





آموزش هوش مصنوعی

ترجمه: محمد مهدی امانی

آبان 1402

آموزش هوش مصنوعی

این آموزش، دانشی مقدماتی در مورد هوش مصنوعی ارائه می دهد. اگر قصد یادگیری هوش مصنوعی را داشته باشید، به شما کمک بزرگی خواهد کرد. در این آموزش، می توانید به طور خلاصه درباره زمینه‌هایی از هوش مصنوعی که در آنها تحقیقات در حال پیشرفت است، اطلاعاتی کسب کنید.

مخاطبان

این آموزش برای دانش پژوهان در سطح مقدماتی و مبتدی که مشتاق یادگیری هوش مصنوعی هستند تهیه شده است.

پیشنیاز آموزش

دانش اولیه علوم کامپیوتر الزامی است. دانش ریاضیات، زبان، علوم، مکانیک یا مهندسی برق امتیاز محسوب می شود.

آموزش هوش مصنوعی

- 10..... هوش مصنوعی - بررسی اجمالی
- 10..... هوش مصنوعی چیست؟
- 10..... فلسفه‌ی هوش مصنوعی
- 11..... اهداف هوش مصنوعی
- 11..... چه چیزی به هوش مصنوعی کمک می کند؟
- 13..... برنامه‌نویسی با هوش مصنوعی و بدون هوش مصنوعی
- 14..... تکنیک‌های کلیدی در هوش مصنوعی چیست؟
- 14..... کاربردهای هوش مصنوعی
- 16..... تاریخچه هوش مصنوعی
- 18..... هوش مصنوعی - سیستم‌های هوشمند
- 18..... هوش چیست؟
- 18..... انواع هوش
- 19..... هوش از چه چیزهایی تشکیل شده است؟
- 22..... تفاوت بین هوش انسان و هوش ماشین
- 22..... هوش مصنوعی - حوزه‌های تحقیق
- 23..... یکی از حوزه‌ها در این زمینه، تشخیص گفتار و صدا
- 23..... عملکرد سیستم‌های تشخیص گفتار و صدا
- 24..... کاربردهای عملی حوزه‌های تحقیق
- 25..... طبقه‌بندی وظایف هوش مصنوعی
- 27..... هوش مصنوعی - عوامل و محیط‌ها
- 27..... عامل و محیط چیست؟
- 28..... اصطلاحات عامل

- 28..... عقلانیت
- 29..... عامل ایده آل عقلانی چیست؟
- 29..... ساختار عوامل هوشمند
- 30..... عوامل بازتاب مدل
- 31..... عوامل مبتنی بر هدف
- 32..... عوامل مبتنی بر سود
- 32..... طبیعت محیط ها
- 33..... ویژگی های محیط
- 34..... هوش مصنوعی - الگوریتم های جستجوی محبوب
- 34..... مشکلات مسیریابی عامل واحد
- 34..... اصطلاحات جستجوی
- 35..... استراتژی های جستجوی **brute-force**
- 35..... جستجوی پهنه ای
- 36..... جستجوی عمق-اول
- 37..... جستجوی دوطرفه
- 37..... جستجوی هزینه یکنواخت
- 38..... جستجوی عمق-اول تعمیق تکراری
- 38..... مقایسه پیچیدگی الگوریتم های مختلف
- 39..... استراتژی های جستجوی آگاهانه (هوشیار)
- 39..... توابع ارزیابی هیستریک
- 39..... جستجوی هیستریک خالص
- 40..... جستجوی A^*
- 40..... جستجوی بهترین اول حریصانه

- 40..... الگوریتم‌های جستجوی محلی
- 40..... جستجوی صعودی
- 41..... جستجوی پرتو محلی
- 42..... آنیلینگ شبیه‌سازی شده
- 43..... مسئله فروشنده دوره‌گرد
- 43..... هوش مصنوعی - سیستم‌های منطق فازی
- 44..... منطق فازی چیست؟
- 44..... پیاده‌سازی
- 45..... چرا منطق فازی؟
- 45..... معماری سیستم‌های منطق فازی
- 47..... مثالی از یک سیستم منطق فازی
- 48..... الگوریتم
- 48..... توسعه
- 48..... مرحله 1 - تعریف متغیرها و اصطلاحات زبانی
- 48..... مرحله 2 - ساختن توابع عضویت برای آنها
- 49..... مرحله 3 - ساختن قوانین پایگاه دانش
- 49..... مرحله 4 - مقدار فازی را بدست آورید
- 50..... مرحله 5 - دوددهی را انجام دهید
- 50..... موارد کاربرد منطق فازی
- 52..... هوش مصنوعی - پردازش زبان طبیعی
- 52..... اجزای NLP
- 53..... فهم زبان طبیعی (NLU)
- 53..... تولید زبان طبیعی (NLG)

- 53..... مشکلات در NLU
- 54..... اصطلاحات NLP
- 55..... مراحل در پردازش زبان طبیعی
- 56..... جنبه‌های پیاده‌سازی تجزیه و تحلیل نحوی
- 57..... گرامر آزاد از متن (CFG)
- 59..... تجزیه‌کننده بالا به پایین (Top-Down Parser)
- 60..... هوش مصنوعی - سیستم‌های خبره
- 60..... سیستم‌های خبره چیست؟
- 60..... ویژگی‌های سیستم‌های خبره
- 60..... قابلیت‌های سیستم‌های خبره
- 61..... اجزای سیستم‌های خبره
- 62..... پایگاه دانش
- 62..... دانش چیست؟
- 62..... اجزای پایگاه دانش
- 63..... نمایش دانش
- 63..... یادگیری دانش
- 63..... موتور استنتاج
- 65..... رابط کاربری
- 66..... نیازهای رابط کاربری کارآمد سیستم‌های خبره
- 66..... محدودیت‌های سیستم‌های خبره
- 66..... کاربردهای سیستم‌های خبره
- 67..... تکنولوژی سیستم‌های خبره
- 68..... توسعه سیستم‌های خبره: گام‌های عمومی

71	هوش مصنوعی - رباتیک
71	ربات‌ها چیستند؟
71	هدف
71	رباتیک چیست؟
71	جنبه‌های رباتیک
71	تفاوت در سیستم ربات و دیگر برنامه‌های هوش مصنوعی
72	حرکت ربات
72	حرکت پایه‌ها
74	حرکت چرخ‌دار
75	اجزای یک ربات
76	بینایی کامپیوتر
76	سخت‌افزار سیستم بینایی کامپیوتر
78	کاربردهای رباتیک
79	هوش مصنوعی - شبکه‌های عصبی
79	شبکه‌های عصبی مصنوعی چیستند؟
79	ساختار اساسی شبکه‌های عصبی مصنوعی
81	انواع شبکه‌های عصبی مصنوعی
81	شبکه عصبی پیش‌رو
82	شبکه عصبی بازخوردی
83	عملکرد شبکه‌های عصبی مصنوعی
83	یادگیری در شبکه‌های عصبی مصنوعی
84	الگوریتم بازگشت به عقب (Back Propagation)
84	شبکه‌های بیزین (BN)

85.....	ساخت یک شبکه بیزین
89.....	کاربردهای شبکه‌های عصبی
91.....	هوش مصنوعی - مسائل و معضلات
91.....	تهدید حریم خصوصی
91.....	تهدید کرامت انسانی
91.....	تهدید ایمنی
91.....	هوش مصنوعی - مجموعه اصطلاحات

فرهاد مهدی املانی

هوش مصنوعی - بررسی اجمالی

از زمان اختراع کامپیوتر، قدرت محاسباتی آن به صورت نمایی رشد کرده است. نه تنها انسان توانسته است کارایی کامپیوتر را در حوزه‌های مختلف افزایش دهد، بلکه سرعت رشد آن را نیز بالا برده و اندازه‌ی آن را کوچک‌تر کرده است.

هوش مصنوعی، یکی از شاخه‌های علوم کامپیوتر، تلاش دارد تا کامپیوترها و ماشین‌هایی هوشمند مشابه انسان بسازد.

هوش مصنوعی چیست؟

به گفته جان مک‌کارتی (پدر علم هوش مصنوعی)، هوش مصنوعی، "علم و مهندسی ساخت ماشین‌های هوشمند، به ویژه برنامه‌های کامپیوتری هوشمند" می‌باشد.

هوش مصنوعی روشی برای ساخت کامپیوتر، ربات یا نرم‌افزاری است که بتواند مانند انسان‌ها، به طور هوشمندانه فکر کند و عمل نماید.

هوش مصنوعی از طریق مطالعه نحوه فکر کردن مغز انسان و فرایندهای یادگیری، تصمیم‌گیری و حل مسئله توسط انسان‌ها، به دست می‌آید. سپس نتایج این مطالعات به عنوان پایه و اساسی برای توسعه نرم‌افزارها و سیستم‌های هوشمند مصنوعی مورد استفاده قرار می‌گیرد.

فلسفه‌ی هوش مصنوعی

هنگامی که انسان از قدرت سیستم‌های کامپیوتری بهره برد، یک پرسش مهم مطرح شد که آیا یک ماشین می‌تواند مانند انسان‌ها فکر کند و رفتار نماید؟ به همین دلیل، توسعه هوش مصنوعی با هدف ایجاد هوشی مشابه هوش انسانی که در انسان‌ها یافت می‌شود و ارزشمند است، آغاز گردید.

اهداف هوش مصنوعی

- ایجاد سیستم‌های خبره :سیستم‌هایی که قادرند رفتار هوشمندانه‌ای از خود بروز دهند، یاد بگیرند، الگوبرداری کنند، توضیح دهند و مشاوره‌ای به کاربران خود ارائه نمایند.
- پیاده‌سازی هوش انسانی در ماشین‌ها :ساخت سیستم‌هایی که قادر به درک مفاهیم، تفکر، یادگیری و رفتار مشابه انسان‌ها باشند.

چه چیزی به هوش مصنوعی کمک می‌کند؟

هوش مصنوعی یک علم میان‌رشته‌ای است که بر پایه رشته‌هایی همچون علوم کامپیوتر، زیست‌شناسی، روان‌شناسی، زبان‌شناسی، ریاضیات و مهندسی شکل گرفته است. یکی از اصول کلیدی هوش مصنوعی، توسعه توانایی‌های رایانه‌ای مرتبط با توانایی‌های شناختی انسان‌ها از قبیل استدلال، یادگیری و حل مسئله است.

بسیاری از حوزه‌های علمی می‌توانند در ساخت سیستم‌های هوشمند مشارکت داشته باشند، از جمله:

-علوم کامپیوتر: شامل مباحثی چون هوش مصنوعی، یادگیری ماشین، شبکه عصبی و الگوریتم‌نویسی.

-ریاضیات: شامل آمار، منطق، بهینه‌سازی و نظریه احتمالات.

-مهندسی: شامل مهندسی نرم‌افزار، مهندسی سخت‌افزار و مهندسی سیستم‌ها.

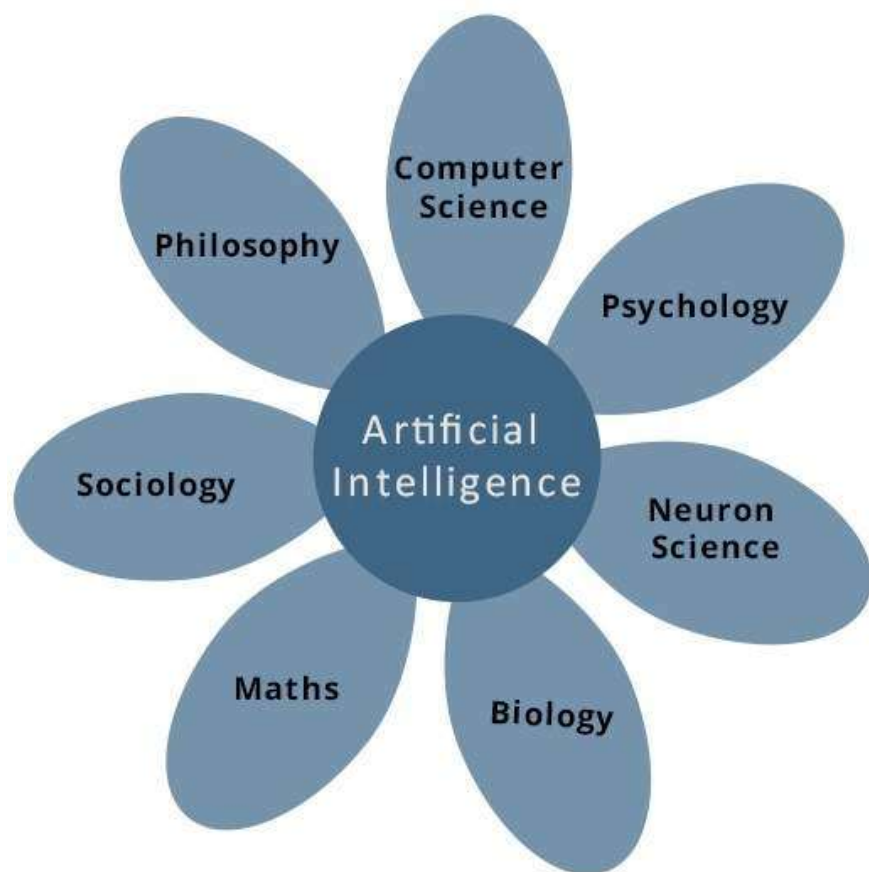
-زیست‌شناسی: شامل علوم اعصاب و مدل‌سازی سیستم عصبی.

-روان‌شناسی: شامل شناخت و فرایندهای ذهنی انسان.

-زبان‌شناسی: شامل پردازش زبان طبیعی و تحلیل متن.

-فلسفه: شامل منطق و استدلال.

بنابراین، یک رویکرد میان‌رشته‌ای برای ساخت سیستم‌های هوشمند بسیار مفید است.



برنامه‌نویسی با هوش مصنوعی و بدون هوش مصنوعی

برنامه‌نویسی بدون و با هوش مصنوعی به صورت زیر تفاوت دارند-

برنامه‌نویسی بدون هوش مصنوعی	برنامه‌نویسی با هوش مصنوعی
<p>یک برنامه کامپیوتری بدون هوش مصنوعی تنها قادر است به سوالات از پیش تعریف شده و مشخصی که در آن برنامه‌نویسی شده است، پاسخ دهد. این برنامه‌ها فاقد قابلیت یادگیری و تعمیم‌پذیری هستند و تنها می‌توانند بر اساس الگوریتم‌ها و قوانین تعریف شده از پیش به سوالات خاص و محدودی پاسخ دهند.</p>	<p>یک برنامه کامپیوتری مجهز به هوش مصنوعی، قادر است به طیف وسیع‌تری از سوالات و مسائل عمومی که در حیطه دانش آن قرار می‌گیرد، پاسخ دهد. این برنامه‌ها از طریق الگوریتم‌های یادگیری ماشینی و شبکه‌های عصبی، توانایی تعمیم‌پذیری و انطباق با سوالات و مسائل جدید را دارند و می‌توانند به طیف گسترده‌تری از سوالات عمومی که در حیطه موضوعی آموزش دیده‌اند، پاسخ دهند.</p>
<p>تغییر در یک برنامه کامپیوتری می‌تواند منجر به تغییر در ساختار و منطق درونی آن شود. زیرا برنامه‌های کامپیوتری اجزای مختلفی دارند که به هم وابسته هستند. تغییر در یک بخش می‌تواند نیاز به تغییر در بخش‌های دیگر را برای حفظ یکپارچگی و صحت عملکرد برنامه ایجاد کند. بنابراین تغییر یک جزء از برنامه بدون در نظر گرفتن تأثیر آن بر کلیت برنامه می‌تواند منجر به اشکال در ساختار و منطق برنامه شود.</p>	<p>برنامه‌های هوش مصنوعی معمولاً یکپارچه و به هم وابسته هستند و تغییر یک بخش می‌تواند بر کل سیستم تأثیر بگذارد. این برنامه‌ها اطلاعات جدید را به صورت تدریجی و با توجه به دانش قبلی خود جذب می‌کنند و یاد می‌گیرند. تغییر یک بخش کوچک می‌تواند باعث ایجاد خطا در عملکرد یا نتایج غیرمنتظره شود. بنابراین تغییرات باید با دقت و بر اساس درک کامل از ساختار و ارتباطات درونی برنامه انجام شود.</p>
<p>تغییر در یک برنامه کامپیوتری اغلب سریع و آسان نیست</p>	<p>تغییر در یک برنامه کامپیوتری اغلب سریع و آسان است.</p>

تکنیک‌های کلیدی در هوش مصنوعی چیست؟

در دنیای واقعی، دانش انسان دارای ویژگی‌هایی است که چالش‌برانگیز می‌باشند:

- حجم دانش بشر بسیار زیاد و درک آن دشوار است.
- دانش انسانی فاقد سازمان‌دهی و نظم مناسبی است.
- دانش به طور مداوم در حال تغییر و تحول است.

بنابراین، دانش انسان دارای ویژگی‌هایی است که استفاده مؤثر از آن با چالش مواجه می‌کند.

تکنیک هوش مصنوعی یک روش است برای سازماندهی و استفاده بهینه از دانش به گونه‌ای که:

- دانش باید به شکلی سازماندهی و نمایش داده شود که برای افراد قابل فهم باشد.
- دانش باید به راحتی قابلیت تغییر و به‌روزرسانی داشته باشد تا بتوان اشتباهات را اصلاح کرد.
- دانش باید حتی با وجود نقص‌ها یا عدم دقت کامل، در شرایط مختلف مفید واقع شود.

تکنیک‌هایی که هوش مصنوعی به آن مجهز شده است سرعت اجرای برنامه پیچیده را افزایش می‌دهند.

کاربردهای هوش مصنوعی

هوش مصنوعی در حوزه‌های مختلفی نظیر:

- **بازی‌ها - (Gaming):** یکی از کاربردهای هوش مصنوعی در حوزه بازی‌ها است. هوش مصنوعی نقش مهمی در بازی‌های استراتژیک مانند شطرنج، گو و تیک تاک تو دارد. در این بازی‌ها، ماشین‌های مجهز به هوش مصنوعی قادرند با استفاده از دانش هوریستیک، تعداد بسیار زیادی حالت و موقعیت ممکن را در نظر بگیرند و بهترین حرکت بعدی را انتخاب کنند.
- **پردازش زبان طبیعی - (Natural Language Processing):** یکی دیگر از کاربردهای هوش مصنوعی، پردازش زبان طبیعی است. پردازش زبان طبیعی به ماشین‌ها این قابلیت را می‌دهد که بتوانند زبان گفتاری و نوشتاری انسان را درک و

تفسیر کنند. با پردازش زبان طبیعی، امکان برقراری ارتباط و تعامل به زبان طبیعی انسان و ماشین فراهم می‌شود.

- **سامانه‌های خبره - (Expert Systems):** یکی دیگر از کاربردهای هوش مصنوعی، سامانه‌های خبره هستند. سامانه‌های خبره از ترکیب سخت‌افزار، نرم‌افزار و دانش تخصصی برای انجام استدلال و مشاوره استفاده می‌کنند. این سیستم‌ها می‌توانند توضیحات و مشاوره‌هایی مرتبط با حوزه تخصصی خود را به کاربر ارائه دهند و جایگزین متخصصان انسانی شوند.
- **سیستم‌های بینایی (Vision Systems):** یکی دیگر از حوزه‌های کاربردی هوش مصنوعی، سیستم‌های بینایی هستند. این سامانه‌ها قادرند ورودی‌های تصویری و بصری را دریافت و تفسیر کنند .

○ هواپیماهای جغرافیایی برای تهیه نقشه و استخراج اطلاعات فضایی، از تصاویر گرفته شده استفاده می‌کنند.

○ پزشکان برای تشخیص بیماری‌ها از سیستم‌های خبره بالینی و تصاویر پزشکی بهره می‌برند.

○ پلیس با استفاده از نرم‌افزارهای شناسایی چهره، مجرمان را شناسایی می‌کند.

- **تشخیص گفتار (Speech Recognition):** برخی از سامانه‌های هوشمند قادرند صدای انسان را در هنگام صحبت بشنوند و مفهوم جملات گفته شده را درک کنند. این سامانه‌ها می‌توانند با وجود موانعی مانند لهجه‌های متفاوت، اصطلاحات عامیانه، نویز پس‌زمینه و تغییرات صدا به دلیل شرایط محیطی، باز هم صحبت‌های انسان را تشخیص دهند.

- **تشخیص خط‌نویسی (Handwriting Recognition):** نرم‌افزارهای تشخیص نوشتار دستی قادرند متن نوشته شده به صورت دست‌نویس روی کاغذ یا صفحه نمایش با استفاده از قلم را تشخیص دهند. این نرم‌افزارها شکل حروف نوشته شده را تشخیص داده و آن را به متنی قابل ویرایش و پردازش تبدیل می‌کنند
- **ربات‌های هوشمند (Intelligent Robots):** ربات‌های مجهز به هوش

مصنوعی قادرند وظایفی را که توسط انسان تعریف شده انجام دهند. این ربات‌ها مجهز به سنسورهایی هستند که اطلاعاتی مانند نور، گرما، دما، حرکت، صدا، برخورد و فشار را از محیط فیزیکی دریافت می‌کنند. همچنین پردازنده‌های قدرتمند، حافظه

بالا و سنسورهای متعدد به آن‌ها قابلیت هوشمندی می‌دهد. این ربات‌ها می‌توانند از تجربیات خود یاد بگیرند و با محیط‌های جدید سازگار شوند.

تاریخچه هوش مصنوعی

در ادامه، تاریخچه هوش مصنوعی در طول قرن بیستم را مشاهده می‌کنید:

سال	بانک اطلاعات/نوآوری
۱۹۲۳	نمایشنامه "R.U.R" نوشته کارل چاپک در سال 1920 میلادی برای اولین بار در لندن به روی صحنه رفت و واژه "Robot" را به معنای امروزی آن به زبان انگلیسی معرفی کرد. این نمایشنامه داستان موجودات مصنوعی به نام ربات را روایت می‌کند که توسط انسان‌ها ساخته شده‌اند. پیش از این نمایشنامه، واژه ربات در زبان انگلیسی وجود نداشت.
۱۹۴۳	در دهه 1940 و 1950 میلادی، مدل‌های اولیه و پایه‌ای شبکه‌های عصبی مصنوعی ابداع و معرفی شدند. عبارت "پایه‌های شبکه‌های عصبی به وجود می‌آیند" به درستی بیان می‌کند که در این دوره، مفاهیم و الگوهای اولیه‌ای که بعدها منجر به توسعه شبکه‌های عصبی پیشرفته‌تر شدند، مطرح و مورد مطالعه قرار گرفتند.
۱۹۴۵	ایزاک آسیموف، نویسنده مشهور علمی-تخیلی و استاد دانشگاه بوستون، واژه "رباتیک" (robotics) را در دهه 1940 میلادی ابداع کرد. وی فارغ‌التحصیل رشته شیمی از دانشگاه کلمبیا بود و پس از آن به نویسندگی و تدریس علوم پرداخت. آسیموف واژه رباتیک را برای توصیف مطالعه علمی ربات‌ها ابداع کرد.
۱۹۵۰	-آلن تورینگ تست تورینگ را برای سنجش هوش ماشین معرفی کرد و کتاب "ماشین‌های محاسباتی و هوش مصنوعی" را منتشر نمود. -کلود مقاله‌ای درباره تجزیه و تحلیل دقیق بازی شطرنج به عنوان یک مسئله جستجو منتشر کرد. این دو مورد نشان‌دهنده پیشرفت‌های اولیه در زمینه هوش مصنوعی و الگوریتم‌های جستجو در آن دوران است.
۱۹۵۶	جان مک‌کارتی اصطلاح "هوش مصنوعی" را ابداع می‌کند. نمایش اولین برنامه‌ی عملی هوش مصنوعی در دانشگاه کارنگی ملون.
۱۹۵۸	جان مک‌کارتی زبان برنامه‌نویسی LISP را برای هوش مصنوعی ابداع می‌کند.

۱۹۶۴	پایان نامه‌ی دنی بوبرو در دانشگاه MIT نشان می‌دهد که کامپیوترها به اندازه‌ای زبان طبیعی را می‌فهمند که بتوانند مسائل جبری را به درستی حل کنند.
۱۹۶۵	جوزف ویزنباوم در MIT ربات تعاملی "الیزا" را ساخت که در انگلیسی به صورت دیالوگ می‌تواند گفت‌وگو کند.
۱۹۶۹	دانشمندان در موسسه تحقیقات استنفورد ربات "شیکی" را توسعه دادند که با حرکت، ادراک و حل مسئله تجهیز شده بود.
۱۹۷۳	گروه رباتیک موسسه دانشگاه ادینبورگ ربات مشهور اسکاتلندی "فردی" را ساختند که قادر به استفاده از بینایی برای یافتن و ترکیب مدل‌ها بود.
۱۹۷۹	اولین خودروی خودکار کنترل شده توسط کامپیوتر به نام "کارت استنفورد" ساخته می‌شود.
۱۹۸۵	هارولد کوهن برنامه‌ی نقاشی "آرون" را ایجاد و نمایش می‌دهد.
۱۹۹۰	پیشرفت‌های قابل توجهی در تمام حوزه‌های هوش مصنوعی - <ul style="list-style-type: none"> • نمونه‌برداری در یادگیری ماشین • استدلال بر پایه‌ی مورد • برنامه‌ریزی چندعاملی • زمانبندی • استخراج داده، پویاگری وب • فهم و ترجمه زبان طبیعی • بینایی، واقعیت مجازی • بازی‌ها
۱۹۹۷	برنامه شطرنج دیپ بلو بازیکن شطرنج معروف جهان، گری کاسپاروف، را شکست می‌دهد.
۲۰۰۰	ربات‌های خانگی تعاملی در دسترس تجاری قرار می‌گیرند. MIT رباتی به نام "کیسمت" که چهره‌ای دارد که احساسات را بیان می‌کند را نمایش می‌دهد. ربات "نومد" مناطق دورافتاده‌ای از قطب جنوب را بررسی می‌کند و شهاب‌سنگ‌ها را پیدا می‌کند.

هوش مصنوعی - سیستم‌های هوشمند

برای یادگیری هوش مصنوعی، باید بدانید که هوش مصنوعی چه معنایی دارد. این درس شامل مفهوم هوش، انواع و اجزا هوش است

هوش چیست؟

قابلیت یک سیستم برای محاسبه، استدلال، درک روابط و قیاس، یادگیری از تجربه، ذخیره و بازیابی اطلاعات از حافظه، حل مسائل، درک ایده‌های پیچیده، بهره‌گیری به طور روان از زبان طبیعی، طبقه‌بندی، تعمیم و سازگاری با شرایط جدید است.

انواع هوش

همانطور که هوارد گاردنر، روانشناس توسعه‌ای آمریکایی، توصیف می‌کند، هوش از چند بخش تشکیل شده است:

مثال	توضیحات	هوش
روایتگران، سخنگوها	قابلیت صحبت کردن، شناختن و استفاده از مکانیزم‌های آوایی (صداها، گفتاری)، نحو (دستور زبان) و معناشناسی (معنی).	هوش زبانی (Linguistic intelligence)
موسیقیدانان، خوانندگان، آهنگسازان	قابلیت ایجاد، ارتباط برقرار کردن و درک معانی ساخته‌شده از صدا، درک نت و ریتم.	هوش موسیقایی (Musical intelligence)
ریاضیدانان، دانشمندان	قابلیت استفاده و درک روابط در غیاب عمل یا اشیاء. درک ایده‌های پیچیده و غیر ملموس.	هوش منطقی-ریاضی (Logical-mathematical intelligence)
خوانندگان نقشه، فضانوردان، فیزیکدانان	قابلیت درک اطلاعات بصری یا فضایی، تغییر آن و بازسازی تصاویر بصری بدون ارجاع به اشیاء، ساخت تصاویر سه‌بعدی و حرکت و چرخش آن‌ها.	هوش فضایی (Spatial intelligence)

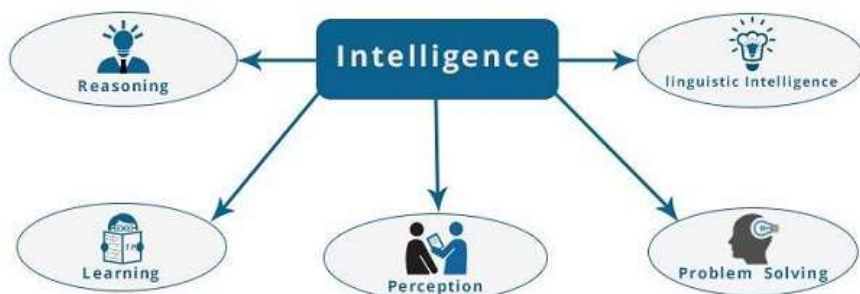
هوش بدنی-حرکتی (Bodily-Kinesthetic intelligence)	ورزشکاران و بازیکنان	قابلیت استفاده از تمام یا بخشی از بدن برای حل مسائل یا ساخت محصولات، کنترل مهارت‌های حرکتی دقیق و درشت، و کنترل اشیاء.
هوش درون-شخصی (Intra-personal intelligence)	خودشناسی	قابلیت تمییز بین احساسات، نیت‌ها و انگیزه‌های خود.
هوش بین-شخصی (Interpersonal intelligence)	مصاحبه‌گران	قابلیت تشخیص و تمایز بین احساسات، باورها و نیت‌های افراد دیگر.

می‌توانید یک ماشین یا یک سیستم را **هوش مصنوعی** بنامید وقتی که حداقل یکی و حداکثر همه انواع هوش در آن تجهیز شده باشد.

هوش از چه چیزهایی تشکیل شده است؟

هوش غیر قابل لمس است. آن شامل موارد زیر است:

- استدلال
- یادگیری
- حل مسئله
- درک
- هوش زبانی



بیایید به طور خلاصه به تمام اجزا بپردازیم -

- **استدلال** - مجموعه‌ای از فرآیندهایی است که به ما امکان می‌دهد پایه‌ای برای ارزیابی، تصمیم‌گیری و پیش‌بینی فراهم کنیم. به طور کلی دو نوع وجود دارد -

استدلال استقرائی (Inductive Reasoning)	استدلال استنباطی (Deductive Reasoning)
این نوع استدلال براساس مشاهدات خاص، اقدام به ارائه حکم عمومی می‌کند.	این نوع استدلال با یک حکم عمومی شروع شده و امکانات را بررسی می‌کند تا به یک نتیجه خاص و منطقی برسد.
حتی اگر تمامی فرضیات یک عبارت درست باشند، استدلال استقرائی امکان دارد نتیجه غلطی را بیان کند.	اگر یک مورد برای یک دسته از اشیاء به طور کلی درست باشد، همچنین برای تمام اعضای آن دسته درست است.
مثال - "نیتا معلم است. نیتا خودساخته است. بنابراین، تمام معلمان خودساخته هستند".	مثال - "تمام زنانی که سنشان بالای ۶۰ سال است، مادر بزرگ هستند. شالینی ۶۵ سال دارد. بنابراین، شالینی مادر بزرگ است".

- **یادگیری** - فعالیتی است که با مطالعه، تمرین، تدریس یا تجربه یک موضوع، دانش

یا مهارتی کسب می‌کند. یادگیری آگاهی افراد را درباره موضوعات مورد مطالعه افزایش می‌دهد.

توانایی یادگیری در انسان‌ها، برخی حیوانات و سامانه‌های هوشمند مصنوعی (هوش مصنوعی) وجود دارد. یادگیری به شکل زیر طبقه‌بندی می‌شود -

- یادگیری شنیداری - این نوع یادگیری به وسیله گوش دادن و شنیدن انجام می‌شود. به عنوان مثال، دانش‌آموزان گوش دادن به سخنرانی‌های صوتی ضبط شده را تجربه می‌کنند.
- یادگیری حادثه‌ای - به معنای یادگیری از خاطره توالی رویدادهایی است که فرد شاهد آن بوده یا تجربه کرده است. این نوع یادگیری خطی و منظم است.

- یادگیری حرکتی — این نوع یادگیری به وسیله حرکت دقیق عضلات انجام می‌شود. به عنوان مثال، برداشتن اشیاء، نوشتن و غیره.
- یادگیری مشاهده‌ای — به معنای یادگیری از طریق تماشا و تقلید از دیگران است. به عنوان مثال، کودک تلاش می‌کند با تقلید از پدر و مادرش یاد بگیرد.
- یادگیری ادراکی — به معنای یادگیری برای تشخیص محرک‌هایی است که قبلاً دیده شده است. به عنوان مثال، شناخت و طبقه‌بندی اشیاء و وضعیت‌ها.
- یادگیری رابطه‌ای — شامل یادگیری برای تمایز بین محرک‌های مختلف براساس خصوصیات ارتباطی، نه خصوصیات مطلق است. به عنوان مثال، اضافه کردن "کمی کمتر" نمک در زمان پخت کردن سیب‌زمینی که در دفعات گذشته شور شده‌اند، هنگام پخت با اضافه کردن به عنوان مثال یک قاشق غذاخوری نمک.
- یادگیری فضایی — به معنای یادگیری از طریق محرک‌های بصری مانند تصاویر، رنگ‌ها، نقشه‌ها و غیره است. به عنوان مثال، یک شخص می‌تواند قبل از واقعی دنبال کردن جاده، نقشه ذهنی ایجاد کند.
- یادگیری تحریک-واکنش — به معنای یادگیری برای انجام رفتار خاصی است که زمانی که یک تحریک خاص حاضر است، انجام می‌دهد. به عنوان مثال، یک سگ وقتی صدای زنگ در راهرو را می‌شنود، گوشش را بلند می‌کند.
- **حل مسئله** — فرآیندی است که فرد در آن از یک وضعیت فعلی با شناخت و تلاش برای رسیدن به یک راه‌حل مطلوب استفاده می‌کند، که توسط موانع شناخته شده یا ناشناخته مسدود شده است.
- حل مسئله شامل همچنین تصمیم‌گیری است که فرآیند انتخاب بهترین جایگزین ممکن از میان جایگزین‌های مختلف برای رسیدن به هدف موردنظر است.
- **دریافت** — این فرآیند هستی است که در آن اطلاعات حسی را به دست می‌آورد، تفسیر می‌کند، انتخاب می‌کند و اطلاعات حسی را به نحو مناسبی سازماندهی می‌دهد.
- فرض می‌شود که مفهوم دریافت حس کردن را شامل می‌شود. در انسان‌ها، دریافت با کمک اعضای حسی انجام می‌شود. در حوزه هوش مصنوعی، مکانیزم دریافت، اطلاعاتی که توسط حسگرها به دست می‌آید را به صورت معناداری یکجا می‌گذارد.

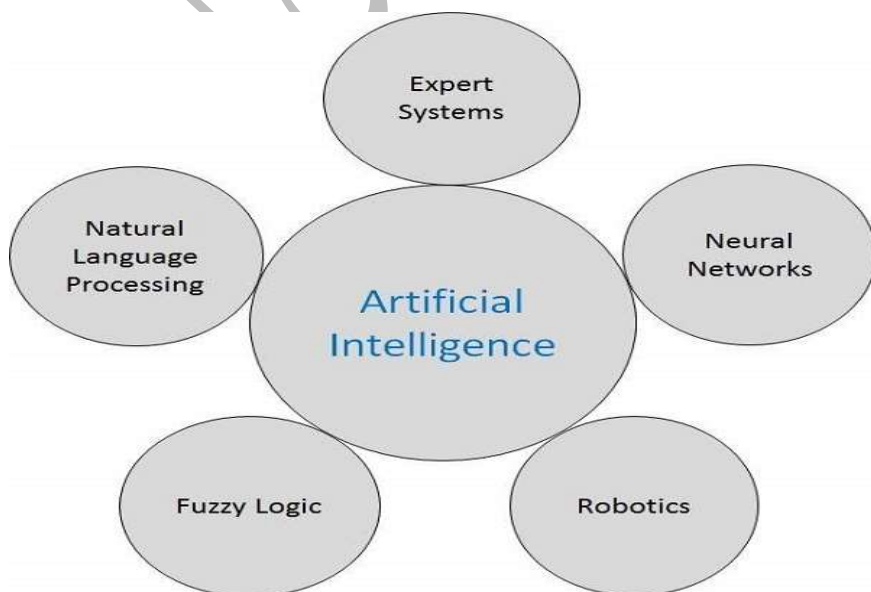
- **هوش زبانی** – این توانایی یک فرد است که از زبان گفتاری و نوشتاری استفاده کند، درک کند، صحبت کند و بنویسد. این در ارتباطات بین فردی مهم است.

تفاوت بین هوش انسان و هوش ماشین

- انسان‌ها از طریق الگوها درک می‌کنند، در حالی که ماشین‌ها از مجموعه قوانین و داده‌ها درک می‌کنند.
- انسان‌ها اطلاعات را با استفاده از الگوها ذخیره و بازخوانی می‌کنند، در حالی که ماشین‌ها این کار را با استفاده از الگوریتم‌های جستجو انجام می‌دهند. به عنوان مثال، به یاد آوردن، ذخیره کردن و بازخوانی کردن عدد ۴۰۴۰۴۰ راحت است زیرا الگوی آن ساده است.
- انسان‌ها می‌توانند شیء کامل را شناسایی کنند حتی اگر بخشی از آن مفقود یا به هم ریخته باشد، در حالی که ماشین‌ها نمی‌توانند به درستی این کار را انجام دهند.

هوش مصنوعی – حوزه‌های تحقیق

حوزه هوش مصنوعی بسیار وسیع است. در ادامه، ما به عناوین موضوعی و تحقیقات گسترده و موفق در حوزه هوش مصنوعی نگاهی خواهیم کرد –



یکی از حوزه ها در این زمینه، تشخیص گفتار و صدا

این دو عبارت در رباطیک، سیستم‌های خبره و پردازش زبان طبیعی، معمول هستند. اگرچه این دو عبارت به صورت قابل تعویض استفاده می‌شوند، اهداف آن‌ها متفاوت است.

تشخیص صدا	تشخیص گفتار
هدف تشخیص صدا شناسایی چه کسی است که در حال صحبت است.	تشخیص گفتار هدفش درک و فهمیدن چه گفته شده است.
برای شناسایی یک شخص با تجزیه و تحلیل تن، پیچش صدا و لهجه و غیره استفاده می‌شود.	در محاسبات بدون دخالت دست، نقشه‌کشی یا ناوبری منو استفاده می‌شود.
این سیستم تشخیص نیازمند آموزش است زیرا به شخص مرتبط است.	سیستم تشخیص گفتار نیاز به آموزش ندارد زیرا به صورت وابسته به سخنران نیست.
سیستم‌های تشخیص گفتار وابسته به سخنران نسبتاً آسان برای توسعه هستند.	توسعه سیستم‌های تشخیص گفتاری مستقل از سخنران سخت است.

عملکرد سیستم‌های تشخیص گفتار و صدا

ورودی کاربر که با صدای ضبط شده از میکروفون گرفته شده است، به کارت صدا سیستم می‌رود. تبدیل کننده سیگنال آنالوگ را به سیگنال دیجیتال معادل تبدیل می‌کند تا برای پردازش گفتار استفاده شود. برای شناسایی کلمات، پایگاه داده برای مقایسه الگوهای صدا استفاده می‌شود. در نهایت، بازخورد برعکس به پایگاه داده داده می‌شود.

این متن منبع زبان ورودی برای موتور ترجمه است که آن را به متن زبان مقصد تبدیل می‌کند. آن‌ها با رابط کاربری تعاملی، پایگاه داده بزرگ واژگان و غیره پشتیبانی می‌شوند.

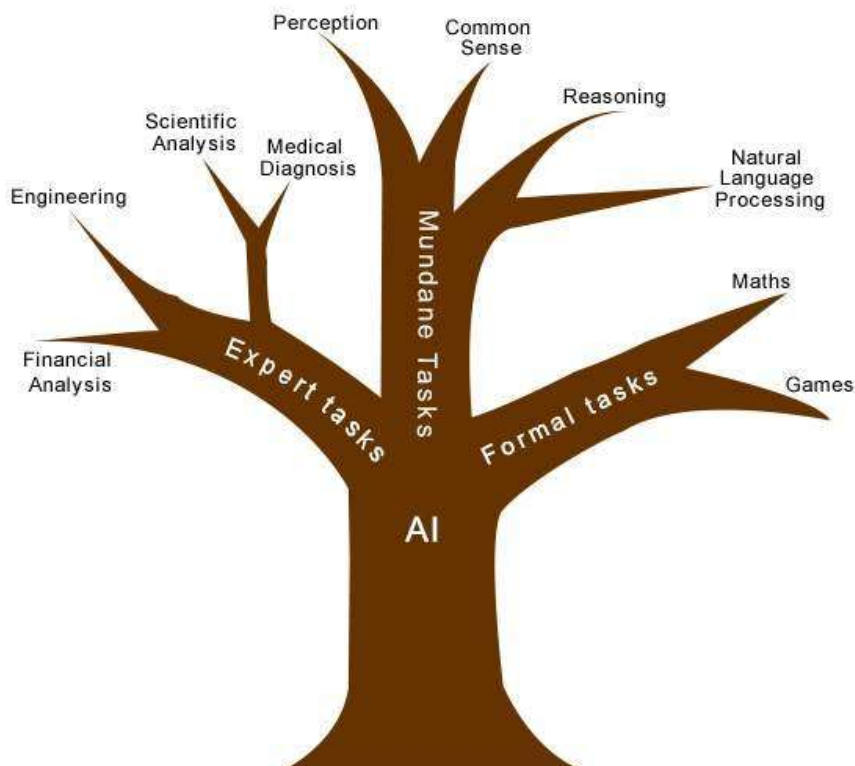
کاربردهای عملی حوزه‌های تحقیق

مجموعه گسترده‌ای از کاربردها وجود دارد که هوش مصنوعی در زندگی روزمره افراد عادی به آن‌ها خدمت می‌کند -

شماره	حوزه‌های تحقیق	کاربرد در زندگی واقعی
1	سیستم‌های خبره مثال‌ها - سیستم‌های ردیابی پرواز، سیستم‌های بالینی.	
2	پردازش زبان طبیعی مثال‌ها: ویژگی Google Now، تشخیص گفتار، خروجی صدای خودکار.	
3	شبکه‌های عصبی مثال‌ها - سیستم‌های تشخیص الگو مانند تشخیص چهره، تشخیص نوشتار، تشخیص خط‌نویسی.	
4	رباتیک مثال‌ها - ربات‌های صنعتی برای حرکت، پاشش، نقاشی، بررسی دقیق، حفاری، تمیزکاری، روکش، حک‌کاری، و غیره.	
5	سیستم‌های منطق فازی مثال‌ها - لوازم الکترونیک مصرفی، خودروها و غیره.	

طبقه‌بندی وظایف هوش مصنوعی

حوزه هوش مصنوعی به وظایف رسمی، وظایف عادی و وظایف خبره تقسیم می‌شود



حوزه‌های وظایف هوش مصنوعی		
وظایف عادی (معمولی)	وظایف رسمی	وظایف خبره
<ul style="list-style-type: none"> ادراک بینایی کامپیوتری 	<ul style="list-style-type: none"> ریاضیات هندسه 	<ul style="list-style-type: none"> مهندسی یافتن خطا

<ul style="list-style-type: none"> • گفتار ، صدا 	<ul style="list-style-type: none"> • منطق • انتگرال و مشتق 	<ul style="list-style-type: none"> • تولید • نظارت
<p>پردازش زبان طبیعی</p> <ul style="list-style-type: none"> • فهمیدن • تولید زبان • ترجمه زبان 	<p>بازی‌ها</p> <ul style="list-style-type: none"> • شطرنج (Deep Blue) • Go • Checkers 	<p>Natural Language Processing</p> <ul style="list-style-type: none"> • Understanding • Language Generation • Language Translation
احساس مشترک	تایید	تحلیل مالی
استدلال	اثبات قضیه	تشخیص پزشکی
برنامه‌ریزی		خلاقیت
<p>رباتیک</p> <ul style="list-style-type: none"> • حرکتی 		

انسان‌ها از تولدشان وظایف عادی را یاد می‌گیرند. آن‌ها وظایف عادی را با درک، گفتار، استفاده از زبان و حرکات یاد می‌گیرند. وظایف رسمی و وظایف خبره را بعداً و به همین ترتیب یاد می‌گیرند.

برای انسان‌ها، وظایف عادی به راحتی قابل یادگیری هستند. همین موضوع قبل از تلاش برای پیاده‌سازی وظایف عادی در ماشین‌ها نیز درست برخوردار بود. در ابتدا، تمام کارهای هوش مصنوعی در حوزه وظایف عادی متمرکز بود.

بعدها مشخص شد که ماشین برای انجام وظایف عادی نیاز به دانش بیشتری، نمایش دانش پیچیده و الگوریتم‌های پیچیده‌تر دارد. به همین دلیل کارهای هوش مصنوعی در حوزه وظایف خبره بهتر پیشرفت می‌کنند، زیرا حوزه وظایف خبره بدون عقل مشترک، نیاز به دانش خبره‌ای دارد که می‌تواند به راحتی نمایش داده و کنترل شود.

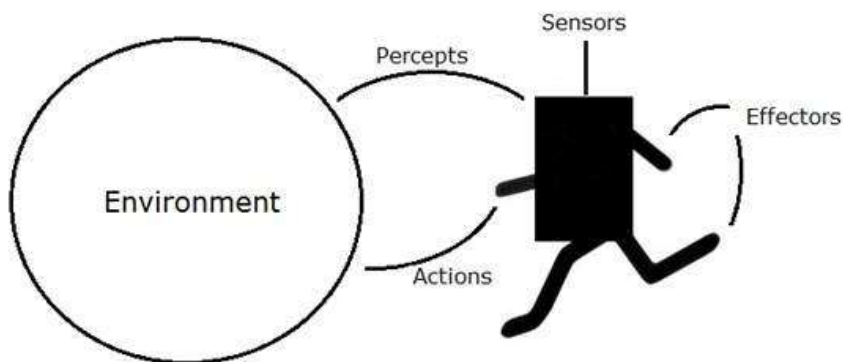
هوش مصنوعی - عوامل و محیط‌ها

یک سیستم هوش مصنوعی از یک عامل و محیط آن تشکیل شده است. عوامل در محیط خود عمل می‌کنند. محیط ممکن است حاوی عوامل دیگر نیز باشد.

عامل و محیط چیست؟

یک عامل هر چیزی است که می‌تواند محیط خود را از طریق سنسورها درک کند و بر آن محیط از طریق اثرگذارها عمل کند.

- یک عامل انسانی دارای اندام‌های حسی مانند چشم‌ها، گوش‌ها، بینی، زبان و پوست موازی با سنسورها و سایر اندام‌ها مانند دست‌ها، پاها، دهان، برای اثرگذارها است.
- یک عامل رباتیک دوربین‌ها و ردیاب‌های مادون قرمز را برای سنسورها و موتورهای مختلف و عملگرها برای اثرگذارها جایگزین می‌کند.
- یک عامل نرم افزاری رشته‌های بیت کدگذاری شده را به عنوان برنامه‌ها و اقدامات خود دارد.



اصطلاحات عامل

- **معیار عملکرد عامل** - این معیار است که تعیین می کند یک عامل چقدر موفق است.
- **رفتار عامل** - این عمل است که عامل پس از هر توالی ادراک داده شده انجام می دهد.
- **ادراک** - این ورودی های حسی عامل در یک زمان معین است.
- **توالی ادراک** - این تاریخچه همه چیزهایی است که یک عامل تا به امروز درک کرده است.
- **تابع عامل** - این یک نقشه از توالی ادراک به یک عمل است.

عقلانیت

عقلانیت چیزی جز وضعیت معقول بودن، منطقی بودن و داشتن حس قضاوت خوب نیست. عقلانیت با انتظارات عمل و نتایج بسته به آنچه عامل درک کرده است، مرتبط است. انجام اقدامات با هدف به دست آوردن اطلاعات مفید، بخشی مهم از عقلانیت است.

عامل ایده آل عقلانی چیست؟

یک عامل ایده آل عقلانی یکی است که قادر است اقدامات مورد انتظار را برای به حداکثر رساندن معیار عملکرد خود، بر اساس -

- توالی ادراک خود
- پایگاه دانش داخلی خود

عقلانیت یک عامل به موارد زیر بستگی دارد -

- معیارهای عملکرد
- توالی ادراک عامل تا کنون
- دانش قبلی عامل در مورد محیط
- اقداماتی که عامل می تواند انجام دهد

یک عامل عقلانی همیشه عمل درست را انجام می دهد، که به این معنی است که عمل باعث می شود عامل در توالی ادراک داده شده بیشترین موفقیت را داشته باشد. مشکلی که عامل حل می کند با $Actuators$ ، $Environment$ ، $Performance Measure$ و $Sensors$ (PEAS) مشخص می شود.

ساختار عوامل هوشمند

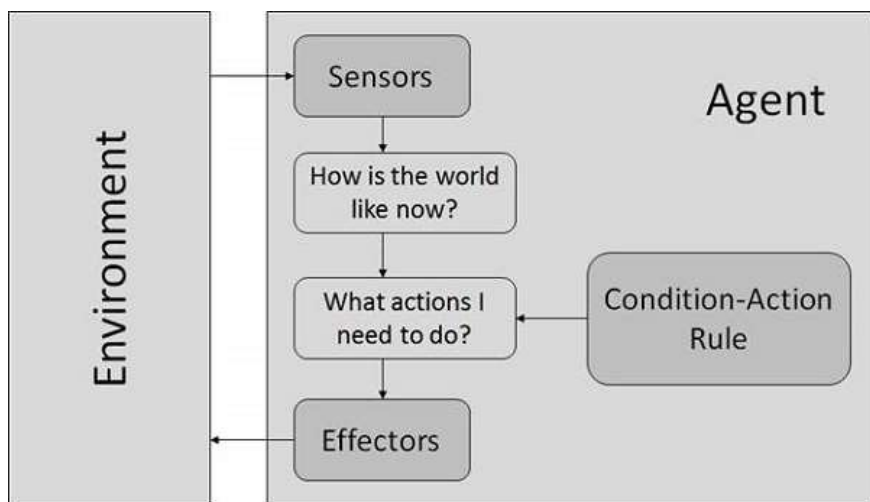
ساختار عامل را می توان به عنوان -

- عامل = معماری + برنامه عامل
- معماری = ماشینی که عامل روی آن اجرا می شود.
- برنامه عامل = پیاده سازی یک تابع عامل.

عوامل رفلکس ساده

- آنها فقط بر اساس ادراک فعلی عمل می کنند.
- آنها فقط عقلانی هستند اگر تصمیم درست فقط بر اساس فعل حال گرفته شود.
- محیط آنها کاملاً قابل مشاهده است.

قاعده شرط-عمل - این یک قاعده است که یک حالت (شرایط) را به یک عمل مرتبط می کند.



عوامل بازتاب مدل

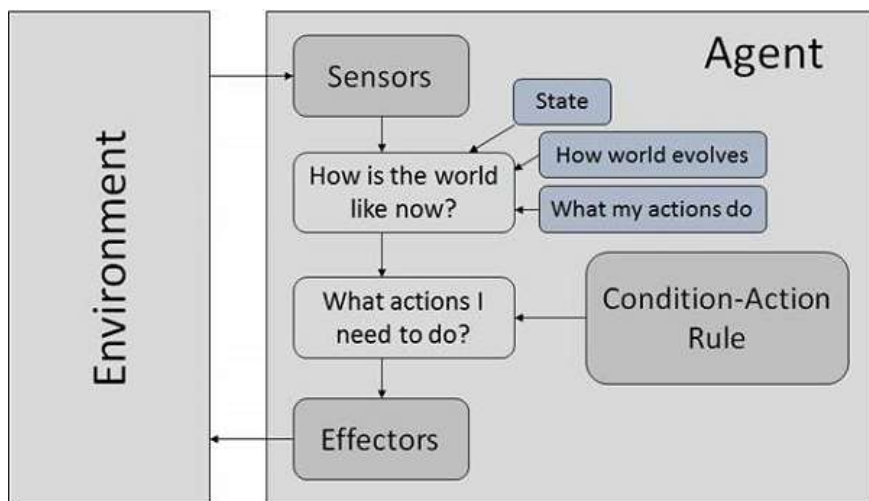
آنها از یک مدل جهان برای انتخاب عمل خود استفاده می کنند. آنها یک حالت داخلی را حفظ می کنند.

مدل - دانش در مورد "نحوه اتفاق افتادن چیزها در جهان."

وضعیت داخلی - این یک representation of unobserved aspects of current state است که به سابقه ادراک بستگی دارد.

به روزرسانی وضعیت به اطلاعات زیر نیاز دارد

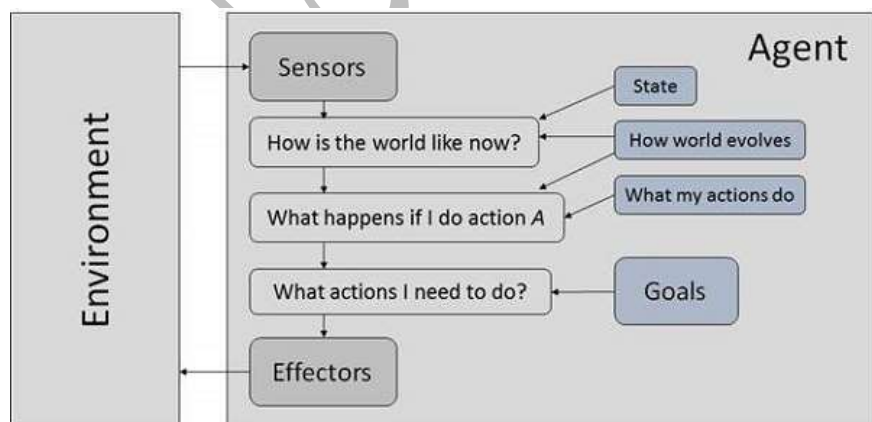
- نحوه تکامل جهان.
- نحوه تأثیر اقدامات عامل بر جهان.



عوامل مبتنی بر هدف

آنها عمل خود را برای رسیدن به اهداف انتخاب می کنند. رویکرد مبتنی بر هدف انعطاف پذیرتر از عامل رفلکس است زیرا دانشی که از تصمیم پشتیبانی می کند به صورت صریح مدل سازی می شود، که امکان تغییر آن را فراهم می کند.

هدف - این توصیف موقعیت های مطلوب است.

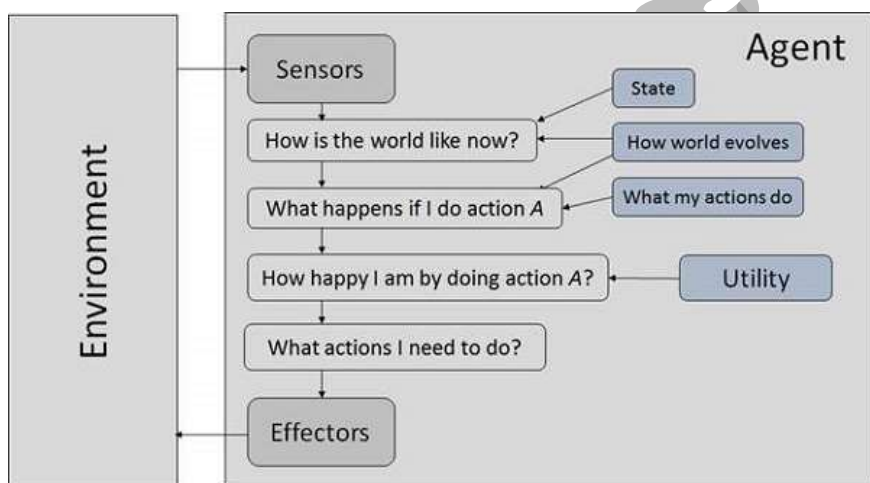


عوامل مبتنی بر سود

آنها بر اساس ترجیح (سود) برای هر حالت عمل می کنند.

اهداف ناکافی هستند زمانی که

- چند هدف متعارض وجود دارد که فقط می توان چند مورد از آنها را محقق کرد.
- اهداف دارای برخی عدم قطعیت در تحقق هستند و شما باید احتمال موفقیت را در برابر اهمیت هدف بسنجید.



طبیعت محیط ها

برخی برنامه ها در محیط کاملاً مصنوعی محدود به ورودی صفحه کلید ، پایگاه داده ، سیستم های فایل رایانه و خروجی کاراکتر روی صفحه نمایش عمل می کنند.

در مقابل ، برخی از نرم افزارهای عامل (روبات های نرم افزاری) در قلمروهای نرم افزاری غنی و نامحدود وجود دارند. شبیه ساز دارای یک محیط بسیار دقیق و پیچیده است. نرم افزار عامل باید از طیف گسترده ای از اقدامات در زمان واقعی انتخاب کند. یک نرم افزار نرم طراحی شده برای اسکن ترجیحات آنلاین مشتری و نشان دادن اقلام جالب به مشتری در محیط واقعی و همچنین مصنوعی کار می کند.

مشهورترین محیط مصنوعی محیط **آزمون تورینگ** است، در آن یک عامل واقعی و سایر عوامل مصنوعی در یک سطح مساوی آزمایش می شوند. این یک محیط بسیار چالش برانگیز است زیرا برای یک نرم افزار عامل بسیار دشوار است که به خوبی یک انسان عمل کند.

آزمون تورینگ

موفقیت رفتار هوشمندانه یک سیستم را می توان با آزمون تورینگ اندازه گیری کرد.

دو نفر و یک ماشین برای ارزیابی در آزمون شرکت می کنند. یکی از دو نفر نقش آزمایشگر را بازی می کند. هر یک از آنها در اتاق های مختلف نشسته اند. آزمایشگر نمی داند که چه کسی ماشین و چه کسی انسان است. او سوالات را تایپ می کند و آنها را به هر دو هوش می فرستد، که به او پاسخ های تایپ شده می دهد.

این آزمایش به هدف فریب دادن آزمایشگر است. اگر آزمایشگر نتواند پاسخ ماشین را از پاسخ انسان تشخیص دهد، آنگاه گفته می شود که ماشین هوشمند است.

ویژگی های محیط

محیط دارای چند ویژگی است -

- **متناوب / پیوسته**: اگر تعداد محدودی از حالات محیط به وضوح تعریف شده وجود داشته باشد، محیط گسسته است (به عنوان مثال شطرنج)؛ در غیر این صورت پیوسته است (به عنوان مثال رانندگی).
- **قابل مشاهده - partially observable**: اگر امکان تعیین وضعیت کامل محیط در هر نقطه زمانی از ادراکات آن وجود داشته باشد، قابل مشاهده است؛ در غیر این صورت فقط partially observable است.
- **static / dynamic**: اگر محیط در حالی که یک عامل عمل می کند تغییر نکند، static است؛ در غیر این صورت dynamic است.
- **عامل واحد / عوامل چندگانه**: محیط ممکن است حاوی عوامل دیگری باشد که ممکن است از نوع مشابه یا متفاوتی از عامل باشد.

- **قابل دسترسی / غیر قابل دسترسی** : اگر دستگاه حسی عامل می تواند به وضعیت کامل محیط دسترسی داشته باشد ، آنگاه محیط برای آن عامل قابل دسترسی است.
- **deterministic / non-deterministic** : اگر حالت بعدی محیط به طور کامل توسط وضعیت فعلی و اعمال عامل تعیین شود ، محیط deterministic است ؛ در غیر این صورت non-deterministic است.
- **episodic / non-episodic** : در یک محیط episodic ، هر قسمت شامل ادراک و سپس عمل عامل است. کیفیت عمل آن فقط به خود قسمت بستگی دارد. قسمت های بعدی به اقدامات قسمت های قبلی بستگی ندارند. محیط های episodic بسیار ساده تر هستند زیرا عامل نیازی به فکر کردن به جلو ندارد.

هوش مصنوعی - الگوریتم های جستجوی محبوب

جستجو، تکنیک جهانی حل مسئله در هوش مصنوعی است. برخی از بازی های تک نفره مانند بازی های رومیزی، سودوکو، جدول کلمات متقاطع و غیره وجود دارند. الگوریتم های جستجو به شما کمک می کنند تا در چنین بازی ها برای یک موقعیت خاص جستجو کنید.

مشکلات مسیریابی عامل واحد

بازی هایی مانند 3X3 هشت تایی، 4X4 پانزده تایی و 5X5 بیست و چهار تایی پازل های تک عامل مسیریابی هستند. آنها از یک ماتریس کاشی با یک کاشی خالی تشکیل شده اند. بازیکن باید کاشی ها را با کشیدن یک کاشی به صورت عمودی یا افقی به یک فضای خالی با هدف انجام برخی از اهداف، مرتب کند.

سایر مثال های مشکلات مسیریابی عامل واحد عبارتند از: مسئله فروشنده دوره گرد، مکعب روبیک و اثبات قضیه.

اصطلاحات جستجوی

- **مورد مسئله** - شامل حالت اولیه و حالت هدف است.
- **گرافیک فضای مسئله** - نشان دهنده وضعیت مسئله است. حالات با گره ها نشان داده می شوند و عملگرها با لبه ها نشان داده می شوند.

- **عمق یک مسئله** - طول کوتاهترین مسیر یا کوتاهترین دنباله ای از عملگرها از حالت اولیه تا حالت هدف است.
- **پیچیدگی فضایی** - حداکثر تعداد گره هایی که در حافظه ذخیره می شوند.
- **پیچیدگی زمانی** - حداکثر تعداد گره هایی که ایجاد می شوند.
- **قابل قبول بودن** - خاصیت یک الگوریتم برای همیشه یافتن یک راه حل بهینه است.
- **عامل شاخه بندی** - میانگین تعداد گره های فرزند در گرافیک فضای مسئله.
- **عمق** - طول کوتاهترین مسیر از حالت اولیه تا حالت هدف.

استراتژی های جستجوی brute-force

آنها ساده ترین هستند، زیرا به هیچ دانش دامنه خاصی نیاز ندارند. آنها با تعداد کمی از حالات ممکن کار می کنند.

نیازهای -

- توضیحات حالت
- مجموعه ای از عملگرهای معتبر
- حالت اولیه
- توضیحات حالت هدف

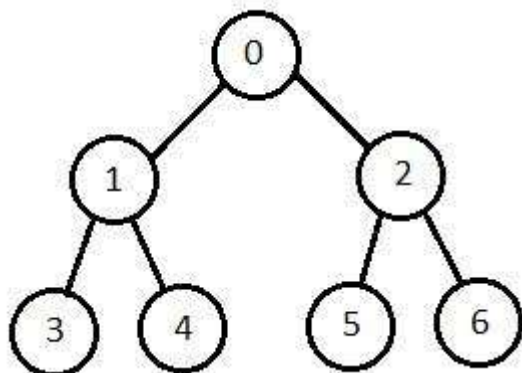
جستجوی پهنه ای

از ریشه گره شروع می شود، گره های مجاور را اول بررسی می کند و به سمت گره های مجاور سطح بعدی حرکت می کند. این یک درخت را در هر بار تا زمانی که راه حل پیدا شود، تولید می کند. می توان آن را با استفاده از ساختار داده صف FIFO پیاده سازی کرد. این روش کوتاهترین مسیر را به راه حل ارائه می دهد.

اگر عامل شاخه بندی (تعداد متوسط فرزند برای یک گره داده شده) b و عمق d ، پس تعداد گره ها در سطح d b^d است.

کل تعداد گره هایی که در بدترین حالت ایجاد می شوند $b + b^2 + b^3 + \dots + b^d$ است.

معایب -از آنجایی که هر سطح از گره ها برای ایجاد مرحله بعدی ذخیره می شود، فضای حافظه زیادی را مصرف می کند. فضای مورد نیاز برای ذخیره گره ها نامایی است. پیچیدگی آن به تعداد گره ها بستگی دارد. می تواند گره های تکراری را بررسی کند.

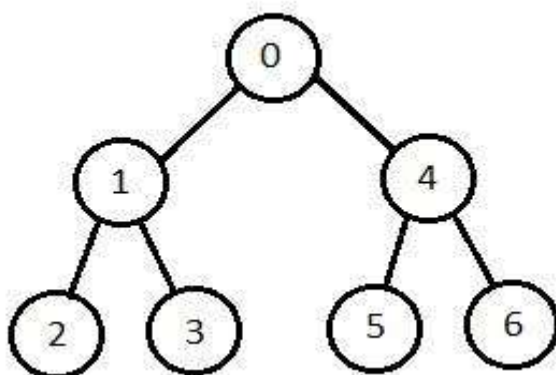


جستجوی عمق-اول

با استفاده از ساختار داده پشته LIFO با بازگشت پیاده سازی می شود. این روش همان مجموعه گره ها را مانند روش Breadth-First ایجاد می کند، فقط به ترتیب متفاوتی . از آنجایی که گره ها در یک مسیر منفرد در هر تکرار از ریشه تا گره برگ ذخیره می شوند، فضای مورد نیاز برای ذخیره گره ها خطی است. با عامل شاخه ای b و عمق به عنوان m ، فضای ذخیره سازی bm است .

معایب -این الگوریتم ممکن است خاتمه نیابد و در یک مسیر به طور نامحدود ادامه دهد. راه حل این مشکل این است که یک عمق برش را انتخاب کنیم. اگر عمق برش ایده آل d باشد، و اگر عمق برش انتخاب شده کمتر از d باشد، ممکن است این الگوریتم شکست بخورد. اگر عمق برش انتخاب شده بیشتر از d باشد، زمان اجرای آن افزایش می یابد.

پیچیدگی آن به تعداد مسیرها بستگی دارد. نمی تواند گره های تکراری را بررسی کند.



جستجوی دوطرفه

از حالت اولیه به جلو و از حالت هدف به عقب جستجو می‌کند تا زمانی که هر دو با هم ملاقات کنند و یک حالت مشترک را شناسایی کنند.

مسیر از حالت اولیه با مسیر معکوس از حالت هدف پیوند داده می‌شود. هر جستجو فقط تا نیمی از مسیر total انجام می‌شود.

جستجوی هزینه یکنواخت

مرتب‌سازی بر اساس هزینه increasing مسیر به یک گره انجام می‌شود. این الگوریتم همیشه گره با کمترین هزینه را گسترش می‌دهد. این الگوریتم با جستجوی

Breadth-First یکسان است اگر هر گذار هزینه یکسانی داشته باشد.

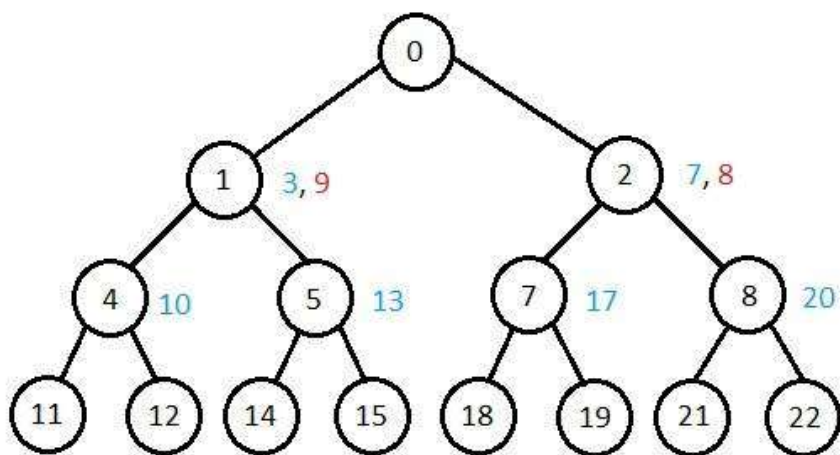
مسیرها را به ترتیب increasing هزینه بررسی می‌کند.

معایب — ممکن است چندین مسیر طولانی با هزینه $C \leq$ وجود داشته باشد. جستجوی هزینه یکنواخت باید همه آنها را بررسی کند.

جستجوی عمق-اول تعمیق تکراری

جستجوی عمق-اول را تا سطح 1 انجام می‌دهد، دوباره شروع می‌کند، یک جستجوی کامل عمق-اول را تا سطح 2 انجام می‌دهد و به همین ترتیب ادامه می‌دهد تا زمانی که راه حل پیدا شود.

گره‌ای ایجاد نمی‌کند تا زمانی که همه گره‌های پایین‌تر ایجاد شده باشند. فقط یک پشته از گره‌ها را ذخیره می‌کند. الگوریتم زمانی به پایان می‌رسد که یک راه حل در عمق d پیدا کند. تعداد گره‌های ایجاد شده در عمق b^d و در عمق $b^{d-1} d-1$ است.



مقایسه پیچیدگی الگوریتم‌های مختلف

بیا باید عملکرد الگوریتم‌ها را بر اساس معیارهای مختلف ببینیم-

معیار	جستجوی عرضی	جستجوی عمقی	جستجوی دوطرفه	جستجوی هزینه یکنواخت	جستجوی عمقی تعمیق تکراری
زمان	b^d	b^m	$b^{d/2}$	b^d	b^d
فضا	b^d	b^m	$b^{d/2}$	b^d	b^d

بهینه بودن	بله	خیر	بله	بله	بله
کامل بودن	بله	خیر	بله	بله	بله

استراتژی‌های جستجوی آگاهانه (هوشیار)

برای حل مسائل بزرگ با تعداد زیادی حالت ممکن، باید دانش خاص مسئله را اضافه کرد تا کارایی الگوریتم‌های جستجو افزایش یابد.

توابع ارزیابی هیستریک

آنها هزینه مسیر بهینه بین دو حالت را محاسبه می‌کنند. یک تابع هیستریک برای بازی‌های کاشی‌های لغزان با شمارش تعداد حرکاتی که هر کاشی از حالت هدف خود انجام می‌دهد و افزودن این تعداد حرکت برای همه کاشی‌ها محاسبه می‌شود.

جستجوی هیستریک خالص

گره‌ها را به ترتیب مقادیر هیستریک آنها گسترش می‌دهد. دو لیست ایجاد می‌کند، یک لیست بسته برای گره‌های قبلاً گسترش یافته و یک لیست باز برای گره‌های ایجاد شده اما گسترش نیافته.

در هر تکرار، یک گره با حداقل مقدار هیستریک گسترش می‌یابد، همه گره‌های فرزند آن ایجاد می‌شوند و در لیست بسته قرار می‌گیرند. سپس، تابع هیستریک به گره‌های فرزند اعمال می‌شود و آنها را بر اساس مقدار هیستریک آنها در لیست باز قرار می‌دهد. مسیرهای کوتاه‌تر ذخیره می‌شوند و مسیرهای طولانی‌تر حذف می‌شوند.

جستجوی A^*

مشهورترین شکل جستجوی بهترین اول است. از گسترش مسیرهایی که قبلاً گران هستند اجتناب می‌کند، اما اولین مسیرهای امیدوارکننده را گسترش می‌دهد.

$$f(n) = g(n) + h(n), \text{ که در آن}$$

- $g(n)$ هزینه (تاکنون) برای رسیدن به گره
- $h(n)$ هزینه تخمینی برای رسیدن از گره به هدف
- $f(n)$ هزینه تخمینی کل مسیر از n به هدف. با افزایش $f(n)$ پیاده‌سازی شده است.

جستجوی بهترین اول حریصانه

گره‌ای را که تخمین زده می‌شود نزدیک‌ترین به هدف است گسترش می‌دهد. گره‌ها را بر اساس $f(n) = h(n)$ گسترش می‌دهد. با استفاده از صف اولویت پیاده‌سازی شده است.

معایب - ممکن است در حلقه‌ها گیر کند. بهینه نیست.

الگوریتم‌های جستجوی محلی

آنها از یک راه حل احتمالی شروع می‌کنند و سپس به یک راه حل همسایه حرکت می‌کنند. حتی اگر در هر زمان قبل از پایان کار قطع شوند، می‌توانند یک راه حل معتبر برگردانند.

جستجوی صعودی

این یک الگوریتم تکراری است که با یک راه حل دلخواه برای یک مسئله شروع می‌کند و سعی می‌کند با تغییر تدریجی یک عنصر از راه حل، یک راه حل بهتر پیدا کند. اگر تغییر، یک راه حل بهتر تولید کند، یک تغییر تدریجی به عنوان یک راه حل جدید گرفته می‌شود. این فرآیند تا زمانی که بهبودی بیشتر وجود نداشته باشد تکرار می‌شود.

تابع کوه‌نوردی (Hill-Climbing)، یک state که یک ماکزیمم محلی است را

باز می‌گرداند.


```

inputs: problem, a problem
local variables: current, a node
                  neighbor, a node
current <- Make_Node(Initial-State[problem])
loop
  do neighbor <- a highest_valued successor of
current
    if Value[neighbor] ≤ Value[current] then
      return State[current]
    current <-
neighbor
end

```

معایب - این الگوریتم کامل یا بهینه نیست.

جستجوی پرتو محلی

در این الگوریتم، در هر زمان k تعداد حالت وجود دارد. در ابتدا، این حالت‌ها به صورت تصادفی تولید می‌شوند. جانشینان این k حالت با کمک تابع هدف محاسبه می‌شوند. اگر هر یک از این جانشین‌ها حداکثر مقدار تابع هدف باشد، الگوریتم متوقف می‌شود.

در غیر این صورت، (k حالت اولیه و k تعداد جانشین حالت $= 2k$) حالت در یک لیست قرار می‌گیرند. سپس لیست به صورت عددی مرتب می‌شود k . حالت برتر به عنوان حالت‌های اولیه جدید انتخاب می‌شوند. این فرآیند تا زمانی که مقدار حداکثری حاصل شود ادامه می‌یابد.

(تابع BeamSearch (مشکل، k) یک حالت راه حل را برمی‌گرداند .

start with k randomly generated states

loop

generate all successors of all k states

if any of the states = solution, then return the state

else select the k best successors

end

آنیلینگ شبیه‌سازی شده

آنیلینگ فرایند گرم کردن و سرد کردن یک فلز برای تغییر ساختار داخلی آن و اصلاح خواص فیزیکی آن است. هنگامی که فلز سرد می‌شود، ساختار جدید آن تثبیت می‌شود و فلز خواص جدید خود را حفظ می‌کند. در فرایند آنیلینگ شبیه‌سازی شده، دما متغیر است.

ما ابتدا دما را بالا می‌گذاریم و سپس اجازه می‌دهیم به تدریج «خنک» شود، زیرا الگوریتم پیش می‌رود. هنگامی که دما بالا است، الگوریتم مجاز است راه‌حل‌های بدتر را با بسامد بالا قبول کند.

شروع

- $k = 0$ را مقداردهی اولیه کنید؛ $L =$ عدد صحیح تعداد متغیرها؛
- از $i \rightarrow 1$ ، تفاوت عملکرد Δ را جستجو کنید.
- اگر $\Delta \leq 0$ باشد، قبول کنید در غیر این صورت
- اگر $\exp(-\Delta/T(k)) > \text{random}(0,1)$ باشد، قبول کنید؛
- مراحل 1 و 2 را برای $L(k)$ مرحله تکرار کنید.
- $k = k + 1$;

مراحل 1 تا 4 را تا زمانی که معیارها برآورده شوند، تکرار کنید.

پایان

مسئله فروشنده دوره گرد

در این الگوریتم، هدف یافتن یک تور کم هزینه است که از یک شهر شروع می شود، دقیقاً یک بار از همه شهرها در مسیر بازدید می کند و در همان شهر شروع به کار می کند.

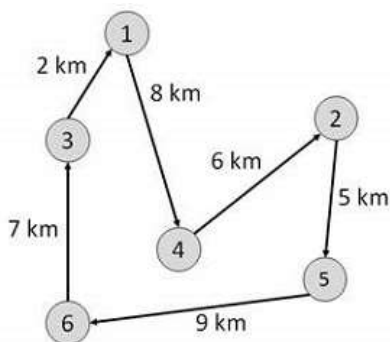
Start

Find out all $(n-1)!$ Possible solutions, where n is the total number of cities.

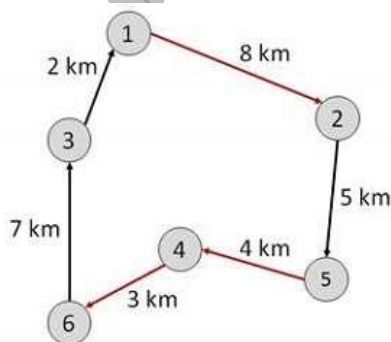
Determine the minimum cost by finding out the cost of each of these $(n-1)!$ solutions.

Finally, keep the one with the minimum cost.

end



Total Distance = 37km



Total Distance = 31km

هوش مصنوعی - سیستم های منطق فازی

سیستم های منطق فازی (FLS) در پاسخ به ورودی ناقص، مبهم، تحریف شده یا غیر دقیق (مبهم) خروجی قابل قبول اما قطعی تولید می کنند .

منطق فازی چیست؟

منطق فازی (FL) روشی برای استدلال است که شبیه استدلال انسانی است. رویکرد FL روش تصمیم‌گیری در انسان را که شامل تمام امکانات میانی بین مقادیر دیجیتالی بله و خیر است، تقلید می‌کند.

بلوک منطقی معمولی که یک کامپیوتر می‌تواند درک کند، ورودی دقیقی را می‌گیرد و خروجی قطعی به عنوان TRUE یا FALSE تولید می‌کند، که معادل بله یا خیر انسان است.

لفظی زاده، مخترع منطق فازی، مشاهده کرد که برخلاف رایانه‌ها، تصمیم‌گیری انسانی طیف وسیعی از امکانات بین بله و خیر را شامل می‌شود، مانند:

قطعاً بله
ممکن است بله
نمی‌توان گفت
ممکن است نه
قطعاً نه

منطق فازی بر سطوح امکانات ورودی برای دستیابی به خروجی قطعی کار می‌کند.

پیاده‌سازی

- می‌توان آن را در سیستم‌هایی با اندازه‌ها و قابلیت‌های مختلف، از میکروکنترلرهای کوچک تا سیستم‌های کنترلی بزرگ، شبکه‌ای و مبتنی بر ایستگاه کاری پیاده‌سازی کرد.
- می‌توان آن را در سخت‌افزار، نرم‌افزار یا ترکیبی از هر دو پیاده‌سازی کرد.

چرا منطق فازی؟

منطق فازی برای اهداف تجاری و عملی مفید است.

- می‌تواند ماشین‌ها و محصولات مصرفی را کنترل کند.
- ممکن است استدلال دقیقی ارائه ندهد، اما استدلال قابل قبولی ارائه دهد.
- منطق فازی به مقابله با عدم قطعیت در مهندسی کمک می‌کند.

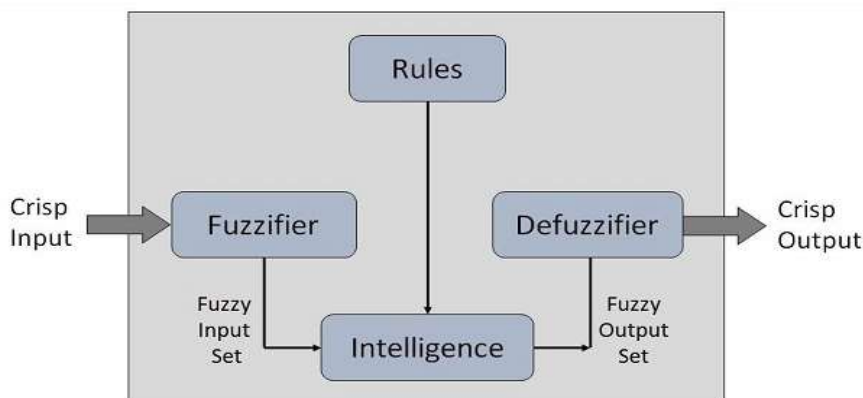
معماری سیستم‌های منطق فازی

این سیستم دارای چهار بخش اصلی است که در زیر نشان داده شده است:

- مدول — fuzzification این سیستم ورودی‌های سیستم را که اعداد واضح هستند، به مجموعه‌های فازی تبدیل می‌کند. این سیگنال ورودی را به پنج مرحله تقسیم می‌کند، مانند:

LP	مقدار بزرگ مثبت است X
MP	مقدار متوسط مثبت است X
S	مقدار کوچک است X
MN	مقدار متوسط منفی است X
LN	مقدار بزرگ منفی است X

- پایگاه دانش - مجموعه قوانین IF-THEN ارائه شده توسط کارشناسان را ذخیره می‌کند.
- موتور استنتاج - فرایند استدلال انسانی را با انجام استنتاج فازی بر روی ورودی‌ها و قوانین IF-THEN شبیه سازی می‌کند.
- ماژول دفریووز - مجموعه فازی حاصل از موتور استنتاج را به یک مقدار دقیق تبدیل می‌کند.



تابع عضویت بر روی مجموعه های فازی متغیرها کار می کند .

تابع عضویت به شما امکان کمی سازی اصطلاحات زبانی و نمایش گرافیکی یک مجموعه فازی را می دهد. یک تابع عضویت برای یک مجموعه فازی A در جهان گفتمان X به عنوان $\mu_A: X \rightarrow [0,1]$ تعریف می شود.

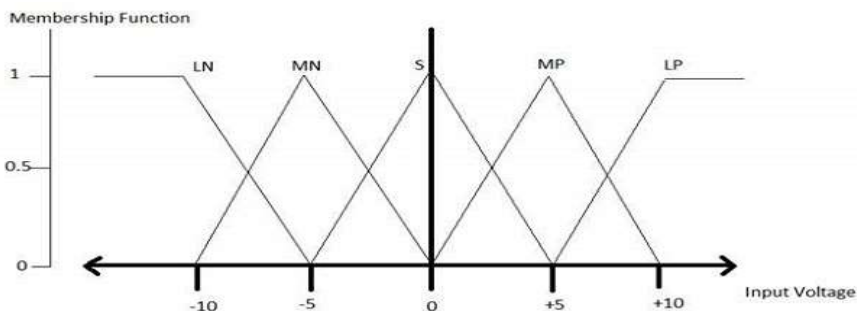
در اینجا، هر عنصر x به یک مقدار بین 0 و 1 نقشه می شود. این را مقدار عضویت یا درجه عضویت می نامند. این میزان عضویت عنصر در X در مجموعه فازی A را کمی می کند.

- محور x جهان گفتمان را نشان می دهد.

- محور y درجات عضویت در بازه $[0, 1]$ را نشان می دهد.

می توان از چندین تابع عضویت برای فازی کردن یک مقدار عددی استفاده کرد. از توابع پیچیده استفاده نمی شود زیرا استفاده از توابع پیچیده دقت بیشتری در خروجی ایجاد نمی کند.

همه توابع عضویت برای **LN**، **MP**، **S**، **MN** و **LP** به شرح زیر نشان داده شده است -

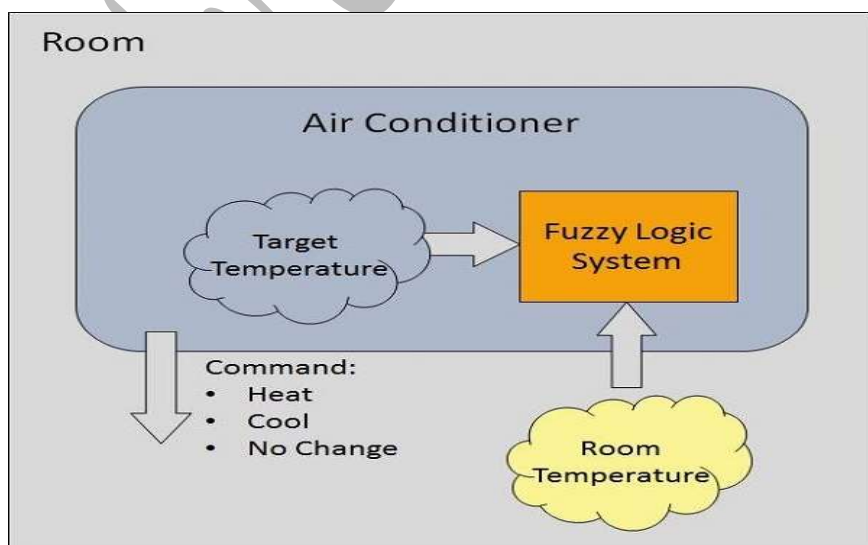


شکل های تابع عضویت مثلی در میان اشکال مختلف تابع عضویت مانند دوزنقه ای، تکی و گاوسی رایج ترین هستند.

در اینجا، ورودی به 5- سطح فازی کننده از 10- ولت تا 10+ ولت متغیر است. بنابراین خروجی مربوطه نیز تغییر می کند.

مثالی از یک سیستم منطق فازی

فرض کنید یک سیستم تهویه مطبوع با سیستم منطق فازی 5 سطحی داریم. این سیستم دمای تهویه مطبوع را با مقایسه دمای اتاق و مقدار دمای هدف تنظیم می کند.



الگوریتم

- تعریف متغیرها و اصطلاحات زبانی (start)
- ساختن توابع عضویت برای آنها (start).
- ساختن پایگاه دانش قوانین (start)
- تبدیل داده های دقیق به مجموعه های فازی با استفاده از توابع عضویت. (فازی سازی)
- ارزیابی قوانین در پایگاه قوانین. (موتور استنتاج)
- ترکیب نتایج هر قانون. (موتور استنتاج)
- تبدیل داده های خروجی به مقادیر غیر فازی. (دفرفیوز)

توسعه

مرحله 1 - تعریف متغیرها و اصطلاحات زبانی

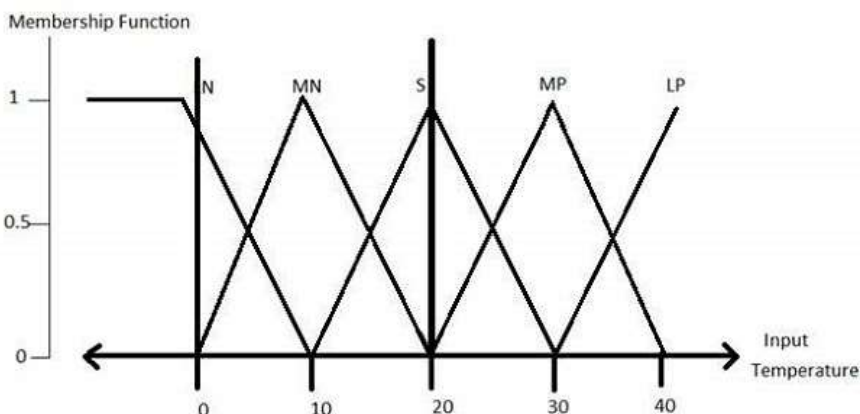
متغیرهای زبانی ورودی و خروجی هستند که به صورت کلمات یا جملات ساده هستند. برای دمای اتاق، سرد، گرم، داغ و غیره، اصطلاحات زبانی هستند.

دمای (t) = {بسیار سرد، سرد، گرم، بسیار گرم، داغ}

هر عضو این مجموعه یک اصطلاح زبانی است و می تواند بخشی از مقادیر کلی دما را پوشش دهد.

مرحله 2 - ساختن توابع عضویت برای آنها

تابع عضویت متغیر دما به صورت زیر نشان داده شده است -



مرحله 3 - ساختن قوانین پایگاه دانش

یک ماتریس از مقادیر دمای اتاق در مقابل مقادیر هدف دما ایجاد کنید که انتظار می رود یک سیستم تهویه مطبوع ارائه دهد.

بسیار گرم	داغ	گرم	سرد	بسیار سرد	دمای اتاق / هدف
گرم	گرم	گرم	گرم	بدون تغییر	بسیار سرد
گرم	گرم	گرم	بدون تغییر	خنک	سرد
گرم	گرم	بدون تغییر	خنک	خنک	گرم
گرم	بدون تغییر	خنک	خنک	خنک	داغ
بدون تغییر	خنک	خنک	خنک	خنک	بسیار گرم

یک مجموعه قوانین را به پایگاه دانش به صورت ساختارهای IF-THEN-ELSE بسازید.

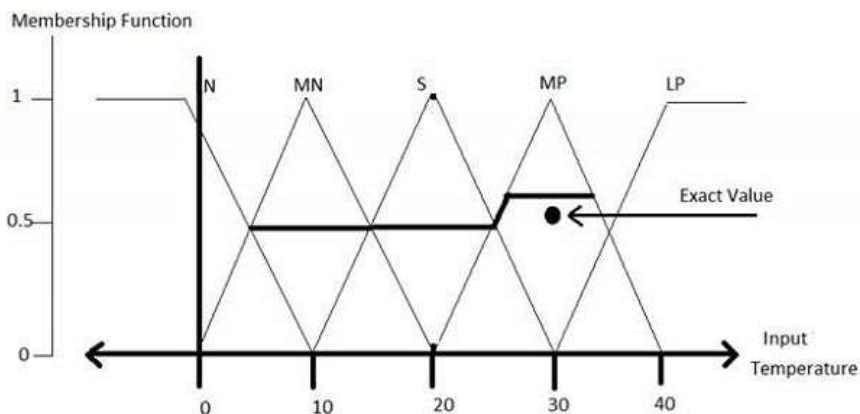
عمل	شرایط	شماره ردیف
گرم	اگر دما=(سرد یا بسیار سرد) AND هدف=گرم THEN	1
خنک	اگر دما=(داغ یا بسیار گرم) AND هدف=گرم THEN	2
No_Change	اگر (دما = گرم) AND هدف = گرم THEN)	3

مرحله 4 - مقدار فازی را بدست آورید

عملیات مجموعه‌های فازی ارزیابی قوانین را انجام می‌دهند. عملیات‌های مورد استفاده برای OR و AND به ترتیب Max و Min هستند. تمام نتایج ارزیابی را برای تشکیل یک نتیجه نهایی ترکیب کنید. این نتیجه یک مقدار فازی است.

مرحله 5 - دوددهی را انجام دهید

دوددهی سپس مطابق با تابع عضویت برای متغیر خروجی انجام می شود.



موارد کاربرد منطق فازی

موارد کاربرد کلیدی منطق فازی به شرح زیر است:

سیستم های خودرو

- جعبه دنده های خودکار
- سیستم فرمان چهار چرخ
- کنترل محیط خودرو

کالاهای الکترونیکی مصرفی

- سیستم های صوتی
- پرینترهای کپی
- دوربین های فیلمبرداری و عکاسی

- تلویزیون

کالاهای خانگی

- مایکروویو
- یخچال‌ها
- توسترها
- جاروبرقی‌ها
- ماشین‌های لباسشویی

کنترل محیط

- سیستم‌های تهویه مطبوع/خشک‌کن/بخاری
- مه‌سازها

مزایای FLS‌ها

- مفاهیم ریاضی در منطق فازی بسیار ساده هستند.
- می‌توانید یک FLS را فقط با افزودن یا حذف قوانین به دلیل انعطاف‌پذیری منطق فازی تغییر دهید.
- سیستم‌های منطق فازی می‌توانند اطلاعات ورودی غیر دقیق، تحریف شده و پر سر و صدا را بپذیرند.
- FLS‌ها برای ساخت و درک آسان هستند.

- منطق فازی یک راه حل برای مشکلات پیچیده در همه زمینه‌های زندگی، از جمله پزشکی است، زیرا شبیه استدلال و تصمیم‌گیری انسان است.

معایب FLS ها

- هیچ روش سیستماتیک برای طراحی سیستم فازی وجود ندارد.
- فقط زمانی قابل فهم هستند که ساده باشند.
- آنها برای مشکلاتی که نیازی به دقت بالایی ندارند مناسب هستند.

هوش مصنوعی - پردازش زبان طبیعی

پردازش زبان طبیعی (NLP) به معنای روشی بر مبنای هوش مصنوعی جهت ارتباط با سیستم‌های هوشمند با استفاده از یک زبان طبیعی مانند انگلیسی است.

زمانی که می‌خواهید یک سیستم هوشمند مانند ربات به دستورات شما عمل کند یا زمانی که می‌خواهید تصمیمی از یک سیستم خبره بر اساس گفتگو بشتوید، استفاده از پردازش زبان طبیعی ضروری است

زمینهٔ NLP شامل کارهایی است که کامپیوترها را قادر می‌سازد تا وظایف مفیدی را با زبان‌های طبیعی انجام دهند که انسان‌ها استفاده می‌کنند. ورودی و خروجی یک سیستم NLP می‌تواند به شرح زیر باشد -

- گفتار
- متن نوشتاری

اجزای NLP

دو جزء در NLP وجود دارند که به شرح زیر است -

فهم زبان طبیعی (NLU)

فهم شامل وظایف زیر است -

- نگاشت ورودی داده شده به زبان طبیعی به ارائه‌های مفید.
- تجزیه و تحلیل جنبه‌های مختلف زبان.

تولید زبان طبیعی (NLG)

این فرآیند تولید عبارات و جملات معنی‌دار به شکل زبان طبیعی است.

این شامل وظایف زیر است -

- برنامه‌ریزی متن - این شامل بازیابی محتوای مرتبط از پایگاه دانش است.
- برنامه‌ریزی جمله - این شامل انتخاب کلمات مورد نیاز، تشکیل عبارات معنی‌دار، تنظیم لحن جمله است.
- واقعی‌سازی متن - این نگاشت جمله به ساختار جمله است.

NLU سخت‌تر از NLG است.

مشکلات در NLU

زبان طبیعی یک ساختار بسیار غنی و پیچیده دارد.

این مطلب می‌تواند سطوح مختلفی از ابهام داشته باشد -

- ابهام لغوی - این در سطح بسیار ابتدایی مانند سطح واژه است.
- به عنوان مثال، چگونه کلمه "تخته" را در نظر بگیریم؟ به عنوان اسم یا فعل؟

- ابهام ساختاری — یک جمله می‌تواند به روش‌های مختلفی تجزیه و تحلیل شود.
- به عنوان مثال، "او بلند کرد مورچه را با کلاه قرمز." — آیا او از کلاه برای بلند کردن مورچه استفاده کرد یا او مورچه ای را بلند کرد که کلاه قرمز داشت؟
- ابهام ارجاعی — به چیزی با استفاده از ضمائر ارجاع می‌شود. به عنوان مثال، سارا پیش مریم رفت. او گفت، "من خوشحالم —".
- دقیقاً کی خوشحال است؟
- یک ورودی می‌تواند معانی مختلفی داشته باشد.
- ممکن است چند ورودی به یک معنی برسند.

اصطلاحات NLP

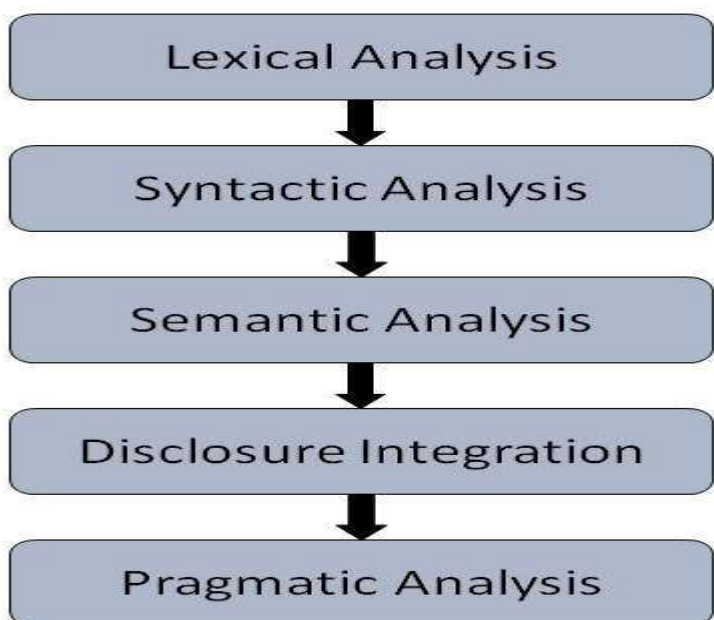
- فونولوژی — این مطالعه نحوه‌ی سیستماتیک سازماندهی صداهاست.
- مورفولوژی — این مطالعه ساخت واژه‌ها از واحدهای ابتدایی معنایی است.
- مورفم — این واحد ابتدایی معنا در یک زبان است.
- نحو — این به ترتیب کلمات برای ساخت یک جمله اشاره دارد. همچنین درگیر تعیین نقش ساختاری کلمات در جمله و عبارات است.
- معناشناسی — این مربوط به معنای کلمات و نحوه ترکیب کلمات به عبارات و جملات معنی‌دار است.
- عملکردی — این درباره استفاده و درک جملات در شرایط مختلف است و نحوه تفسیر جمله بررسی می‌کند.

- گفتمان - این به چگونگی اینکه جمله‌ای که فوراً قبل از آن آمده است چگونه تفسیر جمله بعدی را تحت تأثیر قرار می‌دهد، می‌پردازد.
- دانش جهان - این شامل دانش عمومی درباره جهان است.

مراحل در پردازش زبان طبیعی

پنج مرحله عمومی وجود دارد -

- تجزیه و تحلیل واژگانی - این شامل شناسایی و تجزیه و تحلیل ساختار کلمات است. واژه‌نامه یک زبان به مجموعه‌ای از کلمات و عبارات در یک زبان اشاره دارد. تجزیه و تحلیل واژگانی به معنای تقسیم تکه کلان متن به پاراگراف‌ها، جملات و کلمات است.
- تجزیه و تحلیل نحوی (تجزیه) - این شامل تجزیه و تحلیل کلمات در جمله برای گرامر و ترتیب کلمات به نحوی است که نشان‌دهنده رابطه بین کلمات است. جملاتی مانند "مدرسه به پسر می‌رود" توسط تجزیه‌کننده نحوی انگلیسی رد می‌شود.



- تجزیه و تحلیل معنایی - این معنای دقیق یا معنای دیکشنری را از متن استخراج می‌کند. متن برای معنی‌دار بودن بررسی می‌شود. این کار با نگاشت ساختارهای نحوی و اشیاء در دامنه وظیفه انجام می‌شود. تجزیه‌کننده معنایی جملاتی مانند "بستنی داغ" را نادیده می‌گیرد.
- ادغام گفتمانی - معنای هر جمله به معنای جمله فوراً قبل از آن وابسته است. به علاوه، این همچنین به معنای جمله فوراً بعد از آن اشاره دارد.
- تجزیه و تحلیل عملی - در این مرحله، آنچه گفته شده، مجدداً بر اساس آنچه واقعاً می‌خواسته است تفسیر می‌شود. این شامل استخراج آن جنبه‌های زبان است که نیازمند دانش واقعی جهان هستند.

جنبه‌های پیاده‌سازی تجزیه و تحلیل نحوی

تعدادی الگوریتم برای تجزیه و تحلیل نحوی توسعه داده شده است، اما ما فقط به دو روش ساده زیر می‌پردازیم -

- گرامر آزاد از متن (CFG)

• تجزیه‌کننده بالا به پایین (Top-Down Parser)

بیا باید جزئیات آنها را بررسی کنیم -

گرامر آزاد از متن (CFG)

این گرامر شامل قوانینی است که یک نماد تنها در سمت چپ قوانین بازنویسی دارد. بیا باید یک گرامر بسازیم تا یک جمله را تجزیه و تحلیل کنیم -

”یک پرنده دانه‌ها را می‌خورد“

حرف تعریف (DET) - یک

اسم‌ها (Nouns) - پرنده | پرندگان | دانه | دانه‌ها

عبارت اسمی (NP) - حرف تعریف + اسم | حرف تعریف + صفت + اسم

= DET N | DET ADJ N

فعل‌ها (Verbs) - می‌خورد | می‌خورند | می‌خوردند

عبارت فعلی (VP) - NP V | V NP

صفات (ADJ) - زیبا | کوچک

درخت تجزیه و تحلیل جمله را به اجزای ساختاری شکسته و تقسیم می‌کند تا کامپیوتر بتواند آن را به راحتی درک و پردازش کند. برای الگوریتم تجزیه و تحلیل این درخت تشکیل دهد، یک مجموعه از قوانین بازنویسی باید ساخته شود، که توصیف می‌کنند چه ساختارهای درختی مجاز هستند.

این قوانین می‌گویند که یک نماد خاص ممکن است با دنباله‌ای از نمادهای دیگر در درخت گسترده شود. طبق قاعدهٔ منطقی اول، اگر دو رشته «عبارت اسمی» (NP) و «عبارت فعلی» (VP) وجود داشته باشد، رشته ایجاد شده توسط NP دنباله VP یک جمله است. قوانین بازنویسی برای جمله به شرح زیر هستند -

$S \rightarrow NP VP$

$NP \rightarrow DET N \mid DET ADJ N$

$VP \rightarrow V NP$

(Lexicon) –

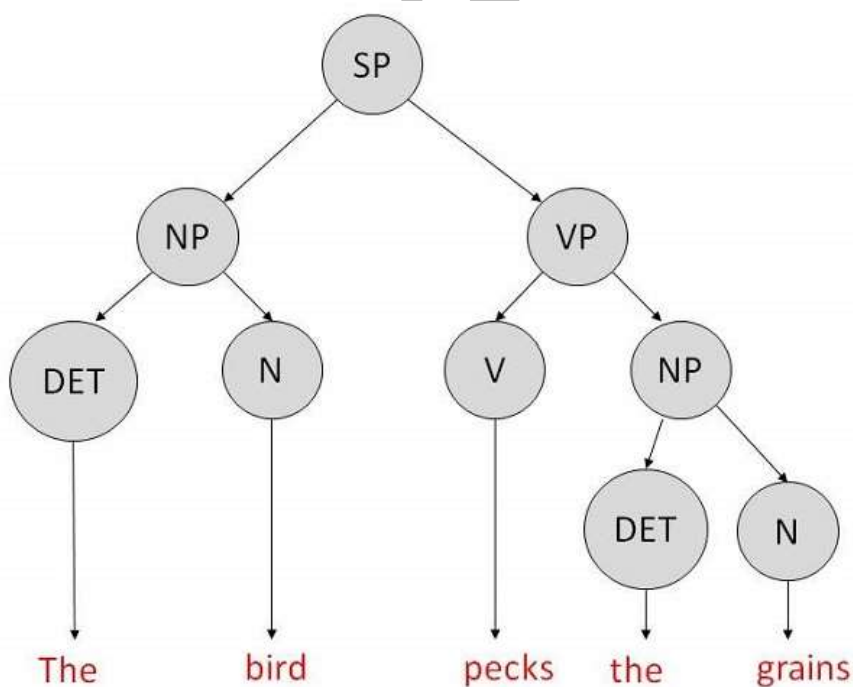
$DET \rightarrow a \mid the$

$ADJ \rightarrow \text{زیبا} \mid \text{کوچک}$

$N \rightarrow \text{پرنده} \mid \text{پرندگان} \mid \text{دانه} \mid \text{دانه‌ها}$

$V \rightarrow \text{می‌خورد} \mid \text{می‌خورند} \mid \text{می‌خوردند}$

درخت تجزیه و تحلیل می‌تواند به شکل زیر ایجاد شود –



حالا قوانین بازنویسی فوق را در نظر بگیرید. از آنجایی که V می تواند توسط "می خورد" یا "می خورند" جایگزین شود، جملاتی مانند "پرنده دانه ها را می خورند" به اشتباه ممکن است مجاز شوند. به عبارت دیگر، خطا در تطابق فاعل و فعل به عنوان درست تأیید می شود.

مزیت — ساده ترین نوع گرامر، بنابراین گسترده ترین استفاده را دارد.

نقاط ضعف —

- آنها بسیار دقیق نیستند. به عنوان مثال، "دانه ها پرنده را می خورند"، از نظر نحوی درست است، اما حتی اگر معنایی نداشته باشد، پارسر آن را به عنوان یک جمله درست می پذیرد.
- برای دقت بالا، نیاز به چندین مجموعه گرامر و قوانین مختلف برای تجزیه و تحلیل نیاز است. این ممکن است نیاز به مجموعه های کاملاً متفاوت از قوانین برای تجزیه و تحلیل تغییرات تک و جمع، جملات فعال و ... داشته باشد که ممکن است منجر به ایجاد مجموعه های بزرگ و نامداری شود که قابل مدیریت نیستند.

تجزیه کننده بالا به پایین (Top-Down Parser)

در اینجا، پارسر با نماد S شروع به کار می کند و سعی می کند آن را به یک دنباله از نمادهای پایانی بازنویسی کند که با کلاس های کلمات در جمله ورودی همخوانی دارد تا کاملاً از نمادهای پایانی تشکیل شود.

سپس اینها با جمله ورودی بررسی می شوند تا ببینیم آیا همخوانی دارند یا خیر. اگر ندارند، فرآیند با مجموعه دیگری از قوانین دوباره آغاز می شود. این کار تا زمانی ادامه دارد که یک قاعده خاص پیدا شود که ساختار جمله را توصیف می کند.

مزیت — پیاده سازی آن ساده است.

نقاط ضعف -

- کارایی آن کم است، چرا که در صورت بروز خطا فرآیند جستجو باید مجدداً تکرار شود.
- سرعت کار کم است.

هوش مصنوعی - سیستم‌های خبره

سیستم‌های خبره (ES) یکی از حوزه‌های مهم تحقیقات هوش مصنوعی هستند. این مفهوم توسط پژوهشگران دانشگاه استنفورد، دانشکده علوم کامپیوتر، معرفی شده است.

سیستم‌های خبره چیست؟

سیستم‌های خبره، نرم‌افزارهای کامپیوتری هستند که برای حل مسائل پیچیده در یک حوزه خاص، به سطح استعداد انسانی و تخصصی رسیده‌اند.

ویژگی‌های سیستم‌های خبره

- کارایی بالا
- قابل فهم
- قابل اعتماد
- پاسخگویی بسیار سریع

قابلیت‌های سیستم‌های خبره

سیستم‌های خبره قادر به -

- مشاوره
- دستور دادن و کمک به انسان در تصمیم‌گیری

- نمایش دادن
- یافتن راه حل
- تشخیص دادن
- توضیح دادن
- تفسیر کردن ورودی
- پیش‌بینی نتایج
- توجیه کردن نتیجه
- پیشنهاد دادن گزینه‌های جایگزین به یک مسئله هستند.

آنها قادر به —

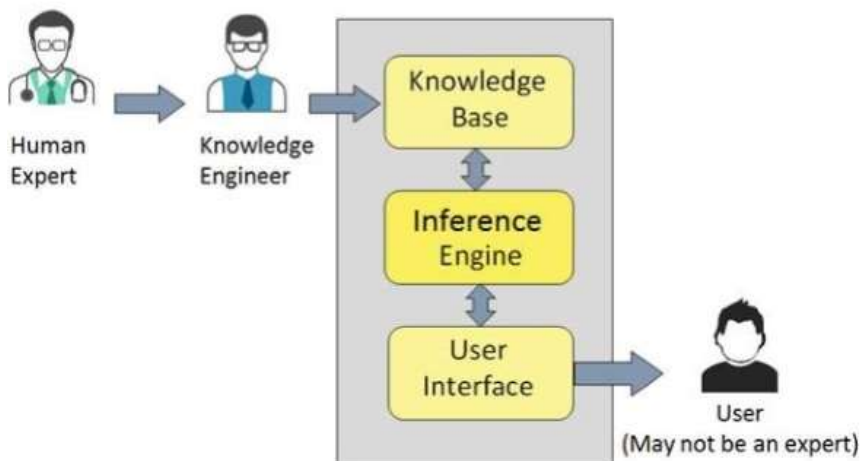
- جایگزینی تصمیم‌گیران انسانی نیستند
- دارای قابلیت‌های انسانی نیستند
- تولید خروجی دقیق برای پایگاه دانش ناکافی نیستند
- دانش خود را بهبود نمی‌دهند

اجزای سیستم‌های خبره

اجزای یک سیستم خبره عبارتند از —

- پایگاه دانش
- موتور استنتاج
- رابط کاربری

بیایید به تفصیل هر کدام را بررسی کنیم -



پایگاه دانش

این پایگاه دانش شامل دانش خاص حوزه و با کیفیت بالا است.

برای نمایش هوش نیاز به دانش داریم. موفقیت هر سیستم خبره به طور عمده به جمع‌آوری صحیح و دقیق دانش بستگی دارد.

دانش چیست؟

داده مجموعه‌ای از حقایق است. اطلاعات به عنوان داده‌ها و حقایق در مورد حوزه وظیفه، سازماندهی می‌شوند. داده، اطلاعات، و تجربیات گذشته با هم به عنوان دانش شناخته می‌شوند.

اجزای پایگاه دانش

پایگاه دانش یک سیستم خبره، ذخیره‌سازی دانش واقعی و هیوریستیک است.

- دانش واقعی - اطلاعاتی است که توسط مهندسان دانش و دانشمندان حوزه وظیفه به عنوان صحیح شناخته شده‌اند.

- دانش هیوریستیک – در مورد عمل، ارزیابی دقیق، توانایی ارزیابی و حدس زدن فرد است.

نمایش دانش

این روشی است که برای سازماندهی و فرمالیزه کردن دانش در پایگاه دانش استفاده می‌شود. این به شکل قوانین IF-THEN-ELSE است.

یادگیری دانش

موفقیت هر سیستم خبره به طور عمده به کیفیت، کمال و دقت اطلاعات ذخیره شده در پایگاه دانش بستگی دارد.

پایگاه دانش توسط خواندن مقالات از انواع متخصصان، دانشمندان و مهندسان دانش شکل می‌گیرد. مهندس دانش فردی است با ویژگی‌های انتقال فکر، یادگیری سریع و مهارت‌های تحلیل موردی.

او اطلاعات را از متخصص موضوع با ضبط، مصاحبه و مشاهده در حال کار، و غیره به دست می‌آورد. سپس اطلاعات را به شکل معنی‌دار و به شکل قوانین IF-THEN-ELSE دسته‌بندی و سازماندهی می‌کند تا توسط ماشین استنتاج استفاده شود. مهندس دانش نیز پیشرفت سیستم خبره را نظارت می‌کند.

موتور استنتاج

استفاده از روش‌ها و قوانین کارآمد توسط موتور استنتاج در استنتاج یک راه حل صحیح و بی‌نقص ضروری است.

در مورد سیستم‌های خبره مبتنی بر دانش، موتور استنتاج اطلاعات را از پایگاه دانش به دست می‌آورد و آن را به یک راه حل خاص می‌رساند.

در مورد سیستم‌های خبره مبتنی بر قوانین، این –

- قوانین را به حقایق که از برنامه قبلی حاصل شده‌اند، اعمال می‌کند.
- اگر لازم باشد دانش جدید را به پایگاه دانش اضافه می‌کند.
- هنگامی که چندین قانون برای یک مورد خاص قابل اعمال باشند، اختلافات قوانین را حل می‌کند.

برای پیشنهاد یک راه حل، موتور استنتاج از استراتژی‌های زیر استفاده می‌کند -

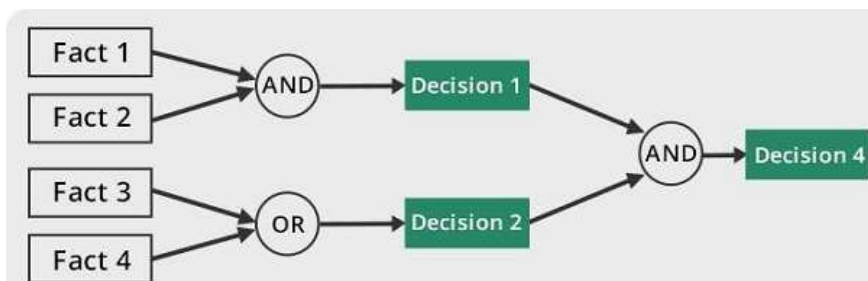
- پیشروی گام به گام (Forward Chaining)
- پسروی گام به گام (Backward Chaining)

پیشروی گام به گام

این استراتژی یک سیستم خبره برای پاسخ به سوال «چه اتفاقی ممکن است بیفتد؟» استفاده می‌شود.

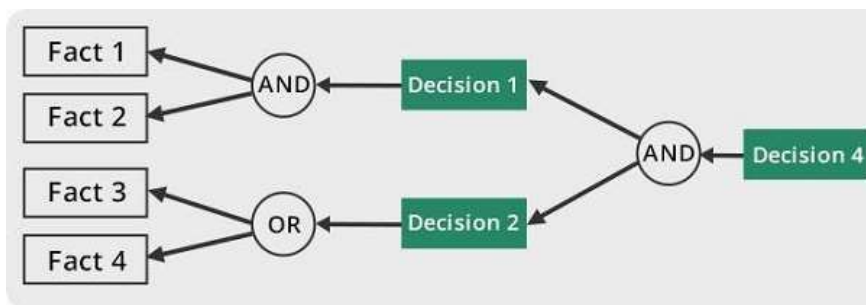
در اینجا، موتور استنتاج زنجیره شرایط و استنتاجات را دنبال می‌کند و در نهایت به نتیجه می‌رسد. تمام حقایق و قوانین را در نظر می‌گیرد و مرتب می‌کند پیش از اینکه به یک راه حل نهایی برسد.

این استراتژی برای کار روی استنتاج، نتیجه یا اثر انجام می‌شود. به عنوان مثال، پیش‌بینی وضعیت بازار سهام به عنوان یک اثر تغییرات نرخ.



پسروی گام به گام

با این استراتژی، یک سیستم خبره پاسخ به سوال «چرا این اتفاق افتاد؟» را پیدا می‌کند. بر اساس آنچه قبلاً اتفاق افتاده است، موتور استنتاج سعی می‌کند بیابد که در گذشته چه شرایطی ممکن است برای این نتیجه رخ داده باشد. این استراتژی برای یافتن علت یا دلیل استفاده می‌شود. به عنوان مثال، تشخیص برخی بیماری‌ها در انسان‌ها.



رابط کاربری

رابط کاربری ارتباط بین کاربر سیستم خبره و خود سیستم خبره فراهم می‌کند. این عموماً پردازش زبان طبیعی است تا توسط کاربری که در حوزه وظیفه ماهر است استفاده شود. کاربر سیستم خبره لازم نیست الزاماً یک متخصص در هوش مصنوعی باشد. توضیحات سیستم خبره زمانی که به یک توصیه خاص رسیده است ممکن است به یکی از این اشکال ظاهر شود-

- زبان طبیعی که روی صفحه نمایش نشان داده شود.
- توضیحات شفاهی به زبان طبیعی.
- لیست شماره‌های قوانین که روی صفحه نمایش داده شده‌اند.

رابط کاربری امکان پیگیری قابلیت اطمینان از استنتاج‌ها را فراهم می‌کند.

نیازهای رابط کاربری کارآمد سیستم‌های خبره

- باید به کاربران کمک کند تا با کوتاه‌ترین راه ممکن اهداف خود را دستیابی کنند.
- باید طراحی شده باشد تا برای روش‌های کاری موجود یا مطلوب کاربر کار کند.
- فناوری آن باید قابل تطبیق با نیازهای کاربر باشد؛ نه برعکس.
- باید از ورودی کاربر به بهترین نحو استفاده کند.

محدودیت‌های سیستم‌های خبره

هیچ فناوری‌ای نمی‌تواند راه‌حل آسان و کاملی ارائه دهد. سیستم‌های بزرگ هزینه بالا، زمان توسعه قابل توجه و منابع کامپیوتری نیاز دارند. سیستم‌های خبره محدودیت‌های خود را دارند که شامل -

- محدودیت‌های فناوری
- یادگیری دانش دشوار
- نگهداری دشوار سیستم‌های خبره
- هزینه‌های توسعه بالا

کاربردهای سیستم‌های خبره

جدول زیر نشان می‌دهد که سیستم‌های خبره کجا می‌توانند به کار گرفته شوند.

توضیحات	برنامه کاربردی
طراحی لنز دوربین، طراحی خودرو.	حوزه طراحی

سیستم‌های تشخیص برای استنتاج علت بیماری از داده‌های مشاهده‌شده، انجام عملیات پزشکی بر روی انسان‌ها.	حوزه پزشکی
مقایسه داده‌ها به صورت مداوم با سامانه‌ای که مشاهده می‌شود یا با رفتار مشخص مانند نظارت بر نشتی در لوله‌های نفت بلند.	سیستم‌های نظارت
کنترل یک فرآیند فیزیکی بر اساس نظارت.	سیستم‌های کنترل فرآیند
پیدا کردن اشکال در وسایل نقلیه، کامپیوترها.	حوزه دانش
شناسایی تقلب ممکن، معاملات مشکوک، معاملات بورس، برنامه‌ریزی ارلاین، برنامه‌ریزی بارگیری.	مالی/تجارت

تکنولوژی سیستم‌های خبره

چندین سطح از فناوری‌های سیستم‌های خبره وجود دارد. فناوری‌های سیستم‌های خبره شامل -

- محیط توسعه سیستم خبره - محیط توسعه سیستم خبره شامل سخت‌افزار و ابزارها است. آنها-
 - ایستگاه‌های کاری، مینی کامپیوترها، مین فریم‌ها.
 - زبان‌های برنامه‌نویسی سطح بالا مانند LISP Programming و PROgrammation en LOGique (PROLOG).
 - پایگاه‌های داده بزرگ.
- ابزارها - تلاش و هزینه مربوط به توسعه یک سیستم خبره را به حد زیادی کاهش می‌دهند.
 - ویرایشگرهای قدرتمند و ابزارهای اشکال‌زدایی با چند پنجره.

○ پروتوتایپ سازی سریع فراهم می کنند.

○ تعاریف توده ای مدل، نمایش دانش و طراحی استنتاجی دارند.

- پوسته ها - یک پوسته چیزی نیست جز یک سیستم خبره بدون پایگاه دانش. یک پوسته توسعه دهندگان را با یادگیری دانش، موتور استنتاج، رابط کاربری و امکان توضیحات فراهم می کند. به عنوان مثال، چند پوسته زیر وجود دارد

○ پوسته سیستم خبره جاوا (JESS) که یک API جاوا کاملاً توسعه

یافته برای ایجاد یک سیستم خبره فراهم می کند.

○ ویدوان، یک پوسته که در مرکز ملی فناوری نرم افزار مومبای در

سال 1993 توسعه داده شده است. این اجازه را می دهد که دانش به

شکل قوانین IF-THEN کد گذاری شود.

توسعه سیستم های خبره: گام های عمومی

فرآیند توسعه سیستم های خبره (ES) به صورت مکرر انجام می شود. گام های توسعه ES شامل موارد زیر هستند -

شناسایی حوزه مسئله

- مشکل باید مناسب برای یک سیستم خبره باشد تا آن را حل کند.
- کارشناسان حوزه وظیفه را برای پروژه سیستم خبره پیدا کنید.
- اقتصادی بودن سیستم را تأیید کنید.

- شناسایی فناوری سیستم خبره
- درک و تأیید درجه ادغام با سایر سیستم‌ها و پایگاه‌های داده.
- درک کنید که چگونه مفاهیم می‌توانند دانش حوزه را بهتر نمایند.

توسعه نمونه کارکرد (پروتوتایپ)

از پایگاه دانش: مهندس دانش کار می‌کند تا-

- دانش حوزه را از کارشناس به دست آورد.
- آن را به صورت قوانین IF-THEN-ELSE نمایش دهد.

آزمایش و بهبود نمونه کارکرد (پروتوتایپ)

- مهندس دانش از موارد نمونه برای آزمایش پروتوتایپ برای هر نقصان در عملکرد استفاده می‌کند.
- کاربران نهایی پروتوتایپ‌های سیستم‌های خبره را آزمایش می‌کنند.

توسعه و به پایان رساندن سیستم خبره

- آزمایش و اطمینان حاصل کنید که سیستم خبره با تمام اجزای محیط خود، از جمله کاربران نهایی، پایگاه‌های داده و سایر سیستم‌های اطلاعاتی ارتباط دارد.
- پروژه سیستم خبره را به خوبی مستند سازی کنید.
- کاربر را آموزش دهید تا از سیستم خبره استفاده کند.

نگهداری سیستم

- اطمینان حاصل کنید که پایگاه دانش با بازبینی و به روزرسانی منظم به روز است.
- برای واسطه‌های جدید با سایر سیستم‌های اطلاعاتی که آن سیستم‌ها تکامل می‌کنند، مرتب کنید.

مزایای سیستم‌های خبره

- دسترسی - به دلیل تولید گسترده نرم‌افزار، به راحتی دسترسی پیدا می‌کنند.
- هزینه تولید کمتر - هزینه تولید معقول است. این باعث می‌شود که قابل انجام باشند.
- سرعت - سرعت بالایی ارائه می‌دهند. میزان کاری که یک فرد انجام می‌دهد را کاهش می‌دهند.
- نرخ خطا کمتر - نرخ خطا کمتر است نسبت به خطاهای انسانی.
- کاهش خطر - می‌توانند در محیط‌هایی که برای انسان‌ها خطرناک است کار کنند.
- پاسخ پایدار - آنها بدون حالت هیجانی، تنش و خستگی به طور پایدار کار می‌کنند.

هوش مصنوعی - رباتیک

رباتیک یک حوزه در هوش مصنوعی است که به مطالعه ایجاد ربات‌های هوشمند و کارآمد می‌پردازد.

ربات‌ها چیستند؟

ربات‌ها عامل‌های مصنوعی هستند که در محیط واقعی عمل می‌کنند.

هدف

ربات‌ها وظیفه دارند که با درک، دریافت، حرکت دادن، تغییر دادن ویژگی‌های فیزیکی یا تأثیر گذاشتن بر آن بدون خسته شدن، به انسان‌ها در انجام وظایف تکراری یاری دهند.

رباتیک چیست؟

رباتیک یک شاخه از هوش مصنوعی است که از مهندسی برق، مهندسی مکانیک و علوم کامپیوتر برای طراحی، ساخت و کاربرد ربات‌ها تشکیل شده است.

جنبه‌های رباتیک

- ربات‌ها دارای ساختار مکانیکی، فرم یا شکلی هستند که برای انجام یک وظیفه خاص طراحی شده‌اند.
- آن‌ها دارای مؤلفه‌های الکتریکی هستند که قدرت و کنترل ماشین‌آلات را فراهم می‌کنند.
- آن‌ها شامل سطحی از برنامه کامپیوتری هستند که تعیین می‌کند ربات وظایف خود را چگونه و کی انجام می‌دهد.

تفاوت در سیستم ربات و دیگر برنامه‌های هوش مصنوعی

اینجا تفاوت بین این دو وجود دارد -

برنامه‌های هوش مصنوعی	ربات‌ها
-----------------------	---------

آن‌ها در دنیای فیزیکی واقعی عمل می‌کنند	آن‌ها معمولاً در دنیاهای شبیه‌سازی شده کامپیوتری عمل می‌کنند.
ورودی‌ها به ربات‌ها نشانه آنالوگ به صورت امواج صوتی یا تصاویر هستند	ورودی به یک برنامه هوش مصنوعی در قالب نمادها و قوانین است.
آن‌ها نیاز به سخت‌افزارهای ویژه با حسگرها و اثرگذارها دارند.	آن‌ها نیاز به کامپیوترهای عمومی دارند تا بر روی آن‌ها عمل کنند.

حرکت ربات

حرکت، مکانیزمی است که ربات را قادر به جابجایی در محیط خود می‌سازد. انواع مختلفی از حرکت وجود دارد –

- پایه‌ها
- چرخ‌دار
- ترکیبی از حرکت پایه‌ها و چرخ‌ها
- لغزش/سرخوردگی با پایه‌ها

حرکت پایه‌ها

- این نوع حرکت در حالت پیاده‌روی، پرش، دویدن، بالا یا پایین و ... از نیروی بیشتری مصرف می‌کند.
- برای انجام یک حرکت نیاز به تعداد بیشتری موتور دارد. این نوع حرکت برای سطوح ناهموار و همچنین صاف مناسب است، جایی که سطح نامنظم یا خیلی صاف باعث می‌شود در حرکت چرخ‌دار نیروی بیشتری مصرف کند. به دلیل مشکلات پایداری اجرای آن کمی دشوار است.

- این نوع حرکت با تنوع یک، دو، چهار و شش پا معمولاً انجام می شود. اگر ربات چند پا دارد، هماهنگی پاها برای حرکت ضروری است.

تعداد کل قدم‌ها (یک دنباله دوره‌ای از رویدادهای بالابردن و آزاد کردن برای هر یک از پاها) که یک ربات می‌تواند انجام دهد، به تعداد پاهایش وابسته است.

اگر ربات k پا داشته باشد، تعداد رویدادهای ممکن $N = (2k-1)!$ خواهد بود.

در صورت ربات دو پا ($k=2$)، تعداد رویدادهای ممکن

$$N = (2k-1)! = (2*2-1)! = 3! = 6$$

است.

بنابراین شش رویداد مختلف ممکن وجود دارد—

- بالا بردن پای چپ
- آزاد کردن پای چپ
- بالا بردن پای راست
- آزاد کردن پای راست
- بالا بردن هر دو پا به‌طور همزمان
- آزاد کردن هر دو پا به‌طور همزمان

در صورت $k=6$ پا، 39916800 رویداد ممکن وجود دارد. بنابراین پیچیدگی ربات‌ها به طور مستقیم به تعداد پاها وابسته است.



حرکت چرخ‌دار

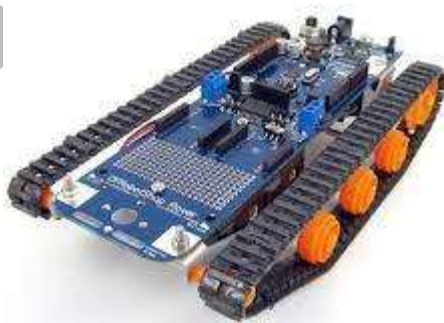
برای انجام حرکت به تعداد کمتری از موتورها نیاز دارد. اجرای آن نسبت به حرکت پایه‌ها آسان‌تر است چرا که در صورت وجود تعداد بیشتری چرخ مشکلات پایداری کمتری دارد. نسبت به حرکت پایه‌ها مصرف انرژی بهتری دارد.

- چرخ استاندارد – دوران در اطراف محور چرخ و در اطراف مختصات تماس دارد.
- چرخ کاستور – دوران در اطراف محور چرخ و مفصل جابجایی آفست دارد.
- چرخ‌های سوئدی 45° و سوئدی 90° – چرخ Omni، در اطراف نقطه تماس، در اطراف محور چرخ و در اطراف رولرها دور می‌شود.
- چرخ توپی یا کروی – چرخ همجهت، اجرای فنی سخت‌تری دارد.



حرکت لغزشی/سرخوردگی

در این نوع، وسایل نقلیه از چرخ‌دنده‌ها استفاده می‌کنند. ربات با جابجایی دنده‌ها با سرعت‌های مختلف در جهت یکسان یا مخالف هدایت می‌شود. این نوع پایداری را ارائه می‌دهد به دلیل ناحیه تماس بزرگ دنده و زمین.



اجزای یک ربات

ربات‌ها با لوازم زیر ساخته می‌شوند -

- منبع تغذیه - ربات‌ها توسط باتری‌ها، انرژی خورشیدی، هیدرولیکی یا هواپیمايي تغذیه می‌شوند.
- عملگرها - انرژی را به حرکت تبدیل می‌کنند.

- موتورهای الکتریکی (جریان مستقیم/متناوب) - برای حرکت چرخشی نیاز دارند.
- عضلات هواپیمایی - تقریباً 40٪ کم می‌شوند وقتی هوا در آنها مکش می‌شود.
- سیم‌های عضلات 5 -٪ کم می‌شوند وقتی جریان الکتریکی از آنها عبور می‌شود.
- موتورهای پیرو و موتورهای اولتراسونیک - برای ربات‌های صنعتی بهترین هستند.
- حسگرها - اطلاعات زمان واقعی در مورد محیط کار را فراهم می‌کنند. ربات‌ها با حسگرهای دیداری تجهیز می‌شوند تا عمق محیط را محاسبه کنند. حسگر لمسی ویژگی‌های مکانیکی گیرنده‌های لمس انگشتان انسان را تقلید می‌کند.

بینایی کامپیوتر

این یک فناوری هوش مصنوعی است که ربات‌ها می‌توانند ببینند. بینایی کامپیوتر نقش حیاتی در زمینه‌های ایمنی، امنیت، سلامتی، دسترسی و سرگرمی دارد.

بینایی کامپیوتر به طور خودکار اطلاعات مفیدی را از یک تصویر یا آرایه تصاویر استخراج، تجزیه و تحلیل می‌کند. این فرآیند شامل توسعه الگوریتم‌ها برای دستیابی به درک بصری خودکار است.

سخت‌افزار سیستم بینایی کامپیوتر

این سخت‌افزار شامل موارد زیر می‌باشد -

- منبع تغذیه
- دستگاه تصویربرداری مانند دوربین

- پردازنده
- یک نرم افزار
- یک دستگاه نمایش برای نظارت بر سیستم
- لوازم جانبی مانند استندهای دوربین، کابل ها و اتصالات

وظایف بینایی کامپیوتر

- OCR – در حوزه کامپیوتر، خواننده نویسه های نوری، یک نرم افزار برای تبدیل اسناد اسکن شده به متن قابل ویرایش است که همراه با یک اسکنر عرضه می شود.

- تشخیص چهره – بسیاری از دوربین های پیشرفته این ویژگی را دارند که امکان خواندن چهره و گرفتن عکس از عبارت مدنظر را فراهم می کند. از آن برای اجازه دادن به کاربر برای دسترسی به نرم افزار در صورت همخوانی صحیح استفاده می شود.

- تشخیص اشیاء – این امکان در دوربین ها، خودروهای لوکس مانند بامو، جی ام و ولوو استفاده شده است.

- تخمین موقعیت – تخمین می زند که یک شیء نسبت به دوربین در کجاست.

حوزه های کاربردی بینایی کامپیوتر

- کشاورزی
- وسایل نقلیه خودکار
- شناسایی اثر انگشت
- شناسایی نویسه ها

- کاربردهای نظارتی
- بازرسی کیفیت صنعتی
- شناسایی چهره
- تجزیه و تحلیل حرکات اشاره
- علوم زمین
- تصویربرداری پزشکی
- نظارت آلودگی
- کنترل فرآیند
- حسگری در دوربین از راه دور
- رباتیک
- حمل و نقل

کاربردهای رباتیک

رباتیک در حوزه‌های مختلفی مانند -

- صنعت - ربات‌ها برای مدیریت مواد، برش، جوش، رنگ‌آمیزی، حفاری، تراشکاری و غیره استفاده می‌شوند.
- پزشکی - ربات‌ها قادر به انجام صدها آزمایش بالینی به صورت همزمان، ترمیم بخشی افراد دائمی معلول و انجام عملیات پیچیده مانند جراحی هستند.

- اکتشاف - ربات‌های کوه‌نوردی برای اکتشاف فضا، زیرآبی‌های استفاده شده برای اکتشاف اقیانوس و ... از جمله مواردی هستند که می‌توان نام برد.
- و بسیاری از حوزه‌های دیگر

هوش مصنوعی - شبکه‌های عصبی

یکی دیگر از حوزه‌های تحقیقاتی در هوش مصنوعی، شبکه‌های عصبی می‌باشد که از شبکه عصبی طبیعی سیستم عصبی انسان الهام گرفته شده است.

شبکه‌های عصبی مصنوعی چیستند؟

مخترع اولین رایانه‌نورویی، دکتر رابرت هکت-نیلسن، یک شبکه عصبی را به شرح زیر تعریف می‌کند -

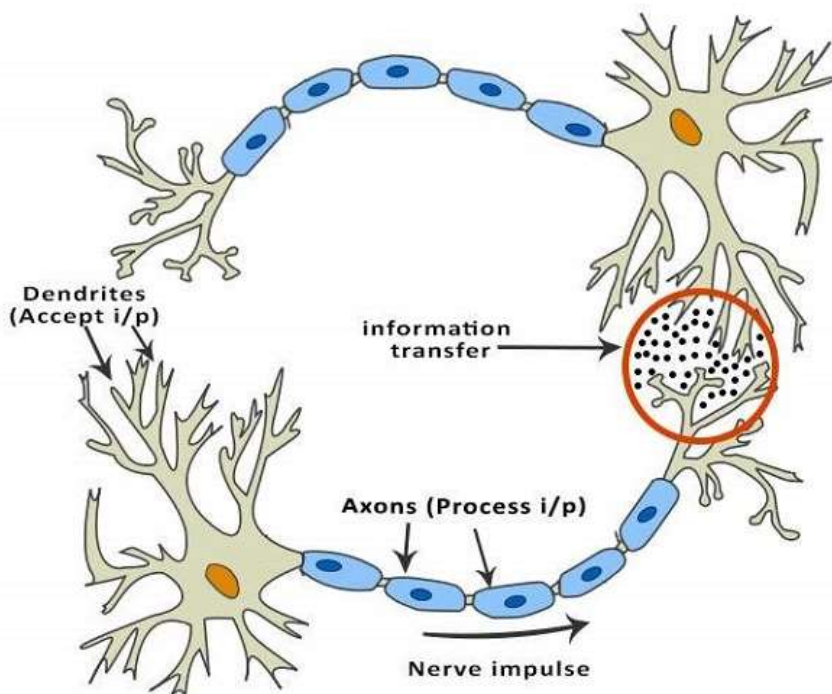
... "یک سامانه محاسباتی است که از تعدادی عناصر ساده و مرتبط پردازشی تشکیل شده است، که اطلاعات را در پاسخ به ورودی‌های خارجی در حالت پویا پردازش می‌کنند".

ساختار اساسی شبکه‌های عصبی مصنوعی

ایده شبکه‌های عصبی مصنوعی بر این باور مبنی است که عملکرد مغز انسان با ایجاد اتصالات درست، می‌تواند با استفاده از سیلیکون و سیم‌ها به عنوان **نورون‌ها** و **دندریت‌ها** شبیه‌سازی شود.

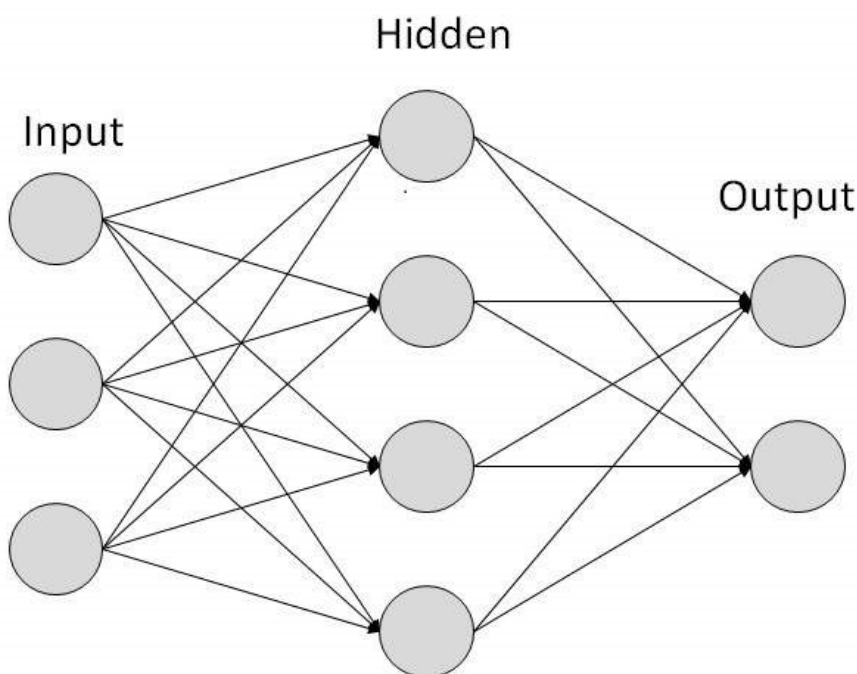
مغز انسان از 86 میلیارد سلول عصبی به نام **نورون‌ها** تشکیل شده است. آنها با هزاران سلول دیگر توسط **آکسون‌ها** متصل هستند. محرک‌ها از محیط خارجی یا ورودی‌ها از اعضای حسی توسط دندریت‌ها پذیرفته می‌شوند. این ورودی‌ها نوسانات الکتریکی ایجاد

می‌کنند که به سرعت از طریق شبکه عصبی حرکت می‌کنند. یک نورون می‌تواند پیام را به نورون دیگری ارسال کند تا مسئله را حل کند یا این پیام را به جلو نفرستد.



شبکه‌های عصبی مصنوعی از چندین **گره** تشکیل شده‌اند که **نورون‌های** زیستی مغز انسان را شبیه‌سازی می‌کنند. نورون‌ها توسط پیوندها متصل هستند و با یکدیگر تعامل دارند. گره‌ها می‌توانند داده‌های ورودی را دریافت کرده و عملیات ساده‌ای روی داده‌ها انجام دهند. نتیجه این عملیات به نورون‌های دیگر منتقل می‌شود. خروجی هر گره به نام **فعالیت یا مقدار گره** آن نامیده می‌شود.

هر پیوند با وزن مرتبط است. شبکه‌های عصبی مصنوعی قابلیت یادگیری دارند که از طریق تغییر مقادیر وزن انجام می‌شود. تصویر زیر یک شبکه عصبی مصنوعی ساده را نشان می‌دهد -

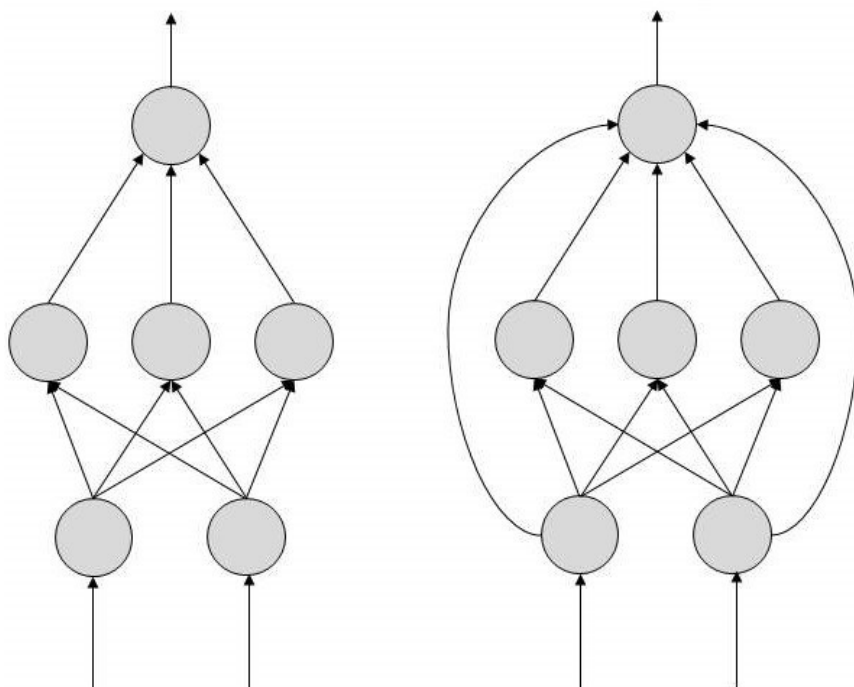


انواع شبکه‌های عصبی مصنوعی

دو نوع توپولوژی شبکه عصبی مصنوعی وجود دارد - پیش‌رو (FeedForward) و بازخوردی (Feedback).

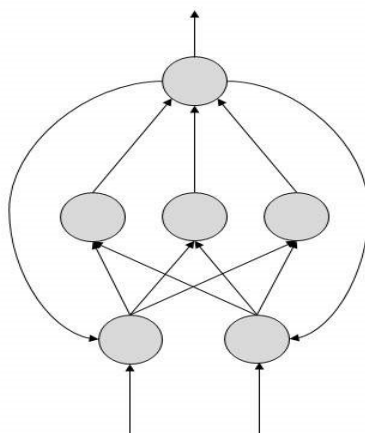
شبکه عصبی پیش‌رو

در این شبکه عصبی، جریان اطلاعات یک‌طرفه است. یک واحد اطلاعات را به واحد دیگری ارسال می‌کند که از آن واحد هیچ اطلاعاتی دریافت نمی‌کند. هیچ حلقه بازخوردی وجود ندارد. این نوع شبکه در تولید الگو/تشخیص/طبقه‌بندی استفاده می‌شود. ورودی‌ها و خروجی‌های آنها ثابت هستند.



شبکه عصبی بازخوردی

در اینجا، وجود حلقه‌های بازخوردی مجاز هست. اینها در حافظه‌های قابل دسترسی به محتوا استفاده می‌شوند.



عملکرد شبکه‌های عصبی مصنوعی

در نمودارهای توپولوژی نشان داده شده، هر پیکان یک اتصال بین دو نورون را نشان می‌دهد و مسیر جریان اطلاعات را نشان می‌دهد. هر اتصال دارای یک وزن می باشد. وزن، یک عدد صحیح است که سیگنال بین دو نورون را کنترل می‌کند.

اگر شبکه یک خروجی "خوب یا مطلوب" ایجاد کند، نیازی به تنظیم وزن‌ها نیست. با این حال، اگر شبکه یک خروجی "ضعیف یا نامطلوب" یا خطا ایجاد کند، سیستم وزن‌ها را تغییر می‌دهد تا نتایج آینده را بهبود بخشد.

یادگیری در شبکه‌های عصبی مصنوعی

شبکه‌های عصبی مصنوعی قابلیت یادگیری دارند و نیاز به آموزش دارند. چندین استراتژی یادگیری وجود دارد —

یادگیری نظارت‌شده — در اینجا یک معلم وجود دارد که دانایتر از شبکه عصبی خود است. به عنوان مثال، معلم داده‌هایی بعنوان نمونه می‌دهد که در مورد آنها پاسخ‌ها را از قبل می‌داند

. به عنوان مثال، شناسایی الگو. شبکه عصبی در هنگام شناسایی حدس می‌زند. سپس معلم به شبکه عصبی پاسخ می‌دهد. سپس شبکه مقایسه می‌کند حدسهای خود را با "صحیح" پاسخ‌های معلم و تنظیمات لازم را بر اساس خطاها انجام می‌دهد.

- یادگیری بدون نظارت — این نوع در مواردی که مجموعه داده‌ای با پاسخ‌های معلوم وجود ندارد مورد نیاز است. به عنوان مثال، جستجو برای یافتن یک الگوی پنهان. در این مورد، خوشه‌بندی، به اصطلاح تقسیم یک مجموعه از عناصر به گروه‌ها بر اساس یک الگوی ناشناخته بر اساس داده‌های موجود انجام می‌شود.

- یادگیری تقویتی - این استراتژی بر اساس مشاهده ساخته شده است. شبکه عصبی با مشاهده محیط خود تصمیم می‌گیرد. اگر مشاهده منفی باشد، شبکه وزن‌های خود را تنظیم می‌کند تا در زمان بعدی تصمیم متفاوتی بگیرد.

الگوریتم بازگشت به عقب (Back Propagation)

این الگوریتم آموزش یا یادگیری است. از طریق مثال یاد می‌گیرد. اگر به الگوریتم مثالی از آنچه که می‌خواهید شبکه انجام دهد، ارائه دهید، وزن‌های شبکه را تغییر می‌دهد تا بتواند در پایان آموزش خروجی مطلوبی برای ورودی مشخص تولید کند.

شبکه‌های بازگشت به عقب برای شناسایی الگوهای ساده و وظایف نقشه‌برداری ایده‌آل هستند.

شبکه‌های بیزین (BN)

اینها ساختارهای گرافیکی هستند که برای نمایش رابطه احتمالی میان مجموعه‌ای از متغیرهای تصادفی استفاده می‌شوند. شبکه‌های بیزین همچنین به نام **شبکه‌های باور** یا **شبکه‌های بیز** شناخته می‌شوند. این شبکه‌ها در مورد دامنه ناقص اندیشه می‌کنند.

در این شبکه‌ها، هر گره یک متغیر تصادفی با گزاره‌های خاص را نمایندگی می‌کند. به عنوان مثال، در حوزه تشخیص پزشکی، گره سرطان گزاره می‌کند که یک بیمار سرطان دارد.

یال‌هایی که گره‌ها را به هم متصل می‌کنند، وابستگی‌های احتمالی میان این متغیرهای تصادفی را نمایش می‌دهند. اگر از دو گره، یکی بر دیگری تأثیر بگذارد، آنها باید به صورت مستقیم در جهت تأثیر متصل شوند. قوت رابطه بین متغیرها توسط احتمال مرتبط با هر گره اندازه‌گیری می‌شود.

تنها محدودیتی که در یک شبکه بیزین وجود دارد، این است که نمی‌توانید به یک گره فقط با دنبال کردن یال‌های جهت‌دار برگردید. بنابراین شبکه‌های بیزین به عنوان گراف‌های جهت‌دار بدون دور (DAGs) نامیده می‌شوند.

شبکه‌های بیزین قادر به کنترل همزمان متغیرهای چندمقداری هستند. متغیرهای شبکه بیزین از دو بُعد تشکیل شده‌اند —

- دامنه گزاره‌ها
- احتمال اختصاص یافته به هر یک از گزاره‌ها.

یک مجموعه متناهی $X = \{X_1, X_2, \dots, X_n\}$ از متغیرهای تصادفی گسسته را در نظر بگیرید، جایی که هر متغیر X_i ممکن است ارزش‌ها را از یک مجموعه متناهی به نام $V_i(X_i)$ بگیرد. اگر یک پیوند جهت‌دار از متغیر X_i به متغیر X_j وجود داشته باشد، آنگاه متغیر X_i والد متغیر X_j خواهد بود که وابستگی‌های مستقیم میان متغیرها را نشان می‌دهد.

ساختار BN برای ترکیب دانش اولیه و داده‌های مشاهده شده ایده‌آل است. شبکه بیزین می‌تواند برای یادگیری ارتباطات علی و درک حوزه‌های مسائل مختلف و پیش‌بینی رویدادهای آینده، حتی در صورت وجود داده‌های از دست رفته، استفاده شود.

ساخت یک شبکه بیزین

یک مهندس دانش می‌تواند یک شبکه بیزین ایجاد کند. در حین ایجاد آن، تعدادی مراحل وجود دارد که مهندس دانش باید آن را انجام دهد.

مثال مسئله — سرطان ریه. یک بیمار از تنگی نفس رنج می‌برد. او با مشکوک بودن به سرطان ریه به پزشک مراجعه می‌کند. پزشک می‌داند که به استثنای سرطان ریه، بیماری‌های احتمالی دیگری نیز وجود دارد که ممکن است بیمار به آن مبتلا باشد، مانند سل و برونشیت.

جمع‌آوری اطلاعات مرتبط مسئله

- آیا بیمار سیگاری است؟ اگر بله، پس احتمال ابتلا به سرطان و برونشیت زیاد است.
- آیا بیمار در معرض آلودگی هوا قرار دارد؟ اگر بله، چه نوع آلودگی هوا؟
- گرفتن عکس اشعه ایکس مثبت نشان دهنده سل یا سرطان ریه است.

شناسایی متغیرهای جالب

مهندس دانش سعی می‌کند به سوالات پاسخ دهد —

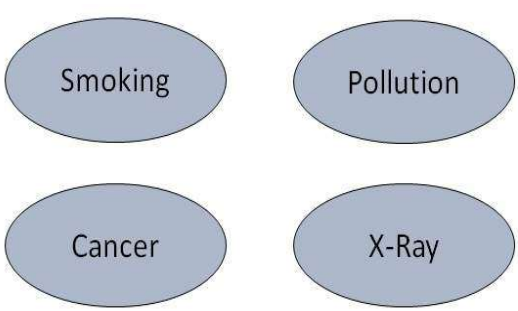
- کدام گره‌ها را نشان دهیم؟
- آنها چه ارزش‌هایی می‌توانند داشته باشند؟ در چه حالتی می‌توانند باشند؟

در حال حاضر بیاید تنها گره‌ها را در نظر بگیریم که فقط ارزش‌های گسسته دارند. متغیر باید در هر زمان دقیقاً یکی از این ارزش‌ها را داشته باشد.

انواع متداول گره‌های گسسته عبارتند از —

- گره‌های بولی — آنها گزاره‌ها را نمایندگی می‌کنند و ارزش‌های دودویی ($TRUE (T)$) و ($FALSE (F)$) را می‌پذیرند.
- مقادیر مرتب — آلودگی گره ممکن است نشان دهنده و مقادیری از {کم، متوسط، زیاد} باشد که درجه قرار گرفتن بیمار در معرض آلودگی را توصیف می‌کند.
- مقادیر صحیح — گرهی به نام Age ممکن است سن بیمار را با مقادیر احتمالی 1 تا 120 نشان دهد. حتی در این مرحله اولیه، انتخاب‌های مدل سازی در حال انجام است.

گره‌ها و ارزش‌های ممکن برای مثال سرطان ریه به شرح زیر هستند —

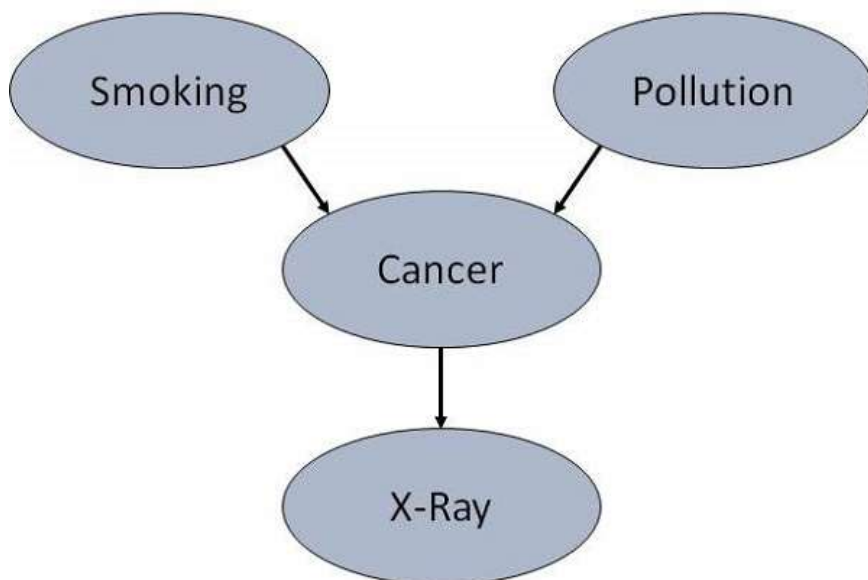
نام گره	نوع	ارزش	ایجاد گره‌ها
Pollution	دودویی	{کم، زیاد، متوسط}	
Smoker	بولی	{TRUE, FALSE}	
Lung-Cancer	بولی	{TRUE, FALSE}	
X-Ray	دودویی	{مثبت، منفی}	

ایجاد یال‌ها بین گره‌ها

توپولوژی شبکه باید روابط کیفی بین متغیرها را نشان دهد.

به عنوان مثال، چه چیزی باعث می‌شود بیمار سرطان ریه داشته باشد؟ - آلودگی و سیگار کشیدن. سپس یال‌ها را از گره *Pollution* و گره *Smoker* به گره *Lung-Cancer* اضافه کنید.

به همین ترتیب اگر بیمار سرطان ریه داشته باشد، نتیجه ایکس-ری مثبت خواهد بود. سپس یال‌ها را از گره *Lung-Cancer* به گره *X-Ray* اضافه کنید.



مشخص کردن توپولوژی

معمولاً، BNها به صورتی طراحی می‌شوند که یال‌ها از بالا به پایین اشاره کنند. مجموعه گره‌های والد یک گره X توسط $Parents(X)$ نمایش داده می‌شود.

گره $Lung-Cancer$ دو والد (دلایل یا علت‌ها) دارد $Pollution$ و $Smoker$ ؛ در حالی که گره $Smoker$ یک نیاکان از گره $X-Ray$ است. به طور مشابه، $X-Ray$ یک فرزند (نتیجه یا اثرات) گره $Lung-Cancer$ است و جانشین گره‌های $Smoker$ و $Pollution$ است.

احتمالات شرطی

حالا روابط بین گره‌های متصل را کمی کنید: این کار با مشخص کردن یک توزیع احتمال شرطی برای هر گره انجام می‌شود. چون تنها متغیرهای گسسته در نظر گرفته شده‌اند، این به صورت یک جدول احتمالات شرطی (CPT) انجام می‌شود.

اولاً، برای هر گره باید به تمام ترکیب‌های ممکن ارزش‌های این والد‌ها نگاه کنیم. هر چنین ترکیبی، یک نمایش از مجموعه والد است. برای هر نمایش متمایز از ارزش‌های والد، باید احتمالی که فرزند ممکن است بپذیرد، مشخص شود.

به عنوان مثال، والد‌های گره $Lung-Cancer$ و $Smoking$ هستند. آنها مقادیر ممکن را دارند $\{(H,T), (H,F), (L,T), (L,F)\}$ = جدول CPT احتمال سرطان برای هر یک از این موارد را به ترتیب $0.05, 0.02, 0.03, 0.001 >$ مشخص می‌کند.

هر گره احتمالات شرطی مرتبط دارد به صورت زیر -

Smoking	Pollution
$P(S = T)$	$P(P = L)$
0.30	0.90

Lung-Cancer		
P	S	P (C = T P, S)
H	T	0.05
H	F	0.02
L	T	0.03
L	F	0.001

X-Ray	
C	X = (Pos C)
T	0.90
F	0.20

کاربردهای شبکه‌های عصبی

آنها می‌توانند وظایفی را انجام دهند که برای انسان آسان است اما برای یک ماشین سخت است -

- هوا فضا - هواپیماهای خودران، تشخیص خطاهای هواپیما.
- اتومبیل - سیستم‌های راهنمایی خودرو.

- الکترونیک - پیش‌بینی دنباله کد، طراحی چیپ IC، تجزیه و تحلیل خرابی چیپ، بینایی ماشین، سنتز صدا.
- مالی - ارزیابی املاک، مشاور تسهیلات وام، انتخاب تسهیلات تورم، امتیازدهی بندرگاه‌های شرکتی، برنامه معاملات پورته، تحلیل مالی شرکت‌ها، پیش‌بینی ارزش ارز، خوانندگان اسناد، ارزیابان درخواست‌های اعتبار.
- صنعتی - کنترل فرآیند تولید، طراحی و تجزیه و تحلیل محصول، سیستم‌های بازرسی کیفیت، تجزیه و تحلیل کیفیت جوهر، طراحی و تجزیه و تحلیل محصولات شیمیایی، مدل‌سازی پویا سیستم‌های فرایند شیمیایی، تجزیه و تحلیل نگهداری ماشین، مناقصه‌نگاری، برنامه‌ریزی و مدیریت پروژه.
- پزشکی - تجزیه و تحلیل سلول‌های سرطان، تجزیه و تحلیل EEG و ECG، طراحی پروتز، بهینه‌ساز زمان پیوند.
- گفتار - شناسایی گفتار، طبقه‌بندی گفتار، تبدیل متن به گفتار.
- مخابرات - فشرده‌سازی تصویر و داده، خدمات اطلاعات خودکار، ترجمه زبان زنده گفتار به صورت زمان واقعی.
- حمل و نقل - تشخیص سیستم ترمز کامیون، برنامه‌ریزی و ردیابی وسایل نقلیه.
- نرم‌افزار - تشخیص الگو در شناسایی چهره، تشخیص نویسه‌های نوری و غیره.
- پیش‌بینی سری‌های زمانی - شبکه‌های عصبی برای پیش‌بینی درباره‌ی سهام و آتشفشان‌ها استفاده می‌شوند.
- پردازش سیگنال - شبکه‌های عصبی می‌توانند آموزش داده شوند تا سیگنال صوتی را پردازش کنند.
- کنترل - شبکه‌های عصبی اغلب برای انجام تصمیمات کنترلی خودروهای فیزیکی استفاده می‌شوند.
- تشخیص نقص - از آنجاییکه شبکه‌های عصبی در تشخیص الگوها ماهر هستند، آموزش داده می‌شوند تا هنگامی که چیزی غیرعادی اتفاق بیفتد که با الگو ناسازگار است خروجی تولید کنند.

هوش مصنوعی - مسائل و معضلات

هوش مصنوعی با سرعتی فوق العاده در حال توسعه است، گاهی اوقات به نظر می‌رسد که جادویی است. در میان برخی پژوهشگران و توسعه‌دهندگان این نظر وجود دارد که هوش مصنوعی به اندازه‌ای قدرتمند می‌شود که برای انسان‌ها دشوار خواهد بود که کنترلش کنند.

انسان‌ها سیستم‌های هوش مصنوعی را با وارد کردن هر نوع هوش ممکن که می‌توانستند، ایجاد کرده‌اند، که به نظر می‌رسد ممکن است تهدیدی برای خود انسان‌ها باشد

تهدید حریم خصوصی

یک برنامه هوش مصنوعی که قابلیت فهم سخن و زبان طبیعی را دارد، این قابلیت را احتمالاً داشته باشد که هر گفتگویی را در ایمیل‌ها و تلفن‌ها متوجه شود و پردازش کند.

تهدید کرامت انسانی

سیستم‌های هوش مصنوعی در حال حاضر جایگزین انسان‌ها در چند صنعت شده‌اند. نباید در مناصب شرافتمندانه‌ای که مربوط به اخلاق هستند مانند پرستاری، جراح، قاضی، افسر پلیس و غیره جایگزین انسان شوند.

تهدید ایمنی

سیستم‌های هوش مصنوعی خود-بهبود می‌توانند آنقدر از انسان‌ها قدرتمندتر شوند که متوقف کردن دستیابی به اهدافشان بسیار دشوار شود، که ممکن است منجر به عواقب ناخواسته شود.

هوش مصنوعی - مجموعه اصطلاحات

در اینجا لیستی از اصطلاحات متداول در زمینه هوش مصنوعی وجود دارد -

شماره	واژه و معنی
-------	-------------

1	<p>عامل (Agent)</p> <p>عوامل، سیستم‌ها یا برنامه‌های نرم‌افزاری هستند که قابلیت عملی خودمختار، هدفمند و استدلالی را به سوی یک یا چند هدف دارند. به آنها دستیار، کارگزار، ربات، دروید، عامل هوشمند و عامل نرم افزار نیز گفته می شود.</p>
2	<p>ربات خودمختار (Autonomous Robot)</p> <p>رباتی است که از کنترل یا تأثیرات خارجی آزاد است و قادر به کنترل خود است.</p>
3	<p>زنجیره عقب گرد (Backward Chaining)</p> <p>استراتژی کار به عقب برای یافتن دلیل یا علت یک مسأله.</p>
4	<p>تخته سیاه (Blackboard)</p> <p>این حافظه در داخل کامپیوتر استفاده می‌شود که برای ارتباط بین سیستم‌های متخصص همکار استفاده می‌شود.</p>
5	<p>محیط (Environment)</p> <p>این بخشی از دنیای واقعی یا محاسباتی است که عامل در آن ساکن است.</p>
6	<p>زنجیره پیشرو (Forward Chaining)</p> <p>استراتژی کار به جلو برای رسیدن به نتیجه یا حل یک مسأله.</p>
7	<p>استنتاج‌ها (Heuristics)</p> <p>دانش مبتنی بر امتحان و خطا، ارزیابی‌ها و آزمایش‌ها است.</p>
8	<p>مهندسی دانش (Knowledge Engineering)</p> <p>به دست آوردن دانش از انسان‌های متخصص و منابع دیگر.</p>

9	دریافت‌ها (Percepts) فرمتی است که عامل اطلاعات درباره محیط دریافت می‌کند.
10	هرس (Pruning) لغو اعتبارها و ملاحظات غیر ضروری در سیستم‌های هوش مصنوعی.
11	قاعده (Rule) فرمتی برای نمایش پایگاه دانش در سیستم‌های خبره. به شکل IF-THEN-ELSE است.
12	پوسته (Shell) یک نرم‌افزار است که در طراحی موتور استنتاج، پایگاه دانش و رابط کاربری یک سیستم خبره کمک می‌کند.
13	وظیفه (Task) هدفی است که عامل سعی در دستیابی به آن دارد.
14	آزمون تورینگ (Turing Test) یک آزمونی است که توسط آلن تورینگ توسعه داده شده است تا هوش یک ماشین را در مقایسه با هوش انسان ارزیابی کند.

محمد مهدی املانی