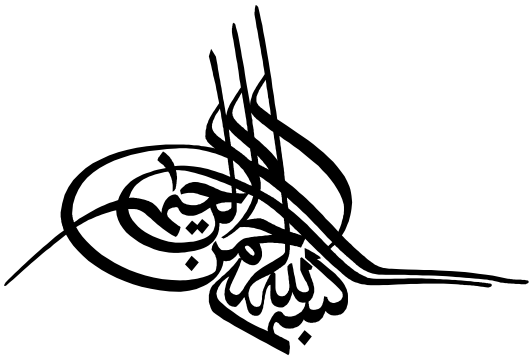


هوش مصنوعی

سری کتاب‌های کمک آموزشی کارشناسی ارشد

مجموعه مهندسی کامپیوتر و فناوری اطلاعات

مؤلف: گروه مولفان

ویراستار علمی: یاسر برجسته

سرشناسه	: گروه مولفان
عنوان	: هوش مصنوعی
مشخصات نشر	: تهران : مشاوران صعود ماهان، ۱۴۰۱
مشخصات ظاهری	: ۱۶۹ ص
فروست	: سری کتاب‌های کمک آموزشی کارشناسی ارشد
شابک	: ۹۷۸-۶۰۰-۴۵۸-۸۱۷-۱
وضعیت فهرست نویسی	: فیپای مختصر
یادداشت	:این مدرک در آدرس http://opac.nlai.ir قابل دسترسی است.
شناسه افزوده	: یاسر برجسته ، ۱۳۶۴
شماره کتابشناسی ملی	: ۳۷۷۴۱۶۹

نام کتاب: ..... هوش مصنوعی

مولف:..... گروه مولفان

ویراستار علمی .....یاسر برجسته

مدیر تولید محتوی: .....سمیه بیگی

ناشر:.....مشاوران صعود ماهان

نوبت و تاریخ چاپ: .....اول / ۱۴۰۱

تیراژ: .....۱۰۰۰ نسخه

قیمت:..... ۲/۰۹۰/۰۰۰ ریال

شابک: ..... ISBN: ۹۷۸-۶۰۰-۴۵۸-۸۱۷-۱

انتشـــارات مشاوران صعود ماهان: خیابان ولیعصر، بالاتر از تقـاطع مطهری،

روبروی قنادی هتل بزرگ تهـــران، جنب بانک ملی، پلاک ۲۰۵۰

تلفن: ۴- ۸۸۱۰۰۱۱۳

سخن ناشر

«ن والقلم و ما یسطرون»

کلمه نزد خدا بود و خدا آن را با قلم بر ما نازل کرد.

به پاس تشکر از چنین موهبت الهی، موسسه ماهان درصدد برآمده است تا در راستای انتقال دانش و مفاهیم با کمک اساتید مجرب و مجموعه کتب آموزشی خود برای شما داوطلبان ادامه تحصیل در مقطع کارشناسی ارشد گام موثری بردارد. امید است تلاش‌های خدمتگزاران شما در این موسسه پایه‌گذار گام‌های بلند فردای شما باشد.

مجموعه کتاب‌های کمک آموزشی ماهان به‌منظور استفاده داوطلبان کنکور کارشناسی ارشد سراسری و آزاد تالیف شده‌اند. در این کتاب‌ها سعی کرده‌ایم با بهره‌گیری از تجربه اساتید بزرگ و کتب معتبر داوطلبان را از مطالعه کتاب‌های متعدد در هر درس بی‌نیاز کنیم.

دیگر تالیفات ماهان برای سایر دانشجویان به‌صورت ذیل می‌باشد.

- **مجموعه کتاب‌های ۸ آزمون:** شامل ۵ مرحله کنکور کارشناسی ارشد ۵ سال اخیر به همراه ۳ مرحله آزمون تالیفی ماهان همراه با پاسخ تشریحی می‌باشد که برای آشنایی با نمونه سوالات کنکور طراحی شده است. این مجموعه کتاب‌ها با توجه به تحلیل ۳ ساله اخیر کنکور و بودجه‌بندی مباحث در هریک از دروس، اطلاعات مناسبی جهت برنامه‌ریزی درسی در اختیار دانشجو قرار می‌دهد.
- **مجموعه کتاب‌های کوچک:** شامل کلیه نکات کاربردی در گرایش‌های مختلف کنکور کارشناسی ارشد می‌باشد که برای دانشجویان جهت جمع‌بندی مباحث در ۲ ماهه آخر قبل از کنکور مفید می‌باشد.

بدین‌وسیله از مجموعه اساتید، مولفان و همکاران محترم خانواده بزرگ ماهان که در تولید و به‌روزرسانی تالیفات ماهان نقش موثری داشته‌اند، صمیمانه تقدیر و تشکر می‌نماییم.

دانشجویان عزیز و اساتید محترم می‌توانند هرگونه انتقاد و پیشنهاد درخصوص تالیفات ماهان را از طریق سایت ماهان به آدرس mahan.ac.ir با ما در میان بگذارند.

موسسه آموزش عالی آزاد ماهان

سخن مؤلف

هوش مصنوعی رشته گسترده‌ای است که جزو علوم جدید محسوب می‌شود و در حال حاضر شاخه‌های بسیار متنوعی از قبیل منطق، احتمال، استدلال، یادگیری، ادراکات، پرولوگ، مهندسی هستان‌شناسی، مهندسی دانش و غیره می‌باشند به طوری که اعمال هوشمند را به صورت خودکار ارئه می دهد و در نتیجه با هر کار هوشمندانه انسان در ارتباط است.

کتاب حاضر، حاصل سال‌ها تدریس در موسسات و مراکز آموزش عالی و مطالعه منابع گوناگون می‌باشد.

این کتاب منبع بسیار مناسبی برای داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر، مهندسی فناوری اطلاعات و مکاترونیک است.

کتاب حاضر شامل نه فصل است که در پایان هر فصل تست‌های کنکور سالهای گذشته با پاسخ تشریحی آورده شده است، همچنین با توجه به مثال‌ها و تمرین‌های حل شده و متن روان و شیوای درس‌نامه این کتاب برای دانشجویان دوره کارشناسی کامپیوتر و فناوری اطلاعات در دانشگاه‌های سراسری، آزاد، غیرانتفاعی و پیام‌نور به‌عنوان یک کتاب کامل آموزشی مناسب است.

در انتها از شما خواننده گرامی تقاضا دارم در صورت مشاهده هرگونه اشکال و یا پیشنهاد اینجانب را مطلع سازید.

یاسر برجسته

آدرس پستی:yaser.barjesteh@gmail.com

فهرست

عنوان	صفحه

فصل اول: تاریخچه هوش مصنوعی و بررسی عامل‌های هوشمند ۹
مبانی هوش مصنوعی ۱۱
تاریخچه هوش مصنوعی ۱۱
عامل‌های هوشمند ۱۲
ویژگی‌های عامل‌ها ۱۳
دنیای ماشین جاروبرقی ۱۳
PEAS: مقیاس کارآیی، محیط، اثرگرها و حسگرها ۱۴
ساختار عامل‌های هوشمند ۱۵
برنامه‌های عامل (agent program) ۱۵
ماهیت عامل ۱۶
عامل واکنشی ساده ۱۶
عامل‌های واکنشی مبتنی بر مدل ۱۷
عامل مبتنی بر هدف ۱۸
عامل مبتنی بر سودمندی ۱۸
عامل‌های یادگیرنده ۱۹
محیط‌ها ۱۹
برنامه عمومی محیط ۲۱
تست تورینگ ۲۲
مباحث مرتبط با هوش مصنوعی ۲۳
سوالات چهارگزینه‌ای و پاسخنامه سراسری فصل اول ۲۶
فصل دوم: بررسی مسائل مختلف و ویژگی‌های آنها ۲۹
حل مسأله ۳۰
چهار گام اساسی برای حل مسائل ۳۰
قوانین حاکم بر fact ها ۳۱
استراتژی کنترل (راه‌حل) ۳۲
مسئله مبلغ‌های مذهبی و آدم خواران ۳۲
مسئله سکه دو ریالی و پنج ریالی ۳۳
مسئله ۸ وزیر ۳۳
مسئله ۴ وزیر ۳۳
انواع مسائل و ویژگی‌های آنها ۳۴

جهت‌های مختلف استدلال ۳۴
معیارهای کارایی استراتژی کنترل (جستجو) ۳۴
انواع مسائل ۳۷
مسائل دنیای واقعی ۳۹
سوالات چهارگزینه‌ای و پاسخنامه سراسری فصل دوم ۴۰
فصل سوم: جستجوی ناآگاهانه ۴۱
جستجوی عرضی یا اول سطح (breadth-first search) ۴۲
الگوریتم bfs ۴۲
جستجو با هزینه یکنواخت (Uniform cost search) ۴۳
الگوریتم جستجو با هزینه یکنواخت ۴۳
روش جستجوی عمقی یا اول- عمق (Depth-first search) ۴۴
الگوریتم جستجویDFS ۴۵
جستجوی عمقی محدود شده(Depth-limited search) ۴۵
الگوریتم جستجوی عمقی محدود شده ۴۶
جستجوی عمقی تکرار شونده (Iterative deepening search) ۴۶
جستجوی دو طرفه (Bidirectional search) ۴۶
جستجو با اطلاعات جزئی ۴۸
مسأله عامل بدون حسگر(Sensorless problem) ۴۸
مسائل احتمالی(Contingency problem) ۴۹
مسائل اکتشافی (exploration) ۴۹
سوالات چهارگزینه‌ای و پاسخنامه سراسری فصل سوم ۵۰
فصل چهارم: جستجوی آگاهانه (informed search) ۵۳
جستجوی اول بهترین (Best-first search) ۵۴
الگوریتم کلی جستجوی آگاهانه ۵۴
جستجوی حریصانه (Greedy search) ۵۴
ویژگی‌های روش حریصانه ۵۵
Beam search ۵۶
جستجوی*A ۵۷
تابع مکاشفه‌ای قابل قبول (admissible beuristic function) ۵۷
الگوریتم *A ۵۷
یکنوایی (Monotonicity) ۵۸
جستجوی اکتشافی با حافظه محدود(IDA*:Iterative Deepening A*) ۵۹
جستجوی اول- بهترین بازگشتی (RBFS) ۶۰
SMA(Simplified memory Bounded A) ۶۲
ویژگی‌های روش جستجو *SMA ۶۲
الگوریتم*SMA ۶۲
الگوریتم‌های اصلاح تکراری(Iterative Improvement) ۶۳

جستجوی تپه‌نوردی(Hill-climbing) ۶۳
جستجوی التهاب گداخت شبیه‌سازی شده(Simulated annealing) ۶۶
سوالات چهارگزینه‌ای و پاسخنامه سراسری فصل چهارم ۶۷
فصل پنجم: جستجوی رقابتی (تئوری بازی) ۸۱
استراتژی بهینه ۸۳
استراتژی (minimax) ۸۳
الگوریتم minimax در بازی‌های چند نفره ۸۵
هرس آلفا- بتا (Alpha-Beta pruning) ۸۶
درک بهتر الگوریتم هرس $\alpha-\beta$ ۸۷
الگوریتم minimax در بازی‌هایی که دارای عنصر شانس می‌باشند ۸۹
سوالات چهارگزینه‌ای و پاسخنامه سراسری فصل پنجم ۹۱
فصل ششم: مسائل ارضای محدودیت ۱۰۱
جستجوی عقب‌گرد برای مسائل ارضای محدودیت ۱۰۵
جستجوی بررسی پیشرو (Forward checking) ۱۰۷
پخش محدودیت ۱۰۷
الگوریتم عقب‌گرد هوشمند ۱۰۸
کاربرد جستجوهای محلی در محل مسائل ارضای محدودیت ۱۰۹
سوالات چهارگزینه‌ای و پاسخنامه سراسری فصل ششم ۱۱۱
فصل هفتم: عامل‌های منطقی ۱۱۵
منطق گزاره‌ای ۱۱۷
استنتاج ۱۱۸
هم ارزی منطقی ۱۱۹
اعتبار(Validity) ۱۲۰
قضیه استنتاج ۱۲۰
ارضا شدن (satisfiability) ۱۲۰
الگوهای استدلال استاندارد ۱۲۰
قانون (Modus ponens) ۱۲۰
قاعده حذف AND (And Elimination) ۱۲۱
Resolution ۱۲۱
الگوریتم Resolution ۱۲۲
سوالات چهارگزینه‌ای و پاسخنامه سراسری فصل هفتم ۱۲۵
فصل هشتم: منطق مرتبه اول ۱۲۹
نحو در منطق مرتبه اول ۱۳۱
ترم (term) ۱۳۲
جملات اتمی (atomic sentence) ۱۳۲
جملات پیچیده (complex sentence) ۱۳۳
سورها (quantifiers) ۱۳۳

سورهای تودرتو (Nested quatifiers) ۱۳۳
نقیض کردن سورها ۱۳۴
ادعاها و پرسش‌ها در منطق مرتبه اول ۱۳۴
اعداد ۱۳۵
مجموعه‌ها ۱۳۵
لیست‌ها ۱۳۶
سوالات چهارگزینه‌ای و پاسخنامه سراسری فصل هشتم ۱۳۷
فصل نهم: استنتاج در منطق مرتبه اول ۱۴۱
قانون انتزاع تعمیم‌یافته (GMP) ۱۴۳
یکسان‌سازی ۱۴۴
ذخیره و بازیابی در پایگاه دانش ۱۴۵
زنجیره‌سازی به جلو ۱۴۶
فراکردهای معین مرتبه اول ۱۴۶
الگوریتم زنجیره‌سازی به جلوی اولیه ۱۴۶
زنجیره‌سازی به جلوی کارا ۱۴۷
زنجیره‌سازی به جلوی افزایشی (Incremental forward chaining) ۱۴۸
زنجیره‌سازی به عقب ۱۴۹
فرم نرمال عطفی برای منطق مرتبه اول ۱۴۹
قانون استنتاج Resolution در منطق مرتبه اول ۱۵۰
تساوی ۱۵۱
سوالات چهارگزینه‌ای و پاسخنامه سراسری فصل نهم ۱۵۲
سوالات کنکور سراسری ۹۵ ۱۶۵
منابع ۱۶۹

فصل اول

تاریخچه هوش مصنوعی و بررسی عامل‌های هوشمند

- مبانی هوش مصنوعی
- تاریخچه هوش مصنوعی
- عامل‌های هوشمند
- برنامه‌های عامل
- ماهیت عامل
- عامل واکنشی ساده
- عامل‌های واکنشی مبتنی بر مدل
- عامل مبتنی بر هدف
- عامل مبتنی بر سودمندی
- عامل‌های یادگیرنده
- محیط‌ها
- تست تورینگ

تاریخچه هوش مصنوعی و بررسی عامل‌های هوشمند

موجود مصنوعی موجودی است که می‌خواهد مانند انسان رفتار کند بنابراین نیاز به هوش مصنوعی دارد. تعاریف متفاوتی برای هوش مصنوعی وجود دارند که به چهار دسته‌ای که در جدول زیر آمده‌اند، تقسیم می‌شوند.

سیستم‌هایی که منطقی فکر می‌کنند	سیستم‌هایی که مانند انسان فکر می‌کنند
سیستم‌هایی که منطقی عمل می‌کنند	سیستم‌هایی که مانند انسان عمل می‌کنند

تعاریفی که در چارچوب تقسیمات بالای جدول قرار می‌گیرند برفرآیند فکر و استدلال و تعاریف پایینی بر عمل و رفتار متمرکز شده‌اند. از طرف دیگر تعاریف سمت راست به مفهوم ایده‌آل هوشمندی یعنی معقول و منطقی بودن و تعاریف سمت چپ با کارآیی انسان مرتبط می‌باشند.

سیستم‌هایی که منطقی فکر می‌کنند (Thinking rationally)

در این بخش تعاریف زیر از هوش مصنوعی قرار می‌گیرند:

- مطالعه قابلیت‌های ذهنی از طریق مدل‌های محاسباتی (چارنیاک و مک درموت، ۱۹۸۵)
- مطالعه محاسباتی که امکان مشاهده، استدلال و عمل را فراهم می‌نمایند. (وینسون ۱۹۹۲)

سیستم‌هایی که منطقی عمل می‌کنند (Acting rationally)

- هوش محاسباتی، مطالعه عامل‌های هوشمند است. (پل اتال ۱۹۹۸)
- هوش مصنوعی با رفتار هوشمند در ساخته‌های انسانی در ارتباط است. (نیلسون ۱۹۹۸)

سیستم‌هایی که مانند انسان فکر می‌کنند (Thinking humanly)

- تلاش جدید و هیجان‌انگیز برای ساخت ماشین‌هایی متفکر و با حس کامل (هاوگلند ۱۹۸۵)
- خودکارسازی فعالیت‌های مرتبط با تفکر انسان، فعالیت‌هایی مانند تصمیم‌گیری، حل مسأله و یادگیری (بلمن ۱۹۷۸)

سیستم‌هایی که مانند انسان عمل می‌کنند. (Acting humanly)

- هنر ساخت ماشین‌هایی که کارهایی را انجام می‌دهند که آن کارها توسط انسان با فکر کردن انجام می‌شوند. (کروزیل ۱۹۹۰)
- مطالعه برای ساخت کامپیوترها برای انجام کارهایی که فعلاً انسان آنها را بهتر انجام می‌دهد. (ریچ و نایت ۱۹۹۱)

در تعاریفی که در بالا ذکر شدند، تعاریف مرتبط با انسان شامل فرضیات و تجربیات می‌باشند و تعاریف مرتبط با منطق،

ترکیبی از ریاضیات و مهندسی هستند. هم‌چنین دو تعریف زیر از هوش مصنوعی را می‌توان در دسته سوم جای داد.

- علم و مهندسی ساخت ماشین‌هایی هوشمند و به‌خصوص برنامه‌های هوشمند (جان مک کارتی از دانشگاه استنفورد)
- هوش مصنوعی عبارتست از ایجاد ظرفیت برای انجام وظایفی که عموماً به‌عنوان ویژگی‌های انسان شناخته می‌شوند، مانند استدلال، اکتشاف، مفهوم، تعمیم، یادگیری و ... (هربرت سیمون). هوش مصنوعی عموماً به‌عنوان زیر شاخه‌ای از کامپیوتر محسوب شده و ارتباط تنگاتنگی با عصب‌شناسی، علوم شناختی، روان‌شناسی شناختی، منطق ریاضی و مهندسی دارد.

مبانی هوش مصنوعی

مبانی هوش مصنوعی به‌طور مختصر در دسته‌بندی زیر جای می‌گیرد:

- **فلسفه:** منطق، استدلال، ناشی شدن تفکر از مغز فیزیکی، مبانی یادگیری، زبان و عقلانیت.
- **روان‌شناسی:** تطبیق، اثر طبیعی ادراک و تأثیر آن بر محیط.
- **زبان‌شناسی:** علم ارائه، گرامر.
- **ریاضیات:** نمایش رسمی الگوریتم‌ها، محاسبات، تصمیم‌پذیری و تصمیم‌ناپذیری، احتمال.
- **نظریه کنترل و سایبرنتیک:** تحت کنترل درآوردن مصنوعات، ثبات و پایداری، طراحی عامل بهینه.
- **علوم عصبی:** نحوه پردازش اطلاعات توسط مغز.
- **اقتصاد:** نظریه تصمیم‌های عقلانی، نظریه بازی.
- **مهندسی کامپیوتر:** ساخت کامپیوترهای سریع.

تاریخچه هوش مصنوعی

تاریخچه تحولات و کارهای انجام گرفته در هوش مصنوعی از سال ۱۹۴۳ و با ارائه مدل نرون مصنوعی بیتی قابل یادگیری برای محاسبه هر تابع قابل محاسبه توسط مک کلوچ و والتر پیترز آغاز می‌شود و این سیر تحولات به‌طور مختصر تا به حال به روال زیر می‌باشد:

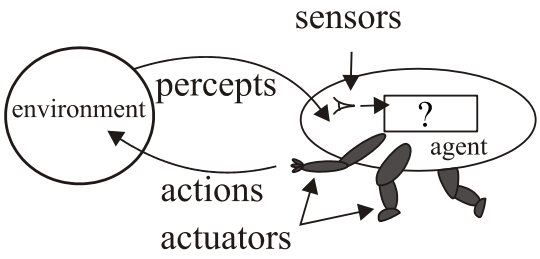
- ۱۹۴۳، مک کلوچ و والتر پیترز: ارائه مدل نرون مصنوعی بیتی (دو حالته) قابل یادگیری به منظور محاسبه هر تابع قابل محاسبه.
- ۱۹۵۰، آلن‌تورینگ اولین بار دیدکاملی از هوش مصنوعی را تحت عنوان «محاسبات ماشینی و هوشمند» ارائه نمود.
- ۱۹۵۱، هینسکی و ادموندز اولین کامپیوتر شبکه عصبی را طراحی کردند.
- ۱۹۵۲، آرتورساموبل برنامه‌ای ساخت که یاد می‌گرفت بهتر از نویسنده‌اش بازی کند، در نتیجه این تصور را که «کامپیوتر فقط کاری را انجام می‌دهد که به آن گفته شود» نقض کرد.
- ۱۹۵۶، نشست کار گروهی دورتموند: انتخاب نام هوش مصنوعی.
- ۱۹۵۹، هربرت جلونتر برنامه (GTP) را ساخت که قضایا را با اصول موضوعات مشخص ثابت می‌کرد.
- ۱۹۵۸، جان مک کارتی: تعریف زبان لیسپ که بهترین زبان هوش مصنوعی شد.
- ۱۹۵۸-۱۹۷۳، جیمز اسلاگل: برنامه حل مسایل انتگرال‌گیری فرم بسته.
- تام ایوانز: برنامه حل مشابهت‌های هندسی.
- دانیل بابروز: برنامه حل مسایل جبری.
- دیوید هافمن: پروژه محدوده بینایی روبات در جهان بلوک‌ها.
- دیوید والتز: سیستم بینایی و انتشار محدود.
- پاتریک وینستون: نظریه یادگیری.
- ۱۹۷۳-۱۹۶۶: کند شدن مسیر تحقیقات هوش مصنوعی.
- پیچیده شدن الگوریتم برنامه‌های جدید.

- برنامه ترجمه متون.
- انجام‌ناپذیری بسیاری از مسائلی که سعی در حل آنها بود.
- عدم موفقیت اثبات قضایا با مفروضات بیشتر.
- بکارگیری بعضی محدودیت‌ها روی ساختارهای اساسی.
- محدودیت نمایش پرسپترون دو ورودی.
- (۱۹۶۹-۱۹۷۹) سیستم‌های مبتنی بر دانش.
- جست و جوی همه منظوره که سعی بر یادگیری داشت تا پیمودن راه‌حل کامل.
- مثل برنامه DENDRAL، بوچانان و همکارانش در سال ۱۹۶۹.
- مزیت برنامه DENDRAL این بود که اولین سیستم پاداش غنی بود.
- متدولوژی جدید سیستم خبره.
- مثل سیستم MYCIN که برای تشخیص عفونت‌های خونی طراحی شد.
- استفاده از فاکتورهای قطعیت.
- افزایش تقاضا برای شِمای نمایش دانش.
- استفاده از منطق در پرولوگ، استفاده از ایده مینسکی یعنی قاب‌ها و ...
- ۱۹۸۰ تاکنون: تبدیل هوش مصنوعی به یک صنعت.
- ۱۹۸۶ تاکنون: برگشت به شبکه‌های عصبی.
- ۱۹۸۷ تاکنون: هوش مصنوعی به علم تبدیل می‌شود.
- ۱۹۹۵ تاکنون: ظهور عامل‌های هوشمند.

عامل‌های هوشمند (Agents)

به هر چیزی که محیط خود را از طریق حسگرها[1] ادراک می‌کند و از طریق محرک‌ها[2] روی محیط عملی[3] را انجام را عامل گوییم.

به‌عنوان مثال در یک عامل انسانی، گوش، چشم و دیگر اعضای حسی، حسگر می‌باشند و دست‌ها، پاها، دهان و دیگر اعضای بدن، محرک‌ها می‌باشند. همچنین در یک عامل روبوتیک، حسگرها می‌توانند دوربین‌ها و ردیاب‌های مادون قرمز باشند و محرک‌ها می‌توانند موتورهای موجود درون آن باشند. در یک عامل نرم‌افزاری رشته‌های بیتی به‌عنوان مشاهدات[4] و اعمالش[5] مورد استفاده قرار می‌گیرد. ارتباط بین یک عامل و محیط آن در شکل زیر آورده شده است:



1- sensors
2- actuators
3- action
4- percepts
5- actions

ویژگی‌های عامل‌ها

۱- ادراک[1]: یک عامل باید توانایی درک محیط اطراف خود را داشته باشد. این درک عمدتاً شامل درک تصویر و درک صدا می‌باشد.

یکی از مسائل مهم دیگر عامل، شناخت از وجود خود[2] است. این بدان معناست که عامل از وجود خود آگاه بوده و بتواند روی محیط خود محرک باشد. تاکنون عامل‌های روبوتیکی که قادر به درک خود باشند به وجود نیامده‌اند. دانشمندان آلمانی رباتی به‌نام Gaak ساختند که قادر به تفکر به‌صورت مستقل بود. این ربات در مسابقات «زنده ماندن قوی‌ترها»[3] شرکت نمود و سعی کرد از یکی از مبارزاتش فرار کند. بعد از فرار از محل مبارزه، در تصادفی از بین رفت. این حادثه می‌تواند دلیلی بر هوشمند بودن این عامل باشد ولی دلیل قاطعی برای اثبات این موضوع نمی‌باشد.

۲- یادگیری[4]: عامل باید قابلیت یادگیری الگوهای جدید پیرامون خود را داشته باشد و بتواند مسائل جدید را بیاموزد و در جایی مشابه مغز انسان این اطلاعات را نگهداری نماید.

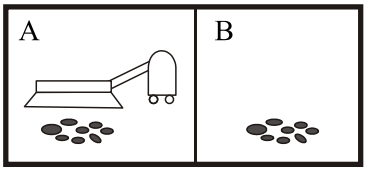
۳- تطابق و پذیرش: عامل باید بتواند خود را با محیط تطابق دهد و در صورت تغییر محیط باید توانایی پذیرش الگوهای جدید را داشته باشد.

۴- قابلیت استدلال: پایگاه دانش عامل برای استدلال باید کامل باشد.

۵- پاسخ‌دهی مناسب به رویدادها: منظور از پاسخ‌دهی مناسب به رویدادها، تصمیم‌گیری‌های صحیح نسبت به رویدادهای محیط، واکنش نشان دادن نسبت به رخدادهای محیطی و غیره می‌باشد.

دنیای ماشین جاروبرقی

به‌عنوان مثال از یک عامل می‌توان دنیای یک ماشین جاروبرقی که در شکل زیر آمده است را در نظر گرفت:



در این مثال، ادراکات، عمل‌ها و ورودی‌ها به‌صورت زیر می‌باشند:

ادراکات: موقعیت و محتوا، به‌عنوان مثال مقدار کثیفی.

عمل‌ها: حرکت به سمت چپ و راست، قدرت مکش و ... .

ورودی‌ها: جدول‌هایی مانند جدول جاروبرقی، جدول عامل، جدول تابع عامل و... .

عامل عقلانی: عاملی است که کار را درست انجام می‌دهد. کار درست از کار نادرست بهتر است. اقدام درست عملی است که سبب موفقیت بیشتر عامل شود.

مقیاس کارایی: معیار موفقیت رفتار یک عامل می‌باشد.

به‌عنوان مثال، مقیاس کارآیی یک جاروبرقی می‌تواند میزان تمیز کردن، مدت زمان تمیز کردن، مقدار برق مصرفی و مقدار نویز ایجاد شده باشد.

نکته: منطقی بودن[5] و دانش کامل[6] با یکدیگر متفاوتند و در واقع داشتن دانش کامل، غیرممکن می‌باشد. مثالی برای واضح شدن این نکته در ادامه آمده است:

1- perception
2- sentient
3- survival of the fittest
4- learning
5- Rationality
6- Omniscience

مثال: من در خیابان شانزلیزه در حال قدم زدن هستم که یک دوست قدیمی را در تقاطع خیابان می‌بینم. از آنجایی‌که در اطرافم ماشینی مشاهده نمی‌کنم، منطقی است که از خیابان عبور کنم. در این زمان شیئی از هواپیما جدا شده و قبل از عبور از خیابان، روی سر من می‌افتد! در این حالت آیا عبور من از خیابان منطقی بوده است؟ واضح است که از آنجایی که همواره عبور من از خیابان موفقیت‌آمیز بوده است، بنابراین این کار منطقی بوده است ولی از آنجایی که راهی برای مشاهده قطعه جدا شده از هواپیما را ندارم، بنابراین دانش من کامل نبوده است، در نتیجه منطقی بودن عمل نیازی به دانش کامل ندارد، زیرا بر اساس تاریخچه ادراکی انجام می‌شود.

نکته: بخش مهمی از منطق بودن، جمع‌آوری اطلاعات می‌باشد.

نکته: منطقی بودن، عامل به ماکزیمم رساندن کارآیی مورد انتظار[1] و کامل بودن به معنای ماکزیمم کردن کارآیی واقعی[2] می‌باشد.

نکته: منطقی بودن یک عامل باید در حیطه دانشی‌که عامل نسبت به محیط خود دارد و اعمالی که می‌تواند داشته باشد، سنجیده شود.

به طور خلاصه، منطقی بودن یک عامل، به چهار مورد زیر وابسته است:

۱- معیار کارآیی که درجه موفقیت را تعیین می‌کند.

۲- مشاهداتی که عامل تا به حال به‌وسیله حسگرهایش دریافت کرده است. این تاریخچه ادراکی کامل را رشته ادراکی[3] می‌نامند.

۳- آنچه عامل در رابطه با محیط می‌داند.

۴- اعمالی که عامل می‌تواند انجام دهد.

عامل منطقی ایده‌آل: عاملی است که به‌ازای هر رشته ادراکی ممکن با استفاده از مشاهدات، باید براساس رشته ادراکی و پیش زمینه دانشی، اعمالی را انجام دهد که انتظار می‌رود معیار کارایی‌اش را ماکزیمم کند.

در تعریف فوق، مفهومی به‌نام دانش درونی[4] یا دانش پیش‌زمینه‌ای[5] وجود دارد. اگر اعمال عامل تنها براساس دانش درونی (که همان دانش طراح است) باشد، آنگاه عامل به مشاهدات خود توجهی ندارد و در این صورت می‌گوییم عامل فاقد استقلال یا خودمختاری است.

بهترین حالت این است که یک عامل هوشمند را با مقداری دانش اولیه و همراه با قدرت یادگیری طراحی کنیم تا بتواند در انواع وسیعی از محیط‌ها موفق باشد و بعد از آنکه عامل تجربه کافی از محیط خود به‌دست آورد، می‌تواند تا حد زیادی مستقل از دانش قبلی خود عمل کند، به‌عبارت دیگر یک عامل منطقی باید خودمختار باشد.

PEAS: مقیاس کارآیی، محیط، محرک‌ها و حسگرها: PEAS از سر واژه واژگان مقیاس کارآیی، محیط، محرک‌ها و حسگرها تشکیل شده است در اولین گام برای طراحی یک عامل باید محیط کار در حد امکان مشخص باشد. به‌عنوان مثال یک راننده خودکار تاکسی را درنظر بگیرید برای این راننده PEAS به‌صورت زیر می‌باشد:

مقیاس کارآیی: رسیدن به مقصد صحیح، کمینه کردن مصرف سوخت و استهلاک، کمینه کردن زمان یا هزینه سفر، کمینه کردن نقض قوانین راهنمایی و رانندگی، بیشینه کردن امنیت و سود و راحتی مسافران

محیط: جاده، ترافیک، عابران پیاده و مسافران

محرک‌ها: فرمان اتومبیل، شتاب‌دهنده، ترمز، چراغ‌ها و بوق

حسگرها: دوربین‌ها، sonar، سرعت‌سنج‌ها، GPS، کیلومترشمار و...

1- expected performance
2- actual performance
3- percept sequence
4- built – in knowledge
5- background knowledge

چند مثال از انواع عامل‌ها و توصیف PEAS برای هریک از آنها

مثال اول:

- عامل: سیستم تشخیص پزشکی
- مقیاس کارآیی: بهبود بیمار، کمینه کردن هزینه‌ها و مسائل حقوقی
- محیط: بیمار، بیمارستان و کارکنان
- محرک‌ها: پرسش‌ها، آزمایش‌ها، تشخیص‌ها، تجویزها و ارجاع‌ها
- حسگرها: وارد کردن علایم بیماری، یافته‌ها و پاسخ‌های بیمار از طریق صفحه کلید

مثال دوم:

- عامل: رباتی که قطعات را جابه‌جا می‌کند.
- مقیاس کارآیی: درصد قطعاتی که در سطل درست قرار گرفته‌اند.
- محیط: کمربند نقاله همراه با قطعات، سطل‌ها.
- محرک‌ها: بازو و دست مفصل‌دار.
- حسگرها: دوربین، حسگرهای زاویه مفصل.

مثال سوم:

- عامل: معلم انگلیسی محاوره‌ای
- مقیاس کارآیی: پیشینه کردن نمره امتحان دانش‌آموز
- محیط: مجموعه دانش‌آموزان، موسسه امتحان گیرنده
- محرک‌ها: تمرین‌ها، پیشنهادات و اصلاحات
- حسگرها: ورودی صفحه کلید

ساختار عامل‌های هوشمند

وظیفه اصلی طراح هوش مصنوعی، طراحی برنامه عامل[1] می‌باشد. این کار به معنای طراحی تابع عاملی[2] است که رشته مشاهدات را به یک عمل نگاشت کند. تابع عامل یک توصیف ریاضی خلاصه است و شامل جزئیات پیاده‌سازی نمی‌باشد. درحالی‌که برنامه عامل، پیاده‌سازی این تابع روی قسمتی از دستگاه‌های محاسباتی که معماری[3] نامیده می‌شود می‌باشد. معماری، ابتدا مشاهدات و ادراکات را از حسگرها می‌گیرد و برای برنامه‌های عامل، قابل دسترس می‌سازد. سپس برنامه عامل را اجرا نموده و اعمال انتخاب شده را به تأثیرگذارها می‌دهد. معماری می‌تواند یک کامپیوتر یا سخت‌افزار خاص منظوره مانند پردازشگر تصاویر دوربین یا فیلترکننده ورودی صدا باشد. ارتباط بین عامل، معماری و برنامه به‌صورت زیر می‌باشد:

معماری + برنامه عامل = عامل

برنامه‌های عامل[4]

عامل‌ها دارای چارچوب یکسانی هستند که شامل دریافت مشاهدات از محیط و تولید عمل مناسب می‌باشد. فرم ساده اولین نوع برنامه عامل در زیرآورده شده است:

```
fhunction SKELETON-AGENT (percept) return action
station: memory; the agent's memory of the world
memory ← UP DATE – MEMORY (memory, percept)
action ← CHOOSE – BEST – ACTION (memory)
memory ← UP DATE – MEMORY (memory, action)
return action
```

1- agent program
2- Agent function
3- architecture
4- agent program

این برنامه از ساختارهای داده داخلی استفاده می‌کند که با دریافت مشاهدات جدید، به‌روزرسانی می‌شود. رویه‌های تصمیم‌گیری، از این ساختار داده برای تولید و انتخاب یک عمل استفاده نموده و در نهایت عمل انتخاب شده توسط بخش معماری اجرا می‌شود.

ساده‌ترین راه نوشتن برنامه عامل، استفاده از جدول است. در زیر یک برنامه عامل با استفاده از جدول آورده شده است:

function TABLE – DRIVEN – AGENT (percept) returns action
static: percepts, a sequence initially empty table, a table indexed by percept sequences, initially fully specified append percept to the end of percepts
action ← LOOK UP (percept, table) returen action

در این عامل، رشته مشاهدات به‌طورکامل در حافظه نگهداری می‌شود و از مشاهدات برای اشاره به مدخل‌های جدول که شامل عمل مناسب برای همه مشاهدات ممکن است، استفاده می‌شود.

معایب این روش عبارتند از:

۱- تعداد مدخل‌های مورد نیاز: به‌طور مثال در بازی شطرنج که حدود $۳۵^{۱۰۰}$ می‌باشد.

۲- زمانبر بودن ساخت جدول.

۳- عدم استقلال عامل.

۴- زمانبر بودن بروزرسانی مدخل‌های جدول حتی با وجود یادگیری.

ماهیت عامل

اصول زیربنایی هر سیستم هوشمند (برنامه عامل) شامل چهار نوع اصلی زیر می‌باشد:

۱- عامل واکنشی ساده (simple reflex agent)

۲- عامل واکنشی مبتنی بر مدل (model – based reflex agent)

۳- عامل مبتنی بر هدف (goal – based agent)

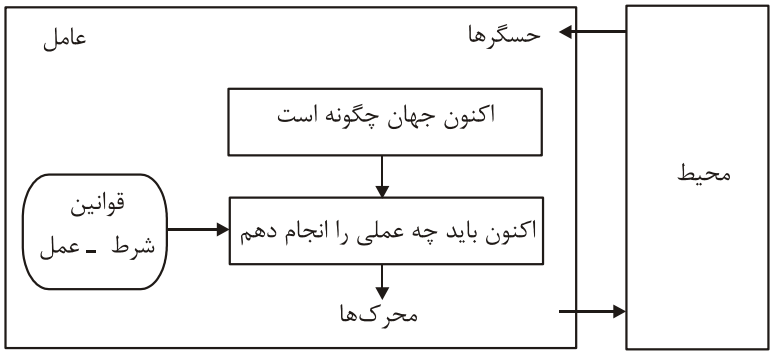
۴- عامل مبتنی بر سودمندی (utility – based agent)

عامل واکنشی ساده

در این نوع عامل، با اعمال برخی از پردازش‌ها روی ورودی، شرطی مشخص می‌شود که منجر به انتخاب عمل توسط عامل می‌گردد. به‌عنوان مثال اگر ماشین جلویی ترمز نماید و چراغ‌های ترمز آن روشن شوند، راننده باید اقدام به ترمز گرفتن کند. در اینجا «روشن شدن چراغ ترمز ماشین جلویی» به‌عنوان ورودی وارد سیستم می‌شود و با اعمال برخی پردازش‌ها شرط «ماشین جلویی در حال ترمز گرفتن» مشخص می‌شود و سپس عمل مناسب که همان «ترمز گرفتن» است. انتخاب می‌شود. چنین رابطه‌ای را یک قانون شرط – عمل (condition – action rule) می‌نامیم و به‌صورت زیر نوشته می‌شود:

اگر ماشین جلویی در حال ترمز است آنگاه شروع به ترمز کن.

شکل زیر شمای یک عامل واکنشی ساده را نشان می‌دهد و همچنین برنامه عامل واکنشی ساده در زیر آن آمده است:



```
function SIMPLE – REFLEX – AGENT (percept) return action
static: rules, a set of condition – action rules
state ← INTERPRET – INPUT (percept)
rule ← RULE – MATCH (state, rules)
action ← RULE – Action (rule)
return action
```

در برنامه فوق، تابع INTERPRET – INPUT براساس ادراک، توصیفی خلاصه از حالت جاری محیط تولید می‌کند و تابع RULE-MATCH اولین قانون در مجموعه قوانین که با حالت توصیفی منطبق است را به‌عنوان خروجی برمی‌گرداند. اگرچه این عامل‌ها می‌توانند خیلی مؤثر پیاده‌سازی شوند ولی کاربرد آنها کم است.

نکته: همواره در حالتی که تصمیم‌گیری درست براساس مشاهده جاری، امکان‌پذیر باشد، عامل واکنشی ساده، درست کار خواهد کرد.

عامل‌های واکنشی مبتنی بر مدل

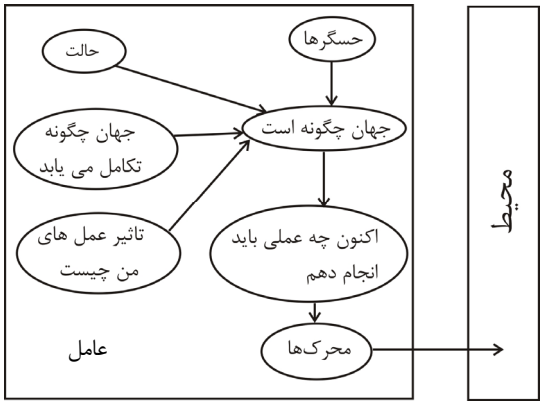
در این نوع عامل، مشاهده جاری (percept) با اطلاعات ساختار داخلی (internal state) ترکیب‌شده و توصیف به‌روزرسانی شده‌ای از حالت جاری تولید می‌کند. در برنامه این عامل، تابع UPDATE STATE مسئول ایجاد توصیف جدید حالت داخلی می‌باشد و به محض تفسیر مشاهده جدید با استفاده از اطلاعاتی که درباره رشد دنیا و نتیجه اعمال عامل موجود است، مسیر قسمت‌های دیده نشده دنیا را پیدا نموده و در ساختار داخلی حفظ می‌کند.

دانش در رابطه با اینکه «دنیا چگونه کار می‌کند» یک مدل از دنیا نامیده می‌شود و عاملی که از این مدل استفاده می‌کند، عامل مبتنی بر مدل نامیده می‌شود. به این عامل، عاملی که مسیر دنیا را حفظ می‌کند (agent that keep track of the world) نیز گفته می‌شود. مثالی از موقعیتی که باید از عامل مبتنی بر مدل استفاده نمود در ادامه آمده است.

مثال: راننده برای بررسی موقعیت وسیله نقلیه مجاور لحظه به لحظه در آینه جلوی ماشین نگاه می‌کند. اگر راننده به آینه نگاه نکند، وسیله نقلیه خط بعدی را نمی‌بیند ولی در تغییر خط، راننده باید بداند که آیا ماشین آنجا قرار دارد یا خیر.

در این مورد نیاز است که عامل اطلاعات قبلی را در ساختارهای داخلی خود نگهداری کند تا قادر باشد که حالت‌های متفاوت محیط را که ورودی یکسانی تولید می‌کنند را تشخیص دهد.

شمای یک عامل واکنشی مبتنی بر مدل در شکل زیر نشان داده شده است.

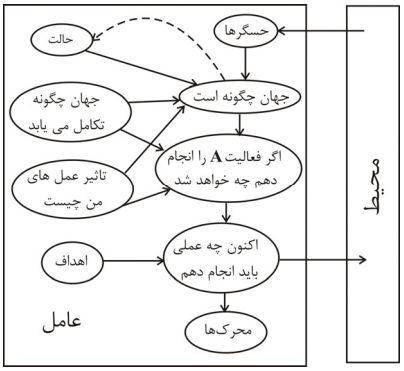


برنامه عامل واکنشی مبتنی بر مدل نیز به‌صورت زیر می‌باشد:

```
function REFLEX – AGENT – WITH – STATE (percept) returns an action
static: state, a description of the current world state rules, a set of condition – action rules
action, the most recent action, initially none
state ← UPDATE – STATE (state, action, percept)
rule ← RULE – MATCH (state, rules)
action ← RULE – ACTION (rule)
return action
```

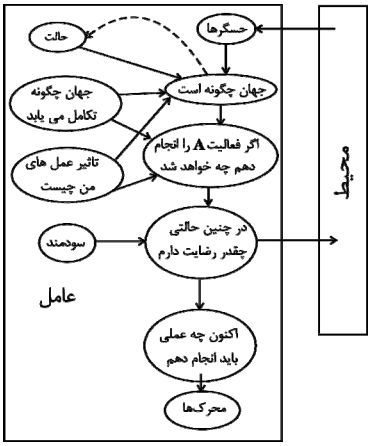
عامل مبتنی بر هدف

دانش موجود درباره حالت جاری محیط نمی‌تواند همواره برای تصمیم‌گیری کافی باشد. به‌عنوان مثال یک راننده همواره در یک چهارراه می‌تواند به چپ، راست بپیچد یا مستقیم حرکت کند و اینکه کدام‌یک از اینها را انتخاب می‌کند، بستگی به هدف و مقصد دارد. بنابراین به‌طورکلی در این نوع عامل، علاوه بر داشتن اطلاعاتی درباره حالت جاری، به اطلاعاتی در رابطه با هدف نیز نیاز است. برنامه عامل می‌تواند این اطلاعات را با اطلاعات مربوط به نتایج اعمال ممکن ترکیب نموده و اعمالی را که او را به هدف می‌رساند انتخاب کند. اگر هدف از یک عمل نتیجه شود، کار ساده است ولی هنگامی که عامل مجبور باشد برای رسیدن به هدف، رشته‌ای طولانی را بررسی کند، کار مشکل می‌شود. در هوش مصنوعی، جستجو[1] و برنامه‌ریزی[2] زمینه‌هایی هستند که رشته‌ای از اعمال را برای رسیدن عامل به هدف، پیدا می‌کنند. توجه داشته باشید که عامل مبتنی بر هدف، به سادگی با توصیف مقاصد جدید می‌تواند خود را در رسیدن به مقاصد مختلف وفق دهد. در حالی‌که در عامل واکنشی ساده، کلیه قوانینی که براساس مقصد قبلی تولید شده‌اند، باید براساس مقصد جدید تغییر نمایند. در شکل زیر ساختار عامل مبتنی بر هدف، آورده شده است:



عامل مبتنی بر سودمندی

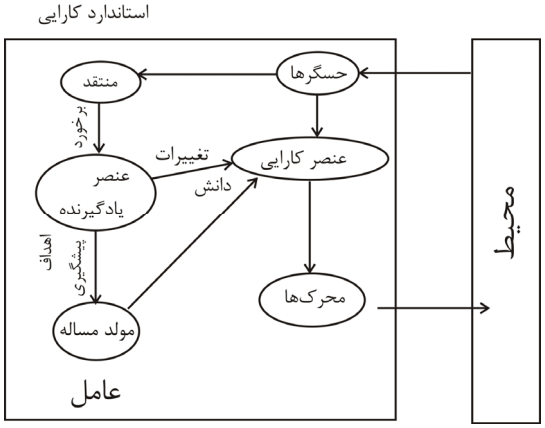
برای درک عامل مبتنی بر سودمندی، ماشینی را درنظر بگیرید که می‌خواهد به مقصدی برسد. این ماشین می‌تواند از راه‌های مختلف به آن مقصد برسد که این راه‌ها از نظر امنیت، سرعت، هزینه با یکدیگر متفاوتند. عامل مبتنی به سودمندی برای ایجاد رفتار با کیفیت بالا از تابع سودمندی استفاده می‌کند. این تابع، هر حالت را به یک عدد حقیقی می‌نگارد و منجر به تصمیم‌های منطقی در دو حالت می‌شود: اول حالتی که اهداف مخالف و متناقض (مانند سرعت و امنیت در مثال رانندگی) هستند و تنها برخی از آنها تأمین می‌شود که در این حالت تابع سودمندی تعادل مناسبی بین آنها ایجاد می‌کند. حالت دوم حالتی است که چندین هدف وجود دارد و عامل نمی‌تواند با اطمینان تعیین کند که باید به کدام یک دست یابد. در این حالت براساس تابع سودمندی، هدفی که سودمندی بیشتری دارد انتخاب می‌شود. ساختار عامل مبتنی بر سودمندی به‌صورت زیر می‌باشد:



1- search
2- planning

عامل‌های یادگیرنده

شکل کلی یک عامل یادگیرنده به‌صورت زیر می‌باشد.



همان‌طور که در شکل دیده می‌شود، این عامل از چهار مولفه تشکیل شده است. عنصر یادگیری مسئول انجام بهبودها و عنصر کارایی مسئول انتخاب اعمال خارجی می‌باشد. عنصر کارایی در واقع همان عامل می‌باشد که مشاهدات را می‌گیرد و اعمال را انتخاب می‌کند. عنصر یادگیری از بازخورد[1] و منتقد[2] استفاده می‌کند. منتقد مشخص می‌سازد که عامل تا چه حد موفق عمل کرده است و عنصر کارایی چگونه تغییر کند تا عامل در آینده موفق‌تر عمل کند.

آخرین مولفه یک عامل یادگیرنده، مولد مسأله[3] است که مسئول پیشنهاد اعمالی است که منجر به تجربه‌های ارزشمند و مفید می‌گردد. برای روشن شدن مطلب به مثال زیر توجه کنید:

مثال: عامل: راننده تاکسی

عنصر کارایی: دانش و رویه‌هایی که تاکسی برای انتخاب اعمال خود استفاده می‌کند.

تاکسی با استفاده از عنصر کارایی خود به جاده می‌رود و رانندگی می‌کند. منتقد پس از مشاهده دنیا، اطلاعات را به عنصر یادگیری می‌فرستد. به‌عنوان مثال بعد از یک گردش ناگهانی به سمت چپ، منتقد شکایات سایر رانندگان را مشاهده کرده و عنصر یادگیری با استفاده از این تجربه برای خود یک قانون وضع می‌کند که این موقعیت اصلاً مناسب نمی‌باشد و عنصر کارآیی نیز با وضع این قانون تغییر می‌کند.

محیط‌ها

در اینجا انواع محیط‌های مختلف را مورد بررسی قرار می‌دهیم و در رابطه با ارتباط عامل با این محیط‌ها توضیح می‌دهیم. به‌طور کلی محیط می‌تواند دارای خواص و ویژگی‌های زیر باشد:

کاملاً مشاهده‌پذیر در مقابل پاره‌ای مشاهده‌پذیر (fully observable vs. partially observable)

اگر حسگر بتواند حالت کامل محیط را در اختیار عامل قرار دهد، آنگاه محیط برای عامل کاملاً مشاهده‌پذیر یا قابل دسترس[4] خواهد بود. چنین محیطی، ساده می‌باشد و عامل نیازی به نگهداری حالات داخلی برای حفظ اثرات دنیا را ندارد.

1- feedback
2- critic
3- problem generator
4- accessible

مثال: از آنجایی که راننده تاکسی نمی‌داند که چه چیزی از ذهن سایر رانندگان خطور می‌کند، محیط آن محیطی پاره‌ای مشاهده‌پذیر یا غیرقابل دسترس (inaccessible) خواهد بود.

قطعی در مقابل تصادفی (deterministic vs. stochastic)

اگر بتوان حالت بعدی محیط را با استفاده از حالت جاری و عمل انتخاب شده به‌طورکامل مشخص نمود، محیط قطعی می‌باشد. محیط‌های غیرقابل دسترس، اکثراً تصادفی می‌باشند، زیرا نگهداری مسیر همه موارد غیرقابل دسترس در یک محیط پیچیده، بسیار دشوار می‌باشد. قطعی یا غیرقطعی بودن یک محیط، به دیدگاه عامل بستگی دارد.

نکته: اگر محیطی برای یک عامل، قطعی باشد ولی برای سایر عامل‌ها قطعی نباشد به آن محیط استراتژیک گفته می‌شود.

تقسیم‌پذیر در مقابل پی‌درپی (episodic vs. sequential)

در محیط‌های تقسیم‌پذیر، وظیفه عامل به بخش‌هایی تقسیم می‌شود. هر بخش شامل دریافت مشاهدات انجام عمل است و کیفیت هر عمل، تنها به بخش مربوطه بستگی دارد و بخش‌های بعدی به اعمالی که در بخش‌های قبلی اتفاق افتاده است، بستگی ندارند.

نکته: محیط‌های تقسیم‌پذیر بسیار ساده می‌باشند. زیرا در آنها نیازی به آینده‌نگری نمی‌باشد. ولی در محیط‌های پی‌درپی یا تقسیم‌ناپذیر[1] تصمیم جاری بر کلیه تصمیمات آینده تأثیر می‌گذارد.

ایستا در مقابل پویا (static vs. dynamic)

اگر محیط در زمان تصمیم‌گیری عامل برای انتخاب عمل مناسب، تغییر کند به آن محیط پویا گفته می‌شود و در غیر این صورت محیط ایستا می‌باشد. در محیط‌های پویا، باید عامل اتفاقات محیط را در حین تصمیم‌گیری، نگهداری کند و گذر زمان را مدنظر قرار دهد. اگر با گذشت زمان محیط تغییر نکند ولی کارآیی تغییر کند، محیط را نیمه پویا[2] گویند.

گسسته در مقابل پیوسته (discrete vs. continuous)

اگر بتوان مشاهدات و اعمال محیط را با اعداد گسسته متناظر کرد، محیط را گسسته گویند. به‌عنوان مثال از آنجایی که در بازی شطرنج در هر نوبت تنها تعداد متناهی حرکت وجود دارد، محیط شطرنج گسسته می‌باشد و از آنجایی‌که عامل‌هایی مانند سرعت، موقعیت و ... در حرکت یک ماشین دارای مقادیر پیوسته‌ای می‌باشند، محیط یک راننده، محیطی پیوسته می‌باشد.

تک عامله در مقابل چند عامله (single agent vs. multi agent)

تشخیص تک عامله یا چند عامله بودن یک محیط ساده است. برای مثال محیط یک شطرنج از محیطی دو عامله است و محیط عاملی که جدول کلمات متقاطع را حل می‌کند، تک عامله است. در بازی شطرنج از آنجایی که هر یک از بازیکن‌ها سعی در ماکزیمم نمودن معیار کارایی خود و در عین حال مینیمم نمودن معیار کارایی حریف دارد، محیط آن یک محیط چند عاملی رقابتی[3] می‌نامند. در محیط راننده، جلوگیری از تصادف، معیار کارآیی همه رانندگان را ماکزیمم می‌کند. بنابراین این محیط چندعامله پاره‌ای همکار[4] می‌باشد.

1- nonepisodic
2- semidynamic
3- competitive multiagent
4- partially cooperative multiagent

محیط رانندگان در عین‌حال، یک محیط پاره‌ای رقابتی[1] نیز می‌باشد، زیرا برای پیدا کردن مکانی برای پاک کردن، تمام رانندگان با یکدیگر به رقابت می‌پردازند.

نکته: سخت‌ترین محیط، محیط غیرقابل دسترس، تصادفی، تقسیم‌ناپذیر، پویا، و پیوسته و چند عامله است.

در جدول زیر تعدادی از مثال‌ها به همراه ویژگی‌های محیط آنها آورده شده است.

محیط	قابل دسترس	قطعی	تقسیم‌پذیر	ایستا	گسسته	تک‌عامله
شطرنج با درنظر گرفتن زمان	بلی	استراتژیک	خیر	نیمه پویا	بلی	چند
شطرنج بدون در نظر گرفتن زمان	بلی	استراتژیک	خیر	بلی	بلی	چند
پوکر (poker)	خیر	استراتژیک	خیر	بلی	بلی	چند
تخته نرد (backgammon)	بلی	خیر	خیر	بلی	بلی	چند
راننده تاکسی	خیر	خیر	خیر	خیر	خیر	تک
سیستم تشخیص پزشکی (medical diagnosis)	خیر	خیر	خیر	خیر	خیر	چند
سیستم آنالیز تصاویر	بلی	بلی	بلی	نیمه پویا	خیر	تک
ربات جابه‌جاکننده قطعات (part-picking-robot)	خیر	خیر	بلی	خیر	خیر	تک
کنترل‌کننده پالایشگاه (refinery controller)	خیر	خیر	خیر	خیر	خیر	تک
جدول کلمات متقاطع (crossword puzzle)	بلی	بلی	خیر	بلی	بلی	تک
معلم خصوصی درس انگلیسی (interactive English tutor)	خیر	خیر	خیر	خیر	بلی	چند

برنامه عمومی محیط

این برنامه ارتباط اولیه بین عامل‌ها و محیط را بیان می‌کند. این برنامه، با یک یا چند عامل در ارتباط است و مرتباً به هر عامل، مشاهده صحیح را داده و از هر عامل، عمل مناسب را دریافت می‌کند. سپس این شبیه‌ساز، محیط را براساس اعمال و فرآیندهای پویای دیگری که به‌عنوان عامل درنظر گرفته نمی‌شوند (مانند باران) به روزرسانی می‌کند. این برنامه برای هر عامل یک معیار کارآیی را به‌کار می‌برد و لیستی از امتیازات عامل را برمی‌گرداند. برنامه شبیه‌ساز محیط به‌صورت زیر می‌باشد:

```
procedure RUN-ENVIRONMENT (state, UPDATE-FN, agents, termination)
input: state, the initial state of the environment
UPDATE-FN, function to modify the environment
agent, a set of agents,
termination, a predicate to test when we are done
repeat
for each agents in agents do
PERCEPT [agent] ← GET – PERCEPT (agent, state)
End
for each agent in agents do
ACTION [agent] ← PRO GRAM [agent] (PERCEPT [agent])
End
state ← UPDATE – FN (action, agents, state)
until termination (state)
```

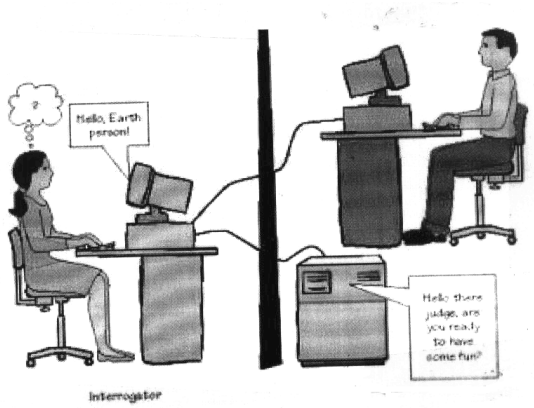
1- partially competitive

تابع RUN-EVAL-ENVIRONMENT معیار کارآیی را برای یک محیط برمی‌گرداند. معمولاً عامل یک مسأله در یک کلاس از محیط‌ها طراحی می‌شود. برای مثال، برنامه شطرنج برای بازی با افراد مختلف و یا حریف‌های ماشینی طراحی می‌شود.

```
function RUN-EVAL-ENVIRONMENT (state, UPDATE-FN, agents,
termination, PERFORMANCE-FN) return scores
local variables: scores, a vector the same size as agents, all 0
repeat
for each agent in agents do
PERCEPT [agent] ← GET-PERCEPT (agent, state)
End
for each agent in agents do
ACTION [agent] ← PROGRAM [agent] (PERCEPT [agent])
End
state ← UPDATE-FN (actions, agents, state)
scores ← PERFORMANCE-FN (scores, agents, state)
until termination (state)
return scores /* change*/
```

تست تورینگ

از این تست در هوش مصنوعی برای بررسی هوشمندی عامل‌هایی که مانند انسان رفتار می‌کنند، استفاده می‌شود و به‌طور ساده به این معنا می‌باشد که سیستم هوش مصنوعی در وضعیت ایده‌آل قرار دارد. اگر شخصی که در جلوی یک ترمینال قرار دارد نتواند تشخیص دهد که شخصی که در آن طرف است، یک کامپیوتر است یا یک انسان و به‌صورت نمادین زیر می‌باشد:



یک سیستم برای موفقیت در تست تورینگ باید دارای قابلیت‌های زیر باشد:

۱- پردازش زبان طبیعی یا NLP (Natural Language Processing) : این ویژگی به معنای درک معنایی جملات نوشته شده برای محاوره می‌باشد در این جا فهم ضرب‌المثل‌ها اندکی دشوار است.

۲- بازنمایی دانش (knowledge represention): جهت ذخیره‌سازی اطلاعات تولید شده قبلی یا حین آزمون می‌باشد.

۳- استدلال خودکار (Automated Reasoning): از اطلاعات ذخیره شده برای پاسخ به پرسش‌ها استفاده کرده و نتایج جدید را استخراج نماید.

۴- یادگیری ماشین (Machine Learning): خود را با شرایط جدید وفق داده و الگوها را کشف کند.

مزیت کامپیوترهای هوشمند نسبت به انسان

این مزیت عبارتند از:

۱- خسته نمی‌شوند.

۲- گرانی نیروی کار انسانی.

۳- عدم شک و اشتباه.

۴- انسان‌ها سرحال نیستند و ...

۵- عدم فراموشی

۶- سرعت بالا، دقت بالا و ...

مباحث مرتبط با هوش مصنوعی

۱- رباتیک: ربات‌ها به دو دسته ثابت و متحرک تقسیم می‌شوند و تقسیمات سینماتیک آنها به‌صورت زیر می‌باشد:

- سینماتیک ربات‌ها
  - هندسه
    - سینماتیک مستقیم
    - سینماتیک معکوس
  - حرکت دیفرانسیلی
    - سینماتیک مستقیم
    - سینماتیک معکوس

۲- ادراک: اکثر مباحث ادراکی که در هوش مصنوعی مطرح است، مربوط به بینایی[1] و سخنرانی[2] می‌باشد. مراحل مختلف بینایی در یک ربات به‌صورت زیر می‌باشد:

و هم‌چنین درک صدا، به‌صورت زیر تقسیم‌بندی می‌شود:

- speech
  - شناسایی صدا
    - تعیین هویت
    - تصدیق هویت
  - شناسایی کلمات ادا شده و تبدیل آنها به متن
  - NLP پردازش زبان طبیعی

۳- حل مسائل و بازی‌ها: مسائلی که در این بخش مطرح می‌شوند، مسائل هوشمند بوده و مسائلی هستند که از قبل برای آنها الگوریتم خاصی نداریم و این به‌دلیل پیچیدگی این گونه مسائل می‌باشد. این مسائل عبارتند از: معمای ۸، شطرنج، کشیش‌ها و

1- vision
2- speech

آدم‌خوارها و... . در این‌گونه مسائل نیاز به دانش[1] داریم. دانش یعنی داشتن دو چیز: حقایق[2] و قوانین حاکم بر حقایق[3]. در ضمن همواره در کنار دانش، تجربه نیز وجود دارد. حال با دانش و تجربه‌ای که از یک مسأله داریم (داشتن factها و Ruleها) به هدف مسأله رسیده و آن را حل می‌کنیم. مسأله اصلی هوش مصنوعی انتخاب بهترین مسیر از بین مسیرهای موجود است که در نهایت برای مسائل از سیستم تولید استفاده می‌کنیم که شامل ۳ مرحله اساسی است:

الف – پایگاه داده (factها)

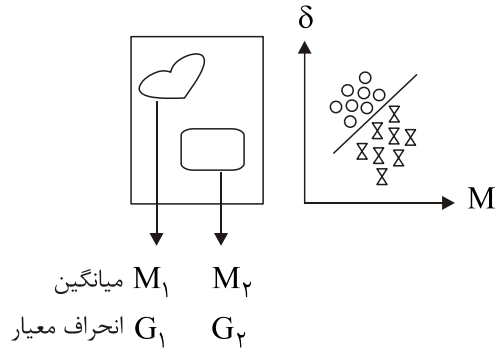
ب – قوانین تولید (Ruleها)

پ – استراتژی کنترل (انتخاب بهترین مسیر)

بحث در ارتباط با نحوه ارائه پایگاه دانش نیاز به زبان منطق، زبان‌های رویه‌ای، شبکه معنایی، سیستم‌های تولید و قاب‌ها دارد.

۴- شناسایی الگو[4]: شاخه‌ای از علوم کامپیوتر است که به دو دسته‌بندی موثر الگوها در داخل دسته‌ها یا کلاس‌های معین می‌پردازد و از دو بعد آماری و ساختاری به این مسأله نگاه می‌کند. در ادامه مثالی از کاربرد این مسأله در هوش مصنوعی آمده است.

مثال: فرض کنید رباتی تصویری را دریافت کند و در این محیط دو شیء مربع و دایره وجود دارد.



به $\begin{bmatrix} m \\ \delta \end{bmatrix}$ بردار ویژگی گفته می‌شود و در حالت کلی، بردار ویژگی می‌تواند برداری nتایی باشد و در این حالت، فضای ویژگی را n بعدی گوییم. در شناسایی الگو، هدف یافتن روشی است که بتوانیم الگوهای جدید را دسته‌بندی کنیم. حال اگر در تصویر ارسالی جدید شی‌ای با ویژگی $\begin{bmatrix} m_x \\ \delta_x \end{bmatrix}$ داشته باشیم باید بررسی کرد که این الگوی جدید مربع است یا دایره. اگر هم‌چنین تعداد الگوهای Δ زیاد باشد، باید بررسی کرد که آیا این الگوها می‌توانند دسته جدیدی تشکیل دهند یا خیر. در اینجا معیار نزدیکی دو الگو را می‌توان یکی از مقیاس‌های زیر درنظر گرفت:

معیار نزدیکی

فاصله همینگ	$H = \sum \lvert x_i - g_i \rvert$
فاصله اقلیدسی	$d(x,y) - \sqrt{۴(n_i - gJ^۲)}$
فاصله شهری	$dc_i = \sum_n \lvert xi - gi \rvert$
فاصله منهتن	$D = MAX \lvert xi - gi \rvert$
نزدیکترین همسایه	

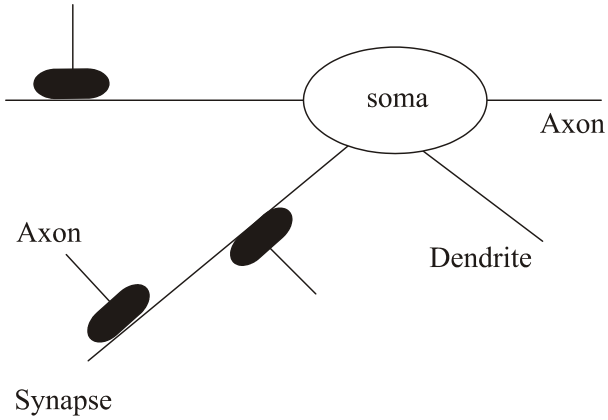
1- knowledge
2- facts
3- Rules
4- pattern Recognition

۵- شبکه‌های عصبی[1]: با وجود آنکه مغز یک ساختار پردازش توزیع شده است ولی این ساختار به مرور زمان وظایف مختلف را تقسیم نموده و هر بخش آن وظیفه‌ای را برعهده دارد. البته این تقسیم کار با این موضوع که این سیستم یک ساختار پردازش موازی بزرگ است در تضاد نیست.

ساختار مغزی انسان در حدود 10^{10} واحد پایه به نام نرون دارد که هر نرون با حدوداً 10^{4} نرون دیگر در ارتباط است. ساختار کامپیوتر به‌این گونه است که باید اعمال را به ترتیب و پشت سرهم با سرعت بالا انجام دهد. درحالی‌که مغز انسان دارای اجزای بیشتر است و با سرعت کمتری (حدود ۱۰۰ هرتز) به طور همزمان کار می‌کند. بنابراین مغز انسان برخلاف کامپیوتر ساختاری موازی دارد.

مهمترین ویژگی مغز انسان، آموزش‌پذیر بودن آن می‌باشد. حال سوال این است که چگونه باید رفتار مغز را تقلید کنیم. ابتدا ساختار مغز را بررسی می‌کنیم.

مغز از 10^{10} نرون تشکیل شده است. نحوه عملیات نرون بسیار پیچیده است و هنوز در سطح میکروسکوپی شناخته شده نمی‌باشد. هر چند که قوانین پایه آن مشخص است. هر نرون دارای تعدادی ورودی است که با یکدیگر به طریقی جمع می‌شوند. اگر تعداد ورودی‌های فعال نرون در یک لحظه به حد کفایت برسد، نرون نیز فعال می‌شود، در غیر این‌صورت نرون غیرفعال می‌شود.



Axon خروجی نرون است که از نظر الکتریکی همواره فعال است. در این حالت Axon در ارتباط‌های بین نرون غایب می‌باشد. رشته اکسون به‌وسیله فاصله سیناپسی از دندریت جدا شده است. این اتصال از طریق ماده شیمیایی موقت صورت می‌گیرد. اگر Axon فعال شود، باعث افزایش پتانسیل سیناپس شده و مواد شیمیایی داخل سیناپس ماده محرکی به‌نام Neurotransmitter ترشح می‌کنند. برای این اثر ترشح ممکن است بیش از یک سیگنال Axon لازم باشد. این ماده ترشح شده در شکاف بین دندریت و Axon پخش شده و باعث افزایش پتانسیل دندریت می‌گردد.

این افزایش باعث ایجاد یک پالس به دندریت شده و سپس وارد بدنه نرون می‌گردد. (در ضمن در هنگام ترشح دروازه‌های باز شده بستگی به میزان ترشح دارد.)

احتمالاً آنچه که باعث یادگیری می‌شود تغییرات synapse هاست. که هم اطلاعات گذشته حفظ می‌شود و هم اطلاعات جدید. حال باید ساختار مغز را شبیه‌سازی کنیم که این کار دارای مراحل تشخیص نوع نرون، ساختار نرون و نحوه آموزش می‌باشد.

۶- سیستم‌های خبره

۷- سیستم‌ها و کنترل فازی

1- Neural Networks

سوالات چهارگزینه‌ای سراسری فصل اول

۱- کدام‌یک از گفته‌های زیر صحیح است؟ **(مکاترونیک ۸۴)**

۱) عامل انعکاسی در محیط‌های پویا قابل استفاده نیست.
۲) عامل خود مختار در محیط‌های پویا قابل استفاده است.
۳) فقط برای محیط‌های پویا عامل به نگهداری حالت احتیاج دارد.
۴) یک عامل خود مختار لزوماً هدف مبنا (Goal based) است.

۲- کدام‌یک از عبارات زیر صحیح است؟ **(مهندسی کامپیوتر ۸۴)**

۱) عاملی که تنها بخشی از محیط را درک می‌کند، نمی‌تواند عامل عقلایی یا معقول باشد.
۲) هر عاملی که از رویه‌های استنتاج Sound استفاده کند، می‌تواند در تست تورینگ موفق شود.
۳) یک عامل معقول Rational همیشه بهتر از عامل‌های غیرمعقول عمل می‌کند، چون نتیجه واقعی اعمالش را می‌داند.
۴) عاملی که به زبان طبیعی ارتباط برقرار می‌کند معمولا در یک محیط «پاره‌ای قابل مشاهده» یا Partially Observable عمل می‌کند.

۳- کدام‌یک از عبارات زیر صحیح است؟ **(کامپیوتر ۸۵)**

۱) همه محیط‌های نیمه قابل مشاهده، غیرقطعی هستند.
۲) عاملی که به زبان طبیعی محاوره می‌کند در یک محیط نیمه قابل مشاهده عمل می‌کند.
۳) هر عاملی که فقط بخشی از محیط را حس (دریافت) می‌کند. نمی‌تواند عقلایی (rational) باشد.
۴) عاملی که در محیط کاملاً قابل مشاهده عمل می‌کند نیازی به حالات درونی ندارد (rational static).

۴- کدام‌یک از عبارات زیر در مورد عامل‌های هوشمند و محیط عملکرد آن‌ها درست نیست؟ **(فناوری اطلاعات ۸۶)**

۱) عامل انعکاسی ساده نمی‌تواند در محیط‌های پاره‌ای مشاهده‌پذیر (partially observable) بکار رود.
۲) ممکن است محیطی پاره‌ای مشاهده‌پذیر (partially observable) و در عین حال قطعی باشد.
۳) محیطی وجود دارد که در آن همه عامل‌ها عملکرد عقلایی (rational) داشته باشند.
۴) هیچ عاملی نمی‌تواند در یک محیط پاره‌ای مشاهده‌پذیر (partially observable) عملگر عقلایی (rational) داشته باشد.

۵- کدام‌یک از جملات زیر صحیح است؟ **(کامپیوتر ۸۶)**

۱) ممکن است agent function ای وجود داشته باشد که نتوان آن را با هیچ agent program ای پیاده سازی نمود.
۲) یک عامل مبتنی بر دانش (knowledge based) را نمی‌توان با کمک معماری انعکاسی ساده ساخت.
۳) عامل مبتنی بر مدل (Model based) برای محیط‌های با حالات و اعمال پیوسته مناسب نیست.
۴) در محیط‌های کاملاً قابل مشاهده دلیلی برای داشتن حالات داخلی (internal state) نیست.

۶- به کدام‌یک از دلایل زیر استفاده از مدل در یک عامل می‌تواند مفید باشد؟ **(فناوری اطلاعات ۸۷)**

۱) گسستگی محیط
۲) استفاده از روش‌های جستجو
۳) پیوستگی محیط
۴) مشاهده ناپذیر بودن محیط

۷- عاملی که تخته نرد بازی می‌کند در چه محیطی قرار دارد؟ **(کامپیوتر ۸۷)**

۱) ایستا- قطعی- مشاهده‌پذیر- گسسته- ترتیبی
۲) پویا- قطعی- مشاهده‌پذیر- پیوسته- واقعه‌ای
۳) ایستا- غیرقطعی- مشاهده‌پذیر- گسسته- ترتیبی
۴) ایستا- قطعی- مشاهده‌پذیر- گسسته- واقعه‌ای

پاسخنامه سوالات چهارگزینه‌ای سراسری فصل اول

۱- گزینه «۲»

☜جمله ۱ نادرست است زیرا به‌عنوان مثال در عامل راننده تاکسی که محیطی پویا دارد از عامل انعکاسی استفاده می‌شود.

جمله ۲ صحیح است زیرا در محیط‌های پویا عامل خود مختار است.

جمله ۳ نادرست است زیرا تنها در محیط‌های غیرقابل دسترس یا پاره‌ای مشاهده‌پذیر عامل به نگهداری حالت نیاز دارد. محیط ممکن است ایستا باشد ولی باز هم به نگهداری حالات داخلی نیاز باشد.

جمله ۴ نادرست است. خودمختاری به یادگیری نیاز دارد و قدرت یادگیری را به هر عاملی می‌توان داد.

۲- گزینه «۴»

۳- گزینه «۲»

☜ عاملی که به زبان طبیعی با عامل‌های دیگر ارتباط برقرار می‌کند، در هر لحظه تنها کلمات فعلی را توسط حسگرهای خود احساس می‌کند و کلمات قبلی در لحظه فعلی جزء ادراک عامل نیستند. بنابراین محیط این عامل به‌صورت جزئی قابل مشاهده است. معمولاً عامل در چنین محیطی نیاز به حافظه‌ای برای ذخیره‌سازی جملات قبلی می‌باشد.

۴- گزینه «۴»

☜جمله ۱ صحیح است زیرا در محیط‌های پاره مشاهده‌پذیر از عامل واکنشی مبتنی بر مدل استفاده می‌شود.

جمله ۲ صحیح است زیرا اگر جاروبرقی خودکار هیچ حسگری نداشته باشد، محیط پاره‌ای مشاهده‌پذیر اما قطعی می‌باشد. جمله ۳صحیح است به‌عنوان مثال محیط بسیار ساده (کاملاً مشاهده‌پذیر، قطعی، تقسیم‌پذیر، ایستا، گسسته، تک عامله) برای عملکرد عقلانی عامل مشکلی ندارد.

جمله ۴ نادرست است زیرا عملکرد عقلایی می‌تواند در محیط اطلاعاتی که عامل می‌تواند داشته باشد سنجیده شود.

۵- گزینه «۱»

☜ممکن است توابع قابل محاسبه‌ای وجود داشته باشند که روی هیچ ماشینی در یک مدت زمان معقول قابل محاسبه نباشند.

گزینه ۲ نادرست است زیرا به‌عنوان مثال در عامل منطق گزاره‌ای جمله $A \to B$ می‌تواند به‌صورت $if - then$ تبدیل شود.

گزینه ۳ نادرست است زیرا عامل مبتنی بر مدل تنها باید حالات داخلی داشته باشد و به پیوستگی محیط بستگی ندارد. در محیط کاملاً مشاهده‌پذیر، نیازی به حالات داخلی نمی‌باشد. بنابراین گزینه ۴ صحیح است.

۶- گزینه «۴»

☜اگر محیط مشاهده‌پذیر نباشد بایستی وضعیت دنیا در محیط داخلی حفظ شود و در این حالت عامل مبتنی بر مدل نامیده می‌شود.

۷- گزینه «۳»

☜با توجه به جدولی که در متن آمده است عامل تخته نرد باز در محیط ایستا، غیرقطعی، مشاهده‌پذیر، گسسته و ترتیبی قرار دارد.

فصل دوم

بررسی مسائل مختلف و ویژگی‌های آنها

- حل مسأله
- چهارگام اساسی برای حل مسائل
- مسأله تنگ آب
- استراتژی کنترل (راه حل)
- مساله ۸ وزیر
- جهت‌های مختلف استدلال
- معیارهای استراتژی کنترل (جستجو)
- مسائل دنیای واقعی
- مسیریابی
- طرح VLSI
- هدایت ربات

بررسی مسائل مختلف و ویژگی‌های آنها

حل مسأله[1]

مسائل را می‌توان به دو دسته « روتین‌دار و فرمول‌دار» یا «بدون راه‌حل معین و جستجویی» تقسیم‌بندی نمود. مسائل هوش مصنوعی مسائلی هستند که از ابتدا راه‌حل مشخصی ندارند و این به علت ماهیت خود مسأله می‌باشد. به‌عنوان مثال در بازی شطرنج که با انجام هر حرکتی راه‌حل‌های متفاوتی را جستجو می‌کنیم، روش معینی برای پیروزی در بازی وجود ندارد. در این‌گونه مسائل، جستجو منجر به ایجاد درخت یا گراف می‌شود. برای حل اینگونه مسائل، ابتدا باید نسبت به مسأله دانش کافی داشته باشیم و سپس راه‌حل مناسبی برای آن ارائه دهیم.

کسب دانش نسبت به مسأله یعنی اینکه بتوانیم تمامی حقایق [2] و اصول حاکم بر حقایق[3] را به‌دست آوریم.

سیستم تولید[4] دارای اجزای زیر می‌باشد:

۱- واقعیات یا facts که به آن پایگاه داده کلی یا Global data base نیز گفته می‌شود.

۲- قوانین یا Rules که به آن سیستم مجموعه قواعد یا production rules نیز گفته می‌شود.

۳- ارائه راه‌حل (استراتژی کنترل برای حل مسأله) که به آن سیستم کنترلی یا control system نیز گفته می‌شود.

چهار گام اساسی برای حل مسائل

۱- فرموله کردن هدف: یعنی اینکه وضعیت‌های نهایی را تعیین کنیم.

۲- فرموله کردن مسأله: یعنی اینکه چه فعالیت‌ها و وضعیت‌هایی برای رسیدن به هدف وجود دارد.

۳- جستجو: یعنی انتخاب یک دنباله از فعالیت‌هایی که منجر به رسیدن به هدف می‌شود.

۴- اجرا: هنگامی که دنباله فعالیت مطلوب پیدا شد، فعالیت‌های پیشنهادی آن اجرا می‌شود.

به مجموعه fact و ruleها، دانش[5] گفته می‌شود و به محل نگهداری این دانش knowledge base یا پایگاه دانش گفته می‌شود. به مراحل یافتن fact و rule، تعریف مسأله گفته می‌شود. در این مراحل باید اهداف به طورکامل مشخص شوند و حالات اولیه و نهایی تعیین شوند.

مهم‌ترین بخش حل مسأله، استراتژی کنترل است. در این بخش، مجموعه‌ای از قوانین را انتخاب می‌کنیم تا به هدف برسیم. استراتژی کنترل می‌تواند ناآگاهانه یا آگاهانه (هیوریستیک) باشد. در ادامه مسائل گوناگونی را در هوش مصنوعی مطرح می‌کنیم و بخش‌های مختلف آن را تشخیص می‌دهیم.

1- problem solving
2- knowledge
3- Rule
4- production system
5- knowledge