



آشنایی با الگوریتم ژنتیک

ایمان اشکانوند

الگوریتم

سال ششم، شماره ۱۶، بهار ۸۶

۳۸

الگوریتم ژنتیک (Genetic Algorithm- GA) تکنیک جستجویی در علم رایانه برای یافتن راه حل تقریبی برای بهینه سازی و مسائل جستجو است. الگوریتم ژنتیک نوع خاصی از الگوریتمهای تکامل است که از تکنیکهای زیست شناسی فرگشتی مانند وراثت و جهش استفاده می کند. الگوریتمهای ژنتیک معمولاً به عنوان یک شبیه ساز کامپیوتری که در آن جمعیت یک نمونه انتزاعی (کروموزومها) از نامزدهای راه حل یک مسئله بهینه سازی که به راه حل بهتری منجر شود، پیاده سازی می شوند. به طور سنتی راه حلها به شکل رشته هایی از ۰ و ۱ بودند، اما امروزه به گونه های دیگری هم پیاده سازی شده اند. فرضیه با جمعیتی کاملاً تصادفی منحصر بفرد آغاز می شود و در نسلها ادامه می یابد. در هر نسل گنجایش تمام جمعیت ارزیابی می شود، چندین فرد منحصر در فرایندی تصادفی از نسل جاری انتخاب می شوند (بر اساس شایستگیها) و برای شکل دادن نسل جدید، اصلاح می شوند (کسر یا دوباره ترکیب می شوند) و در تکرار بعدی الگوریتم به نسل جاری تبدیل می شود.

۱- مقدمه

اخیراً، دو تکنیک جستجوی هیورستیک که در هوش مصنوعی از آنها استفاده می شود به وجود آمده است که شامل: ۱- الگوریتم های ژنتیک (GAs) ۲- شبکه های عصبی (NNs) می باشند. هر دو تکنیک GAs و NNs متکی بر مدل هایی از طبیعت هستند. یک الگوریتم ژنتیک بر مبنای علم ژنتیک و فرضیه داروین مدل شده است در حالیکه یک شبکه عصبی متکی بر مدل هایی از شناخت انسان می باشد.

یک کاربرد معمولی الگوریتم ژنتیک استفاده به عنوان تابع بهینه ساز می باشد. از دیگر کاربردهای الگوریتم ژنتیک می توان به استنتاج ارگانیزم اجرای خوب و موفق در یک محیط معین اشاره نمود. هر کدام از این کاربردهای GA متکی بر بقای افراد بهتر (انتخاب طبیعی) بر مبنای نظریه داروین می باشد.

بندی (encapsulate) تعدادی از رفتارهای مطلوب استفاده می شود. به عبارت دیگر یک بار دیگر وزنهای خوب پیدا می شوند.

شبکه عصبی می تواند به وسیله خودش مورد استفاده قرار گیرد تا رفتاری مطلوبی را از خود نشان دهد. از یک مجموعه از وزنهای خوب آنهایی که به پیشروی (حرکت به جلو) آشنا نیستند باید تعلیم ببینند. به جای الگوریتمهای تعلیم قدیمی شبکه عصبی، De Garis از الگوریتم ژنتیک برای تعلیم یک مجموعه از وزنهای خوب استفاده کرد. تعلیم به وسیله خود شبکه عصبی انجام نمی شود. این روش Genetic programming نامیده می شود.

همانطور که ذکر شد الگوریتم ژنتیک یک جمعیت از کروموزومها را بر طبق فرایند انتخاب طبیعی استنتاج می کند. در طی این فرایند اپراتورهای ژنتیکی کروموزومهای جدید (فرزندان) را بر اساس کارایی بیشتر نسل قبل (والدها) ایجاد می کنند. ترکیب (در این جا کراس آور نامیده می شود) یک اپراتور ژنتیکی و یک کلید قدرتمند

شبکه های عصبی از طرف دیگر به نظر می رسد در مکانیزمهای کنترلی برای ارگانیزم ها مفید هستند (یک ارگانیزم باید از خطر دوری کند و جستجو کند بهتر را). این دو روش به طور طبیعی یک تفاوت در مقیاس را منعکس می کنند. در حالیکه شبکه عصبی می تواند در کنترل یک ارگانیزم مخصوص الگوریتم ژنتیک استفاده شود.

در استنتاج یک جمعیت از ارگانیزم ها، الگوریتم ژنتیک در یک محیط معلوم (معین) خوب اجرا می شود اگر در یک شبکه عصبی که در بسته بندی (encapsulate) یک رفتار مخصوص استفاده شده است می تواند الگوریتم ژنتیک استفاده شود تا آن رفتار را، بوسیله استنتاج یک جمعیت از شبکه های عصبی استنتاج کند.

یک روش مخصوص استنتاج رفتار، بوسیله De Garis توضیح داده شده است. در این روش از الگوریتم ژنتیک برای استنتاج یک جمعیت از شبکه های عصبی استفاده می شود. هر شبکه عصبی یک مجموعه از وزنهای قابل تنظیم دارد و برای بسته



Software

نرون است. قانون sigmoid activation از تابع استفاده می کند.

هر Genetic programming Neural Network (که به وسیله Gen Net ، De Garis نامیده می شود) یک مجموعه از نرونهای ورودی ، نرونهای خروجی و نرونهای مخفی را شامل می شود. نرون های ورودی ، نرون هایی هستند که اطلاعات را از محیط دریافت می کنند. نرون های خروجی اعمالی را فراهم می کنند که بر محیط تاثیر می گذارند. اکنون به ما اجازه دهید که به بررسی یک مثال [۱] که همچنین برای توضیح آزمایش ها در این گزارش استفاده شده است بپردازیم.

ما می خواهیم به مثال راه پیما (Walker) رجوع کنیم. در این مثال یک GenNet کنترل یک مجموعه پایه های به هم متصل را بر عهده دارد. به وسیله الگوریتم ژنتیک شیوه حرکت آموخته می شود تا بتوانند راه روند. GenNet به وسیله De Garis مطرح شد که شامل ۱۲ نرون می باشد که هشت نرون ورودی و چهار نرون خروجی می باشد. پاهای به هم متصل از دو پا که در مفصل ران به هم متصل شده اند تشکیل میشوند هر پا شامل یک ران و یک ماهیچه ساق پا است که در زانو به هم متصل می شوند.

این به وسیله چهار قطعه نشان داده شده است. (یکی برای هر ماهیچه ساق پا و ران) چنانچه پاها حرکت کنند ، می توانیم زاویه و سرعت زاویه ای هر قطعه را دریافت کنیم. این اطلاعات به وسیله ۸ نرون ورودی دریافت می شود چهار نرون خروجی برای نمایش شتاب زاویه ای که به هر قطعه فرستاده می شود به کار می رود. (شکل را ببینید)

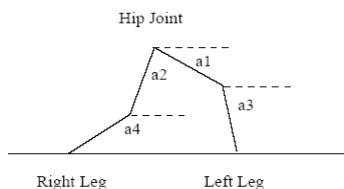


Figure 1. The Stick Legs

در این مثال می توانیم از شبکه ها به عنوان یک کنترلگر برای پاهای به هم متصل استفاده کنیم برای حالت ورودی (به عبارت دیگر زاویه ها و سرعت زاویه ای ها) GenNet یک مجموعه شتاب زاویه ای که پاهای به هم متصل را کنترل می کند تولید خواهد کرد. این شتاب های زاویه ای، زاویه ها و سرعت زاویه ای قطعات را تغییر خواهند داد و برای GenNet ورودی های جدیدی را فراهم می کنند.

هدف از GenNet به عنوان یک شبیه سازی زمان جداگانه که این پاهای به هم متصل در بازه زمانی کوتاه حرکت کنند استفاده می شود. یک سری وزن ها با مقادیر معین اختیاری در GenNet پاهای به هم متصل را در مدلهای تصادفی حرکت خواهند داد در صورتیکه با وزن های مناسب این امکان برای GenNet وجود دارد که پاهای به هم متصل را در چنین راهی به عنوان مثال برای نمایش رفتار راه رفتن کنترل کند در واقع هدف ، یاد گرفتن وزن های مناسب است.

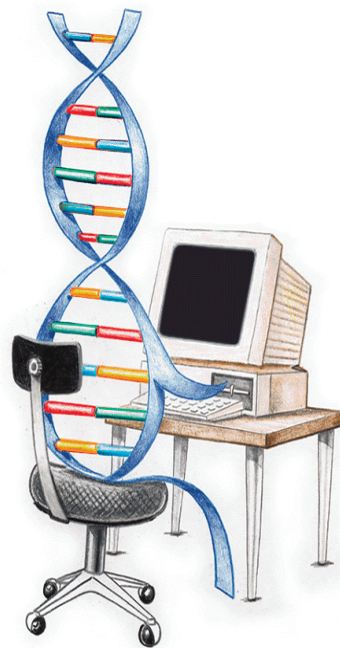
بعد از بیت پنجم اتفاق افتاده باشد ، در نتیجه هر فرزند جدید یک نیمه از بیتیهای والدش را دریافت می کند .

Child ۱: ۱۰۱۰۱۱۰۰۰۰

Child ۲: ۱۰۰۰۰۰۱۰۱۰

در Genetic programming ، یک GA برای استنتاج یک جمعیت که fit شبکه عصبی را افزایش می دهد استفاده می شود .

Fitness هر شبکه عصبی با چگونگی خوب کردن یک رفتار مطلوب که در حال اجرا است اندازه گیری می شود . برای استفاده از GA برای این هدف ، ارائه یک رشته بیتی از شبکه عصبی مورد نیاز است همچنین به یک تابع Fitness که آن چگونگی خوب شدن یک شبکه عصبی را نشان می دهد نیز نیاز خواهد بود. این ها موضوعاتی هستند که در قسمت بعد به آنها پرداخته می شود .



۲- الگوریتم ژنتیک

کتاب "Adaptation in Natural and Artificial Systems" مسائل مهم مربوط به الگوریتمهای ژنتیک را مطرح می کند . یک الگوریتم ژنتیک تشکیل شده است از یک جمعیت از کروموزمهای اولیه ، که آنها کروموزمهای جدید (به همراه تعدادی تولید نسل) را بر طبق کارایی شان (fitness) در یک محیط تولید می کنند. کروموزمهایی که دارای بیشترین کارایی هستند ، بیشترین احتمال را برای بقاء دارند. کروموزمها دو به دو با هم ترکیب می شوند و فرزندان را ایجاد می کنند . فرزندان به وسیله اعمال رندم اپراتورهای ژنتیکی بر روی کروموزمها (والدها) ایجاد می شوند . کروموزمهای یک جمعیت، دو به دو به همراه اپراتورهای ژنتیکی ترکیب می شوند تا یک راهبرد جستجو را اجرا کنند ، که این راهبرد جستجو فرضیاتی درباره فضای جستجو در نظر می گیرد و کروموزم کارآمدتر را انتخاب می کند .

هر کروموزم یک جمعیت ، یک نقطه در فضای جستجو است . یک کروموزم در GA با یک رشته بیتی به طول n نمایش داده می شود . در نتیجه هر کروموزم نماینده یک نقطه در فضای به اندازه 2^n است . برای ارائه یک رشته بیتی معین ، الگوریتم ژنتیک نسل جدید را با استفاده از اپراتورهای ژنتیکی کراس آور (ترکیب) و جهش تولید خواهد کرد .

اپراتور جهش در سطح یک بیت ، به طور تصادفی بیتها را در داخل جمعیت جابجایی می کند . نرخ جهش کم است ، معمولاً در حدود یک هزارم . اپراتور کراس آور روی کروموزمها عمل می کند . نقطه کراس آور به طور رندم انتخاب می شود ، برای دو کروموزمی (parents) که به طور رندم انتخاب شده اند . این نقطه بین بیتیهای کروموزمها واقع شده است و هر کروموزم را به دو قسمت سمت چپ و راست تقسیم می کند ، کراس آور در نهایت قسمت سمت چپ (یا راست) دو کروموزم را با یکدیگر جابجا می کند .

یک مثال در رابطه با کراس آور : دو کروموزم (parents) زیر را در نظر بگیرید

Parent ۱: ۱۰۱۰۱۰۱۰۱۰

Parent ۲: ۱۰۰۰۰۰۱۰۰۰۰

فرض کنید نقطه کراس آور به طور تصادفی

۳- شبکه عصبی و برنامه نویسی ژنتیکی

کلاس شبکه های عصبی (NNs) یک زیر کلاس از مدلهای پردازش توزیع شده موازی (PDP) است . مدلهای PDP فرض می کنند که اطلاعات در حال پردازش یک نتیجه از محاوره بین عناصر ساده در حال پردازش است . در Genetic programming تعداد نرونها مجموعه ای با کاربرد است . این تعداد به رفتاری که می آموزند وابسته هستند .

شبکه کاملاً متصل است (هر نرون یک ارتباط از خودش به تمام نرونهای دیگر دارد به همراه مقدار خالص وزنشان در رنج -۱.۰ تا ۱.۰). ورودی خالص هر نرون (قانون propagation) در واقع مجموع حاصلضربهای ورودیهایش در وزنهایش است . خروجی هر نرون ، فعالیت (activation) آن



در مثال نامبرده، الگوریتم ژنتیک برای یادگیری وزن های GenNet استفاده شده است. این به یک رشته بیت برای نمایش شبکه های عصبی نیاز دارد و یک تابع Fitness کمیتی است که تعیین می کند چگونه روش خوبی برای راه رفتن به دست آوریم. در برنامه نویسی ژنتیکی هر مقدار واقعی وزن به وسیله یک مجموعه عدد از بیتها نمایش داده می شوند. از این رو هر نرون با بقیه (از جمله خودش) ارتباط برقرار می کند. وزنها n نرون شبکه می تواند در یک ماتریس n تایی ذخیره شود. هر عنصر ماتریس وزن یکی از ارتباط ها را ذخیره می کند. هر ماتریس یک GenNet منحصر به فرد

را تعیین می کند و می تواند به عنوان یک رشته بیت در سطرهای بزرگ منظم نمایش داده شود. هر فرد متعلق به GA، در یک ماتریس وزن طولی نمایش داده می شود اگر n نرون عصبی در شبکه وجود داشته باشد Bn دو بیتی هستند که برای هر فرد لازم است. برای پاهای به هم متصل De Garis هر وزن را به وسیله هفت بیت نمایش داد نتیجه اینکه در هر فرد بعد از اینکه ۱۲ نرون در شبکه وجود داشته باشد، ۱۰۰۸ بیت وجود دارد. تابع Fitness که برای ارزیابی مشخصات برتر افراد استفاده می شود باید به وسیله هر فردی از جمعیت پذیرفته شود و بعضی کمیت هایی که

شیوه صحیح راه رفتن را انجام می دهند به وسیله آن فرد برمی گردد. یک تابع Fitness درست مسافت پیموده شده به وسیله پاهای به هم متصل را محاسبه می کند. فرد بهتر، جلوتر حرکت می کند (به عبارت دیگر بهتر راه می رود) و اشتباهات فردی در حرکت او خیلی کم، رخ می دهد. هدف پیدا کردن یک مجموعه از وزن هایی است که به وسیله Gen Net نتیجه می شود و شیوه صحیح راه رفتن را تولید خواهد کرد. قسمتی از طرح کلی که یک برنامه نویسی ژنتیکی خوب در رسیدن به این هدف موفق گردیده است را دنبال نمایید.



آشنایی با ActiveX ها

همانند هر صنعت دیگری، کارآیی یکی از دغدغه های اصلی برنامه نویسان است و یکی از جنبه های مهم کارآیی اجتناب از دوباره کاری است. اگر کدی نوشته اید که کار خاصی را انجام می دهد، چرا باید دفعه بعد همان کد را دوباره بنویسید؟ اولین گام در راه استفاده مجدد از کدها، مفهوم زیر مجموعه یا روال Procedure است. کدی را یکبار می نویسید و از آن به بعد در هر جای برنامه که لازم بود فقط کافی است آنرا احضار Call کنید. قرار دادن روال های کلی در یک مدول برنامه نویسی قدمی به پیش بود. با این تمهید می توان از آن روال ها در برنامه های مختلف استفاده کرد. برنامه نویسی مدولار سالها برنامه نویسان را به خود مشغول کرده بود و در واقع یکی از محرکهای اصلی توسعه سیستم عامل ویندوز هم همین ایده کدهای قابل اشتراک و قابل استفاده مجدد بوده است. اولین گام در راه پیاده سازی عناصر مدولار تکنولوژی OLE یا Object Linking and Embedding بود. هدف اولیه OLE ایجاد سندهای مرکب Compound Documents با استفاده از برنامه های مختلف بود. سندی که مقداری متن و مقداری نمودار دارد، و هر کدام آنها با نرم افزار خاص خود ایجاد شده اند، نمونه ای از یک سند مرکب است. وقتی با متن کار می کنید نرم افزار واژه پرداز کنترل را بدست می گیرد و وقتی با نمودارها کار می کنید نرم افزار ترسیمی مسئولیت را به عهده می گیرد. OLE با وجود کندی و مشکلات دیگر مسلماً قدمی به پیش بود.

تکنولوژی OLE خود بر یک استاندارد کلی تر بنام COM یا Component Object Model استوار است. بزودی COM از سندهای مرکب فراتر رفت و OLE را هم بدنبال خود کشاند و از آن به بعد OLE اصطلاحی شد برای هر چیزی که از تکنولوژی COM استفاده می کرد. سالها بعد که میکروسافت بطور جدی درگیر اینترنت شد، و اصطلاح Activex هم وارد ادبیات کامپیوتری شد. ابتدا این اصطلاح فقط در رابطه با اینترنت و وب بود اما اوضاع بدین منوال باقی نماند و اکنون Activex به آن بخش از تکنولوژیهای COM گفته می شود که در آنها یک قطعه نرم افزاری امکانات خود را در اختیار برنامه های دیگر می گذارد. یکی از ادعاهای Activex (که بویژه به اینترنت مربوط می شود) پشتیبانی از نرم افزارهای توزیع شده Distributed است، و این یعنی، کنترل های Activex به شما سرویس خواهند داد، حتی اگر در کامپیوتری دیگر (و هزاران کیلومتر دورتر باشند اما چه بر سر OLE آمد؟ این تکنولوژی دوباره به وضعیت اولیه اش برگشت و اکنون فقط با سندهای مرکب سروکار دارد.

نقاط ضعف و قوت: Activex

تکنولوژی Activex بهترین وسیله برای ایجاد محتویات فعال در وب است. البته برخی با این عقیده مخالفند و باید گفت که این تکنولوژی مسلماً تنها ابزار محتویات فعال وب نیست. پس اجازه دهید نگاهی به جنبه های مثبت و منفی این تکنولوژی بیندازیم.

نقاط قوت :

یکی از مهمترین نقاط قوت Activex قدرت آن است. یک کنترل Activex تقریباً از عهده هر کاری که یک برنامه معمولی بتواند انجام دهد، برمی آید. از دیدگاه یک برنامه نویس وب این بهترین جنبه یک ابزار خلق محتویات دینامیک است. دیگر ابزارهای محتویات فعال، مانند جاوا و CGI در این زمینه بسیار محدودتر از Activex هستند. نکته مثبت دیگر، حداقل برای بسیاری از افراد، امکان استفاده از مهارتهای عادی برنامه نویسی در خلق کنترل های Activex است. روش انجام بارگیری های وب هم یکی از مزایای Activex است.

نقاط ضعف - هیچ چیز کامل نیست!

شاید مهمترین جنبه منفی Activex ضعف امنیتی آن باشد. این ضعف ظاهراً یکی از تبعات اجتناب ناپذیر قدرت و انعطاف تکنولوژی Activex است. یک برنامه نویس شرور می تواند ب راحتی با این تکنولوژی نرم افزاری بسازد که به کامپیوتر دیگران صدمه بزند. یکی دیگر از نقاط ضعف تکنولوژی Activex آن است که فقط برخی از کاوشگرهای امروزی از آن پشتیبانی می کنند. اگر در یک صفحه وب نرم افزارهای Activex وجود داشته باشد و فردی با یک کاوشگر که از Activex پشتیبانی نمی کند این صفحه را باز کند قادر به استفاده از مزایای آن نخواهد بود.