

جواب کوییز 2 هوش مصنوعی

1. چهار مرحله کلی برای حل یک مساله در هوش

مصنوعی را با مثال شهر رومانی شرح دهید:

حل یک مسئله در هوش مصنوعی معمولاً به چندین

مرحله تقسیم می شود. در زیر، چهار مرحله کلی برای

حل یک مساله در هوش مصنوعی را با مثال شهر

رومانی شرح می دهیم:

تعریف مسئله: در این مرحله، مسئله به طور دقیق

تعریف می شود. باید مشخص شود که مسئله دقیقاً چه

هدفی دارد و چه محدودیت هایی دارد. به عنوان مثال،

می توانیم مسئله مسیریابی در شهرهای رومانی را در

نظر بگیریم که هدف آن پیدا کردن کوتاه ترین مسیر

بین دو شهر است و محدودیت ها شامل مسافت های

بین شهرها و محدودیت های مربوط به طول و زمان

سفر است. طراحی مدل: در این مرحله، باید یک مدل

ریاضی برای حل مسئله طراحی کنیم. این مدل معمولاً

با استفاده از مفاهیم ریاضی و الگوریتم های مختلف

ایجاد می شود. در مثال مسیریابی در شهرهای رومانی،

می توانیم از یک گراف استفاده کنیم که شهرها را به

عنوان گره ها و مسافت های بین شهرها را به عنوان

یال ها نشان می دهد. سپس با استفاده از الگوریتم های

جستجو، می توانیم کوتاه ترین مسیر را پیدا کنیم.

پیاده سازی و اجرا: در این مرحله، مدل طراحی شده

را پیاده سازی می کنیم و بر روی داده های واقعی

اجرا می کنیم. می توانیم از زبان برنامه نویسی مانند

Python یا Java برای پیاده سازی استفاده کنیم.

در مثال مسیریابی در شهرهای رومانی، باید گراف و

مسافت های بین شهرها را به صورت داده های واقعی

وارد کنیم و الگوریتم مسیریابی را بر روی آنها اجرا

کنیم. ارزیابی و بهبود: در این مرحله، عملکرد مدل

و الگوریتم را ارزیابی می کنیم و در صورت نیاز، آنها

را بهبود می بخشیم. این مرحله ممکن است نیاز به

تکرار و تحلیل داده های خروجی و تغییر در طراحی

داشته باشد. به عنوان مثال، می توانیم در مساله

مسیریابی، عملکرد الگوریتم را با تغییر پارامترها یا



مدل داشته باشد. به عنوان مثال، می‌توانیم در مساله مسیریابی، عملکرد الگوریتم را با تغییر پارامترها یا استفاده از الگوریتم‌های بهتر بهبود دهیم. 2. انواع مساله را نام ببرید و شرح مختصری از هر یک با ذکر یک مثال بیان کنید؟

مساله جستجو و کاوش: در این نوع مساله، هدف پیدا کردن یک راه‌حل مناسب از میان یک فضای جستجو است. مثالی از این نوع مساله می‌تواند جستجوی مسیر کوتاه‌ترین بین دو شهر در یک نقشه باشد. مساله بهینه‌سازی: در این نوع مساله، هدف پیدا کردن بهترین راه‌حل بر اساس یک معیار است. به عنوان مثال، مساله برنامه‌ریزی تولید بهینه برای توزیع محصولات به فروشگاه‌ها. مساله تصمیم‌گیری: در این نوع مساله، هدف انتخاب بهترین اقدام بر اساس ورودی‌های داده شده است. به عنوان مثال، مساله تصمیم‌گیری برای خرید سهام بر اساس اطلاعات بازار و تحلیل ریسک. مساله پردازش زبان طبیعی: در این نوع مساله، هدف درک و تفسیر زبان طبیعی توسط سامانه‌های مصنوعی است. به عنوان مثال، سیستم‌های تشخیص گفتار و ترجمه ماشینی. مساله یادگیری ماشین: در این نوع مساله، هدف آموزش سامانه‌های مصنوعی برای یادگیری از داده‌ها و بهبود عملکرد آنها است. به عنوان مثال، سیستم توصیه‌گر بر اساس سابقه خرید کاربران. مساله شناخت تصویر: در این نوع مساله، هدف تشخیص و تفسیر تصاویر توسط سامانه‌های مصنوعی است. به عنوان مثال، سیستم تشخیص چهره در عکس‌ها. مساله منطق و استدلال: در این نوع مساله، هدف استدلال بر اساس قواعد و منطق است. به عنوان مثال، سیستم‌های تشخیص تقلب در مالیات بر اساس الگوریتم‌های منطقی. این تنها چند نمونه از انواع مختلف مسائل در هوش مصنوعی است و هر کدام از این مسائل به روش‌ها و الگوریتم‌های مختلفی نیاز دارند. 3. مسئله 8 وزیر را با دو روش فرموله سازی (مثال n وزیر را طوری در صفحه شطرنج بگذارید همدیگر را تهدید نکنند)؟



روش اول: استفاده از مساله فرمولاسیون CSP (محدودیت برنامه‌ریزی) یک روش برای فرموله‌سازی این مسئله استفاده از فرمولاسیون CSP (محدودیت برنامه‌ریزی) است. در این روش، هر خانه در صفحه شطرنج به عنوان یک متغیر در نظر گرفته می‌شود و مقدار هر متغیر می‌تواند عدد 1 تا N (تعداد وزیرها) باشد. سپس با استفاده از محدودیت‌ها، تهدیدهای ممکن بین وزیرها را مدل می‌کنیم. برای مثال، محدودیتی می‌تواند باشد که دو وزیر نباید در یک ستون، یک سطر یا قطری باشند. روش دوم: الگوریتم ژنتیک روش دیگری برای فرموله‌سازی این مسئله استفاده از الگوریتم ژنتیک است. در این روش، هر حالت (موقعیت وزیرها در صفحه شطرنج) را با یک کروموزوم و هر ویژگی (موقعیت هر وزیر) را با یک ژن نمایش می‌دهیم. سپس با استفاده از عملگرهای تکاملی مانند انتخاب، ترکیب، و جهش، بهترین حالت‌ها را پیدا کرده و تا جای ممکن تهدیدها را حذف می‌کنیم. به این ترتیب، ترکیبی از وزیرها را در صفحه شطرنج بدون تهدیدها پیدا می‌کنیم. برای مثال، فرض کنید بخواهیم 8 وزیر را در صفحه شطرنج 8x8 قرار دهیم. این مسئله به این صورت فرموله می‌شود:

18:06 ✓

هر وزیر می‌تواند در یک سطر از 1 تا 8 قرار بگیرد.  
هر وزیر می‌تواند در یک ستون از 1 تا 8 قرار بگیرد.  
هر وزیر می‌تواند در یک قطر اصلی یا فرعی از 1 تا 8 قرار بگیرد. جستجوی درختی را با ذکر یک مثال شرح دهید؟

در این روش، یک درخت جستجوی ساخته می‌شود که هر گره آن یک حالت ممکن در مسئله را نمایش می‌دهد و یال‌های درخت نشان‌دهنده روابط بین حالت‌ها هستند. هدف جستجوی درختی پیدا کردن مسیری است که به یک حالت هدف منتهی می‌شود. برای مثال، می‌توانیم یک مسئله جستجوی کوتاه‌ترین مسیر در یک نمودار جهانی حل کنیم. در این مسئله،



هر وزیر می‌تواند در یک سطر از 1 تا 8 قرار بگیرد.  
هر وزیر می‌تواند در یک ستون از 1 تا 8 قرار بگیرد.  
هر وزیر می‌تواند در یک قطر اصلی یا فرعی از 1 تا 8 قرار بگیرد. جستجوی درختی را با ذکر یک مثال شرح دهید؟

در این روش، یک درخت جستجوی ساخته می‌شود که هر گره آن یک حالت ممکن در مسئله را نمایش می‌دهد و یال‌های درخت نشان‌دهنده روابط بین حالت‌ها هستند. هدف جستجوی درختی پیدا کردن مسیری است که به یک حالت هدف منتهی می‌شود. برای مثال، فرض کنید می‌خواهیم یک مسئله جستجوی کوتاه‌ترین مسیر در یک نمودار جهانی حل کنیم. در این مسئله، هر گره از درخت جستجو یک شهر را نمایش می‌دهد و یال‌ها نشان‌دهنده مسیرهای ممکن بین شهرها هستند. همچنین، هر گره شامل اطلاعاتی مانند مسافت تا شهرهای دیگر و هزینه سفر است. با شروع از گره شهر مبدأ، می‌توانیم با استفاده از الگوریتم جستجوی درختی مانند الگوریتم جستجوی سطح اول (BFS) یا الگوریتم جستجوی عمق اول (DFS)، درخت جستجو را پیمایش کنیم. هر بار که به یک گره جدید می‌رسیم، بررسی می‌کنیم که آیا این گره حالت هدف است یا خیر. اگر گره حالت هدف باشد، مسیری که تا این گره رسیده‌ایم را پیدا کرده‌ایم. در غیر این صورت، ادامه می‌دهیم و به تمام گره‌های فرزند این گره می‌رویم و جستجو را در آنها ادامه می‌دهیم. به عنوان مثال، فرض کنید می‌خواهیم کوتاه‌ترین مسیر بین شهر A و شهر B را در یک نمودار جهانی پیدا کنیم. درخت جستجوی ساخته شده از شهر A شروع می‌شود و با پیمایش گره‌های فرزند، به تمام شهرهای ممکن در نمودار می‌رسیم. هر بار که به گره شهر B می‌رسیم، مسیری که تا اینجا پیموده‌ایم را پیدا کرده‌ایم. این مسیر می‌تواند از تعدادی شهر و مسیرهای مختلف تشکیل شده باشد که با استفاده از جستجوی درختی، دست پیدا می‌کنیم.

نقشه فضای حالت و Fringe را تعریف کنید؟



5. فضای حالت و Fringe را تعریف کنید؟

فضای حالت (State Space) در مفهوم عمومی، مجموعه‌ای است از تمام حالت‌های ممکن که یک سامانه می‌تواند به آن‌ها وارد شود یا در آن‌ها قرار بگیرد. در حوزه هوش مصنوعی و جستجوی مسئله، فضای حالت به معنای مجموعه‌ای از تمام حالت‌های ممکن است که یک مسئله را می‌توان با آن‌ها مدل کرد. هر حالت در فضای حالت می‌تواند یک توصیف کامل از وضعیت سامانه در یک زمان خاص باشد. به عنوان مثال، در یک مسئله جستجوی مسیر، هر حالت می‌تواند مکان واقعی یک عامل در یک زمان خاص باشد. Fringe یا لبه (Fringe) در جستجوی مسئله، مجموعه‌ای است از گره‌ها که در حالت‌هایی قرار دارند که هنوز بررسی نشده‌اند و در مراحل بعدی ممکن است بررسی شوند. به عبارت دیگر، Fringe شامل گره‌هایی است که در حالت‌های مجاور به حالت‌های فعلی قرار دارند. در جستجوی مسئله، از Fringe برای نگهداری گره‌هایی که باید در مراحل بعدی بررسی شوند و به آن‌ها دسترسی داشته باشیم، استفاده می‌شود. در هر مرحله از جستجو، یک گره از Fringe برای بررسی انتخاب می‌شود و به عنوان گره فعلی در نظر گرفته می‌شود. به عنوان مثال، در جستجوی کوتاه‌ترین مسیر بین دو شهر، فضای حالت شامل تمام شهرهای ممکن است که می‌توان به آن‌ها رسید. Fringe در این مسئله مجموعه‌ای از گره‌هایی است که در حالت‌های مجاور به حالت فعلی قرار دارند و هنوز بررسی نشده‌اند. در هر مرحله، یک گره از Fringe انتخاب می‌شود و بررسی می‌شود که آیا به حالت هدف رسیده است یا خیر. اگر به حالت هدف رسیده باشد، مسیری که تا این گره طی شده است، پیدا می‌شود. در غیر این صورت، گره‌های فرزند این گره به Fringe اضافه می‌شوند تا در مراحل بعدی بررسی شوند.

جستجوی ناآگاهانه را تعریف کنید و انواع آن را نام ببر.

جستجوی ناآگاهانه (Uninformed Search) یک



جستجوی ناآگاهانه (Uninformed Search) یک روش جستجو در هوش مصنوعی است که در آن الگوریتم‌ها بر اساس اطلاعات کمتری درباره فضای حالت و مسئله عمل می‌کنند. در جستجوی ناآگاهانه، الگوریتم‌ها بر اساس قوانین عمومی جستجو عمل می‌کنند و به صورت سیستماتیک و بدون اطلاعات خاص درباره مسئله، حالات را بررسی می‌کنند. انواع جستجوی ناآگاهانه عبارتند از: جستجوی سطح اول (Breadth-First Search): در این روش، ابتدا گره‌های مجاور به گره فعلی بررسی می‌شوند و سپس به گره‌های مجاور به آن‌ها می‌رسیم. این جستجو به صورت وسیع و پهن از درخت جستجو پیش می‌رود و تمام گره‌های هر سطح را در ابتدا بررسی می‌کند. جستجوی عمق اول (Depth-First Search): در این روش، ابتدا یک گره را بررسی می‌کنیم و سپس به گره فرزند آن می‌رسیم. این روش به صورت عمقی در درخت جستجو پیش می‌رود و تا زمانی که دیگر گره‌ای برای بررسی نباشد، به صورت پیوسته به سمت پایین حرکت می‌کند. جستجوی یکنواخت (Uniform-Cost Search): در این روش، هر گره به عنوان نقطه شروع در نظر گرفته می‌شود و هزینه حرکت به هر گره در نظر گرفته می‌شود. الگوریتم در هر مرحله گره با کمترین هزینه را بررسی می‌کند و به سمت آن حرکت می‌کند.

18:06 ✓

جستجوی عمق محدود (Limited-Depth Search): در این روش، محدودیتی برای عمق جستجو تعیین می‌شود و تا آن حد عمق، جستجو انجام می‌شود. این روش می‌تواند به صورت محدودیت زمانی یا محدودیت عمق باشد. جستجوی عمق تدریجی (Iterative Deepening Search): این روش ترکیبی از جستجوی عمق اول و جستجوی عمق محدود است. ابتدا با عمق یک شروع می‌کند و در هر مرحله عمق افزایش می‌دهد تا به حل مسئله برسد. 7. الگوریتمی



جستجوی عمق محدود (Limited-Depth Search):

در این روش، محدودیتی برای عمق جستجو تعیین می‌شود و تا آن حد عمق، جستجو انجام می‌شود.

این روش می‌تواند به صورت محدودیت زمانی

یا محدودیت عمق باشد. جستجوی عمق تدریجی

(Iterative Deepening Search): این روش ترکیبی

از جستجوی عمق اول و جستجوی عمق محدود است.

در ابتدا با عمق یک شروع می‌کند و در هر مرحله عمق

را افزایش می‌دهد تا به حل مسئله برسد. 7. الگوریتمی

که از لحاظ زمانی از مرتبه جستجوی اول سطح است

ولی از لحاظ پیچیدگی حافظه از مرتبه جستجوی اول

عمق می‌باشد کدام است، شرح دهید؟

الگوریتمی که از لحاظ زمانی از مرتبه جستجوی

اول سطح (Breadth-First Search) است ولی

از لحاظ پیچیدگی حافظه از مرتبه جستجوی اول

عمق (Depth-First Search) می‌باشد، الگوریتم

جستجوی بهینه‌شده (Optimized Search) نامیده

می‌شود. در الگوریتم جستجوی بهینه‌شده، از هر دو

روش جستجوی اول سطح و جستجوی عمق اول

بهره‌برداری می‌شود. الگوریتم به این صورت عمل

می‌کند که ابتدا با استفاده از جستجوی اول سطح،

گره‌های مجاور به گره فعلی را بررسی می‌کند و در

صورتی که به هدف نرسیده باشد، به گره‌های مجاور

به آن‌ها می‌رسد. این روند تا زمانی ادامه می‌یابد که به

یک عمق مشخصی برسد. در این مرحله، الگوریتم به

جستجوی عمق اول تغییر می‌دهد و از روش جستجوی

عمق اول برای بررسی گره‌های مجاور به گره‌های فعلی

استفاده می‌کند. این کار ادامه می‌یابد تا زمانی که به

هدف برسد یا تمام گره‌ها را بررسی کند. با استفاده

از این الگوریتم، زمان اجرای جستجوی اول سطح را

داریم که به صورت وسیع و پهن از درخت جستجو

پیش می‌رود و تمام گره‌های هر سطح را در ابتدا

بررسی می‌کند. اما با تغییر به جستجوی عمق اول،

بدگی حافظه نیز از مرتبه جستجوی اول عمق

شود. بنابراین الگوریتم جستجوی عمق اول تنها یک



پیچیدگی حافظه نیز از مرتبه جستجوی اول عمق می‌شود، زیرا در الگوریتم جستجوی عمق اول، تنها یک مسیر را در حافظه نگه می‌دارد و در صورت نرسیدن به هدف، به گره‌های فرزند آن می‌رود.

8. کارایی انواع جستجوهای نا آگاهانه را بر حسب چهار پارامتر کامل بودن، بهینگی، پیچیدگی زمانی و فضایی بیان کنید؟

انواع جستجوهای نا آگاهانه به ترتیب کامل بودن، بهینگی، پیچیدگی زمانی و پیچیدگی فضایی مورد بررسی قرار می‌گیرند:

کامل بودن (Completeness): یک الگوریتم جستجوی نا آگاهانه کامل است اگر در صورت وجود پاسخ، همواره به پاسخ برسد. به عبارت دیگر، الگوریتم باید تمام مسیرهای ممکن را بررسی کند تا به پاسخ برسد. الگوریتم جستجوی سطح اول (Breadth-First Search) و جستجوی عمق اول (Depth-First Search) از نوع جستجوهای نا آگاهانه هستند و برای مسائل محدود شده کامل هستند. بهینگی (Optimality): یک الگوریتم جستجوی نا آگاهانه بهینه است اگر در صورت وجود پاسخ، به پاسخ بهینه برسد. به عبارت دیگر، الگوریتم باید بهترین پاسخ ممکن را ارائه کند. الگوریتم جستجوی سطح اول و جستجوی عمق اول بهینه نیستند، زیرا ممکن است به پاسخ غیر بهینه برسند. پیچیدگی زمانی (Time Complexity): پیچیدگی زمانی الگوریتم جستجوی نا آگاهانه به تعداد عملیاتی که در طول اجرا انجام می‌دهد بستگی دارد. الگوریتم جستجوی سطح اول و جستجوی عمق اول هر دو از مرتبه  $O(b^d)$  هستند، که در آن  $b$  تعداد حالت‌های ممکن در هر گره و  $d$  عمق مسئله را نشان می‌دهد. الگوریتم جستجوی سطح اول معمولاً زمان بیشتری نسبت به جستجوی عمق اول می‌برد زیرا تمام گره‌های هر سطح را در ابتدا بررسی می‌کند. پیچیدگی فضایی (Space Complexity): پیچیدگی فضایی الگوریتم جستجوی نا آگاهانه به لحظه‌ای که در طول اجرا استفاده می‌شود بستگی



فضایی بیان کنید؟

انواع جستجوهای نا آگاهانه به ترتیب کامل بودن ، بهینگی ، پیچیدگی زمانی و پیچیدگی فضایی مورد بررسی قرار می گیرند:

کامل بودن (Completeness): یک الگوریتم جستجوی نا آگاهانه کامل است اگر در صورت وجود پاسخ، همواره به پاسخ برسد. به عبارت دیگر، الگوریتم باید تمام مسیرهای ممکن را بررسی کند تا به پاسخ برسد. الگوریتم جستجوی سطح اول (Breadth-First Search) و جستجوی عمق اول (Depth-First Search) از نوع جستجوهای نا آگاهانه هستند و برای مسائل محدود شده کامل هستند. بهینگی (Optimality): یک الگوریتم جستجوی نا آگاهانه بهینه است اگر در صورت وجود پاسخ، به پاسخ بهینه برسد. به عبارت دیگر، الگوریتم باید بهترین پاسخ ممکن را ارائه کند. الگوریتم جستجوی سطح اول و جستجوی عمق اول بهینه نیستند، زیرا ممکن است به پاسخ غیر بهینه برسند. پیچیدگی زمانی (Time Complexity): پیچیدگی زمانی الگوریتم جستجوی نا آگاهانه به تعداد عملیاتی که در طول اجرا انجام می دهد بستگی دارد. الگوریتم جستجوی سطح اول و جستجوی عمق اول هر دو از مرتبه  $O(b^d)$  هستند، که در آن  $b$  تعداد حالت های ممکن در هر گره و  $d$  عمق مسئله را نشان می دهد. الگوریتم جستجوی سطح اول معمولاً زمان بیشتری نسبت به جستجوی عمق اول می برد زیرا تمام گره های هر سطح را در ابتدا بررسی می کند. پیچیدگی فضایی (Space Complexity): پیچیدگی فضایی الگوریتم جستجوی نا آگاهانه به حافظه ای که در طول اجرا استفاده می شود بستگی دارد. الگوریتم جستجوی سطح اول به مقدار  $O(b^d)$  حافظه نیاز دارد زیرا تمام گره های هر سطح را در حافظه نگه می دارد. الگوریتم جستجوی عمق اول به مقدار  $O(bd)$  حافظه نیاز دارد زیرا تنها یک مسیر را در حافظه نگه می دارد.