



پردیس علوم  
دانشکده ریاضی، آمار و علوم کامپیوتر

# بررسی الگوریتم‌ها و روش‌های هوش مصنوعی در بازی‌های کامپیوتری

نگارنده

**فاطمه حسین پور**

استاد راهنما: استاد دکتر ساجدی

پایان‌نامه برای دریافت درجه کارشناسی  
در رشته علوم کامپیوتر

۱۵ مرداد ۱۴۰۰

## چکیده

از زمان تولد ایده هوش مصنوعی، بازی ها به پیشرفت تحقیقات هوش مصنوعی کمک نموده اند. بسیاری از نوآوری های هوش مصنوعی، در صنعت بازی سازی رخ داده است و بازی ها تبدیل به ابزاری برای آزمایش کاربردهای هوش مصنوعی در دنیای واقعی شده اند. اما این فقط هوش مصنوعی نیست که از طریق بازی ها پیشرفت می کند، بازی ها نیز از طریق تحقیقات هوش مصنوعی پیشرفته شده اند. در حقیقت، هوش مصنوعی و بازی های کامپیوتری، وابستگی جدایی ناپذیری به یکدیگر دارند. در این پروژه به بررسی هوش مصنوعی در بازی ها پرداخته شده است و الگوریتم های هوش مصنوعی مورد استفاده در بازی ها بررسی گردیده است.

## سپاس گزاری

در این پایان نامه از زحمات استاد گرانقدرم سرکار خانم دکتر هدیه ساجدی تقدیر و تشکر می نمایم.

## پیشگفتار

هوش مصنوعی (AI) در سال های اخیر پیشرفت گسترده ای داشته است. هوش مصنوعی هم یک زمینه پر رونق در تعداد گسترده ای از حوزه های مهم تحقیقاتی و هم یک فن آوری اصلی برای تعداد زیادی از حوزه های کاربردی است. علاوه بر نوآوری های الگوریتمی، پیشرفت سریع در هوش مصنوعی اغلب به افزایش قدرت محاسباتی به دلیل پیشرفت های سخت افزاری نسبت داده می شود. نمونه های موفق هوش مصنوعی از طریق بسیاری از کاربردهای عملی آن می تواند در زندگی روزمره ما تجربه شود. پیشرفت های هوش مصنوعی درک بهتر تصاویر و گفتار، تشخیص احساسات، اتومبیل های خودران، جستجوی وب، طراحی خلاقانه با کمک هوش مصنوعی و بازی، در میان بسیاری از کارهای دیگر قادر ساخته است. برای برخی از این وظایف ماشین آلات به وضعیت تراز انسانی یا فراتر از آن رسیده اند.

با این حال، تفاوتی بین آنچه ماشین ها می توانند به خوبی انجام دهند و آنچه انسان ها در آن مهارت دارند وجود دارد. در روزهای ابتدایی هوش مصنوعی، محققان سیستم های محاسباتی را در نظر گرفتند که می تواند جنبه هایی از هوش انسانی را نشان دهد و مهارت های حل مسئله یا تصمیم گیری در سطح انسان را به دست آورد. این مسائل به صورت مجموعه ای از مفاهیم ریاضی رسمی (قاعده مند) در فضاها نسبتاً باریک و کنترل شده به ماشین ها ارائه شده است که با نوعی دستکاری نماد یا جستجو در فضای نمادین قابل حل است. به طور طبیعی، بازی ها - به ویژه بازی های رومیزی - یک دامنه محبوب برای تلاش های اولیه هوش مصنوعی بوده اند زیرا محیط های تصمیم گیری رسمی (قاعده مند) و بسیار محدود، در عین حال پیچیده هستند.

در طول سالها، تمرکز بسیاری از تحقیقات هوش مصنوعی به سمت کارهایی تغییر کرده است که انجام آنها برای انسانها نسبتاً ساده است اما توصیف نحوه انجام آنها برای ما دشوار است، مانند یادآوری چهره یا تشخیص صدای دوستان از طریق تلفن. در نتیجه، محققان هوش مصنوعی شروع به طرح سوالاتی از قبیل: چگونه هوش مصنوعی می تواند احساسات را تشخیص و بیان کند؟ چگونه هوش مصنوعی می تواند به افراد آموزش دهد، خلاق باشد یا از نظر هنری بدیع باشد؟ چگونه هوش مصنوعی می تواند بازی را که قبلاً ندیده انجام دهد؟ چگونه هوش مصنوعی می تواند از حداقل آزمایشات یاد بگیرد؟ چگونه هوش مصنوعی می تواند احساس گناه کند؟ همه این سوالات هوش مصنوعی را با چالش های جدی روبرو می کند و با وظایفی مطابقت دارد که رسمیت یا تعریف عینی آنها برای ما آسان نیست.

شاید جای تعجب باشد، وظایفی که نیاز به تلاش شناختی نسبتاً کمی از ما دارند، اغلب انجام آن‌ها برای ماشین‌ها بسیار دشوارتر است. باز هم، بازی‌ها دامنه‌ای محبوب برای بررسی چنین توانایی‌هایی فراهم کرده‌اند زیرا بازی‌ها جنبه‌هایی از ماهیت ذهنی دارند که به راحتی رسمی (قاعدۀ مند) نمی‌شوند. از جمله این موارد می‌توان برای نمونه به تجربه بازی یا روند خلاقانه طراحی بازی اشاره کرد.

از زمان تولد ایده هوش مصنوعی، بازی‌ها به پیشرفت تحقیقات هوش مصنوعی کمک نموده‌اند. بازی‌ها نه تنها مسائل جالب و پیچیده‌ای را برای هوش مصنوعی ایجاد می‌کنند - به عنوان مثال، انجام یک بازی خوب - بلکه یک بوم نقاشی برای خلاقیت و بیان ارائه می‌دهند که توسط کاربران (مردم یا حتی ماشین‌ها) تجربه می‌شود. بنابراین، بدون شک، بازی‌ها یک حوزه نادر هستند که در آن علم (حل مسئله) با هنر و تعامل تلفیق می‌شود: این موارد باعث شده است که بازی‌ها به یک حوزه منحصر به فرد و مورد علاقه برای مطالعه هوش مصنوعی تبدیل شوند. اما این فقط هوش مصنوعی نیست که از طریق بازی‌ها پیشرفت می‌کند، بازی‌ها نیز از طریق تحقیقات AI پیشرفته شده‌اند. هوش مصنوعی در چندین جبهه به بهتر شدن بازی‌ها کمک کرده است: در نحوه بازی کردن آنها، در درک عملکردهای داخلی آنها، در طراحی آنها و در درک بازی، تعامل و خلاقیت.

هوش مصنوعی بازی (game AI) - به ویژه بازی ویدیویی یا بازی رایانه‌ای هوش مصنوعی - در پانزده سال وجود خود به عنوان یک زمینه تحقیقاتی مستقل، پیشرفت‌های عمده‌ای داشته است. در طول این مدت، این رشته شاهد ایجاد و رشد جلسات مهم سالانه بوده است - از جمله کنفرانس IEEE در زمینه بازی‌ها و هوش محاسباتی (CIG) و مجموعه کنفرانس هوش مصنوعی و سرگرمی دیجیتال تعاملی - AAAI Artificial Artificial and Interactive Digital Entertainment (AIIDE).

از همان روزهای ابتدایی هوش مصنوعی، ما در چندین زمینه شاهد نمونه‌های موفقیت بی‌شماری بوده‌ایم. امروزه می‌توانیم از AI برای انجام بسیاری از بازی‌ها بهتر از هر انسانی استفاده کنیم، می‌توان ربات‌های AI را طراحی کرد که از بازیکنان انسانی باورپذیرتر و شبیه به انسان باشند، می‌توانیم برای طراحی بازی‌های بهتر و غیر متعارف با AI همکاری کنیم، ما می‌توانیم بهتر بازیکنان را درک کرده و با مدلسازی تجربه کلی بازی، بازی کنیم، ما می‌توانیم با مدلسازی بازی به صورت یک الگوریتم، طرح آن را بهتر درک کنیم و می‌توانیم با تحلیل مقدار زیادی از داده‌های بازیکن، طراحی بازی را بهبود بخشیم و استراتژی کسب درآمد خود را تنظیم کنیم.

در فصل اول این پروژه ابتدا تاریخچه مختصری از هوش مصنوعی و بازی‌ها ارائه گردیده است. سپس به تحولات دانشگاه و صنعت در این حوزه اشاره شده و در ادامه به دلایل استفاده از هوش مصنوعی در بازی‌ها و دلایل استفاده از بازی‌ها برای تحقیقات هوش مصنوعی پرداخته شده است. در فصل دوم روش‌های هوش مصنوعی مورد استفاده در بازی‌ها معرفی شده و در فصل سوم مطالبی در زمینه آینده هوش مصنوعی و بازی‌ها بیان شده است و از کتاب Artificial Intelligence and Games نوشته ی Georgios N. Yannakakis و Julian Togelius بهره گرفته شده است.

# فهرست مطالب

۱	مفاهیم مقدماتی	۱
۱	۱.۱ تاریخچه ای خلاصه از هوش مصنوعی و بازی ها	۱
۴	۱.۱.۱ دانشگاه	۴
۵	۲.۱.۱ صنعت	۵
۶	۲.۱ چرا بازیها برای هوش مصنوعی	۶
۶	۱.۲.۱ بازی ها مسئله های سخت و جذاب هستند	۶
۷	۲.۲.۱ تعامل انسان غنی و کامپیوتر	۷
۸	۳.۲.۱ بازی ها عمومی هستند	۸
۹	۴.۲.۱ برای همه حوزه های هوش مصنوعی چالش هایی وجود دارد	۹
۱۰	۵.۲.۱ بازی ها بهترین تحقق اهداف بلندمدت هوش مصنوعی	۱۰
۱۱	۳.۱ چرا هوش مصنوعی برای بازی ها	۱۱
۱۱	۱.۳.۱ هوش مصنوعی بازی شما را بازی می کند و بهبود می بخشد	۱۱
۱۲	۲.۳.۱ محتوای بیشتر، محتوای بهتر	۱۲
۱۲	۳.۳.۱ تجربه بازیکن و تجزیه و تحلیل داده های رفتاری	۱۲
۱۲	۴.۱ خلاصه	۱۲
۱۳	۲ روش های هوش مصنوعی	۱۳
۱۳	۱.۲ نگارش رفتارهای موقت	۱۳
۱۳	۱.۱.۲ ماشینهای حالت محدود	۱۳
۱۴	۲.۱.۲ درختان رفتار	۱۴
۱۵	۳.۱.۲ هوش مصنوعی مبتنی بر سود	۱۵
۱۶	۲.۲ جستجوی درختی	۱۶
۱۷	۱.۲.۲ جستجوی ناآگاهانه	۱۷
۱۸	۲.۲.۲ جستجوی بهترین-اول	۱۸
۱۹	۳.۲.۲ مینی مکس Minimax	۱۹
۲۰	۴.۲.۲ درخت جستجوی مونت کارلو	۲۰
۲۱	۳.۲ محاسبات تکاملی	۲۱

۲۲	جستجوی محلی	۱.۳.۲
۲۳	الگوریتم های تکاملی	۲.۳.۲
۲۴	یادگیری تحت نظارت	۴.۲
۲۶	یادگیری تقویتی	۵.۲
۲۶	یادگیری Q (Q-Learning)	۱.۵.۲
۲۷	یادگیری بدون نظارت	۶.۲
۲۷	خوشه بندی	۱.۶.۲
۲۸	استخراج الگوی مکرر	۲.۶.۲
۲۹	الگوریتم های ترکیبی قابل توجه	۷.۲
۲۹	تکامل عصبی	۱.۷.۲
۳۰	یادگیری TD با برآوردگرهای تابع شبکه ANN	۲.۷.۲
۳۱	آینده هوش مصنوعی بازی	۳
۳۴	نتیجه گیری	۴

# فصل ۱

## مفاهیم مقدماتی

یکی از مقوله‌های جذاب در دنیای تکنولوژی استفاده از هوش مصنوعی در بازی‌های کامپیوتری و ویدئویی است. بسیاری از نوآوری‌های هوش مصنوعی، در صنعت بازی سازی رخ داده است و بازی‌ها تبدیل به ابزاری برای آزمایش کاربردهای هوش مصنوعی در دنیای واقعی شده‌اند. امروزه، توسعه دهندگان هوش مصنوعی به دنبال راه‌هایی هستند که تا قدرت تفکر، یادگیری و توسعه‌ی فردی را در برنامه‌های کامپیوتری ایجاد کنند؛ نوآوری‌هایی که نه تنها تجربه‌ی گیم‌پلی بازی‌ها را دگرگون می‌کنند، بلکه زندگی روزمره‌ی ما را نیز تحت تاثیر قرار خواهند داد. در حقیقت، هوش مصنوعی و بازی‌های کامپیوتری، وابستگی جدایی ناپذیری به یکدیگر دارند.

از همان روزهای ابتدایی هوش مصنوعی، ما در چندین زمینه از این زمینه تحقیقاتی در حال رشد و شکوفایی، شاهد نمونه‌های موفقیت بی شماری بوده ایم. امروزه می‌توانیم از AI برای انجام بسیاری از بازی‌ها بهتر از هر انسانی استفاده کنیم، می‌توان ربات‌های AI را طراحی کرد که از بازیکنان انسانی باورپذیرتر و شبیه به انسان باشند، می‌توانیم برای طراحی بازی‌های بهتر و غیر متعارف (جنبه‌های) با AI همکاری کنیم، ما می‌توانیم بهتر بازیکنان را درک کرده و با مدل‌سازی تجربه کلی بازی، بازی کنیم، ما می‌توانیم با مدل‌سازی بازی به صورت یک الگوریتم، طرح آن را بهتر درک کنیم و می‌توانیم با تحلیل مقدار زیادی از داده‌های بازیکن، طراحی بازی را بهبود بخشیم و استراتژی کسب درآمد خود را تنظیم کنیم.

در این تحقیق به هوش مصنوعی برای بازی‌ها پرداخته شده است. الگوریتم‌های مورد استفاده در هوش مصنوعی برای بازی‌ها و بررسی برخی نمونه‌های موفق در این حوزه پرداخته شده است.

### ۱.۱ تاریخچه‌ای خلاصه از هوش مصنوعی و بازی‌ها

بازی‌ها و هوش مصنوعی در کنار هم سابقه طولانی دارند. بسیاری از تحقیقات در مورد هوش مصنوعی برای بازی‌ها مربوط به ساخت عامل‌های هوشمند برای انجام بازی‌ها، با یا بدون یک



مولفه یادگیری است. از نظر تاریخی، این اولین و برای مدت طولانی تنها روش استفاده از هوش مصنوعی در بازی ها بوده است. حتی قبل از اینکه هوش مصنوعی به عنوان یک زمینه شناخته شود، پیشگامان اولیه علوم کامپیوتر برنامه های بازی را نوشتند زیرا آنها می خواستند آزمایش کنند که آیا کامپیوترها می توانند وظایفی را که به نظر می رسد نیاز به "هوش" دارند، حل کنند. آلن تورینگ، مخترع اصلی علوم کامپیوتر، (دوباره) الگوریتم Minimax را اختراع کرد و از آن برای بازی شطرنج استفاده کرد. اولین نرم افزاری که موفق به تسلط بر یک بازی شد توسط A. S. Douglas در سال ۱۹۵۲ بر روی نسخه دیجیتالی بازی Tic-Tac-Toe و به عنوان بخشی از پایان نامه دکترای وی در کمبریج برنامه ریزی شد. چند سال بعد، آرتور ساموئل اولین کسی بود که نوعی یادگیری ماشین را ابداع کرد که اکنون یادگیری تقویتی با استفاده از یک برنامه نامیده می شود که یاد گرفت بازی چکرز (Checkers) را با بازی در برابر خودش انجام دهد.

بیشتر تحقیقات اولیه در مورد هوش مصنوعی بازی، معطوف به بازی های رومیزی کلاسیک مانند Chess و Checkers بود. یک تصور وجود داشت که این بازی ها، که در آن ها پیچیدگی های زیادی می تواند ناشی از قوانین ساده ایجاد شود و بهترین ذهن های انسانی را برای صدها یا حتی هزاران سال به چالش می کشید، به نحوی جوهره تفکر را به دست آورد. پس از بیش از سه دهه تحقیق بر روی جستجوی درختی (tree search)، در سال ۱۹۹۴، Chinook بازیکن Checkers مدیریت شد تا ماریون تینسلی قهرمان چکرز جهانی را شکست دهد؛ این بازی در نهایت در سال ۲۰۰۷ حل شد. دهه هاست که روش های بی شمار جدید AI در بازی شطرنج به عنوان "the drosophila of AI" به معنای "ارگانیسم مدل" آزمایش شده است [۴]. نرم افزاری که برای اولین بار قابلیت شطرنج فوق بشری را به نمایش گذاشت، IBM's Deep Blue، متشکل از یک الگوریتم مینی مکس (Minimax) بود که بر روی یک ابر رایانه سفارشی اجرا می شود. Deep Blue در یک رویداد تبلیغاتی که در سال ۱۹۹۷ برگزار شد، در برابر استاد بزرگ حاکم شطرنج، گری کاسپاروف، پیروز شد.

یک نقطه عطف در تحقیقات هوش مصنوعی در بازی ها چند سال قبل از موفقیت Deep Blue و Chinook، نرم افزار تخته نرد به نام TD-Gammon است که توسط Gerald Tesauro در سال ۱۹۹۲ ساخته شده است. TD-Gammon از یک شبکه عصبی مصنوعی استفاده می کند که از طریق یادگیری تفاوت زمانی با بازی چند میلیون بار تخته نرد علیه خود آموزش می بیند. TD-Gammon موفق شد تخته نرد را در سطح یک بازیکن برتر تخته نرد انسانی بازی کند. پس از Deep Blue IBM نمونه موفقیت بعدی، واتسون بود، یک سیستم نرم افزاری قادر به پاسخگویی به سوالات مطرح شده به زبان طبیعی. در سال ۲۰۱۱، واتسون در مسابقه Jeopardy یک بازی تلویزیونی شرکت کرد و یک میلیون دلار در برابر برندگان قبلی بازی برنده شد.

به دنبال موفقیت AI در بازی های رومیزی سنتی، جدیدترین نقطه عطف بازی رومیزی AI در سال ۲۰۱۶ در بازی Go به دست آمد. بلافاصله پس از Chinook و Deep Blue، بازی Go به معیار جدیدی برای بازی AI تبدیل شد با ضریب انشعاب نزدیک به ۲۵۰ و یک فضای جستجوی وسیع چند برابر بزرگتر از شطرنج. در حالی که بازی Go در سطح انسانی انتظار می رفت در آینده

ای دور باشد، در سال ۲۰۱۶ لی سدول - بازیکن حرفه ای ۹ نفره Go - در یک مسابقه پنج بازیه مقابل نرم افزار Google DeepMind's AlphaGo شکست خورد که شامل یک روش یادگیری تقویتی عمیق بود. در سال ۲۰۱۷ AlphaGo در یک بازی Go در سه بازی مقابل بازیکن شماره ۱ رده بندی جهان Jie Ke ، با یک رایانه، برنده مسابقه شد. با این پیروزی، Go آخرین بازی رومیزی کلاسیک عالی بود که کامپیوترها به عملکرد فوق العاده انسانی دست یافته اند. در یک و نیم دهه گذشته، یک انجمن تحقیقاتی در مورد استفاده از هوش مصنوعی در بازی های غیر از بازی های رومیزی (با مکانیزم گسسته مبتنی بر نوبت و جایی که وضعیت کامل بازی برای هر دو بازیکن قابل مشاهده است)، به ویژه بازی های ویدیویی، رشد کرده است. بخش عمده ای از تحقیقات در این انجمن بر توسعه هوش مصنوعی برای انجام بازیها - هر چه موثرتر، یا به سبک انسانها (یا یک انسان خاص)، یا با دارایی های دیگر تمرکز دارد. در سال ۲۰۱۴ هنگامی که الگوریتم های توسعه یافته توسط Google DeepMind یاد گرفتند که چندین بازی را از کنسول کلاسیک بازی ویدیویی Atari 2600 در سطح مهارت فوق العاده انسانی فقط از ورودی های پیکسل خام انجام دهند، یک نقطه عطف قابل توجه در بازی هوش مصنوعی به دست آمد. یکی از بازی های آتاری که سختی بازی کردن خوب با این رویکرد را ثابت نمود MS Pac-Man (Namco, 1982) است. این بازی عملاً در ژوئن ۲۰۱۷ توسط تیم Microsoft Maluuba با استفاده از یک تکنیک یادگیری تقویت معماری پاداش ترکیبی (hybrid reward architecture reinforcement learning technique) حل شد. سایر کاربردهای هوش مصنوعی در بازی های ویدیویی نیز بسیار مهم بوده است. یکی از اینها تولید محتوای رویه ای است. از اوایل دهه ۱۹۸۰، برخی از بازی های ویدیویی برخی از محتوای خود را به جای اینکه توسط انسان طراحی شده باشد، در زمان اجرا به صورت الگوریتمی ایجاد کردند. دو بازی در اوایل بسیار تأثیرگذار بودند. Rogue (Toy and Wichmann, 1980) ، که هر بار که یک بازی جدید شروع می شود، سیاه چال ها و قرار دادن موجودات و وسایل در آنها تولید می شود و Elite (Acornsoft , 1984) ، که یک جهان بزرگ را به عنوان مجموعه ای از دانه های تصادفی ذخیره می کند و با اجرای بازی سیستم های ستاره ای ایجاد می کند. نوید بزرگ بازی هایی که می توانند برخی از محتوای خودشان را تولید کنند این است که شما می توانید محتوای بیشتری را دریافت کنید بدون اینکه آن را با دست طراحی کنید، همچنین می تواند به کاهش فضای ذخیره سازی در میان سایر مزایای بالقوه کمک کند. تأثیر این بازی ها را می توان در موفقیت های اخیر مانند No , Diablo III (Blizzard Entertainment 2012) و Chalice Dungeons of Bloodborne (Sony و Man's Sky ( Hello Games, 2016) Computer Entertainment, 2015) ، مشاهده کرد. نسبتاً اخیراً ، استفاده از هوش مصنوعی برای تجزیه و تحلیل بازی ها و مدل سازی بازیکنان بازی ها نیز آغاز شده است. این امر به دلیل نیاز توسعه دهندگان بازی برای ایجاد بازی هایی که بتواند مخاطبان مختلف را به خود جلب کند، اهمیت بیشتری پیدا می کند، و چون بیشتر بازی ها اکنون از اتصال به اینترنت بهره مند می شوند، به طور فزاینده مورد توجه قرار می گیرد. بازی های فیس بوک مانند FarmVille (Zynga 2009) ، از جمله اولین بازی هایی بودند که از جمع آوری

مداوم داده ها، تجزیه و تحلیل داده ها با پشتیبانی هوش مصنوعی سازگاری نیمه اتوماتیک بازی بهره مند شدند. امروزه، بازی هایی مانند (Nevermind (Flying Mollusk, 2016) می توانند تغییرات احساسی بازیکن را ردیابی کرده و بازی را متناسب با آن تطبیق دهند. اخیراً، تحقیقات در مورد عوامل هوشمند باورپذیر در بازی ها، افق های جدیدی را در هوش مصنوعی بازی گشوده است. یکی از راه های تصور باورپذیری، ساختن عامل های هوشمندی است که بتوانند آزمونهای تورینگ (مداوم) مبتنی بر بازی را پشت سر بگذارند. آزمون تورینگ بازی نوعی از آزمون تورینگ است که در آن تعدادی از داوران باید به درستی حدس بزنند که آیا یک رفتار مشاهده شده در یک بازی رفتار یک انسان است یا یک ربات بازی تحت کنترل هوش مصنوعی. در ادامه، تحولات موازی دانشگاه و صنعت را در این به صورت خلاصه شرح خواهیم داد که دو جامعه موفق به تبادل روشها و انتقال دانش برای یک هدف مشترک دوگانه شدند: پیشرفت AI و بهبود بازیها.

### ۱.۱.۱ دانشگاه

در هوش مصنوعی بازی آکادمیک، دو حوزه اصلی و فعالیت تحقیقاتی مربوطه را از هم تفکیک می کنیم: بازی های رومیزی و بازی های ویدیویی (یا رایانه ای). در زیر، ما دو دامنه را در یک ترتیب زمانی به صورت خلاصه بیان می کنیم.

وقتی نوبت به تحقیق هوش مصنوعی می رسد، بازی های رومیزی کلاسیک مانند Chess، Checkers و Go به وضوح برای کار مفید هستند، زیرا مدل سازی آنها در کد بسیار ساده است و می تواند بسیار سریع شبیه سازی شود - با یک کامپیوتر مدرن به راحتی می توان میلیون ها حرکت در ثانیه انجام داد - که برای بسیاری از تکنیک های هوش مصنوعی ضروری است. همچنین، به نظر می رسد بازی های رومیزی برای داشتن یک بازی خوب و داشتن خاصیتی که "برای یادگیری، یک دقیقه، اما برای تسلط بر آنها برای یک عمر طول می کشد" نیاز به تفکر دارند. در واقع اینگونه است که بازی ها ارتباط زیادی با یادگیری دارند و بازی های خوب می توانند به طور مداوم درباره نحوه بازی آنها بیشتر به ما یاد دهند. در واقع، تا حدی لذت بازی کردن در یادگیری آن است و وقتی چیز دیگری برای یادگیری وجود ندارد، لذت بردن از آنها را تا حد زیادی متوقف می کنیم. بازی های رومیزی از اوایل دهه ۱۹۵۰ تا همین اواخر حوزه غالب تحقیقات AI بود. البته بازی های رومیزی، همچنان یک حوزه محبوب در زمینه تحقیقات AI هستند حتی اگر ورود بازی های ویدیویی و آنلاین در دهه ۱۹۸۰ بخش عمده ای از تمرکز را از آن زمان تغییر داده باشد.

اولین کنفرانس بازی های ویدیویی در سال ۱۹۸۳ در دانشگاه تحصیلات تکمیلی هاروارد برگزار شد. محور اصلی این کنفرانس بر مزایای آموزشی و تأثیر مثبت اجتماعی بازی های ویدیویی بود. تاریخ تولد بازی دیجیتال زمینه AI را می توان حدود سال ۲۰۰۱ در نظر گرفت. مقاله اصلی توسط لایرد و ون لنت پایه های بازی هوش مصنوعی را ایجاد کرد و از کارهای اولیه در این زمینه الهام گرفت. در آن روزهای اولیه هوش مصنوعی در بازی های دیجیتال بیشتر مربوط به انجام بازی ها، معماری عامل برای رفتار NPC بود، گاهی اوقات در یک نمایش تعاملی و مسیریابی.

بیشتر کارهای اولیه در زمینه هوش مصنوعی بازی توسط محققانی با هوش مصنوعی، بهینه سازی و کنترل زمینه و تجربه تحقیق در رفتار انطباقی، رباتیک و سیستم های چند عاملی انجام شده است. دانشگاهیان هوش مصنوعی از بهترین هوش محاسباتی و ابزارهای هوش مصنوعی خود برای تقویت رفتار NPC در پروژه های معمولاً ساده، متمرکز بر تحقیقات، غیر مقیاس پذیر با ارزش و دیدگاه تجاری پایین استفاده کردند.

## ۲.۱.۱ صنعت

اولین بازی های ویدیویی منتشر شده در دهه ۱۹۷۰ شامل مقدار کمی از چیزی بود که ما آن را هوش مصنوعی می نامیم؛ رفتارهای NPC نوشته شده به قوانین ساده ای تکیه شده است، بخشی از آن به دلیل وضعیت ابتدایی تحقیقات هوش مصنوعی است، یا به دلیل سخت افزار ابتدایی آن زمان. با این وجود، به موازات تحولات دانشگاهی، صنعت بازی بتدریج گامهایی را در جهت تلفیق هوش مصنوعی پیشرفته در بازیهای خود در طی روزهای ابتدایی بازی AI برداشت. تعدادی از روش های هوش مصنوعی و ویژگی های بازی که باعث پیشرفت هوش مصنوعی بازی در صنعت شد عبارت است از: اولین کاربرد عمومی شبکه های عصبی در بازی Creatures (Millennium Interactive, 1996)، با هدف مدلسازی رفتار موجودات؛ سیستم حسی پیشرفته نگهبانان در Thief (EIDOS, 1998)؛ تاکتیک های تیمی و صحنه های باورپذیر مبارزه در سری بازی های Halo (Microsoft Studios 2011-2017)- Halo2 به ویژه استفاده از درختان رفتار را در بازی ها؛ هوش مصنوعی مبتنی بر رفتار Blade Runner (Virgin Interactive 1997)؛ تاکتیک های پیشرفته حریف در بازی Half-Life (Valve, 1998)؛ تلفیق تکنیک های یادگیری ماشینی مانند گیرنده ها، درختان تصمیم گیری و یادگیری تقویت همراه با مدل شناختی باور-تمایل-قصد در بازی Black and White (EA, 2000)، عوامل باورپذیر بازی The Sims (Electronic Arts, 2000-2017)؛ سیستم رانندگی یادگیری تقلیدی Forza Motorsport (MS Game Studios, 2005)؛ جهان های ایجاد شده رویه ای در سری های بازی Civ-ization (MicroProse, Activision, Infogrames Entertainment, SA and 2K Games, 1991-2016) و قلعه کوتوله ها (Dwarf Fortress) (Bay 12 Games, 2006)؛ مدیر هوش مصنوعی در بازی Left 4 Dead (Valve, 2008)؛ جنگهای واقع بینانه در بازی Red Dead Redemption (Rockstar Games 2010)؛ اقتباس مبتنی بر شخصیت در بازی Silent Hill: Shattered Memories (Konami, 2010)؛ آموزش تکامل عصبی جوخه ها در بازی Supreme Commander 2 (Square Enix, 2010)؛ روایت های تعاملی در Blood and Laurels (Emily Short, 2014)؛ و جهان های تولید شده رویه ای Spelunky (Mossmouth LLC 2013) و No Man's Sky (Hello Games, 2016).

معیار اصلی که هوش مصنوعی موفقیت آمیز را در بازی های استاندارد تجاری متمایز می کند، همیشه سطح ادغام و درهم آمیختن هوش مصنوعی در طراحی بازی بوده است. یک ترکیب ناموفق در طراحی بازی و هوش مصنوعی ممکن است منجر به رفتارهای NPC غیر قابل توجهی شود. یک نمونه متداول از چنین عدم تطابق بین هوش مصنوعی و طراحی، ناوبری ناگهانی ربات هایی است

که در بن بست سطح گیر می کنند؛ در چنین مواردی یا سطح طراحی متناسب با طراحی هوش مصنوعی (دوباره) در نظر گرفته نمی شود یا هوش مصنوعی به اندازه کافی آزمایش نشده است یا هر دو. از طرف دیگر، یکپارچه سازی موفقیت آمیز هوش مصنوعی در فرآیند طراحی احتمالاً نتایج رضایت بخشی را برای تجربه بازی تضمین می کند.

## ۲.۱ چرا بازیها برای هوش مصنوعی

دلایل مختلفی وجود دارد که بازی ها حوزه ایده آل برای مطالعه هوش مصنوعی را ارائه می دهند. در این بخش مهمترین آنها را بیان می کنیم.

### ۱.۲.۱ بازی ها مسئله های سخت و جذاب هستند

بازی ها به دلیل تلاش و مهارت های مورد نیاز مردم برای تکمیل آنها یا حل کردن آنها در مورد معماها، جذاب هستند. پیچیدگی و جذابیت بازی ها به عنوان مسئله ای است که آنها را برای هوش مصنوعی مطلوب می کند. بازی ها سخت هستند زیرا فضاهای حالت محدود آنها، مانند استراتژی های احتمالی یک عامل هوشمند، اغلب گسترده هستند. پیچیدگی آنها به عنوان یک حوزه افزایش می یابد زیرا فضاهای جستجوی گسترده آنها اغلب دارای فضاهای کوچک قابل اجرا هستند (فضاهای راه حل). بعلاوه، معمولاً ارزیابی صحیح خوب بودن هر وضعیت بازی سخت (یا حتی غیرممکن) است.

از منظر پیچیدگی محاسباتی، بسیاری از بازی ها NP-hard هستند (NP به "زمان چند جمله ای غیرقطعی" اشاره دارد)، بدین معنی که پیچیدگی بدترین حالت "حل" آنها بسیار زیاد است. به عبارت دیگر، در حالت کلی یک الگوریتم برای حل یک بازی خاص می تواند برای مدت زمان طولانی اجرا شود. بسته به ویژگی های بازی، پیچیدگی می تواند به میزان قابل توجهی متفاوت باشد. لازم به ذکر است که این ویژگی پیچیدگی محاسباتی ارتباط چندانی با سختی بازی ها برای انسانها ندارد و لزوماً در مورد چگونگی روشهای ابتکاری هوش مصنوعی نمی تواند حرف زیادی بزند. با این حال، واضح است که حداقل در تئوری و برای نمونه هایی با اندازه دلخواه، بسیاری از بازی ها بسیار سخت هستند.

تحقیقات در مورد ظرفیت هوش مصنوعی در انجام بازی های سخت و پیچیده از طریق تعدادی بازی مهم انجام شده است. بازی هایی مانند شطرنج، چکرز، Go، StarCraft و Super Mario Bros از مهمترین آن ها هستند. بازی رایانه Go با چندین دهه تحقیق فعال، یکی از معیارهای اصلی و سنتی بازی هوش مصنوعی است. به عنوان معیار پیچیدگی مسئله، یک بازی معمولی در Go حدود  $10^{170}$  حالت دارد.

بازی استراتژیک در زمان واقعی (Blizzard Entertainment, 1998) StarCraft را می توان به عنوان یکی از سخت ترین بازی ها برای رایانه ها توصیف کرد. تا سال ۲۰۱۸، بهترین ربات های StarCraft فقط به سطح بازیکنان آماتور می رسیدند. پیچیدگی بازی عمدتاً از وظیفه

چند هدفه و کنترل چندین واحد مختلف و غیر مشابه در محیط بازی با اطلاعات جزئی ناشی می شود. اگر چه تقریب فضای حالت StarCraft کار ساده ای نیست، بر اساس یک مطالعه اخیر [۷]، بازی معمولی حداقل  $10^{1685}$  حالت ممکن دارد. در مقام مقایسه، تعداد پروتون ها در جهان مشاهده شده فقط حدود  $10^{80}$  است.

مطمئناً می توان بازی هایی را طراحی کرد که هدف آنها سخت تر باشد، اما هیچ کس تضمین نمی کند که کسی بخواهد آن بازی ها را انجام دهد. هنگام انجام تحقیقات هوش مصنوعی، کار بر روی بازی هایی که مردم به آنها اهمیت می دهند بدان معناست که شما روی مسائل مرتبط کار می کنید. این به این دلیل است که بازی ها برای به چالش کشیدن مغز انسان طراحی شده اند و بازی های موفق معمولاً در این زمینه مهارت دارند. بازی StarCraft (Blizzard Entertainment) (1998 و بعد از آن StarCraft II (Blizzard Entertainment 2010) توسط میلیون ها نفر در سراسر جهان بازی می شود.

## ۲.۲.۱ تعامل انسان غنی و کامپیوتر

بازی های رایانه ای طبق تعریف رسانه ای پویا هستند و مسلماً یکی از غنی ترین اشکال تعامل انسان و رایانه (HCI) را ارائه می دهند. غنای تعامل با توجه به گزینه های موجود یک بازیکن در هر لحظه و راه های (روش های) یک بازیکن برای تعامل با رسانه تعریف می شود. در بازی هایی مانند StarCraft II (Blizzard Entertainment 2010) گزینه های موجود برای بازیکن به فضای عمل بازی و پیچیدگی مربوط به آن مرتبط هستند. بعلاوه، روشهایی که امروزه ممکن است برای تعامل با بازیها فراتر از صفحه کلید سنتی، ماوس لپ تاپ های و تبلت، تا کنترل کننده های بازی، فیزیولوژی مانند تغییر ضربان قلب، حرکت بدن مانند وضعیت بدن و حرکات، متن و گفتار باشد. در نتیجه، بسیاری از بازی ها به راحتی لیست بیت های اطلاعاتی رد و بدل شده بین آنها و کاربرانیشان در هر ثانیه را در مقایسه با سایر رسانه های HCI قرار می دهند.

واضح است، بازی ها یکی از بهترین و معنی دارترین حوزه ها را برای تحقق حلقه عاطفی ارائه می دهند، که چارچوبی را تعریف می کند که قادر به استخراج، شناسایی و پاسخ موفق به الگوهای شناختی، رفتاری و احساسی کاربر آن است. پتانسیلی که بازی ها می توانند روی بازیکنان تأثیر بگذارند عمدتاً به دلیل توانایی آنها در قرار دادن بازیکن در حالت تعامل مداوم با بازی است که پاسخ های پیچیده شناختی، عاطفی و رفتاری را به بازیکن منتقل می کند. این حالت تعامل مداوم با اشکال سریع و چند حالتی تعامل کاربر غالباً غنی می شود که اغلب در بازی ها امکان پذیر است. از آنجا که هر بازی دارای یک بازیکن - یا تعدادی از بازیکنان - است تعامل بین بازیکن و بازی برای تحقیقات هوش مصنوعی مهم است زیرا به الگوریتم ها امکان دسترسی به محرک های تجربه بازیکن غنی و تظاهرات عاطفی بازیکن را می دهد. چنین مظاهر (تجلیات) پیچیده ای را نمی توان به طور پیش پا افتاده با روش های استاندارد در یادگیری ماشین و علم داده گرفت. بدون شک، مطالعه تعامل بازی-بازیکن از طریق هوش مصنوعی نه تنها دانش ما را در مورد رفتار و احساسات انسان ارتقا می بخشد بلکه به طراحی تعامل بهتر انسان و کامپیوتر نیز کمک می کند. در نتیجه،

مرزهای روشهای هوش مصنوعی را بیشتر برطرف می کند تا به چالش های تعامل مبتنی بر بازی بپردازد.

## ۳.۲.۱ بازی ها عمومی هستند

در حالی که بازی های ویدئویی ، در دهه ۱۹۸۰ ، به عنوان یک فعالیت خوب برای کسانی که به بازی های ویدئویی یا کنسول هایی مانند Atari 2600 دسترسی داشتند، معرفی شد، آنها در دهه ۲۰۱۰ به تدریج به یک صنعت چند میلیارد دلاری تبدیل شدند، یک درآمد بازار جهانی بالاتر از هر شکل دیگر صنعت خلاق، از جمله فیلم و موسیقی. در سال ۲۰۱۸، بازی ها در کل درآمد تقریبی ۱۰۰ میلیارد دلاری در جهان دارند که انتظار می رود تا سال ۲۰۱۹ به ۱۲۰ میلیارد دلار برسد.

اما چرا بازی ها اینقدر محبوب شدند؟ فراتر از استدلال آشکار این که بازی ها می توانند با ارائه ظرفیت های تعاملی، انگیزه و تعامل ذاتی کاربر را با یک محیط مجازی افزایش دهند، این پیشرفت های فن آوری در ۴۰ سال گذشته بود که جمعیت بازیکنان را به شدت تغییر داد [۵]. بازی های اوایل دهه ۱۹۸۰ فقط در ماشین های سرگرمی بازی انجام می شد. با این حال، امروزه می توان آنها را با استفاده از تعداد زیادی دستگاه از جمله رایانه شخصی (به عنوان مثال، بازی های چند نفره آنلاین) ، تلفن همراه ، لبتاپ ، دستگاه دستی، دستگاه واقعیت مجازی یا کنسول بازی کرد. فراتر از پیشرفت های تکنولوژیکی که باعث افزایش قابلیت دسترسی و عمومی شدن بازی شده است، این فرهنگ نیز است که از یک رسانه جدید پیروی می کند و آن را به شکل جدیدی از هنر و بیان توسعه می دهد. نه تنها صحنه مستقل طراحی و توسعه بازی به این فرهنگ کمک کرده است ، بلکه همچنین ارتباط گسترده ای از اهداف و موضوعات بازی ها بیش از سرگرمی به دست آمده است: بازی برای هنر، بازی به عنوان هنر، بازی برای تغییر، بازی های تعاملی فیزیکی، بازی برای تحصیل، بازی برای آموزش و سلامت، بازی برای کشف علمی، و بازی برای فرهنگ و موزه ها. به طور خلاصه، نه تنها بازی ها در همه جا وجود دارند و در زندگی روزمره ما حضور دارند ، بلکه به طور کلی ارزش های فرهنگی و اجتماعی ما را نیز شکل می دهند.

با محبوبیت بیشتر بازی ها، رشد کمیت و پیچیدگی آنها، راه حل های جدید AI برای پاسخگویی به چالش های جدید فناوری به طور مداوم مورد نیاز است. این جایی است که هوش مصنوعی با حوزه ای با پشتوانه صنعتی قوی و تمایل به پشتیبانی از فن آوری پیشرفته برای بهبود تجربه بازیکن مواجه می شود. علاوه بر این ، حوزه های بسیار کمی برای هوش مصنوعی امتیاز دسترسی روزانه به محتوا و داده های جدید را از کاربردهای عمومی (محبوب) آنها ارائه می دهند.

هرچه افراد بیشتری (بیشتر) بازی کنند ، محتوای بیشتری برای بازی ها مورد نیاز است. ایجاد محتوا به تلاش نیاز دارد ، اما با گذشت سالها ، سازوکارهایی ساخته شده اند که به ماشین ها و بازیکنان امکان می دهد انواع مختلفی از محتوا را در بازی ها طراحی و ایجاد کنند. بازی ها به تدریج به عنوان برنامه های نرم افزاری پرمحتوا توسعه یافته اند که مطالبی را می طلبند که هم در بازی استفاده مستقیم داشته و هم از تازگی کافی برخوردار باشد. تقاضای طاقت فرسا برای تجربه های جدید و بدیع بازی از طرف جامعه عظیم کاربران دائماً مرزهای خلاقیت انسانی و محاسباتی

را به سمت زمینه های جدید و به طور طبیعی هوش مصنوعی سوق می دهد. یکی دیگر از مهمترین تأثیرات محبوبیت بازی ها، تولید گسترده داده ها از بازی و رفتار بازیکن است. از اواخر دهه ۲۰۰۰، شرکت های بازی سازی به خدمات دقیق تله متری بازی دسترسی دارند که به آنها امکان می دهد خریدهای بازی، ریزش و تعامل مجدد یا پیشرفت بازی برای اشکال زدایی از بازی یا تجربه بازیکنان را ردیابی و کنترل کنند. چالشهای الگوریتمی مواجه شده در اینجا از چالشهای کلی داده های کلان و تحقیقات داده کاوی کلان پیروی می کنند، که شامل فیلتر کردن داده ها در هنگام دستیابی به داده ها، تولید فراداده ها، استخراج اطلاعات از داده های اشتباه و گمشده، تجزیه و تحلیل خودکار داده ها در میان مجموعه های داده غیر مشابه، و ... است. خوشبختانه امروزه برخی از این مجموعه های داده به طور آشکار برای تجزیه و تحلیل بازی و تحقیقات داده کاوی بازی در دسترس هستند.

## ۴.۲.۱ برای همه حوزه های هوش مصنوعی چالش هایی وجود دارد

بازی ها تمام زمینه های اصلی هوش مصنوعی را به چالش می کشند. این را می توان با استفاده از تعدادی از زمینه های پذیرفته شده از AI و بحث در مورد چالش های موجود برای آن مناطق در بازی ها، مشاهده کرد. پردازش سیگنال، برای مبتدیان، در بازی ها با چالش های بزرگی روبرو می شود. به عنوان مثال، داده های بازیکنان نه تنها در تفکیک های مختلف - در رویدادهای درون بازی در مقابل حالت سر در مقابل فیزیولوژی بازیکن - وجود دارد، بلکه از چندین روش تعامل سریع در محیطی ایجاد می شود که الگوهای پیچیده شناختی و عاطفی را برای بازیکن استخراج می کند. تعامل چند حالت و همجوشی چند حالت در ساختن عوامل گفتگوی مجسم و شخصیت های مجازی از مسائل سخت و مشکل هستند. علاوه بر این، پیچیدگی کار پردازش سیگنال در بازی ها به دلیل ماهیت مکانی-زمانی سیگنال ها که ناشی از تعامل غنی و سریع گام با بازی است، افزایش می یابد.

ارتباط تاریخی بین دستاوردهای هوش مصنوعی و بازی ها در حال حاضر شواهد روشنی را نشان می دهد که تمام مناطق فوق به صورت متداول توسط بازی ها به چالش کشیده می شوند. سرانجام، وقتی نوبت به برنامه ریزی و پیمایش می رسد، بازی ها به طور سنتی محیط هایی با پیچیدگی بالا و فزاینده را برای الگوریتم های آزمایش شده ارائه می دهند. در حالی که بازی هایی مانند StarCraft به طور واضح نقاط عطف اصلی را برای الگوریتم های برنامه ریزی مشخص می کنند، ناوبری و مسیر یابی از طریق محیط های بازی شبیه سازی شده و واقع گرایانه با چندین عامل به درجه بلوغ خاصی رسیده اند. یک مزیت دیگر بازی ها به عنوان یک دامنه برای برنامه ریزی رفتاری این است که آنها یک آزمایشگاه واقع بینانه اما در عین حال راحت تر و ارزان تر در مقایسه با رباتیک ارائه می دهند. فراتر از آزمایش گسترده و پیشرفت انواع  $A^*$  از طریق بازی، انواع محبوب و بسیار موثر جستجوی درخت مانند الگوریتم جستجوی درخت مونت کارلو [۳] در پاسخ به مشکلات ناشی از بازی اختراع شده است.



## ۵.۲.۱ بازی‌ها بهترین تحقق اهداف بلندمدت هوش مصنوعی

یکی از سوالاتی که مدت زیادی AI با آن روبرو است این است که هدف نهایی بلند مدت AI چیست؟ در حالی که مباحث و کتابهای زیادی به این موضوع اختصاص یافته است، تلاش مشترک نویسندگان برای پرداختن به این سوال، زمینه‌های هوش اجتماعی و تعامل عاطفی، خلاقیت (محاسباتی) و هوش عمومی را به عنوان مهمترین اهداف بلند مدت هوش مصنوعی نشان می‌دهد. سه حوزه فوق‌الذکر در مجموع به سیستم‌های هوش مصنوعی بهتر کمک می‌کنند و بازی‌ها به بهترین وجه این سه هدف را تحقق می‌بخشند. این سه هدف بلند مدت زمینه‌های تحقیقات مرزی را برای هوش مصنوعی بازی تعریف می‌کند.

محاسبات عاطفی زمینه چند رشته‌ای تحصیل در بین رشته‌های علوم کامپیوتر، علوم شناختی و روانشناسی است که طراحی و توسعه نرم افزار هوشمندی را که قادر به استخراج، تشخیص، مدل سازی و بیان احساسات و هوش اجتماعی است، بررسی می‌کند. هدف نهایی از محاسبات عاطفی تحقق به اصطلاح حلقه عاطفی است، سیستمی را تعریف می‌کند که قادر است با موفقیت احساسات کاربر خود را استخراج، شناسایی و پاسخ دهد. به طور طبیعی، هر دو جنبه عاطفی و اجتماعی هوش برای سیستمی مرتبط است که حلقه عاطفی را تحقق می‌بخشد.

بازی‌ها می‌توانند درک بسیار معناداری از حلقه عاطفی و تعامل عاطفی ارائه دهند. بازیها طبق تعریف هم سرگرم کننده هستند (چه برای رضایت صرف، آموزش یا تحصیل استفاده شوند) و هم فعالیت‌های تعاملی که در دنیاهای خیالی انجام می‌شود. علاوه بر این، بازی‌ها برای ارائه تجربه‌های احساسی تحت تأثیر بازخورد بازیکنان طراحی شده‌اند و بازیکنان مایلند دوره‌های بازی ناامید کننده، مضطرب و ترسناک را برای تجربه درگیری تجربه کنند. برای این منظور، یک کاربر در شرایط بازی - بیش از هر شکل دیگری از تعامل انسان و کامپیوتر - به طور کلی در معرض تغییرات مبتنی بر عاطفه در تعامل و تأثیرات وضعیت احساسی خود است.

خلاقیت محاسباتی پتانسیل نرم افزار را برای تولید مستقل نتایج مورد بررسی قرار می‌دهد که می‌تواند فرایندهای خلاقانه یا الگوریتمی تلقی شود که خلاق به نظر می‌رسند. بازی‌های رایانه‌ای را می‌توان به عنوان حوزه کاربرد قاتل (killer application) برای خلاقیت محاسباتی در نظر گرفت. این تنها ویژگی‌های منحصر به فرد آنها نیست. از همه مهمتر ماهیت چند وجهی آنهاست. به طور خاص، این تلفیق دامنه‌های خلاقانه متعدد و بسیار متنوع است - هنر تصویری، طراحی صدا، طراحی گرافیک، طراحی تعامل، روایت، فیلمبرداری مجازی، زیبایی شناسی و زیبا سازی محیط - در یک برنامه نرم افزاری واحد که باعث می‌شود بازی‌ها به عرصه ایده آل برای مطالعه خلاقیت محاسباتی تبدیل شوند. همچنین توجه به این نکته مهم است که هر شکل هنری (یا وجهی) که در بازی‌ها ملاقات می‌کند، تجربیات مختلفی را برای کاربران خود به همراه دارد. تلفیق آنها در نرم افزار نهایی با هدف قرار دادن مخاطبان بسیار زیاد و متنوع یک چالش اضافی برای خلاقیت محاسباتی است. در نتیجه، مطالعه خلاقیت محاسباتی در داخل و برای بازی‌های رایانه‌ای [۸] هم در زمینه هوش مصنوعی و هم در حوزه بازی‌ها پیشرفت می‌کند.

همچنین هوش مصنوعی بیش از هر حوزه دیگری ظرفیت کلی هوش ماشین‌ها را در دامنه بازی‌ها بررسی کرده است به لطف ویژگی‌های ایده آل بازی‌ها برای آن منظور: مسائل کنترل شده در

عین حال جالب و محاسباتی سخت [۶]. به طور خاص، ظرفیت هوش مصنوعی برای انجام خوب بازی های دیده نشده - به عنوان مثال، بازی های عمومی - در سال های اخیر پیشرفت های زیادی داشته است.

## ۳.۱ چرا هوش مصنوعی برای بازی ها

استفاده های مختلف از هوش مصنوعی در بازی ها به دلایلی برای طراحی بازی های بهتر مفید است. در این بخش ما بر مزایای حاصل از اجازه دادن به هوش مصنوعی برای انجام بازی، تولید محتوا و تجزیه و تحلیل تجربه و رفتار بازیکن تمرکز می کنیم.

### ۱.۳.۱ هوش مصنوعی بازی شما را بازی می کند و بهبود می بخشد

هوش مصنوعی می تواند فقط با انجام بازی ها از چند طریق بهبود یابد. صنعت بازی معمولاً از هوش مصنوعی بازی های خود - به ویژه AI غیر بازیکن یا حریف - تعریف و تمجید می کند، وقتی هوش مصنوعی بازی به ارزش تجاری بازی می افزاید، به بررسی بهتر بازی کمک می کند و تجربه بازیکن را بهبود می بخشد. این که آیا هوش مصنوعی زیربنایی مبتنی بر یک درخت رفتار ساده است، یک هوش مصنوعی مبتنی بر سود یا یک گزینه دیگر در یک ماشین کنترل کننده واکنشی آموخته پیشرفته از نظر ارتباط محدود تا زمانی که اهداف فوق الذکر را تأمین کند از اهمیت محدودی برخوردار است. یک راه حل غیر متعارف و موثر برای یک وظیفه NPC اغلب می تواند عاملی مهم باشد که استراتژی های مدیریت، بازاریابی و کسب درآمد را در حین تولید و بعد از آن شکل می دهد.

هوش مصنوعی با دو هدف اصلی بازی ها را انجام می دهد: بازی خوب و باورپذیری بازی (یا شبیه انسان یا جالب). به علاوه هوش مصنوعی می تواند شخصیت بازیکن یا شخصیت غیر بازیکن بازی را کنترل کند. هوش مصنوعی که به خوبی شخصیت بازیکن را بازی می کند، روی بهینه سازی عملکرد بازی متمرکز است - عملکرد به عنوان درجه ای است که بازیکن فقط اهداف بازی را برآورده می کند. چنین هوش مصنوعی می تواند برای تست اتوماتیک بازی و برای ارزیابی کلیت طراحی بازی از اهمیت فوق العاده ای برخوردار باشد. در عوض هوش مصنوعی که به عنوان یک شخصیت غیر بازیکن به خوبی بازی می کند، می تواند قابلیت تنظیم پویایی مسئله و مکانیزم های تعادل خودکار بازی را فراهم کند که به نوبه خود تجربه بازیکن را شخصی سازی و افزایش می دهد اگر تمرکز هوش مصنوعی بر روی کنترل شخصیت های بازیکنی باشد که باورپذیرانه یا شبیه به انسان بازی می کنند هوش مصنوعی می تواند به عنوان وسیله ای برای اشکال زدایی تجربه بازیکن یا نمایش بازی واقع گرایانه برای اهداف طراحی باشد. سرانجام، بازی ای که دارای تعامل غنی با NPC است تنها می تواند از هوش مصنوعی بهره ببرد که NPC هایی را بیان می کند که رفتارهای انسان مانند و باورپذیری را به تصویر می کشد.

## ۲.۳.۱ محتوای بیشتر، محتوای بهتر

دلایل مختلفی برای علاقه مند شدن طراحان و توسعه دهندگان بازی به هوش مصنوعی و به ویژه تولید محتوا وجود دارد. اولین و تاریخی ترین دلیل، مصرف حافظه است. محتوا را می توان معمولاً با ننگه داشتن "گسترش نیافته" تا زمان نیاز فشرده کرد. بعلاوه، تولید محتوا ممکن است باعث ایجاد یا تقویت بیشتر خلاقیت انسان شود و اجازه ظهور انواع کاملاً جدیدی از بازیها، ژانرهای بازی یا فضاهای کاملاً جدید کاوش و بیان هنری را بدهد [۸]. علاوه بر این، اگر بتوان محتوای جدیدی را با تنوع، کیفیت و کمیت کافی تولید کرد، ممکن است ایجاد بازی های واقعاً بی پایان با ارزش بازی نهایی امکان پذیر باشد. سرانجام، هنگامی که تولید محتوا با جنبه های بازی همراه باشد، می توانیم انتظار داشته باشیم که بازی شخصی و انطباقی از طریق اصلاح محتوا ظهور کند. برخلاف سایر زمینه های هوش مصنوعی بازی - مانند بازی عمومی که ممکن است بیشتر به عنوان یک فعالیت دانشگاهی در نظر گرفته شود - تولید محتوا یک ضرورت تجاری است [۸]. قبل از علاقه دانشگاهی به تولید محتوا، سیستم های تولید محتوا سابقه طولانی در پشتیبانی از بازی های استاندارد تجاری برای ایجاد تجارب جذاب و در عین حال غیر قابل پیش بینی بازی داشتند. به طور طبیعی، بازی هایی که از سیستم های تولید محتوای پیچیده ای بهره می برند می توانند برای فناوری های خود ستایش کنند.

## ۳.۳.۱ تجربه بازیکن و تجزیه و تحلیل داده های رفتاری

استفاده از هوش مصنوعی برای درک تجربه بازیکن می تواند روند طراحی بازی ها را تقویت و ارتقا دهد. طراحان بازی معمولاً مجموعه ای از مکانیک و پویایی بازی را که الگوهای تجربه ای را که می خواهند بازیکن را به کار گیرند، کشف می کنند و آزمایش می کنند. حالت های بازیکن مانند تعامل، ترس و استرس، ناامیدی، انتظار و چالش جنبه های مهمی از طراحی تجربه بازیکن را تعریف می کند که به ژانر، روایت و اهداف بازی بستگی دارد. در نتیجه، جام مقدس طراحی بازی - که همان تجربه بازیکن است - می تواند برای هر بازیکن بهبود یافته و متناسب با آن باشد بلکه از طریق تعامل مبتنی بر تجربه غنی تر، تقویت شود. بعلاوه، به عنوان یک نتیجه مستقیم از طراحی بهتر و سریعتر، کل فرایند توسعه بازی تقویت و بهبود می یابد.

## ۴.۱ خلاصه

هوش مصنوعی رابطه ای دیرینه و خوب با بازی ها دارد. الگوریتم های AI از طریق بازی ها پیشرفته یا حتی اختراع شده اند. بازی ها، طراحی و توسعه آنها، به نوبه خود، عمدتاً توسط نقش های زیادی که AI در بازی ها بر عهده داشته است، بهره مند شده اند. در فصل بعدی، روش ها و الگوریتم های اصلی مورد استفاده در زمینه هوش مصنوعی بازی را بیان می کنیم.

## فصل ۲

# روش های هوش مصنوعی

این فصل تعدادی از روشهای اساسی هوش مصنوعی را ارائه می دهد که معمولاً در بازیها استفاده می شود. در این فصل، بخشهای مربوط به هوش مصنوعی را به شش دسته تقسیم می کنیم: نویسندگی موقت، جستجوی درختی، محاسبات تکاملی، یادگیری تحت نظارت، یادگیری تقویتی و یادگیری بدون نظارت [۱].

### ۱.۲ نگارش رفتارهای موقت

در این بخش، ما اولین و مسلماً محبوب ترین کلاس روش های هوش مصنوعی را برای توسعه بازی بحث می کنیم. ماشینهای حالت محدود، درختان رفتار و هوش مصنوعی مبتنی بر سود، روشهای نگارش رفتارهای موقت هستند که به طور سنتی بر کنترل شخصیتهای غیر بازیکن در بازیها حاکم هستند. غلبه آنها با این واقعیت مشهود است که امروزه اصطلاح بازی AI در صحنه توسعه بازی مترادف با استفاده از این روش ها است.

#### ۱.۱.۲ ماشینهای حالت محدود

ماشین حالت محدود (FSM) - و انواع FSM مانند FSM های سلسله مراتبی - روش هوش مصنوعی بازی است که تا اواسط دهه ۲۰۰۰ بر روندهای کنترل و تصمیم گیری شخصیت های غیر بازیکن در بازی ها حاکم بود. FSM به حوزه سیستم های دانش خبره تعلق دارند و به صورت گراف نشان داده می شوند. نمودار FSM نمایشی خلاصه از مجموعه ای از اشیا، نمادها، رویدادها، کنش ها یا خصوصیات به هم پیوسته است که نیاز به طراحی (نمایش) دارد. به طور خاص، نمودار شامل گره ها (حالت ها) است که برخی انتزاع های ریاضی را نشان می دهد و لبه ها (انتقال ها) که نشان دهنده یک رابطه شرطی بین گره ها است. FSM می تواند همزمان فقط در یک حالت

باشد. در صورت تحقق شرط در گذار مربوطه، وضعیت فعلی می تواند به حالت دیگری تغییر کند. به طور خلاصه، FSM توسط سه جزء اصلی تعریف می شود:

- تعدادی از حالات که اطلاعات مربوط به یک وظیفه را ذخیره می کنند - به عنوان مثال، شما در حال حاضر در حالت کاوش هستید.
- تعدادی انتقال بین حالتها که نشان دهنده تغییر وضعیت است و با شرایطی توصیف می شود که باید انجام شود - به عنوان مثال، اگر صدای شلیک شنیدید، به حالت هشدار بروید.
- مجموعه اقداماتی که باید در هر حالت دنبال شود - به عنوان مثال، در حالی که در حالت کاوش هستید به طور تصادفی حرکت کنید و به دنبال مخالفان باشید.

طراحی، اجرا، تجسم و اشکال زدایی FSM ها فوق العاده ساده است. بعلاوه آنها ثابت کرده اند که در طول سالهای زندگی مشترکشان با بازی ها به خوبی کار می کنند. با این وجود، طراحی آنها در مقیاس وسیع می تواند بسیار پیچیده باشد و از این رو، از نظر محاسباتی به برخی از وظایف موجود در هوش مصنوعی بازی محدود می شود. یک محدودیت حیاتی دیگر در FSM ها (و همه روش های نگارش موقت) این است که انعطاف پذیر و پویا نیستند (مگر اینکه به طور هدفمند طراحی شده باشند). پس از اتمام طراحی، آزمایش و رفع اشکال، فضای محدودی برای سازگاری و تکامل وجود دارد. در نتیجه، FSM در نهایت رفتارهای بسیار قابل پیش بینی در بازی ها را به تصویر می کشد. تا حدودی می توانیم با ارائه انتقالات به عنوان قوانین مبهم یا احتمالات، بر چنین نقصی غلبه کنیم.

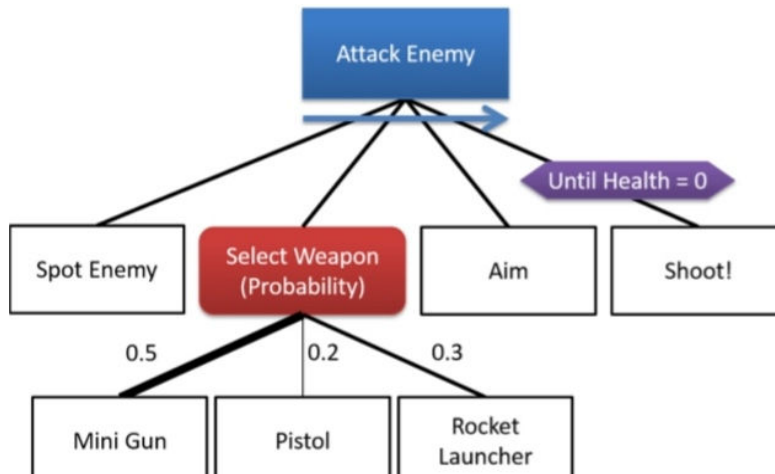
## ۲.۱.۲ درختان رفتار

یک درخت رفتار (BT)، یک سیستم دانش خبره است که، به طور مشابه، FSM انتقال بین مجموعه ای از وظایف (یا رفتارها) را مدل سازی می کند. قدرت BT در مقایسه با FSM مدولار بودن آن است: اگر خوب طراحی شود، می تواند رفتارهای پیچیده متشکل از کارهای ساده را به همراه داشته باشد. تفاوت اصلی بین BT و FSM (یا حتی FSM های سلسله مراتبی) در این است که آنها بیشتر از رفتارها نسبت به حالت ها تشکیل شده اند. همانند ماشین های حالت محدود، BT ها به راحتی طراحی، آزمایش و اشکال زدایی می شوند، که پس از کاربرد موفقیت آمیز در بازی هایی مانند Halo 2 (Microsoft Game Studios, 2004) و Bioshock (2K Games, 2007)، آنها را در صحنه توسعه بازی مسلط می کند.

- توالی (مستطیل آبی را در شکل ببینید): اگر رفتار فرزند موفقیت آمیز باشد، توالی ادامه پیدا می کند و در نهایت اگر همه رفتارهای فرزند موفق شود گره والدین موفق می شود. در غیر این صورت توالی از کار می افتد.
- انتخابگر (مستطیل گرد قرمز را در شکل ببینید): دو نوع اصلی گره انتخابگر وجود دارد: احتمالات و انتخاب کننده های اولویت. هنگامی که از یک انتخابگر احتمال استفاده می

شود، رفتارهای فرزند بر اساس احتمالات والد- فرزند تعیین شده توسط طراح BT انتخاب می شوند. از طرف دیگر در صورت استفاده از انتخاب کننده های اولویت، رفتارهای فرزند در یک لیست مرتب شده و یکی پس از دیگری امتحان می شوند. فارغ از نوع انتخابگر استفاده شده، اگر رفتار فرزند موفق شود، انتخاب کننده موفق می شود. اگر رفتار فرزند شکست بخورد، فرزند بعدی به ترتیب (در انتخابگرهای اولویت دار) انتخاب می شود یا انتخاب کننده (در انتخابگرهای احتمال) شکست می خورد.

- دکوراتور (به شش ضلعی بنفش شکل نگاه کنید): گره دکوراتور باعث پیچیدگی و افزایش ظرفیت رفتار فرزند تنها می شود. مثالهای دکوراتور شامل تعداد دفعاتی است که رفتار فرزند انجام می شود یا زمانی که برای انجام کار به رفتار فرزند اختصاص داده شده است.



شکل ۱.۲: یک نمونه درخت رفتار

در مقایسه با FSM، BT ها انعطاف پذیرتر هستند و آزمایش آنها آسان تر است. با این حال هنوز هم از اشکالات مشابهی رنج می برند. به طور خاص، پویایی آنها با توجه به اینکه نمایندگان دانش ساکن هستند، نسبتاً کم است. گره های انتخاب احتمال ممکن است به غیرقابل پیش بینی بودن آنها بیفزایند و روش های سازگاری ساختارهای درختانشان قبلاً نوید داده اند.

### ۳.۱.۲ هوش مصنوعی مبتنی بر سود

همانطور که توسط چندین توسعه دهنده هوش مصنوعی بازی های صنعتی اشاره شده است، عدم تعدیل رفتار در بازی ها و وظایف درون بازی برای توسعه هوش مصنوعی با کیفیت بالا ضرر دارد.

یک روش بسیار محبوب برای نوشتن رفتارهای موقت که محدودیت های مدولار بودن FSM ها و BT ها را از بین می برد، رویکرد AI مبتنی بر سود است که می تواند برای طراحی سیستم های کنترل و تصمیم گیری در بازی ها استفاده شود. به دنبال این رویکرد، به مواردی از بازی یک تابع سود خاص اختصاص داده می شود که مقداری را برای اهمیت نمونه خاص تعیین می کند. به عنوان مثال، اهمیت حضور دشمن در یک فاصله خاص یا اهمیت سلامت یک عامل در این زمینه خاص. با توجه به مجموعه تمام توابع سود موجود برای یک عامل و تمام گزینه هایی که دارد، هوش مصنوعی مبتنی بر سود تصمیم می گیرد که مهمترین گزینه ای است که در این لحظه باید بررسی کند. رویکرد مبتنی بر سود در تئوری سودمندی اقتصاد پایه ریزی شده و مبتنی بر طراحی تابع سود است. این رویکرد مشابه طراحی توابع عضویت در یک مجموعه مبهم است.

یک سود می تواند همه چیز را از داده های عینی قابل مشاهده (به عنوان مثال، سلامت دشمن) تا مفاهیم ذهنی مانند احساسات، خلق و خو و تهدید اندازه گیری کند. سودهای مختلف مربوط به اقدامات یا تصمیمات احتمالی را می توان در فرمول های خطی یا غیرخطی جمع کرد و عامل را به تصمیم گیری بر اساس سود جمع شده راهنمایی کرد. مقادیر سود را می توان در هر  $n$  فریم بازی بررسی کرد. بنابراین در حالی که FSM ها و BT ها همزمان یک تصمیم را بررسی می کنند، معماری های AI مبتنی بر سود همه گزینه های موجود را بررسی می کنند، یک سود برای آنها اختصاص می دهند و گزینه مناسب ترین (بالا ترین سود) را انتخاب می کنند. هوش مصنوعی مبتنی بر سود دارای مزایای خاصی در مقایسه با سایر تکنیک های نوشتاری موقت است. این حالت مدولار است زیرا تصمیم عامل بازی به عوامل مختلفی (یا ملاحظات) بستگی دارد. این لیست از عوامل می تواند پویا باشد. هوش مصنوعی مبتنی بر سود نیز قابل توسعه است زیرا ما به راحتی و در صورت صلاحدید انواع جدید ملاحظات را به راحتی می توانیم بنویسیم. سرانجام، این روش قابل استفاده مجدد است زیرا می توان اجزای سود را از یک تصمیم به تصمیم دیگر و از یک بازی به یک بازی دیگر منتقل کرد. در نتیجه این مزایا، هوش مصنوعی مبتنی بر سود به تدریج در صحنه صنعت بازی جذاب می شود. هوش مصنوعی مبتنی بر سود کاربرد گسترده ای را در ژانرهای بازی مشاهده کرده است و از جمله در بازی Kohan 2: Kings of War (Take Iron Man (Sega, در بازی Two Interactive and Global Star Software, 2004) Red Dead Redemption (Rockstar Games, 2010) برای کنترل رئیس، در بازی Killzone 2 (Sony Computer Entertainment, و در بازی و گفتگو، و در بازی FEAR (Sierra Entertainment 2005) برای تصمیم گیری تاکتیکی پویا.

## ۲.۲ جستجوی درختی

تا حد زیادی ادعا شده است که در واقع هوش مصنوعی اغلب، فقط جستجو است. تقریباً هر مسئله هوش مصنوعی می تواند به عنوان یک مسئله جستجو مطرح شود، که با یافتن بهترین (طبق برخی

اندازه گیری) برنامه، مسیر، مدل، عملکرد و غیره می توان آن را حل کرد. بنابراین الگوریتم های جستجو اغلب در هسته هوش مصنوعی دیده می شوند، تا جایی که بسیاری از کتابهای درسی با رفتار الگوریتم های جستجو شروع می شوند.

الگوریتم های ارائه شده در زیر می توانند به صورت الگوریتم های جستجوی درختی شناخته شوند زیرا می توان آنها را به عنوان ساخت یک درخت جستجو در جایی که ریشه گره نمایانگر وضعیت شروع جستجو است، مشاهده کرد. لبه های این درخت نمایانگر اعمالی است که عامل برای رسیدن از یک حالت به حالت دیگر انجام می دهد و گره ها نشان دهنده حالات هستند. از آنجا که به طور معمول چندین عمل مختلف وجود دارد که می تواند در یک حالت مشخص انجام شود، درخت شاخه می شود. الگوریتم های جستجوی درختی عمدتاً در اینکه کدام شاخه ها کاوش می شوند و به چه ترتیب متفاوت هستند.

## ۱.۲.۲ جستجوی ناآگاهانه

الگوریتم های جستجوی ناآگاهانه، الگوریتم هایی هستند که بدون هیچ گونه اطلاعات بیشتری در مورد هدف، فضای حالت را جستجو می کنند. الگوریتم های جستجوی ناآگاهانه عمومی معمولاً به عنوان الگوریتم های بنیادی علوم کامپیوتر دیده می شوند و حتی گاهی به عنوان AI نیز مشاهده نمی شوند.

Depth-first search یک الگوریتم جستجو است که هر شاخه را تا آنجا که ممکن است قبل از عقب نشینی و آزمایش شاخه دیگری بررسی می کند. در هر بار تکرار از حلقه اصلی خود، جستجوی عمق اول شاخه ای را انتخاب می کند و سپس به جستجوی گره حاصله در تکرار بعدی می پردازد. با رسیدن به یک گره ترمینال - گره ای که امکان پیشرفت بیشتر از آن وجود ندارد - جستجوی عمق اول لیست گره های بازدید شده را بالاتر می برد تا زمانی که یکی را پیدا کند که اقدامات کاوش نشده ای را پیدا کند. هنگامی که برای انجام یک بازی استفاده می شود، جستجوی عمق اول نتایج یک حرکت را تا زمان برنده شدن یا باختن بازی بررسی می کند و سپس به دنبال عواقب انجام یک حرکت متفاوت در نزدیکی حالت های نهایی است.

Breadth-first search برعکس جستجوی عمق اول است. به جای کاوش در تمام عواقب یک عمل، جستجوی گسترده اول قبل از کاوش در هر یک از گره های حاصل از انجام این اقدامات، همه اقدامات را از یک گره بررسی می کند. بنابراین، همه گره ها در عمق اول قبل از همه گره ها در عمق دو، سپس همه گره ها در عمق سه و غیره کاوش می شوند.

تغییرات و ترکیبات بسیاری از این الگوریتم ها وجود دارد و الگوریتم های جستجوی ناآگاه جدید در حال توسعه هستند. به ندرت مشاهده الگوریتم های جستجوی ناآگاهانه که به طور موثر در بازی ها استفاده می شوند وجود دارد، اما موارد استثنایی مانند جستجوی عرض تکراری وجود دارد که به طور شگفت انگیزی در انجام بازی های ویدیویی عمومی و استفاده از جستجوی عرض اول برای ارزیابی جنبه های نقشه های بازی استراتژی در Sentient Sketchbook [۹] بسیار



خوب عمل می کند. همچنین، مقایسه عملکرد الگوریتم های پیشرفته با الگوریتم جستجوی ساده و بدون اطلاع، غالباً روشن کننده است.

## ۲.۲.۲ جستجوی بهترین-اول

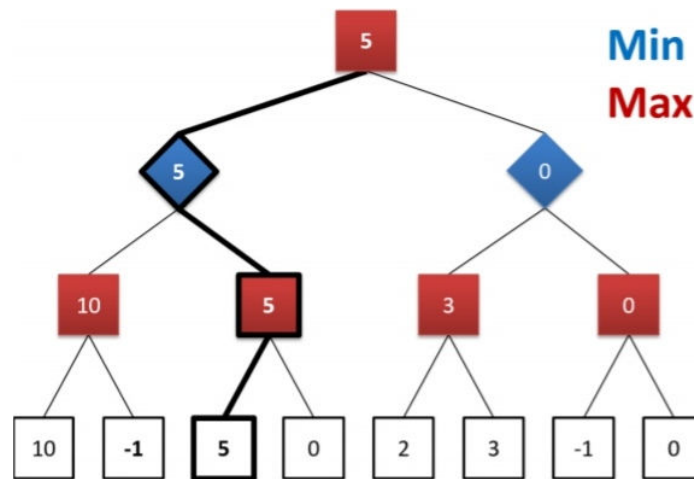
در جستجوی بهترین اول، گسترش گره ها در درخت جستجو با مقداری دانش در مورد وضعیت هدف اطلاع داده می شود. به طور کلی، گره ای که براساس برخی معیارها به حالت هدف نزدیک تر است، ابتدا گسترش می یابد. شناخته شده ترین الگوریتم جستجوی بهترین اول  $A^*$  است (تلفظ A ستاره). الگوریتم  $A^*$  لیستی از گره های "باز" را نگه می دارد، که در کنار یک گره کشف شده قرار دارند اما خود آنها کشف نشده اند. برای هر گره باز، برآوردی از فاصله آن با هدف انجام می شود. گره های جدید برای کاوش بر اساس مبنای کمترین هزینه انتخاب می شوند، جایی که هزینه فاصله از گره مبدا به علاوه برآورد فاصله تا هدف است.

$A^*$  را به راحتی می توان نوبری در فضای دو یا سه بعدی دانست. از انواع این الگوریتم معمولاً برای یافتن مسیر در بازی ها استفاده می شود. در بسیاری از بازی ها، "AI" اساساً به شخصیت های غیر بازیکن با استفاده از پیدا کردن مسیر  $A^*$  برای عبور از نقاط اسکرپت شده (نوشته شده) مربوط می شود. به منظور کنار آمدن با فضاهای بزرگ و فریبده، تغییرات زیادی در این الگوریتم اساسی ارائه شده است، از جمله نسخه های سلسله مراتبی  $A^*$ ، جستجوی اکتشافی در زمان واقعی، جستجوی نقطه پرش برای شبکه های با هزینه یکنواخت، الگوریتم های یافتن مسیر سه بعدی، الگوریتم های برنامه ریزی برای دنیای بازی های پویا [۱۰] که امکان انیمیشن جمعیت در مسیرهای بدون برخورد و رویکردهای جهت یابی مسیر در مش های نوبری را فراهم می کنند. کارهای استیو رابین و ناتان استورتوانت در مورد مسیر یابی مبتنی بر شبکه و معماری مسیر یابی نمونه های قابل توجهی هستند. مسیر یابی مبتنی بر شبکه در بازی هایی مانند Dragon Age: Origins (Electronic Arts 2009)، StarCraft (Blizzard Entertainment 1998) و WarCraft III: Reign of Chaos (Blizzard Entertainment 2002) بکار گرفته شده است.

با این حال، از  $A^*$  همچنین می توان برای جستجو در فضای حالت های بازی استفاده کرد، در حالی که به سادگی جستجو در مکان های فیزیکی نیست. به این ترتیب می توان جستجوی بهترین اول را برای برنامه ریزی به جای استفاده از نوبری، به کار برد. این تفاوت در نظر گرفتن وضعیت تغییر یافته جهان است (و نه فقط تغییر وضعیت یک عامل واحد). برنامه ریزی با  $A^*$  می تواند به طرز شگفت انگیزی موثر باشد، همانطور که توسط برنده 2009 Mario AI Competition نشان داده می شود - جایی که رقبا نمایندگانی را ارائه می دهند که با Super Mario Bros (Nintendo) بازی می کنند - بر اساس یک برنامه ریز  $A^*$  ساده که تلاش در رسیدن به انتهای سمت راست صفحه را در هر زمان آسان می کند.

## ۳.۲.۲ مینی مکس Minimax

برای بازی های تک نفره، می توان از الگوریتم های جستجوی ساده ناآگاهانه یا آگاهانه برای یافتن راهی به حالت بهینه بازی استفاده کرد. با این حال، برای بازی های خصمانه دو نفره، بازیکن دیگری وجود دارد که او نیز سعی می کند برنده باشد و عملکردهای هر بازیکن بستگی زیادی به عملکرد بازیکن دیگر دارد. برای چنین بازی هایی به جستجوی خصمانه نیاز داریم که شامل اقدامات دو بازیکن خصمانه (یا بیشتر) است. الگوریتم اصلی جستجوی خصمانه Minimax نام دارد. از این الگوریتم برای اجرای بازیهای کلاسیک دو نفره با اطلاعات کامل مانند Chess و Checkers بسیار موفقیت آمیز استفاده شده است و در واقع به طور خاص برای ساخت یک برنامه بازی شطرنج ابداع شده است.



شکل ۲.۲: یک درخت بازی انتزاعی که الگوریتم Minimax را نشان می دهد.

حلقه اصلی الگوریتم Minimax بین بازیکن ۱ و بازیکن ۲ - مانند بازیکن سفید و سیاه در شطرنج - به عنوان بازیکن min و max متناوب است. برای هر بازیکن، تمام حرکات ممکن بررسی می شود. برای هر یک از حالت های به دست آمده، تمام حرکات احتمالی بازیکن دیگر نیز مورد بررسی قرار می گیرد و به همین ترتیب ادامه می یابد تا زمانی که تمام ترکیبات احتمالی حرکات تا جایی که بازی به پایان می رسد، مورد بررسی قرار گیرند (به عنوان مثال، با برد، باخت یا تساوی) نتیجه این فرآیند تولید درخت، درخت کامل از گره ریشه تا برگها است. نتیجه بازی تابع سود را که بر روی گره های برگ اعمال می شود، اطلاع می دهد. تابع سود تخمین می زند که تنظیمات بازی فعلی چقدر برای یک بازیکن مناسب است. سپس، الگوریتم درخت جستجو را طی می کند تا تعیین کند که هر بازیکن با گرفتن مقادیر پشتیبان از برگها از طریق گره های شاخه، در

هر حالت مشخص چه اقدامی انجام می دهد. با این کار فرض بر این است که هر بازیکن سعی در بازی بهینه دارد. بنابراین، از نقطه نظر بازیکن، max سعی می شود نمره خود را به حداکثر برساند، در حالی که min سعی می کند امتیاز حداکثر را به حداقل برساند. از این رو، نام Minimax است.

به عبارت دیگر، یک گره حداکثر از درخت حداکثر مقادیر فرزند خود را محاسبه می کند در حالی که یک گره حداقل، حداقل ارزش فرزند آن را محاسبه می کند. سپس استراتژی بهینه برنده برای حداکثر بدست می آید اگر در نوبت حداقل، حداکثر برای همه حرکاتی که حداقل می تواند برنده باشد، یک برنده بدست آید. استراتژی مطلوب مربوط به حداقل زمانی است که یک برد مستقل از حداکثر حرکت ممکن باشد. برای به دست آوردن یک استراتژی برنده برای حداکثر، به عنوان مثال، ما از ریشه درخت شروع می کنیم و حرکتهایی را که منجر به گره های فرزند با بالاترین مقدار می شوند، انتخاب می کنیم (در نوبت حداقل گره های فرزند با کمترین مقدار به جای آنها انتخاب می شوند).

مطمئناً کاوش در همه حرکات و خلفهای احتمالی برای هر بازی با پیچیدگی جالب غیرقابل اجرا است، زیرا اندازه درخت جستجو با عمق بازی یا تعداد حرکاتی که شبیه سازی می شود به طور تصاعدی افزایش می یابد. به طور مشخص، tic-tac-toe دارای اندازه درخت بازی  $9! = 362$  حالت است که امکان عبور از آن وجود دارد. با این حال، درخت بازی شطرنج تقریباً  $10^{154}$  گره دارد که جستجوی آن با رایانه های مدرن غیرممکن است. بنابراین، تقریباً همه برنامه های واقعی الگوریتم Minimax جستجو را در یک عمق مشخص قطع می کنند و از یک تابع ارزیابی حالت برای ارزیابی مطلوبیت هر حالت بازی در آن عمق استفاده می کنند. به عنوان مثال، در شطرنج یک تابع ارزیابی حالت ساده این است که فقط تعداد مهره های سفید روی تخته جمع شود و تعداد قطعات سیاه کم شود. هرچه این عدد بیشتر باشد، وضعیت بازیکن سفید بهتر است. (البته، از توابع ارزیابی پیچیده بسیار پیچیده تری معمولاً استفاده می شود).

## ۴.۲.۲ درخت جستجوی مونت کارلو

بازی های زیادی وجود دارد که Minimax آنها را به خوبی بازی نخواهد کرد. به طور خاص، بازی هایی با ضریب انشعاب بالا (جایی که اقدامات بالقوه زیادی برای انجام هر زمان وجود دارد) منجر به این می شوند که Minimax فقط یک درخت بسیار کم عمق را جستجو می کنند. یکی دیگر از جنبه های بازی هایی که به طور مکرر کلیدهایی در کارهای Minimax ایجاد می کنند، ساختن یک عملکرد خوب ارزیابی وضعیت دشوار است. بازی رومیزی Go یک بازی اطلاعاتی قطعی و کامل است که نمونه خوبی از هر دو پدیده است. Go دارای یک عامل انشعاب تقریباً ۳۰۰ است، در حالی که Chess معمولاً حدود ۳۰ اقدام برای انتخاب دارد. ماهیت موقعیتی بازی Go، که همه چیز در اطراف دشمن است، تخمین صحیح ارزش یک صفحه تخته داده شده را بسیار دشوار می کند. برای مدت طولانی، بهترین برنامه های Go-playing در جهان، که بیشتر آنها مبتنی بر Minimax بودند، به سختی می توانستند از قدرت بازی یک مبتدی انسانی فراتر روند.

در سال ۲۰۰۷، جستجوی درخت مونت کارلو (MCTS) اختراع شد و قدرت بازی بهترین برنامه های Go به شدت افزایش یافت.

فراتر از اطلاعات کامل پیچیده، بازی های جبرگرا مانند Go، Chess و Checkers، بازی های اطلاعاتی ناقص مانند Battleship، Poker، Bridge و یا بازی های غیر قطعی مانند تخته نرد و انحصار به دلیل ماهیت الگوریتم از طریق Minimax قابل حل نیستند. در چنین بازی هایی، MCTS نه تنها بر محدودیت اندازه درخت Minimax غلبه می کند بلکه با توجه به محاسبه کافی، درخت Minimax بازی را تقریب می زند.

بنابراین چگونه MCTS از عوامل انشعاب بالا، عدم عملکرد مناسب ارزیابی وضعیت و عدم داشتن اطلاعات کامل و جبرگرایی می پذیرد؟ برای شروع، همه شاخه های درخت جستجو را تا عمق یکسان جستجو نمی کند، در عوض روی شاخه های امیدوار کننده متمرکز می شود. این امر امکان جستجو در شاخه های خاص را تا عمق قابل توجهی فراهم می کند حتی اگر فاکتور شاخه سازی زیاد باشد. علاوه بر این، برای دستیابی به کمبود توابع ارزیابی خوب، جبرگرایی و اطلاعات ناقص، فرمول استاندارد MCTS برای تخمین کیفیت وضعیت بازی از rollouts استفاده می کند، به طور تصادفی از حالت بازی تا پایان بازی برای دیدن نتیجه برد مورد انتظار (یا از دست دادن) بازی می کند. مقادیر سودمندی به دست آمده از طریق شبیه سازی های تصادفی می تواند به طور کارآمد برای تنظیم خط مشی به سمت استراتژی بهترین اول (تقریب درخت Minimax) استفاده شود.

در شروع اجرای الگوریتم MCTS، درخت از یک گره تشکیل شده است که نمایانگر وضعیت فعلی بازی است. سپس الگوریتم با اضافه کردن و ارزیابی گره های جدید نشان دهنده حالات بازی، به صورت تکراری درخت جستجو می سازد. این فرایند می تواند در هر زمان قطع شود و MCTS را هر زمان و الگوریتمی ارائه دهد. MCTS برای کار فقط به دو قسمت اطلاعات احتیاج دارد: قوانین بازی که به نوبه خود حرکت های موجود در بازی و ارزیابی وضعیت ترمینال را به دست می دهد - این یک برد، یک باخت، یک تساوی یا یک بازی است. حلقه اصلی الگوریتم MCTS را می توان به چهار مرحله تقسیم کرد: انتخاب، گسترش (دو مرحله اول نیز به عنوان سیاست درختی شناخته می شوند)، شبیه سازی و تولید مجدد.

## ۳.۲ محاسبات تکاملی

در حالی که الگوریتم های جستجوی درخت از گره ریشه که نمایانگر یک حالت مبدا است، شروع می شوند و یک درخت جستجو را بر اساس اقدامات موجود می سازند، الگوریتم های بهینه سازی یک درخت جستجو نمی سازند. آنها فقط راه حل های کامل را در نظر می گیرند و نه مسیری که برای رسیدن به آنجا طی شده اند. همه الگوریتم های بهینه سازی فرض می کنند که چیزی برای بهینه سازی راه حل ها وجود دارد. باید یک هدف، جایگزین تابع سود، تابع ارزیابی یا تابع تناسب اندام وجود داشته باشد، که می تواند یک مقدار عددی (تناسب) را به یک راه حل اختصاص دهد،

که می تواند به حداکثر برسد (یا به حداقل برسد). با توجه به یک تابع سود، یک الگوریتم بهینه سازی را می توان به عنوان یک الگوریتم در نظر گرفت که در فضای جستجو راه حل هایی را که بالاترین (یا کمترین) ارزش آن سود را دارند جستجو می کند. یک خانواده گسترده از الگوریتم های بهینه سازی مبتنی بر تغییرات تصادفی راه حل ها است، جایی که یک یا چند راه حل در هر زمان نگهداری می شود و راه حل های جدید (یا نامزدها یا نقاط جستجو؛ اصطلاحات مختلفی توسط نویسندگان مختلف استفاده می شود) از طریق تغییر تصادفی برخی از راه حل های موجود، یا شاید ترکیبی از برخی از آنها. الگوریتم های بهینه سازی تصادفی که چندین راه حل را حفظ می کنند، با قیاس با تکامل طبیعی، الگوریتم های تکاملی نامیده می شوند.

مفهوم مهم دیگر هنگام صحبت در مورد الگوریتم های بهینه سازی، نمایش آنها است. همه راه حل ها به نوعی نشان داده می شوند، به عنوان مثال بردارهایی با اندازه ثابت اعداد واقعی یا رشته های طول متغیر کاراکترها. به طور کلی، یک مصنوع یکسان را می توان به طرق مختلف نشان داد. به عنوان مثال، هنگام جستجوی دنباله ای از اقدامات که یک پیچ و خم را حل می کند، توالی عملکرد را می توان به چندین روش مختلف نشان داد. در مستقیم ترین نمایش، شخصیت در مرحله  $t$  تعیین می کند که در مرحله  $t + 1$  چه اقدامی انجام شود. نمایش تا حدودی غیرمستقیم برای دنباله ای از اقدامات، دنباله ای از چندتایی ها خواهد بود، جایی که شخصیت در مرحله  $t$  تصمیم می گیرد چه اقدامی انجام دهد و تعداد  $t + n$  تعیین می کند که چند مرحله از زمان  $n$  برای انجام آن عمل باشد.

انتخاب عامل تأثیر زیادی بر کارایی و کارایی الگوریتم جستجو دارد و هنگام انجام این انتخابها چندین مصالحه (سبک و سنگین کردن) وجود دارد. بهینه سازی یک مفهوم بسیار کلی است و الگوریتم های بهینه سازی برای طیف وسیعی از وظایف در AI و همچنین در محاسبات به طور کلی مفید است. در هوش مصنوعی و بازی ها، الگوریتم های بهینه سازی مانند الگوریتم های تکاملی نیز در بسیاری از نقش ها استفاده شده است.

## ۱.۳.۲ جستجوی محلی

ساده ترین الگوریتم های بهینه سازی، الگوریتم های بهینه سازی محلی هستند. اینها به این دلیل نامیده می شوند که آنها فقط "محلی"، در بخش کوچکی از فضای جستجو، در هر زمان معین جستجو می کنند. یک الگوریتم بهینه سازی محلی به طور کلی فقط یک نامزد راه حل را در هر زمان معین نگه می دارد و تغییرات آن راه حل را بررسی می کند.

## ۲.۳.۲ الگوریتم های تکاملی

الگوریتم های تکاملی الگوریتم های بهینه سازی جهانی تصادفی هستند. آنها جهانی و نه محلی نامیده می شوند زیرا در بسیاری از نقاط همزمان به جستجوی نقاط می پردازند و این نقاط می توانند از یکدیگر دور باشند. آنها این کار را با حفظ حافظه جمعیتی از راه حلها در هر زمان انجام می دهند. ایده کلی محاسبات تکاملی بهینه سازی با راه حل های ”پرورش” است: تولید بسیاری از راه حل ها، دور ریختن راه حل های بد و نگه داشتن راه حل های خوب (یا حداقل کمتر بد) و ایجاد راه حل های جدید از راه حل های خوب.

ایده نگهداری جمعیت با انتخاب طبیعی از تکامل داروینی گرفته شده که الگوریتم های تکاملی نیز از آن نام می برند. اندازه جمعیت یکی از پارامترهای اصلی الگوریتم تکاملی است.

ایده دیگری که از تکامل در طبیعت گرفته شده همگذری (crossover) است که به آن ترکیب مجدد نیز گفته می شود. این معادل تولید مثل جنسی در دنیای طبیعی است. دو یا چند راه حل (والدین نامیده می شوند) با ترکیب عناصر خود فرزندی تولید می کنند. ایده این است که اگر ما دو راه حل خوب بگیریم، یک راه حل ترکیبی از این دو راه حل - یا بین آنها - باید خوب باشد، شاید حتی بهتر از والدین. اپراتور فرزندان به نمایش راه حل بسیار وابسته است. هنگامی که راه حل به صورت یک رشته یا بردار نمایش داده می شود، عملگرهایی مانند همگذری یکنواخت (که یک سکه منصفانه را تلنگر می زند و به طور تصادفی از هر والد برای هر موقعیت در فرزندان مقادیر را انتخاب می کند) یا همگذری یک نقطه ای (جایی که یک موقعیت  $p$  در فرزندان است به طور تصادفی انتخاب می شود و می توان مقادیر موقعیت ها قبل از  $p$  را از والد ۱ و مقادیر موقعیت ها را پس از  $p$  از والد ۲ گرفت.) می تواند استفاده گردد.

الگوی اصلی الگوریتم تکاملی به شرح زیر است:

۱. شروع: جمعیت با  $N$  راه حلهای ایجاد شده و به طور معمول، به عنوان مثال، نقاط تصادفی در فضای جستجو پر شده است. راه حل های بسیار مناسب شناخته شده نیز می توانند به این جمعیت اولیه اضافه شوند.

۲. ارزیابی: تابع تناسب اندام برای ارزیابی همه راه حل های موجود در جمعیت و تعیین مقادیر تناسب برای آنها استفاده می شود.

۳. انتخاب والدین: بر اساس تناسب اندام و احتمالاً معیارهای دیگر، آن دسته از اعضای جمعیتی که برای تولید مثل استفاده می شوند انتخاب می شوند. استراتژی های انتخاب شامل روش هایی است که مستقیم یا غیر مستقیم به تناسب راه حل ها بستگی دارد، از جمله چرخ رولت (متناسب با تناسب اندام)، رتبه بندی (متناسب با رتبه در جمعیت) و مسابقات.

۴. تولید مثل: فرزندان از طریق همگذری از والدین یا از طریق کپی کردن راه حل های والدین یا ترکیبی از آنها تولید می شوند.

۵. تنوع: جهش در بعضی یا همه والدین و یا فرزندان اعمال می شود.

۶. جایگزینی: در این مرحله ما انتخاب می کنیم که کدام یک از والدین و یا فرزندان به نسل بعدی برسد. استراتژی های جایگزین محبوب جمعیت فعلی شامل روش های نسلی (والدین می میرند؛ فرزندان جایگزین آنها می شوند)، حالت پایدار (فرزندان جایگزین بدترین والدین می شوند، و تنها در صورت بهتر بودن فرزندان) و روش های نخبه گرایی.

۷. خاتمه: آیا ما هنوز کارمان تمام شده است؟ بر اساس تعداد نسل ها یا ارزیابی های گذشته (خستگی)، بالاترین تناسب با هر راه حل (موفقیت) و یا شرایط دیگر خاتمه، تصمیم بگیرید.

۸. به مرحله ۲ بروید.

هر تکرار حلقه اصلی (یعنی هر بار که به مرحله ۲ می رسیم) یک نسل نامیده می شود که مطابق اصطلاحات الهام گرفته از طبیعت است. مجموع ارزیابی های تناسب اندام انجام شده معمولاً متناسب با اندازه جمعیت بر تعداد نسل ها است. این الگوی سطح بالا را می توان به طرق بیشماری متفاوت پیاده سازی و گسترش داد. هزاران الگوریتم تکاملی یا شبیه تکامل وجود دارد، و بسیاری از آنها جریان کلی را تغییر می دهند، مراحل جدید را اضافه می کنند و مراحل موجود را حذف می کنند.

یکی دیگر از خانواده های برجسته الگوریتم های تکاملی، الگوریتم های ژنتیک (GAs) است. اینها با اتکا به همگذری و نه جهش برای تغییرات مشخص می شوند (برخی از الگوریتم های ژنتیک هیچ جهشی ندارند)، انتخاب متناسب با تناسب اندام و راه حل ها اغلب به عنوان رشته بیت یا سایر رشته های جداگانه نشان داده می شوند. با این حال، باید توجه داشت که تمایز بین انواع مختلف الگوریتم های تکاملی عمدتاً بر اساس ریشه های تاریخی آنها است. این روزها، تغییرات و ترکیب گسترده به قدری زیاد است که دسته بندی یک الگوریتم خاص به عنوان متعلق به یک یا خانواده دیگر، اغلب منطقی نیست.

## ۴.۲ یادگیری تحت نظارت

یادگیری تحت نظارت فرآیند الگوریتمی تقریب عملکرد زیربنایی بین داده های برچسب دار (مشخص) و صفات یا ویژگی های مربوطه است. یک مثال رایج از یادگیری تحت نظارت دستگاهی است که از آن خواسته می شود بین سیب و گلابی (داده های دارای برچسب) با توجه به مجموعه ای از صفات یا ویژگی های داده مانند رنگ و اندازه میوه ها، تمایز قائل شود. در ابتدا، دستگاه با طبقه بندی بین سیب و گلابی با دیدن تعدادی از نمونه های میوه موجود - که شامل رنگ و اندازه هر میوه از یک سو و برچسب متناظر آنها (سیب یا گلابی) از طرف دیگر، یاد می گیرد. پس

از اتمام یادگیری، دستگاه به طور ایده آل باید بتواند تشخیص دهد که آیا یک میوه جدید و دیده نشده گلابی است یا سیب فقط بر اساس رنگ و اندازه آن. امروزه فراتر از تمایز بین یادگیری تحت نظارت سیب و گلابی در بسیاری از برنامه ها از جمله خدمات مالی، تشخیص پزشکی، تشخیص تقلب، دسته بندی صفحات وب، تشخیص تصویر و گفتار و مدل سازی کاربر (در میان بسیاری) استفاده می شود.

بدیهی است که یادگیری تحت نظارت مستلزم مجموعه ای از نمونه های آموزشی دارای برچسب است. از این رو تحت نظارت به طور خاص، سیگنال آموزشی به عنوان مجموعه ای از برچسب های تحت نظارت بر داده ها (به عنوان مثال، این یک سیب است در حالی که آن یک گلابی است) که بر اساس مجموعه ای از مشخصه های این برچسب ها عمل می کند (به عنوان مثال، این سیب دارای رنگ قرمز و اندازه متوسط است). در نتیجه، هر مثال داده به صورت یک جفت از مجموعه ای از برچسب ها (یا خروجی ها) و ویژگی های مربوط به این برچسب ها (یا ورودی ها) ارائه می شود. هدف نهایی یادگیری تحت نظارت فقط یادگیری از جفت های ورودی-خروجی نیست، بلکه بدست آوردن یک تابع است که رابطه آنها را نزدیک (بهرتر، تقلید) می کند. تابع مشتق شده باید بتواند به خوبی نمونه های جدید و دیده نشده از جفت های ورودی و خروجی را ترسیم کند (به عنوان مثال، سیب ها و گلابی های دیده نشده در مثال ما)، این ویژگی است که تعمیم نامیده می شود.

به طور قاعده مند، یادگیری تحت نظارت سعی می کند تا تابع  $f: X \rightarrow Y$  را با توجه به مجموعه ای از  $N$  مثال آموزشی  $\{(x_1, y_1), \dots, (x_N, y_N)\}$ ؛ جایی که  $X$  و  $Y$  به ترتیب فضای ورودی و خروجی است؛  $x_i$  بردار ویژگی (ورودی) مثال  $i$  ام و  $y_i$  مجموعه ای از برچسب های مربوط به آن است. یک کار یادگیری تحت نظارت دارای دو مرحله اصلی است. در اولین مرحله آموزش، نمونه های آموزشی - ویژگی ها و برچسب های مربوطه - ارائه می شوند و عملکرد  $f$  بین ویژگی ها و برچسب ها استخراج می شود. همانطور که در لیست الگوریتم های زیر خواهیم دید  $f$  را می توان به عنوان تعدادی از قوانین طبقه بندی، درخت تصمیم یا فرمول های ریاضی نشان داد. در مرحله دوم مرحله  $f$  می تواند برای پیش بینی برچسب داده های ناشناخته با توجه به ویژگی های آنها استفاده شود. برای تأیید تعمیم پذیری  $f$  و جلوگیری از تطبیق بیش از حد با داده ها [۱۱]، معمول این است که  $f$  با استفاده از یک معیار عملکرد مانند دقت، که درصد نمونه های آزمایشی است، روی یک مجموعه داده مستقل (آزمون) جدید ارزیابی می شود. به درستی توسط عملکرد آموزش دیده ما پیش بینی شده است. اگر دقت قابل قبول باشد، می توانیم از  $f$  برای پیش بینی نمونه های داده جدید استفاده کنیم.

اما چگونه این تابع  $f$  را استخراج می کنیم؟ به طور کلی، یک فرآیند الگوریتمی پارامترهای این تابع را اصلاح می کند تا ما مطابقت خوبی بین برچسب های داده شده از نمونه های آموزشی خود و عملکردی که برای تقریبی آن تلاش می کنیم، بدست آوریم. روشهای بی شماری برای یافتن و نمایش آن تابع وجود دارد که هرکدام مربوط به الگوریتم یادگیری تحت نظارت متفاوت است. اینها شامل شبکه های عصبی مصنوعی، استدلال مبتنی بر مورد، یادگیری درخت تصمیم، جنگل های



تصادفی، رگرسیون گاوسی، طبقه بندی های ساده لوح بیز، نزدیکترین همسایگان  $k$  و ماشین های بردار پشتیبانی می شود [۱۱]. انواع الگوریتم های یادگیری تحت نظارت، تا حدی با این واقعیت توضیح داده می شود که هیچ الگوریتم یادگیری واحدی وجود ندارد که در مورد تمام مشکلات یادگیری تحت نظارت بهترین عملکرد را داشته باشد. این به طور گسترده ای به عنوان قضیه ناهار رایگان شناخته می شود.

سه الگوریتم یادگیری تحت نظارت عبارتند از: شبکه های عصبی مصنوعی، ماشین های بردار پشتیبان و یادگیری درخت تصمیم. هر سه الگوریتم یادگیری تحت نظارت را می توان برای کارهای طبقه بندی، پیش بینی یا یادگیری ترجیحی استفاده کرد.

## ۵.۲ یادگیری تقویتی

یادگیری تقویتی (RL) یک رویکرد یادگیری ماشینی است که از روانشناسی رفتارگرایی و به ویژه روش یادگیری تصمیم گیری انسان و حیوانات از طریق پاداش های (مثبت یا منفی) دریافت شده توسط محیط، الهام گرفته شده است. در یادگیری تقویت کننده، نمونه های رفتار خوب معمولاً در دسترس نیستند (مانند یادگیری تحت نظارت). در عوض، مشابه یادگیری تکاملی (تقویت کننده)، سیگنال آموزش الگوریتم بر اساس نحوه تعامل یک عامل با آن توسط محیط ارائه می شود. در یک نقطه خاص از زمان  $t$  عامل در یک حالت خاص قرار دارد و تصمیم می گیرد از همه اقدامات موجود در حالت فعلی خود یک اقدام  $a$  را انجام دهد. به عنوان یک پاسخ، محیط زیست پاداش فوری  $r$  را می دهد. از طریق تعامل مداوم بین عامل و محیط آن، عامل به تدریج می آموزد که اقداماتی را انتخاب کند که مجموع پاداش آن را به حداکثر برساند. RL از منظرهای انضباطی مختلفی از جمله تحقیقات عملیاتی، نظریه بازی، تئوری اطلاعات و الگوریتم های ژنتیک مورد مطالعه قرار گرفته و با موفقیت در مسائلی که شامل تعادل بین پاداش های بلند مدت و کوتاه مدت مانند کنترل ربات و بازی ها است، مورد استفاده قرار گرفته است.

### ۱.۵.۲ یادگیری Q (Q-Learning)

Q-learning یک الگوریتم یادگیری TD بدون مدل، خارج از خط مشی است که متکی بر نمایشی جدولی از مقادیر  $Q(s, a)$  است (از این رو نام آن است). به طور غیررسمی،  $Q(s, a)$  نشان می دهد که انتخاب عمل  $a$  در حالت  $s$  چقدر خوب است. به طور رسمی،  $Q(s, a)$  تقویت پیش بینی شده تخفیف برای اقدام  $a$  در حالت  $s$  است. عامل یادگیری  $Q$  با انتخاب اقدامات و دریافت پاداش از طریق بوت استرپ از تجربه می آموزد. هدف عامل یادگیری  $Q$  این است که با انتخاب عملکرد صحیح در هر حالت، پاداش مورد انتظار خود را به حداکثر برساند. پاداش، به طور خاص، یک جمع وزنی از مقادیر مورد انتظار برای پاداش های تخفیف خورده در آینده است. الگوریتم

یادگیری  $Q$  یک بروزرسانی ساده در مورد مقادیر  $Q$  به صورت تکراری است. در ابتدا، جدول  $Q$  دارای مقادیر دلخواهی است که توسط طراح تنظیم شده است. سپس هر بار که عامل اقدام  $a$  را از حالت  $s$  انتخاب می کند، از حالت  $s'$  بازدید می کند، پاداش فوری  $r$  دریافت می کند و مقدار  $Q(s, a)$  خود را به روز می کند.

یادگیری  $Q$  دارای تعدادی محدودیت است که عمدتاً مربوط به نمایش جدولی آن است. اول از همه، بسته به نمایش حالت-عمل انتخاب شده، اندازه فضای حالت-عمل ممکن است از نظر محاسباتی بسیار گران باشد. همانطور که اندازه جدول  $Q$  نیازهای محاسباتی ما برای تخصیص حافظه و بازیابی اطلاعات افزایش می یابد. بعلاوه، ما ممکن است همگرایی بسیار طولانی را تجربه کنیم زیرا زمان یادگیری به اندازه فضای حالت-عمل نمایی است. برای غلبه بر این موانع و دریافت عملکرد مناسب از زبان آموزان، RL، ما باید راهی برای کاهش فضای حالت-عمل طراحی کنیم.

## ۶.۲ یادگیری بدون نظارت

همانطور که قبلاً بیان شد، نوع سود (یا سیگنال آموزش) کلاس الگوریتم AI را تعیین می کند. در یادگیری تحت نظارت، سیگنال آموزش به عنوان برجسب داده (خروجی هدف) و در یادگیری تقویت به عنوان پاداش از محیط حاصل می شود. در عوض، یادگیری بدون نظارت تلاش می کند با جستجوی الگوهای موجود در تمام ویژگی های داده ورودی و بدون دسترسی به یک خروجی هدف، ارتباطات ورودی را کشف کند. با یادگیری بدون نظارت، به جای تلاش برای تقلید یا پیش بینی مقادیر هدف، بر ساختار ذاتی و ارتباطات داده ها تمرکز می کنیم. ما دو کار یادگیری بدون نظارت را با الگوریتم های مربوطه بررسی می کنیم: خوشه بندی و استخراج الگوی مکرر.

### ۱.۶.۲ خوشه بندی

خوشه بندی وظیفه یادگیری بدون نظارت برای یافتن گروه های ناشناخته از تعدادی از نقاط داده است به طوری که داده های درون یک گروه (یا در غیر این صورت خوشه) شبیه به یکدیگر و غیر از داده های خوشه های دیگر است. خوشه بندی برنامه هایی را در شناسایی گروه های داده در چندین ویژگی و وظایف کاهش داده مانند فشردن داده ها، هموار سازی اختلالات، تشخیص پرت و پارتیشن مجموعه داده پیدا کرده است. خوشه بندی برای بازیها با کاربردها در مدل سازی بازیکن، بازی کردن و تولید محتوا از اهمیت اساسی برخوردار است.

همانند طبقه بندی، خوشه بندی داده ها را در کلاس ها قرار می دهد. اما برجسب های کلاس ها ناشناخته است و الگوریتم های خوشه بندی کشف آنها را با ارزیابی تکراری کیفیت آنها انجام می دهند. از آنجا که خوشه های صحیح ناشناخته هستند، شباهت (و عدم تشابه) فقط به ویژگی های داده مورد استفاده بستگی دارد. خوشه های خوب با دو ویژگی اصلی مشخص می شوند: (۱) تشابه درون خوشه ای بالا، یا در غیر این صورت، فشردگی زیاد و (۲) تشابه بین خوشه ای کم، یا غیر از

این، تفکیک خوب. معیار محبوبیت فشردگی، میانگین فاصله بین هر نمونه در خوشه و نزدیکترین نقطه نمایش است - به عنوان مثال، نقطه مرکزی - همانطور که در الگوریتم k-means استفاده می شود. نمونه هایی از معیارهای جداسازی شامل پیوند منفرد و پیوند کامل است: نمونه اول کمترین فاصله بین هر نمونه در یک خوشه و هر نمونه در خوشه دیگر است. دومی بیشترین فاصله بین هر نمونه در یک خوشه و هر نمونه در خوشه دیگر است. در حالی که فشردگی و تفکیک از معیارهای عینی اعتبار خوشه هستند، توجه به این نکته مهم است که اینها شاخص های معنی داری خوشه نیستند.

فراتر از معیارهای اعتبار توصیف شده در بالا، الگوریتم های خوشه بندی توسط یک تابع عضویت و یک روش جستجو تعریف می شوند. تابع عضویت ساختار خوشه ها را در رابطه با نمونه های داده تعریف می کند. رویه جستجو استراتژی ای است که ما برای خوشه بندی داده های خود با توجه به عملکرد عضویت و معیار اعتبار دنبال می کنیم. نمونه هایی از این استراتژی ها شامل تقسیم همه نقاط داده به یک باره به خوشه ها مانند k-means یا ادغام مجدد خوشه ها (یا تقسیم) خوشه ها (مانند خوشه بندی سلسله مراتبی) است. خوشه بندی از طریق انبوهی از الگوریتم ها از جمله خوشه بندی سلسله مراتبی، k-means، k-medoids، DBSCAN و نقشه های خودسازمانده قابل تحقق است. الگوریتم ها در نحوه تعریف خوشه و شکل گیری آن متفاوت از یکدیگر هستند. انتخاب یک الگوریتم خوشه بندی مناسب و پارامترهای مربوط به آن، از جمله اینکه کدام تابع فاصله را باید استفاده کنید یا تعداد خوشه هایی را که انتظار می رود، بستگی به اهداف مطالعه و داده های موجود دارد.

## ۲.۶.۲ استخراج الگوی مکرر

الگوی استخراج مکرر مجموعه تکنیک هایی است که سعی در استخراج الگوها و ساختارهای مکرر در داده ها دارد. الگوها شامل توالی ها و مجموعه های مختلف هستند. دو نوع استخراج الگوی مکرر وجود دارد که مورد توجه ویژه هوش مصنوعی بازی است: استخراج اقلام مکرر و استخراج دنباله مکرر. هدف اولی یافتن ساختار در میان ویژگی های داده است که ترتیب داخلی خاصی ندارند در حالی که هدف دوم یافتن ساختار در میان ویژگی های داده بر اساس یک نظم زمانی ذاتی است. در حالی که با الگوی یادگیری بدون نظارت همراه است، استخراج الگوی مکرر هم در هدف و هم در روش های الگوریتمی که دنبال می کند متفاوت است. روش های استخراج الگوی مکرر متداول محبوب و مقیاس پذیر شامل الگوریتم Apriori [6] برای استخراج مجموعه ها، SPADE و GSP، برای استخراج توالی است.

## ۷.۲ الگوریتم های ترکیبی قابل توجه

روش های هوش مصنوعی می توانند به روش های مختلفی در هم آمیخته شوند تا الگوریتم های پیچیده جدیدی را جمع کنند که نقاط قوت قطعات ترکیبی آنها را جمع می کند. به عنوان مثال می توانید به GA اجازه دهید درختان رفتاری یا FSM شما را تکامل دهند. در عوض می توانید MCTS را با برآوردهای ANN برای هرس درخت توانمند کنید. یا می توانید در هر الگوریتم جستجو که قبلاً پوشش داده شد، یک جزء از جستجوی محلی اضافه کنید. ما ترکیبات حاصل از روشهای AI را به عنوان الگوریتم های ترکیبی نامگذاری می کنیم و در این بخش دو مورد از تأثیرگذارترین الگوریتم های هوش مصنوعی بازی ترکیبی را پوشش می دهیم: تکامل عصبی و یادگیری تفاوت زمانی با تقریب های عملکرد ANN.

### ۱.۷.۲ تکامل عصبی

تکامل شبکه های عصبی مصنوعی، یا دیگری تکامل عصبی، به طراحی شبکه های عصبی مصنوعی - وزن اتصال آنها، توپولوژی یا هر دو - با استفاده از الگوریتم های تکاملی اشاره دارد. تکامل شبکه های عصبی با موفقیت در حوزه های زندگی مصنوعی، کنترل ربات ها، سیستم های مولد و بازی های رایانه ای اعمال شده است. کاربرد گسترده الگوریتم در درجه اول به دو دلیل اصلی است. اولاً، بسیاری از مشکلات هوش مصنوعی را می توان به عنوان مشکلات بهینه سازی عملکرد مشاهده کرد که عملکرد عمومی اساسی آنها را می توان از طریق ANN تقریب زد. دوم، تکامل عصبی روشی است که در استعاره های بیولوژیکی و نظریه تکامل بنا شده و از روش تکامل مغز الهام گرفته شده است.

این رویکرد یادگیری تکاملی (تقویت کننده) یا هنگامی اعمال می شود که تابع خطای موجود قابل تغییر نباشد یا هنگامی که خروجی های هدف در دسترس نیستند. اولین مورد ممکن است رخ دهد، به عنوان مثال، هنگامی که توابع فعال سازی مورد استفاده در ANN پیوسته نیستند و بنابراین قابل تمایز نیستند. مورد دوم ممکن است در دامنه ای رخ دهد که برای آن نمونه ای از رفتار خوب (یا بد) نداشته باشیم یا تعریف عینی اینکه یک رفتار خوب چگونه است غیرممکن است. در مقابل یادگیری تحت نظارت، تکامل عصبی برای آموزش شبکه های عصبی نیازی به مجموعه داده ای از جفت های ورودی و خروجی ندارد. بلکه فقط به اندازه گیری عملکرد ANN در مورد مسئله تحت بررسی نیاز دارد، به عنوان مثال، امتیاز یک عامل بازی که توسط ANN کنترل می شود. تکامل عصبی کاربردهای گسترده ای را در حوزه بازی ها در نقش هایی مانند ارزیابی فضای عمل-بازی، انتخاب یک اقدام مناسب، انتخاب از بین استراتژی های ممکن، مدل سازی استراتژی های حریف، تولید محتوا و مدل سازی تجربه بازیکن پیدا کرده است. کارآیی، مقیاس پذیری، کاربرد وسیع و یادگیری بی پایان الگوریتم، برخی از دلایلی است که تکامل عصبی را به عنوان یک روش کلی خوب برای بسیاری از وظایف هوش مصنوعی بازی در می آورد.

## ۲.۷.۲ یادگیری TD با برآوردگرهای تابع شبکه ANN

یادگیری تقویت به طور معمول از نمایش های جدولی برای ذخیره دانش استفاده می کند. همانطور که قبلاً در بخش RL ذکر شد، نمایش دانش از این طریق ممکن است منابع محاسباتی موجود ما را تخلیه کند زیرا اندازه جدول جستجو با توجه به فضای حالت عمل به طور تصاعدی افزایش می یابد. مشهورترین روش برای پرداختن به این چالش استفاده از ANN به عنوان برآوردگر (تقریب زنده) مقدار یا مقدار  $Q$  است و بدین ترتیب جایگزین جدول می شود. با این کار استفاده از الگوریتم در فضاهای بزرگتر از نمایش های حالت عمل امکان پذیر است. بعلاوه، بعنوان مثال یک ANN به عنوان یک تقریب زنده تابع  $Q$  می تواند مسائل مربوط به فضاهای حالت پیوسته را که بی نهایت بزرگ هستند برطرف کند.

دو نمونه مهم از الگوریتم هایی که از ظرفیت تقریبی جهانی ANN برای یادگیری تفاوت زمانی استفاده می کنند، الگوریتم های TD-Gammon و  $deep\ Q$  به ترتیب برای تسلط بر بازی تخته نرد و انجام بازی های سرگرمی Atari 2600 در سطح فوق انسانی هستند. هر دو الگوریتم برای هر وظیفه RL فراتر از این بازی های خاص قابل اجرا هستند.

## فصل ۳

# آینده هوش مصنوعی بازی

تکنیک های پیشرفته تر هوش مصنوعی به آرامی در صنعت بازی راه پیدا می کنند و احتمالاً طی سالهای آینده افزایش خواهد یافت. علاوه بر این، شرکت ها بیشتر و بیشتر با موسسات تحقیقاتی همکاری می کنند تا آخرین نوآوری ها را در صنعت ارائه دهند. به عنوان مثال، شرکت Massive Entertainment و دانشگاه مالت برای پیش بینی انگیزه بازیکنان در بازی محبوب Tom Clan-cys The Division همکاری کردند. شرکت های دیگر مانند King ، سرمایه گذاری زیادی در روش های یادگیری عمیق انجام می دهند تا به طور خودکار مدل هایی از بازیکنان را یاد بگیرند که می توانند به سرعت برای آزمایش سطح جدید مورد استفاده قرار گیرند.

تولید محتوای رویه ای در حال حاضر برای بسیاری از بازیهای اصلی مانند Spelunky (Moss-Hello Games, 2016) No Man's Sky و mouth, LLC, 2013) استفاده شده است و احتمالاً در آینده شاهد انواع کاملاً جدیدی از بازی خواهیم بود که تحقق آنها بدون تکنیک های پیچیده هوش مصنوعی غیرممکن خواهد بود. بازی اخیر AI Dungeon 2 نشان می دهد که این بازیها چه مسیری را باید دنبال کنند در این بازی ماجراجویی متنی بازیکنان می توانند با مدل زبان GPT-2 Open AI ، که روی ۴۰ گیگابایت از متن برداشته شده از اینترنت آموزش دیده، تعامل داشته باشند. این بازی تقریباً به هر چیزی که بازیکن تایپ می کند به روشی معقول پاسخ می دهد، اگرچه داستان های ایجاد شده نیز اغلب پس از مدتی انسجام خود را از دست می دهند. این مشاهدات به یک چالش مهم اشاره دارد: برای اینکه تکنیک های پیشرفته هوش مصنوعی بیشتر در صنعت بازی به کار گرفته شوند، روشهایی لازم است که توسط طراحان قابل کنترل تر و قابل تفسیر بیشتری هستند.

پیش بینی می شود که در آینده نزدیک، تکنیک های مدل سازی مولد از یادگیری ماشین، مانند شبکه های تولیدی و متخاصم (GAN) ، به کاربران امکان می دهد نمادهای خود را در سطح بی سابقه شخصی کنند یا اجازه ایجاد انواع نامحدود بافت های واقع گرایانه و دارایی ها در بازی ها را بدهند. این ایده تولید محتوای رویه ای از طریق یادگیری ماشینی (PCGML) ، یک منطقه تحقیقاتی جدید است که در حال حاضر منجر به نتایج امیدوار کننده در تولید سطوح برای بازی

هایی مانند Doom یا Super Mario شده است [۲].  
انتظار می رود که تحقیقات آینده (در کنار بازی بهتر در بازی های بیشتر) در بازی AI روی این زمینه ها متمرکز شود:

- همکاری هوش مصنوعی / انسان و همکاری عامل AI / AI از اهمیت بیشتری برخوردار می شوند، این ممکن است تحت عنوان هوش مصنوعی تیمی بکار گرفته شود.
- پردازش بیشتر زبان طبیعی ارتباط بهتر و در برخی موارد ارتباط مستقیم با شخصیت های بازی را امکان پذیر می کند. سیستم های کمکی صوتی تجاری موجود به عنوان Google Assistant یا Alexa نشان می دهد که این امکان وجود دارد.
- نکات قبلی و پیشرفت در مدل سازی بازیکنان و تجزیه و تحلیل بازی ها منجر به هوش مصنوعی رفتاری شبیه تر به انسان می شود، این امر به نوبه خود تست بازی بهتری را فراهم می کند که می تواند تا حدی خودکار باشد.

- (Procedural Content Generation) PCG بیشتر در صنعت بازی و سایر برنامه ها اعمال خواهد شد. به عنوان مثال، در نسخه شبیه ساز پرواز جدید مایکروسافت که اکنون در حالت آزمون آلفا است، به شدت مورد استفاده قرار می گیرد. این امر همچنین باعث تحقیقات بیشتر در این زمینه خواهد شد.

با این وجود، مانند سایر زمینه های هوش مصنوعی، هوش مصنوعی بازی نیز مجبور است با برخی از مسائل که بیشتر از دو پیشرفت جدیدتر ناشی می شود کنار بیاید: روش های یادگیری عمیق با تئوری سبک اما بسیار موفق و محاسبات کاملاً موازی. اولین مورد حاکی از آن است که ما کنترل بسیار کمی بر عملکرد روشهای یادگیری عمیق داریم، پیش بینی اینکه چه چیزی با کدام پارامترها خوب کار می کند دشوار است، و آزمایش دوم بدان معنی است که به دلیل محدودیت های سخت افزاری، بسیاری از آزمایش ها به سختی می توانند تکرار شوند. به عنوان مثال، Open AI Five به مدت طولانی روی GPU 256 و CPU ۱۲۸،۰۰۰ آموزش دیده است. به طور کلی، بخش بزرگی از هوش مصنوعی با یادگیری عمیق در حال حاضر با بحران تکرارپذیری روبرو است. برخی از اینها با استفاده از روش تجربی و آمار بهتر بهبود می یابند، زیرا چندی پیش در محاسبات تکامل نیز به خوبی کار شده است. اولین تلاش ها در بازی هوش مصنوعی همچنین سعی دارد با تعریف رهنمودهایی برای آزمایش به این مسئله نزدیک شود، برای مثال ALE، اما تکرار آزمایش هایی که هفته ها به طول می انجامد مسئله ای است که احتمالاً به راحتی حل نخواهد شد.

قطعاً استفاده از الگوریتم هایی که با موفقیت در بازی های پیچیده سروکار دارند، در سایر زمینه های کاربردی نیز مطلوب است. متأسفانه، این موضوع معمولاً بی اهمیت نیست، اما برخی از نمونه های امیدوار کننده از قبل وجود دارند. روش AlphaGo که مبتنی بر جستجو با استفاده از MCTS در بازنمایی شبکه عصبی از مسئله درمان شده است، به مسئله رترو سنتز شیمیایی منتقل شده است که شامل یافتن مسیر سنتز برای یک جز chemical شیمیایی خاص است. در مورد مسئله سنتز، برخلاف بازی Go، مجموعه حرکات عملی (واکنشهای احتمالی) ارائه نشده است اما باید از داده ها آموخت، این روش شباهت زیادی با MuZero دارد. ایده یادگیری یک مدل پیش رو از داده ها، برنامه جهانی نامیده شده است. به همین ترتیب، از همان سیستم RL توزیع شده که OpenAI برای آموزش تیمی متشکل از پنج عامل برای Dota 2 استفاده می کرد [۱۲]، برای

آموزش یک دست روبات برای انجام دستکاری ماهرانه استفاده شد.  
تحقیقات هوش مصنوعی بازی باعث ایجاد نوآوری در دنیای هوش مصنوعی خواهد شد.



## فصل ۴

# نتیجه گیری

هوش مصنوعی رابطه ای دیرینه و خوب با بازی ها دارد. الگوریتم های AI از طریق بازی ها پیشرفته یا حتی اختراع شده اند. بازی ها، طراحی و توسعه آنها، به نوبه خود، عمدتاً توسط نقش های زیادی که AI در بازی ها بر عهده داشته است، بهره مند شده اند. در این پروژه روش ها و الگوریتم های اصلی مورد استفاده در زمینه هوش مصنوعی بازی را بیان کردیم. دریافتیم که این فقط هوش مصنوعی نیست که از طریق بازی ها پیشرفت می کند، بازی ها نیز از طریق تحقیقات AI پیشرفته شده اند. هوش مصنوعی در چندین جبهه به بهتر شدن بازی ها کمک کرده است: در نحوه بازی کردن آنها، در درک عملکردهای داخلی آنها، در طراحی آنها و در درک بازی، تعامل و خلاقیت. در آینده نیز تحقیقات هوش مصنوعی بازی باعث ایجاد نوآوری در دنیای هوش مصنوعی خواهد شد.

# واژه‌نامه فارسی به انگلیسی

model organism	ارگانیسم مدل
board games	بازی های رومیزی
procedural content generation	تولید محتوای رویه ای
Best-First Search	جستجوی بهترین-اول
tree search	جستجوی درختی
Depth-first search	جستجوی عمق اول
noninformed search	جستجوی ناآگاهانه
Behavior Trees	درختان رفتار
artificial neural network	شبکه عصبی مصنوعی
chess	شطرنج
Agent	عامل هوشمند
finite state machine	ماشین حالت محدود
Computational Intelligence	هوش محاسباتی
Artificial intelligence	هوش مصنوعی

game AI ..... هوش مصنوعی بازی

Utility-based AI ..... هوش مصنوعی مبتنی بر سود

reinforcement learning ..... یادگیری تقویتی

machine learning ..... یادگیری ماشین

# واژه‌نامه انگلیسی به فارسی

عامل هوشمند .....	Agent
هوش مصنوعی .....	Artificial intelligence
شبکه عصبی مصنوعی .....	artificial neural network
درختان رفتار .....	Behavior Trees
جستجوی بهترین-اول .....	Best-First Search
بازی های رومیزی .....	board games
شطرنج .....	chess
هوش محاسباتی .....	Computational Intelligence
جستجوی عمق اول .....	Depth-first search
ماشین حالت محدود .....	finite state machine
هوش مصنوعی بازی .....	game AI
یادگیری ماشین .....	machine learning
ارگانیسم مدل .....	model organism
جستجوی ناآگاهانه .....	noninformed search

تولید محتوای رویه ای .....	procedural content generation
یادگیری تقویتی .....	reinforcement learning
جستجوی درختی .....	tree search
هوش مصنوعی مبتنی بر سود .....	Utility-based AI

## کتابنامه

- Yannakakis, Georgios N., and Julian Togelius. Artificial intelligence [۱] and games. Springer, 2018.
- Risi, Sebastian, and Mike Preuss. "From chess and atari to starcraft [۲] and beyond: How game AI is driving the world of AI." *KI-Künstliche Intelligenz* 34.1 (2020): 7-17.
- Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon [۳] M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of Monte Carlo tree search methods. *Computational Intelligence and AI in Games, IEEE Transactions on*, 4(1):1–43, 2012.
- Nathan Ensmenger. Is Chess the Drosophila of AI? A Social History [۴] of an Algorithm. *Social Studies of Science*, 42(1):5–30, 2012.
- Jesper Juul. A casual revolution: Reinventing video games and their [۵] players. MIT Press, 2010.
- Tom Schaul, Julian Togelius, and Jürgen Schmidhuber. Measuring [۶] intelligence through games. arXiv preprint arXiv:1109.1314, 2011.
- Nicolas Usunier, Gabriel Synnaeve, Zeming Lin, and Soumith Chin- [۷] tala. Episodic Exploration for Deep Deterministic Policies: An Application to StarCraft Micromanagement Tasks. arXiv preprint arXiv:1609.02993, 2016.
- Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. [۸] Computational Game Creativity. In *Proceedings of the Fifth International Conference on Computational Creativity*, pages 285–292, 2014.

- Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. Sentient Sketchbook: Computer-aided game level authoring. In Proceedings of ACM Conference on Foundations of Digital Games, pages 213–220, 2013. [9]
- Kai Ninomiya, Mubbasir Kapadia, Alexander Shoulson, Francisco Garcia, and Norman Badler. Planning approaches to constraint-aware navigation in dynamic environments. *Computer Animation and Virtual Worlds*, 26(2):119–139, 2015. [10]
- Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2006. [11]
- Berner C et al (2019) Dota 2 with large scale deep reinforcement learning. arXiv :1912.06680 [12]

## **Abstract**

Since the birth of the idea of artificial intelligence, games have contributed to the advancement of artificial intelligence research. Many AI innovations have taken place in the gaming industry, and games have become a tool for testing AI applications in the real world. But it's not just AI that is advancing through games, games are also advancing through AI research. In fact, artificial intelligence and computer games are inextricably linked. In this project, artificial intelligence in games has been studied and artificial intelligence algorithms used in games have been studied.





College of Science  
School of Mathematics, Statistics, and Computer Science

# Artificial Intelligence Algorithms And Methods In Games

**Fatemeh Hosseinpour**

Supervisor: Dr.Sajedi

A thesis submitted in partial fulfillment of the requirements for  
the degree of B.Sc. in Computer Science

6 August 2021