بسم ا... الرحمن الرحیم

جزوی درس مباحث ویژه در علوم کامپیوتر

دانشگاه پیام نور

پاییز ۹۳

فهرست مطالب **صفحه**

مروری بر الگوریتم های فرامکاشفه ای

الگوریتم ژنتیک:

الگوریتم ژنتیک که بعنوان یکی از روشهای تصادفی بهینه یابی شناخته شده, توسط جان هالند در سال ۱۹۶۷ ابداع شده است. بعدها این روش با تلاشهای گلدبرگ ۱۹۸۹, مکان خویش را یافته و امروزه نیز بواسطه توانایی های خویش , جای مناسبی در میان دیگر روشها دارد. روال بهینه یابی در الگوریتم ژنتیک براساس یک روند تصادفی- هدایت شده استوار می باشد. این روش , بر مبنای نظریه تکامل تدریجی و ایده های بنیادین داروین پایه گذاری شده است.در این روش , ابتدا برای تعدادی ثابت که جمعیت نامیده می شود مجموعه ای از پارامترهای هدف بصورت اتفاقی تولید می شود , پس از اجرای برنامه شبیه ساز عددی را که معرف انحراف معیار و یا برازش آن مجموعه از اطلاعات است را به آن عضو از جمعیت مذکور نسبت می دهیم . این عمل را برای تک تک اعضای ایجاد شده تکرار می کنیم , سپس با فراخوانی عملگرهای الگوریتم ژنتیک از جمله لقاح , جهش و انتخاب نسل بعد را شکل می دهیم و این روال تا ارضای معیار همگرایی ادامه داده خواهد شد. هنگامی که لغت تنازع بقا به کار می‌رود اغلب بار ارزشی منفی آن به ذهن می‌آید. شاید همزمان قانون جنگل به ذهن برسد و حکم بقای قوی‌تر !البته برای آنکه خیالتان راحت شود می‌توانید فکر کنید که همیشه هم قوی‌ترین‌ها برنده نبوده‌اند. مثلا دایناسورها با وجود جثه عظیم و قوی‌تر بودن در طی روندی کاملا طبیعی بازی بقا و ادامه نسل را واگذار کردند در حالی که موجوداتی بسیار ضعیف‌تر از آنها حیات خویش را ادامه دادند. ظاهرا طبیعت بهترین‌ها را تنها بر اساس هیکل انتخاب نمی‌کند! در واقع درست‌تر آنست که بگوییم طبیعت مناسب ترین‌ها (Fittest) را انتخاب می‌کند نه بهترین‌ها .قانون انتخاب طبیعی بدین صورت است که تنها گونه‌هایی از یک جمعیت ادامه نسل می‌دهند که بهترین خصوصیات را داشته باشند و آنهایی که این خصوصیات را نداشته باشند به تدریج و در طی زمان از بین می‌روند.

مثلا فرض کنید گونه خاصی از افراد، هوش بسیار بیشتری از بقیه افراد یک جامعه یا کولونی دارند. در شرایط کاملا طبیعی این افراد پیشرفت بهتری خواهند کرد و رفاه نسبتا بالاتری خواهند داشت و این رفاه خود باعث طول عمر بیشتر و باروری بهتر خواهد بود(توجه کنید شرایط طبیعیست نه در یک جامعه سطح بالا با

ملاحظات امروزی یعنی طول عمر بیشتر در این جامعه نمونه با زاد و ولد بیشتر همراه است). حال اگر این خصوصیت(هوش)ارثی باشد به طبع در نسل بعدی همان جامعه تعداد افراد باهوش به دلیل زاد و ولد بیشتر این‌گونه افراد بیشتر خواهد بود .اگر همین روند را ادامه دهید خواهید دید که در طی نسل‌های متوالی دائما جامعه نمونه ما باهوش و باهوش‌تر می‌شود. بدین ترتیب یک مکانیزم ساده طبیعی توانسته است در طی چند نسل عملا افراد کم هوش را از جامعه حذف کند علاوه بر اینکه میزان هوش متوسط جامعه نیز دائما در حال افزایش است(البته امکان داشت اگر داروین بی‌عرضگی افراد باهوش امروزی را می‌دید کمی در تئوری خود تجدید نظر می‌کرد اما این مسئله دیگریست بدین ترتیب می‌توان دید که طبیعت با بهره‌گیری از یک روش بسیار ساده(حذف تدریجی گونه‌های نامناسب و در عین حال تکثیر بالاتر گونه‌های بهینه) توانسته است دائما هر نسل را از لحاظ خصوصیات مختلف ارتقا بخشد .البته آنچه در بالا ذکر شد به تنهایی توصیف کننده آنچه واقعا در قالب تکامل در طبیعت اتفاق می‌افتد نیست. بهینه‌سازی و تکامل تدریجی به خودی خود نمی‌تواند طبیعت را در دسترسی به بهترین نمونه‌ها یاری دهد. اجازه دهید تا این مساله را با یک مثال شرح دهیم .پس از اختراع اتومبیل به تدریج و در طی سال‌ها اتومبیل‌های بهتری با سرعت‌های بالاتر و قابلیت‌های بیشتر نسبت به نمونه‌های اولیه تولید شدند. طبیعیست که این نمونه‌های متاخر حاصل تلاش مهندسان طراح جهت بهینه‌سازی طراحی‌های قبلی بوده اند. اما دقت کنید که بهینه‌سازی یک اتومبیل تنها یک "اتومبیل بهتر" را نتیجه می‌دهد .اما آیا می‌توان گفت اختراع هواپیما نتیجه همین تلاش بوده است؟ یا فرضا می‌توان گفت فضا پیماها حاصل بهینه‌سازی طرح اولیه هواپیماها بوده‌اند؟پاسخ اینست که گرچه اختراع هواپیما قطعا تحت تاثیر دستاورهای صنعت اتومبیل بوده است اما به‌هیچ وجه نمی‌توان گفت که هواپیما صرفا حاصل بهینه‌سازی اتومبیل و یا فضا پیما حاصل بهینه‌سازی هواپیماست. در طبیعت هم عینا همین روند حکم‌فرماست. گونه‌های متکامل‌تری وجود دارند که نمی‌توان گفت صرفا حاصل تکامل تدریجی گونه قبلی هستند .در این میان آنچه شاید بتواند تا حدودی ما را در فهم این مساله یاری کند مفهومیست به نام : تصادف یا جهش به عبارتی طرح هواپیما نسبت به طرح اتومبیل یک جهش بود و نه یک حرکت تدریجی. در طبیعت نیز به همین گونه‌است. در

هر نسل جدید بعضی از خصوصیات به صورتی کاملا تصادفی تغییر می‌یابند سپس بر اثر تکامل تدریجی که پیشتر توضیح دادیم در صورتی که این خصوصیت تصادفی شرایط طبیعت را ارضا کند حفظ می‌شود در غیر این‌صورت به شکل اتوماتیک از چرخه طبیعت حذف می‌گردد .در واقع می‌توان تکامل طبیعی را به این‌صورت خلاصه کرد: جست‌وجوی کورکورانه(تصادف یا Blind Search)بقای قوی‌تر .

حال ببینیم که رابطه تکامل طبیعی با روش‌های هوش مصنوعی چیست .هدف اصلی روش‌های هوشمند به کار گرفته شده در هوش مصنوعی یافتن پاسخ بهینه مسائل مهندسی ست. بعنوان مثال اینکه چگونه یک موتور را طراحی کنیم تا بهترین بازدهی را داشته باشد یا چگونه بازوهای یک ربات را محرک کنیم تا کوتاه‌ترین مسیر را تا مقصد طی کند(دقت کنید که در صورت وجود مانع یافتن کوتاه‌ترین مسیر دیگر به سادگی کشیدن یک خط راست بین مبدا و مقصد نیست) همگی مسائل بهینه‌سازی هستند .

روش‌های کلاسیک ریاضیات دارای دو اشکال اساسی هستند. اغلب این روش‌ها نقطه بهینه محلی (Local Optima)را بعنوان نقطه بهینه کلی در نظر می‌گیرند و نیز هر یک از این روش‌ها تنها برای مساله خاصی کاربرد دارند. این دو نکته را با مثال‌های ساده‌ای روشن می‌کنیم .

به شکل زیر توجه کنید. این منحنی دارای دو نقطه ماکزیمم می‌باشد. که یکی از آنها تنها ماکزیمم محلی است. حال اگر از روش‌های بهینه‌سازی ریاضی استفاده کنیم مجبوریم تا در یک بازه بسیار کوچک مقدار ماکزیمم تابع را بیابیم. مثلا از نقطه ۱ شروع کنیم و تابع را ماکزیمم کنیم. بدیهی است اگر از نقطه ۱ شروع کنیم تنها به مقدار ماکزیمم محلی دست خواهیم یافت و الگوریتم ما پس از آن متوقف خواهد شد. اما در روش‌های هوشمند خاصه الگوریتم ژنتیک بدلیل خصلت تصادفی آنها حتی اگر هم از نقطه ۱ شروع کنیم باز ممکن است در میان راه نقطه A به صورت تصادفی انتخاب شود که در این صورت ما شانس دست‌یابی به نقطه بهینه کلی (Global Optima) را خواهیم داشت .در مورد نکته دوم باید بگوییم که روش‌های ریاضی بهینه‌سازی اغلب منجر به یک فرمول یا دستورالعمل خاص برای حل هر مسئله می‌شوند. در حالی که روش‌های

هوشمند دستورالعمل‌هایی هستند که به صورت کلی می‌توانند در حل هر مسئله‌ای به کار گرفته شوند. این نکته را پس از آشنایی با خود الگوریتم بیشتر و بهتر خواهید دید .

الگوریتم ژنتیک چیست؟

الگوریتم های ژنتیک از اصول انتخاب طبیعی داروین برای یافتن فرمول بهینه جهت پیش بینی یا تطبیق الگو استفاده می کنند.الگوریتم های ژنتیک اغلب گزینه خوبی برای تکنیک های پیش بینی بر مبنای رگرسیون هستند.همان طور ساده،خطی وپارامتریک گفته می شود،به الگوریتم های ژنتیک می توان غیر پارامتریک گفت.برای مثال اگر بخواهیم نوسانات قیمت نفت را با استفاده از عوامل خارجی وارزش رگرسیون خطی ساده مدل کنیم،این فرمول را تولید خواهیم کرد:قیمت نفت در زمان=t ضریب ۱ نرخ بهره در زمان +tضریب ۲ نرخ بیکاری در زمان+t ثابت ۱ . سپس از یک معیار برای پیدا کردن بهترین مجموعه ضرایب و ثابت ها جهت مدل کردن قیمت نفت استفاده خواهیم کرد.در این روش ۲ نکته اساسی وجود دارد.اول این روش خطی است و مسئله دوم این است که ما به جای اینکه در میان "فضای پارامترها"جستجو کنیم ،پارامترهای مورد استفاده را مشخص کرده ایم.با استفاده از الگوریتم های ژنتیک ما یک ابر فرمول یا طرح تنظیم می کنیم که چیزی شبیه"قیمت نفت در زمان t تابعی از حداکثر ۴ متغیر است"را بیان می کند. سپس داده هایی برای گروهی از متغیرهای مختلف،شاید در حدود ۲۰ متغیر فراهم خواهیم کرد.سپس الگوریتم ژنتیک اجرا خواهد شد که بهترین تابع و متغیرها را مورد جستجو قرار می دهد.روش کار الگوریتم ژنتیک به طور فریبنده ای ساده،خی

لی قابل درک وبه طور قابل ملاحظه ای روشی است که ما معتقدیم حیوانات آنگونه تکامل یافته اند.هر فرمولی که از طرح داده شده بالا تبعیت کند فردی از جمعیت فرمول های ممکن تلقی می شود خیلی شبیه به این که بگوییم جرج بوش فردی از جمعیت انسان های ممکن است .متغیر هایی که هر فرمول داده شده را مشخص می کنند به عنوان یکسری از اعداد نشان داده شده اند که معادل دی ان ای آن فرد را تشکیل می دهند.

موتور الگوریتم ژنتیک یک جمعیت آغاز از فرمول ایجاد می کند.هر فرد در برابر مجموعه ای از داده ها ی مورد آزمایش قرار می گیرند و مناسبترین آنها شاید ۱۰ درصد از مناسبترین ها باقی می مانند.بقیه کنار گذاشته می شوند. مناسبترین افراد با هم جفتگیری (جابجایی عناصر دی ان ای)وتغییر(تغییر تصادفی عناصر دی ان ای) کرده اند.مشاهده می شود که با گذشت از میان تعدد ریادی از نسلها،الگوریتم ژنتیک به سمت ایجاد فرمول هایی که بیشتر دقیق هستند،میل می کنند.در حالی که شبکه های عصبی هم غیر خطی و غیر پارامتریک هستند،جذابیت زیاد الگوریتم های ژنتیک این است نتایج نهایی قابل ملاحظه ترند.فرمول نهایی برای کاربر انسانی قابل مشاهده خواهد بود،و برای ارائه سطح اطمینان نتایج می توان تکنیک های آماری متعارف رابر روی این فرمول ها اعمال کرد.فناوری الگوریتم های ژنتیک همواره در حال بهبود استفبرای مثال با مطرح کردن معادله ویروس ها که در کنار فرمول ها وبرای نقض کردن فرمول ها ی ضعیف تولید می شوندودر نتیجه جمعیت را کلاً قویتر می سازند.مختصراً گفته می شود که الگوریتم ژنتیک یا GA یک تکنیک برنامه نویسی است که از تکامل ژنتیکی به عنوان یک الگوی حل مسئله استفاده می کند.مسئله ای که باید حل شود ورودی است و راه حلها طبق یک الگو کد گذاری می شودومتریک که تابع fitness هم نام دارد هر راه حل کاندید را ارزیابی می کندکه اکثر آنها به صورت تصادفی انتخاب می شوندالگوریتم ژنتیک GA یک تکنیک جستجو در علم کامپیوتربرای یافتن راه حل بهینه ومسائل جستجو است.الگوریتم های ژنتیک یکی از انواع الگوریتم های تکاملی اند که از علم زیست شناسی مثل وراثت، جهش،انتخاب ناگهانی ، انتخاب طبیعی و ترکیب الهام گرفته شده. عموماً راه حلها به صورت ۲ تایی ۰و۱ نشان داده می شوند ولی روشهای نمایش دیگری هم وجود دارد.تکامل از یک مجموعه کاملاً تصادفی از موجودیت ها شروع می شود و در نسلهای بعدی تکرار می شود.در هر نسل،مناسبترین ها انتخاب می شوند نه بهترین ها.یک راه حل برای مسئله مورد نظر،با یک لیست از پارامترها نشان داده می شود که به آنها کروموزوم یا ژنوم می گویند.کروموزوم ها عموماً به صورت یک رشته ساده از داده ها نمایش داده می شوند،البته انواع ساختمان داده های دیگر هم می توانند مورد استفاده قرار گیرند.در ابتدا چندین مشخصه به صورت تصادفی برای ایجاد نسل اول تولید می شوند. در طول هر نسل ،هر

مشخصه ارزیابی می شود وارزش تناسب (fitness)توسط تابع تناسب اندازه گیری می شود.گام بعدی ایجاد دومین نسل از جامعه است که بر پایه فرآیندهای انتخاب ،تولید از روی مشخصه های انتخاب شده با عملگرهای ژنتیکی است:اتصال کروموزوم ها به سر یکدیگر و تغییر.

برای هر فرد ،یک جفت والد انتخاب می شود.انتخابها به گونه ای اند که مناسبترین عناصر انتخاب شوند تا حتی ضعیفترین عناصر هم شانس انتخاب داشته باشند تا از نزدیک شدن به جواب محلی جلوگیری شود.چندین الگوی انتخاب وجود دارد: چرخ منگنه دار(رولت)،انتخاب مسابقه ای (Tournament) ،. ...

معمولاً الگوریتم های ژنتیک یک عدد احتمال اتصال دارد که بین ۰.۶و۱ است که احتمال به وجود آمدن فرزند را نشان می دهد.ارگانیسم ها با این احتمال با هم دوباره با هم ترکیب می شوند.اتصال ۲ کروموزوم فرزند ایجاد می کند،که به نسل بعدی اضافه می شوند.این کارها انجام می شوند تا این که کاندیدهای مناسبی برای جواب،در نسل بعدی پیدا شوند. مرحله بعدی تغییر دادن فرزندان جدید است.الگوریتم های ژنتیک یک احتمال تغییر کوچک وثابت دارند که معمولاً درجه ای در حدود ۰.۰۱ یا کمتر دارد. بر اساس این احتمال ،کروموزوم های فرزند به طور تصادفی تغییر می کنند یا جهش می یابند.مخصوصاً با جهش بیتها در کروموزوم ساختمان داده مان.

این فرآیند باعث به وجود آمدن نسل جدیدی از کروموزوم ها یی می شود، که با نسل قبلی متفاوت است.کل فرآیند برای نسل بعدی هم تکرار می شود،جفتها برای ترکیب انتخاب می شوند،جمعیت نسل سوم به وجود می آیند و. ...

این فرآیند تکرار می شود تا این که به آخرین مرحله برسیم.

شرایط خاتمه الگوریتم های ژنتیک عبارتند از:

- به تعداد ثابتی از نسل ها برسیم .
- بودجه اختصاص داده شده تمام شود(زمان محاسبه/ پول)
- یک فرد(فرزند تولید شده) پیدا شود که مینیمم (کمترین)ملاک را برآورده کند .

- بیشترین درجه برازش فرزندان حاصل شود یا دیگر نتایج بهتری حاصل نشود .
- بازرسی دستی .
- ترکیبهای بالا.

ایده اصلی :

در دهه هفتاد میلادی دانشمندی از دانشگاه میشیگان به نام جان هلند ایده استفاده از الگوریتم ژنتیک را در بهینه‌سازی‌های مهندسی مطرح کرد. ایده اساسی این الگوریتم انتقال خصوصیات موروثی توسط ژن‌هاست. فرض کنید مجموعه خصوصیات انسان توسط کروموزوم‌های او به نسل بعدی منتقل می‌شوند. هر ژن در این کروموزوم‌ها نماینده یک خصوصیت است. بعنوان مثال ژن ۱ می‌تواند رنگ چشم باشد ، ژن ۲ طول قد، ژن ۳ رنگ مو و الی آخر. حال اگر این کروموزوم به تمامی، به نسل بعد انتقال یابد، تمامی خصوصیات نسل بعدی شبیه به خصوصیات نسل قبل خواهد بود. بدیهیست که در عمل چنین اتفاقی رخ نمی‌دهد. در واقع بصورت همزمان دو اتفاق برای کروموزوم‌ها می‌افتد. اتفاق اول موتاسیون (Mutation) است .موتاسیون به این صورت است که بعضی ژن‌ها بصورت کاملا تصادفی تغییر می‌کنند. البته تعداد این گونه ژن‌ها بسیار کم می‌باشد اما در هر حال این تغییر تصادفی همانگونه که پیشتر دیدیم بسیار مهم است. مثلا ژن رنگ چشم می‌تواند بصورت تصادفی باعث شود تا در نسل بعدی یک نفر دارای چشمان سبز باشد. در حالی که تمامی نسل قبل دارای چشم قهوه‌ای بوده‌اند. علاوه بر موتاسیون اتفاق دیگری که می‌افتد و البته این اتفاق به تعداد بسیار بیشتری نسبت به موتاسیون رخ می‌دهد چسبیدن ابتدای یک کروموزوم به انتهای یک کروموزوم دیگر است. این مساله با نام Crossoverشناخته می‌شود. این همان چیزیست که مثلا باعث می‌شود تا فرزند تعدادی از خصوصیات پدر و تعدادی از خصوصیات مادر را با هم به ارث ببرد و از شبیه شدن تام فرزند به تنها یکی از والدین جلوگیری می‌کند .

در ابتدا تعداد مشخصی از ورودی ها، $X_1, X_2, \ldots, X_n$ که متعلق به فضای نمونه X هستند را انتخاب می کنیم و آنها را در یک عدد بردای $X=(x1,x2,\ldots xn)$ نمایش می دهیم..در مهندسی نرم افزار اصطلاحاً به

آنها ارگانیسم یا کروموزوم گفته می شود.به گروه کروموزوم ها Colony یا جمعیت می گوییم.در هر دوره Colonyرشد می کند و بر اساس قوانین مشخصی که حاکی از تکامل زیستی است تکامل می یابند.

برای هر کروموزوم Xi ،ما یک ارزش تناسب (Fitness)داریم که آن را f(Xi) هم می نامیم.عناصر قویتر یا کروموزوم هایی که ارزش تناسب آنها به بهینه Colony نزدیکتر است شانس بیشتری برای زنده ماندن در طول دوره های دیگر و دوباره تولید شدن را دارند و ضعیفترها محکوم به نابودی اند. به عبارت دیگر الگوریتم ورودی هایی که به جواب بهینه نزدیکترندرانگه داشته واز بقیه صرف نظر می کند.

یک گام مهم دیگر درالگوریتم،تولد است که در هر دوره یکبار اتفاق می افتد. محتویات دو کروموزومی که در فرآیند تولید شرکت می کنند با هم ترکیب میشوند تا ۲ کروموزوم جدید که ما انها را فرزند می نامیم ایجاد کنند.این هیوریستیک به ما اجازه می دهد تا ۲ تا از بهترین ها را برای ایجاد یکی بهتر از آنها با هم ترکیب کنیم (evolution).به علاوه در طول هر دوره،یک سری از کروموزوم ها ممکن است جهش یابند (Mutation)

الگوریتم :

هر ورودی x در یک عدد برداری $X=(x_1,x_2,..,x_n)$ قرار دارد .برای اجرای الگوریتم ژنتیک مان باید هر ورودی را به یک کروموزوم تبدیل کنیم.می توانیم این را با داشتن log(n) بیت برای هر عنصرو تبدیل ارزش Xiانجام دهیم مثل شکل زیر .

e(X۱)

e(X۱)

e(X۱)

۰۱۱۱۱۱۱ ... ۱۰۱۰۱۱۱ ۱۱۱۱۰۱۱

(X۱, X۲,…,Xn)= (۱۲۳, ۸۷,…, ۶۳)

می توانیم از هر روش کد کردن برای اعداد استفاده کنیم.در دوره ۰، یک دسته از ورودی های X را به صورت تصادفی انتخالب می کنیم.بعد برای هر دورهi ام ما ارزش مقدار Fitness را تولید،تغییر وانتخاب را اعمال می کنیم.الگوریتم وقتی پایان می یابد که به معیارمان برسیم .

سود و کد :

Choose initial population

Repeat

Evaluate the individual fit nesses of a certain proportion of the population

Select pairs of best-ranking individuals to reproduce

Apply crossover operator

Apply mutation operator

Until terminating condition

روش های نمایش :

قبل از این که یک الگوریتم ژنتیک برای یک مسئله اجرا شود،یک روش برای کد کردن ژنوم ها به زبان کامپیوتر باید به کار رود. یکی از روش های معمول کد کردن به صورت رشته های باینری است:رشته های ۰و۱. یک راه حل مشابه دیگر کدکردن راه حل ها در آرایه ای از اعداد صحیح یا اعشاری است ،که دوباره هر جایگاه یک جنبه از ویژگی ها را نشان می دهد.این راه حل در مقایسه با قبلی پیچیده تر و مشکل تر است. مثلاً این روش توسط استفان کرمر،برای حدس ساختار ۳ بعدی یک پروتئین موجود در آمینو اسید ها استفاده شد.الگوریتم های ژنتیکی که برای آموزش شبکه های عصبی استفاده می شوند،از این روش بهره می گیرند.سومین روش برای نمایش صفات در یک GA یک رشته از حروف است،که هر حرف دوباره نمایش دهنده یک خصوصیت از راه حل است.خاصیت هر ۳تای این روشها این است که آنها تعریف سازنده ایی را که تغییرات

تصادفی در آنها ایجاد می کنند را آسان می کنند:۰را به ۱ وبرعکس،اضافه یا کم کردن ارزش یک عدد یا تبدیل یک حرف به حرف دیگر.یک روش دیگر که توسط John Koza توسعه یافت،برنامه نویسی ژنتیک (Genetic programming)است.که برنامه ها را به عنوان شاخه های داده در ساختار درخت نشان می دهد.در این روش تغییرات تصادفی می توانند با عوض کردن عملگرها یا تغییر دادن ارزش یک گره داده شده در درخت،یا عوض کردن یک زیر درخت با دیگری به وجود آیند .

روش های انتخاب :

روش های مختلفی برای الگوریتم های ژنتیک وجود دارند که می توان برای انتخاب ژنوم ها از آنها استفاده کرد.اما روش های لیست شده در پایین از معمولترین روش ها هستند.

انتخاب: Elitist مناسبترین عضو هر اجتماع انتخاب می شود.

انتخاب :Roulette یک روش انتخاب است که در آن عنصری که عدد برازش(تناسب)بیشتری داشته باشد،انتخاب می شود.

انتخاب: Scaling به موازات افزایش متوسط عدد برازش جامعه،سنگینی انتخاب هم بیشتر می شودوجزئی تر.این روش وقتی کاربرد دارد که مجموعه دارای عناصری باشد که عدد برازش بزرگی دارند وفقط تفاوت های کوچکی آنها را از هم تفکیک می کند.

انتخاب : Tournament یک زیر مجموعه از صفات یک جامعه انتخاب می شوندواعضای آن مجموعه با هم رقابت می کنندو سرانجام فقط یک صفت از هر زیر گروه برای تولید انتخاب می شوند.

بعضی از روشهای دیگر عبارتند از :

:Rank Selection, Generational Selection, Steady-State Selection Hierarchical Selection.

روش های تغییر:

وقتی با روش های انتخاب کروموزوم ها انتخاب شدند،باید به طور تصادفی برای افزایش تناسبشان اصلاح شوند.۲ راه حل اساسی برای این کار وجود دارد.اولین وساده ترین جهش (Mutation) نامیده می شود.درست مثل جهش در موجودات زنده که عبارت است از تغییر یک ژن به دیگری، در الگوریتم ژنتیک جهش تغییر کوچکی در یک نقطه از کد خصوصیات ایجاد می کند.

دومین روش Crossover نام دارد و ۲ کروموزوم برای معاوضه سگمنتهای کدشان انتخاب می شوند.این فرآیند بر اساس فرآیند ترکیب کروموزوم ها در طول تولید مثل در موجودات زنده شبیه سازی شده. اغلب روش های معمول Crossover شامل Single-point Crossover هستند ، که نقطه تعویض در جایی تصادفی بین ژنوم ها است.بخش اول قبل از نقطه ،و بخش دوم سگمنت بعد از آن ادامه پیدا می کند،که هر قسمت برگرفته از یک والد است،که ۵۰/۵۰ انتخاب شده.

شکل های بالا تاثیر هر یک از عملگر های ژنتیک را روی کروموزوم های ۸ بیتی نشان می دهد. شکل بالاتر ۲ ژنوم را نشان می دهد که نقطه تعویض بین ۵امین و ۶امین مکان در ژنوم قرار گرفته،ایجاد یک ژنوم جدید از پیوند این ۲ والد بدست می آیند.شکل ۲وم ژنومی را نشان می دهد که دچار جهش شده و ۰ در آن مکان به ۱ تبدیل شده.

تقاط قوت الگوریتم های ژنتیک:

اولین و مهمترین نقطه قوت این الگوریتم ها این است که الگوریتم های ژنتیک ذاتاً موازی اند .اکثر الگوریتم های دیگر موازی نیستند و فقط می توانند فضای مسئله مورد نظر را در یک جهت در یک لحظه جستجو کنند واگر راه حل پیدا شده یک جواب بهینه محلی باشدویا زیر مجموعه ای از جواب اصلی باشد باید تمام کارهایی که تا به حال انجام شده را کنار گذاشت ودوباره از اول شروع کرد.از آنجایی که GA چندین نقطه شروع دارد،در یک لحظه می تواند فضای مسئله را از چندجهت مختلف جستجو کند. اگر یکی به نتیجه نرسید سایر راه ها ادامه می یابند و منابع بیشتری را در اختیار شان قرار می گیرد.در نظر بگیرید: همه ۸ عدد رشته

باینری یک فضای جستجو را تشکیل می دهند،که می تواند به صورت ******** نشان داده شود.رشته ۰۱۱۰۱۰۱۰یکی از اعضای این فضاست.همچنین عضوی از فضاهای

********و*******۰۱و۰۰******* و*۱*۱*۱*۰و ۰**۰۱*۰۱ والی آخر باشد.

به دلیل موازی بودن واین که چندین رشته در یک لحظه مورد ارزیابی قرار می گیرند GA ها برای مسائلی که فضای راه حل بزرگی دارند بسیار مفید است .اکثر مسائلی که این گونه اند به عنوان"غیر خطی" شناخته شده اند.در یک مسئله خطی، Fitness هر عنصر مستقل است،پس هر تغییری در یک قسمت بر تغییر وپیشرفت کل سیستم تاثیر مستقیم دارد.می دانیم که تعداد کمی از مسائل دنیای واقعی به صورت خطی اند.در مسائل غیر خطی تغییر در یک قسمت ممکن است تاثیری ناهماهنگ بر کل سیستم ویا تغییر در چند عنصر تاثیر فراوانی بر سیستم بگذارد .خوشبختانه موازی بودن GA باعث حل این مسئله می شود ودر مدت کمی مشکل حل می شود.مثلاً برای حل یک مسئله خطی ۱۰۰۰ رقمی ۲۰۰۰ امکان حل وجود دارد ولی برای یک غیر خطی ۱۰۰۰ رقمی $۲^{۱۰۰۰}$امکان. یکی از نقاط قوت الگوریتم های ژنتیک که در ابتدا یک کمبود به نظر می رسد این است که GA: ها هیچ چیزی در مورد مسائلی که حل می کنند نمی دانندو اصطلاحاً به آنها Blind Watchmakers می گوییم . آنها تغییرات تصادفی را در راه حل های کاندیدشان می دهند وسپس از تابع برازش برای سنجش این که آیا آن تغییرات پیشرفتی ایجاد کرده اند یا نه، استفاده می کنند.مزیت این تکنیک این است که به GA اجازه می دهند یا با ذهنی باز شروع به حل کنند.از آنجایی که تصمیمات آن اساساً تصادفی است،بر اساس تئوری همه راه حلهای ممکن به روی مسئله باز است،ولی مسائلی که محدود به اطلاعات هستند باید از راه قیاس تصمیم بگیرند ودر این صورت بسیاری از راه حلهای نو وجدید را از دست می دهند . یکی دیگر از مزایای الگوریتم ژنتیک این است که آنها می توانند چندین پارامتر را همزمان تغییردهند.بسیاری ازمسائل واقعی نمی توانند محدود به یک ویژگی شوند تا آن ویژگی ماکسیمم شود یا مینیمم و باید چند جانبه در نظر گرفته شوندGAها در حل این گونه مسائل بسیار مفیدند،و در حقیقت قابلیت موازی کار کردن آنها این

خاصیت را به آنها می بخشد.و ممکن است برای یک مسئله ۲ یا چند راه حل پیدا شود،که هر کدام با در نظر گرفتن یک پارامتر خاص به جواب رسیده اند.

محدودیتهای GA ها :

یک مشکل چگونگی نوشتن عملگر Fitness است که منجر به بهترین راه حل برای مسئله شود.اگر این کارکرد برازش به خوبی و قوی انتخاب نشود ممکن است باعث شود که راه حلی برای مسئله پیدا نکنیم یا مسئله ای دیگر را به اشتباه حل کنیم. به علاوه برای انتخاب تابع مناسب برای Fitness ،پارامترهای دیگری مثل اندازه جمعیت،نرخ جهش و Crossover، قدرت ونوع انتخاب هم باید مورد توجه قرار گیرند.

مشکل دیگر،که آن را نارس می نامیم این است که اگر یک ژنوم که فاصله اش با سایر ژنوم های نسل اش زیاد باشد(خیلی بهتر از بقیه باشد)و خیلی زود دیده شود(ایجاد شود)ممکن است محدودیت ایجاد کند و راه حل را به سوی جواب بهینه محلی سوق دهد.این اتفاق معمولاً در جمعیت های کم اتفاق می افتد.روش های Rank ,Scaling tournament selectionبر این مشکل غلبه می کنند

چند نمونه از کاربرد های الگوریتم های ژنتیک :

نرم‌افزار شناسایی چهره با استفاده از تصویر ثبت شده به همت مبتکران ایرانی طراحی و ساخته شد .در این روش، شناسایی چهره براساس فاصله اجزای چهره و ویژگی‌های محلی و هندسی صورت می‌گیرد که تغییرات ناشی از گیم، تغییرات نور و افزایش سن کمتین تأثیر را خواهد داشت .همچنین گراف‌ها برای چهره‌های جدید با استفاده از الگویتم‌های ژنتیک ساخته شده و با استفاده از یک تابع تشابه، قابل مقایسه با یکدیگر هستند که این امر تأثیر به‌سزایی در افزایش سرعت شناسایی خواهد داشت .

توپولوژی های شبکه های کامپیوتی توزیع شده.

بهینه سازی ساختار ملکولی شِمیایی(شیمی)

مهندسی برق برای ساخت آنتنهایCrooked-Wire Genetic Antenna

مهندسی نرم افزار

بازی های کامپیوتری

مهندسی مواد

مهندسی سیستم

رباتیک(Robotics)

تشخیص الگوو استخراج داده(Data mining)

حل مسئله فروشنده دوره گرد

آموزش شبکه های عصبی مصنوعی

یاددهی رفتار به رباتها با GA

یادگیری قوانین فازی با استفاده از الگویتم های ژنتیک.

یک مثال ساده:

ما یک مربع ۳*۳ داریم که می خواهیم اعدادی بین ۱تا۱۵ را در این مربع قرار دهیم به طوری که جمع اعداد در هر سطرو ستون برابر ۲۴ شود .

=۲۴
۲۴

۲۴N

ابن مسئله تا حدودی پیچیده است.ممکن است یک انسان بتواند آن را در مدت زمانی مشخص حل کند ولی هیچ گاه یک کامپیوتر نخواهد توانست آن رادر مدت زمان کوتاهی با استفاده از اعداد تصادفی حل کند. ولی الگوریتم ژنتیک می تواند این مشکل را حل کند.

نسل اول :

اولین گام ایجاد کردن یک نسل ابتدایی برای شروع کار است که شامل تعدادی ژنوم تصادفی است.این ژنوم ها به صورت باینری(۰و۱) نشان داده می شوند. حالا مثال مان:

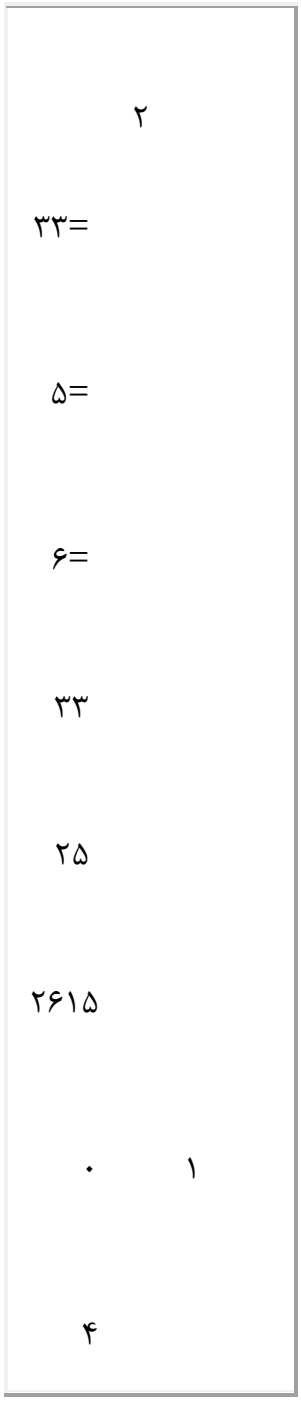
اول یکسری عدد به صورت تصادفی تولید می شوند. هر ژنوم یا کروموزوم شامل اطلاعاتی برای هر ۹ جای خالی است .چون این اعداد مقادیر بین ۰تا۱۵ دارند می توان آنها را با ۴ بیت یا ژن داده نمایش داد. پس هر ژنوم شامل ۳۶ بیت است .

یک نمونه ژنوم می تواند به شکل زیر باشد:

Bits (Genes)	۰۱۱۰	۱۱۰۰	۱۱۱۱	۱۰۱۱	۰۱۰۰	۱۰۱۰	۰۱۱۱	۰۱۰۱	۱۱۱۰
Values(Traits)	۶	۱۲	۱۵	۱۱	۴	۱۰	۷	۵	۱۴

حالاباید به هر ژنوم در مجموعه یک عدد تناسب (Fitness)بنابر تاثیر آن در حل مسئله نسبت داد.فرآیند وروش محاسبه این عدد برای هر مسئله فرق می کند.انتخاب الگوی مناسب برای مسئله مشکلترین و

حساسترین بخش در حل مسئله ژنتیک است.دراین مثال ما اعداد را در مکان هایشان جایگذاری می کنیم و بررسی می کنیم که چقدر با جواب اصلی فاصله دارند .



مقادیر معادل عبارتند از۳۳ و۲۵و۲۶و۲۴و۲۱و۳۹.واضح است که این مقادیر مسئله را حل نمی کنند.پس باید مقادیر تناسب را برای این ژنوم محاسبه کرد.برای این کار ابتدا فاصله هرمجموع را از۲۴ محاسبه کرده،سپس معکوس مجموع تفاصل آنها را محاسبه می کنیم .بنابراین درجه تناسب برای این ژنوم تقریباً برابر ۰.۰۳۳ است.هرچقدر که اعداد ما به جواب نزدیکتر باشند عدد تناسب بزرگتر خواهد شد.اما اگر مخرج ما برابر ۰شود چه اتفاقی می افتد؟ دراین صورت همه اعداد ما برابر ۲۴ شده اند وما به جواب رسیده ایم.

نسل بعدی :

دو ژنوم به طور تصادفی برای تولید نسل بعدی انتخاب می شوند. این اصلی ترین بخش الگوریتم ژنتیک است که از ۳ مرحله تشکیل شده:

انتخاب :

دو ژنوم به طور تصادفی از نسل قبل انتخاب می شوند.این ژنوم ها دارای اعداد تناسب بزرگتری هستند و بعضی صفات آنها به نسل بعدی منتقل می شوند. این بدین معنی است که عدد تناسب در حال افزایش خواهد بود.بهترین روش برای تابع انتخاب (Fitness)در این مسئله روشی به نام رولت (Roulette)است.اول یک عدد تصادفی بین صفر وعدد تناسب نسل قبلی انتخاب می شود. تابع انتخاب به صورت زیر خواهد بود:

```
RouletteSelection()
{
   float ball  = rand_float_between(۰.۰, total_fitness);
   float slice = ۰.۰;
   for each genome in population
   {
```

```
            slice += genome. fitness;
            if ball < slice
               return genome;
         }
      }
```

تغییر از یک نسل به نسل بعدی (Cross over):

حالا دو ژنوم بخشی از ژنهایشان را برای ایجاد نسل بعدی اهدا می کنند. اگر آنها تغییر پیدا نکنند همانطور بی تغییر به نسل بعدی منتقل خواهند شد.درجه Crossover نشان دهنده این است که هر چند وقت یکبار ژنوم ها تغییر پیدا خواهند کرد و این عدد باید در حدود ۶۵-۸۵ %باشد.

عملگر تغییر در ژنوم های باینری مثال ما با انتخاب یک مکان تصادفی در ژنوم برای تغییر آغاز می شود. بخش اول ژنهای پدر و بخش دوم ژنهای مادر با هم ترکیب می شوند(و بالعکس) تا۲ فرزند تولید شوند. در زیریک عمل تغییر را می بینیم.

Before Crossing

Father ۰۱۱۱۱۰۰۱۰۰۱۱ ۰۰۱۰۱۱۰۱۱۰۰۰۱۱۱۰۱۱۰۱۰۰۰۰

Mother ۰۱۰۱۰۰۱۱۱۱۱۰ ۰۱۰۱۰۱۱۱۱۱۰۱۰۰۰۱۰۰۰۱۰۰۱۰

After Crossing

Child۱ ۰۱۱۱۱۰۰۱۰۰۱۱ ۰۱۰۱۰۱۱۱۱۱۰۱۰۰۰۱۰۰۰۱۰۰۱۰

Child۲ ۰۱۰۱۰۰۱۱۱۱۱۰ ۰۰۱۰۱۱۰۱۱۰۰۰۱۱۱۰۱۱۰۱۰۰۰۰

جهش (Mutation):

قبل از این که ژنوم ها در نسل بعدی قرار بگیرند،احتمال دارد دچار جهش یا تغییر ناگهانی شوند شوند.جهش یک تغییر ناگهانی در ژن است.در ژنهای باینری این تغییر به معنای تغییر یک بیت از ۰به ۱ یا از ۱ به ۰ است. درجه جهش نشان دهنده احتمال بروز جهش در یک ژن است و تغریباً بین ۱-۵٪ برای ژنهای باینری و ۵-۲۰٪برای ژنهای عددی است.

نمونه برنامه این مسئله به زبان C++ همراه این تحقیق آورده شده است

هایپر هیوریستیک :

مثال های بسیاری وجود دارد که برای حل ان ها از الگوریتم ژنتیک استفاده شده است از جمله Traveling salesman& bin packing&scheduling مسئله ی جدول زمانی پرسنل وکارکنان با استفاده از الگوریتم ژنتیک به صورت موفقیت امیزی حل شده است aickelin& dowslamd. از الگوریتم ژنتیک لیست کار پرستاران در یکی از بیمارستان های بزرگ انگلستان به کا ر برده شد eston & mansour از الگوریتم ژنتیک برای حل مسئله ی جدول زمان بندی کار استفاده کردند. در واقع برای حل این مسئله از الگوریم ژنتیک توزیع شده که به صورت موازی روی شبکه ی محل کار وجود داشت دست یافتند.آنچه از این تحقیق بدست آمد سه دسته ی متفاوت زمان بندی برای مجموعه ای از کارهای متفاوت بود. این محققین مسائلی را که با هیوریستیک و متا هیوریستیک کار میکرند با هم مقایسه کردند و نتیجه ان بود که الگوریتم ژنتیک بهتر از همه ی آنها کار می کرد در نتیجه الگوریتم های ژنتیک با کروموزوم های مستقیم و کروموزوم های غیر مستقیم به صورت گسترده ای مورد مطالعه قرار گرفت . به عنوان مثال hart & ross& nelson از الگوریتم ژنتیک با کروموزوم های مستقیم برای حل مسئله ی زمان بندی امتحانات طراحی شد. انه استراتژی های به عنوان پارامتر در ده خانه ای ارایه ایجاد کردند. بنابراین کروموزوم ها ساختار جدول زمان بندی را ایجاد می کنند نه خود جدول زمان بندی .کروموزوم های غیر مستقیم برای رفع محدودیت های کروموزوم های مستقیم مورد استفاده قرار می گیرند. که این محدودیت همان شکست هماهنگ بین قسمت های مختلف حل مسئله وجود داشت.

از الگوریم ژنتیک بیشتر در مسائلی مورد استفاده قرار می گرفت که روی مسئله ی زمان و زمان بندی تاکید داشت و در این نوع مسائل ما نیاز به دامنه ی دانش مسئله داریم . در واقع کروموزوم ها که به عنوان ساختار حل مسئله بودند دانش مسئله در طراحی کروموزوم ها در الگوریتم ژنتیک بسیار لازم و ضروری است . وابستگی زیاد این طراحی به دامنه ی دانش مسئله موجب می شود تا نتوانیم الگوریتم ژنتیک بدست آمده در مسائل دیگر به کار ببریم.الگوریتم های ژنتیک که از کروموزوم های غیر مستقیم بر پایه ی سلسله مراتب تکاملی هیوریستیک طراحی شده است را در مسائلی همچون مسئله ی زمان بندی کار پرسنل می توان به صورت عمومی ودر همه ی مکان ها استفاده کرد.متد هایپر هیوریستیک که از ان در مسئله ی حمل و نقل مرغ های زنده تو سعه و تکامل دادنداین مسئله به این شیوه حل شد که این مسئله را به دو زیر مسئله تقسم کردند هر یک از این دو زیر مسئله از الگوریتم زنتیک جدا استفاده میکردند. هر کدام از این دو الگوریتم ژنتیک یک استراتژی برای تولید برنامه معین می کنند نه اینکه خودشان یک برنامه باشند تمامی اطلاعات شرکت به عنوان یکسری قانون است که به وسیله ی توانایی جستجوی الگوریتم ژنتیک پشتیبانی می شود.سلسله مراتب در الگوریتم هیوریستیک معین میکند که کدام هیوریستیک برای قرار دادن کارها در برنامه استفاده شود. در نتیجه ی این تحقیق ساختار برنامه ی زمان بندی برای حمل و نقل تعداد زیادی مرغ زنده در یک کار خانه طراحی شد.الگوریتم های هایپر هیوریستیک که با الگوریتم های زنتیک با کرو موزوم های غیر مستقیم طراحی کرده ایم .که در این نوع الگوریتم کروموزوم ها دارای اندازه های متغیر انداین نوع الگوریتم هم برای مسائل زمان بندی مورد استفاده قرار می گیرد.الگوریتم هایپرهیوریستیک غیر مستقیم الگوریتمی است که از یکسری هیوریستیک سطح پایین و یکسری هیوریستیک سطح بالا که هایپر هیوریستیک که فقط میداند که کی توابع هدف ماکسیمم یا مینیمم هستندو هیچ اطلاعاتی در مورد اینکه تابع هدف چه چیزی را نشان میدهد ندارد و هیچ گونه دامنه ی دانشی در هایپر هیوریستیک و هیوریستیک های سطح پایین مرتبط با ان موجود نمی باشد . همچنین با توجه به چهارچوب کلی یکسری توابع انتخاب داریم که که تصمیم میگیرند کدام هیوریستیک را صدا بزنند .{۸}

الگوریتم ژنتیکی ساده :

معرفی الگوریتم :

گلد برگ ، الگوریتم ژنتیکی هلند را با عنوان الگوریتم ژنتیکی ساده معرفی می کند. این الگوریتم از نظر محاسباتی ساده و در عین حال جستجویشان قدرتمند می باشند. الگوریتم های ژنیتکی اصطلاحات فنی خود را از ژنتیک طبیعی اقتباس کرده اند. در الگوریتم ژنتیکی ، مجموعه ای از متغییرهای طراحی را توسط رشته هایی با طول ثابت یا متغیر کدگذاری می کنند که در سیستم های بیولوژیکی آن را کروموزوم می نامند.هر رشته نشان دهنده یک قطعه پاسخ در فضای جستجو است . به ساختمان رشته ها (مجموعه ای از پارامترها که توسط یک کروموزوم خاص نمایش داده می شوند) ژنوتیپ و به مقدار رمزگشایی شده آن فنوتیپ گویند الگوریتمهای ژنتیکی فرآیندی تکراری هستند که هر مرحله تکراری را نسل و مجموعه ای از پاسخها در هر نسل را جمعیت نام نهاده اند.

الگوریتم های ژنتیکی ، جستجوی اصلی را در فضای پاسخ به اجراء می گذارند این الگوریتم ها با تولید نسل آغاز می شود که وظیفه ایجاد مجموعه نقاط جستجوی اولیه ای به نام جمعیت اولیه را به عهده دارند و به طور انتخابی یا تصادفی تعیین می شوند از آنجا که الگوریتم های ژنتیکی برای هدایت عملیات جستجو به طرف نقطه بهینه از روشهای آماری استفاده می کنند ، در فرآیندی که به انتخاب طبیعی وابسته است جمعیت موجود به تناسب برازندگی افراد آن برای نسل بعد انتخاب می شوند. پس از این جمعیت جدیدی جایگزین جمعیت پیشین می شود و این چرخه ادامه می یابد. معمولاً جمعیت جدید ، برازندگی بیشتری دارد این بدان معناست که از نسلی به نسل دیگر جمعیت بهبود می یابد. هنگامی جستجو نتیجه بخش خواهد بود که به بیشینه نسل مورد نظر رسیده باشیم یا همگرایی حاصل شده باشد یا معیارهای توقف برآورده شده باشد.

محدوده کاربرد الگوریتمهای ژنتیکی :

الگوریتم های ژنتیکی برای حل مسائل بهینه کردن عددی که حل کلاسیک آنها مشکل است مفید می‌باشند. درجه مشکل بودن با دو فاکتور تعیین می‌شود. اندازه فضای جستجو و حضور زیرمجموعه‌های بهینه. اگرچه الگوریتم ژنتیکی در زمینه های علمی زیادی به کار گرفته شده است ولی این کاربردها به طور کلی به سه دسته مهم تقسیم می‌شوند. مسائل عددی، مسائل توالی و مسائل انتخاب زیر مجموعه (انتخاب متغیر).

مسائل دسته اول شامل حل مدهای محاسباتی عددی هستند. به عنوان مثال در شیمی می‌توان از فیت کردن منحنی یک طیف IR با استفاده از یک تعداد از پیکهای گوسین نام برد.توالی مثل تعیین طیف NMR پروتئین‌ها یا طراحی توالی ستون‌های متغیر، یک مثال از دسته سوم، انتخاب متغیر می‌باشد. برای مثال می‌توان از انتخاب تعدادی طول موج از یک طیف برای کالیبراسیون چند متغیره نام برد.

اصول الگوریتم های ژنتیکی :

تکامل تدریجی بیولوژیکی در طراحی و خلق ارگانیسم های پیچیده، موفق بوده است. مکانیسم های زیر برای ارزیابی بیولوژیکی مهم بوده است :

قوانین کلاسیک داروین درباره تکامل تدریجی طبیعی: تلاش برای زندگی (قانون بقای نسل) این قوانین کلاسیک برروی یک فرم کددارشده از زندگی یعنی کروموزوم و نه برروی خود موجود زنده عمل می‌کنند.

تغییرات اتفاقی به وسیله جهش طبیعی ایجاد می‌شود.

این موضوع دانشمندان زیادی را تشویق کرده است تا این فرآیندها را در کامپیوتر شبیه سازی کنند. و به این وسیله محاسباتی پیچیده را حل کنند. هولاند یکی از پیشکسوتان در توسعه الگوریتم‌های ژنتیکی بوده است. برای شبیه سازی تکامل تدریجی به منظور بهینه سازی، مؤلفه‌های تکامل تدریجی مناسبی باید طراحی شوند.

یک تکنیک کددار کردن برای حلهای ارائه شده:

رقابت بین این حلهای ارائه شده بازترکیبی بین حلهای بقایافته به طوری که یک نسل جدید از حل های بهتر بتوانند انتخاب شوند و جایگزین حل های حاضر شود.

جهش: تغییرات اتفاقی:

در اینجا برروی الگوریتم ژنتیکی مرسوم که الگوریتم ژنتیکی ساده (SGA) نامیده می‌شود و به وسیله هولند معرفی شد متمرکز می‌شویم.

حلهای کاندید شده :

حلهای انتخاب شده مسئله باید در کامپیوتر به صورت کددار در آید . در الگوریتم ژنتیکی مرسوم ، این ها رشته های بایتی شامل صفر و یک هستند که بعداً با الگوریتم ساخته می شوند.حلهای انتخاب شده را می توان به صورتهای مختلف نشان داد. مقادیر پارامتری ممکن است مقیاس بندی شوند و به صورت مقادیر عددی صحیح کدگذاری شوند.معمول ترین روش ،کد دار کردن دودویی مقادیر پارامترها است که در الگوریتم ژنتیکی مورد استفاده قرار می گیرد. در این روش کددار کردن، کاراکترها در یک رشته بایتی ،صفر و یک هستند. به طور معمول همه رشته ها(حلهای انتخاب شده)طول مشابهی دارند. هر زیر قسمت در رشته یک پارامتر مورد نظر را بیان می کند و یک طول مینیمم یک بایتی دارد.از نظر انسان کدار کردن با اعداد حقیقی بهتر به نظر می رسد زیرا انسان بیشتر در مورد حل با اعداد حقیقی فکر می کند. این باعث می شود که بتوان مساله را به صورت محسوس تر بیان کرد. بیشتر پژوهشگران در زمینه الگوریتم ژنتیکی،تکنیک کددار کردن حقیقی را توصیه نمی کنند. یک تکنیک دیگر که بیشتر استفاده می شود کددار کردن گری است. یک گری کد، اعداد صحیح (۱...و۱- ۲ N) را به عنوان یک رشته دودویی با طول N بیان می کند. به طوری که بیان گری ، عدد صحیح مجاور فقط در یک موقعیت بایتی فرق می کند. حرکت در طول رشته (توالی) عددی صحیح بنابر این نیاز به تغییر فقط یک بایت در زمان دارد.

مثل پیدا کردن پیکربندی سه بعدی یک بیوماکرومولکول که با استفاده از طیف NMR که به صورت تجربی بدست آمده است سازگار است. به دلیل اینکه پیکربندی‌های مختلفی با چرخش پیچشی حول پیوندها ایجاد می‌شود به راحتی می‌توان پیکربندی سه بعدی ملکول را با استفاده از زوایای پیچشی در ملکول بیان نمود. در نتیجه حلهای انتخاب شده شامل یک تعداد از مقادیری است که زوایای پیچشی مختلف را بیان می‌کند.

مثال بعدی فیت کردن منحنی است. تنها راه فیت کردن توابع پیچیده (که دارای پارامترهای غیرخطی هستند) برروی داده ها این است که از اسراتژی تکرار استفاده شود. این استراتژی نیاز به اطلاعات زیادی در مورد مقادیر پارامترها دارد. اگر مقادیر اولیه پارامترها تا حد کافی به مقدار صحیح نزدیک باشد روش در یک بهینه محلی پایان می‌باشد که منتج به یک فیت کردن نامناسب می‌شود. الگوریتم های ژنتیکی اجازه استفاده از اطلاعات کمتری درباره مقادیر پارامترها را می‌دهند.

شروع:

مرحله اول در الگوریتم ژنتیکی ایجاد یک تعداد حل اولیه است. بدون اطلاع قبلی، حلهای انتخاب شده به صورت نقاطی انتخاب می‌شوند. در این جا همه فضای جستجو به صورت مساوی گسترش داده می‌شود. برای پرهیز از اینکه فضای جستجو خیلی بزرگ شود مفید است که از اطلاعات قبلی استفاده کنیم. تعداد حلهای انتخاب شده معمولاً در گستره ۵۰ تا ۱۰۰ برای بیشتر مسائل انتخاب می شود. این مجموعه از حلها یا جمعیت های انتخاب شده اولین نسل را ایجاد می‌کند. از این نسل اولیه، نسل‌های بعدی ایجاد می‌شود. عقیده براین است که بالاخره در نسل نهایی، یک حل انتخاب شده که حل بهینه مسئله است بدست می‌آید.

ارزیابی:

مرحله بعدی در الگوریتم ژنتیکی ارزیابی کیفیت حلهای انتخاب شده است. این نیاز به یک محک برای ارزیابی یا در اصطلاح الگوریتم ژنتیکی، یک مقدار شایستگی (fitness value) دارد. مقدار هدف یا بهینه محک ارزیابی می‌بایست تعریف شود. بدلیل اینکه قضاوت در مورد یک حل نیاز به اطلاعات حدودی دارد این مرحله

نسبت به همه الگوریتم ها به آن حدود وابسته تر است. برای مثال NMR ممکن است ک طیف NMR متعلق به هر حل انتخاب شده را محاسبه کنیم. اختلاف کوچکتر بین طیف تجربی و محاسبه شده، نشانه نزدیکی به پیکربندی واقعی است. اختلاف مربع میانگین بین دو طیف به عنوان یک محک ارزیابی استفاده می‌شود. مقدار هدف برای مقدار شایستگی صفر یا عددی کوچک، بسته به فضای تجربی است. به طور کلی مقدار شایستگی به وسیله یک تابع ارزیابی به دست می آید:

$$F(x) = g(f(x))$$

که f ، تابع هدف بوده، g مقادیر تابع هدف را به یک مقدار غیر منفی تبدیل می کندو F مقدار برازندگی نسبی است.در مرحله ارزیابی باید همچنین اثبات کنیم که آیا رشته ها همه قیدهای فضای حل را پوشش می‌دهند یا نه. مرحله بعدی چک کردن این موضوع است که آیا الگوریتم به پایان رسیده است یا نه. پایان هنگامی است که یک حل انتخاب شده ای وجود دارد که با محک هدف برای مقدار شایستگی مطابقت دارد. در نسل اول بعید است که چنین حلی در میان حلهای انتخاب شده وجود داشته باشد. چون این حلها به صورت اتفاقی انتخاب می‌شوند. وقتی که حلهای انتخاب شده نسل جدید خیلی خوب نیستند اجرای الگوریتم می تواند متوقف شود. محک پایان ساده دیگر ، یک تعداد ماکزیمم نسل است. این از پیشرفت نا محدود و نامشخص الگوریتم وقتی که همگرایی نتواند حاصل شود جلوگیری می کند.

مرحله انتخاب:

مرحله بعدی از قانون انتخاب داروین الهام می‌گیرد. بقای نسل در این مرحله همچنین مرحله بهره‌برداری از الگوریتم نامیده می‌شود. یک مجموعه جدید از حلهای انتخاب شده از جمعیت حاضر ایجاد میشود. این حلها یک جمعیت موقت و جدید می‌سازند. این جمعیت بدین صورت ایجاد می شود که یک حل انتخاب شده از جمعیت حاضر با توجه به یک استراتژی انتخاب از قبل تعریف شده انتخاب می‌شود. یک کپی از این رشته گرفته می‌شود و این کپی به عنوان عضو اول این جمعیت جدید قرار داده می‌شود. این عمل N مرتب تکرار می‌شود تا جمعیت جدید کامل شود. وقتی که استراتژی انتخاب، حلهای با مقدار شایستگی بالاتر را ترجیح

دهد، جمعیت جدید شامل تعداد بیشتری حل خوب از جمعیت اولیه می‌شود. این همچنین فشار انتخاب یاSelection pressure نامیده می‌شود. به هرحال هیچ حل جدیدی در این مرحله ایجاد نمی شود جمیت جدید انتظار می رود که به طور میانگین یک کیفیت بالاتری نسبت به جمعیت قبلی داشته باشد یک استراتژی انتخاب معمول این است که شاخص نمونه برداری متناسب با مقدار شایستگی داشته باشد روشهای زیادی برای انتخاب وجود دارد نظیر انتخاب آستانه ، انتخاب خطی، انتخاب چرخ رولت .

چرخه رولت :

در این روش انتخاب، یک مقدار مجموع (sum) در نظر گرفته می شود که این مقدار مجموع می تواند مجموع احتمال انتخاب یا مجموع مقادیر شایستگی تمامی راه حلها در جمعیت باشد. سپس راه حلها تک تک در محدوده [sum,۰] وارد می شوند. اندازه هر راه حل بستگی به مقدار برازندگی آن در جمعیت دارد.

فشار انتخاب که در مرحله قبل توضیح داده شد ضروری است اما کافی نیست تا فضای مسئله را به صورت کارا جستجو کند. همه آنچه تاکنون انجام شد برای این است که در جمعیت های آینده بهترین حلهای جمعیت اصلی یا اولیه اکثریت داشته باشد. در عین حال باید اطلاعاتی که تاکنون در جمعیت وجود داشته حفظ شوند. این می‌تواند به وسیله آنچه که عملگرهای ژنتیک نامیده می‌شود. یعنی جهش و عملگر ترکیب، برآورده شوند.

ترکیب :

این عملگر از جابجایی بیولوژیکی بین کروموزومها الهام گرفته شده است. قسمتهایی از دو حل انتخاب شده (رشته‌های مادر) با هم ترکیب می‌شوند،به طوری که دو حل جدید (رشته‌های نوزاد) بدست می‌آید. ساده‌ترین نوع جابجایی یا ترکیب، نوع تک – نقطه‌ای آن است. که در آن دو رشته در یک نقطه ای که به صورت اتفاقی انتخاب می‌شود به دو قسمت تقسیم می‌شوند. قسمتهای مختلف سپس با هم ترکیب می‌شوند. عقیده براین است که محلهای بهتر ممکن است با این نوع ترکیب به دست آیند. اطلاعاتی که تاکنون وجود داشته معمولاًحفظ می‌شود و به رشته‌های نوزاد منتقل می‌شود.

Chromosome ۱

۰۰۱۰۰۱۱۰۱۱۰ | ۱۱۰۱۱

Chromosome ۲

۱۱۰۰۰۰۱۱۱۱۰ | ۱۱۰۱۱

Offspring ۱

۱۱۰۰۰۰۱۱۱۱۰ | ۱۱۰۱۱

Offspring ۲

۰۰۱۰۰۱۱۰۱۱۰ | ۱۱۰۱۱

یک نقطه ترکیب برای هر جفت از رشته ها به صورت اتفاقی انتخاب می‌شود. در این نقطه، رشته‌های مادر می‌توانند شکسته شوند تا رشته های نوزاد را تشکیل دهند. این ترکیب والدین با احتمال pr انجام می‌شود که معمولاً بین ۰/۱ تا ۰/۵ قرار می‌گیرند.واضح است که عملگرهای ترکیب، حلهای انتخاب شده جدیدی در الگوریتم ژنتیکی ایجاد می‌کنند. خیلی از دست اندکاران وخبرگان الگوریتم ژنتیکی ادعا می‌کنند که عملگر ترکیب چیزی است که الگوریتم ژنتیکی را از سایر تکنیک‌های بهینه کردن متمایز می‌کند. این نکته قابل ذکر است که هرنوع ترکیب که استفاده می‌شود همیشه این خطر وجود دارد که حلهای بی‌ارزش تولید شود. این مشکل در مرحله ارزیابی با چک کردن اینک آیا حل انتخاب شده همه قیدها را در برمی‌گیرد یا نه حل می‌شود. به عنوان یک قانون می‌توان بیان کرد که وقتی تعداد زیادی حل غیرمجاز پیدا شوند بدین معنی است که طریق بیان کردن حل و عملگر ترکیب، خوب با هم ترکیب نشده اند.

جهش :

در یک وضعیت به خصوص، عملگر ترکیب قادر نیست که تغییر لازم را ایجاد کند. وقتی که یک بایت در همه رشته ها یکسان باشد، این بایت نمی‌تواند به وسیله عملگر ترکیب، تغییر کند. برای مثال جمعیت رشته های زیر را بررسی می‌کنیم.

۰۱۱۰۰۱۰۱ ۱۱۱۰۱۰۰۱

۱۰۰۱۱۱۰۱ ۰۰۱۱۱۱۱۱

بایت آخر در همه رشته ها برابر ۱ است. این بدان معنا است که فقط اعداد فرد در جمعیت حضور دارند و ممکن نیست که با عملگر ترکیب به تنهایی یک عدد زوج بدست آید. با این مکانیسم الگوریتم ژنتیکی ممکن است در بهینه محلی در فضای جستجو به دام افتد. برای فائق آمدن براین مشکل، یک عملگر معرفی می‌شود، عملگر جهش. جهش یک تغییر محلی اتفاقی در رشته است. هر موقعیتی در هر رشته در معرض جهش با احتمال pm که معمولاً پایین تر از ۰/۰۵ است قرار می‌گیرد. با بیان دودویی، جهش باعث می‌شود که آن بایت هایی که انتخاب می‌شوند از صفر به یک و بالعکس تغییر کنند. جهش، در جمعیت تنوع ایجاد می‌کند و به پرهیز از همگرایی زودرس کمک می‌کند. تغییر یک بایت عملاً رشته را به یک مکان کاملاً متفاوت در فضای حل حرکت می‌دهد، جهش می‌تواند برای این منظور استفاده شود که رشته هایی که تاکنون خوب عمل کرده اند را روی یک قله فضای حل قرار دهد و بتوان بررسی ها را روی این قله انجام داد. این می‌تواند با تغییر کم مقادیر پارامتری حاضر انجام شود.

Original offspring ۱

۱۱۰۱۱۱۱۰۰۰۱۱۱۱۰

Original offspring ۲

۱۱۰۱۱۰۰۱۰۰۱۱۰۱۱۰

Mutated offspring ۱

۱۱۰۰۱۱۱۰۰۰۱۱۱۱۰

Mutated offspring ۲

۱۱۰۱۱۰۱۱۰۰۱۱۰۱۱۰

ابتدا به جهش در الگوریتم ژنتیکی یک نقش زمینه ای نسبت داده می‌شد چون سرعت جهش بالا، به اطلاعات مفیدی که تاکنون در جمعیت وجود داشته، آسیب می‌رساند. بنابراین یک الگوریتم ژنتیکی که از یک احتمال جهش پایین استفاده می‌کند نسبت به الگوریتم که از احتمال بالا استفاده می‌کند کارآیی بهتری دارد. یک احتمال جهش بالا بدون ترکیب، مشابه با یک جستجوی اتفاقی است. سرعت جهش برابر صفر ممکن است که باعث شود الگوریتم ژنتیکی در یک بهینه محلی، به دام افتد. اخیراً نقش جهش مورد بازیابی مجدد قرار گرفته است. در چند نوع از مسائل تأثیر عملگر جهش روی کارآیی الگوریتم ژنتیک به نظر می‌رسد که بیشتر از تأثیر عملگر ترکیب باشد. یک سرعت جهش بالا و یک سرعت ترکیب پایین در چنین مواردی ترجیح داده می‌شود.

جایگزینی و ادامه:

بعد از اعمال ترکیب و جهش، جمعیت حاضر می‌تواند با جمعیت نسل جدید و موقتی جایگزین شود. وقتی جمعیت موقت و اصلی از نظر اندازه و بزرگی یکسان هستند فرآیند جایگزینی آسان است. هنگامی که جمعیت موقت کوچکتر از جمعیت اصلی انتخاب شود. فرآیند جایگزینی نیاز به مقداری توجه دارد.

اندازه کارآیی یک نسل:

کارآیی یک نسل می‌توانند با روشهای مختلف اندازه گیری شود. به هرحال هیچ راه خوب و منحصر به فردی وجود ندارد که کارآیی یک الگوریتم ژنتیکی بخصوص را اندازه‌گیری کند. می‌توانیم میانگین مقادیر شایستگی همه اعضای جمعیت، میانگین k عضو یک نسل یا بهترین حل هرنسل یا ترکیبی از این ها را محاسبه کنیم. معمولاً شایستگی میانگین جمعیت همراه با بهترین رشته به صورت تابعی از تعداد نسل رسم می‌شود.

پیکربندی الگوریتم‌های ژنتیکی :

در قسمت قبل خیلی از جنبه های پیکربندی الگوریتم های ژنتیک مورد بحث قرار گرفت.. کاربران با تجربه امکان زیاد دارند تا الگوریتم ژنتیکی را با احتیاجات خود منطبق کنند. بهرحال تاکنون هیچ روش

استانداردی برای یک مسئله به خصوص قابل دسترسی نبوده است. این بدان معنی است که تجربه و تعدادی از قوانین سرانگشتی کاربر را راهنمایی خواهد کرد. باید تأکید شود که فهم مناسب و معقول از مکانیسم های جستجوی تأثیر پارامترهای مختلف یک ضرورت است. بهینه کردن با یک حدس اولیه برای تنظیم پارامتری پیکربندی الگوریتم ژنتیکی از جایی که اصلاح بیشتر می‌تواند انجام شود شروع می‌شود. استراتژی های مختلفی می‌تواند استفاده شود تا پیکربندی بهینه گردد. الگوریتم ژنتیکی معمولاً بهینه کلی در فضای جستجو را پیدا می‌کند. اما تنظیم غیربهینه الگوریتم، باعث صرف زمان زیادی برای رسیدن به همگرایی می‌شود. یک مورد خیلی مشکل هنگامی اتفاق می‌افتد که الگوریتم ژنتیکی همگرا شونده نباشد. در نتیجه مشکل است که پی ببریم که کدام تنظیم می‌بایست تغییر داده شود تا به همگرایی برسیم.

مهمترین پارامترهای پیکربندی الگوریتم ژنتیکی :

انتخاب روش بیان حل

انتخاب استراتژی برای شروع الگوریتم

انتخاب رشته ها و پارامترهای مناسب

ترکیب رشته ها و جهش

جایگزینی رشته ها

Np : اندازه جمعیت اولیه

'N : اندازه جمعیت موقتی

Pr : احتمال ترکیب

Pm : احتمال جهش

صحت و دقت جستجو

فهم این نکته مهم است که کارآیی یک الگوریتم ژنتیکی براساس یک جمعیت از حلها است تا یک حل بخصوص. در این جمعیت یک صحت جستجو می‌تواند تعریف شود. منظور از صحت این است که بهینه صحیح در فضای حل، با یک تعداد کافی از رشته ها در جمعیت بدست می آید.

به طور کلی مشاهده می‌شود که الگوریتم‌های ژنتیکی صحت جستجوی خیلی خوبی را از این نظر که اغلب به قله صحیح در شایستگی می‌رسند نشان می‌دهند. با توجه به طبیعت احتمالی عملگرها در الگوریتم ژنتیکی مشکل است که به این قله برسیم. به این دلیل است که گفته می‌شود الگوریتم ژنتیکی دقت جستجوی خوبی ندارند.

الگوریتم ژنتیکی در انتخاب متغیر:

در مسائل انتخاب متغیر، ابتدا به صورت اتفاقی یک جمعیت از رشته‌ها تولید می‌شود در اینجا هررشته یک بردار سطری است که عناصر آن را متغیرها تشکیل می‌دهند و هر عنصر به صورت صفر یا یک در رشته کددار شده است. صفر برای مواقعی که متغیر وجود ندارد و یک برای موقعی که متغیر وجود دارد. شایستگی می‌تواند برابر با RMSE برای سری کالیبراسیون یا پیش‌بینی باشد که از رابطه زیر بدست می‌آید. که باید مینیمم باشد. در این رابطه yi مقدار واقعی پاسخ مورد نظر و oi مقدار پیش بینی شده آن با استفاده از مدل یا متغیرهای انتخاب شده است. برای هررشته از این جمعیت که به طور اتفاقی تولید شده مقدار شایستگی محاسبه می‌شود و بهترین رشته‌ها انتخاب می‌شود و تحت عمل ترکیب و جهش قرار می‌گیرند. این فرآیند چند مرتبه تکرار می‌شود تا در نهایت به حل بهینه برسیم .

کاربردهای الگوریتم ژنتیک :

زمینه الگوریتم‌های ژنتیکی نسبتاً نو و تازه است. اولین کاربردهای آن در حدود سال ۱۹۶۰ وقتی که هولاند این روش را پیشنهاد کرد گزارش شد. به هرحال فقط از سال ۱۹۸۰ که تعداد مقالات به صورت نمایی افزایش یافت که بیشتر به دلیل توسعه تکنولوژی کامپیوتر بود رشد نمود. در سال ۱۹۸۵ اولین کنفرانس

الگوریتم ژنتیکی برگزار شد و اولین کتاب در این زمینه در سال ۱۹۸۹ منتشر شد. الگوریتم ژنتیک کاربردهای بسیار وسیعی در علوم دارا می‌باشد که مهمترین آنها عبارتند از:

فیت کردن طیف‌های IR

رفتار کروماتوگرافی غیرخطی

پارامترهای سینتیکی

پارامترهای رگرسیون

بهینه کردن پارامترهای کنترل کیفیت

ارتباطات کمی ساختار – فعالیت

آموزش شبکه عصبی

مشخص کردن ساختارهای شیمیایی

تعیین طیف NMR پروتئین‌ها

طراحی توالی ستون‌های متغیر

انتخاب طول موج برای کالیبراسیون چند متغیره

دسته‌بندی مجموعه اطلاعات بزرگ

نتیجه‌گیری کلی (الگوریتم ژنتیک):

برای حل مسئله به روش ژنتیک ابتدا باید پاسخ مسئله را کدگذاری کرده، به گونه‌ای که در ادامه اجرای الگوریتم بتوان این پاسخ را مورد ارزیابی قرار داد و عملگرهای مختلف را بر آن اعمال کرده اجرای الگوریتم با ایجاد یک مجموعه ابتدایی از جوابهای تصادفی که جمعیت اولیه نامیده می‌شود شروع می‌گردد. هر عضو در جمعیت یک کروموزوم نامیده می‌شود که نمایانگر یک حل برای مسئله موجود است.

یک کروموزوم رشته‌ای از اعداد است که در اصطلاح ژن نامیده می‌شوند و معمولاً نه لزوماً یک رشته دودویی است. طی هر تکرار الگوریتم ژنتیک، مجموعه جدیدی از کروموزومها تولید می‌شوند. جمعیت در زمان

معلوم را نسل می‌نامند.و طی هر نسل میزان برازش کروموزومها یا تابع برازش یک کروموزوم را با توجه به تابع هدف مسئله برآورد می‌کند، تعیین می‌شود. طی فرآیند باز تولید، عملگرهای ژنتیک یعنی عملگر ترکیب و عملگر جهش برروی کروموزومها اعمال می‌شود.

به کروموزومهایی که از این طریق تولید می‌شوند نوزاد اطلاق می‌شود. سپس برازندگی نوزادان ارزیابی شده و به وسیله یکی از روشهای انتخاب، کروموزومهای بهتر انتخاب و به نسل بعد منتقل می‌شوند. برای هریک از عملگرهای ژنتیک پارامتر احتمال تعریف می‌شود که عملگرها با این احتمالات بر کروموزومها اعمال می‌شوند. هر تکرار این روند یک نسل را ایجاد می‌کند که تعداد نسلها به دلخواه تعیین می‌شوند. در این فرآیند، الگوریتم به بهترین کروموزوم همگرا می‌شود که نمایانگر جواب بهینه یا زیر بهینه مسأله است.

الگوریتم مورچگان:

سیستم مورچه CE یک سیستم هوشمند پر ازدحام (مانند کندوی زنبوران) است و رفتار کاوش آذوقه مورچگان را تقلید می کند که شامل تعداد زیادی عامل با رفتارهای ساده است که بطور غیر مستقیم و غیر همزمان با هم ارتباط برقرار می کنند . همه عاملها مأموریت جستجوی مسیرهای حلقه ای و گزارش کیفیت مسیر که بوسیله مفهوم " عملکرد مسیر " تعریف می شود , را دارند.

کاربردهای الگوریتم مورچگان :

- ایجاد و نگهداری اتصال مسیرهای اولیه و پشتیبانی از هم گسیخته .
- حل مسایل مربوط به پیدا کردن و نگهداری مسیرهای مجازی در یک **شبکه** ارتباطات با تغییر شرایط .
- ساختار اتوماتیک سایتهای پرتال در وب .
- استفاده در مسیریاب های نرم افزاری و عاملهای موبایل (مثال زیر)

مثال :

مورچگان بین منبع غذا و لانه یک مسیر را حفظ می کنند .

نکات کلی:

- لانه مورچه در کامپیوتر میزبان است .
- مورچگان عاملهای موبایل هستند .
- هدف : جلوگیری از بکارگیری اتصالات متراکم .
- پاکتهای داده در هر مسیریاب , مسیریابی می شوند .

مسیریاب:

- پاکتهای داده با کلیک مسیریابی می شوند .
- کد مورچه در Java VM اجرا می شود .
- دو زیر سیستم اطلاعاتشان مورد نیازشان را با هم رد و بدل می کنند .

میزبانان :

- مورچگان و تست ترافیک تولید می شوند .
- آمار نرخ پاکتها برای هر اتصال .

تست انطباق:

- تمام اتصالات بین مسیریابها دارای ظرفیت ۵۰ پاکت در ثانیه اند .
- ابتدا , دو اتصال بدون ترافیک ایجاد می شود .
- پس از مدتی , اتصال ۱ شروع به فرستادن ترافیک CBR تقویت شده با نرخ ۴۰ پاکت در ثانیه می کند .

نتیجه اتصال ۲:

- در ابتدا , کوتاهترین مسیر ترجیح داده می شود .
- ترافیک ناشی از اتصال ۱ منجر به افزایش هزینه می شود .
- اکنون مسیر دیگری کمترین هزینه را خواهد داشت .
- سیستم به مسیر منتخب سویچ می کند .

ارزیابی:

- این الگوریتم برای پیاده سازی مسیریابهای نرم افزاری و عاملهای موبایل موجه است .
- عملکرد محدود است .
- پیاده سازی بر پایه سیستم عامل موبایل است .
- برای اجرای سیستم در شبکه بزرگتر , عملکرد باید بهبود یابد .

مقدمات رمز نگاری

رمزگذاری یعنی تبدیل اطلاعات به یک شکل غیر قابل فهم و انتقال آن و سپس برگرداندن اطلاعات رمز شده به حالت اولیه و قابل خواندن. عناصر مهمی که در رمزگذاری مورد استفاده قرار می‌گیرند به شرح زیر می‌باشد:

۱- معرفی و اصطلاحات

رمزنگاری علم کدها و رمزهاست. یک هنر قدیمی است و برای قرنها بمنظور محافظت از پیغامهایی که بین فرماندهان، جاسوسان، عشاق و دیگران ردوبدل می‌شده، استفاده شده است تا پیغامهای آنها محرمانه بماند.

هنگامی که با امنیت دیتا سروکار داریم، نیاز به اثبات هویت فرستنده و گیرنده پیغام داریم و در ضمن باید از عدم تغییر محتوای پیغام مطمئن شویم. این سه موضوع یعنی محرمانگی، تصدیق هویت و جامعیت در قلب امنیت ارتباطات دیتای مدرن قرار دارند و می‌توانند از رمزنگاری استفاده کنند.

اغلب این مساله باید تضمین شود که یک پیغام فقط میتواند توسط کسانی خوانده شود که پیغام برای آنها ارسال شده است و دیگران این اجازه را ندارند. روشی که تامین کننده این مساله باشد "رمزنگاری" نام دارد. رمزنگاری هنر نوشتن بصورت رمز است بطوریکه هیچکس بغیر از دریافت کننده موردنظر نتواند محتوای پیغام را بخواند.

رمزنگاری مخففها و اصطلاحات مخصوص به خود را دارد. برای درک عمیق‌تر به مقداری از دانش ریاضیات نیاز است. برای محافظت از دیتای اصلی (که بعنوان plaintext شناخته می‌شود)، آنرا با استفاده از یک کلید (رشته‌ای محدود از بیتها) بصورت رمز در می‌آوریم تا کسی که دیتای حاصله را می‌خواند قادر به درک آن نباشد. دیتای رمزشده (که بعنوان ciphertext شناخته می‌شود) بصورت یک سری بی‌معنی از بیتها بدون داشتن رابطه مشخصی با دیتای اصلی بنظر می‌رسد. برای حصول متن اولیه دریافت‌کننده آنرا رمزگشایی می‌کند. یک شخص ثالث (مثلا یک هکر) می‌تواند برای اینکه بدون دانستن کلید به دیتای اصلی دست یابد، کشف رمزنوشته (cryptanalysis) کند. بخاطرداشتن وجود این شخص ثالث بسیار مهم است.

رمزنگاری دو جزء اصلی دارد، یک الگوریتم و یک کلید. الگوریتم یک مبدل یا فرمول ریاضی است. تعداد کمی الگوریتم قدرتمند وجود دارد که بیشتر آنها بعنوان استانداردها یا مقالات ریاضی منتشر شده‌اند. کلید، یک رشته از ارقام دودویی (صفر و یک) است که بخودی‌خود بی‌معنی است. رمزنگاری مدرن فرض می‌کند که الگوریتم شناخته شده است یا می‌تواند کشف شود. کلید است که باید مخفی نگاه داشته شود و کلید است که در هر مرحله پیاده‌سازی تغییر می‌کند. رمزگشایی ممکن است از همان جفت الگوریتم و کلید یا جفت متفاوتی استفاده کند.

دیتای اولیه اغلب قبل از رمزشدن بازچینی می‌شود؛ این عمل عموما بعنوان scrambling شناخته می‌شود. بصورت مشخص‌تر، hash functionها بلوکی از دیتا را (که می‌تواند هر اندازه‌ای داشته باشد) به طول از پیش مشخص‌شده کاهش می‌دهد. البته دیتای اولیه نمی‌تواند از hashed value بازسازی شود. Hash functionها اغلب بعنوان بخشی از یک سیستم تایید هویت مورد نیاز هستند؛ خلاصه‌ای از پیام (شامل مهم‌ترین قسمتها مانند شماره پیام، تاریخ و ساعت، و نواحی مهم دیتا) قبل از رمزنگاری خود پیام، ساخته و hash می‌شود.

۱-۱ یک چک تایید پیام (Check Message Authentication) یا MAC یک الگوریتم ثابت با تولید یک امضاء برروی پیام با استفاده از یک کلید متقارن است. هدف آن نشان دادن این مطلب است که پیام بین ارسال و دریافت تغییر نکرده است. هنگامی که رمزنگاری توسط کلید عمومی برای تایید هویت فرستنده پیام استفاده می‌شود، منجر به ایجاد امضای دیجیتال (digital signature) می‌شود.

۱-۲ Public Key یا کلید عمومی اعداد یا کلماتی که با یک شخص یا سازمان در ارتباط می‌باشد. کلید عمومی جزئی از جفت کلید عمومی/خصوصی می‌باشد وبه صورت عمومی در دسترس کسانی که قصد انتقال اطلاعات رمز شده را دارند، می‌باشد.

۱-۳ Private Key یا کلید خصوصی اعداد یا کلماتی که با یک شخص یا سازمان در ارتباط می‌باشد. کلید خصوصی جزئی از جفت کلید عمومی/خصوصی می‌باشد. کلید خصوصی فقط در دسترس مالک جفت کلید عمومی/خصوصی می‌باشد و برای بازگشایی اطلاعاتی که توسط کلید عمومی رمزگذاری شده استفاده می‌شود.

۱-۴ ایجادکننده های جفت کلید برای ایجاد یک جفت کلید عمومی و خصوصی طبق یک الگوریتم رمزگذاری مشخص استفاده می‌شود.

۱-۵ Key Factories برای تبدیل کلید های نامشخص به کلیدهای مشخص به کار می‌رود.

۱-۶ Keystores بانکی که برای مدیریت تعدادی از کلید ها به کار می‌رود.

۱-۷ الگوریتم های رمزگذاری الگوریتم ها و روشهایی که برای رمزگذاری اطلاعات به کار می‌رود. RSA و DES نام دو تا از معروفترین الگوریتم ها می‌باشد.

۲- الگوریتم‌ها

طراحی الگوریتمهای رمزنگاری مقوله‌ای برای متخصصان ریاضی است. طراحان سیستمهایی که در آنها از رمزنگاری استفاده می‌شود، باید از نقاط قوت و ضعف الگوریتمهای موجود مطلع باشند و برای تعیین الگوریتم مناسب قدرت تصمیم‌گیری داشته باشند. اگرچه رمزنگاری از اولین کارهای شانون (Shannon) در اواخر دهه ۴۰ و اوایل دهه ۵۰ بشدت پیشرفت کرده است، اما کشف رمز نیز پابه‌پای رمزنگاری به پیش آمده است و الگوریتمهای کمی هنوز با گذشت زمان ارزش خود را حفظ کرده‌اند. بنابراین تعداد الگوریتمهای استفاده شده در سیستمهای کامپیوتری عملی و در سیستمهای برپایه کارت هوشمند بسیار کم است.

۱-۲ سیستمهای کلید متقارن

یک الگوریتم متقارن از یک کلید برای رمزنگاری و رمزگشایی استفاده می‌کند. بیشترین شکل استفاده از رمزنگاری که در کارتهای هوشمند و البته در بیشتر سیستمهای امنیت اطلاعات وجود دارد data encryption algorithm یا DEA است که بیشتر بعنوان DES شناخته می‌شود. DES یک محصول دولت ایالات متحده است که امروزه بطور وسیعی بعنوان یک استاندارد بین‌المللی شناخته می‌شود. بلوکهای ۶۴بیتی دیتا توسط یک کلید تنها که معمولا ۵۶بیت طول دارد، رمزنگاری و رمزگشایی می‌شوند. DES از نظر محاسباتی ساده است و براحتی می‌تواند توسط پردازنده‌های کند (بخصوص آنهایی که در کارتهای هوشمند وجود دارند) انجام گیرد.

این روش بستگی به مخفی‌بودن کلید دارد. بنابراین برای استفاده در دو موقعیت مناسب است: هنگامی که کلیدها می‌توانند به یک روش قابل اعتماد و امن توزیع و ذخیره شوند یا جایی که کلید بین دو سیستم مبادله می‌شوند که قبلا هویت یکدیگر را تایید کرده‌اند عمر کلیدها بیشتر از مدت تراکنش طول نمی‌کشد. رمزنگاری DES عموما برای حفاظت دیتا از شنود در طول انتقال استفاده می‌شود.

کلیدهای ۴۰ DESبیتی امروزه در عرض چندین ساعت توسط کامپیوترهای معمولی شکسته می‌شوند و بنابراین نباید برای محافظت از اطلاعات مهم و با مدت طولانی اعتبار استفاده شود. کلید ۵۶بیتی عموما توسط سخت‌افزار یا شبکه‌های بخصوصی شکسته می‌شوند. رمزنگاری DESسه‌تایی عبارتست از کدکردن دیتای اصلی با استفاده از الگوریتم DES که در سه مرتبه انجام می‌گیرد. (دو مرتبه با استفاده از یک کلید به سمت جلو (رمزنگاری) و یک مرتبه به سمت عقب (رمزگشایی) با یک کلید دیگر) مطابق شکل زیر:

این عمل تاثیر دوبرابر کردن طول مؤثر کلید را دارد؛ بعدا خواهیم دید که این یک عامل مهم در قدرت رمزکنندگی است.

الگوریتمهای استاندارد جدیدتر مختلفی پیشنهاد شده‌اند. الگوریتمهایی مانند Blowfish و IDEA برای زمانی مورد استفاده قرار گرفته‌اند اما هیچکدام پیاده‌سازی سخت‌افزاری نشدند بنابراین بعنوان رقیبی برای DES برای استفاده در کاربردهای میکروکنترلی مطرح نبوده‌اند. پروژه استاندارد رمزنگاری پیشرفته دولتی

ایالات متحده (AES) الگوریتم Rijndael را برای جایگزینی DES بعنوان الگوریتم رمزنگاری اولیه انتخاب کرده است. الگوریتم Twofish مشخصا برای پیاده‌سازی در پردازنده‌های توان-پایین مثلا در کارتهای هوشمند طراحی شد.

در ۱۹۹۸ وزارت دفاع ایالات متحده تصمیم گرفت که الگوریتمها Skipjack و مبادله کلید را که در کارتهای Fortezza استفاده شده بود، از محرمانگی خارج سازد. یکی از دلایل این امر تشویق برای پیاده‌سازی بیشتر کارتهای هوشمند برپایه این الگوریتمها بود.

برای رمزنگاری جریانی (streaming encryption) (که رمزنگاری دیتا در حین ارسال صورت می‌گیرد بجای اینکه دیتای کدشده در یک فایل مجزا قرار گیرد) الگوریتم RC۴ سرعت بالا و دامنه‌ای از طول کلیدها از ۴۰ تا ۲۵۶ بیت فراهم می‌کند. RC۴ که متعلق به امنیت دیتای RSA است، بصورت عادی برای رمزنگاری ارتباطات دوطرفه امن در اینترنت استفاده می‌شود.

۲-۲ سیستمهای کلید نامتقارن

سیستمهای کلید نامتقارن از کلید مختلفی برای رمزنگاری و رمزگشایی استفاده می‌کنند. بسیاری از سیستمها اجازه می‌دهند که یک جزء (کلید عمومی یا key public) منتشر شود در حالیکه دیگری (کلید اختصاصی یا private key) توسط صاحبش حفظ شود. فرستنده پیام، متن را با کلید عمومی گیرنده کد می‌کند و گیرنده آن را با کلید اختصاصی خودش رمزنگاری میکند. بعبارتی تنها با کلید اختصاصی گیرنده می‌توان متن کد شده را به متن اولیه صحیح تبدیل کرد. یعنی حتی فرستنده نیز اگرچه از محتوای اصلی پیام مطلع است اما نمی‌تواند از متن کدشده به متن اصلی دست یابد، بنابراین پیام کدشده برای هرگیرنده‌ای بجز گیرنده مورد نظر فرستنده بی‌معنی خواهد بود. معمولترین سیستم نامتقارن بعنوان RSA شناخته می‌شود (حروف اول پدیدآورندگان آن یعنی ، Shamir Rivest و Adlemen است). اگرچه چندین طرح دیگر وجود دارند. می‌توان از یک سیستم نامتقارن برای نشاندادن اینکه فرستنده پیام همان شخصی است که ادعا

می‌کند استفاده کرد که این عمل اصطلاحا امضاء نام دارد. RSA شامل دو تبدیل است که هرکدام احتیاج به بتوان‌رسانی ماجولار با توانهای خیلی طولانی دارد:

امضاء، متن اصلی را با استفاده از کلید اختصاصی رمز می‌کند؛ رمزگشایی عملیات مشابه‌ای روی متن رمزشده اما با استفاده از کلید عمومی است. برای تایید امضاء بررسی می‌کنیم که آیا این نتیجه با دیتای اولیه یکسان است؛ اگر اینگونه است، امضاء توسط کلید اختصاصی متناظر رمزشده است.

به بیان ساده‌تر چنانچه متنی از شخصی برای دیگران منتشر شود، این متن شامل متن اصلی و همان متن اما رمز شده توسط کلید اختصاصی همان شخص است. حال اگر متن رمزشده توسط کلید عمومی آن شخص که شما از آن مطلعید رمزگشایی شود، مطابقت متن حاصل و متن اصلی نشاندهنده صحت فرد فرستنده آن است، به این ترتیب امضای فرد تصدیق می‌شود. افرادی که از کلید اختصاصی این فرد اطلاع ندارند قادر به ایجاد متن رمزشده نیستند بطوریکه با رمزگشایی توسط کلید عمومی این فرد به متن اولیه تبدیل شود.

اساس سیستم RSA این فرمول است: $X = Yk \pmod r$

که X متن کد شده، Y متن اصلی، k کلید اختصاصی و r حاصلضرب دو عدد اولیه بزرگ است که با دقت انتخاب شده‌اند. برای اطلاع از جزئیات بیشتر می‌توان به مراجعی که در این زمینه وجود دارد رجوع کرد. این شکل محاسبات روی پردازنده‌های بایتی بخصوص روی ۸ بیتی‌ها که در کارتهای هوشمند استفاده می‌شود بسیار کند است. بنابراین، اگرچه RSA هم تصدیق هویت و هم رمزنگاری را ممکن می‌سازد، در اصل برای تایید هویت منبع پیام از این الگوریتم در کارتهای هوشمند استفاده می‌شود و برای نشاندادن عدم تغییر پیام در طول ارسال و رمزنگاری کلیدهای آتی استفاده می‌شود.

سایر سیستمهای کلید نامتقارن شامل سیستمهای لگاریتم گسسته می‌شوند مانند Diffie-Hellman، ElGamal و سایر طرحهای چندجمله‌ای و منحنی‌های بیضوی. بسیاری از این طرحها عملکردهای یک‌ـطرفه‌ای دارند که اجازه تاییدهویت را می‌دهند اما رمزنگاری ندارند. یک رقیب جدیدتر الگوریتم RPK است که از یک تولیدکننده مرکب برای تنظیم ترکیبی از کلیدها با مشخصات مورد نیاز استفاده می‌کند. RPK

یک پروسه دو مرحله‌ای است: بعد از فاز آماده‌سازی در رمزنگاری و رمزگشایی (برای یک طرح کلید عمومی) رشته‌هایی از دیتا بطور استثنایی کاراست و می‌تواند براحتی در سخت‌افزارهای رایج پیاده‌سازی شود. بنابراین بخوبی با رمزنگاری و تصدیق‌هویت در ارتباطات سازگار است.

طولهای کلیدها برای این طرحهای جایگزین بسیار کوتاهتر از کلیدهای مورد استفاده در RSA است که آنها برای استفاده در چیپ‌کارتها مناسب‌تر است. اما RSA محکی برای ارزیابی سایر الگوریتمها باقی مانده است؛ حضور و بقای نزدیک به سه‌دهه از این الگوریتم، تضمینی در برابر ضعفهای عمده بشمار می‌رود.

۳- روشهای رمزگذاری

۳-۱ روش متقارن Symmetric در این روش هر دو طرفی که قصد رد و بدل اطلاعات را دارند از یک کلید مشترک برای رمزگذاری و نیز بازگشایی رمز استفاده می‌کنند.در این حالت بازگشایی و رمزگذاری اطلاعات دو فرآیند معکوس یکدیگر می‌باشند. مشکل اصلی این روش این است که کلید مربوط به رمزگذاری باید بین دو طرف به اشتراک گذاشته شود و این سوال پیش می‌آید که دو طرف چگونه می‌توانند این کلید را به طور امن بین یکدیگر رد و بدل کنند. انتقال از طریق انترانت و یا به صورت فیزیکی تا حدی امن می‌باشد اما در انتقال آن در اینترنت به هیچ وجه درست نمی‌باشد.در این قبیل سیستم ها، کلید های رمزنگاری و رمزگشایی یکسان هستند و یا رابطه ای بسیار ساده با هم دارند .این سیستم ها را سیستم های متقارن یا " تک کلیدی " مینامیم. به دلیل ویژگی ذاتی تقارن کلید رمزنگاری و رمزگشایی، مراقبت و جلوگیری از افشای این سیستم ها یا تلاش در جهت امن ساخت آنها لازم است در بر گیرنده " جلوگیری از استراق سمع " و " ممانعت از دستکاری اطلاعات " باشد .

۳-۲ روش نامتقارن Asymmetric این روش برای حل مشکل انتقال کلید در روش متقارن ایجاد شد. در این روش به جای یک کلید مشترک از یک جفت کلید به نام های کلید عمومی و خصوصی استفاده می‌شود. در این روش از کلید عمومی برای رمزگذاری اطلاعات استفاده می‌شود. طرفی که قصد انتقال اطلاعات را به صورت رمزگذاری شده دارد اطلاعات را رمزگذاری کرده و برای طرفی که مالک این جفت کلید است استفاده

می‌شود. مالک کلید، کلید خصوصی را پیش خود به صورت محرمانه حفظ می‌کند. در این دسته، کلید های رمزنگاری و رمزگشایی متمایزند و یا اینکه چنان رابطه پیچیده ای بین آنها حکم فرماست که کشف کلید رمزگشایی با در اختیار داشتن کلید رمزنگاری، عملا ناممکن است.

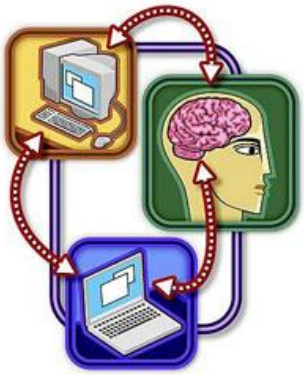
۳-۳ مقایسه رمزنگاری الگوریتم های متقارن و الگوریتم های کلید عمومی: بحث های زیادی شده که کدام یک از این الگوریتم ها بهترند اما جواب مشخصی ندارد. البته بررسی هایی روی این سوال شده به طور مثال Needham و Schroeder بعد از تحقیق به این نتیجه رسیدند که طول پیغامی که با الگوریتم های متقارن میتواند رمزنگاری شود از الگوریتم های کلید عمومی کمتر است. و با تحقیق به این نتیجه ریسیدند که الگوریتم های متقارن الگوریتم های بهینه تری هستند. اما وقتی که بحث امنیت پیش می آید الگوریتم های کلید عمومی کارایی بیشتری‌دارند. و بطور خلاصه می‌توان گفت که الگوریتم های متقارن دارای سرعت بالاتر و الگوریتم های کلید عمومی دارای امنیت بهتری هستند. در ضمن گاهی از سیستم ترکیبی از هردو الگوریتم استفاده میکنند که به این الگوریتم ها الگوریتم های ترکیبی (hybrid)گفته میشود. اما اگر به طور دقیق تر به این دو نگاه کنیم آنگاه متوجه خواهیم شد که الگوریتم های کلید عمومی و الگوریتم های کلید متقارن دارای دو ماهیت کاملا متفاوت هستند و کار برد های متفاوتی دارند به طور مثال در رمزنگاری های ساده که حجم داده ها بسیار زیاد است از الگوریتم متقارن استفاده میشود زیرا داده ها با سرعت بالاتری رمزنگاری و رمزگشایی شوند. اما در پروتکل هایی که در اینترنت استفاده میشود، برای رمز نگری کلید هایی که نیاز به مدیریت دارند از الگوریتم های کلید عمومی استفاده میشود.

۴-۳ Key Agreement همانطور که در بالا گفته شد به علت کند بودن و محدودیت رمزگذاری با روش نامتقارن از این روش فقط برای رمزگذاری کلید مشترک استفاده می‌شود. اما این روش نیز یک مشکل دارد و آن اینست که هر شخص نیاز به کلید عمومی و خصوصی مربوط به خود را دارد و باید برای انتقال اطلاعات آنرا برای طرف مقابل بفرستد. یک راه برای حل مشکل استفاده از کلید عمومی و یک مکانیزم به نام Agreement Key می‌باشد که به طبق آن یک توافق بر روی کلید مخفی بین طرفین به وجود می‌آید و به

این ترتیب نیازی به انتقال کلید نمی‌باشد. وقتی که یک بار بر روی یک کلید مشترک توافق حاصل شد از آن می‌توان برای رمزگذاری و رمزگشایی اطلاعات مربوطه استفاده کرد. معمولا در این روش از الگوریتم Diffie-Hellman استفاده می‌شود. مراحل انتقال اطلاعات از این روش به صورت زیر می‌باشد: - آغازگر ابتدا یک جفت کلید عمومی و خصوصی ایجاد کرده و کلید عمومی را همراه با مشخصات الگوریتم (Algorithm Specification) به سمت طرف مقابل می‌فرستد. - طرف مقابل نیز یک جفت کلید عمومی و خصوصی همراه با مشخصات الگوریتم آغازگر ساخته و کلید عمومی را برای آغازگر می‌فرستد. - آغازگر یک کلید مخفی بر اساس کلید خصوصی خود و کلید عمومی طرف مقابل ایجاد میکند. - طرف مقابل نیز با استفاده از کلید خصوصی خود و کلید عمومی آغازگر یک کلید مخفی می‌سازد. الگوریتم Diffie-Hellman تضمین می‌کند که کلید مخفی هر دو طرف یکسان می‌باشد.

شبکه‌های عصبی (Neural Networks)

آشنایی با شبکه‌های عصبی (Neural Networks) - قسمت اول



شبکه‌های عصبی را می‌توان با اغماض زیاد، مدل‌های الکترونیکی از ساختار عصبی مغز انسان نامید. مکانیسم فراگیری و آموزش مغز اساساً بر تجربه استوار است. مدل‌های الکترونیکی شبکه‌های عصبی طبیعی نیز بر اساس همین الگو بنا شده‌اند و روش برخورد چنین مدل‌هایی با مسائل، با روش‌های محاسباتی که به‌طور معمول توسط سیستم‌های کامپیوتری در پیش گرفته شده‌اند، تفاوت دارد. می‌دانیم که حتی ساده‌ترین مغزهای جانوری هم قادر به حل مسائلی هستند که اگر نگوییم که کامپیوترهای امروزی از حل آنها عاجز هستند، حداقل در حل آنها دچار مشکل می‌شوند. به عنوان مثال، مسائل مختلف شناسایی الگو، نمونه‌ای از مواردی هستند که روش‌های معمول محاسباتی برای حل آنها به نتیجه مطلوب نمی‌رسند. درحالی‌که مغز ساده‌ترین جانوران به‌راحتی از عهده چنین مسائلی بر می‌آید. تصور عموم کارشناسان IT بر آن است که مدل‌های جدید محاسباتی که بر اساس شبکه‌های عصبی بنا می‌شوند، جهش بعدی صنعت IT را شکل می‌دهند. تحقیقات در این زمینه نشان داده است که مغز، اطلاعات را همانند الگوها (pattern) ذخیره می‌کند. فرآیند ذخیره‌سازی اطلاعات به‌صورت الگو و تجزیه و تحلیل آن الگو، اساس روش نوین محاسباتی را تشکیل می‌دهند. این حوزه از دانش محاسباتی (computation) به هیچ وجه از روش‌های برنامه‌نویسی سنتی استفاده نمی‌کند و به‌جای آن از شبکه‌های بزرگی که به‌صورت موازی آرایش شده‌اند و تعلیم یافته‌اند، بهره می‌جوید. در ادامه این نوشته به این واژگان که در گرایش شبکه‌های عصبی، معانی ویژه‌ای دارند، بیشتر خواهیم پرداخت.

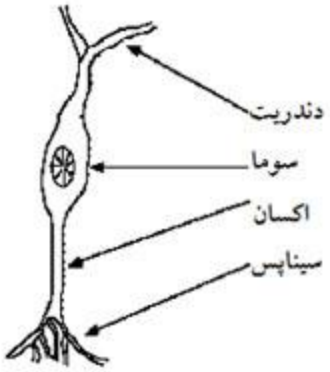
شباهت با مغز:

اگرچه مکانیسم‌های دقیق کارکرد مغز انسان (یا حتی جانوران) به‌طور کامل شناخته شده نیست، اما با این وجود جنبه‌های شناخته شده‌ای نیز وجود دارند که الهام بخش تئوری شبکه‌های عصبی بوده‌اند. به‌عنوان مثال، یکی ازسلول‌های عصبی، معروف به نرون (Neuron) است که دانش بشری آن را به‌عنوان سازنده اصلی مغز می‌انگارد. سلول‌های عصبی قادرند تا با اتصال به‌یکدیگر تشکیل شبکه‌های عظیم بدهند. گفته می‌شود که هر نرون می‌تواند به هزار تا ده هزار نرون دیگر اتصال یابد (حتی در این مورد عدد دویست هزار هم به عنوان یک حد بالایی ذکر شده است). قدرت خارق‌العاده مغز انسان از تعداد بسیار زیاد نرون‌ها و ارتباطات بین آنها ناشی می‌شود. ساختمان هر یک از نرون‌ها نیز به‌تنهایی بسیار پیچیده است. هر نرون از بخش‌ها و زیرسیستم‌های زیادی تشکیل شده است که از مکانیسم‌های کنترلی پیچیده‌ای استفاده می‌کنند. سلول‌های عصبی می‌توانند از طریق مکانیسم‌های الکتروشیمیایی اطلاعات را انتقال دهند. برحسب مکانیسم‌های به‌کاررفته در ساختار نرون‌ها، آنها را به بیش از یکصدگونه متفاوت طبقه‌بندی می‌کنند. در اصطلاح فنی، نرون‌ها و ارتباطات بین آنها، فرایند دودویی(Binary)، پایدار (Stable) یا همزمان (Synchronous)

محسوب نمی‌شوند. در واقع، شبکه‌های عصبی شبیه‌سازی شده یا کامپیوتری، فقط قادرند تا بخش کوچکی از خصوصیات و ویژگی‌های شبکه‌های عصبی بیولوژیک را شبیه‌سازی کنند. در حقیقت، از دید یک مهندس نرم‌افزار، هدف از ایجاد یک شبکه عصبی نرم‌افزاری، بیش از آنکه شبیه‌سازی مغز انسان باشد، ایجاد مکانیسم دیگری برای حل مسائل مهندسی با الهام از الگوی رفتاری شبکه‌های بیولوژیک است.

روش کار نرون‌ها :

در شکل یک، نمای ساده شده‌ای از ساختار یک نرون بیولوژیک نمایش داده شده است. به‌طور خلاصه، یک نرون بیولوژیک، پس از دریافت سیگنال‌های ورودی (به شکل یک پالس الکتریکی) از سلول‌های دیگر، آن سیگنال‌ها را با یکدیگر ترکیب کرده و پس از انجام یک عمل (operation) دیگر بر روی سیگنال ترکیبی، آن را به‌صورت خروجی ظاهر می‌سازد.

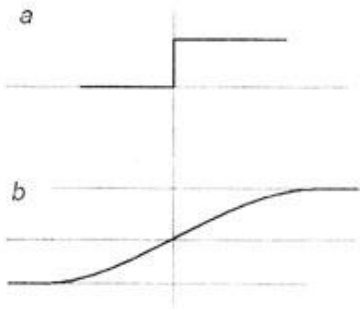


همان‌طور که در تصویر مشاهده می‌کنید، نرون‌ها از چهار بخش اصلی ساخته شده‌اند. دندریت‌ها (Dendrite)، سوما (Soma)، اکسان (Axon) و بالاخره، سیناپس (synapse) دندریت‌ها، همان اجزایی هستند که به‌شکل رشته‌های طویل از مرکز سلول به اطراف پراکنده می‌شوند. دندریت‌ها نقش کانال‌های ارتباطی را برای انتقال‌دادن سیگنال‌های الکتریکی به مرکز سلول بر عهده دارند. در انتهای دندریت‌ها، ساختار بیولوژیکی ویژه‌ای به‌نام سیناپس واقع شده است که نقش دروازه‌های اتصالی کانال‌های ارتباطی را ایفا می‌کند. در واقع سیگنال‌های گوناگون از طریق سیناپس‌ها و دندریت‌ها به مرکز سلول منتقل می‌شوند و در آنجا با یکدیگر ترکیب می‌شوند. عمل ترکیب که به آن اشاره کردیم، می‌تواند یک عمل جمع جبری ساده باشد. اصولاً اگر چنین نیز نباشد، در مدل‌سازی ریاضی می‌توان آنرا یک عمل جمع معمولی در نظر گرفت که پس از آن تابع ویژه‌ای بر روی سیگنال اثر داده می‌شود و خروجی به شکل سیگنال الکتریکی متفاوتی از طریق اکسان (و سیناپس آن) به سلول‌های دیگر انتقال داده می‌شود. البته تحقیقات جدید نمایانگر این واقعیت هستند که نرون‌های بیولوژیک بسیار پیچیده‌تر از مدل ساده‌ای هستند که در بالا تشریح شد. اما همین مدل ساده می‌تواند زیربنای مستحکمی برای دانش شبکه‌های عصبی مصنوعی (Artificial Neural Network = ANN) تلقی گردد و متخصصان گرایش شبکه‌های عصبی یا هوش مصنوعی می‌توانند با پیگیری کارهای دانشمندان علوم زیست‌شناسی، به بنیان‌گذاری ساختارهای مناسب‌تری در آینده دست بزنند.

مدل ریاضی :

در متون فنی برای نمایش مدل ساده‌ای که در بالا تشریح گردید، به‌طور معمول از شکلی مشابه شکل ۲ استفاده می‌شود. در این شکل کلاسیک، از علامت p برای نشان دادن یک سیگنال ورودی استفاده می‌شود. در واقع در این مدل، یک سیگنال ورودی پس از تقویت (یا تضعیف) شدن به اندازه پارامتر w، به‌صورت یک سیگنال الکتریکی با اندازه pw وارد نرون می‌شود. به‌جهات ساده‌سازی مدل ریاضی، فرض می‌شود که در هسته سلول عصبی، سیگنال ورودی با سیگنال دیگری به اندازه b جمع می‌گردد. در واقع سیگنال b خود به معنی آن است که سیگنالی به اندازه واحد در پارامتری مانند b ضرب (تقویت یا تضعیف) می‌شود. مجموع حاصل، یعنی سیگنالی به اندازه pw + b، قبل از خارج شدن از سلول تحت عمل یا فرآیند دیگری واقع می‌شود که در اصطلاح فنی به آن تابع انتقال (Transfer Function) می‌گویند. این موضوع در شکل به‌وسیله جعبه‌ای نمایش داده شده است که روی آن علامت f قرار داده شده است. ورودی این جعبه همان سیگنال pw + b است و خروجی آن یا همان خروجی سلول، با علامت a نشانه گذاری شده است. در ریاضی، بخش آخر مدل‌سازی توسط رابطه (a = f(pw + b نمایش داده می‌شود. پارامتر w یا همان ضریبی که سیگنال ورودی p در آن ضرب می‌شود، در اصطلاح ریاضی به نام پارامتر وزن یا weight نیز گفته می‌شود. زمانی‌که از کنار هم قرار دادن تعداد بسیار زیادی از سلول‌های فوق یک شبکه عصبی بزرگ ساخته شود، شبکه‌ای در دست خواهیم داشت که رفتار آن علاوه بر تابع خروجی f، کاملاً به مقادیر w و b وابسته خواهد بود. در چنین شبکه بزرگی، تعداد بسیار زیادی از پارامترهای w و b باید توسط طراح شبکه مقداردهی شوند. این پروسه از کار، در اصطلاح دانش شبکه‌های عصبی، به فرآیند یادگیری معروف است. در واقع در یک آزمایش واقعی، پس از آن‌که سیگنال‌های ورودی چنین شبکه بزرگی اتصال داده شدند، طراح شبکه با اندازه‌گیری خروجی و با انتخاب پارامترهایw و b به‌گونه‌ای که خروجی مطلوب به‌دست آید، شبکه را <آموزش> می‌دهد. به این ترتیب پس از آنکه چنین شبکه به ازای مجموعه‌ای از ورودی‌ها برای ساختن خروجی‌های مطلوب <آموزش> دید، می‌توان از آن برای حل مسائلی که از ترکیب متفاوتی از ورودی‌ها ایجاد می‌شوند، بهره برد. تابع f می‌تواند بر حسب کاربردهای گوناگون به‌طور ریاضی، به شکل های متفاوتی انتخاب شود. در برخی از کاربردها، پاسخ مسائل از نوع دودویی است. مثلاً مسأله به‌گونه‌ای است که

خروجی شبکه عصبی باید چیزی مانند <سیاه> یا <سفید> (یا <آری> یا <نه>) باشد. در واقع چنین مسائلی نیاز به آن دارند که ورودی‌های دنیای واقعی به مقادیر گسسته مانند مثال فوق تبدیل شوند. حتی می‌توان حالاتی را در نظر گرفت که خروجی دودویی نباشد، اما همچنان گسسته باشد. به عنوان مثال، شبکه‌ای را در نظر بگیرید که خروجی آن باید یکی از حروف الفبا، مثلاً از بین کاراکترهای اسکی (یا حتی یکی از پنجاه هزار کلمه متداول زبان انگلیسی) باشد. در چنین کاربردهایی، روش حل مسئله نمی‌تواند صرفاً بر جمع جبری سیگنال‌های ورودی تکیه نماید. در این کاربردها، ویژگی‌های خواسته شده فوق، در تابع خروجی یا تابع انتقال f گنجانیده می‌شوند. مثلاً اگر قرار باشد خروجی فقط یکی از مقادیر صفر یا یک را شامل شود، در این صورت می‌توان تابع خروجی شبکه عصبی را به شکل بخش a شکل شماره ۳ انتخاب کرد. در این حالت، خروجی چنین شبکه‌ای فقط می‌تواند بر حسب ورودی‌های متفاوت، مقدار یک یا صفر باشد.

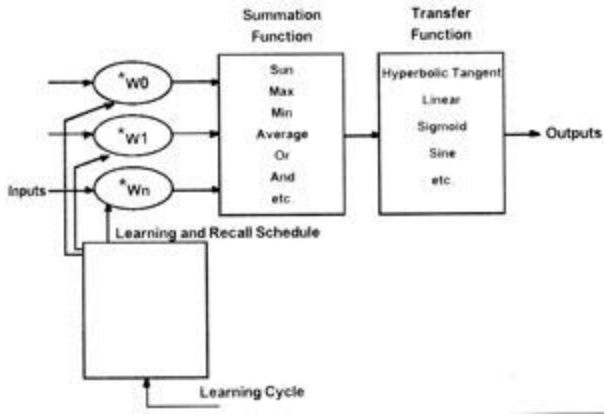


در گروه دیگری از مسائلی که حل آن‌ها به شبکه‌های عصبی واگذار می‌شود، خروجی شبکه عصبی الزاماً بین مقادیر معلوم و شناخته شده واقع نمی‌شود. مسائلی از نوع شناسایی الگوهای تصویری، نمونه‌ای از چنین مواردی محسوب می‌شوند. شبکه‌های عصبی در این موارد، باید به‌گونه‌ای باشند که قابلیت تولید مجموعه نامتناهی از پاسخ‌ها را داشته باشند. رفتار حرکتی یک روبات نمونه‌ای از < هوشی> است که چنین شبکه‌های عصبی می‌توانند فراهم آورند. اما در چنین شبکه‌هایی هم لازم خواهد بود که خروجی بین مقادیر مشخصی محدود شده باشد (موضوع محدود شدن خروجی بین دو مقدار حدی ماکزیمم و مینیمم را در اینجا با موضوع قبلی اشتباه نگیرید. در این مورد خروجی مسأله اساساً گسسته نیست و حتی در بین چنین مقادیر حدی،

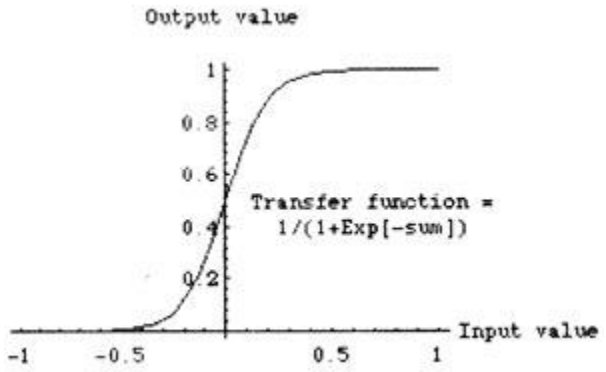
می‌توان به تعداد نامتناهی خروجی دست یافت). اهمیت این موضوع زمانی آشکار می‌شود که از مثال واقعی کمک گرفته شود. فرض کنید قراراست از شبکه عصبی برای کنترل حرکت بازوی یک روبات استفاده شود. در صورتی‌که خروجی یک شبکه عصبی برای کنترل نیروی حرکتی به‌کار گرفته شود، طبیعی خواهد بود که اگر خروجی شبکه محدود نشده باشد، ممکن است بازوی روبات بر اثر حرکت بسیار سریع، به خود و یا محیط اطراف آسیب برساند. در چنین مواردی ممکن است از تابع انتقال به‌شکل بخش b شکل شماره ۳ استفاده شود. قبل از آنکه به بخش دیگری از موضوع شبکه‌های عصبی بپردازیم، باید یک نکته را یادآوری کنیم که همان‌طور که در ابتدای این بخش تشریح شد، سلول‌های عصبی دارای ورودی‌های متعددی هستند و خروجی آنها نیز الزاماً محدود به یک خروجی نیست. بر این اساس زمانی که بخواهیم از مدل‌سازی ریاضی برای مدل کردن یک سلول عصبی استفاده کنیم، به‌جای آن‌که همانند شکل ۲ از یک ورودی p و یک خروجی a استفاده کنیم، از یک بردار p و یک بردار a سخن می‌گوییم. بدین ترتیب بدون آنکه نیاز به اعمال تغییری در این تصویر داشته باشیم، می‌توانیم از آن برای مدل‌سازی سلولی با n ورودی (p۱,p۲,p۳ . . . pn) و به همین ترتیب m خروجی (a۱,a۲,am) استفاده کنیم. توجه داشته باشید که در این صورت، تعداد عناصر b و w نیز به تناسب افزایش می‌یابند و هر یک به n عدد افزایش می‌یابند.

پیاده‌سازی‌های الکترونیکی نرون‌های مصنوعی :

نرم‌افزارهایی که در آن‌ها از شبکه‌های عصبی استفاده شده است، نرون‌های شبکه را المان پردازنده (element Processing) می‌نامند. به‌طور معمول در محصولات نرم‌افزاری، المان‌های پردازنده قابلیت بسیار بیشتری از نمونه ساده‌شده‌ای که در بخش‌های پیشین تشریح کردیم، دارند. در شکل شماره ۴، نمایی با جزئیات بیشتر از یک نرون مصنوعی را نشان می‌دهد.



در این مدل، ورودی‌ها در نخستین گام، در ضریب وزنی (weighting factor) متناظر خود ضرب می‌شوند. در مرحله بعد، ورودی‌هایی که تغییر مقیاس داده‌اند وارد واحدی می‌شوند که در آن سیگنال‌های ورودی با هم ترکیب می‌شوند. به‌طور معمول عمل ترکیب در این واحد همان عمل جمع جبری است، اما در اصول، می‌توان در این واحد، ورودی‌ها را به روش‌های دیگری علاوه بر عمل جمع معمولی نیز با یکدیگر ترکیب کرد. مثلاً می‌توان به‌جای عمل جمع، از عمل متوسط گیری، انتخاب بزرگترین، انتخاب کوچک‌ترین، عمل OR یا AND منطقی هم استفاده کرد. در واقع هدف نهایی در این واحد آن است که از تعداد n ورودی، یک سیگنال خروجی برای ارائه به بخش‌های بعدی فرایند، به‌دست آید. انتخاب نوع <عمل> در این واحد، موضوع دقیقی است که کاملاً به کاربرد مسأله وابسته است.



به‌طور معمول در نرم‌افزارهای تجاری، امکان انتخاب و حتی ساختن توابع گوناگون برای این واحد، از طرف نرم‌افزار به کاربران داده می‌شود. حتی می‌توان کاربردهایی یافت که در آنها، عمل ترکیب در این واحد، وابسته به زمان باشد و در زمان‌های گوناگون پردازش مسأله، عملیات مختلفی برای ترکیب کردن ورودی‌ها به‌کار برده شوند. در هر صورت، پس از آنکه ورودی‌ها با یکدیگر ترکیب شدند، سیگنال حاصل به واحد دیگری که در آن تابع انتقال یا Function Transfer به سیگنال اعمال می‌شود، هدایت می‌گردد. خروجی این بخش، سیگنال‌های حقیقی خواهند بود. بدین ترتیب جعبه‌ای در دست خواهیم داشت که تعداد n عدد سیگنال ورودی را به m عدد سیگنال خروجی تبدیل می‌کند. در عمل توابع انتقالی که در بخش انتهایی نمودار شکل ۴ به‌کار برده می‌شوند، معمولاً یکی از توابع سینوسی، تانژانت هذلولی، sigmoid و نظایر این‌ها است. در تصویر شماره ۵، نمونه‌ای از یک تابع انتقال از نوع sigmoid نمایش داده شده است. همانطور که در این شکل مشاهده می‌کنید، این تابع انتقال، سیگنال خروجی واحد ترکیب را به سیگنال خروجی تبدیل می‌کند که مقدار (یا اندازه آن) بین صفر و یک می‌تواند باشد. در عمل، سیگنالِ خروجی یک المان پردازنده می‌تواند برحسب نوع کاربرد، به المان‌های پردازشی دیگر و یا به اتصالات دیگر خارج از شبکه عصبی هدایت شود. در واقع تمامی شبکه‌های عصبی بر اساس ساختار المان‌های پردازشی فوق کار می‌کنند. در قسمت بعدی این مقاله به تشریح عملیات در شبکه‌های عصبی و آموزش این شبکه‌ها می‌پردازیم.{۱}

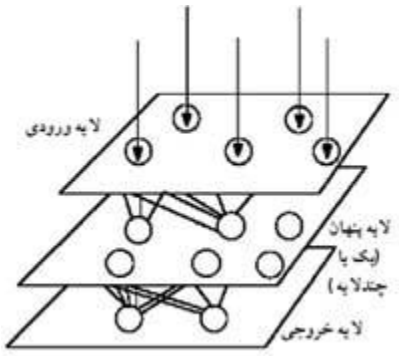
عملیات شبکه‌های عصبی - Neural Networks (قسمت دوم):

اشاره :

در قسمت قبلی مقاله (ماهنامه شبکه - شماره۵۱) با مفاهیم کلی شبکه‌های عصبی و اجزای آن و نحوه پردازش در آن‌ها آشنا شدید. در این قسمت به مراحل طراحی یک شبکه عصبی می پردازیم . سپس نظری به مقوله بسیار مهم آموزش در چنین شبکه‌هایی خواهیم داشت و روش‌های مختلف آموزش را بیان می‌کنیم.

عملیات شبکه‌های عصبی:

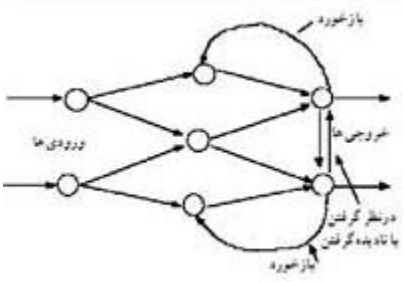
تا اینجا تمام توجه ما معطوف ساختار درونی یک نرون مصنوعی یا المان پردازشی بود. اما بخش مهم دیگری در مراحل طراحی یک شبکه عصبی نیز وجود دارد. در واقع هنر یک طراح شبکه‌های عصبی می‌تواند در چگونگی ترکیب نرون‌ها در یک شبکه (neuran Clustering)، متجلی شود. علوم بیولوژی نشان داده‌اند که کلاسترینگ نرون‌ها در شبکه عصبی مغز ما به‌گونه‌ای است که ما را قادر می‌سازد تا اطلاعات را به صورتی پویا، تعاملی و خودسامان (Selforganizing) پردازش کنیم. در شبکه‌های عصبی بیولوژیک، نرون‌ها در ساختاری سه بعدی به یکدیگر اتصال یافته‌اند. اتصالات بین نرون‌ها در شبکه‌های عصبی بیولوژیک آنقدر زیاد و پیچیده‌است که به هیچ وجه نمی‌توان شبکه مصنوعی مشابهی طراحی کرد. تکنولوژی مدارات مجتمع امروزی به ما امکان می‌دهد که شبکه‌های عصبی را در ساختارهای دو بعدی طراحی کنیم. علاوه بر این، چنین شبکه‌های مصنوعی دارای تعداد محدودی لایه و اتصالات بین نرون‌ها خواهند بود. بدین ترتیب، این واقعیات و محدودیت‌های فیزیکی تکنولوژی فعلی، دامنه کاربردهای شبکه‌های عصبی مبتنی‌بر تکنولوژی سیلیکونی را مشخص می‌سازند. ساختار شبکه‌های عصبی امروزی، از لایه‌های نرونی تشکیل شده است. در چنین ساختاری، نرون‌ها علاوه بر آنکه در لایه خود به شکل محدودی به یکدیگر اتصال داده شده‌اند، از طریق اتصال بین لایه‌ها نیز به نرون‌های طبقات مجاور ارتباط داده می‌شوند. در شکل ۱ نمونه‌ای از ساختار لایه‌ای یک شبکه عصبی مصنوعی نمایش داده شده است (تعداد اتصالات ممکن بین نرون‌ها را در چنین ساختاری با تعداد اتصالات بین نرون‌های مغز انسان، مقایسه کنید).



در این توپولوژی، گروهی از نرون‌ها از طریق ورودی‌های خود با جهان واقعی ارتباط دارند. گروه دیگری از نرون‌ها نیز از طریق خروجی‌های خود، جهان خارج را می‌سازند. در واقع این <جهان خارج> تصویری است که شبکه عصبی از ورودی خود می‌سازد یا می‌توان چنین گفت که جهان خارج <تصوری> است که شبکه عصبی از ورودی خود دارد. خلاصه آنکه در توپولوژی فوق، مابقی نرون‌ها از دید پنهان هستند. تلاش محققان در زمینه شبکه‌های عصبی نشان داده است که شبکه‌های عصبی، چیزی بیشتر از یک مشت نرون که به یکدیگر اتصال داده شده‌اند، هستند. حتی گروهی از محققان سعی داشته‌اند که از اتصالات تصادفی برای ارتباط دادن نرون به یکدیگر استفاده کنند که در این زمینه به نتایج جالب توجهی دست نیافتند. امروزه مشخص شده است که در ساده‌ترین مغزهای بیولوژیک مانند مغز مارها هم ارتباطات بین نرون‌ها بسیار ساخت‌یافته است. در حال حاضر یکی از ساده‌ترین روش‌های ارتباط دهی نرون‌ها در شبکه‌های عصبی، آن است که ابتدا نرون‌ها در گروه‌های مشخصی به صورت لایه‌های نرونی سازمان‌دهی می‌شوند و پس از تامین ارتباطات بین‌نرونی در هر لایه، ارتباطات بین لایه‌ها نیز برقرار می‌شوند. اگرچه در کاربردهای مشخصی می‌توان با موفقیت از شبکه‌های عصبی تک لایه استفاده کرد، اما رسم بر آن است که شبکه‌های عصبی حداقل دارای سه لایه باشند (همانطور که قبلاً اشاره شد، لایه ورودی، لایه خروجی و نهایتاً لایه پنهان یا لایه میانی).

در بسیاری از شبکه‌های عصبی، اتصالات بین‌نرونی به گونه‌ای است که نرون‌های لایه‌های میانی، ورودی خود را از تمام نرون‌های لایه پایینی خود (به طور معمول لایه نرون‌های ورودی) دریافت می‌کنند. بدین ترتیب در یک شبکه عصبی، سیگنال‌ها به تدریج از یک لایه نرونی به لایه‌های بالاتر حرکت می‌کنند و در نهایت به لایه آخر و خروجی شبکه می‌رسند. چنین مسیر در اصطلاح فنی feed forward نامیده می‌شود. ارتباطات بین‌نرونی در شبکه‌های عصبی از اهمیت بسیار زیادی برخوردار هستند و به نوعی قدرت یک شبکه عصبی را تعیین می‌کنند. قاعده آن است که ارتباطات بین نرونی را به دو گروه تقسیم‌بندی می‌کنند. یک نوع از ارتباطات بین نرونی، به‌گونه‌ای هستند که باعث جمع شدن سیگنال در نرون بعدی می‌شوند. گونه دوم ارتباطات بین نرونی باعث تفریق سیگنال در نرون بعدی می‌شوند. در اصطلاح محاوره‌ای گروهی از ارتباطات انگیزش ایجاد می‌کنند و گروه دیگر

ممانعت به عمل می‌آورند.در مواردی، نرون مشخصی از شبکه عصبی تمایل دارد که سیگنال دیگر نرون‌های لایه خود را نادیده بگیرد. چنین حالتی به‌طور معمول در لایه خروجی ایجاد می‌شود. به عنوان مثال، در کاربردهای تشخیص متن (OCR)، فرض کنید که احتمال آنکه کاراکتر مورد شناسایی، حرف P باشد برابر با ۸۵ درصد تعیین شده است و به همین ترتیب احتمال آنکه کاراکتر مورد نظر حرف F باشد، ۶۵ درصد تخمین زده است. در این وضعیت، سیستم باید کاراکتری را برگزیند که دارای درصد احتمال بزرگ‌تر است. در نتیجه در این شبکه عصبی، نرون‌هایی که خروجی F را تجویز می‌کنند، باید نادیده گرفته شوند یا inhibit شوند. به چنین فرایندی، lateral inhibition گفته می‌شود. نوع دیگری از ارتباط بین نرونی در شبکه‌های عصبی به ارتباط بازخورد یا feedback معروف است. در این نوع از ارتباطات، خروجی یک لایه نرونی به لایه قبلی (یا به لایه‌ای که چند مرحله پایینتر است) اتصال داده می‌شود. در شکل ۲ نمونه‌ای از یک شبکه عصبی نمایش داده شده که در آن از ارتباط بازخوردی استفاده شده است. در نرم‌افزارهای پیشرفته شبکه‌های عصبی، کاربر و طراح شبکه عصبی می‌تواند نوع ارتباطات بین نرون‌ها و لایه‌های آنها را تعیین کند.



آموزش شبکه‌های عصبی:

تا اینجا از ساختار شبکه‌های عصبی صحبت کردیم. گفتیم که شبکه‌های عصبی می‌توانند بر اساس طراحی خود سیگنال‌های ورودی را پردازش کنند و به سیگنال‌های خروجی مورد نظر تبدیل نمایند. به طور معمول، پس از آنکه یک شبکه عصبی طراحی و پیاده‌سازی شد، باید پارامترهای w و b (که قبلاً معرفی کردیم) به ازای مجموعه‌هایی از سیگنال‌های ورودی، به‌گونه‌ای تنظیم شوند که سیگنال‌های خروجی شبکه خروجی

مطلوب را تشکیل دهند. چنین فرایندی را آموزش دیدن شبکه عصبی می‌نامند (در نخستین مرحله آموزش، مقادیر w و b به‌طور تصادفی انتخاب می‌شوند. زیرا تا این پارامترها مقدار نداشته باشند، شبکه عصبی قابل استفاده نخواهد بود) در حین آموزش دیدن شبکه عصبی (یعنی به تدریج همزمان با افزایش دفعاتی که مقادیر پارامترها برای رسیدن به خروجی مطلوب‌تر، تنظیم می‌شوند) مقدار پارامترها به مقدار حقیقی و نهایی خود نزدیک‌تر می‌شوند. به‌طور کلی دو روش برای آموزش دادن شبکه‌های عصبی وجود دارد. روش supervised و روش unsupervised. روش نخست، شامل مراحلی است که در بخش قبل، به‌طور مختصر تشریح شد. اما در روش unsupervised، شبکه عصبی باید بدون کمک گرفتن از جهان خارج، بتواند کار آموزش را انجام دهد. واقعیت آن است که در عمل از روش supervised و یا حداکثر از روش‌های ترکیبی استفاده می‌شود و فرایند آموزش unsupervised به شکل خالص تنها وعده‌ای است که شاید در آینده بتواند تحقق یابد. در حال حاضر و در کاربردهای پیشرفته، از روش آموزش unsupervised برای ایجاد تنظیمات اولیه بر روی سیگنال‌های ورودی شبکه‌های عصبی استفاده می‌شود و باقی مراحل آموزش شبکه به روش supervised ادامه می‌یابد. همان‌طور که قبلاً اشاره کردیم، در روش معمول آموزش شبکه‌های عصبی، از مجموعه شناخته‌شده‌ای از داده‌های ورودی و خروجی‌های متناظر آنها (training set data) برای آموزش دادن شبکه استفاده می‌شود. در چنین فرایندی، پس از اعمال مجموعه‌های داده‌های آموزشی، پارامترهای شبکه به تدریج به سمت مقادیر نهایی خود همگرا می‌شوند. بسته‌های نرم‌افزاری پیشرفته تولید و طراحی شبکه‌های عصبی، دارای ابزارهایی هستند که بر روند آموزش شبکه مدیریت می‌کنند. چنین ابزارهایی می‌توانند سرعت همگرایی پارامترهای شبکه را زیر نظر بگیرند و به عنوان مثال، اجازه دهند که پارامترهای یک شبکه مشخص، در طول چندین روز به دقت کافی و مورد نظر طراحان خود برسد.

در مواردی ممکن است که شبکه عصبی اصولاً موفق به فراگیری نشود. بدین معنی که پارامترهای شبکه پس از زمان‌های طولانی به مقدار مشخصی همگرا نشود. چنین مواردی ممکن است بر اثر ناکافی بودن داده‌های آموزشی و یا اصولاً نقص طراحی شبکه ایجاد شوند. حتی مواردی در عمل وجود دارند که شبکه عصبی مشخصی،

بر اثر آموزش بیش از حد، اصطلاحا over trained شود. توجه داشته باشید که فرایند آموزش شبکه‌های عصبی فقط به ازای زیر مجموعه‌ای از داده‌هایی که قرار شبکه آنها را در کاربرد حقیقی خود پردازش کند، آموزش داده می‌شوند. درصورتی‌که تعداد داده‌های آموزشی یک شبکه عصبی بیش از اندازه زیاد باشد (در واقع از تمامی داده‌های مسئله برای آموزش دادن به شبکه استفاده شود)، شبکه عصبی به جای آنکه آموزش ببیند، به حالتی می‌رسد که به آن حفظ کردن اطلاعات می‌گویند. در واقع به جای آنکه یک شبکه عصبی برای حل مسئله از هوش خود کمک بگیرد، از محفوظات خود استفاده می‌کند!

پس از آنکه یک شبکه عصبی به اندازه کافی آموزش دید، طراح یا کاربر شبکه می‌تواند پارامترهای شبکه را قفل کند (هر چند که در مواردی پارامترهای شبکه آزاد گذارده می‌شوند تا در طول کاربرد واقعی بازهم شبکه آموزش ببیند). در این مرحله شبکه عصبی برای کاربرد واقعی خود و حل مسائل آماده خواهد بود. در برخی از ابزارهای تولید و طراحی شبکه‌های عصبی، کل شبکه عصبی به همراه پارامترهای قفل شده آن، تبدیل به نرم‌افزار مستقلی (مثلاً یک فایل dll) می‌شوند که می‌توان از آن در پروژه‌های مشخصی استفاده کرد. در برخی از موارد دیگر، چنین شبکه‌هایی پس از آموزش دیدن، به شکل سخت‌افزاری در قالب یک مدار مجتمع (IC) به تولید انبوه یا نیمه انبوه می‌رسند.

آموزش unsupervised یا تطبیقی (Adaptive) :

در مورد این روش آموزش گفتیم که شبکه عصبی بدون در اختیار داشتن داده‌های خروجی، در معرض آموزش قرار می‌گیرد. در واقع سیستم به تنهایی و بدون کمک خارجی باید با توجه به شکل سیگنال‌های خروجی خود، درباره درستی و نادرستی آنها تصمیم‌گیری نماید. در دنیای واقعی شرایط بسیار زیادی وجود دارند که در آنها مجموعه اطلاعات کافی برای آموزش دادن به سیستم فراهم نیستند. تحقیقات نظامی یکی از گرایش‌هایی است که به این موضوع توجه دقیقی دارد. به عنوان مثال گفته می‌شود که شرایط جنگی به دلیل فراوانی پارامترها و تکنیک‌های نظامی متغیر و پیشرفت‌های تکنولوژی نظامی، از نمونه مواردی است که در آنها به هیچ وجه نمی‌توان مجموعه داده‌های آموزشی کافی به دست آورد. در این زمینه یکی از محققان شبکه‌های عصبی، به نام

Tuevo Kohonen (از دانشگاه هلسینکی) فعالیتی جدی دارد. کوهنن با تحقیقات در ساختارهای عصبی غیرمتعارف، به پژوهش در این زمینه ادامه می‌دهد. کوهنن، نرون‌های شبکه‌عصبی را فیلدهای مختلفی تقسیم‌بندی می‌کند. در روش کوهنن، نرون‌های هر فیلد <مرتب توپولوژیک> یا Topologically ordered محسوب می‌شوند (توپولوژی نام شاخه‌ای از ریاضیات است که در آن نگاشت از یک فضا به فضای دیگر بدون تغییر مشخصه‌های هندسی، مورد بررسی قرار می‌گیرد). گروه‌بندی‌های سه‌بعدی که در ساختار مغز پستانداران یافت شده است، نمونه‌ای از مرتب‌سازی توپولوژیک محسوب می‌شوند. کوهنن معتقد است که فقدان ملاحظات توپولوژیک در مدل‌های عصبی امروزی، باعث می‌شود که شبکه‌های عصبی امروزی، مدل‌های ساده شده‌ای از شبکه‌های عصبی واقعی موجود در مغز محسوب شوند. در هر صورت این حوزه از مبحث شبکه‌های عصبی، هنوز در مرحله تحقیقات آزمایشگاهی قرارداد و کاربرد واقعی نیافته است.

تفاوت‌های شبکه‌های عصبی با روش‌های محاسباتی متداول و سیستم‌های خبره :

گفتیم که شبکه‌های عصبی روش متفاوتی برای پردازش و آنالیز اطلاعات ارائه می‌دهند. اما نباید این گونه استنباط شود که شبکه‌های عصبی می‌توانند برای حل تمام مسائل محاسباتی مورد استفاده واقع شوند. روش‌های محاسباتی متداول همچنان برای حل گروه مشخصی از مسائل مانند امور حسابداری، انبارداری و محاسبات عددی مبتنی بر فرمول‌های مشخص، بهترین گزینه محسوب می‌شوند. جدول ۱، تفاوت‌های بنیادی دو روش محاسباتی را نشان می‌دهد.

شبکه‌های عصبی مصنوعی	روش محاسباتی متداول (شامل سیستم‌های خبره)	مشخصه
موازی	ترتیبی	روش پردازش
(estault (right brained	منطقی(left brained)	توابع
با مثال(Socratically)	به کمک قواعد (didactically)	روش فراگیری

کار برد	**حسابداری، واژه پردازی، ریاضیات، ارتباطات دیجیتال**	**پردازش حسگرها، تشخیص گفتار، نوشتار، الگو**

جدول ۱

سیستم‌های خبره، انشعابی از روش محاسباتی متداول محسوب می‌شود و در مواردی هم به آن نسل پنجم محاسبات نام داده‌اند (نسل اول از کلید و سیم‌بندی استفاده می‌کرد، نسل دوم با اختراع ترانزیستور ایجاد شد، نسل سوم از فناوری مدارات مجتمع استفاده می‌کرد، نسل چهارم با به وجود آمدن زبان‌های سطح بالا آغاز شد و نسل پنجم شامل هوش مصنوعی می‌شود). به طور معمول، یک سیستم خبره شامل دو بخش عمده می‌شود. یک بخش یا موتور استنتاجی و یک پایگاه دانایی (Knowledge base). موتور استنتاجی، بخشی است که رابط کاربر را مدیریت می‌کند و بر فایل‌ها و دسترسی به برنامه‌ها و برنامه‌ریزی کنترل دارد. پایگاه دانایی شامل اطلاعاتی در ارتباط با یک مسئله مشخص است. این پایگاه به متخصصان اجازه می‌دهد که قواعد فرایند مشخصی را تعریف نماید. چنین متخصصی نیازی به دانستن روش‌های برنامه‌نویسی نخواهد داشت. او تنها باید کاری که از کامپیوتر می‌خواهد را درک کند و شناخت کافی از روش عمل سیستم داشته باشد. درواقع پوسته سیستم بخشی است که به کامپیوتر می‌گوید چه‌کار باید انجام دهد. برنامه لازم برای حل مسئله توسط خود سیستم تولید خواهد شد. تلاش‌هایی که برای اجرایی کردن سیستم‌های خبره به کار گرفته شده‌اند، با مشکلات مشترکی مواجه بوده‌اند. با افزایش سطح پیچیدگی سیستم‌ها، منابع کامپیوتری مورد نیاز سیستم به شدت افزایش می‌یابند و سیستم با کندی بیش از حد روبرو می‌شود. در حقیقت تجربه نشان داده است که در وضعیت فعلی، سیستم‌های خبره تنها می‌توانند در مواقعی مفید واقع شوند که هدف محدود و مشخصی تعیین شده باشد. شبکه‌های عصبی در مسیری گام برمی‌دارند که ابزارها توانایی فراگیری و برنامه‌ریزی خود را داشته باشند. ساختارشبکه‌های عصبی به گونه‌ای است که قابلیت حل مسئله را بدون کمک فرد متخصص و برنامه‌ریزی خارجی داشته باشند. شبکه‌های عصبی قادر به یافتن الگوهایی در اطلاعات هستند که هیچ‌کس، هیچ‌گاه از وجود آنها اطلاع نداشته است.

درحالی‌که سیستم‌های خبره در عمل به موفقیت‌های بسیاری دست یافته‌اند، شبکه‌های عصبی در کاربردهایی همچون دید مصنوعی، تشخیص و تولید پیوسته گفتار، فراگیری ماشینی و نظایر آن با مشکلاتی روبرو بوده‌اند. در حال حاضر شبکه‌های عصبی کاملاً وابسته به سرعت پردازنده سیستم اجرا کننده هستند.{۱}

انواع شبکه‌ها از نظر برگشت پذیری :

۱. شبکه‌های پیش‌خور (Feed Forward) :

با توجه به مباحث بند ششم، در شبکه‌های تک لایه - بردار ورودی X توسط نرون‌ها را میتوان طبق رابطه ذیل نشان داد :

$$(y = f (W *X = b))$$

به بردار خروجی مرتبط می‌شود. این شبکه شکل ساده‌ای از شبکه‌های پیش‌خور است که مسیر پاسخ همواره رو به جلو پردازش می‌شود و به نرون‌های لایه (لایه‌های قبل) باز نمی‌گردد. اگر ایده شبکه‌های feed forward را به چند لایه تعمیم دهیم، هر لایه ماتریس وزن W و بردار ورودی X و بردار خروجی مختص خود را دارد.

v شبکه‌های برگشتی (Recurrent) :

تفاوت شبکه‌های برگشتی یا Recurrent با شبکه‌های پیش‌خور در آن است که در شبکه‌های برگشتی حداقل یک سیگنال برگشتی از یک نرون به همان نرون یا نرون‌های همان لایه یا لایه‌های قبل وجود دارد. شبکه‌های Recurrent بهتر می‌توانند رفتار مربوط به ویژه‌گی‌های زمانی و پویایی سیستم‌ها را نشان دهند. در این نوع شبکه ها که با توجه به ماهیت پویای مسئله طراحی می‌شوند بعد از مرحله یادگیری شبکه نیز پارامترها تغییر کرده و تصحیح می شوند (به طور مثال در پیش‌بینی ها بعد از گذر زمان مقایسه انجام شده و پارامترهای آزاد شبکه تنظیم می‌شوند) نوع خاصی از شبکه‌های برگشت پذیر به شبکه های هاپفیلد (Hopfield Network) موسوم هستند. در شبکه‌های هاپفیلد نرون‌ها نخست توسط ورودی مقدار اولیه می‌گیرند و شبکه به

گونه‌ای خود را تکرار می‌کند که نتیجه آن همگرایی به سمت الگوی مرجع است. در این شبکه همه نرون‌ها شبیه به یکدیگر عمل کرده و هیچ کدام از نرون‌ها به عنوان ورودی یا خروجی از هم متمایز نمی‌شوند. در تشکیل ساختارها یعنی تعداد سلول‌های عصبی و لایه‌ها و همچنین شرایط اولیه مسئله باید دقت نمود که موارد عملی و پیچیده به یک یا حتی تعداد بیشتری لایه مخفی ورودی و خروجی و تعداد زیادی وزن برای عوامل ورودی احتیاج دارند. بسیاری از ANN های تجاری شامل ۴ و یا حتی ۵ لایه هستند که هر کدام شامل ۱۰ تا ۱۰۰۰ المان پردازشی هستند. برخی از ANN ها حاوی میلیون‌ها المان پردازشی هستند.

برخی اهم کاربرد شبکه‌های عصبی در مباحث مهندسی صنایع :

شبکه‌های عصبی هنگامی کاربرد دارند که قانون شناخته شده‌ای وجود ندارد به عبارت دیگر هنگامی که یک دستورالعمل یا روش خاصی برای حل مسائل وجود دارد (همانند روش‌های مرسوم جبری در حل معادلات یا روش Simplex) بهترآنست که همان روش به صورت نرم افزار تهیه شود تا امکان دسترسی دقیق‌تر به پاسخ صحیح فراهم گردد.بنابراین شبکه‌های عصبی توان بالقوه‌ای برای حل مسائلی دارد که شبیه‌سازی آنها از طریق منطقی، تکنیک‌های تحلیلی و تکنولوژیهای استاندارد نرم‌افزاری مشکل است. بنابراین در مسائلی که تعداد زیادی از عوامل به صورت علت و معلولی به یکدیگر وابسته هستند و تشخیص مرز سیستم براحتی امکان پذیر نیست (همانند سیستم‌های پیچیده اقتصادی ، اجتماعی ، پیش‌بینی سری‌های زمانی مسائل مالی، بیمه و ...) و یا مسائل فنی که محاسبه یک پارامتر خاص به تعداد عوامل زیاد و نامشخص بستگر دارد (همانند کنترل بازوی ربات‌ها و ...) شبکه‌های عصبی می‌تواند کاربرد موثر داشته باشد. در ادامه برخی کاربردهای شبکه عصبی در مهندسی صنایع معرفی میگردد.

v دسته‌بندی و شناسایی الگو :

امروزه شبکه های عصبی طراحی شده اند که قادر هستند الگوهای مختلف را دسته بندی کرده و از یک دیگر تفکیک کنند. از این توانایی شبکه های عصبی در شناسایی و دسته بندی نقاط خارج از کنترل در کنترل کیفیت ((QC و همچنین شناسایی نقاطی که دارای خود همبستگی هستند بسیار استفاده شده است.کاربرد

شبکه هایی که برای تشخیص الگو طراحی شده اند در سیستمهای پشتیبان تصمیم گیری (DSS) ، آنجا که لازم است نظرات خبرگان از عامه تفکیک و دسته بندی گردد کاربرد دارد. اینگونه شبکه ها در بسیاری دیگر از تکنیکهای مهندسی صنایع نظیر تکنولوژی گروهی (GT) و مباحث دسته بندی بهینه ماشین آلات کاربرد فراوان دارد.

v پیش بینی :

هم اکنون بسیاری از شبکه های عصبی به گونه ای طراحی و آموزش داده شده اند که قادر به پیش بینی آینده بر اساس یادگیری و حفظ تجارب گذشته هستند. شبکه های پیش بینی قیمت نفت، بازار بورس از جمله این مسائل محسوب میشوند. شبکه های پیش بینی کننده بکارگیری وسیعی در مباحث کنترل موجودی، کنترل کیفیت و برنامه‌ریزی تعمیرات دارند. هم اکنون مقلاتی وجود دارد که بر اساس مدل های باکس-جنکینز طراحی و آموزش داده شده اند.

v مدل‌سازی و بهینه یابی :

این گونه شبکه ها بطور وسیع در مسائل برنامه‌ریزی تولید و OR کاربرد دارند، شبکه هایی طراحی شده اند که قادر به جستجوی نقطه بهینه سراسری Optimization Global در زمینه برنامه ریزی های ذیل هستند:

l Linear/Quadratic/Goal/Stochastic/Dynamic Programming

l TSP, Job- shop Scheduling

بیوانفورماتیک

❋ مقدمه

درک زبان خاموش اما جذاب سلولهای زنده، تلاش بیولوژی مولکولی مدرن است. از یک الفبای ساده چهار حرفی (آدنین، گوانین، سیتوزین، تیمین) دستورالعمل فرآیندهای حیاتی شکل می گیرد که پیچیده ترین بیان و گفتار آن‌ها "انسان" است. حجم فوق العاده زیاد داده های مولکولی و الگوهای مبهم و مرموز آن‌ها ما را بر آن داشته است تا به نیاز مطلق به پایگاه های داده کامپیوتری و ابزارهای تحلیل داده‌ها پی ببریم. تلاش و مبارزه ما یافتن روش های جدید، جهت مدیریت این داده‌ها حجیم و پیچیده است و نیز فراهم کردن بستری جهت دسترسی به ابزارهای کامپیوتری و تحلیل داده هاست تا میراث ژنتیکی خود را بیش تر درک کنیم و نقش آن را در سلامت و بیماری‌ها بیابیم.

❋ تعریف بیوانفورماتیک

بیوانفورماتیک شاخه ای از علم است که از کامپیوتر بهره می گیرد تا اطلاعات بیولوژی مولکولی- توالی‌های DNA یا پروتئین‌ها- را در کامپیوتر ذخیره نماید و با ابزارهای کامپیوتری و الگوریتمهای قدرتمند ریاضی آنالیز و تحلیل نماید.

❋ نیاز علم ژنتیک به کامپیوتر

بیولوژی مولکولی و علم ژنتیک مسایلی در پیش دارند که بیوانفورماتیک می تواند با به کار بردن این اطلاعات کامپیوتری شده به حل آن‌ها کمک نماید. حجم فوق العاده زیاد داده‌ها جهت نگهداری و مقایسه های گاهاً میلیونی رکوردها بسیار مشکل و گاهی غیر ممکن است. یکی از کاربردهای بیوانفورماتیک تحلیل این داده‌ها جهت پی بردن به معمای تکامل هستی است. حل این معما در میلیاردها نوکلئوتید درون ژنوم موجودات زنده نهفته است.

❋ پایگاه داده

پایگاه داده یا database یک برنامه کامپیوتری است که با روش های بسیار منظمی اطلاعات را در کامپیوتر ذخیره می کند و با سرعت فوق العاده ای در آن اطلاعات جست و جو کرده و نتیجه‌ی جستجو را ارایه می دهد.

❋ نقش پایگاه‌های داده

مشهورترین کاربرد بیوانفورماتیک در تحلیل توالی هاست. توالی های DNA مربوط بـه ارگـانیزم‌هـای مختلف جهت دستیابی سریع و مقایسه آن‌ها با یکدیگر، در پایگاه های داده ذخیره می شوند. پروژه ژنوم انسـان که از سال ۱۹۹٦ تا سال ۲۰۰۳ به طول انجامید نمونه ای از تحلیل توالی هاست. در ایـن پـروژه بـا اسـتفاده از کامپیوترهای بزرگ و روش های مختلف به دست آوردن توالی‌ها، همه ژنوم انسان تعیین توالی گردیـد و درون یک پایگاه داده قرار گرفت. با کامل شدن نقشه ژنوم انسان بیوانفورماتیک در تحقیقات سرطان‌ها به امید رسیدن به یک درمان موفق و نهایی بسیار با اهمیت شده است.

پایگاه های داده به دو دسته اصلی و فرعی تقسیم می شوند. نتایج تجربی حاصـل از تحقیقـات علمـی مانند توالی های نوکلئوتیدی یک ژن خاص، که در یک آزمایش تجربی به دست آمده است درون پایگـاه هـای داده‌ی اصلی قرار می گیرند. این داده‌ها خام و بدون تحلیل هستند. نمونه ای از این نـوع پایگـاه‌هـا GenBank است که توالی‌های نوکلئوتیدی را نگهداری می کند. این پایگاه توسط NCBI مدیریت می شود.

NCBI

در نوامبر ۱۹۸۸ بخش جدیدی به کتابخانه پزشکی ملی آمریکا (NLM) افـزوده شـد تـا روش‌هـای کـامپیوتری پردازش اطلاعات جهت هدایت تحقیقات Biomedical ایجاد شـود. ایـن بخـش مهـم، مرکـز ملـی اطلاعـات بیوتکنولوژی یا NCBI نام گرفت. کتابخانه ملی پزشکی آمریکا در سازمان ملـی بهداشـت آمریکـا (NIH) قـرار گرفته است. تجربه های موفق این کتابخانه در ایجاد و نگهداری پایگـاه‌هـای داده در زمینـه Biomedical و بـه عنوان بخشی از NIH باعث ایجاد یک برنامه تحقیقاتی در زمینـه بیولـوژی مولکـولی کـامپیوتری، شـده اسـت. امروزه NCBI به عنوان بزرگترین مرکز تحقیقات Biomedical در دنیا شـناخته شـده اسـت. انسـتیتوی NCBI جهت رسیدن به اهداف خود فعالیت های زیر را دنبال می کند: ۱- هدایت تحقیقاتی در زمینه مسـایل اصـولی Biomedicalدر سطح مولکولی با بهره‌گیری از روش های ریاضی و کـامپیوتری ۲- همکـاری بـا انسـتیتوهای دیگر NIH، دانشگاه‌ها، صنعت و دیگر سازمان‌های دولتی ۳- ایجاد ارتباط های علمی به وسـیله برگـزار کـردن همایش‌ها، کارگاه‌های آموزشی و سلسله سخنرانی‌ها ٤- برنامه های تحصیلاتی برای دانشجویان در زمینه هـای بیولوژی کامپیوتری توسعه و انتشار نرم افزارها و پایگاه‌های داده مختلف.

سـازمان NIH

سازمان NIH علاوه بر کتابخانه NLM از ۲۷ انستیتو و مرکز تحقیقاتی دیگر تشکیل شـده اسـت. از جملـه ایـن انستیتوها می توان به انستیتوی ملی سرطان انستیتوی ملی تحقیقات ژنوم انسان انستیتوی ملی دیابت و بیمـاری های کلیه و گوارش، و اانستیتوی ملی کودکان اشاره کرد.

وب سایت سازمان NIH با آدرس www.nih.gov قابل دسترسی است. این ۲۷ انستیتو زیر نظر بخش مرکزی NIH به نام Office of Director یا OD رهبری و مدیریت می شود. OD مسؤول تعیین سیاست های NIHدر برنامه‌ها، مدیریت‌ها و فعالیت‌ها است. در سایت NIH می توان با رؤسای این سازمان‌ها و انستیتوها به وسیله Email ارتباط برقرار کنید. آدرس هایEmail ، آدرس پستی و مشخصات کامل آنان در این سایت درج شده است.

موتور Entrez, The Life Sciences Search Engine

مقدمه

یکی از پر استفاده ترین ابزارهای جست و جوی اطلاعات بیولوژی، موتور جست و جوی Entrez است که NCBI آن را ایجاد کرده است. این موتور جست و جو قادر است به طور همزمان، بانک های اطلاعاتیMed Pub، توالی های نوکلئوتیدی(GenBank)، توالی های پروتئینی(protein) ، ساختمان پروتئین‌ها(Structure)، ژنوم کامل انسان و بعضی حیوانات، تاکسونومی، پایگاه داده بیماری‌های ژنتیکی (OMIM)و بسیاری موارد دیگر را جست و جو کند.

موتور جست و جوی Entrez از ارتباط تنگاتنگ رکوردها در بانک های مختلف استفاده می کند تا اطلاعات بیولوژی در مورد یک موضوع خاص را از پایگاه داده‌های مختلف بازیابی کرده و در اختیار کاربر قرار دهد. بنابراین Entrez یک پایگاه داده نیست بلکه یک سیستم یکپارچه بازیابی اطلاعات است. محققین می توانند از این دستور جست و جو بخواهند همه بانک های اطلاعاتی را همزمان جست و جو نماید یا این که جست و جوی خود را محدود به بانک اطلاعاتی خاصی نمایند. در این جزوه نحوه استفاده از این موتور جست و جو را برای شما شرح می دهیم.

جست و جوی همزمان در تمام پایگاه‌های داده

جهت استفاده از این موتور جست و جو، وب سایتNCBI را با آدرس www.ncbi.nlm.nih.gov باز کنید. در ابتدا، در مورد بخش های این صفحه توضیحاتی ارایه می دهیم. در بالا و سمت چپ این صفحه اینترنتی، لوگوی سایت NCBI را مشاهده می کنید. در زیر این لوگو، نوار سرمه ای رنگی وجود دارد که همواره در پایگاه‌های داده مختلف با شما خواهد بود. بر روی این نوار پر کاربردترین پایگاه‌های داده مانند PubMed و سرویس های مهمی مانند BLAST در دسترس هستند. برای استفاده از پایگاه‌های داده و سرویس‌ها کافی است روی نام آن بر روی این نوار کلیک نمایید. در زیر این نوار سرمه ای، نوار سرمه ای رنگ دیگری وجود دارد که مخصوص انتخاب پایگاه داده و وارد کردن کلمات یا عبارت جست و جو است.

این نوار از سه بخش تشکیل گردیده است. در سمت چپ این نوار منوی کشویی با عنوان Search وجود دارد که لیست پایگاه‌های اطلاعاتی NCBI را در اختیار شما قرار می دهد. گزینهAll Databases به صورت پیش فرض برای شما انتخاب گردیده است تا همه بانک‌های اطلاعاتی را جست و جو نماید. با کلیک بر روی فلش رو به پایین این منو آن را باز نمایید. بانک‌های مهمی چونPubMed ، Protein، Structrue و Booksرا در این منو مشاهده می نمایید. در صورتی که بخواهید بانک خاصی را به تنهایی جست و جو نمایید بر روی نام آن بانک کلیک نمایید. جست و جوی همزمان در همه بانک‌ها به وسیله موتور جست و جوEntrez از سه طریق امکان پذیراست: انتخاب گزینه All Database از منوی کشویی با عنوانSearch کلیک بر روی لینک All Database ، در نوار سرمه ای رنگ کلیک بر روی لینک Enterz Home در ستون راست با عنوان Hot Spots لینک Enterz Home در قسمت Hot Spot یا گزینه All Database در نوار سرمه ای رنگ بالای صفحه را کلیک نمایید تا به صفحه اصلی Enterz دست یابید.

در بالای صفحه، لوگوی سبز و آبی رنگ Enterz را مشاهده می نمایید. در این صفحه نام پایگاه‌های داده ای که موتور جست و جوی Enterz قادر است همزمان همه آن‌ها را جست و جو نمایید، به چشم می خورد. در بین آن‌هاPubMed، PubMed CentralT، Nucleotide، Structure، Booksو Protein پایگاه‌های مشهوری هستند که در این جزوه به آن‌ها خواهیم پرداخت. برای استفاده از این موتور جست و جو، کلمه یا عبارت مورد نظر خود را در کادر جلوی Search across database تایپ، و دکمه Go را کلیک نمایید. قبل از این که این موتور جست و جو را بیش تر بررسی نماییم کمی در مورد تکنیک های جست و جو صحبت خواهیم کرد. این تکنیک‌ها به شما کمک خواهند کرد تا جست و جوی های موفق تری داشته باشید. لازم به ذکر است که این عبارات و تکنیک های ارایه شده تنها در مورد این موتور جست و جو کاربرد دارد و در موتورهای جست و جوی عمومی مانند Google عمل نخواهد کرد. تکنیک های جست و جو بهترین روش مطالعه این سیستم در نظر گرفتن یک مثال است. فرض کنید که بخواهید درباره ژن DCC تحقیق نمایید. کلمه DCCمخففDeleted in Colorectal Cancer است. این ژن در سرطان روده و معده نقش اساسی دارد. ما علاقه‌مند به مقالاتی هستیم که نویسنده آنBert Vogelstein است. این محقق در این زمینه، مقالات متعددی دارد. چنین جست و جوی باید با استفاده از operator و qualifier به صورت مناسبی محدود گردد. این operatorها و qualifier با قرار گرفتن در کنار کلمات کلیدی جست و جو، عبارتی را تشکیل می دهند که به صورت زیر است[AU] "DCCAND"Vogelstein" : عبارت [AU] بعد از نام نویسنده، به Enterz می گوید که این عبارت نام یک نویسنده است و Enterz هنگام ارزیابی عبارت قرار گرفته در گیومه، تنها باید فیلد

نویسنده‌ها را در نظر بگیرید. به این عبارت‌ها که درون کروشه قرار گرفته و جست و جو را محدودتر و سریع تر می نماید qualifier می گویند.

موتور جست و جوی Entrez یک جست و جو بر روی تمام پایگاه‌های داده‌ی زیر نظر خود انجام می دهد و تعداد رکوردهای یافت شده در هر پایگاه را سمت چپ نام آن پایگاه نمایش می دهد. برای مشاهده این رکوردها بر روی عدد سمت چپ این بانک کلیک نمایید. اگر تعداد رکوردهای یافت شده در این قسمت بسیار زیاد باشد می توانید جست و جو را با اضافه کردن کلمات کلیدی بیش تری محدود تر کنید .جست و جو در یک بانک اطلاعاتی خاص علاوه بر جست و جوی همزمان همه بانک ها می توان یک بانک اطلاعاتی خاص را مورد جست و جو قرار داد. در صفحه اصلی NCBI از منوی کشویی Search، بانک اطلاعاتی مورد نظر خود را انتخاب، و سپس عبارت جست و جوی خود را در کادر جلوی for تایپ کنید و بر روی دکمه Go کلیک نمایید.

* ساختار سه بعدی پروتئین‌ها

مقدمه

یکی از کارهای جالب بشری به تصویر کشیدن مفاهیم علمی است. تصاویر، مفاهیم علمی را گویاتر و جذاب‌تر می‌کنند. زیست‌شناسان و محققین علم ژنتیک نیز از این ابزار به خوبی بهره برده و از آن در کارهای تحقیقاتی و انتقال مفاهیم به دانش‌اندوزان استفاده کرده‌اند. دستاوردهای زیادی در همکاری مهندسین کامپیوتر با دانشمندان زیست‌شناس حاصل شده است که این تلاشها در خور تقدیر است. تجسم پیچ و خم‌های DNA و ماکرومولکولهای مرموزی چون پروتئین‌ها و درک ویژگی‌های وابسته به این ساختارها، برای انسان بسیار مشکل است. در این زمینه تصاویر سه بعدی بسیار راهگشا و پر فایده هستند. تصاویر و انیمیشن‌های زیادی به همراه نرم‌افزارهای با ارزشی که این تصاویر را نمایش دهند به وجود آمده‌اند که در این فصل به نمونه‌ای از آنها توجه خواهیم کرد. پژوهشگران به روش‌های مختلفی از جمله تابش اشعه ایکس ساختار سه بعدی پروتئین‌ها را مورد بررسی قرار می‌دهند. نتایج حاصل از این پرتونگاری‌ها، اطلاعاتی است که در به تصویر کشیدن این ساختارها مورد نیاز است. یک گروه از پایگاه داده‌ها به نام Structure توسط NCBIتدارک دیده شده است که اطلاعات مربوط به ساختار سه بعدی پروتئین‌ها را در خود نگه می‌دارد .این گروه شامل مواردزیراست:

MMDB

پایگاه داده‌ای شامل ساختار سه بعدی ماکرومولکول‌ها (The Molecular Modeling DataBase) به همراه ابزارهایی برای نمایش و مقایسه این ساختارها است. ساختار سه بعدی این پروتئین‌ها به روش‌های

تجربی و از پروتئین‌های موجود در پایگاه داده پروتئین‌ها (PDB) حاصل شده است. نرم‌افـزاری بـه نـام Cn۳d توسط NCBI تدارک دیده شده است که به وسیله آن می‌توان این ساختارهای سه بعدی را مشاهده کرد. در این جزوه طریقه استفاده از این نرم‌افزار را مفصلا برا ی شما شرح خواهیم داد. شما می‌توانید با توالی یک پـروتئین شروع نمایید و با استفاده از BLAST تمام پروتئین‌هایی که با این نسبتی دارند مشخص نمایید و سپس سـاختار سه بعدی هر یک را که تاکنون شناخته شده‌اند مشاهده نمایید.

❋ نرم افزار Cn۳D

اکنون وقت آن رسیده است تا ساختار سه بعـدی پـروتئین‌هـا را توسـط نـرم‌افـزار Cn۳D بـه نمـایش درآوریم. همچنان که در ابتدای جزوه گفته شد، ساختار فضایی بعضی از پروتئین‌ها به روش‌های تجربی تعیـین شده است. جهت نمایش چنین ساختارهای فضایی زیبا نیاز بـه نـرم‌افزارهـای ویـژه‌ای اسـت. سـازمان NCBI نرم‌افزار مناسب و قدرتمندی را ایجاد کرده است که در بخش طریقه دانلود کردن، نصب کردن و اسـتفاده از آن را برای شما شرح می‌دهیم. مطمئنا تماشای ساختار سه بعدی پروتئین‌هـا توجـه شـما را جلـب خواهـد کـرد و امیدواریم که از آن در کارهای تحقیقاتی خود استفاده کنید. قابلیت‌های این نرم‌افزار به اختصار به صـورت زیـر است:

نمایش مدل‌های مختلف پروتئین مانند مدل فضا پر کن، توپ و میله، میلـه‌ای و چنـدین مـدل زیبـای دیگر. ۲- نمایش پروتئین بر اساس بار الکتریکی اتم‌ها . ۳- نمایش دامین‌های پروتئین. ٤- نمایش ساختار دوم پروتئین. ٥- نمایش توالی هر زنجیره از پروتئین و جایگاه هر اسید آمینه بر روی پروتئین. ٦- نمایش پـروتئین بر اساس دما نسبی اتم‌ها.

دانلودنرم‌افزارCn۳D

لینک این نرم‌افزار در بخش‌های مختلف سایتNCBI قرار داده شده است. ساده‌تـرین مسـیر را بـرا ی دانلود کردن آن به شما ارائه می‌دهیم.

صفحه اصلی سایت NCBI بـه آدرسwww.ncbi.nlm.nih.gov را بـاز نماییـد. از منـویی کـه لیسـت بانک‌ها در آن قرار دارد(Search) ، پایگاه داده Structure را انتخاب نمایید و دکمه Go را کلیک کنید. این منو و انتخاب پایگاه داده Structure را به شما نشان می‌دهد. بعد از کلیک روی دکمه Go صفحه اصـلی پایگـاه داده Stucture باز می‌شود. در میان صفحه لینک آبی‌رنگ نرم‌افزارCn۳D را می‌بینید. روی آن کلیک نمایید تا صـفحه بعد باز شود. لینک دانلود نرم‌افزار در بالای صفحه دیده می‌شود. این نرم‌افزار برای سیستم عامـل‌هـای مختلـف مانند Windows وUnix طراحی شده است. روی کلمهDownload کلیک نمایید تا صفحه بعد باز شود. سیستم عامل را انتخاب و بر روی آن کلیک نمایید.

در نهایت لینک ftp:/ftp.ncbi.nih.gov/cn۳d/ cn۳D-٤.١.msi را برای دریافت فایل نصبی این نرم‌افزار کلیک نمایید. توجه کنید که در این صفحه نیازمندی‌های نرم‌افزار Cn۳D نیز لیست شده است. این نرم‌افزار بر روی سیستم عامل‌های ویندوز از تمام نسخه‌ها نصب می‌شود. توصیه می‌شود که کارت گرافیک شما ۳۲ بیتی باشد. این نرم‌افزار نیازمندی خاصی ندارد. بعد از کلیک بر روی این لینک، پنجره ذخیره کردن نرم‌افزار باز می‌شود. بر روی دکمه Save کلیک نمایید و فایل مربوطه را در مسیر دلخواه خود روی کامپیوتر ذخیره نمایید. نام فایل این نرم‌افزار Cn۳D-٤.msi است . بعد از ذخیره این فایل روی آن دابل کلیک کنید تا اجرا شود. مراحل نصب را تا آخر دنبال کنید. پس از اتمام مراحل نصب، از منوی start و از مسیر زیر می‌توانید اقدام به اجرای آن کنید: start /program files/NCBI

معرفی منوهای نرم‌افزارCn۳D

این نرم‌افزار قابلیت‌های زیادی دارد که شما را با تعدادی از آن‌ها آشنا می‌کنیم.این نرم‌افزار این قابلیت جالب را دارد می‌توانید آن را در زاویه‌های مختلف بچرخانید. برای این کار موس را در قسمتی از تصویر قرار داده، دکمه چپ موس را پایین نگه داشته و آن را حرکت دهید. مولکول شما به زیبایی و در یک فضای سه بعدی می‌چرخد.

منوی view

اگر بخواهید بخش‌هایی از مولکول را دقیق‌تر مورد بررسی قرار دهید، بهتر است که ابتدا آن قسمت را با چرخاندن در جلو تصویر قرار دهید را دقیق‌تر مورد بررسی قرار دهید، بهتر است که ابتدا آن قسمت را با چرخاندن در جلوی تصویر قرار دهید و سپس تصویر را بزرگ نمایید. برای بزرگ کردن تصویر منوی View را باز نمایید و گزینه Zoom in را انتخاب کنید. برای کوچک‌تر کردن تصویر نیز از همین منو گزینه zoom out را کلیک کنید .اگر می‌خواهید به اندازه و زاویه اولیه تصویر برگردید یکی از گزینه‌های Restore یا Reset را انتخاب کنید. اگر شکل فضایی مولکول را بهتر درک کنید، نرم‌افزار این توانایی را دارد که مولکول را به صورت افقی و پیوسته بچرخاند. برای اینکار در پایین منویView گزینه Animation را انتخاب کنید .این گزینه شامل چندین زیر منو است. گزینه spin را انتخاب کنید تا تصویر شروع به چرخش کند. برای توقف چرخش گزینه Stopرا انتخاب کنید. به هنگام چرخیدن تصویر می‌توانید با موس خود مولکول را گرفته و آن را به سمت بالا یا پایین حرکت دهید تا زاویه چرخش مولکول تغییر نماید. می‌توانید در حین چرخش تصویر، آن را بزرگ کنید تا جزییات پروتئین بیش‌تر درک نمایید.

منویHide / Show

قبل از توضیحاتی در مورد این منو لازم است که پنجره توالی پروتئین را معرفی نماییم. در ابتدا که نرم‌افزار Cn۳D را باز می‌نمایید دارای پنجره‌ای در پایین صفحه است که زنجیرها و توالی اسید آمینه هر زنجیره را نمایش می‌دهد. در سمت چپ تصویر نام چهار زنجیره هموگلوبین قرار گرفته است .مثلا زنجیره A با نام ۲ H۳۵-Aاولین زنجیره است. توالی این زنجیره در جلو آن با رنگ‌های سبز فیروزه‌ای قرار دارد. بخش‌های سبز هلیکس‌ها را تشکیل می‌دهند. همچنان که گفته شد این مولکول دارای چهار زنجیره است. ممکن است نمایش هر چهار زنجیره با هم شلوغ به نظر برسد. می‌توانید تنها یکی از زنجیره‌های هموگلوبین را انتخاب کنید و آن را مورد بررسی قرار دهید. برای این کار منوی Hide / Show را باز نمایید و گزینهPick Structure را انتخاب کنید. بعد از انتخاب این گزینه پنجره‌ای باز می‌شود که تمام زنجیره‌های هموگلوبین را لیست می‌کند . در ابتدا تمام زنجیرهای انتخاب هستند و به همین دلیل همه آنها نمایش داده می‌شوند. آبی بودن نام زنجیره‌ها مبین انتخاب بودن آنهاست. اگر بر روی نام یکی از زنجیره‌ها کلیک نمایید، به رنگ سفید در آمده و از انتخاب خارج می‌شود. برای انتخاب مجدد، کافی است روی نام زنجیره دوباره کلیک نمایید .بعد از هر تغییری در این پنجره دکمه Done را کلیک کنید تا زنجیره انتخابی شما نمایش داده شود. بعد از هر تغییر در این پنجره می‌توانید از منویShow/Hide گزینهShow Everything را انتخاب کنید تا تمام زنجیره‌ها و اجزای پروتئین دوباره نمایش داده شود. نمایش پروتئین را می‌توان به یک هلیکس یا تعداد دلخواهی اسید آمینه محدود کرد. برا ی این کار در پنجره توالی اسید آمینه که در زیر پنجره اصلی نرم‌افزار قرار دارد و در بالا در مورد آن توضیح داده شد، با درگ کردن بر روی تعدادی از اسید آمینه‌ها آنها را انتخاب نمایید. هرگاه اسید آمینه‌ای را انتخاب می‌کنید، نام اسید آمینه در پنجره توالی‌ها و نیز جایگاه آن اسید آمینه در پنجره نمایش سه بعدی پروتئین به رنگ زرد در می‌آید تا انتخاب شما قابل تشخیص باشد . اگر تصمیم دارید چندین تکه متوالی را انتخاب نمایید که از هم فاصله دارند باید بعد از انتخاب هر تکه کلید shift را پایین نگه دارید. دو هلیکس از زنجیره A که با تعدادی اسید آمینه از هم جدا شده‌اند، انتخاب شده است. هر دو هلیکس در تصویر سه بعدی به رنگ زرد مشخص هستند. دقت کنید که جهت جلوگیری از شلوغی تنها زنجیره A را نمایش داده‌ایم. اگر بخواهید تنها توالی‌های انتخاب شده را نمایش داده شود، از منوی Show/Hide گزینه Show selected Residue را انتخاب نمایید. اگر بخواهید دامین خاصی از زنجیره‌ها را نمایش دهید کافی است در پنجره توالی پروتئین، یکی از اسید آمینه‌های آن دامین را انتخاب نمایید و از منوی Show/Hide گزینهDomain show selected را برگزینید.

منویStyle

در این منو گزینه‌هایی وجود دارد که به شما امکان می‌دهد اجزای مختلف پروتئین مانند دامین‌ها مولکول‌ها لیگاندها و... و همچنین ویژگی‌های پروتئین مانند دمای نسبی مولکلوها آبگیزی بارالکتریکی و...را

جداگانه و بر اساس رنگ‌های انتخابی خودتان نمایش دهید. اولین گزینه این منو Edit Global style است که از طریق آن می‌توانید رنگ‌های بخش‌های مختلف پروتئین را تغییر دهید. بعد از اجرای این دستور پنجره‌ای باز می‌شود که چندین گزینه از آن را بررسی می‌نماییم. در پایین و سمت راست این پنجره کلیک نمایید، پنجره دیگری باز می‌شود که می‌توانید رنگ دلخواه خود را انتخاب کنید.

منوی Window

از این منو می توانید برای نمایش پنجره‌های جانبی نرم افزار استفاده کنید. گزینه Show Sequence Viewerباعث نمایش یا عدم نمایش پنجره نمایش توالی‌های اسید آمینه پروتئین می شود. هرگاه به پنجره توالی نیاز ندارید می توانید با اجرای این دستور آن را نمایش ندهید. گزینه Show Message باعث نمایش یا عدم نمایش پنجره ای می شود که در آن دستورات اجرا شده توسط شما را ثبت می نماید.

BLASTX

در این نوع بلاست توالی مورد تقاضا نوکلئوتیدی است که در ٦ قالب خواندنی به پروتئین ترجمه میشود یک توالی DNA به عنوان مثال:

ATGAGCCTAAGATGACTG را در نظر بگیرید اگر این ابتدای یک ژن باشد ، از روی یکی از این رشته ها رونویسی انجام می شود یکی رشته + و یکی رشته – پس تا اینجا فریم خواندن خواهیم داشت. حالا رشته مثبت را در نظر بگیرید: زمانی که می خواهد رونویسی انجام گیرد می تواند از ابتدای رشته نوکلئوتید A شروع شود مسیر ۱، می تواند از نوکلئوتید دوم T شروع شود مسیر ۲، و یا می تواند از نوکلئوتید سوم G آغاز شود مسیر ۳. چون هر سه نوکلئوتید یک اسید آمینه را کد می کند در نتیجه از هر کدام از ۳ مسیر بالا را که انتخاب کند در آخر ۳ پپتید با اندازه های متفاوت تولید می کند، زیرا هر کدام به یک کدون پایان در نقاط متفاوت می رسند. پس تا اینجا دو رشته و هر رشته دارای سه فرم خواندن هستند که در مجموع ٦ فریم خواندن خواهیم داشت. حالا زمانی که روی یک توالی DNA ئ BLASTX انجام می دهیم در واقع توالی DNAرا به توالی اسید آمینه تبدیل می کند، توالی موجود در بانک اطلاعاتی را نیز به توالی اسید تبدیل می کند سپس BLAST را انجام می دهد. در نتیجه هر یک از رشته ها را که می خواهد به اسید آمینه ترجمه کند طبق توضیحات بالا ار سه مسیر می تواند آن را انجام دهد که در مجموع ٦ فریم خواندن به ما می دهد. زمانی نیز که نتایج BLAST را می بینید در بالای هر کدام مشخص کرده که از کدام مسیر رفته است.

منابع

۱- مروری بر ژنتیک مولکولی (زیست مولکولی)، فصل دهم.

۲- ابزارهای بیوانفورماتیک.

۳- A practical guide to bioinformatics by:mohammad ali malboobi & tahmineh lohrasebi