری برای حل مسائل پیچیده با الهام از فرایند تکامل طبیعی است. در این روش، جمعیت هایی از برنامه های کامپیوتری به طور تصادفی تولید می شوند و سپس با استفاده از اپراتورهای ژنتیکی مانند جهش و تقاطع، تکامل می یابند. یکی از کاربردهای جالب GP، تقریب توابع است. در این روش، هر فرد از جمعیت بیانگر یک عبارت ریاضی است که به شکل یک درخت نمایش داده می شود. عناصر میانی این درخت شامل عملگرهای ریاضی، منطقی و یا هر گونه عملگر تعریف شده توسط کاربر هستند. برگ های درخت حاوی مقادیر ثابت و متغیرهای ورودی هستند.

در این پروژه، تابع هدف یک جعبه سیاه (Black Box) است و تنها چیزی که از آن می دانیم (یا قادریم بدست آوریم)، مقدار تابع در نقاط دلخواه است. بدین ترتیب، در تعدادی نقطه دلخواه مقدار تابع را یافته و از آنها به عنوان مجموعه آموزشی یاد می کنیم.

## نکات تکمیلی:

۱. الگوریتم ژنتیک را مرحله به مرحله پیاده سازی کنید. استفاده از کتابخانه در این پروژه مجاز نیست. (البته برای نمایش درخت خروجی یا کارهای اضافه دیگر می توانید از کتابخانه های موجود استفاده کنید، اما الگوریتم ژنتیک توسط خودتان باید پیاده سازی شود.)

۲. عملگرهایی که در درختان به کار برده می شود حداقل شامل موارد زیر باشد (موارد بیشتری نیز می توانید اضافه کنید)

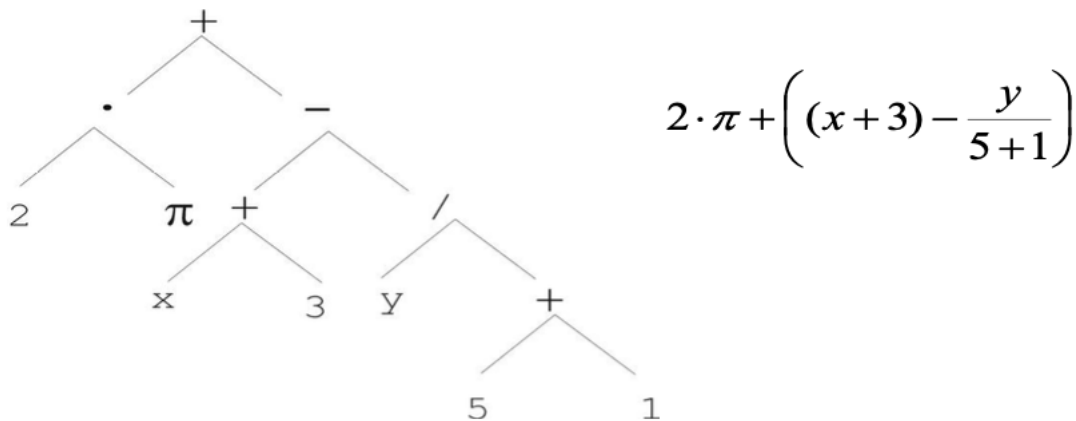
- عملگرهای دو عملوندی:

$+, -, \times, /, ^$

- عملگرهای تک عملوندی:

$\sin(x), \cos(x)$

برای مثال یک نمونه از درخت می‌تواند به این شکل باشد:



در این نمونه، در نقاطی دلخواه خروجی ضابطه را به دست آورید و به عنوان نمونه به الگوریتم بدهید. الگوریتم شما باید بتواند در مراحل پایانی، درختی نزدیک به درخت بالا خروجی دهد.

۳. حتما در آزمایش‌های خود 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. برای مثال اگر در فهرست عملگرهای مجاز الگوریتم خود از لگاریتم یا تانژانت استفاده نکرده‌اید، نقاط آموزشی را از تابعی در نظر بگیرید که شامل لگاریتم یا تانژانت اند. نتیجه را گزارش دهید.

۴. در یکی از آزمایش‌ها رفتار GP را در تقریب تابعی که در بخش‌های مختلف دامنه خود، رفتار متفاوتی دارند، بررسی کنید. به این منظور، باید نقاط گسستگی در تابع ایجاد کنید.

برای مثال تابع زیر را در نظر بگیرید:

$$f(x) = \begin{cases} 2^y, & \text{if } x < c_1 \\ 3 - y, & \text{if } c_1 \leq x \leq c_2 \\ < \text{some other function} >, & \text{if } x > c_2 \end{cases}$$

در این مثال، تابع در سه بخش مختلف دامنه (X) رفتار متفاوتی دارد:

- بخش اول: ( $X < c_1$ ) تابع به صورت  $2^y$  است.
- بخش دوم: ( $c_1 \leq X \leq c_2$ ) تابع به صورت  $3 - y$  است.
- بخش سوم: ( $X > c_2$ ) تابع به صورت دلخواه انتخاب می‌شود.

رفتار الگوریتم را برای این تابع مشاهده کنید و دقت تقریب در بخش‌های مختلف تابع را بسنجید.

۵. از مثال‌های ساده شروع کنید و ابتدا مطمئن شوید کد سالم است و درست کار می‌کند سپس سراغ مثال‌های ساده‌تر بروید.

۶. علاوه بر آزمایش روی توابعی با ورودی یک بعدی (ورودی فقط X) توابعی با ورودی بیش از یک بعد (حداقل ۲ بعد) را نیز آزمایش کنید. در مورد توابع دو بعدی (یا با ورودی بیش از ۲ بعد) صرفاً توابع ساده را آزمایش کنید. لزومی ندارد مورد ۳ یا ۴ را برای این نوع توابع آزمایش کنید.

در نهایت تلاش شما نمره اصلی را خواهد داشت و حتی اگر نتایج دلچسب نباشند، امکان گرفتن نمره کامل در صورت صرف وقت و یادگرفتن مشکلات راه، وجود دارد. همچنین در صورتی که نتایج کاملی از همه بخش‌ها به دست‌آورید، ممکن است باعث نمره مثبت (نمره اضافی) هم بشود.

موفق باشید!