

# بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## Scribing شنبه - 19 اسفند 1402

محمد پور خاکساری

هوش مصنوعی از نظر لغوی به معنای "هوش دست‌ساز" است. هوش مجموعه‌ای از توانایی‌ها شامل یادگیری، تفکر و استفاده از دانش است. یادگیری توانایی کسب دانش و اطلاعات است. تفکر به معنای دسترسی به اطلاعات در حافظه یا پایگاه داده و پردازش آنها است. هوش مصنوعی به عنوان یک کلیت، علم ساخت و مدل‌سازی عامل‌های هوشمند است. هر علمی به دنبال پاسخ به سوالات "چگونه" و "چطور" است. در علم هوش مصنوعی، به این سوالات پاسخ داده می‌شود که چگونه یک عامل هوشمند بسازیم و یک عامل هوشمند چگونه کار می‌کند.

عامل هوشمند موجودی است که کاری انجام می‌دهد. اگر طبق تعریف هوش، عامل ما توانایی کسب دانش، تفکر و استفاده از آن را داشته باشد، یعنی هوشمند است.

عامل‌های هوشمند با استفاده از حسگرها محیط اطراف خود را درک می‌کنند. این محیط می‌تواند مجازی (مانند کامپیوتر) یا طبیعی (مانند ربات) باشد.

عامل‌های هوشمند با استفاده از عملگرها، تغییر حالت برای خود یا محیط ایجاد می‌کنند.

بنابراین می‌توان گفت که عامل ما هوشمند (intelligent) دارای ادراک (perception) و یک سری اعمال (Actions) است.

علم هوش مصنوعی به طور غیرمستقیم مدل‌های مختلف هوشمندی را برای ما تعریف می‌کند و به ما می‌گوید که دنیای بدون هوشمندی چگونه است و هوشمندی چگونه کار می‌کند.

تفاوت ساختن و مدل‌سازی در ساختن، از خلاقیت استفاده می‌کنیم و چیزی را از نو می‌سازیم، اما در مدل‌سازی، از موجودات و پدیده‌های طبیعی الهام می‌گیریم.

تست تورینگ آزمونی برای سنجش توانایی یک ماشین در شبیه‌سازی رفتار هوشمندانه انسان، به ویژه در مکالمه، است. آلن تورینگ، دانشمند برجسته کامپیوتر، این تست را در سال 1950 معرفی کرد.

در این تست، سه نفر حضور دارند: بازیکن A که یک انسان، بازیکن B که ماشینی شبیه‌ساز گفتار انسان، و داور که یک انسان است. داور با هر دو بازیکن به طور جداگانه مکالمه می‌کند و سپس باید حدس بزند که کدام یک انسان است.

هدف تست تورینگ تعیین این است که آیا ماشین‌ها می‌توانند مانند انسان‌ها فکر کنند و صحبت کنند یا خیر. اگر ماشینی بتواند داور را فریب دهد یا در تشخیص او تردید ایجاد کند، آن ماشین "هوشمند" تلقی می‌شود.

عامل‌های هوشمند به دو دسته منطقی (منطقه‌گرا) و انسان‌گرا تقسیم می‌شوند. عامل‌های منطقی در هر لحظه منطقی‌ترین کار را انجام می‌دهند، در حالی که عامل‌های انسان‌گرا به دنبال مدل‌سازی رفتار و کردار انسان هستند.

حل مسئله مهمترین وجه هر علمی است. علم با یک مسئله شروع می‌شود و با حل آن به پایان می‌رسد. علم هوش مصنوعی نیز به حل مسائل مختلف در حوزه‌های مختلف می‌پردازد.

انواع مسائل در علم هوش مصنوعی

1. جستجو

2. ارضای محدودیت

3. مسائل بازنمایی دانش

4. مسائل یادگیری

مسائل جستجو: پیدا کردن راهحل از نقاط شروع تا نقطه هدف

انواع مسائل جستجو:

غیر آگاهانه: عاملی در محیط قرار داده میشود و هیچ اطلاعاتی به آن داده نمیشود.

آگاهانه: عاملی در محیط قرار داده میشود و اطلاعاتی به آن داده میشود.

مسائل ارضای محدودیت: مسائلی که با رفع یک سری محدودیت، خود به خود به هدف میرسیم.

مسائل بازنمایی دانش: نحوه نشان دادن دانش کسب شده ابزار ما برای نشان دادن دانش: منطق

یادگیری: مدل سازی یادگیری با استفاده از الگوریتمهای یادگیری

انواع مسائل یادگیری:

1. یادگیری ماشین

2. یادگیری عمیق

3. یادگیری تقویتی

الگوریتم (BM) (موزه بریتانیا) الگوریتمی است که برای یافتن تمام راه حل های ممکن یک مسئله، تمام حالت های ممکن را امتحان می کند. این الگوریتم حتی پس از یافتن جواب، جستجو را ادامه می دهد تا زمانی که تمام حالت های ممکن را بررسی کند.

مزیت این الگوریتم این است که تضمین می کند که تمام راه حل های ممکن را پیدا می کند. با این حال، این الگوریتم می تواند بسیار پرهزینه و زمان بر باشد، به خصوص برای مسائل پیچیده با تعداد زیادی حالت ممکن.

امروزه از الگوریتم های (Brute Force) بیشتر برای رمز شکنی استفاده می شود

الگوریتم Brute Force ساده ترین و ابتدایی ترین الگوریتمی است که برای حل یک مسئله می توان از آن استفاده کرد. این الگوریتم بر قدرت محاسباتی تکیه می کند و به دنبال بهینه سازی نیست رویکرد Brute Force ساده و تکراری است. این الگوریتم اولین رویکردی است که با دیدن مشکل به ذهن می رسد و تمام امکانات موجود برای حل مشکل را امتحان می کند.

مزیت الگوریتم Brute Force این است که تضمین می کند که به جواب می رسد (در صورت وجود). با این حال، این الگوریتم می تواند بسیار پرهزینه و زمان بر باشد، به خصوص برای مسائل پیچیده با تعداد زیادی حالت ممکن

الگوریتم جستجوی اول عمق (DFS)

الگوریتم DFS ایده ای را ارائه می دهد که در آن به عمق گراف حرکت می کنیم. در علوم کامپیوتر، عمق را اغلب از سمت چپ گراف در نظر می گیریم و به دلیل استفاده از حروف الفبا، گره ای که حروف الفبای کمتری دارد را انتخاب می کنیم.

اما مشکل اصلی آن این است که ممکن است به گره های بیخودی دسترسی داشته باشد و در صورتی که حجم گراف یا درخت بسیار بزرگ باشد، زمان زیادی طول می کشد.

برای حل این مشکل، محدودیتی را برای عمق DFS ارائه می دهیم. به عبارت دیگر، الگوریتم DFS با عمق محدود، مانند DFS با  $d=2$ ، به وجود آمده است. این به این معنی است که تا عمق 2 در گراف حرکت می کنیم و اگر به هدف نرسیدیم، باز می گردیم و از راه دیگری سعی می کنیم. اگر به عمق 2 رفتیم و هدف را پیدا نکردیم، الگوریتم غیر می کند و جوابی را ارائه نمی دهد.

تکرارهای اجرای DFS با عمق محدود با ++D ساخته می شوند. تفاوت آن با DFS این است که هنگامی که به جواب می رسد، عملیات را به پایان می رساند و ادامه مسیر را ادامه نمی دهد.

الگوریتم جستجوی اول سطح (BFS) (یک الگوریتم پیمایش گراف یا درخت است که به صورت سطحی در گراف حرکت می کند.

الگوریتم BFS از یک نقطه شروع (معمولاً ریشه) شروع می کند و تمام گره های همسایه را در سطح اول بررسی می کند. سپس، به سطح بعدی می رود و تمام گره های همسایه را که هنوز بررسی نشده اند، بررسی می کند. این کار را تا زمانی که تمام گره ها را بررسی کند، ادامه می دهد.

الگوریتم A\* مسیری را انتخاب می کند که مجموع هزینه فعلی و هزینه تخمینی آن کمترین باشد. این الگوریتم را می توان به عنوان یک الگوریتم حریمانه محلی در نظر گرفت که در هر مرحله به دنبال بهترین گره بعدی برای جستجو است.

\* الگوریتم محلی یک الگوریتم است که فقط به یک قسمت خاص از یک مسئله بزرگ نگاه می کند. این الگوریتم ها سعی می کنند با اعمال تغییرات کوچک در یک قسمت از مسئله، به یک جواب قابل قبول برسند.

الگوریتم غیر محلی یک الگوریتم است که به کل یک مسئله بزرگ نگاه می کند و تلاش می کند تا یک جواب بهینه را پیدا کند. این الگوریتم ها برخلاف الگوریتم های محلی، به یک قسمت کوچک از مسئله محدود نمی شوند.