دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

گزارش درس روش پژوهش و ارائه

آشنایی با یادگیری تقویتی و نقش آن در بازی‌ها

نگارش:

امیرحسین سرور

استاد راهنما:

دکتر رضا صفابخش

آبان ۱۴۰۰

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

گزارش درس روش پژوهش و ارائه

آشنایی با یادگیری تقویتی و نقش آن در بازی‌ها

نگارش:

امیرحسین سرور

استاد راهنما:

دکتر رضا صفابخش

آبان ۱۴۰۰

سپاس‌گزاری

لازم می‌دانم از استاد دلسوز و گران‌قدر **جناب آقای دکتر رضا صفابخش** تشکر کنم که مطالب بسیار زیادی را به من آموختند و بدون راهنمایی‌های ایشان، تهیه این گزارش امکان‌پذیر نبود. از زحماتشان صمیمانه سپاس‌گزارم و برایشان آرزوی توفیق روزافزون دارم.

امیرحسین سرور

آبان ۱۴۰۰

چکیده

بسیاری از مسائلی که در زندگی روزمره با آن‌ها مواجه می‌شویم و نیاز به تصمیم‌گیری داریم، ماهیتی متوالی و پیوسته دارند. در این دسته از مسائل، بازده نهایی به یک تصمیم مجزا وابسته نیست، بلکه به دنباله‌ای از تصمیمات بستگی دارد و برای بیشینه کردن بازده کل، تصمیم‌گیرنده ممکن است در برخی موارد از پاداش‌های آنی چشم‌پوشی کند تا بتواند در آینده پاداش و ارزش بیش‌تری را کسب کند. مسئله‌ی یافتن یک خط مشی مناسب برای تصمیم‌گیری، در حوزه‌ی یادگیری تقویتی بررسی می‌شود. در واقع، یادگیری تقویتی دانشی است که در آن به مطالعه و بررسی چگونگی رفتار یک عامل در مواجهه با محیط برای یادگیری سیاستی مناسب جهت اتخاذ اعمال می‌پردازد؛ با این هدف که عامل بتواند پاداش‌های تجمعی مورد انتظار برای انجام یک کار را به حداکثر برساند.

به عنوان یک روش شناخته‌شده برای حل مسائلی که در آن‌ها تصمیم‌گیری‌های متوالی نقش دارند، می‌توان به فرآیندهای تصمیم‌گیری مارکوف اشاره کرد. مهم‌ترین ویژگی این فرآیندها آن است که تصمیم بهینه در یک حالت معین، مستقل از حالات قبلی است که تصمیم‌گیرنده با آن‌ها مواجه شده است. فرآیندهای تصمیم‌گیری مارکوف، چارچوبی مناسب برای مدل‌سازی مسائل تصمیم‌گیری در حوزه یادگیری تقویتی به شمار می‌روند.

همچنین، یادگیری تقویتی و بازی‌ها تاریخچه پربار و مشترکی دارند. از یک طرف، بازی‌ها حوزه‌هایی غنی و چالش‌برانگیز برای آزمایش الگوریتم‌های یادگیری تقویتی محسوب می‌شوند و از طرفی دیگر، در بسیاری از بازی‌های رایانه‌ای عوامل هوشمند بازی از یادگیری تقویتی استفاده می‌کنند. در این گزارش، در ابتدا به بررسی مفهوم یادگیری ماشینی و انواع آن خواهیم پرداخت تا بتوانیم در فصل‌های بعد از آن، به طور دقیق‌تری یادگیری تقویتی را مورد مطالعه قرار دهیم. پس از مروری بر یادگیری تقویتی، به طور خاص نقش آن را در بازی‌ها بررسی می‌کنیم و جهت انسجام موضوع، به معرفی چند بازی پرداخته و عملکرد یادگیری تقویتی در آن‌ها را شرح می‌دهیم و در انتها، چالش‌های پیش‌رو در استفاده از یادگیری تقویتی در بازی‌ها را مورد بررسی قرار خواهیم داد.

واژه‌های کلیدی:

تصمیم‌گیری، خط مشی، یادگیری تقویتی، سیاست، پاداش‌های تجمعی، اتخاذ عمل، فرآیندهای تصمیم‌گیری مارکوف، مدل‌سازی، عوامل هوشمند بازی، یادگیری ماشینی

فهرست مطالب

عنوان صفحه

فهرست اشکال

عنوان صفحه

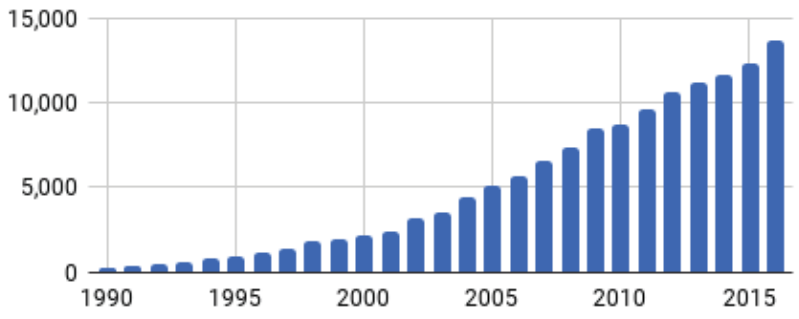
فصل اول

مقدمه

۱- مقدمه

یادگیری تقویتی عبارت است از یادگیری نگاشتی از حالات یا موقعیت‌ها به مجموعه‌ای از اقدامات یا اعمال، به منظور بیشینه‌کردن سیگنال پاداش. در این نوع یادگیری برخلاف سایر رویکردهای یادگیری ماشین، به یادگیرنده گفته نمی‌شود که چه اقدامی را باید انجام دهد، اما در عوض، عامل یادگیرنده باید خود با امتحان‌کردن اعمال مختلف، کشف کند که کدام اقدامات بیش‌ترین پاداش را برایش به همراه دارد. در برخی از حالات چالش‌برانگیز، اعمال لحظه‌ای ممکن است نه تنها بر پاداش آنی عامل، بلکه بر موقعیت بعدی و از این طریق بر تمام پاداش‌های بعدی تاثیر بگذارد. این دو ویژگی – آزمون و خطا و پاداش‌های تجمیعی – دو ویژگی متمایزکننده برای یادگیری تقویتی هستند[۱].

یادگیری تقویتی به نوعی هم موضوعی جدید و هم قدیمی در حوزه هوش مصنوعی به شمار می‌رود. از اولین پژوهش‌هایی که در این زمینه انجام شد، می‌توان به برنامه‌ی چکرزباز ساموئل (Samuel's Checkers-playing Program) اشاره کرد که از یادگیری تفاوت‌زمانی برای مدیریت پاداش‌های تجمعی استفاده می‌کرد. البته بحث «یادگیری» و «تقویت» تقریباً در حدود یک قرن است که جزو مطالعات حوزه روان‌شناسی محسوب می‌شود که تاثیرات زیادی هم بر حوزه هوش مصنوعی و مهندسی داشته است. در واقع، یادگیری تقویتی را می‌توان تماماً نوعی مهندسی معکوس از فرآیندهای یادگیری روان‌شناختی (مثلاً شرطی‌سازی عامل یا تقویت ثانویه) در نظر گرفت. علی‌رغم این مسائل، یادگیری تقویتی تا حد زیادی در اواخر دهه ۱۹۶۰ و همین‌طور در دهه ۱۹۷۰ فراموش شد، تا زمانی که در اوایل دهه ۱۹۸۰ به تدریج به یک حوزه فعال پژوهشی در یادگیری ماشین تبدیل شد[۱]. در شکل ۱-۱ ، نمودار تعداد مقالات منتشرشده در حوزه یادگیری تقویتی از سال‌های ۱۹۹۰ تا ۲۰۱۵ قابل مشاهده است[۲]. محور افقی نشان‌دهنده سال و محور عمودی تعداد مقالات ثبت‌شده در سایت google scholar را نشان می‌دهد.



شکل ۱-۱ رشد تعداد مقالات منتشرشده در حوزه یادگیری تقویتی[۲]

بخشی از جذابیت یادگیری تقویتی آن است که به نوعی یک مسئله هوش مصنوعی در یک جهان کوچک بررسی می‌شود. وظیفه یک عامل یادگیری مستقل این است که با دنیای خود برای رسیدن به یک هدف تعامل داشته باشد. این چارچوب، ساده‌سازی‌های لازم برای پیشرفت را تعیین می‌کند و در عین حال مواردی را شامل می‌شود که به وضوح فراتر از توانایی‌های فعلی ما هستند و آن‌ها را برجسته‌تر می‌کند؛ مواردی که تا زمانی که بسیاری از مشکلات کلیدی در یادگیری و بازنمایی حل نشوند، قادر به حل آن‌ها نخواهیم بود. در واقع این موضوع، چالش یادگیری تقویتی است[۱].

حال اگر از زاویه‌ای دیگر نگاه کنیم و به بررسی نقش یادگیری تقویتی در یکی از کاربردهای آن، یعنی بازی‌ها بپردازیم، می‌توان گفت بازی‌ها بستر بسیار مناسبی برای پژوهش در حوزه یادگیری تقویتی محسوب می‌شوند. بازی‌ها برای سرگرم‌کردن و به چالش‌کشیدن انسان‌ها طراحی شده‌اند؛ بنابراین با مطالعه بازی‌ها می‌توان امیدوار بود در مورد هوش انسان و چالش‌هایی که هوش انسانی باید از عهده حل‌کردن آن‌ها بربیاید، بیاموزیم. در عین حال، بازی‌ها خود نیز در حوزه یادگیری تقویتی چالش‌برانگیز محسوب می‌شوند؛ شاید به همان علتی که برای انسان‌ها و هوش انسانی نیز چنین است: آن‌ها برای تصمیم‌گیری‌ها طراحی شده‌اند[۳]. در فصل سوم، با جزئیات بیشتری به انواع چالش‌های موجود در مسیر پژوهش در حوزه بازی‌ها با رویکرد یادگیری تقویتی آشنا می‌شویم.

در این گزارش، هدف، آشنایی با یادگیری تقویتی و بررسی نقش آن در بازی‌هاست. در این راستا، لازم است ابتدا با مفهوم یادگیری و به طور خاص یادگیری ماشین آشنا شویم که در فصل اول به طور خلاصه به این موضوع خواهیم پرداخت. پس از آشنایی اولیه با مفاهیم یادگیری ماشین، روی یکی از رویکردهای آن یعنی یادگیری تقویتی، متمرکز خواهیم شد و این نوع از یادگیری را در فصل دوم مورد بررسی قرار می‌دهیم. پس از مروری بر یادگیری تقویتی، در فصل سوم نقش آن را در بازی‌ها خواهیم دید و همچنین با چند بازی و جایگاه یادگیری تقویتی در آن‌ها آشنا می‌شویم و در نهایت، به جمع‌بندی و نتیجه‌گیری موضوعات مورد بحث در گزارش خواهیم پرداخت.

فصل دوم

مروری بر یادگیری ماشین

۲- یادگیری ماشین

در این بخش می‌خواهیم به بررسی مفاهیم پایه در یادگیری خودکار یا همان یادگیری ماشین بپردازیم. در واقع، هدفمان این است که بتوانیم رایانه‌ها را طوری برنامه‌ریزی کنیم تا بتوانند از داده‌هایی که به عنوان ورودی به آن‌ها داده می‌شود، «یاد بگیرند» و عمل خود را بر مبنای «تجربه» انجام دهند.

۲-۱ یادگیری چیست؟

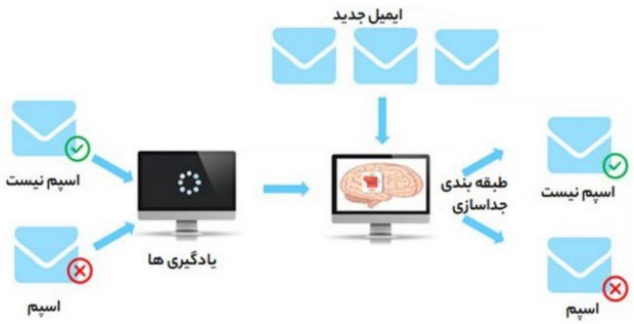
بحث را با مثالی از دنیای واقعی شروع می‌کنیم. هنگامی که موش‌ها با مواد غذایی با ظاهر یا بوی بدی مواجه می‌شوند، ابتدا مقادیر بسیار کمی از آن می‌خورند. خوردن یا نخوردن آن غذا در دفعات بعد، به طعم غذا و اثر فیزیولوژیکی آن بستگی دارد. اگر غذا تاثیر بدی داشته باشد یا باعث بیماری شود، متعاقباً موش‌ها آن را نخواهند خورد. واضح است که یک سازوکار[1] یادگیری در اینجا وجود دارد. حیوان از تجربیات گذشته خود در مورد بعضی غذاها، برای تشخیص ایمنی غذاهایی که در آینده با آن‌ها برخورد می‌کند استفاده می‌کند. اگر تجربه قبلی از آن غذا با برچسب منفی همراه بوده باشد، حیوان پیش‌بینی می‌کند که در آینده نیز خوردن این غذا تاثیر منفی خواهد گذاشت[۴].

بسیاری از مفاهیم پایه در یادگیری که اکثریت ما نیز با آن‌ها آشنا هستیم، در غالب همین مثال مطرح شده‌اند. به طور کلی، یادگیری را می‌توان به «فرآیند تبدیل تجربه به تخصص یا دانش» تعبیر کرد[۴]. اکنون فرض کنید می‌خواهیم عملیات یادگیری را توسط یک ماشین انجام دهیم. به عنوان مثال، می‌خواهیم ماشینی را برنامه‌ریزی کنیم که بتواند هرزنامه[2]ها را از رایانامه[3]های معمولی تشخیص دهد. یک راه حل ساده را می‌توان مطابق شکل ۲-۱ بر مبنای همان روشی در نظر گرفت که موش‌ها یاد می‌گیرند چگونه از طعمه‌های سمی اجتناب کنند. کافیست این ماشین تمام رایانامه‌های قبلی که توسط کاربر انسانی به عنوان هرزنامه برچسب‌گذاری شده بودند، به خاطر بسپارد و هنگامی که یک رایانامه جدید رسید، آن را در مجموعه هرزنامه‌های قبلی جست‌وجو کند؛ اگر با یکی از آن‌ها مطابقت داشته باشد، آن رایانامه را حذف یا به عنوان یک هرزنامه جدید برچسب‌گذاری کند. در غیراین صورت، رایانامه به صندوق ورودی کاربر منتقل می‌شود.

[1] Mechanism

[2] Spam

[3] Email



شکل ۲-۱ جداسازی هرزنامه‌ها توسط یک ماشین

۲-۲ اهمیت یادگیری ماشین

در اینجا می‌توان این سوال را مطرح کرد: چرا به جای برنامه‌ریزی مستقیم رایانه‌هایمان برای انجام عملیات‌های مورد نیاز، از یادگیری ماشین استفاده می‌کنیم؟ در حل یک مسئله داده‌شده، دو جنبه می‌تواند مطرح شود که در آن‌ها حل مشکل، نیاز به استفاده از برنامه‌هایی داشته باشد که عملکرد آن‌ها بر اساس «تجربه» تعیین شده و بهبود می‌یابد و ما را از برنامه‌ریزی مستقیم به سمت یادگیری ماشین سوق می‌دهد: پیچیدگی مشکل و نیاز به سازگاری.

۲-۲-۱ پیچیدگی مسئله

دسته‌ای از امور هستند که به علت پیچیدگی بیش از حد، قابل برنامه‌ریزی نیستند:

- *اموری که توسط انسان‌ها/حیوانات انجام می‌شوند*: امور متعددی وجود دارند که ما انسان‌ها به طور معمول انجام می‌دهیم، اما نحوه‌ی انجام آن‌ها توسط ما به اندازه کافی برای تعریف یک برنامه خوش‌فرم[۴]، دقیق نیست. نمونه‌هایی از این امور عبارتند از: رانندگی، تشخیص گفتار و درک تصویر. در این دسته از امور ، برنامه‌های پیشرفته یادگیری ماشین که بر مبنای «تجربه» هستند، زمانی که به اندازه کافی در معرض داده‌های آموزشی[۵] قرار می‌گیرند، به نتایج بسیار رضایت‌بخشی دست می‌یابند.

[4] Well-formed

[5] Training data

- *اموری که انجام آن‌ها خارج از توانایی انسان است:* خانواده بزرگ دیگری از امور که از فنون یادگیری ماشین بهره می‌برند، مربوط به تجزیه و تحلیل داده‌های بسیار بزرگ و پیچیده هستند. نمونه‌هایی از این امور عبارتند از: پیش‌بینی آب و هوا، موتورهای جست‌وجوی وب و تجارت الکترونیک. با وجود ذخیره روزافزون داده‌های دیجیتالی، مشخص می‌شود که گنجینه‌هایی از اطلاعات معنادار در بایگانی داده‌ها دفن شده‌اند که بسیار بزرگ‌تر و پیچیده‌تر از محدوده درک انسان‌ها هستند. یادگیری الگوهای معنادار در مجموعه داده‌های بسیار بزرگ و پیچیده حوزه امیدوارکننده‌ای است که با وجود حافظه و سرعت پردازش روزافزون رایانه‌ها، افق‌های جدیدی را می‌نمایاند.

۲-۲-۲ سازگاری

یکی از ویژگی‌های محدودکننده‌ی برنامه‌ریزی مستقیم، منعطف نبودن آن است. یک برنامه نوشته‌شده و نصب‌شده، بدون تغییر باقی می‌ماند. با این حال، بسیاری از امور در طول زمان یا از یک کاربر به کاربر دیگر، دچار تغییراتی می‌شوند. ابزارهای یادگیری ماشین – برنامه‌هایی که عملکرد آن‌ها بر مبنای داده‌های ورودی است – راه حلی برای این دسته از امور ارائه می‌دهند؛ چرا که طبیعتاً با تغییرات محیطی که با آن در حال تعامل هستند، سازگارند. از کاربردهای موفق یادگیری ماشین برای حل چنین مسائلی می‌توان به برنامه‌هایی اشاره کرد که متن دست‌نویس را تشخیص می‌دهند و می‌توانند با تغییرات بین دست‌خط کاربران مختلف سازگار شوند؛ و یا برنامه‌های تشخیص هرزنامه که به طور خودکار با تغییر در ماهیت هرزنامه‌ها سازگار می‌شوند، و برنامه‌های تشخیص گفتار.

۲-۳ انواع یادگیری ماشین

فرض اصلی در یادگیری از داده‌ها، استفاده از مجموعه‌ای از مشاهدات برای بدست آوردن سازوکار یک فرآیند است. این فرض به قدری گسترده است که نمی‌توان آن را در یک چارچوب واحد قرار داد. در نتیجه، رویکردهای متفاوتی برای مقابله با موقعیت‌ها و مفروضات مختلف پدید آمده‌اند[۵]. به طور متداول، سه نوع رویکرد اصلی در مسائل حوزه یادگیری ماشین وجود دارند که در می‌توان در شکل ۲-۲ آن‌ها را مشاهده کرد. در اینجا به معرفی اجمالی برخی از این رویکردها می‌پردازیم و در فصل‌های آینده، رویکرد «یادگیری تقویتی» و کاربرد آن را به طور خاص در بازی‌ها مورد بررسی قرار خواهیم داد.

شکل ۲-۲ انواع یادگیری ماشین

۲-۳-۱ یادگیری با نظارت[۶]

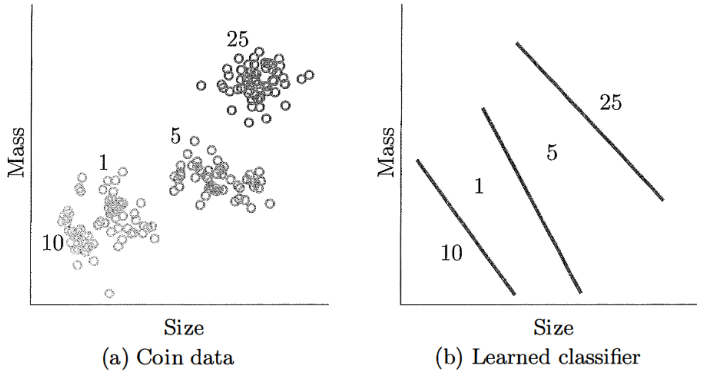
رویکردی که در ابتدای این فصل برای معرفی یادگیری ماشین به آن پرداخته بودیم، یادگیری تحت نظارت نامیده می‌شود که به نوعی پرکاربردترین نوع یادگیری ماشین نیز محسوب می‌شود. زمانی که در مجموعه داده‌های آموزشی برچسب یا خروجی صحیح به ازای هر ورودی مشخص شده باشد، این نوع از یادگیری به صورت نظارت‌شده خواهد بود. مسئله تشخیص رقم‌های دست‌نویس[۷] را به عنوان نمونه در نظر بگیرید. یک مجموعه داده آموزشی معقول برای این مسئله، مجموعه‌ای از تصاویر ارقام دست‌نویس و مقدار واقعی و عددی عدد نوشته‌شده در تصویر است. بنابراین ما مجموعه‌ای از زوج‌های مرتب (تصویر، رقم) را خواهیم داشت که در آن‌ها برچسب خروجی «تصویر» صریحاً به عنوان یک «رقم» معرفی شده است. در اینجا منظور از «نظارت» آن است که گویی یک «ناظر» مسئولیت نگاه کردن به هر تصویر ورودی و تعیین برچسب مقدار خروجی را برعهده گرفته است که در این‌صورت، برچسب خروجی، یکی از اعضای مجموعه $\{۹, ۸, ۷, ۶, ۵, ۴, ۳, ۲, ۱, ۰\}$ خواهد بود[۵].

[6] Supervised learning

[7] Hand-written recognition problem

۲-۳-۲ یادگیری بدون نظارت[8]

در یادگیری بدون نظارت، مجموعه داده آموزشی هیچ‌گونه اطلاعاتی درباره خروجی در اختیار ما نمی‌گذارد و تنها شامل داده‌های ورودی بدون برچسب است. در اینجا ممکن است این سوال پیش بیاید که چگونه می‌توان از این مجموعه داده ورودی چیزی یاد گرفت. این مسئله را در نظر بگیرید: تعداد بسیار زیادی سکه یک، پنج، ده و ۲۵ تومانی و همین‌طور اطلاعاتی از قبیل اندازه و جرم هرکدام از آن‌ها را در اختیار داریم و با استفاده از آن‌ها می‌خواهیم نوع یک سکه داده‌شده را پیدا کنیم. تا اینجای کار، مسئله از نوع دسته‌بندی[9] یا همان نظارت‌شده است؛ چرا که همان‌طور که در شکل ۲-۳ می‌بینیم، به ازای اطلاعات هر سکه، مقدار یا برچسب خروجی آن را در اختیار داریم و دسته‌بند[10]های ما برچسب ورودی‌ها را در هر دسته به طور صریح مشخص خواهند کرد.



شکل ۲-۳ دسته‌بندی سکه‌ها به صورت نظارت‌شده[۵]

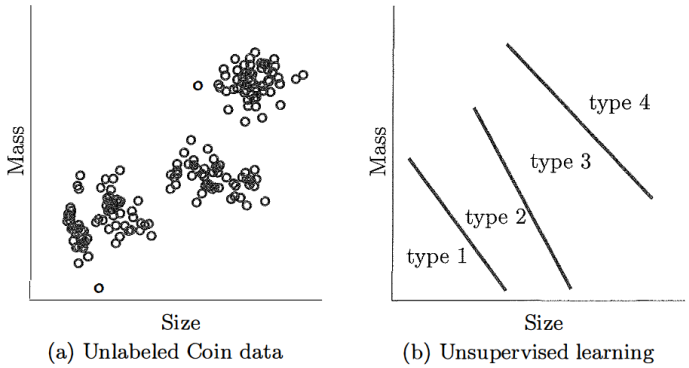
اکنون فرض کنید همان مجموعه داده ورودی قبلی را در اختیار داریم، با این تفاوت که ارزش هیچ‌کدام از سکه‌ها را نمی‌دانیم. در واقع، همان‌طور که در شکل ۲-۴ مشاهده می‌شود، داده‌های ورودی هیچ‌گونه برچسب خروجی ندارند. در نهایت، ما همان دسته‌بندها را – که این‌بار داده‌ها را «خوشه‌بندی[11]» کرده‌اند – خواهیم داشت، با این تفاوت که این دسته‌های جدید برچسب خاصی ندارند و تنها در یک «خوشه» قرار می‌گیرند؛ مثلا می‌توان به همه‌ی آن‌ها یک رنگ را نسبت داد. با این وجود، خوشه‌بندی صحیح و تعداد خوشه‌ها می‌تواند چالش‌برانگیز باشد[۵].

[8] Unsupervised Learning

[9] Classification

[10] Classifier

[11] Clustering



(a) Unlabeled Coin data (b) Unsupervised learning

شکل ۲-٤ خوشه‌بندی سکه‌ها به صورت بدون نظارت[۵]

یادگیری بدون نظارت زمانی مناسب‌تر است که به حجم انبوهی از داده‌های بدون برچسب دسترسی داشته باشیم. درک معنای پشت این داده‌ها نیازمند الگوریتم‌هایی است که بتوانند بر اساس الگوهایی که در داده‌ها پیدا می‌کنند، آن‌ها را خوشه‌بندی کنند. به عنوان مثال، در زمینه پزشکی جمع‌آوری حجم زیادی از داده‌ها در مورد یک بیماری خاص می‌تواند به پزشکان کمک کند تا الگوهایی در علائم بیماری بدست آورند و آن‌ها را با نتایج بیماران مرتبط کنند. برچسب‌گذاری تمام این داده‌های مرتبط با یک بیماری خاص، مثلا دیابت، زمان زیادی می‌برد؛ در نتیجه یک رویکرد یادگیری بدون نظارت می‌تواند سریع‌تر از یک رویکرد یادگیری تحت نظارت، به تعیین نتایج کمک کند[۶].

۲-۳-۳ یادگیری تقویتی[۱۲]

کودک نوپایی را در نظر بگیرید که یاد می‌گیرد به یک فنجان چای داغ دست نزند. تجربه چنین کودکی عموماً شامل مجموعه‌ای از موارد است که با فنجان چای مواجه می‌شود و تصمیم می‌گیرد که آن را لمس کند یا لمس نکند. هربار که کودک تصمیم به لمس فنجان می‌گیرد، نتیجه آن احتمالا سطح بالایی از درد بوده و هربار که تصمیم می‌گیرد فنجان را لمس نکند، میزان درد بسیار پایین‌تری حاصل شده است (مثل یک کنجکاوی ارضا نشده). در نهایت، کودک یاد می‌گیرد که بهتر است فنجان داغ را لمس نکند. داده‌های آموزشی در این مثال، بیان نمی‌کردند که کودک باید چه کاری انجام دهد؛ اما در عوض، اقدامات مختلفی را که انجام می‌داد درجه‌بندی می‌کردند. با این وجود، کودک از این داده‌ها و رتبه‌بندی آن‌ها استفاده می‌کند تا اقدامات خود را «تقویت» کند و در نهایت یاد می‌گیرد که در موقعیت‌های مشابه، چه کاری باید انجام دهد. از این نوع رویکرد به یادگیری با

[12] Reinforcement learning

عنوان «یادگیری تقویتی» یاد می‌شود که در آن، داده‌های آموزشی صریحاً شامل خروجی هدف نیستند؛ اما در عوض حاوی برخی از خروجی‌های ممکن همراه با معیاری از خوب بودن آن خروجی هستند. برخلاف یادگیری تحت نظارت که در آن داده‌های آموزشی به صورت (ورودی، خروجی صحیح) بودند، در یادگیری تقویتی داده‌های آموزشی به شکل (ورودی، مقداری خروجی، درجه‌ای [پاداشی] برای این خروجی) خواهند بود[۵].

از کاربردهای مفید یادگیری تقویتی می‌توان به یادگیری نحوه انجام یک بازی اشاره کرد. موقعیتی را در بازی تخته نرد تصور کنید که در آن بین اقدامات مختلفی حق انتخاب دارید و می‌خواهید بهترین حرکت را انجام دهید. تعیین بهترین حرکت در هر مرحله معین از بازی، کار ساده‌ای نیست؛ بنابراین، به راحتی نمی‌توان داده‌های آموزشی تحت نظارت تولید کرد. اما اگر از رویکرد یادگیری تقویتی استفاده کنیم، کافیست عملی را برای انجام دادن انتخاب کرده و بررسی کنیم بازی تا چه اندازه‌ای خوب پیش رفته است؛ به این ترتیب می‌توان داده‌های آموزشی لازم را ایجاد کرد. الگوریتم یادگیری تقویتی در اینجا وظیفه دارد تا اطلاعات بدست آمده از داده‌های آموزشی مختلف را برای پیداکردن حرکت بهینه در هر مرحله بررسی کند.

۲-۳-٤ شبکه‌های عصبی و یادگیری عمیق[۱۳]

با وجود اینکه سه رویکرد اصلی را در یادگیری ماشین بررسی کردیم، اما با توجه به کاربرد روزافزون شبکه‌های عصبی در هر سه رویکرد یادشده و همچنین به عنوان یکی از مورد مطالعه‌ترین حوزه‌های یادگیری ماشین، لازم است نگاهی هم به یادگیری عمیق و شبکه‌های عصبی داشته باشیم. همچنین یادگیری عمیق در رویکرد یادگیری تقویتی (که در فصل آینده به تفصیل به آن می‌پردازیم)، خود زیرشاخه جدیدی را با عنوان «یادگیری تقویتی عمیق[۱۴]» معرفی می‌کند که فراتر از بحث ما در این گزارش می‌باشد.

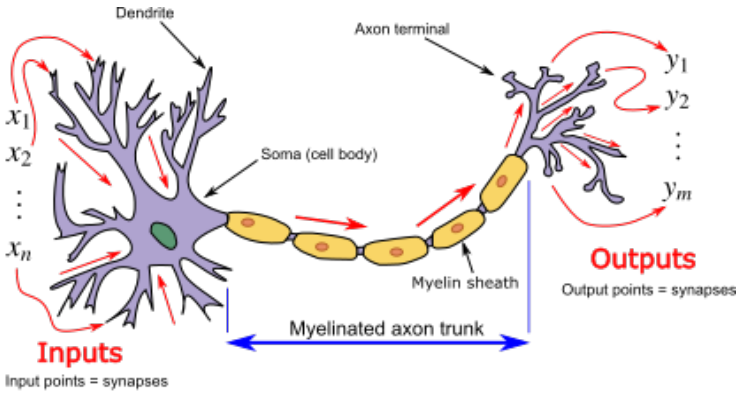
شبکه‌های عصبی مصنوعی، سیستم‌های محاسباتی‌ای هستند که از شبکه‌های عصبی زیستی[۱۵] که مغز را تشکیل می‌دهند، الهام گرفته‌اند. یک شبکه عصبی مصنوعی مبتنی بر مجموعه‌ای از گره[۱۶]های متصل به هم با نام «نورون‌های مصنوعی» است که نورون‌ها را در یک مغز بیولوژیکی مدل‌سازی می‌کنند. تصویر یک نورون و نحوه مدل‌سازی اجزای آن به یک نورون مصنوعی را در شکل ۲-۵ می‌توان دید. هر اتصال، مانند سیناپس‌های یک مغز می‌تواند اطلاعات یک «سیگنال ورودی» را پردازش کرده و از یک نورون مصنوعی به نورون دیگر منتقل کند[۷].

[13] Neural networks and deep learning

[14] Deep reinforcement learning

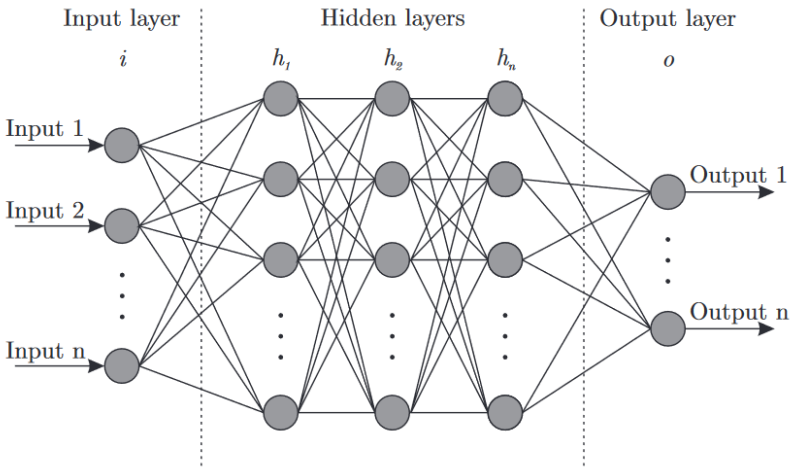
[15] Biological

[16] Node



شکل ۲-۵ بخش‌های یک نورون و نحوه مدل‌سازی آن به صورت یک نورون مصنوعی[۷]

همان‌طور که در شکل ۲-۶ مشاهده می‌شود، یک شبکه عصبی، از سه یا تعداد بیشتری لایه تشکیل شده است: یک لایه ورودی، یک یا چند لایه پنهان و یک لایه خروجی. داده‌ها از طریق لایه ورودی دریافت می‌شوند و سپس در لایه‌های پنهان و لایه خروجی براساس وزن‌های اعمال شده اصلاح می‌گردند. یک شبکه عصبی معمولی ممکن است از هزاران یا حتی میلیون‌ها گره تشکیل شده باشد که به طور متراکم به هم متصل هستند[۶].



شکل ۲-٦ ساختار یک شبکه عصبی مصنوعی[۸]

اصطلاح «یادگیری عمیق» زمانی استفاده می‌شود که لایه‌های پنهان متعددی در یک شبکه عصبی داشته باشیم. با وجود اینکه یک شبکه عصبی تک‌لایه نیز می‌تواند پیش‌بینی‌های تقریبی مناسبی انجام دهد، وجود لایه‌های پنهان بیشتر، می‌تواند کمک شایانی به بهینه‌سازی دقت خروجی کند. هر لایه در شبکه‌های عصبی عمیق، با استفاده از لایه‌های قبلی و بهبود آن‌ها برای

بهینه‌سازی پیش‌بینی یا طبقه‌بندی موردنظر تشکیل می‌شود که به این پیش‌روی محاسبات از طریق شبکه، «انتشار پیش‌رو[17]» گفته می‌شود. فرآیند دیگری به نام «انتشار پس‌رو[18]» نیز وجود دارد که از الگوریتم‌هایی مثل گرادیان نزولی[19] برای محاسبه خطاها در پیش‌بینی هر لایه استفاده می‌کند و سپس وزن‌ها را در هر لایه با حرکت به سمت عقب (از لایه خروجی به سمت لایه ورودی) برای آموزش مدل تنظیم می‌کند. این ساده‌ترین توصیف از عملکرد یک شبکه عصبی عمیق است؛ گرچه الگوریتم‌های یادگیری عمیق بسیار پیچیده‌اند[۹].

۲-٤ جمع‌بندی

در این بخش با مفاهیم یادگیری ماشین آشنا شدیم و دریافتیم که چرا نیاز داریم به‌جای برنامه‌ریزی مستقیم ماشین‌ها، آن‌ها را به نحوی برنامه‌ریزی کنیم که بتوانند بر اساس تجربه و با استفاده از مجموعه داده‌هایی که برای آموزش در اختیارشان می‌گذاریم، به نوعی «یاد بگیرند» که در محیط‌ها و موقعیت‌های مشابه چه عملکردی داشته باشند. در واقع هدفمان این بود که بتوانیم این عملکرد را در رویارویی با داده‌های جدید به بهینه‌ترین حالت ممکن برسانیم. همچنین دیدیم که به علت تنوع در فرضیات مسئله و نوع داده‌هایی که در اختیار ماشین قرار می‌گیرد، نمی‌توان یک رویکرد کلی برای یادگیری ماشین معرفی کرد. بنابراین سعی کردیم جنبه‌ها و رویکردهای مختلفی را که در مسائل یادگیری ماشین وجود دارند، بررسی کنیم و با سه رویکرد اصلی آن یعنی یادگیری تحت نظارت، یادگیری بدون نظارت و یادگیری تقویتی آشنا شدیم. در آخر نیز رویکرد یادگیری عمیق و همین‌طور سازوکار شبکه‌های عصبی مصنوعی را به طور مختصر بررسی کردیم که البته با اینکه جزو سه رویکرد اصلی ما محسوب نمی‌شد، اما به دلیل اهمیت و همین‌طور کاربرد روزافزون آن در هر سه رویکرد اصلی، به طور جداگانه به آن پرداختیم. در فصل آینده با جزئیات بیشتری به رویکرد «یادگیری تقویتی» خواهیم پرداخت.

[17] Forward propagation

[18] Back propagation

[19] Gradient descent

فصل سوم

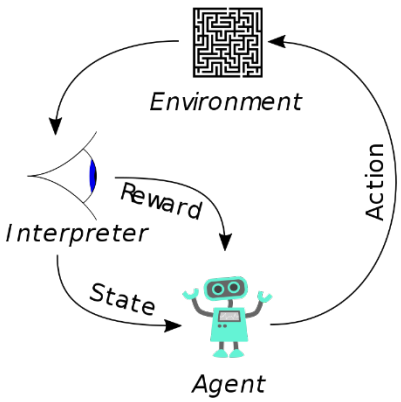
مسئله‌ی یادگیری تقویتی

۳- رویکرد یادگیری تقویتی

در فصل قبل با انواع یادگیری ماشین آشنا شدیم و تا حدودی به مرور رویکردهای مختلف به مسائل این حوزه پرداختیم. در این فصل می‌خواهیم به طور خاص به یادگیری تقویتی عمیق پرداخته و این رویکرد را تشریح کنیم. دانستیم که یک عامل محاسباتی در رویکرد یادگیری تقویتی، تصمیم‌گیری خود را در مواجه با مسائل بر اساس آزمون و خطا انجام می‌دهد. یادگیری تقویتی عمیق، به عامل این اجازه را می‌دهد تا تصمیمات خود را با استفاده از داده‌های بدون ساختار و بدون نیاز به مهندسی فضای حالت[۲۰] در نظر بگیرد[۱۰]. در ادامه بیشتر به این موضوع خواهیم پرداخت.

۳-۱ نحوه عملکرد یک عامل[۲۱]

در یادگیری تقویتی برخلاف دو رویکرد اصلی دیگر در یادگیری ماشین، یک عامل خودمختار یاد می‌گیرد که چگونه عملکرد خود را در مواجه با محیط برای انجام یک وظیفه محول شده بهبود بخشد. در تعریف یک «عامل» آمده است: «هر آنچه بتواند محیط اطراف خود را با استفاده از حسگرها درک کرده و در آن محیط با استفاده از محرک‌ها عملی را انجام دهد»[۱۱]. همانطور که در شکل ۳-۱ مشاهده می‌شود، در رویکرد یادگیری تقویتی، هیچ ناظری برای نشان دادن اقدام درست به عامل وجود ندارد؛ بلکه عملکرد عامل توسط یک تابع پاداش R ارزیابی می‌شود. در هر وضعیت، عامل اقدامی را انجام داده و براساس میزان مفید بودن یا نبودن آن اقدام، از محیط پاداش دریافت می‌کند. به تدریج، عامل می‌تواند پاداش بلندمدت خود را با بهره‌برداری از دانشِ آموخته شده درباره مطلوبیت مورد انتظار از جفت‌های وضعیت – عمل مختلف افزایش دهد.



شکل ۳-۱ عملکرد یک عامل یادگیری تقویتی در مواجهه با محیط[۱۰]

[20] State space

[21] Agent

یکی از چالش‌های اصلی در یادگیری تقویتی، مدیریت مبادله[۲۲] بین اکتشاف[۲۳] و بهره‌برداری[۲۴] است. برای بیشینه کردن پاداش‌های دریافتی، عامل باید با بهره‌برداری از دانش کسب شده، عمل‌هایی را انتخاب کند که بیش‌ترین پاداش را به همراه دارند. از طرف دیگر، برای کشف این عمل‌های سودمند، عامل باید خطر انجام دادن اقدامات جدیدی را بپذیرد که ممکن است به پاداش‌های بالاتری نسبت به بهترین اقدام‌های ارزیابی شده در وضعیت فعلی منجر شود. به عبارت دیگر، عامل باید از آنچه تاکنون یاد گرفته برای به‌دست آوردن بیش‌ترین پاداش بهره‌برداری کند؛ اما همچنان به اکتشاف اقدامات ناشناخته بپردازد تا بتواند در آینده اعمال بهتری انجام دهد[۱۲].

نمونه‌هایی از راهبردهایی که برای مدیریت این مبادله پیشنهاد شده‌اند عبارتند از: روش اپسیلون–حریصانه[۲۵] و روش بیشینه هموار[۲۶]. در روش اپسیلون حریصانه، عامل یا یکی از اعمال را به صورت تصادفی و با احتمال $0<\varepsilon<1$ انتخاب می‌کند، یا پرارزش‌ترین (دارای بیش‌ترین پاداش) عمل موجود در آن وضعیت را با احتمال $1-\varepsilon$ انجام می‌دهد. طبیعتاً در ابتدای فرآیند یادگیری که اطلاعات کمی در مورد محیط وجود دارد، عامل باید بیشتر به اکتشاف بپردازد؛ اما با پیشرفت این فرآیند، عامل ممکن است به تدریج از اکتشاف به سمت بهره‌برداری روی بیاورد[۱۲].

۳-۲ فرآیندهای تصمیم‌گیری مارکوف

برای آنکه بتوان تاثیر «عواقب داشتن اتخاذ عمل» را در نظر گرفت، نیازمند یک چارچوب ریاضیاتی مناسب هستیم که به کمک آن بتوان فرآیندهای تصمیم‌گیری یک عامل یادگیری تقویتی مستقل را به صورت ترتیبی مدل کرد[۱۳]. فرآیندهای تصمیم‌گیری مارکوف مناسب این کار هستند. یک فرآیند تصمیم‌گیری مارکوف عبارت است از:

- مجموعه حالات S
- مجموعه اعمال A
- تابع انتقال T
- تابع پاداش R

[22] Trade-off

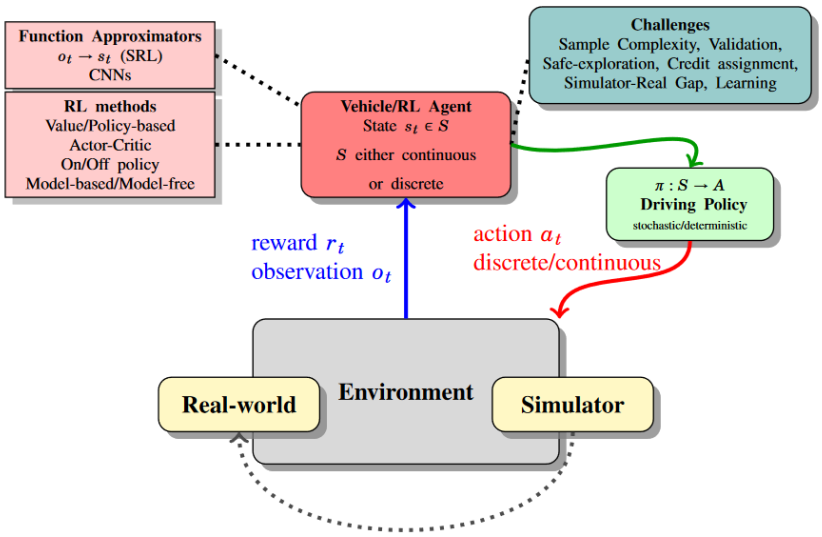
[23] Exploration

[24] Exploitation

[25] ε-greedy

[26] Softmax

بنابراین یک فرآیند تصمیم‌گیری مارکوف را می‌توان به صورت یک چندتایی <S,A,T,R> نمایش داد[۱۴]. هنگامی که در وضعیت دلخواه $s \in S$ هستیم، اتخاذ عمل $a \in A$ عامل را به وضعیت جدید $s' \in S$ با احتمال انتقال $T(s,a,s') \in (۰,۱)$ و پاداش $R(s,a)$ هدایت می‌کند که این فرآیند را می‌توان در شکل ۳-۲ مشاهده کرد.



شکل ۳-۲ فرآیند تصمیم‌گیری مارکوف برای یک عامل خودران[۱۲]

۳-۲-۱ سیاست بهینه و تابع ارزش-عمل

سیاست تصادفی[۲۷] $\pi: S \rightarrow \mathcal{A}$ در شکل ۳-۲ در صفحه ۱۷، یک نگاشت را از فضای حالت به یک احتمال روی مجموعه اعمال نشان می‌دهد. همچنین، $\pi(a|s)$ احتمال اتخاذ عمل a را در وضعیت s محاسبه می‌کند. هدف آن است که سیاست بهینه π^* را پیدا کنیم که منجر به بیشترین مقدار موردانتظار از پاداش‌های تخفیف‌یافته[۲۸] شود[۱۳]:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \underbrace{\mathbb{E}_\pi\{\sum_{k=0}^{H-1} \gamma^k r_{k+1} | s_0 = s\}}_{:=V_\pi(s)} \qquad (۱)$$

[27] Stochastic policy

[28] Discounted rewards

برای هر وضعیت $s \in S$ ، $r_k = R(s_k, a_k)$ نشان‌دهنده پاداش در لحظه k بوده و $V_\pi(s)$ یا «تابع ارزش[۲۹]» در وضعیت s و مبتنی بر سیاست π، خروجی(سودمندی) را با شروع از وضعیت s و سپس دنبال کردن سیاست π نشان می‌دهد[۱۳]. یک مفهوم مهم دیگر تابع عمل – ارزش یا تابع Q بوده که به صورت زیر تعریف می‌شود:

$$Q_\pi(s,a) = \mathbb{E}_\pi\{\sum_{k=0}^{H-1} \gamma^k r_{k+1} \,|s_0 = s\,, a_0 = a\} \qquad (۲)$$

۳-۲-۲ ضریب تخفیف[۳۰] و آینده‌نگری عامل

ضریب تخفیف $\gamma \in [۰,۱]$ میزان اهمیت پاداش‌های آتی را برای یک عامل مشخص می‌کند. اگر γ مقدار نسبتاً کمی باشد، رفتار عامل به وقایع جدیدتر بستگی بیشتری پیدا می‌کند و هدفش بیشینه کردن پاداش‌ها در کوتاه‌مدت می‌شود؛ در حالی که مقادیر بالای γ باعث می‌شود عامل آینده‌نگری بیشتری داشته و پاداش‌ها را در یک بازه بلندمدت بیشینه کند. متغیر H نیز به تعداد مراحل زمانی در فرآیند تصمیم‌گیری مارکوف اشاره دارد. در مسائل افق نامتناهی[۳۱] مقدار H برابر با بی‌نهایت در نظر گرفته می‌شود؛ در حالی که در مسائلی که دامنه محدود و رویدادمحور[۳۲] دارند، مقدار H متناهی است. دامنه‌های رویدادمحور ممکن است پس از تعداد معینی از مراحل زمانی یا هنگامی که عامل به یک حالت هدف مشخص می‌رسد، خاتمه یابند. آخرین حالتی که در یک مسئله رویدادمحور به‌دست می‌آید، حالت پایانی نامیده می‌شود. در این دسته از مسائل افق متناهی و هدف‌محور، ضریب تخفیف نزدیک به یک می‌تواند برای تشویق عامل به تمرکز روی رسیدن به هدف استفاده شود؛ در حالی که در مسائل افق نامتناهی ممکن است ضرایب تخفیف پایین‌تری برای برقراری تعادل بین پاداش‌های کوتاه‌مدت و بلندمدت مورد استفاده قرار گیرد[۱۲].

[29] Value function

[30] Discount factor

[31] Infinite-horizon

[32] Episodic

۳-۲-۳ خاصیت مارکوفی

در یک فرآیند تصمیم‌گیری مارکوف اگر سیاست بهینه را داشته باشیم، آنگاه می‌توان با استفاده از V_{π^*} و شروع از هر حالت اولیه دلخواه، بیشینه مقدار پاداش‌های تجمیعی تخفیف‌یافته را پیدا کرد؛ به این صورت که مسیر را در فضای حالت با اعمال متوالی و ترتیبی سیاست به یک حالت اولیه طی می‌کنیم. در اینجا می‌توان خاصیت مارکوفی را به این صورت تعریف کرد: یک فرآیند تصمیم‌گیری مارکوف خاصیت مارکوفی را برآورده می‌کند اگر تغییر حالت سیستم فقط به آخرین حالت و عمل اتخاذ شده بستگی داشته باشد، نه به تاریخچه کامل حالات و اقدامات در فرآیند تصمیم‌گیری. علاوه بر این، در بسیاری از حوزه‌های کاربردی در دنیای واقعی، امکان مشاهده تمام ویژگی‌های یک محیط برای عامل وجود ندارد. در چنین مواردی، مسئله تصمیم‌گیری به عنوان یک فرآیند تصمیم‌گیری مارکوف تا حدی قابل مشاهده[۳۳] در نظر گرفته می‌شود[۱۲].

۳-۳ روش‌های حل مسئله

حل یک مسئله یادگیری تقویتی به معنی پیداکردن سیاست π است به طوری که مجموع پاداش‌های تخفیف‌یافته مورد انتظار در مسیرهای طی شده در فضای حالت بیشینه گردد. عامل‌ها در یادگیری تقویتی ممکن است برآوردهایی از تابع ارزش، سیاست و یا مدل‌های محیط را به طور مستقیم بیاموزند. برنامه‌ریزی پویا[۳۴] به مجموعه‌ای از فرآیندها گفته می‌شود که می‌توانند برای محاسبه سیاست‌های بهینه با توجه به یک مدل کامل از محیط در قالب پاداش‌ها و توابع انتقال، استفاده شوند. برخلاف برنامه‌ریزی پویا، در روش‌های مونت کارلو[۳۵] هیچ فرضی بر دانش کامل از محیط نداریم. روش‌های مونت کارلو به صورت افزایشی و رویداد به رویداد هستند؛ به طوری که بعد از اتمام یک رویداد، برآوردهای ارزش و سیاست به‌روز می‌شوند. از سوی دیگر، روش‌های تفاوت زمانی[۳۶] به صورت افزایشی و گام‌به‌گام هستند؛ به طوری که آن‌ها را برای مسائل غیررویدادی[۳۷] قابل استفاده می‌کند. مانند روش‌های مونت کارلو، روش‌های تفاوت زمانی می‌توانند مستقیماً از تجربه خام و بدون مدلی از پویایی محیط، یادگیری را انجام دهند. همچنین روش‌های تفاوت زمانی مانند برنامه‌ریزی پویا برآوردهای خود را براساس دیگر برآوردها می‌آموزند[۱۲].

---

[33] Partially-observable Markov decision process (POMDP)

[34] Dynamic Programming (DP)

[35] Monte Carlo methods

[36] Temporal Difference (TD)

[37] Non-episodic

۳-٤ جمع‌بندی

در این بخش، رویکرد یادگیری تقویتی را به طور خاص مورد بررسی قرار دادیم. همچنین به تعریف «عامل» پرداختیم و با نحوه عملکرد یک عامل در یادگیری تقویتی آشنا شدیم. سپس یکی از چالش‌های اصلی را در یادگیری تقویتی که مدیریت مبادله بین اکتشاف و بهره‌برداری از دانش کسب‌شده بود، بررسی کردیم و راهکارهایی برای آن ارائه دادیم. علاوه بر این‌ها نیازمند یک چارچوب ریاضیاتی مناسب برای تعریف مسئله مورد نظرمان بودیم که در همین راستا فرآیندهای تصمیم‌گیری مارکوف را معرفی کرده و به جزئیات آن و نیز خاصیت مارکوفی پرداختیم. در نهایت مروری بر هدف اصلی مسئله یادگیری تقویتی داشتیم و چند روش حل را مورد بررسی قرار دادیم. در فصل بعد به طور خاص به نقش یادگیری تقویتی در بازی‌ها و چالش‌های پیش روی آن و همچنین بررسی راهکارهای موجود برای حل این چالش‌ها خواهیم پرداخت.

فصل چهارم

یادگیری تقویتی در بازی‌ها

٤- نقش یادگیری تقویتی در بازی‌ها

یادگیری تقویتی و بازی‌ها تاریخچه طولانی و پرباری دارند. «برنامه چکرزباز ساموئل[۳۸]» که یکی از برنامه‌های اولیه یادگیری ماشین به شمار می‌رود، ده‌ها سال قبل از اینکه یادگیری تفاوت زمانی (که در فصل قبل به آن پرداختیم) مطرح و تحلیل شوند از آن مفاهیم استفاده کرده بود. همچنین زمانی که بازی TD-Gammon (نوعی تخته نرد) جرالد تزارو[۳۹] توانست به سطح بازیکنان برتر انسانی برسد و حتی از آن فراتر برود و اینکار را کاملاً به تنهایی و با یادگیری انجام دهد، یادگیری تقویتی به اولین موفقیت بزرگ خود دست یافت. از آن زمان، یادگیری تقویتی در بسیاری از بازی‌های دیگر به کار گرفته شده است؛ هرچند نتوانسته در همه بازی‌ها به موفقیت TD-Gammon دست یابد، اما نتایج امیدوارکننده بسیار زیادی وجود دارند[۳].

٤-١ اهداف و ساختار

با وجود اینکه هدف این فصل بررسی کاربردهای یادگیری تقویتی در بازی‌هاست، اما یک موضوع مهم دیگر نیز در اینجا وجود دارد: بررسی اینکه الگوریتم‌های یادگیری تقویتی در عمل چگونه به صورت موفقیت‌آمیز عمل می‌کنند (یا حتی شکست می‌خورند). شاید تجزیه و تحلیل نظری آن‌ها به ما این اطمینان را بدهد که (در شرایط ایده‌آل) عملکرد خوبی دارند، اما می‌توان گفت این ایده‌ها در اکثر بازی‌ها، به علت شرایط مختلفی مثل محدودیت بالای قیود بازی یا سست بودن حالات، غیرعملی هستند. به عنوان مثال می‌دانیم اگر محیط، یک فرآیند تصمیم‌گیری مارکوف محدود باشد، ارزش هر حالت به صورت جداگانه ذخیره شده و نرخ یادگیری به شیوه مناسبی کاهش یافته باشد و اکتشاف به اندازه کافی صورت گیرد، یادگیری تفاوت زمانی[۴۰] به یک سیاست بهینه همگرا خواهد شد؛ در حالی که اکثر این قیود در یک برنامه بازی معمولی نقض می‌شود. با این وجود، یادگیری تفاوت زمانی در بازی تخته نرد به خوبی عمل کرده و در بازی‌های دیگر (مثلا تتریس[۴۱] [نوعی بازی خانه‌سازی]) چنین عملکردی ندارد. با این حال، تلاش‌های بسیار زیادی در راستای شناخت عواملی در بازی‌ها که باعث می‌شود یادگیری تفاوت زمانی و سایر الگوریتم‌های یادگیری تقویتی عملکرد خوبی داشته باشند، وجود دارد[۳].

---

[38] Samuel's Checkers-playing Program

[39] Gerald Tesauro

[۴۰] به طور نظری، یادگیری تفاوت زمانی به یک روش ارزیابی سیاست گفته می‌شود؛ در حالی که در ادبیات مرتبط با بازی، به معنای «یادگیری بازیگر-منتقد با انتقاد به وسلیه یادگیری تفاوت زمانی» به کار می‌رود.

[41] Tetris

٤-٢ معرفی چند بازی

تنوع زیاد بازی‌ها، انسجام موضوع را از بین می‌برد و سازمان‌دهی آن را سخت می‌کند. برای جلوگیری از این مشکل، با مروری از یادگیری تقویتی در چند بازی معروف که مورد مطالعه زیادی هم قرار گرفته‌اند، شروع می‌کنیم: تخته نرد و شطرنج. این بازی‌ها را می‌توان به عنوان موضوعات مورد مطالعه‌ای در نظر گرفت که بسیاری از مسائل مختلف را در حوزه یادگیری تقویتی معرفی می‌کنند.

٤-٢-١ تخته نرد

شکل ۴-۱ نمایی از بازی تخته نرد را نشان می‌دهد. برای طولانی نشدن گزارش، از بیان قواعد موجود در بازی می‌گذریم و فرض می‌کنیم که با قوانین آن آشنا هستیم[۴۲]. راهبردهای اساسی بازی شامل مسدودکردن ستون‌های کلیدی، ساخت بلوک‌های طولانی که پرش از آن‌ها سخت یا غیرممکن است و همچنین پیشبرد بازی تا انتها می‌شود. در عین حال بازیکنان باید بتوانند احتمال رویدادهای مختلف را در هر حرکت به طور نسبتاً دقیق تخمین بزنند.

شکل ٤-١ تخته نرد در چینش اولیه و شروع بازی[۳]

الگوریتم‌های سنتی جست‌وجو برای بازی تخته نرد به دلیل وجود عنصر «شانس»، موثر نیستند. در هر نوبت، ۲۱ نتیجه ممکن برای تاس انداختن وجود دارد که به طور میانگین برای هر یک از آن‌ها بیست حرکت قانونی وجود دارد. در نتیجه ضریب انشعاب بیش از چهارصد خواهد بود که عملاً جست‌وجوی پیش‌رو عمیق[۴۳] را غیرممکن می‌سازد[۳].

[۴۲] برای اطلاع از قوانین کامل بازی تخته نرد، می‌توانید به پیوند روبه‌رو مراجعه کنید: http:www.play65.com/Backgammon.html

[43] Deep look-ahead search

اولین نسخه از بازی TD-Gammon برای ارزیابی موقعیت‌های مختلف از یک شبکه عصبی استفاده می‌کرد؛ به طوری که ورودی شبکه موقعیت بازی و خروجی ارزش آن موقعیت بود. پس از انداختن تاس، برنامه تمام حرکات قانونی را بررسی کرده و ارزش آن‌ها را می‌سنجید، سپس پرارزش‌ترین حرکت را انتخاب کرده و آن را انجام می‌داد. سیگنال پاداش برای برد مقدار یک و برای باخت، صفر در نظر گرفته می‌شد؛ بنابراین مقادیر خروجی احتمال برنده‌شدن بازیکن را تخمین می‌زدند. نسخه‌های بعدی همچنین از ورودی‌هایی با ویژگی‌های پیشرفته‌تری (مثل فاصله کل مهره‌ها از هدف) استفاده می‌کرد. در هر مرحله، خروجی نورون‌ها با استفاده از قانون تفاوت زمانی و وزن‌های شبکه با انتشار پس‌رو به‌روز می‌شدند و این شبکه به صورت خودآموز و بدون هیچگونه اکتشافی آموزش داده می‌شد. به این ترتیب، TD-Gammon به طرز شگفت‌آوری عملکرد رضایت‌بخشی داشت. Neuro-gammon نیز نوعی دیگر از بازی تخته‌نرد قبل از ظهور TD-Gammon بود که از شبکه‌های عصبی با همان ساختار و معماری، اما با استفاده از نمونه‌های برچسب‌گذاری شده انسانی برای آموزش استفاده می‌کرد. با این وجود، TD-Gammon به صورت قابل توجهی قوی‌تر از Neuro-gammon ظاهر شد[۳].

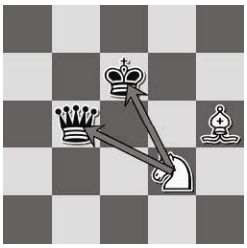
٤-۲-۲ شطرنج

شطرنج یکی از بازی‌های سطح بالایی است که یادگیری تقویتی شاید موفقیت چندانی در آن کسب نکرده است. با این وجود، می‌توان استدلال کرد که عملکرد یادگیری تقویتی در شطرنج به طور کلی بد نبوده است. به هرحال، شطرنج برای ده‌ها سال یکی از فعال‌ترین بازی‌ها در حوزه‌های پژوهشی بوده است. صفحه یک بازی شطرنج را می‌توانید در شکل ۴-۲ مشاهده کنید.

شکل ٤-۲ نمایی از یک صفحه بازی شطرنج[۳]

یادگیری تقویتی در بازی شطرنج، باید بتواند با الگوریتم‌های جست‌وجوی درختی که به طور خاص برای بازی‌های مجموع-صفر[۴۴] و دونفره که اطلاعات کاملی از وضعیت بازی در دسترس است طراحی شده‌اند، رقابت کند. علاوه بر این، تلاش‌های زیادی برای بهینه‌سازی این الگوریتم‌های جست‌وجو و همچنین ایجاد کتاب‌خانه‌های بزرگی برای شطرنج صورت گرفته است. در مقابل، رویکردهای یادگیری تقویتی مثل یادگیری تفاوت زمانی، عمومی‌تر هستند (به عنوان مثال، می‌توانند تصادفی‌بودن را مدیریت کنند)؛ بنابراین لزوماً کمتر موثر خواهند بود. با این حال، این سوال می‌تواند مطرح شود که چرا یادگیری تقویتی در شطرنج نتوانست موفقیت خود را در تخته نرد تکرار کند[۳].

برای روش‌های مبتنی بر ارزش، یکی از دلایل این است که ایجاد ویژگی‌های خوب برای شطرنج کار چندان ساده‌ای نیست. برای مثال، به شکل توجه کنید به این موقعیت اصطلاحاً «چنگال اسب[۴۵]» گفته می‌شود؛ زمانی که اسب همزمان شاه و وزیر را تهدید می‌کند. برای یادگیری اینکه چنگال اسب یک حالت خطرناک است (و بتوان آن را از حالت‌هایی که اسب بی‌خطر است متمایز کرد)، عامل باید بتواند در مورد موقعیت‌های نسبی مهره‌ها یاد بگیرد و آن را به موقعیت‌های مطلق تعمیم دهد، که در نمایش مربعی صفحه شطرنج کار بسیار پیچیده‌ای محسوب می‌شود.



شکل ٤-۳ موقعیت «چنگال اسب» در شطرنج[۳]

جالب اینجاست که شطرنج حتی برای روش‌های نمونه‌برداری فضای حالت (مانند الگوریتم‌های جست‌وجوی درختی مونت‌کارلو) یک چالش محسوب می‌شود. همچنین ماهیت قطعی[۴۶] شطرنج از عوامل دیگری است که کار را برای یادگیری تقویتی نیز دشوار می‌سازد. عامل‌ها باید به طور فعال اکتشاف کنند تا بتوانند از یک زیرمجموعه متنوع از فضای حالت تجربه کسب کنند. در حالی که رویکردهای یادگیری تقویتی در شطرنج به اندازه کافی برای کنترل بازی رقابتی نیستند، اما با قاطعیت می‌توان گفت که برای ارزیابی شرایط مناسب‌اند[۳].

---

[۴۴] در نظریه بازی‌ها، یک بازی مجموع-صفر (Zero-sum game) یک مدل ریاضی از وضعیتی است که سود (یا زیان) یک شرکت کننده، دقیقاً معادل با زیان‌های (یا سودهای) شرکت‌کننده (های) دیگر است. اگر مجموع سودهای شرکت کننده‌ها با هم جمع شود و مجموع زیان‌ها از آن کم شود، حاصل برابر صفر خواهد بود.

[45] Knight fork

[46] Deterministic

٤-٣ چالش‌های پیش رو

در بخش قبل چند بازی و نقش یادگیری تقویتی در آن‌ها را بررسی کردیم. این بازی‌ها به گونه‌ای بودند که هرکدام دسته‌ای از مسائل جالبی که در رویکرد یادگیری تقویتی با آن‌ها مواجه هستیم را شامل می‌شدند؛ اما با این وجود، به هیچ وجه نمی‌توان آن‌ها را نماینده‌ای از چالش‌های موجود در اعمال یادگیری تقویتی در بازی‌ها در نظر گرفت. در این بخش، فهرستی از چالش‌های موجود در یادگیری تقویتی که می‌توان در بازی‌ها به بررسی آن‌ها پرداخت، ارائه می‌کنیم.

٤-٣-١ اکتشاف

محوری‌ترین موضوع در هدایت یک عامل به سمت راه حل بهینه، اکتشاف است. انتخاب عمل به روش بولتزمن[47] و اپسیلون-حریصانه رایج‌ترین نوع اکتشاف محسوب می‌شوند. از نظریه فرآیندهای تصمیم‌گیری مارکوف متناهی می‌دانیم که روش‌های بسیار کارآمدتری نیز برای اکتشاف وجود دارند، اما هنوز مشخص نیست که چگونه می‌توان این روش‌ها را با استفاده از تقریب تابعی[48] به رویکردهای بدون مدل[49] یادگیری تقویتی (که بخش مهمی از کاربرد یادگیری تقویتی در بازی‌ها را پوشش می‌دهند) تعمیم داد[۳].

٤-٣-٢ داده‌های آموزشی

در بازی‌های دو یا چندنفره، محیط یادگیری عامل به بقیه بازیکنان بستگی پیدا می‌کند. در این صورت، هدف عامل برای یادگیری می‌تواند انتخاب بهترین عملکرد در برابر یک حریف ثابت، یا یک عملکرد نسبتاً خوب در برابر مجموعه‌ای از حریفان قوی باشد. با این حال، اگر حریف بسیار قوی‌تر از سطح واقعی عامل باشد، یادگیری می‌تواند به شدت ناکارآمد و یا حتی کاملاً مسدود شود؛ چرا که تمام راهبردهایی که یک عامل مبتدی اتخاذ کند، با احتمال بالایی با ضرر روبه‌رو خواهند شد و بنابراین سیگنال پاداش به طور یکنواخت منفی خواهد بود و هیچ رویکردی جهت بهبود عملکرد نشان نخواهد داد. به زبانی دیگر، حریفان باید به اندازه کافی متنوع باشند تا از همگرایی عامل به بهینه محلی جلوگیری کنند. به همین علت، انتخاب حریفان و داده‌های آموزشی اهمیت بالایی پیدا می‌کند[۳].

[47] Boltzmann

[48] Function approximation

[49] Model-free

٤-٣-٣ نحوه برخورد با اطلاعات ناموجود

عامل‌ها ممکن است چه در بازی‌های دیرینه[50] و چه در بازی‌های رایانه‌ای، با اطلاعات از دست رفته یا ناموجود (مثلا کارت‌های مخفی در بازی پوکر[51]) مواجه شوند. در اینجا، منظور فقط اطلاعاتی است که «واقعاً ناموجودند»، یعنی با بازنمایی بهتری از ویژگی‌ها نمی‌توان به آن‌ها دست یافت. یکی از رویکردهایی که در مواجهه با این چالش در نظر گرفته می‌شود، نادیده گرفتن این مشکل و یادگیری سیاست‌های انفعالی براساس مجموعه مشاهدات فعلی است؛ چرا که کاربرد روش‌های جست‌وجوی سیاست مستقیم[52] تحت تاثیر جزئی بودن مشاهدات قرار نمی‌گیرد. با این حال، این روش در نمونه‌برداری مونت‌کارلو در جست‌وجوی خصمانه دچار مشکل می‌شود؛ چرا که پس از نمونه‌برداری از مقادیر مشاهده‌نشده، عامل آن‌ها را مشاهده‌پذیر فرض می‌کند و همچنین می‌پندارد که حریف قابلیت مشاهده همه‌چیز را دارد. به این ترتیب، «ارزش اطلاعات» در تصمیم‌گیری نادیده گرفته شده و ممکن است یک حرکت تنها به علت داشتن اطلاعات اضافی، ترجیح داده شود[۳].

٤-٣-٤ مدل‌سازی حریف

رفتار و تصمیم‌گیری حریف در بازی، نوعی اطلاعات پنهان محسوب می‌شود. مدل‌های ساخته‌شده از حریف، سعی می‌کنند حرکات حریف را بر اساس مشاهدات گذشته و حتی بازی‌های گذشته پیش‌بینی کنند. در سیستم‌های یادگیری تقویتی با چند بازیکن، مدل‌سازی حریف یک مسئله مهم به‌شمار می‌رود؛ گرچه معمولاً به عنوان یک موضوع جداگانه به آن پرداخته نمی‌شود. با این وجود، اگر حریف کامل نباشد، بهره‌برداری از نقاط ضعف آن نیز می‌تواند مفید واقع شود[۳].

٤-٤ استفاده از یادگیری تقویتی در بازی‌ها

تا اینجای کار به دیدگاه‌های الگوریتمی و انواع مشخصه‌های چالش‌ها بر نقش یادگیری تقویتی در بازی‌ها پرداختیم. در این بخش می‌خواهیم رابطه‌ی میان یادگیری تقویتی و بازی‌ها را از نگاهی دیگر مورد بررسی قرار دهیم. سوال اصلی ما در این بخش این است: یادگیری تقویتی چگونه می‌تواند یک رویکرد مفید در بازی‌ها محسوب شود؟ در اینجا دو دیدگاه را بررسی خواهیم کرد که البته زمانی که هدف، یافتن یک بازیکن قوی با هوش مصنوعی باشد، این دو دیدگاه کم و بیش با هم مطابقت خواهند داشت.

[50] Classic games

[51] Poker

[52] Direct policy search methods

٤-٤-١ بیشینه‌کردن «سرگرمی»

برخلاف پژوهش‌هایی که در حوزه هوش مصنوعی بازی‌ها صورت می‌گیرد، هدف یک بازی ویدئویی سرگرم‌کردن بازیکن است. بخشی از این کار با تنظیم‌کردن درجه سختی روی سطح مناسب (نه خیلی دشوار و نه خیلی آسان) و پاداش‌دادن به بازیکن در صورت پیروزی در بازی با سطح دشوار انجام می‌شود. در این حالت، یک حریف ایده‌آل با هوش مصنوعی، اندکی ضعیف‌تر از بازیکنی است که در بهترین سطح خود قرار دارد. با این حال، به همان اندازه مهم است که عامل بازی به روشی «متقاعدکننده» ببازد. عامل باید این توهم را ایجاد کند که علی‌رغم اینکه هوشمندانه رفتار کرده و تمام تلاش خود را برای برنده‌شدن انجام داده، باخته است. البته، «سرگرم‌کننده»، «چالش‌برانگیز» و «هوشمند» هیچکدام مفاهیم کاملاً تعریف شده‌ای نیستند و امکان دارد تفسیرهای مختلفی از آن‌ها ارائه شود. در بسیاری از بازی‌ها، به همان اندازه که هوشمندکردن یک عامل چالش محسوب می‌شود، ضعیف کردن آن نیز می‌تواند چالش‌برانگیز باشد[۳]. در اینجا دو رویکرد را در راستای این چالش بررسی می‌کنیم:

- *سنجش دشواری*: فنون «سنجش دشواری» سعی می‌کنند درجه سختی بازی را به طور خودکار تنظیم کنند؛ به این صورت که یک راهبرد تطبیقی و سطح بالا در نظر می‌گیرند و در صورت لزوم قدرت بازی آن را در سطح بازیکنان تنظیم می‌کنند.
- *تطبیق‌پذیری*: از یادگیری تقویتی می‌توان برای تطبیق برخط[۵۳] حریفان با سبک بازی بازیکنان استفاده کرد. در اصل، این رویکرد می‌تواند بازیکن را از استفاده مکرر از ترفندهای مشابه به سمت انتخاب‌های متنوع‌تری در بازی هدایت کند و همچنین حریفان چالش‌برانگیزتری را فراهم کند.

٤-٤-٢ یادگیری حین توسعه

بهتر است پیش از انتشار بازی، یادگیری به صورت کامل و برون‌خط انجام شود و نتایج و سیاست‌های حاصل‌شده از بازی به طور کامل توسط توسعه‌دهندگان آزمایش و در صورت نیاز اصلاح شوند. در اینجا دو دیدگاه که نیازمند این نوع از یادگیری برون‌خط هستند را بررسی می‌کنیم: مدیریت خرد[۵۴] و تعادلِ بازی[۵۵].

[53] Online adaptation

[54] Micromanagement

[55] Game Balancing

٤-٤-٢-١ مدیریت خرد

یکی از مشکلات اصلی بازی‌های راهبردی بی‌درنگ[56] و نیز بازی‌هایی که به نوبت انجام می‌شوند (Turned-based) نیاز به مدیریت خرد است. بازیکن در حین گرفتن تصمیمات سطح بالا (مثل مکان حمله، فناوری سلاح و ...) باید بتواند تصمیمات سطح پایین‌تری هم بگیرد (مثل تخصیص نیروی کار در هر شهر یا مدیریت سربازان و کارگران). این‌گونه تصمیمات پس از مدتی خسته‌کننده و تکراری می‌شوند؛ بنابراین در این نوع از بازی‌ها، یک راه حل استاندارد برای جلوگیری از مدیریت خرد، استفاده از یک عامل هوش مصنوعی است که می‌تواند بخشی از این کار را انجام دهد[۳].

٤-٤-٢-٢ تعادل بازی

تعادل در بازی می‌تواند به دو مسئله اشاره داشته باشد:

- متعادل بودن شانس برد هر یک از حریفان در بازی‌هایی با طراحی نامتقارن: عدم تقارن می‌تواند خفیف باشد، مثلاً در یک بازی راهبردی بی‌درنگ که در آن هر دور بازی می‌تواند با نقاط قوت و نقاط ضعف مختلفی شروع شود؛ مثلاً در بازی تخته‌ای Last Night on Earth که یک بازیکن مردگان متحرک فراوان اما ضعیفی را کنترل می‌کند، در حالی که بازیکن دیگر کنترل یک گروه کوچک‌تر اما بسیار قدرتمند را برعهده دارد.
- تعادل بین راهبردهای مختلفی که یک بازیکن ممکن است اتخاذ کند.

به وضوح، اولین نوع تعادل برای داشتن یک بازی منصفانه ضروری است؛ در حالی که نوع دوم بیشتر برای اطمینان از آن است که بازیکن مجبور نباشد یک راهبرد غالب را به طور مکرر در پیش بگیرد و گزینه‌های متنوعی برای انتخاب داشته باشد تا بتواند تصمیمات معناداری گرفته و احتمالاً سرگرمی بیشتری داشته باشد. برقراری هیچکدام از انواع تعادل کار ساده‌ای نیست و نیاز به آزمایش گسترده بازی دارد. با این وجود، عدم تعادل ممکن است در طول آزمایش نیز شناسایی نشود[۳].

[56] Real-time Strategic

٤-٥ جمع‌بندی

در این فصل، برآن شدیم تا به طور خاص، نقش یادگیری تقویتی در بازی‌ها را مورد بررسی قرار دهیم. در همین راستا، ابتدا به تاریخچه استفاده از یادگیری تقویتی در بازی‌ها پرداختیم و سپس اهداف این کار را مشخص کردیم. در ادامه، به دلیل وسعت و تنوع زیاد بازی‌های موجود و جلوگیری از از دست رفتن انسجام موضوع، چند بازی را به طور خاص بررسی کردیم و مسائلی را که شامل استفاده از یادگیری تقویتی در آن‌ها می‌شد، مطرح کردیم. همچنین به بیان چالش‌های موجود در مسیر استفاده از یادگیری تقویتی در بازی‌ها پرداختیم و در نهایت، رابطه‌ی میان یادگیری تقویتی و بازی‌ها را تحلیل کردیم؛ در واقع سعی کردیم به این سوال پاسخ دهیم که یادگیری تقویتی چگونه می‌تواند رویکردی مفید برای بازی‌ها باشد. به طور خلاصه، در این فصل سعی بر آن بود تا موضوع اصلی گزارش که بررسی نقش یادگیری تقویتی در بازی‌هاست، تا حد خوبی پوشش داده شود. اکنون می‌توان گفت تا این قسمت از گزارش، با مفاهیم یادگیری ماشین، رویکرد یادگیری تقویتی به عنوان یکی از زیرشاخه‌های یادگیری ماشین و همچنین کاربرد یادگیری تقویتی در بازی‌ها، به طور مناسبی آشنا شده‌ایم.

فصل پنجم

نتیجه‌گیری و پیشنهادها

۵- نتیجه‌گیری و پیشنهادها

در بخش پایانی گزارش، جمع‌بندی و مروری بر سیر مطالب عنوان‌شده در گزارش خواهیم داشت. همچنین نتایج حاصل را بیان کرده و پیشنهادهایی برای ادامه کار در این موضوع ارائه می‌دهیم.

۵-۱ نتیجه‌گیری

در این گزارش، با هدف تمرکز بر نوعی خاص از یادگیری ماشین یعنی یادگیری تقویتی، و همچنین بررسی نقش آن در بازی‌ها، در ابتدا سعی کردیم مفاهیم یادگیری و انواع یادگیری ماشین را به طور مختصر شرح دهیم. دیدیم که یادگیری ماشین سه رویکرد اصلی دارد: یادگیری با نظارت، یادگیری بدون نظارت و یادگیری تقویتی. در توضیح هر یک از این رویکردها، با بیان مثال‌های کوتاهی سعی شد مفاهیم آن‌ها به صورتی ساده عنوان شود. پس از معرفی یادگیری ماشین، به سراغ موضوع اصلی بحث یعنی یادگیری تقویتی رفتیم و با جزئیات بیشتری این نوع از یادگیری را بررسی کردیم. در بیان مسئله‌ی یادگیری تقویتی، به معرفی عامل و نحوه عملکرد آن در محیط پرداختیم و همچنین فرآیندهای تصمیم‌گیری مارکوف را به عنوان یک چارچوب ریاضیاتی مناسب برای مدل‌سازی مسائل یادگیری تقویتی مطرح کردیم. در حل مسئله یادگیری تقویتی، سیاست بهینه و تابع ارزش – عمل را معرفی کردیم و همچنین مروری بر خاصیت مارکوفی داشتیم و در انتها، روش‌های حل مسئله به اختصار بیان شد. در قسمت پایانی گزارش، عملکرد یادگیری تقویتی در بازی‌ها را مورد ارزیابی قرار دادیم که اینکار با بیان تاریخچه‌ای مختصر از سوابق استفاده از یادگیری تقویتی در بازی‌ها آغاز شد. در ادامه جهت انسجام موضوع، چند بازی را به طور خاص مورد بررسی قرار دادیم و نقش یادگیری تقویتی در آن‌ها را از گذشته تاکنون بررسی کردیم. در این میان، یادگیری تقویتی در مواجهه با بازی‌ها با چالش‌هایی مانند داده‌های آموزشی، اطلاعات ناموجود و ... روبه‌رو بود که در این موارد هم توضیحاتی داده شد و در پایان، نحوه اعمال یادگیری تقویتی در بازی‌ها و در واقع نقشی را که یادگیری تقویتی در بازی‌ها ایفا می‌کند، عنوان کردیم.

شاید مهم‌ترین درسی که باید آموخت این است: فنون یادگیری تقویتی بسیار قدرتمند هستند، اما برای کارآمد بودن آن‌ها باید به مؤلفه‌هایی مثل نمایش مناسب، داده‌های آموزشی موثر، بستر مناسب اکتشاف و ... توجه ویژه‌ای کرد. همان‌گونه که این مؤلفه‌ها خود به تنهایی برای ایجاد عوامل قدرتمند در بازی‌ها یا هر کاربرد دیگری کافی نیستند، فنون یادگیری تقویتی نیز به تنهایی قادر به آموزش چنین عواملی نخواهند بود.

۵-۲ پیشنهادها

بازی‌ها با تعداد روزافزون کاربرانی که دارند، حوزه‌ای فعال و پربار برای پژوهش در راستای یادگیری تقویتی محسوب می‌شوند. یادگیری تفاوت زمانی، روش جستجوی درخت مونت‌کارلو و یادگیری تقویتی تکاملی[۵۷] از جمله محبوب‌ترین فنون کاربردی در حوزه یادگیری تقویتی محسوب می‌شوند. در بسیاری از موارد، رویکردهای یادگیری تقویتی با سایر فنون هوش مصنوعی و یا حتی هوش انسانی، قابل رقابت هستند. بازی‌ها به طور کلی حوزه‌ای هیجان‌انگیز محسوب می‌شوند و هنوز بسیاری از آن‌ها توسط هوش مصنوعی احاطه نشده‌اند؛ بنابراین فرصت‌های زیادی برای پژوهش در این زمینه‌ها وجود دارد.

یادگیری تقویتی خود نیز یک حوزه پژوهشی فعال و امیدوارکننده محسوب می‌شود که هنوز چالش‌ها و مسائل قابل حل زیادی در آن وجود دارند. بازساخت نتایج حاصل از تحقیقات در حوزه یادگیری تقویتی معمولاً دشوار است و وابستگی بسیار زیادی به فرامقادیر[۵۸] انتخاب شده در آزمایشات دارد که در اکثر اوقات با جزئیات کامل گزارش نمی‌شوند. علاوه بر این‌ها، پیاده‌سازی الگوریتم‌های یادگیری تقویتی کاری چالش‌برانگیز برای محققان و متخصصان به شمار می‌رود. محققین در حوزه یادگیری تقویتی باید نقطه شروع قابل اعتمادی داشته باشند که در آن، الگوریتم‌های شناخته‌شده یادگیری تقویتی به خوبی پیاده‌سازی شده و آزمایش شوند و از مستندات کافی نیز برخوردار باشند. برای این کار چارچوب‌های[۵۹] کارآمد و متن‌باز[۶۰] زیادی مانند Tensorflow Agents، OpenAI Baselines، RL Coach و ... وجود دارند که فرصت ارزیابی و توسعه الگوریتم‌های یادگیری تقویتی را فراهم می‌کنند[۱۲]. در نهایت، امید است این گزارش مشوق تحقیقات و پژوهش در زمینه‌ی یادگیری تقویتی واقع شود.

[57] Evolutionary Reinforcement Learning

[58] hyper-parameters

[59] Framework

[60] open-source

منابع و مراجع

[1] Sutton, R.S., 1992. Introduction: The challenge of reinforcement learning. In: *Reinforcement Learning* (pp. 1-3). Springer, Boston, MA.

[2] Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D. and Meger, D., 2018, April. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).

[3] Szita, I., 2012. Reinforcement learning in games. In *Reinforcement learning* (pp. 539-577). Springer, Berlin, Heidelberg.

[4] Shalev-Shwartz, S. and Ben-David, S., 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

[5] Abu-Mostafa, Y.S., Magdon-Ismail, M. and Lin, H.T., 2012. *Learning from data* (Vol. 4, p. 4). New York, NY, USA:: AMLBook.

[6] Mueller, J.P. and Massaron, L., 2021. *Machine learning for dummies*. John Wiley & Sons.

[7] En.wikipedia.org. 2021. *Artificial neural network - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Artificial_neural_network> [Accessed 7 December 2021].

[8] Bre, F., Gimenez, J.M. and Fachinotti, V.D., 2018. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, *158*, pp.1429-1441.

[9] Education, I., 2021. *What is Deep Learning?*. [online] Ibm.com. Available at: <https://www.ibm.com/cloud/learn/deep-learning> [Accessed 7 December 2021].

[10] En.wikipedia.org. 2021. *Artificial neural network - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Artificial_neural_network> [Accessed 7 December 2021].

[11] Brewka, G., 1996. Artificial intelligence—a modern approach by Stuart Russell and Peter Norvig, Prentice Hall. Series in Artificial Intelligence, Englewood Cliffs, NJ. *The Knowledge Engineering Review*, *11*(1), pp.78-79.

[12] Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A.A., Yogamani, S. and Pérez, P., 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

[13] van Otterlo, M., 2012. *Reinforcement learning: State-of-the-Art*. Springer Berlin Heidelberg.

[14] Puterman, M.L., 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.