به نام خدا

پروژه شماره 1 هوش مصنوعی

محمد حسین میرزائی 99522158

پروژه از 3 فایل py. تشکیل شده است که هر کدام توضیح داده خواهد شد :

Project_AI : فایل اصلی برنامه می باشد که بخش های مختلف آن را توضیح می دهیم :

4 لیست اصلی func_list که ،string اولیه ورودی که به وسیله Random تولید شده را می گیرد و در خود ذخیره می کند ، tree_list که func_list اولیه که ورودی string دارد را به درخت تبدیل و در خود ذخیره می کند ، fitness_list که درخت های درست شده را می گیرد و با توجه به inputو output لیست که black box ما را تشکیل می دهد ،fitness را محاسبه و در خود حفظ میکند ، همچنین این لیست در هر بخش علاوه بر مقدار fitness درخت را نیز در خود نگه میدارد ،پس در هر بخش tuple در نظر گرفته شده است که هر 2 مقدار را ذخیره کند .

Max_fit که مرتب شده fitness_list بر اساس مقدار fitness می باشد .

توابع این فایل :

Random_generation که مسئولیت ساخت ورودی ها را به صورت random دارد و 'x' که مجهول ماست و این تابع حتما شامل این مجهول می شود .

Make_funct که لیست funct_list را با مقادیر گرفته شده در تابع قبل مقدار دهی می کند .

Make_tree که لیست عبارات را گرفته و درخت نظیر آن را می سازد و آن را در tree_list ذخیره می کند.

Fitness که inputو output ،black box و tree_list را می گیرد و قدر مطلق اختلاف آن ها را چک می کند که هر چه به صفر نزدیک تر باشد ، آن جواب ارزش بیشتری دارد .

Calc_fit که fitness_list را می سازد و max_fit را به عنوان مرتب شده آن بر می گرداند .

Crossover که تابع کامل آن در فایل دیگر پیدا سازی شده است.

Mutation و swap_mutation نیز شرایط تابع بالا را دارند .

فایل Tree.py :

Class Node که درخت را می سازد و درفایل قبلی هم از آن استفاده شد .

Evaluate که مقدار expression tree را به صورت float محاسبه می کند و بر می گرداند .

Inorder که درخت مورد نظر را پیمایش می کند .

Infix_to_postfix که مقدار اولیه رندوم را می گیرد (که به صورت infix می باشد) و آن را به postfix تبدیل می کند .

cunstructTree که postfix را می گیرد و آن را به درخت تبدیل می کند .

All_nodes که یک درخت را می گیرد (در این جا root)و کل node های آن درخت را بر می گرداند .

Check_parent که یک node را می گیرد و parent آن را در صورت وجود بر می گرداند .

Select_change_node که همان نقش crossover را دارد و 2 node را گرفته (به صورت رندوم انتخاب می کند) و parent آن را با توابع پیدا می کند و با توجه به node مورد نظر آن ها را جابه جا می کند .

Mutation_tree_node ابتدا درخت را به All_node پاس می دهد و بعد از بین node ها به صورت رندوم یکی را انتخاب و اگر operatorجای آن operator و در غیر این صورتoperand قرار می دهد .

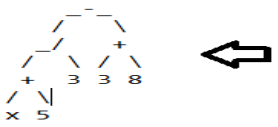
Swap_mutation_tree_node نیز مانند قسمت قبل کار می کند ، با این تفاوت که در این قسمت 2 node انتخاب شده از یک درخت به صورت رندوم انتخاب و مقدار این 2 node با یکدیگر عوض می شود (از هر یک از این 2 مدل می توان برای mutation استفاده کرد .)

Infix_expTree که برای تبدیل ، expression Tree به infix استفاده می شود .

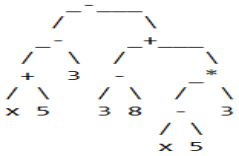
فایل show_tree.py :

در این فایل توابع صرفا برای پیاده سازی شکل درخت استفاده شده است .

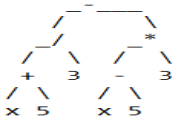
نمونه ای از crossover :



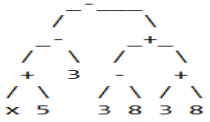
تابع اول



تابع دوم

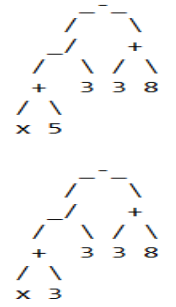


تغییر یافته تابع اول

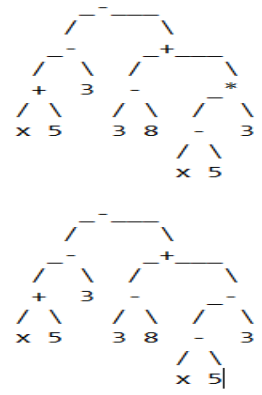


تغییر یافته تابع دوم

نمونه ای از mutation :(نوع یک)

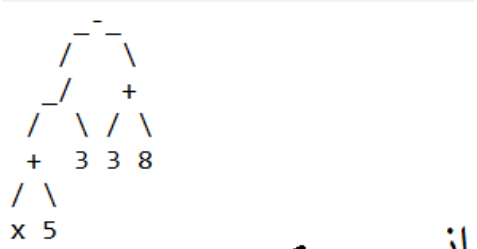


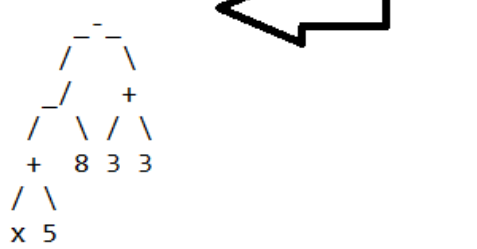
تابع قبل و بعد از mutation



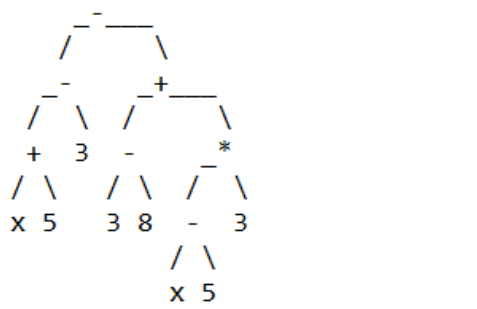
تابع قبل و بعد از mutation

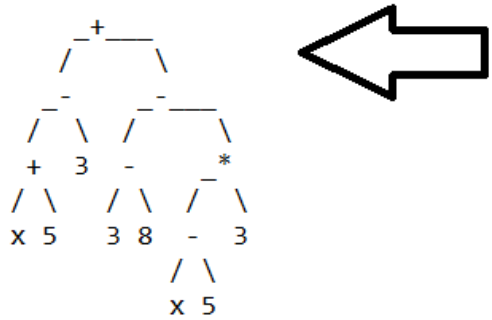
نمونه ای از mutation :(نوع دو)





تابع اول قبل و بعد از mutation





تابع دوم قبل و بعد از mutation