محاسبات نرم

(هوش محاسباتی)

مدرس

امیر جلیلی ایرانی

amirjalili.ir

amirjaliliirani@gmail.com

تحصیلات

لیسانس کامپیوتر – نرم افزار

فوق لیسانس کامپیوتر – هوش مصنوعی

از دانشگاه علوم و تحقیقات تهران

هوش مصنوعی
Artificial Intelligence (A. I.)

- هوش مصنوعی:

الهام از موجودات زنده به ویژه انسان برای ایجاد قابلیت های استدلال، استنتاج، یادگیری و رفتار هوشمند.

- گرایش هوش مصنوعی:

عبارت است از هوشمند سازی کامپیوترها و سیستم های مبتنی بر کامپیوتر.

مرجع درسی

الف) هوش مصنوعی(رهیافتی نوین)

Prentice hall

تالیف: استوارت راسل – پیتر نورویگ ترجمه: ۱. رامین رهنمون ۲. راحتی و بهداد و تیموری

ناقوس دانشگاه امام رضا

ب) اصول و مبانی سیستم های خبره

انشارات دانشگاه علم و صنعت

تالیف: مهدی غضنفری – زهره کاظمی

ج) سیستم های فازی و کنترل فازی

انتشارات دانشگاه صنعتی خواجه نصیر

تالیف: لی وانگ ترجمه: تشنه لب – صفارپور – افیونی

د) اساس شبکه های عصبی مصنوعی

تالیف: لوران فاست ترجمه: ....

مرجع کنکوری

هوش مصنوعی

۱- انتشارات پوران پژوهش

تالیف: مهدیه شادی

۲- انتشارات گسترش علوم پایه

تالیف: رضا رافع – حمید رضا مقسمی

amirjalili.ir

سر فصلها

- هوش مصنوعی چیست؟
- هوش محاسباتی چیست؟
- سیستم های هوشمند
- شبکه های عصبی مصنوعی
- الگوریتم های تکاملی
- سیستم های فازی
- سیستم های ترکیبی

پیش نیاز ها

- ساختمان داده ها
- طراحی الگوریتم ها
- هوش مصنوعی

دلیل اهمیت درس

- **کاربردی و باعث تقویت برنامه نویسی هوشمند.**
- **پایه و پیش نیاز بسیاری از دروس تخصصی هوش.**
- **مطرح در کنکور کارشناسی ارشد گرایش های مختلف.**

سیستم ارزشیابی

☐ کوئیز	۱
☐ تمرینات	۱
☐ میان ترم	۴
☐ پایان ترم	۱۴
☐ حضور مرتب و منظم	مازاد
☐ تحقیق و پروژه	اختیاری

فصل اول

هوش مصنوعی چیست

مقدمه

- انسان موجود کاملی نیست. از اینرو نیازهای فراوانی در زندگی وی ظاهر می‌شوند. این نیازها در ادوار مختلف تمدن بشری، تغییرات فراوانی داشته‌اند.
- اما چیزی که در میان تمام نسل‌های بشری مشترک است، تلاش برای رفع نیازها و کمبودهای ذاتی است که با تکیه بر قدرت تفکر و نوآوری، میتواند با ابداع و اختراع، بخش قابل توجهی از محدودیت‌ها و نیازهایش را جبران نماید.

amirjalili.ir

مقدمه

- انسان از نظر توان فیزیکی، موجودی محدود است. در میان جاندارانی که در خشکی، دریا و هوا حرکت می‌کنند؛ انسان جایگاه قابل توجهی ندارد.
- با درک چنین نیازهایی، در طول قرون و اعصار مختلف، اختراعاتی چون: هواپیما، کشتی، چاقو، جرثقیل، لباس و تجهیزات غواصی و طناب در جهت رفع محدودیت‌های فیزیکی انسان‌ها ایجاد شده‌اند.

مقدمه

- پس از مدتی که تمدن نوپای بشری کار خود را آغاز نمود، نوع دیگری از نیازها برای انسان مطرح شدند. نیازهایی که در مقایسه با نیازهای فیزیکی، قدری لوکس‌تر به نظر می‌رسند.
- انسان، علاقه‌مند به ثبت وقایع و دستاوردهای خود شد. آثار باستانی نشان از علاقه‌مندی بشر به حفظ و انتقال اطلاعات دارد.
- نیاز به ذخیره‌سازی و تبادل اطلاعات، ابداعاتی از قبیل دیسک‌های نوری، دوربین عکاسی، خط، تلفن، کبوتر نامه‌بر و اینترنت را بوجود آورده است.

amirjalili.ir

مقدمه

- دسته‌ی دیگری از محدودیت‌هایی که انسان رفته رفته بیشتر با آن‌ها روبرو شد شامل خستگی، محدودیت سرعت کار، و عدم مصونیت از خطا است.
- انسان نمی‌تواند یک طرح را صدها بار بدون هیچ تفاوت اجرا نماید. قطعا با طولانی‌تر شدن زمان کار، کیفیت کار نیز خدشه‌دار خواهد شد.
- این نوع از نیازها و محدودیت‌ها منجر به ایجاد اختراعاتی چون ساعت، چاپگر، و کامپیوتر شدند.

مقدمه

- بشر امروز، علاوه بر نیازهای فیزیکی، ثبت و انتقال اطلاعات و پردازش دقیق نوع دیگری از نیاز را تجربه می‌کند. نیازی که تا ۵۰ سال پیش، هیچ گاه به صورت جدی برای انسان مطرح نشده بود.
- امروزه انسان در پی ایجاد ابزارهایی است که به جای او فکر کند، یاد بگیرد و تصمیم گیری کند. ابزارهایی که از قدرت تحلیل و هوش انسان تقلید میکنند. به این ترتیب هوش مصنوعی به عنوان یکی از نیازهای عصر جدید مورد توجه قرار گرفت.

amirjalili.ir

مقدمه

- فرض کنید که به جای استفاده از یک هوش انسانی، از سیستمی استفاده شود که نیازمند استفاده از چندین تخصص در رشته‌های مختلف علمی و مهندسی است.
- سیستم‌هایی همچون خلبان خودکار در هواپیما، کنترل کننده‌های ترافیک شهری، ادوات مراقبت پزشکی هوشمند، ابزارهای تبدیل صوت به متن و روبات‌های هوشمند، از جمله مظاهر استفاده از هوش مصنوعی در زندگی روزمره هستند.

تعریف هوش مصنوعی

الهام از موجودات زنده به ویژه انسان برای ایجاد قابلیت های استدلال، استنتاج، یادگیری و رفتار هوشمند.

عاقلانه فکر کردن	مانند انسان فکر کردن
عاقلانه عمل کردن	مانند انسان عمل کردن

هوش مصنوعی

تعاریف AI به چهار قسمت تقسیم شده‌اند:

- پردازش فکري و استدلالي(reasoning)
- پردازش رفتاري
- هوشمندي ايده‌آل (منطقي بودن)
- كارايي انساني

هوش مصنوعی

amirjalili.ir

تست تورینگ

- در سال ۱۹۵۰آلن تورینگ آزمونی به شرح زیر برای تشخیص هوشمند بودن ماشین بیان کرد:

آزمونی از کامپیوتر به عمل آید که در آن آزمون گیرنده نتواند دریابد که در آن طرف انسان قرار دارد یا ماشین.

- برای این منظور ماشین باید قابلیت های خاصی داشته باشد.

هوش محاسباتی

- هوش محاسباتی (Computational Intelligence) یکی از زیر بخش‌های بسیار مهم و کاربردی هوش مصنوعی است، که در آن از ابزارهای مختلفی برای تحقق ایده‌ی هوش مصنوعی استفاده می‌شود.

- ابزارهای مورد استفاده در هوش محاسباتی، غالبا ابزارهایی ریاضیاتی هستند که به نوعی از طبیعت و دنیای اطراف الهام گرفته شده‌اند. مهم‌ترین ابزارها و الگوهایی که در هوش محاسباتی مطرح می‌شوند، شامل موارد زیر هستند:

پردازش تکاملی

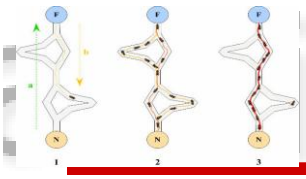
- محاسبات یا پردازش تکاملی شامل مجموعه‌ای از روش‌ها است که به نام الگوریتم‌های تکاملی معروف هستند.

- مشهورترین این الگوریتم‌ها، الگوریتم ژنتیک است که از نظریه تکامل و علم ژنتیک الهام گرفته شده است. در این الگوریتم، فرآیند تکامل، که طی میلیون‌ها سال در طبیعت اتفاق افتاده است، شبیه‌سازی می‌شود.
- اصلی‌ترین مورد کاربرد الگوریتم‌های تکاملی، حل مسائل بهینه‌سازی و برنامه ریزی ریاضی است.

مزایا و معایب

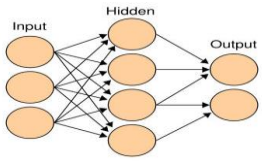
- مزایا:
1. کارایی محاسباتی
2. بهینه سازی عمومی

- معایب:
1. سرعت پایین محاسبات
2. دشوار بودن کد نویسی



هوش گروهی

- روش‌هایی که در این دسته قرار می‌گیرند، الگوی دیگری را برای حل مسائل بهینه‌سازی پیشنهاد می‌کنند.
- در این روش‌ها، تعداد قابل توجهی از عامل‌های بسیار ساده و کم هوش، برای تشکیل نوعی هوش ازدحامی یا هوش جمعی با یکدیگر همکاری یا رقابت می‌کنند.
- به عنوان مثال، الگوریتم بهینه‌سازی مورچگان، که از رفتار جمعی مورچه‌ها الهام گرفته شده است، یکی از الگوریتم‌های هوش ازدحامی است.



شبکه های عصبی مصنوعی

- تقریبا همه دانشمندان یقین دارند که مغز انسان پیچیده ترین ساختار موجود و شناخته شده در کل هستی است. ریاضی‌دان‌ها و مهندسین هوش مصنوعی، با الهام از یافته‌های عصب‌شناسان، شبکه‌های عصبی مصنوعی را معرفی کردند که استفاده‌های فراوانی در مدل‌سازی و طبقه‌بندی اطلاعات دارد.
- شاید بتوان شبکه‌های عصبی را مهم‌ترین ابزار در زمینه یادگیری ماشینی به حساب آورد.

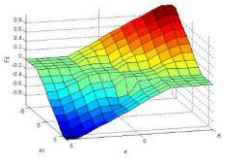
مزایا و معایب

- مزایا:

1. یادگیری
2. تطبیق پذیری
3. تحمل پذیری در برابر خطا
4. توانایی تعمیم یا عمومیت بخشی
5. توانایی تقریب
6. نمودار تناسب با واقعیت

- معایب:

1. عدم توانایی تحلیل و تفسیر لایه های میانی سیستم



سیستم های فازی

- نظریه‌ی مجموعه‌های فازی و محاسبات فازی از ابداعات پرفسور لطفی عسگرزاده، استاد ایرانی-آذربایجانی دانشگاه برکلی آمریکا است.
- در محاسبات فازی، به جای استفاده از اعداد دقیق برای توصیف یک مفهوم، از کلماتی مانند کم یا زیاد استفاده می‌شود.
- سیستم‌هایی که بجای نظریه کلاسیک مجموعه‌ها و محاسبات کلاسیک ریاضی، از نظریه مجموعه‌های فازی و محاسبات فازی بهره میبرند، بنام سیستم‌های فازی شناخته می‌شوند.
- امروزه از سیستم‌های فازی در طراحی سیستم‌های مختلف، از جمله لوازم خانگی هوشمند، استفاده‌های فراوانی می‌شود.

مزایا و معایب

- مزایا:

1. قابلیت تفسیر
2. شفافیت
3. تدریج
4. تحمل عدم دقت

- معایب:

1. اکتساب دانش
2. یادگیری

هوش محاسباتی

- در کنار موارد یاد شده، ابزارهای ریاضی دیگری نیز به کار گرفته می‌شوند تا عملکرد کلی سیستم‌های مبتنی بر هوش محاسباتی بهبود یابند.

- هدف اصلی محققین حوزه‌های هوش مصنوعی و هوش محاسباتی، ایجاد ابزارهایی است که ما را به ایجاد هوش مصنوعی هم تزار با هوش انسانی نزدیک‌تر نماید.

amirjalili.ir

محاسبات نرم

- محاسبات نرم یا soft computing روشی است که نیازی به استفاده دقیق و کامل از اطلاعات ندارد.

- طی چند دهه اخیر برای حل مسائل مختلف و روزمره از آنها استفاده میشود.
- قبلاً از روشهای محاسبات دقیق یا hard computing برای حل مسائل استفاده میشد.

- مدل کامل از S.C. همان مغز انسان است.

ویژگیهای روشهای محاسبات نرم

- محسبات نرم شامل روشهایی است که میتوانند خود را با موارد زیر تطبیق دهند:

1. عدم دقت یا imprecision
2. عدم قطعیت یا uncertainty
3. حقیقت جزئی یا partial truth

amirjalili.ir

فصل دوم

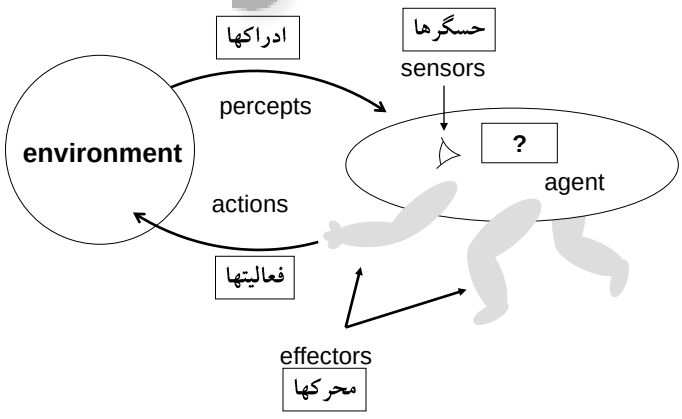
عاملهای هوشمند

حل مساله

عامل یا agent

- هر چيزي كه قادر به درک محيط پيرامون از طريق حس‌گرها (sensor) و اثرگذاري بر روي محيط از طريق اثركننده‌ها (effector) باشد، عامل نام دارد.
- عامل نرم‌افزاري، رشته‌هاي بيتي را به عنوان درک محيط و عمل كدگذاري مي‌كند.

نمایی از یک عامل



ساختار عامل

برنامه + معماري = عامل

- وظيفه هوش مصنوعي طراحي برنامه عامل است.
- برنامه عامل، تابع عامل را محقق ميسازد.
- تابع عامل، اقدامات عامل را در قبال ادراکات مشخص ميکند.
- رفتار عامل وابسته به دنباله ادراکي رويت شده تا حال است.

عاملهای یادگیرنده

- نظر تورینگ: ساخت ماشینهای هوشمند و سپس آموزش آنها.
- عامل های معمولی فاقد یادگیری هستند و دانش آنها در زمان طراحی تهیه میشود. آنها خودمختار نیستند و از تجربیات خود جهت تصمیم گیری و کسب دانش جدید استفاده نمیکنند.
- یادگیری به عامل اجازه عمل در محیطی ناشناخته میدهد تا ماهرتر گردد.
- عامل های یادگیرنده با استفاده از تجربیات خود به دانششان اضافه کرده و در صورت لزوم آنرا اصلاح می نمایند.

عامل حل مسئله

- یک نوع عامل هدفگرا، عامل حل مسئله نامیده می‌شود.
- این عاملها با یافتن ترتیب عملیات، تصمیم به انجام آنها برای رسیدن به حالت‌هاي مطلوب دارند.

سه مرحله اساسي براى حل مسائل

- تدوین: ۱- هدف
  ۲- مساله
- جستجو
- اجرا

عامل بصورت تدوین، جستجو، اجرا طراحی میشود.

تدوین هدف

- اولین گام در حل مساله تدوین هدف بر اساس حالت فعلی و مقیاس کارایی عامل میباشد.
- اهداف با محدود کردن مقاصدی که عامل برای رسیدن به آنها تلاش میکند به سازماندهی رفتار عامل کمک میکند.
- وظیفه عامل فهمیدن اینست که کدام دنباله از اقدامات آنرا به یک حالت هدف میرساند.

تدوین مساله

- فرایند تصمیم گیری در مورد اقدامات و حالات بر اساس یک هدف خاص است.
- چه فعالیتها و وضعیت هایی برای رسیدن به هدف موجود است.

مرحله جستجو

- فرایند بررسی توالی های مختلف ممکن از اقدامات که به حالاتی با مقادیر معلوم منجر میشود و انتخاب بهترین آنها توسط عامل، جستجو نام دارد.
- رفتن از یک حالت ممکن به حالت ممکن دیگر برای رسیدن به حل مساله.(میتواند درخت یا گراف باشد)
- الگوريتم جستجو مسئلهاي را به عنوان ورودي دريافت نموده و راهحلي را به صورت دنباله عملیات بر ميگرداند.

مرحله اجرا

- بمحض اینکه جستجو راه حلی پیدا کرد اقدامات پیشنهاد شده را اجرا میکند.
- پس از تدوین یک هدف و یک مساله برای حل، عامل یک روال جستجو را فراخوانی میکند. راه حل توالی اقدامات بعدی را برای عامل مشخص میکند و عامل اقدام به انجام و اجرای راه حل میکند.
- پس از اجرای راه حل عامل هدف جدیدی تدوین میکند.

مسائل و راه حلهای خوش تعریف

- مسئله: در واقع مجموعه‌اي از اطلاعات است که عامل از آنها براي تصميم‌گيري در مورد اينکه چه کاري انجام دهد، استفاده مي‌کند.

- یک مساله بصورت رسمی با چهار مولفه تعریف میشود:

1. حالت ابتدایی
2. اقدامات یا عملگرها یا حرکت ها
3. آزمون هدف
4. تابع هزینه مسیر

حالت ابتدایی

- حالت نشان دهنده یک وضعیت از مساله است.
  وضعیت منحصر به فرد از محیط مساله.

- حالت شروع حالتی است که عامل از آن شروع بکار میکند.
  عامل خودش از بودن در آن آگاه است.

- حالت ابتدایی ممکن است بینهایت یا بیش از یکی باشد.

اقدامات یا عملگرها

- اقدامات، فعالیتها و عملیات ممکن که در دسترس عامل هستند.
- معمولاً از تابع پسین یا جانشین استفاده میشود.
- هر چیزی که باعث تبدیل حالات به یکدیگر شود.
- عملگر روی حالتی اعمال شده و آنرا به حالت دیگری تبدیل میکند.
- اگر عملگرها بینهایت باشند جستجو امکانپذیر نیست. پس حتماً باید عملگرها محدود باشند.

تابع جانشین + حالت اولیه ← فضای حالت

فضای حالت

- مجموعه تمامی حالتهایی که از حالت اولیه میتوان به آنها رسید.
  مجموعه تمامی حالات ممکن(امکانپذیر و امکان ناپذیر) مساله.
- فضای حالت گرافی را تشکیل میدهد که گره های آن حالتها و کمانها اقدامات میباشند.
- یک مسیر در فضای حالت یک توالی از حالتهایی است که توسط دنباله اقدامات بهم متصل اند.
- فضای حالت با فضای جستجو متفاوت است.

آزمون هدف

- تعیین میکند که آیا یک حالت خاص، حالت هدف است یا خیر.

- هدف صریح: در مثال رومانی، رسیدن به بخارست. در این شکل حالت نهایی را داریم.

- هدف انتزاعی: در شطرنج، رسیدن به حالت کیش و مات. در این شکل به جای حالت نهایی الگویی برای شناسایی حالت نهایی داریم.

تابع هزینه مسیر

- برای هر مسیر یک هزینه عددی تخصیص میدهد.

- عامل حل مساله، تابع هزینه ای را انتخاب میکند که مقیاس کارایی خودش را منعکس کند.

- راه حل یک مسئله مسیری از حالت ابتدایی تا حالت هدف است.

- یک راه حل بهینه (معمولا) کمترین هزینه مسیر را در بین تمامی راه حلها دارد.

مسائل نمونه

- رویکرد حل مساله در مجموعه وسیعی از محیط های کار استفاده میشود.

- مسائل اسباب بازی:

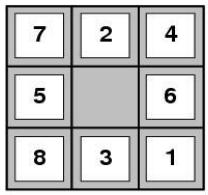
هدف، نمایش یا تمرین روشهای مختلف حل مساله است.
میتوان از آنها برای مقایسه کارایی الگوریتم ها استفاده کرد.

- مسائل دنیای واقعی:

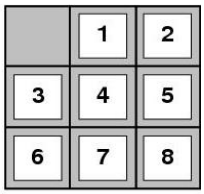
مسائلی که عموماً راه حلهای آن برای مردم واقعاً اهمیت دارد.

مساله اسباب بازی: پازل

- حالتها:
- حالت اولیه:
- تابع جانشین:
- آزمون هدف:
- هزینه مسیر:



Start State

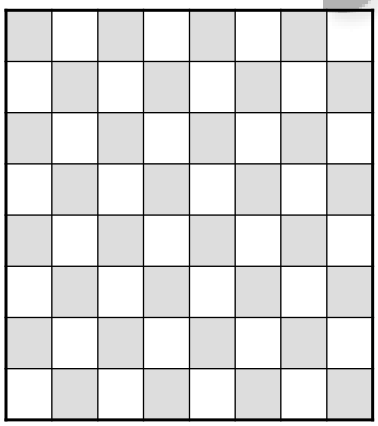


Goal State

مساله اسباب بازی: پازل

- پازل ۸ تایی 9!/2=181440 حالت دارد.
- پازل ۱۵ تایی حدود 1.3 تریلیون حالت دارد.
  حل آن با بهترین الگوریتم ها در حدود چند میلی ثانیه.
- پازل ۲۴ تایی حدود ۱۰۲۵تریلیون حالت دارد.
  حل آن با بهترین الگوریتم ها هنوز هم مشکل است.

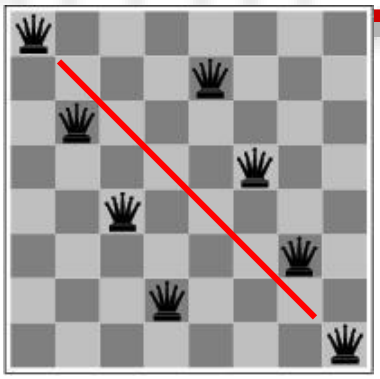
مساله اسباب بازی: مساله n وزیر



تدوین اول: افزایشی

- حالتها:
- حالت اولیه:
- تابع جانشین:
- آزمون هدف:
- هزینه مسیر:؟

مساله اسباب بازی: مساله n وزیر



تدوین دوم: حالت کامل

- حالتها:
- حالت اولیه:
- تابع جانشین:
- آزمون هدف:
- هزینه مسیر:؟

amirjalili.ir

مساله اسباب بازی: مساله n وزیر

- تدوین اول:

در این روش باید 64*....*57=3*10^14 دنباله ممکن بررسی میشود.(برای n=100، 10^400 حالت)

- تدوین دوم:

این روش فضای حالت را از 3*10^14 به 2057 کاهش میدهد.(برای n=100، 10^52 حالت)

مساله دنیای واقعی

- مسیریابی:

الگوريتم‌هاي مسير يابي كاربردهاي زيادي دارند، مانند مسيريابي در شبكه‌هاي كامپيوتري، سيستم‌هاي خودكار مسافرتي و سيستم‌هاي برنامه‌نويسي مسافرتي هوايي.

- مساله فروشنده دوره گرد:

مسئله فروشنده دوره گرد مسئله مشهوري است كه در آن هر شهر حداقل يكبار بايد ملاقات شود. هدف يافتن كوتاهترين مسير است.

اندازه گیری کارایی حل مساله

- کامل بودن: آیا الگوریتم تضمین میکند که در صورت وجود راه حل، آن را بیابد؟
- بهينگي: آیا این راه حل، بهینه ترین جواب را ارائه میکند.
- پيچيدگي زمانی: چقدر طول میکشد تا راه حل را پیدا کند؟
  تعداد گره های تولید شده در حین جستجو
- پيچيدگي فضا: برای جستجو چقدر حافظه نیاز دارد؟
  حداکثر تعداد گره های ذخیره شده در حافظه

فصل سوم

حل مساله از طریق جستجو

راهبردهای جستجوی ناآگاهانه

- ناآگاهی این است که الگوریتم هیچ اطلاعات اضافی غیر از اطلاعاتی که در مسئله آمده درباره حالت دراختیار ندارد.
- این راهبردها فقط میتواند جانشینها یا پسینهایي را تولید و حالت هدف را از غیر هدف تشخیص دهند.
- راهبردهایي که تشخیص میدهد یک حالت غیر هدف نسبت به گره غیر هدف دیگر، امید بخش تر است، جست و جوی آگاهانه یا جست و جوی اکتشافي(هیورستیک)نامیده میشود.

راهبردهای جستجوی ناآگاهانه

- جستجوي اول سطح
- جستجوي با هزينه يكنواخت
- جستجوي اول عمق
- جستجوي عمق محدود
- جستجوي عميق‌شونده تكراري
- جستجوي دوطرفه

* تمام راهبردهای جستجو بر اساس ترتیب گسترش گره ها، از یکدیگر متمایز میشوند.

جستجوي هزينه يكنواخت(UCS)

- این روش گره nای که کمترین هزینه مسیر دارد را گسترش میدهد.
- این روش جستجو به تعداد مراحل یک مسیر اهمیت نمیدهد بلکه فقط هزینه کل آنها را در نظر میگیرد.
- این روش یک جستجوی با هزینه است.

جستجوي هزينه يكنواخت: پارامترها

g(n)= هزینه کم هزینه ترین مسیر از s به n

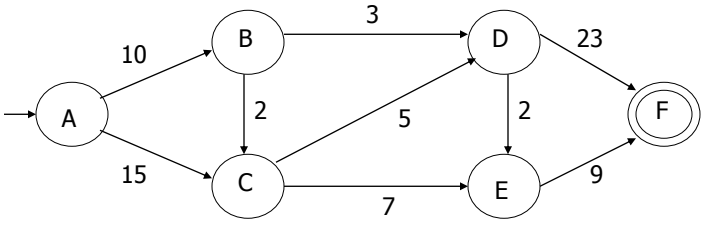
C(x,y)= هزینه رسیدن از x به y (به شرط x پدر y)

g(s)=0

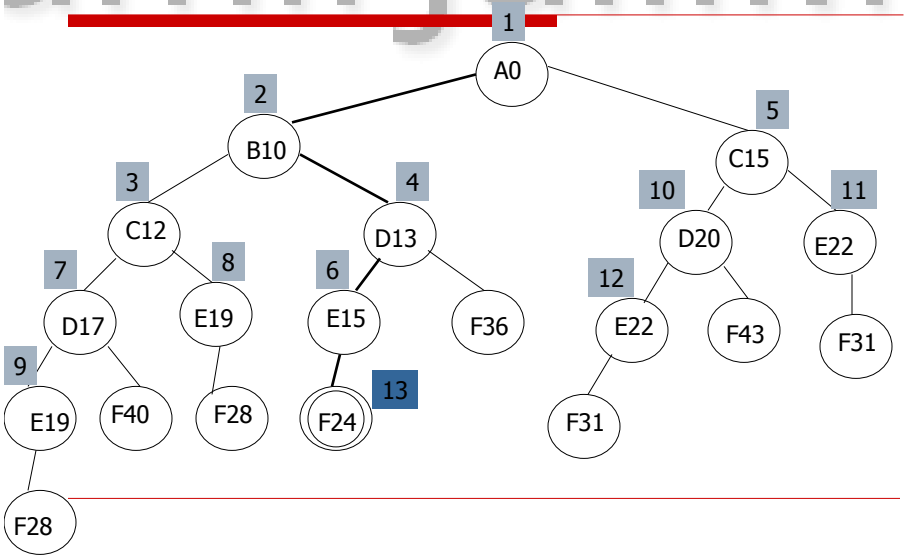
g(y)=g(x)+ C(x,y)

*این الگوریتم حریصانه نیست و بجای C به g توجه دارد.

جستجوي هزينه يكنواخت: مثال



جستجوي هزينه يكنواخت: مثال



مساله و جستجو

برای مساله و جستجو چهار حالت امکانپذیر است:

1. مساله گراف ، جستجو گراف.
2. مساله گراف ، جستجو درخت.
   اگر گراف مساله دور داشته باشد و جستجوی درختی(در عمق نه در پهنا) انجام گیرد ممکن است جستجو در دور بینهایت بیافتد.
3. مساله درخت ، جستجو درخت.
4. مساله درخت ، جستجو گراف.

فقط زمان و فضا را بخاطر تست Ex از دست میدهیم.

٦٤

راهبردهای جستجوی آگاهانه

- راهبردهای جستجوی ناآگاهانه میتوانستند راه حلها را با تولید نظام مند حالتهای جدید و آزمودن آنها در مقابل هدف بیابند، ولی ناکارا هستند.
- این بخش نسخه های آگاهانه الگوریتم های بخش قبل را توصیف میکند.
- یک راهبرد جستجوی آگاهانه با استفاده از دانش خاص مساله(فراتر از تعریف مساله)، میتواند راه حلها را بطور مؤثرتری نسبت به راهبردهای جستجوی ناآگاهانه بیابد.

amirjalili.ir

جستجوی اول بهترینBFS

- نمونه ای از الگوریتم های عمومی (درختی یا گرافی) است که در آنها یک گره براساس یک تابع ارزیاب $f(n)$ جهت گسترش انتخاب میشود.
- گرهی که کمترین ارزیابی را دارد انتخاب میشود زیرا مقیاس ارزیابی فاصله تا هدف میباشد.
- تابع ارزیابی کاملاً دقیق نیست و ممکن است گاهی اوقات ما را به بیراهه نیز بکشاند.
- تابع ارزیابی برای انتخاب بهترین گره ظاهری (نزدیک ترین به هدف)، از یک تابع تخمین استفاده میکند.

توابع هیوریستیک

- برای اعمال آگاهی و هوشمندی به الگوریتم های جستجو از هیوریستیک استفاده میکنیم.
- توابع هیوریستیک معمول ترین شکل رساندن اطلاعات اضافی مساله به الگوریتم هستند.
- در واقع تخمینی است برای زود رسیدن به جواب.
- این توابع حدس ما را در مورد جواب مساله به الگوریتم اعمال میکنند.
- چون حدس وارد الگوریتم شده قطعیت را از دست میدهیم.

تعاریف

- تابع هزینه مسیر g(n) :هزینه مسیر ازگره S تا گره n.
- تابع اکتشافی h(n) :هزینه تخمینی ارزانترین مسیر از گره n به G.
if(n=G) then h(n)=0
- تابع بهترین مسیر h*(n) :ارزان ترین مسیر واقعی از گره n تا گره G.
- تابع ارزیابي f(n) :هزینه تخمینی ارزان ترین مسیر از طریق n.
f(n)= g(n) + h(n)
- f*(n) : هزینه ارزان ترین مسیر از طریقn.
f*(n)= g(n) + h*(n)

انواع توابع ارزیابی

- **f(n)= g(n)**

چون عموماً g(n) قطعی است این روش همان UCS خواهد بود.

- **f(n)= h(n)**

یک الگوریتم حریصانه است که فقط مسیری را بسط میدهد که ظاهراً به هدف نزدیکتر است.

- **f(n)= g(n) + h(n)**

یک تابع ارزیاب مناسب که اثر بایاس g و h را خنثی میکند.

در الگوریتم A* کاربرد دارد.

ویژگی تابع ارزیاب f(n)

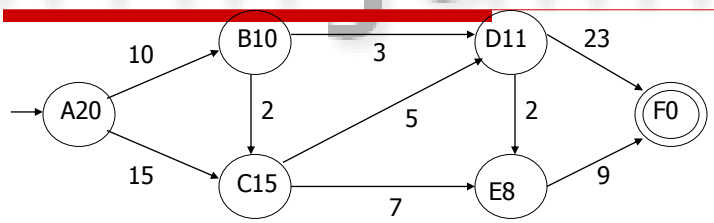
1. یک تخمین نزدیک به واقعیت به ما بدهد.
2. حتی الامکان خطای کمتری داشته باشد.
3. محاسبه آن ساده باشد.

- دقت h در تخمین مساله شدیداً دخالت دارد.
- اگر h(n)=∞ باشد یعنی که این گره اصلاً به جواب نمیرسد.
- اگر f<f* گوئیم دچار underestimate شده و منجر به جواب صحیح خواهد شد.
- اگر f>f* گوئیم دچار overestimate شده و ممکن است منجر به جواب صحیح (بهینه) نشود.
- یک h قابل قبول است اگر overestimate نباشد.

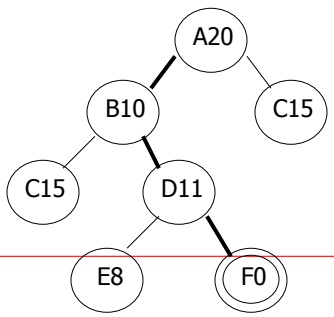
جستجوی اول بهترین حریصانه

- این روش سعی میکند تا نزدیکترین گره به هدف را گسترش دهد.
- پس در این روش $f(n)= h(n)$ خواهد بود.
- چون عموماً h تخمینی است یک اشتباه کوچک ممکن است ما را از رسیدن به جواب بهینه باز دارد.

جستجوي اول بهترين حريصانه: مثال



اعداد داخل گره، h میباشد.



جستجوي کمينه کردن کل هزينه تخمين راه حل: A*

- رايجترين شيوه از جستجوی اول بهترين ميباشد.
- در اين روش تابع ارزيابی عبارت است از:

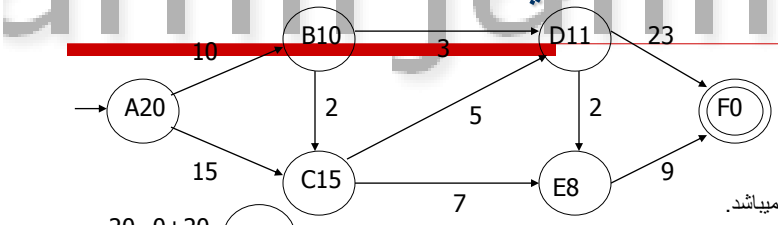
$$f(n) = g(n) + h(n)$$

- چون عموماً g هزينه قطعی (از گره S به گره n) و h هزينه تخمينی (از n گره به گره G) ميباشد پس:

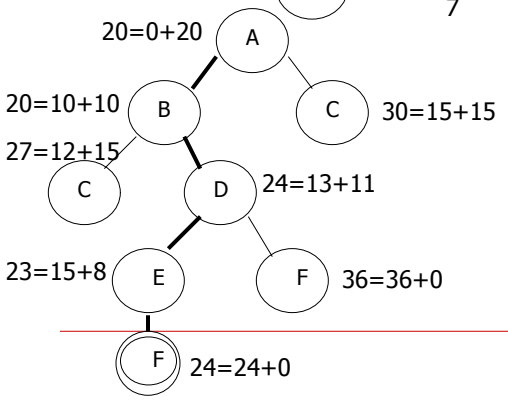
هزينه تخمينی ارزانترين راه حل از طريق n = f(n)

- اين راهبرد کاملاً عاقلانه است.

جستجوي A* : مثال ۱



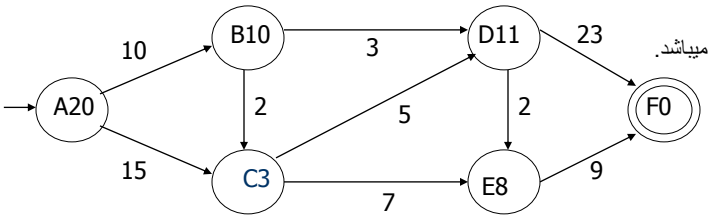
اعداد داخل گره، h ميباشد.



f=g+h

جواب مسير ABDEF

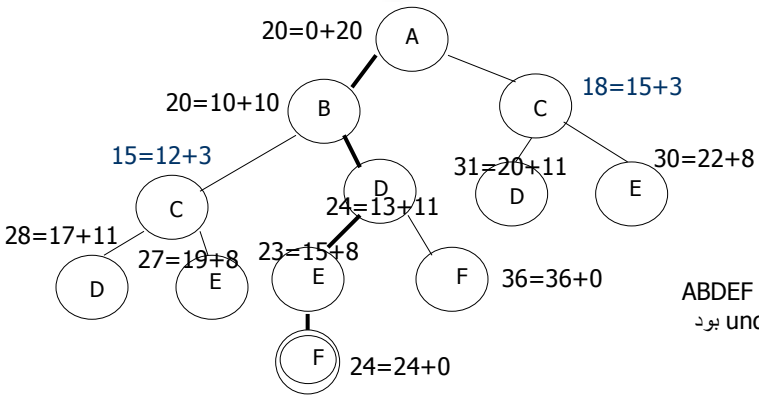
جستجوي A* : مثال ۱ با understimate



اعداد داخل گره، h میباشد.

amirjalili.ir

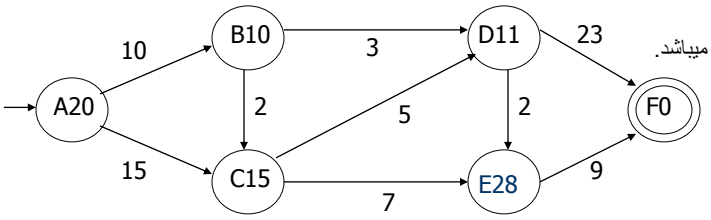
جستجوي A* : مثال ۱ با understimate



f=g+h

همچنان جواب مسیر ABDEF
چون understimate بود

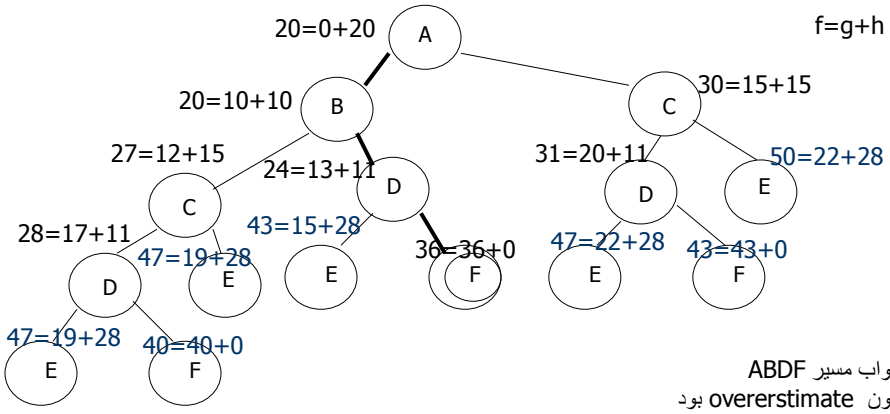
جستجوي A* : مثال ۱ با overstimate



اعداد داخل گره، h میباشد.

amirjalili.ir

جستجوي A* : مثال ۱ با overstimate



جواب مسیر ABDF
چون overerstimate بود

توابع هیوریستیک:مثال پازل

- هزینه متوسط راه حل یک نمونه تصادفی حدود ۲۲ گام است با فاکتور انشعاب تقریباً ۳.(حرکتها ۲یا۳یا۴)
- جست و جوی جامع تا عمق ۲۲ : $3^{22} \approx 3.1 \times 10^{10}$
- با نگهداری سابقه حالات تکراری، از یک حالت فقط 9!/2=181440 حالت متمایز قابل دسترسی وجود دارد.؟؟؟

(کل حالات 9!=362880)

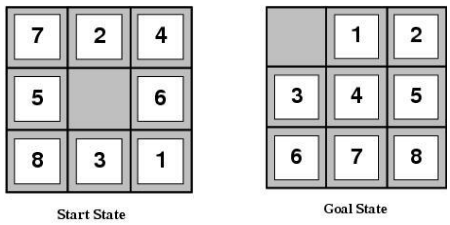
- پس به یک هیوریستیک خوب جهت کاهش مراحل جستجو نیاز داریم.
- یک تابع هیوریستیک خوب هرگز تعداد گامها تا هدف را بیش از اندازه واقعی برآورد نمیکند.

تابع هیوریستیک اول برای پازل

- h1 = تعداد خانه‌هايي كه در مكان‌هاي نادرست هستند.

h1 یک هیوریستیک خوب است، زیرا واضح است که هر خانه‌اي كه خارج از مكان درست باشد حداقل يكبار بايد جابجا شود.

h1=8



تابع هیوریستیک دوم برای پازل

- h2 = مجموع فواصل خانه‌ها از مکان‌هاي هدف صحيحشان است. فاصله‌اي كه ما حساب مي‌كنيم، مجموع فواصل عمودي و افقي است (نه حركت قطری) كه فاصله بلوک شهر يا منتهن ناميده مي‌شود. $h_2 = 3+1+2+2+2+3+3+2 = 18$
- h2 قابل قبول است زيرا هر جابجايي كه ميتواند انجام گيرد به اندازه يک مرحله به هدف نزديک ميشود.
- هيچكدام بيش از هزينه واقعی(۲۶گام) برآورد ندارند.

تاثير دقت هيوريستيک بر كارايي:مثال

	Search Cost			Effective Branching Factor		
d	IDS	A*(h_1)	A*(h_2)	IDS	A*(h_1)	A*(h_2)
2	10	6	6	2.45	1.79	1.79
4	112	13	12	2.87	1.48	1.45
6	680	20	18	2.73	1.34	1.30
8	6384	39	25	2.80	1.33	1.24
10	47127	93	39	2.79	1.38	1.22
12	3644035	227	73	2.78	1.42	1.24
14	–	539	113	–	1.44	1.23
16	–	1301	211	–	1.45	1.25
18	–	3056	363	–	1.46	1.26
20	–	7276	676	–	1.47	1.27
22	–	18094	1219	–	1.48	1.28
24	–	39135	1641	–	1.48	1.26

Figure 4.8 Comparison of the search costs and effective branching factors for the ITERATIVE-DEEPENING-SEARCH and A* algorithms with h_1, h_2. Data are averaged over 100 instances of the 8-puzzle, for various solution lengths.

تاثیر دقت هیوریستیک بر کارایی

- اگر برای هر گره n داشته باشیم: h2(n) >= h1(n)
  - h2 بر h1 برتری دارد.
  - برتر بودن مستقیما به کارایی ترجمه میشود.
  - تعداد گره هایی که A* با بکارگیری h2 بسط میدهد، هرگز بیش از بکارگیری h1 نیست.
- همیشه بهتر است از تابع هیوریستیک با مقادیر بالاتر استفاده کنیم. به شرطی که:
  1. بیش از اندازه تخمین نزند.
  2. زمان محاسبات برای اکتشاف، خیلی بزرگ نباشد.

فصل چهارم

الگوریتم های جستجوی محلی

الگوریتم های جستجوی محلی و مسائل بهینه سازی

- الگوریتمهای جستجو، فضای جستجو را بطور سیستماتیک بررسی میکنند و:
  - تا رسیدن به هدف یک یا چند مسیر نگهداری میشوند.
  - مسیر رسیدن به هدف، راه حل مسئله را تشکیل میدهد.
- در بسیاری از مسائل مسیر به هدف ارتباطی ندارد.
- اگر مسیر هدف اهمیت نداشته باشد، الگوریتم های متعددی مطرح میگردد که اصلاً به مسیر اهمیت نمیدهند.

amirjalili.ir

الگوریتم های جستجوی محلی و مسائل بهینه سازی

- الگوریتم های جستجوی محلی با استفاده از حالت فعلی عمل میکنند و عموماً فقط به همسایگان آن حالت تغییر مکان میدهند.
- معمولاً مسیرهای دنبال شده توسط جستجو، نگهداری نمیشوند.
- این الگوریتم ها اگرچه نظام مند نیستند ولی:

1. بسیار کم از حافظه استفاده میکنند.
2. اغلب در فضاهای بزرگ یا نامتناهی پیوسته(که الگوریتم های نظام مند مناسب نیستند)، به راه حل معقولی میرسند.

الگوریتم های جستجوی محلی و مسائل بهینه سازی

- الگوریتم های جستجوی محلی علاوه بر یافتن اهداف در مسائل خالص بهینه سازی نیز سودمند میباشند.
- در مسائل خالص بهینه سازی هدف، یافتن بهترین حالت بر اساس تابع هدف میباشد.

- یک الگوریتم جستجوی محلی کامل، همیشه هدف را در صورت وجود پیدا میکند.
- یک الگوریتم جستجوی محلی بهینه، همیشه یک کمینه یا بیشینه سراسری را پیدا میکند.

جستجوی تصادفی

- ساده ترین و ابتدایی ترین الگوریتم جستجوی محلی است.
- عدد تصادفی x را تولید و فرض میکنیم که f(x) بهینه است!!!!
- K بار عدد تصادفی x را تولید و برای هر کدام f(x) را پیدا میکنیم. سپس بهترین مقدار بین آنها را بهینه در نظر میگیریم.
- عملاً در این حالت نیز احتمال رسیدن به جواب بهینه کم است.

الگوریتم های جستجوی محلی و مسائل بهینه سازی

- این الگوریتم‌ها به دو گره اصلي تقسيم مي‌شوند:

- الگوريتم‌هاي تپه‌نوردي Hill-Climbing
  - تپه نوردی استاندارد
  - تپه نوردی با k شروع مجدد تصادفی
- سخت سازی شبیه سازی شده Simulated Annealing
  - پرتو محلی
  - الگوریتم ژنتیک

جستجوی تپه نوردی

- در این روش در یک حلقه، مدام در جهت افزایش مقدار حرکت میکنیم و الگوریتم هنگامی پایان مییابد که به قله ای برسیم که هیچیک از همسایه ها مقدار بیشتری نداشته باشد.

- این الگوریتم درخت جستجو را نگهداری نمیکند و هر گره فعلی تنها نیاز به ثبت حالت و مقدار تابع هدفش دارد.

- تپه نوردی فراتر از همسایه های مجاور حالت فعلی را نگاه نمیکند.

جستجوی تپه نوردی

- حلقه اي كه مدام در جهت افزايش مقدار (بطرف بالای تپه) حركت ميكند.
- رسيدن به بلندترين قله در همسايگی حالت فعلی، شرط خاتمه است.
- در صورت وجود بيش از يک بهترين پسين، معمولاً از اين مجموعه به تصادف يکی را انتخاب ميكند.

amirjalili.ir

جستجوی تپه نوردی

- جستجوی تپه نوردی، جستجوی محلی حريصانه نيز نام دارد. چون يک حالت همسايه خوب را بدون فكر قبلي كه از آنجا به كجا خواهد رفت، انتخاب ميكند.
- الگوريتم های حريصانه اغلب بسيار خوب عمل ميكنند.
- تپه نوردی اغلب پيشرفت سريعی به سمت راه حل دارد چون بهبود يک حالت بد خيلی ساده است.

جستجوی تپه نوردی

- در تپه نوردی یک x تصادفی تولید میکنیم و یک مقدار $\pm\Delta x$ به آن اعمال میکنیم هر کدام بیشتر باشد آنرا دنبال میکنیم تا زمانیکه:

$$f(x\pm\Delta x)<f(x)$$

- مشکل این الگوریتم تعیین ΔX میباشد.
- تعداد پسین ها برای فضای n بعدی:

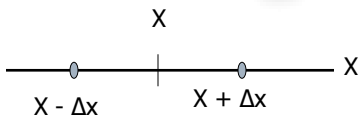
$$3^n-1$$

- این همان فاکتور انشعاب یا b میباشد.

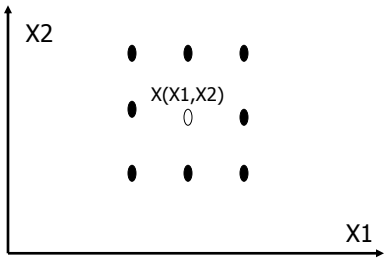
- چون فاکتور انشعاب نمایی است برای ابعاد بالا مناسب نیست.

جستجوی تپه نوردی:مثال

- فضای یک بعدی:



- فضای دو بعدی:

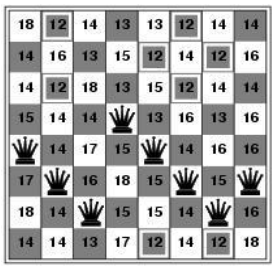


جستجوی تپه نوردی: مثال ۸ وزیر

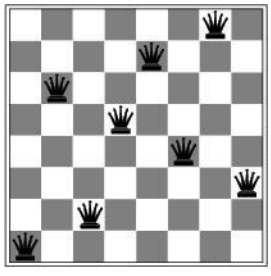
- الگوریتم های جستجوی محلی اغلب از فرمول بندی حالت کامل استفاده میکنند.
- در این روش هر حالت شامل ۸ وزیر، یکی در هر ستون میباشد.
- تابع پسین تمام حالاتی که با جابجا کردن یک وزیر به خانه دیگر در همان ستون تولید میشود.هر حالت 8*7=56 حالت پسین دارد.
- تابع هیوریستیک h، تعداد جفت وزیرهایی که نسبت به هم، مستقیم یا غیر مستقیم گارد دارند(همدیگر را تهدید میکنند).
- کمینه سراسری این تابع صفر است.(در راه حل کامل اتفاق می افتد)

amirjalili.ir

جستجوی تپه نوردی: مثال ۸ وزیر



حالتی با h=17
مقادیر تمام پسینها در شکل مشخص است
بهترین پسین دارای h=12میباشد



یک کمینه محلی با h=1
با ۵ گام فاصله از شکل قبلی
پسینها دارای h بالاتری میباشد

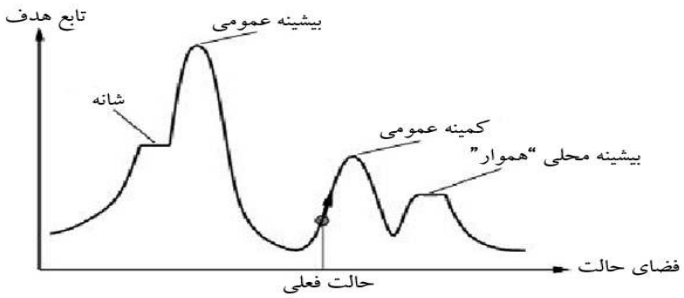
جستجوی تپه نوردی: معایب

1. بیشینه محلی: قله ای که از تمام حالتهای همسایه بلندتر و از بیشینه سراسری کوتاهتر است.
2. دماغه: یک رشته بیشینه محلی هستند که گذر از آنها برای الگوریتمهای حریصانه بسیار مشکل است.

1. فلات: ناحیه ای که در آن تابع ارزیاب ثابت است. میتواند یک بیشینه محلی مسطح نیز باشد که هیچ مسیر رو به بالایی ندارد یا یک شانه باشد که بتوان از آن بالاتر هم رفت. ممکن است.

یک دور نمای فضای حالت یک بعدی

الگوریتم تپه نوردی ممکن است به نقطه ای برسد که قادر به خروج از هیچیک از سه حالت قبلی نباشد.



جستجوی تپه نوردی: انواع

- تپه نوردی اتفاقی: بصورت تصادفی از میان حرکات رو به بالا یکی را انتخاب میکند.
- تپه نوردی اولین گزینه: از تپه نوردی اتفاقی استفاده میکند و بصورت تصادفی پسین تولید میکند تا زمانیکه پسین تولید شده از حالت فعلی بهتر باشد.
- همه این روشها ناکامل هستند و معمولاً در بیشینه محلی گیر میکنند.
- تپه نوردی با K شروع مجدد تصادفی مستقل

amirjalili.ir

تپه نوردی با K شروع مجدد تصادفی

- این روش یک مجموعه جستجوی تپه نوردی را از حالتهای شروع تصادفی اجرا میکند و هنگام پیدا شدن هدف متوقف میشود.
- به احتمال زیاد این روش کامل است. چون بالاخره یک حالت هدف را به عنوان حالت شروع تولید خواهد کرد.

- از بین k تپه نوردی بهترین جواب را انتخاب میکنیم.
- با افزایش k احتمال رسیدن به جواب افزایش پیدا میکند.

جستجوی تپه نوردی: ارزیابی

- موفقیت hill-climbing خيلي به ظاهر دور نمای فضاي حالت بستگي دارد: اگر فلات و ماكزيمم‌هاي محلي كمي وجود داشته باشد، تپه‌نوردي با شروع تصادفي خيلي سريع يک راه حل خوب را پيدا خواهد كرد.
- بسیاری از مسائل واقعی دارای دورنمایی شبیه جوجه تیغی هستند.
- مسائل سخت معمولاً دارای تعداد نمایی بیشینه محلی هستند.

جستجوی سخت سازی شبیه سازی شده

- نامهای مختلفی از جمله شبیه سازی حرارت دارد.
- چون الگوریتم تپه نوردی هرگز به سمت پایین (حالاتی با مقادیر کم یا هزینه بالا) حرکت نمیکرد ناکامل است.
- زیرا ممکن است در یک بیشینه محلی گیر کند.
- در مقابل جستجوی تصادفی کامل ولی ناکارا است.
- لذا ترکیب تپه نوردی و جستجوی تصادفی کار معقول است.
- شبیه سازی حرارت چنین الگوریتمی است.
- حرارت با درجه بالا و به تدریج سرد کردن.

جستجوی سخت سازی شبیه سازی شده

- در این الگوریتم علاوه بر ΔX یک Δt نیز داریم.

$$\Delta t >> \Delta x$$

- Δt ما را از یک بیشینه محلی بیرون می آورد.
- در هر مرحله Δt را کاهش میدهیم.
- زمانیکه قله جدید با قله قبلی یکی شد کار تمام است.

جستجوی شبیه سازی حرارت : مثال

- پایین آمدن توپ از شیب در سطحی ناهموار(کمینه کردن هزینه)
  - توپ در فرود از تپه به کمینه محلی میرود
  - با تکان دادن سطح، توپ از کمینه محلي خارج میشود
  - با تکان شدید شروع(دمای زیاد)
  - بتدریج کاهش شدت تکان (دمای پایین تر)

شدت تکانها باید مناسب باشد تا فقط از کمینه محلی خارج گردد.

اگر زمانبندی دما به اندازه کافی آهسته کاهش یابد، الگوریتم یک بهینه عمومی را مي یابد

جستجوی پرتو محلی

- نگهداری یک گره در حافظه واکنشی افراطی نسبت به مساله محدودیت حافظه است.
- جستجوی پرتو محلی اطلاعات k حالت را نگهداری میکند.
- K حالت تصادفی تولید و تمام پسینهای آنها تولید میشود اگر هر کدام از آنها هدف بود پایان میپذیرد در غیر اینصورت بهترین k پسین از لیست انتخاب و الگوریتم ادامه می یابد.
- اگر فاکتور انشعاب b باشد در هر مرحله bk پسین داریم.

amirjalili.ir

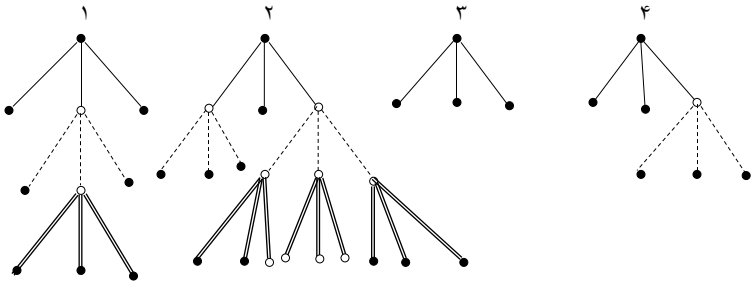
جستجوی پرتو محلی

- تفاوت پرتو محلی با k شروع مجدد تصادفی:
  - در جستجوی پرتو محلی، اطلاعات مفید بین k فرایند موازی مبادله میشود.
  - در جستجوی k شروع مجدد تصادفی، هر فرایند مستقل از بقیه اجرا میشود.

  بهترینها اینجاست!!!!
- پس کاملاً با یکدیگر تفاوت دارند.
- این الگوریتم به سرعت جستجوهای بی ثمر را رها و منابع را به جایی منتقل میکند که بیشترین پیشرفت صورت گیرد.

جستجوی پرتو محلی:مثال

- با فرض k=4 و b=3 :



جستجوی پرتو محلی:بهبود

- جست و جوی پرتو اتفاقی یا غیرقطعی:
  - به جای انتخاب بهترین k پسین، k جانشین تصادفی را انتخاب میکند. بطوریکه احتمال انتخاب یک پسین خاص، یک تابعی صعودی از مقدارش باشد.

- جست و جوی پرتو بهبود یافته:
  - در هر مرحله به bk فرزند تولید شده، N نقطه تصادفی تولید و اضافه کنیم سپس از بین bk+N گره، k گره بهینه را انتخاب کنیم و مرحله بعدی را تکرار کنیم. با اینکار یک تنوع به الگوریتم میدهیم.

الگوریتم های ژنتیک(GA)

- شکلی از جستجوی پرتو غیر قطعی است که:
  حالتهای پسین از طریق ترکیب دو حالت والد تولید میشود.
- از نظریه تکامل گرفته شده است.
- الگوریتم ژنتیک در سه زمینه مطرح است:

1. طبیعت
2. بهینه سازی
3. جستجوی فضای حالت

amirjalili.ir

الگوریتم های ژنتیک

- در GA نیز شبیه جستجوی پرتو با یک مجموعه تصادفی که جمعیت یا population نامیده میشود شروع میکنیم.
- میزان جمعیت بر اساس اندازه جمعیت یا popsize مشخص میگردد.
- براساس تعداد نسل یا maxgen الگوریتم اقدام به تکرار تولید نسل میکند.

الگوریتم ژنتیک: عملگر های اصلی

- افراد جمعیت را توسط تابع برازش یا Fitness function ارزیابی میکنیم.

1. تابع انتخاب یا selection بر اساس ارزیابی Ff اقدام به انتخاب از بین نسلها به اندازه popsize میکند.
2. تابع برش(پیوند یا تقاطع یا ترکیب) یا crossover نقطه پیوند را جهت تولید فرزندان(offspring) از والدین(parents) نشان میدهد.
3. تابع جهش یا mutation نیز نشان دهنده میزان جهش در بیتها(ژنها) میباشد.

الگوریتم های ژنتیک

- هر حالت یا فرد از جمعیت (کروموزوم) بصورت یک رشته از الفبای محدود (ژن) نشان داده میشود.(معمولاً 0,1)
- مثال ۸ وزیر:

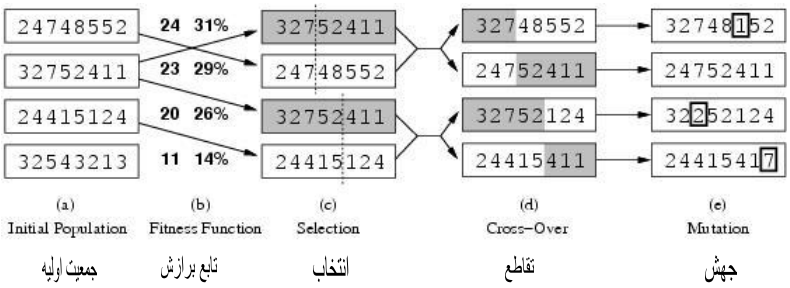
1. در هر حالت باید موقعیت ۸ وزیر را هر یک در یک ستون (با ۸ خانه) نشان داد. پس تعداد بیت(ژن) مورد نیاز:

$$8 * \log_2^8 = 24$$

2. هر حالت با ۸ رقم که هر یک بین ۱ تا ۸ هستند نشان داده شود. پس این روش ۸ ژن خواهد داشت.

الگوریتم های ژنتیک:مثال ۸ وزیر

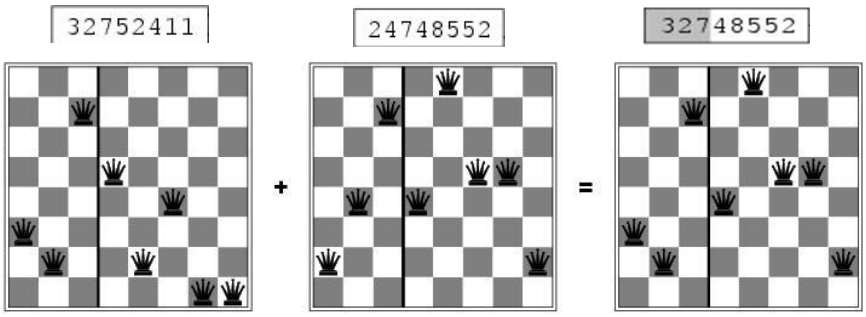
در این شکل در a جمعیت اولیه (۴ رشته ۸ بیتی) دیده میشود.
تولید نسل بعدی حالتها در b تا e دیده میشود.



amirjalili.ir

الگوریتم های ژنتیک:مثال ۸ وزیر

حالتهای ۸ وزیر متناظر با دو والد اول و فرزند اول



الگوریتم های ژنتیک

- یک تابع برازش برای حالتهای بهتر باید عدد بزرگتری برگرداند.
- در این مثال تعداد وزیرهایی که به هم حمله نمیکنند. برای یک راه حل برابر ۲۸میباشد.؟؟؟
- در این روش خاص از GA، احتمال انتخاب مستقیماً متناسب است با Ff.
- یک مورد دو بار انتخاب شده است و موردی دیگر اصلاً.
- برای هر جفت یک نقطه پیوند تصادفی انتخاب میشود.

amirjalili.ir

الگوریتم های ژنتیک

- اغلب در اوایل فرایند تقاطع جمعیت کاملاً متفاوت هستند. بنابراین در اوایل پیوند گامهای بزرگی در فضای حالت بر میدارد (مانند SA) و بعدها که اکثر نمونه ها مشابه شدند گامهای کوچکتری بر میدارد.
- GA شبیه جستجوی پرتو اتفاقی با ترکیب اکتشاف اتفاقی و تبادل اطلاعات میان رشته های موازی بالاتر میرود.
- مزیت اصلی GA عمل پیوند میباشد.

الگوریتم های ژنتیک

- به حاصل یک crossover, mutation, selection نسل جدید میگویند.
- در مسائل بهینه سازی بهترین نفر از جمعیت آخر به عنوان جواب برگردانده میشود.
- Crossover میتواند چند نقطه ای نیز باشد. یعنی برش و اتصال از بیش از یک نقطه صورت گیرد.
- جهش میتواند تصادفی نباشد. یعنی کپی از فرد دیگر باشد یا با چرخ رولت یا با تورنمنت انتخاب گردد.

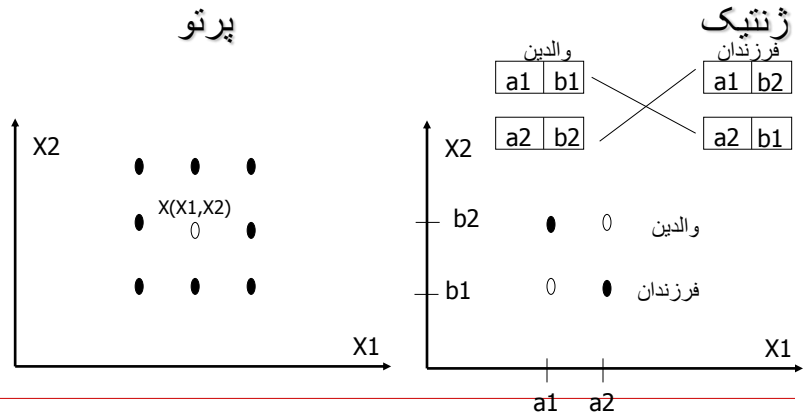
amirjalili.ir

الگوریتم های ژنتیک و جستجوی فضای حالت

- GA حالت خاصی از جستجوی پرتو اتفاقی است که bk فرزند توسط crossover تولید میشود و N فرزند تصادفی نیز توسط mutation ایجاد میگردد.
- GA فضای حالت را به شکل گسسته بررسی میکند.
- برای بررسی پیوسته فضای حالت نیاز به PSO داریم.

الگوریتم ژنتیک :مثال

- فضای دو بعدی گسسته:



بهینه سازی گروهی ذرات(PSO)

- مانند الگوریتم GA تعدادی جمعیت دارد ولی بر خلاف آن اقدام به تولید جمعیت جدید نمیکند (تا ضعیف ها از بین بروند) بلکه اقدام به حرکت دادن جمعیت ها میکند.

- جمعیت یا (swarm) به سمت بهترین فرد یا (partial) حرکت میکنند.

بهینه سازی گروهی ذرات(PSO)

- ممکن است افراد به دو سمت نیز حرکت کنند:
  - Global best:بهترین نقطه که همه به آن رسیده اند.
  - personal best: بهترین نقطه که خود به آن رسیده اند.

- ممکن است وقتی تعداد افراد زیاد است به گروههای مختلف تقسیم کنیم . در اینجا سه حرکت خواهیم داشت:
  - Global best:بهترین نقطه که همه به آن رسیده اند.
  - Personal best:بهترین نقطه که خود به آن رسیده اند.
  - Local best:بهترین نقطه که افراد گروه به آن رسیده اند.

مقایسه GA با PSO

- هر دو بهینه سازی یک هدفه هستند.

- GA فضای حالت را گسسته بررسی میکند ولی PSO پیوسته.

- در PSO نقاط بوجود آمده از بین نمیروند بلکه به سمت بهترین نقطه متمایل میشود.

بهینه سازی چند هدفه(MOO)

- زمانی مطرح است که بیش از یک (معمولاً دو) هدف برای بهینه سازی مطرح باشد.

- هر دو میتواند maximization باشد.
  مانند کارخانه: بیشترین استفاده از خط تولید و بیشترین سود.
- هر دو میتواند minimization باشد.
  مانند سوئیچ شبکه: کمترین ترافیک و کمترین فاصله.
- میتواند ترکیبی باشد.
  مانند خرید: بیشترین کیفیت و کمترین قیمت.

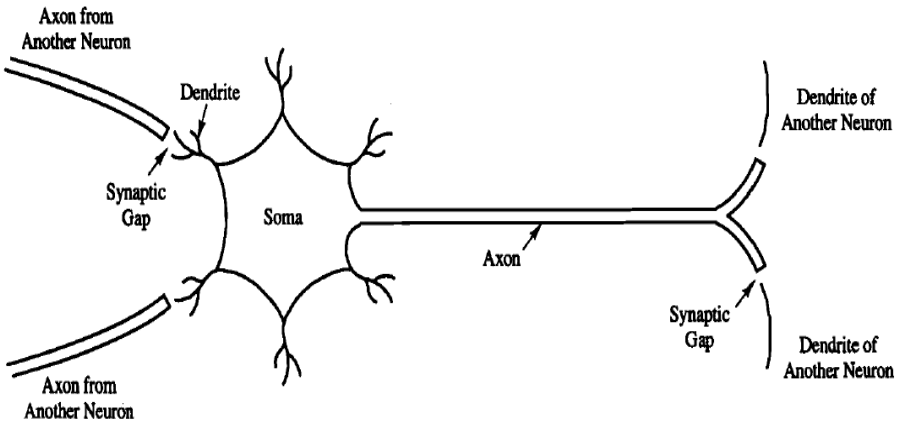
فصل پنجم

شبکه های عصبی مصنوعی

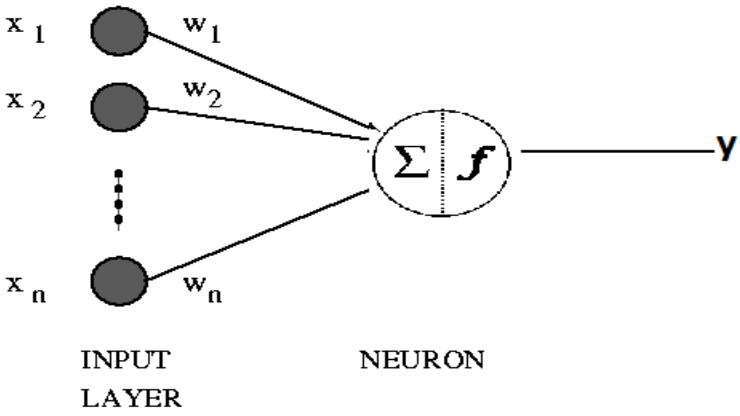
مقدمه

- شبکه های عصبی مصنوعی، از سیستم عصبی انسان الگو گرفته شده است که در آنها یک مجموعه پیچیده از نرونهای به هم متصل در کار یادگیری دخیل هستند.
- مغز انسان از حدود 10^{11} نرون تشکیل شده که هر نرون با تقریباً 10^{4} نرون دیگر در ارتباط است.
- سرعت سوئیچنگ نرونها در حدود 10^{-3} ثانیه است که در مقایسه با کامپیوترها (10^{-10} ثانیه) بسیار ناچیز مینماید. با این وجود آدمی قادر است در 0.1 ثانیه تصویر یک انسان را بازشناسائی نماید. این قدرت فوق العاده باید از پردازش موازی توزیع شده در تعدادی زیادی از نرونها حاصل شده باشد.

الگوی یک عصب بیولوژیکی



الگوی یک نرون مصنوعی



ساختار نرون مصنوعی

- تابع جمع کننده:

$$I = \sum_{i=1}^{n} w_i x_i$$

- تابع فعالساز:

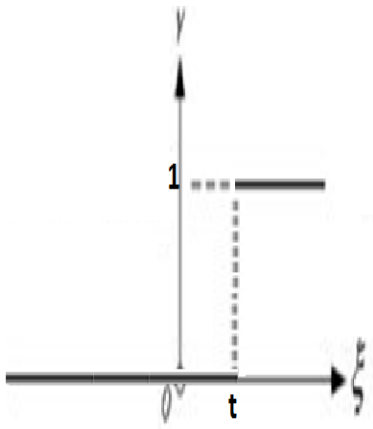
$$y = f(I)$$

انواع توابع فعاساز

1. تابع فعالساز آستانه ای:
2. تابع فعالساز علامت:
3. تابع فعالساز خطی:
4. تابع فعالساز سیگموئید:

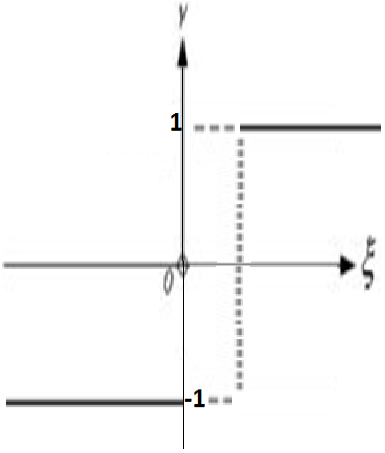
تابع فعالساز آستانه ای

$$Y = f(I) = \begin{cases} 1 \dots I > t \\ 0 \dots I \le t \end{cases}$$



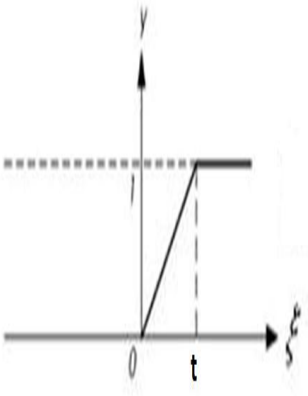
تابع فعالساز علامت

$$Y = f(I) = \begin{cases} 1 \dots I > t \\ -1 \dots I \le t \end{cases}$$



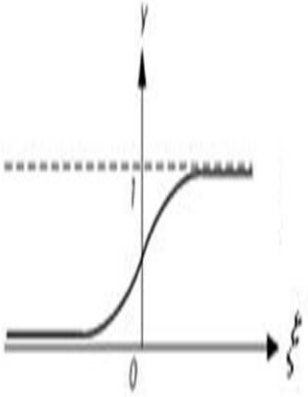
تابع فعالساز خطی

$$Y = f(I) = \begin{cases} 1 \dots I > t \\ I \dots 0 \le I \le t \\ 0 \dots I < 0 \end{cases}$$



تابع فعالساز سیگموئیدی

$$Y = f(I) = \frac{1}{1+e^{\alpha I}}$$



تابع فعالساز سیگموئیدی: مزایا

- غیر خطی است.
  قدرت مدل کردن سیستم را بالا میبرد.
- کران بالای آن معلوم است.
  برای پایداری سیستم مهم است.
- وجود ضریب آلفا.
  باعث انعطاف در عملکرد تابع میشود.
- آستانه ای نیست.
  مقدار حد آستانه نیاز ندارد.

شبکه های عصبی: انواع

- شبکه های عصبی به سه دسته اصلی تقسیم میگردند:

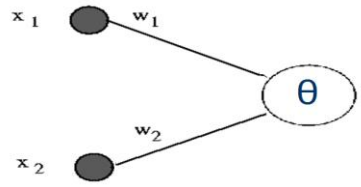
1. شبکه های عصبی بدون حافظه
2. شبکه های عصبی حافظه دار
3. شبکه های عصبی رقابتی

پرسپترون

- در سال ۱۹۵۸ روزن بلات مدل ابتکاری پرسپترون را ارائه کرد که پایه شبکه های عصبی مصنوعی است.
- او برای گره های ورودی وزن گذاشت و آنها را از حالت انحصاری بودن (۰ و ۱) در آورد.
- پرسپترن یک شبکه عصبی بدون حافظه میباشد.
- در طراحی یک شبکه عصبی مصنوعی، مساله اصلی یافتن مقادیر وزنها یا ماتریس وزنها میباشد.

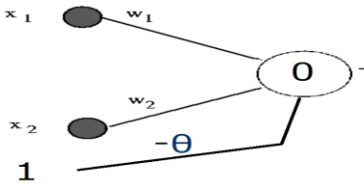
پرسپترون

- مدل اولیه:



$$x_1 w_1 + x_2 w_2 > \theta$$

- مدل توسعه یافته:



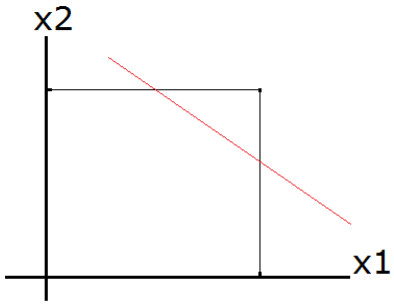
$$x_1 w_1 + x_2 w_2 - \theta > 0$$

- حاوی ورودی بایاس

پرسپترون: مثال

X1	x2	AND	x1w1+x2w2	OR	x1w1+x2w2
0	0	0	0≤θ	0	0≤θ
0	1	0	w2≤θ	1	w2>θ
1	0	0	w1≤θ	1	w1>θ
1	1	1	w1+w2>θ	1	w1+w2>θ

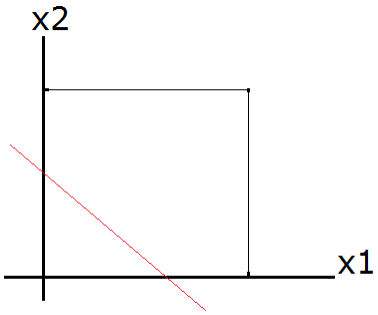
پرسپترون: مثال۱



AND

$$W_1 = W_2 = 0.4$$
$$\theta = 0.5$$

پرسپترون: مثال۲



OR

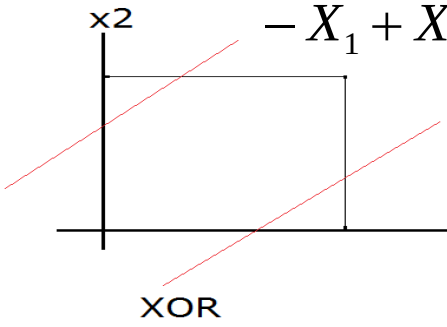
$$W_1 = W_2 = 0.4$$
$$\theta = 0.3$$

پرسپترون: مثال۳

X1	x2	XOR	x1w1+x2w2
0	0	0	0≤θ
0	1	1	w2>θ
1	0	1	w1>θ
1	1	0	w1+w2≤θ

amirjalili.ir

پرسپترون: مثال۳



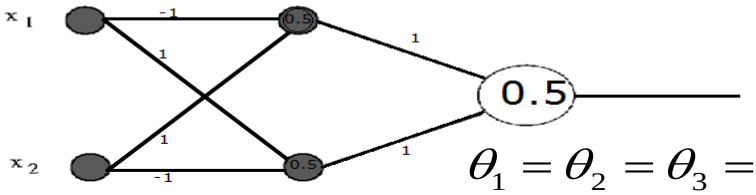
$$-X_1 + X_2 = 0.5$$

$$X_1 - X_2 = 0.5$$

- این یک تناقض است.
- با این روش نمیتوان مسائلی شبیه به XOR را پیاده سازی کرد.

پرسپترون: مثال۳

$$xor \Rightarrow (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2})$$



$$\theta_1 = \theta_2 = \theta_3 = 0.5$$

$$w_{1,1} = w_{2,2} = -1$$

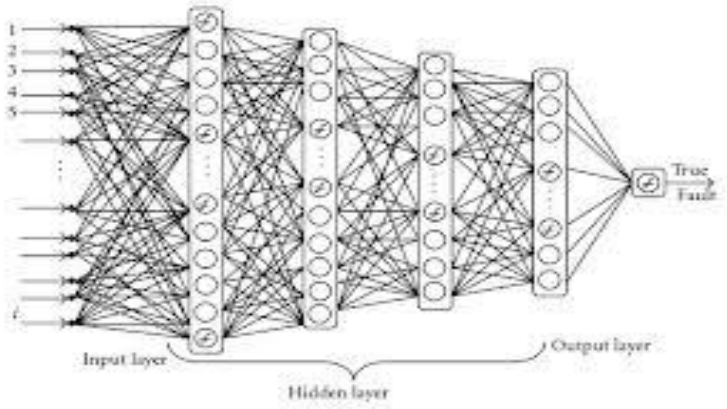
$$w_{1,2} = w_{2,1} = +1$$

$$w_1 = w_2 = +1$$

پرسپترون چند لایه یا MLP

- برای پیاده سازی توابعی شبیه به XOR که بشکل خطی قابل جداسازی نیستند یک لایه جوابگو نیست.
- برای جداسازی غیر خطی از حداقل دو لایه استفاده میکنیم.
- به پرسپترونهای بیش از یک لایه، MLP اطلاق میشود.
- اصلاح مقادیر ماتریس وزنها(که بصورت تصادفی تولید شده اند) جهت واقعی کردن آنها، یادگیری نام دارد.

پرسپترون چند لایه یا MLP

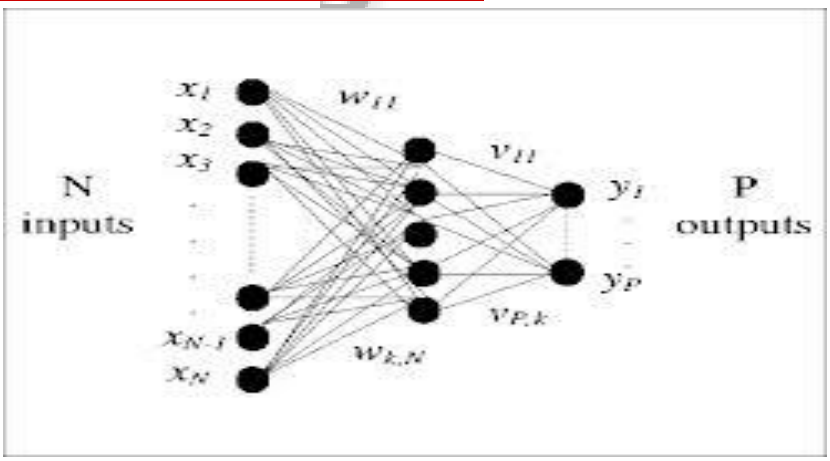


amirjalili.ir

پرسپترون چند لایه یا MLP

- اثبات شده است که فقط با یک لایه میانی میتوان هر نوع تفکیک غیر خطی را انجام داد.
- البه تعداد نرونهای لایه میانی (پنهان) بسیار مهم است.
- طبق تئوری کولموگروف برای این منظور تعداد نرونهای لایه میانی حداکثر باید $2n+1$ باشد.
- تعداد نرونهای لایه میانی در نرسیدن یا دیررسیدن به مقدار واقعی، به همگرایی یا convergency شبکه بسیار موثر است.

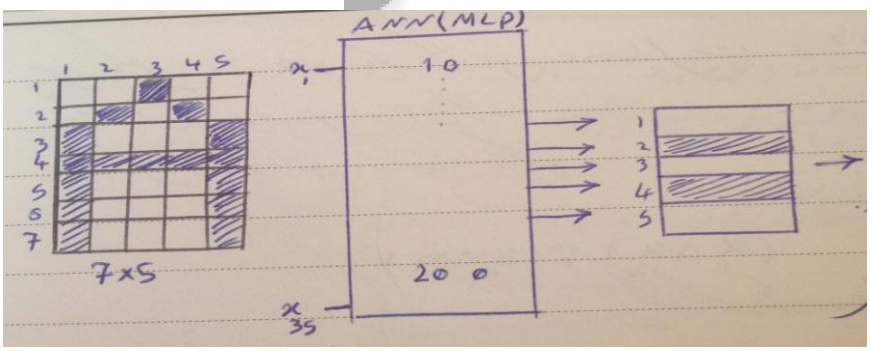
پرسپترون چند لایه یا MLP



پرسپترون چند لایه: کاربردها

- شناسائی الگو
- کدگذاری اطلاعات
- فشرده سازی اطلاعات
- کلاس بندی
- خوشه بندی
- تشخیص
- ....

پرسپترون چند لایه: مثال کاربردها



- اصطلاحاً گوئیم ۸۰۰ درجه آزادی داریم.

Weight=35*20+20*5=800

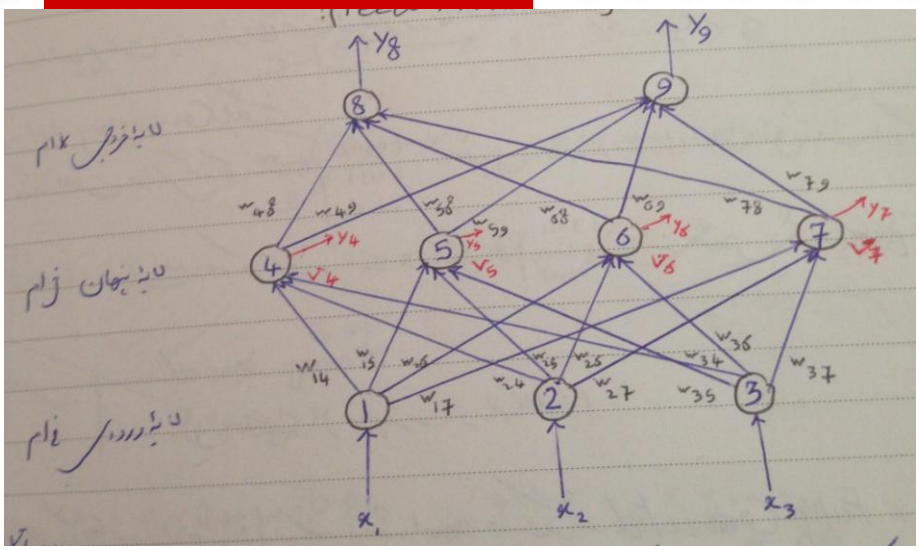
پرسپترون چند لایه

- اگر در یک MLP تمام نود ها با هم ارتباط داشته باشند آنرا اصطلاحاً **اتصال قوی** یا fully connected گویند.

- اگر در یک MLP ارتباط تمام نود ها با لایه بعدی باشند آنرا اصطلاحاً **پیشرو** یا پیش خور یا feed forward گویند.

- اگر در یک MLP ارتباط تمام نود ها با لایه قبلی باشند آنرا اصطلاحاً پسرو یا **بازخور** یا feedback گویند.

پرسپترون چند لایه: مثال



amirjalili.ir

پرسپترون چند لایه: مثال

- در این شبکه MLP پیشرو با اتصال قوی:
  - لایه ورودی دارای ۳ نرون
  - لایه میانی دارای ۴ نرون
  - لایه خروجی دارای ۲ نرون

- با فرض استفاده همه گره ها از تابع خطی، ماتریس وزنها به شکل زیر محاسبه میگردد:

پرسپترون چند لایه: مثال

- محاسبه ماتریس وزن لایه ورودی به لایه میانی:

$$\begin{bmatrix} v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} w_{14} & w_{24} & w_{34} \\ w_{15} & w_{25} & w_{35} \\ w_{16} & w_{26} & w_{36} \\ w_{17} & w_{27} & w_{37} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \Rightarrow v_j = w_{ij}.x_i$$

amirjalili.ir

پرسپترون چند لایه: مثال

- محاسبه ماتریس وزن لایه میانی به لایه خروجی:

$$\begin{bmatrix} y_8 \\ y_9 \end{bmatrix} = \begin{bmatrix} w_{48} & w_{58} & w_{68} & w_{78} \\ w_{49} & w_{59} & w_{69} & w_{79} \end{bmatrix} \times \begin{bmatrix} v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} \Rightarrow \begin{array}{c} y_k = w_{jk}.v_j \\ y_k = w_{jk}.w_{ij}.x_i \end{array}$$

انواع یادگیری یا آموزش شبکه های عصبی

- آموزش با ناظر یا supervised :
  - در این روش خروجی مطلوب در دسترس میباشد و با محاسبه خطا، شبکه راطوری آموزش میدهیم تابطرف آن رهنمون گردد.
  - بیشتر برای classification کاربرد دارد.

- آموزش بدون ناظر یا unsupervised :
  - در این روش خروجی مطلوب در دسترس نمیباشد پس خطلا مطرح نیست و طبق قاعده ای شبکه را آموزش میدهیم.
  - بیشتر برای clustering کاربرد دارد.

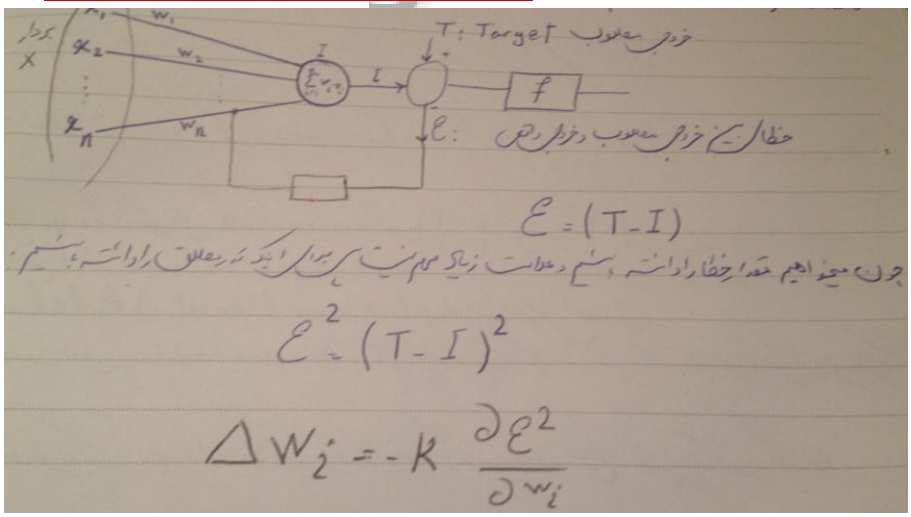
قانون دلتا یا delta rule

- یکی از قوانین رایج برای شبکه های عصبی میباشد که توسط ویدرو و هوف پیشنهاد شده است.

- طبق این قانون: میزان تغییرات بروی وزنها در هر مرحله، متناسب است با مشتق خطا نسبت به وزن ضربدر ضریب یادگیری.

- چون بحث خطا مطرح است پس نوعی روش با نظارت میباشد.

قانون دلتا(ویدرو – هوف) یا delta rule



یادگیری پس انتشار یا back propagation

- خاص شبکه های MLP میباشد که از قانون یادگیری دلتا استفاده میکند.
- الگون آن شامل حرکت رو به عقب، محاسبه خطا و تعدیل وزنهای تک تک لایه میباشد.
- مزیت این روش آنست که چون فقط خروجی مطلوب را برای لایه آخر داریم، در اینجا مشکل ساز نیست.

یادگیری پس انتشار یا back propagation

- طبق این الگوریتم، تغییرات وزنها از فرمول کلی زیر تبعیت میکند:

$$w_{ij}(N+1) = w_{ij}(N) + \alpha(\Delta w_{ij}(N))$$

- آلفا مقداری است بین صفر تا یک و ضریب یادگیری نام دارد.
- محاسبه تغییرات دلتابرای لایه های مختلف متفاوت وپیچیده است.
- محاسبات طولانی و پیچیده، ایراد اصلی این الگوریتم میباشد.

پرسپترون چند لایه: معایب

- همگرایی آن تضمین شده نیست.
- بهینه کردن پارامترهای مختلف آن دشوار است.
- سرعت آموزش آن در مقایسه با سایر شبکه ها، پایین است.

✓ شبکه LVQ ایرادات فوق را مرتفع میسازد.

فصل ششم

سیستم های فازی

سیستم های فازی

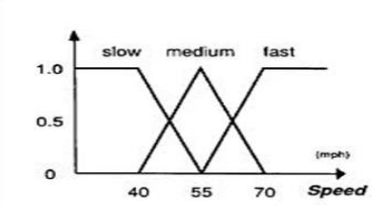
- واژه فازي بصورت مبهم، گنگ، نادقیق، گیج، مغشوش، در هم و نامشخص تعریف شده است.
- سيستم هاي فازي، مبتنی بر دانش یا قواعد میباشند. قلب یک سيستم فازي یک پایگاه دانش بوده که از قواعد اگر آنگاه فازي تشکیل شده است.

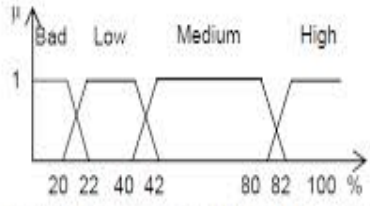
- بعنوان مثال عبارت فازي زیر را در نظر بگیرید:

اگر سرعت اتومبیل بالا است، آنگاه نیروي کمتري به پدال گاز وارد کنید.

سیستم های فازی

- یک قاعده اگر آنگاه فازي یک عبارت اگر آنگاه بوده که بعضی کلمات آن بوسیله توابع تعلق پیوسته مشخص شده اند.
- در سیستم های فازی از یک تابع تعلق استفاده میشود که معمولاً مقداری در بازه صفر و یک دارد.
- مثال: ۱- تابع تعلق نمره ۲- تابع تعلق سرعت





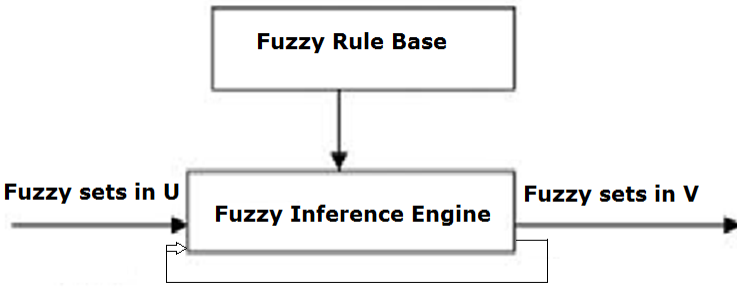
انواع سیستم های فازی

۱- سیستم هاي فازی خالص

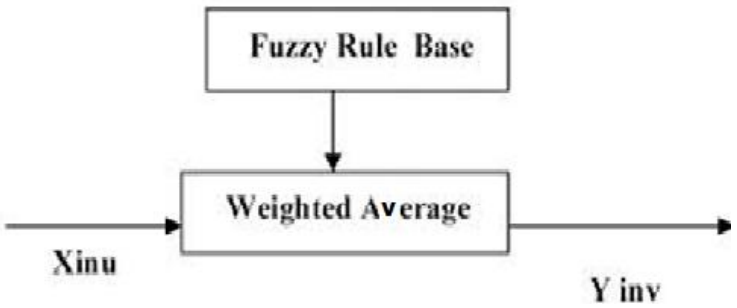
۲- سیستم هاي فازي تاکاگی - سوگنو - کانگ (TSk)

۳- سیستم هاي با فازي ساز و غیر فازي ساز

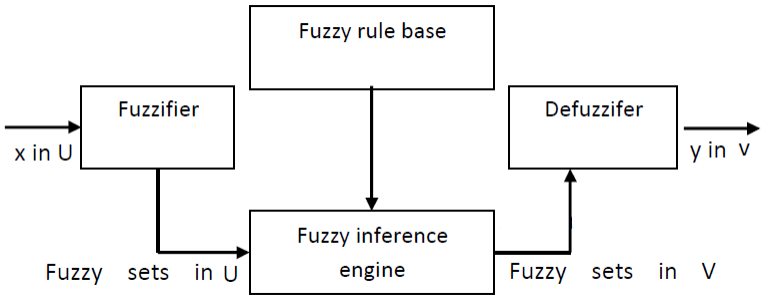
سیستم های فازی خالص



سیستم های فازی تاکاگی – سوگنو - کانگ



سیستم های با فازی ساز و غیر فازی ساز



فصل هفتم

سیستم های ترکیبی

سیستم های ترکیبی

- ترکیبی از ابزارهای مختلف هوش مصنوعی مانند شبکه های عصبی، ژنتیک و فازی است.
- شبکه های عصبی مختلف با یکدیگر نیز ترکیب و شبکه های هیبرید موسوم به HNN تشکیل میدهند.
- رایجترین ترکیب، سیستم های عصبی – فازی هستند.

سیستم های NEURO - FUZZY

- ترکیبی از شبکه های عصبی و فازی است.
- معمولاً ورودیها به شکل فازی وارد شبکه عصبی میگردند.
- انعطاف پذیر است.
- برای پیاده سازی سیستم های واقعی کاربرد دارد.