

بررسی هوش مصنوعی مبتنی بر شبکه عصبی با هوش مصنوعی استاندارد در بازی های کامپیوتری هوشمند

احمد رضا کارگر^۱، لیدا زارعیان^۲

^۱ دانشجوی کامپیوتر، نرم افزار، گروه کامپیوتر، موسسه آموزش عالی اندیشه جهرم، جهرم

^۲ مربی موسسه آموزش عالی اندیشه جهرم، گروه کامپیوتر، موسسه آموزش عالی اندیشه جهرم، جهرم

چکیده

هوش مصنوعی در بازی های رایانه ای یک مفهوم شناخته شده است و به طراحی بازی اشاره دارد. به طوری که حتی در حالی که تنها بازی ها را انجام می دهد، در فعالیت هایی که توسط شخصیت های غیر واقعی^۱ شبیه سازی شده رفتار های انسان مانند وجود دارد. اغلب استدلال می شود که بازی ویدیویی مبتنی بر هوش مصنوعی واقعا هوشمند نیست و به سادگی مجموعه ای از دستورالعمل هایی است که توسط برنامه نویس نوشته شده است. از این رو، هدف این است که مطالعه و مقایسه ای از هوش مصنوعی سنتی را با هوش مصنوعی یادگیری ماشین مطالعه و مقایسه شود و براساس نتیجه استنتاج کنیم. یک بازی در موتور Unity که به اندازه کافی پیچیده باشد؛ توسعه یافته است به طوری که از نظر سختی برای یادگیری بازی، پیچیدگی بالایی وجود داشته باشد و در عین حال به اندازه ای ساده باشد که برای تولید نتیجه تا حد ممکن سریع باشد.

واژه های کلیدی: هوش مصنوعی ، توسعه بازی، یادگیری ماشین

^۱ NPCs

۱. مقدمه

هوش مصنوعی* به طور مخفف با عنوان AI از آن یاد می‌شود. هوش مصنوعی به عنوان شاخه‌ای از علوم کامپیوتر مشخص می‌کند که با خودکارسازی رفتارهای هوشمندانه سروکار دارد. در بازی‌های رایانه‌ای، از هوش مصنوعی برای تولید شبیه‌سازی هوش شخصیت‌های غیرقابل بازی استفاده می‌شود. تکنیک‌هایی که به صورت معمول استفاده می‌شود، از روش‌های موجود در رشته هوش مصنوعی بهره می‌گیرند. با این وجود، اصطلاح هوش مصنوعی بازی، اغلب، به مجموعه‌ای وسیع از الگوریتم‌هایی که شامل تکنیک‌های تئوری کنترل، رباتیک، گرافیک رایانه‌ای و علوم رایانه می‌باشد، مربوط است. از آنجایی که هوش مصنوعی در بازی‌های رایانه‌ای بر روی ظاهر هوش و گیم پلی[†] تمرکز دارد، روشی که از آن استفاده می‌کند با روش‌های معمول هوش مصنوعی سنتی تفاوت زیادی دارد. در اینجا روش‌های دور زدن باگ‌ها[‡] و تقلب در بازی[§] مورد پذیرش است. مثلاً در اکثر موارد، توانایی‌های کامپیوتر باید پایین آورده شود تا بازیکن‌های انسان بتوانند از پس حریف ماشین خود برآیند. این مثال مخصوصاً در بازی‌های تیراندازی اول شخص برجسته‌است. اگر نشانه‌گیری شخصیت‌های غیرقابل بازی فوق‌العاده باشد، بازی کردن با چنین ماشینی فراتر از حد توانایی یک انسان می‌شود. یک نمونه معمول از استفاده از AI در بازی‌ها شطرنج [۵] است که در آن تصمیم‌گیرنده AI بر اساس تکنیک نیروی بی‌رحم با استفاده از الگوریتم minimax است. بازی‌های مدرن از AI استفاده می‌کنند:

۱. طراحی مبارزه

۲. یافتن مسیر توسط NPC ها برای یافتن کوتاهترین مسیر بدون برخورد با زمین.

۳. شناسایی بازیکنان انسانی با مراجعه به نشانه‌ها یا در حوزه‌های بازی مانند صدای قدم‌ها شنوایی یا دیدن.

از این رو AI یک برنامه گسترده‌ای از مفاهیم هوش مصنوعی است که به یک توهم از هوش برای بازیکنان وجود

دارد.

به عنوان یکی از شاخه‌های وسیع و پرکاربرد هوش مصنوعی، یادگیری ماشین به تنظیم و اکتشاف شیوه‌ها و الگوریتم‌هایی می‌پردازد که براساس آن‌ها رایانه‌ها و سامانه‌ها توانایی تعلم و یادگیری پیدا می‌کنند. هدف یادگیری ماشین این است که کامپیوتر بتواند به تدریج و با افزایش داده‌ها کارایی بهتری در انجام وظیفه‌ی مورد نظر پیدا کند. گستره‌ی این وظیفه می‌تواند از تشخیص خودکار چهره با دیدن چند نمونه از چهره‌ی مورد نظر تا فراگیری شیوه‌ی گام‌برداری روبات‌های دویابا دریافت سیگنال پاداش و تنبیه باشد. طیف پژوهش‌هایی که در یادگیری ماشینی می‌شود گسترده است. پژوهشگران بر آن‌اند که روش‌های یادگیری تازه‌ای به وجود بیاورند و امکان‌پذیری و کیفیت یادگیری را برای روش‌هایشان مطالعه کنند و عده‌ای از پژوهشگران سعی می‌کنند روش‌های یادگیری ماشینی را بر مسایل تازه‌ای اعمال کنند. البته این طیف گسترده نیست و پژوهش‌های انجام‌شده دارای مؤلفه‌هایی از هر دو رویکرد هستند.

* Artificial Intelligenc

† Game play

‡ Workaround

§ Cheats

Unity یک موتور بازی چند سکویی است که توسط فناوری‌های یونیتی* ساخته شده است و در ساخت بازی ویدئویی برای کامپیوترهای شخصی، کنسول‌های بازی، دستگاه‌های همراه و وبسایت‌ها استفاده می‌شود. برای اولین بار در کنفرانس جهانی توسعه‌دهندگان اپل در سال ۲۰۰۵ برای سیستم عامل OS X معرفی شد و از آن زمان تا به حال بر روی بیست و یک سکوی دیگر توسعه یافته است. این موتور بازی‌ساز از نرم‌افزار موندولاپ که یک نرم‌افزار متن باز ویرایشگر زبان‌های برنامه‌نویسی است به عنوان ویرایشگر زبان برنامه‌نویسی استفاده می‌کند. این نرم‌افزار کاملاً با موتور بازی ساز یونیتی هماهنگ شده است. از مهم‌ترین قابلیت‌های این نرم‌افزار میتوان به پیشنهاد دهنده خودکار کلمات، سیستم اشکال‌یاب پیشرفته، پشتیبانی کامل از زبان برنامه‌نویسی سی شارپ اشاره نمود. Unity از جمله موتورهای بازی ساز است که می‌تواند از زبان‌های برنامه‌نویسی معروف پشتیبانی کند. زبان‌های برنامه‌نویسی سی‌شارپ، جاوااسکریپت و پایتون. Unity می‌تواند برای بسیاری از پلتفرم‌های موجود خروجی تهیه کند. امروزه تنها حدود ۵ درصد کاربران بازی‌ها را بر روی کامپیوترهای شخصی اجرا می‌کنند و سهم بسیاری به کنسول‌های بازی و موبایل و وب می‌رسد. تقریباً می‌توان گفت که Unity برای تمامی پلتفرم‌های موجود خروجی بازی تولید می‌کند و ممکن است کمتر نیاز به ویرایش اسکریپت‌های پروژه پیدا کنید. این قابلیت از لحاظ تجاری بسیار مهم است، به خصوص اجرای بازی بر روی وب و به صورت چند کاربر که باعث جذب کاربران بسیار می‌شود. تعدادی از بازی‌های موجود در شبکه اجتماعی فیس‌بوک توسط موتور بازی ساز Unity ایجاد شده است.

۲. مروری بر کارهای پیشین

با پیشرفت در دستگاه‌های ذخیره سازی و پردازش قدرتمندتر ماشین آلات، AI کارآمدتری در حال توسعه است. بازی‌ها بهترین نمونه‌هایی را ارائه می‌دهند که AI به طور مستقیم با یک انسان در سطح تصمیم‌گیری ارتباط برقرار می‌کند. این به ما کمک می‌کند که در زمینه‌های دیگری مانند انسان و ماشین‌ها از AI استفاده کنیم. عملکرد OEP در "بازی‌های اکشن چندبعدی: برنامه‌ریزی تکاملی آنلاین در برابر درخت جستجو" یک تکنیک بسیار مفید است، زمانی که AI قادر به درک دقیق به شاخه‌های مسیر برای یافتن راه حل بهینه برای رفع یک مشکل نیست [۱]. "مروری بر استفاده از تکنیک‌های AI در بازی‌های جدی: تصمیم‌گیری و یادگیری ماشین" اطلاعات زیادی در مورد تکنیک‌های توسعه AI در دهه گذشته و طبقه‌بندی آنها در ستون‌های مختلف برای مقایسه ارائه می‌دهد [۲].

ابرتکاملی یک پیشرفت عظیمی در مقایسه با تکنیک‌های تکاملی سنتی هستند که در اینجا ذکر شده است: "روش ابرتکاملی برای ایجاد استراتژی‌های سازگار برای بازی". همچنین در تعیین استراتژی‌های جدید در بازی‌هایی که از استراتژی‌های پیش ساخته استفاده می‌کنند کمک می‌کند [۳]. "اطلاعات بازی ترکیبی براساس درخت جستجو برای بازی ویدیویی عمومی" یک AI است که قادر به پخش انواع مختلفی از بازی‌ها با آشنا شدن با اشیاء بازی است. این ترکیبی است زیرا با استفاده از تکنیک‌های جستجوی قبلی که در بالا ذکر شده است، همزمان یک راه حل بهینه ارائه می‌شود [۴].

علاقه‌مندان در دنیای هوش مصنوعی و بنیانگذار OpenAI در پروژه مشابهی با شرکتش کار کرد تا یک ربات با استفاده از شبکه‌های عصبی را با تغذیه مجموعه داده‌های بازی ۲ DOTA بسازد. در عرض ۶ ماه، ربات به اندازه کافی مهارت داشت که توانست بازیکنان برتر بازی را که ۷ سال از مهارت هایشان در حال پیشرفت بودند، شکست دهد. ربات از بازیابی تقلید

* Unity Technologies

یادرخت جستجو برای حل مشکلات در بازی استفاده نکرد. این مکانیک بازی را از Scratch یاد گرفته و استراتژی های خود را ساخته است که حتی برای بازیکنان باتجربه غیر قابل پیش بینی و چالش برانگیز است [۶].

ایجاد یک AI با استفاده از روش های جستجو سنتی بسیار خسته کننده است و پیچیده تر می شود به عنوان مکانیک بازی و تعداد سناریو ها / تصمیم گیری افزایش می یابد. توسعه براساس شبکه عصبی اجازه می دهد که AI برنامه خود را با رعایت الگوهای تصمیم گیری انسان بارها و بارها، تکرار کند. از این رو در زمان و تلاش صرفه جویی می شود. AI با استفاده از تکنیک های جستجوی سنتی، هر زمان برای یک سناریوی مشابه، راهکارهای مشابهی را دنبال می کند و در تصمیم گیری آنها بسیار قابل پیش بینی و تکراری می شود AI مبتنی بر شبکه عصبی در حال تکامل است و اگر استراتژی فعلی با شکست مواجه شود، سعی دارد یک استراتژی جدید برای سناریو ارائه دهد. Unity، یک موتور توسعه بازی است که قادر است بازی ها و انیمیشن ها را برای چند سیستم عامل از طریق رایانه به دستگاه های تلفن همراه توسعه دهد. در ابتدا برای OS X منحصر به فرد بود اما بعدها به ۲۷ سیستم عامل دیگر نیز دسترسی پیدا کرد. این یک موتور همه منظوره است که اجازه می دهد تا سریع توسعه بازی ها با اجازه دادن به کاربر از استفاده از فیزیک قبل از آن و توانایی برای ایجاد مدل های دوبعدی* و سه بعدی[†] ساده در خود نرم افزار است. همچنین کاربر را قادر می سازد تا از سرویس های ابری استفاده کند تا بتواند از هر نقطه به پروژه خود دسترسی پیدا کند و آسانتر برای مدیریت داده های خود باشد. همچنین دارای مجموعه گسترده رایگان برای استفاده از داده ها است که به طور قابل توجهی بدون نیاز به طراحی مدل های خود و یا خرید آنها هدر رفتن زمان را کاهش می دهد.

۳. شیوه بازی

۳.۱. راه اندازی

محیط بازی یک عرصه با زمین های بیابانی و بسیاری از موانع درون آن است. عرصه با حصار پوشانده شده و شخصیت ها تنها اجازه حرکت در محیط داخلی خود را دارند. این شامل ۲ کاراکتر است که یکی از آن توسط بازیکن قابل کنترل است و دیگری AI که ما طراحی می کنیم. این بازی دارای نقطه نظر از بالا به پایین است و دوربین هم تراز شده و همیشه دارای شخصیت اصلی در مرکز آن است.

۳.۲. مکانیک

حرکت: اسکرپیت های حرکت به W، A، S، D در صفحه کلید ما محدود می شوند. با فشار دادن کلید، شخصیت را در جهت رو به جلو، چپ، عقب، و جهت درست در عرصه حرکت می دهد.

حمله اصلی: ضربه های اصلی به سمت دکمه سمت چپ ماوس محدود می شود. پس با فشار دادن، شخصیت بازیکن در جهت نشانگر ماوس حرکت می کند. این توانایی را می توان ۱۰ بار استفاده کرد که پس از آن بازیکن باید ۳ ثانیه دوباره بارگذاری کند. این بازیکن ۳۰٪ سرعت حرکت را کاهش داده است در حالیکه 'reload' در حال انجام است.

* ۲D

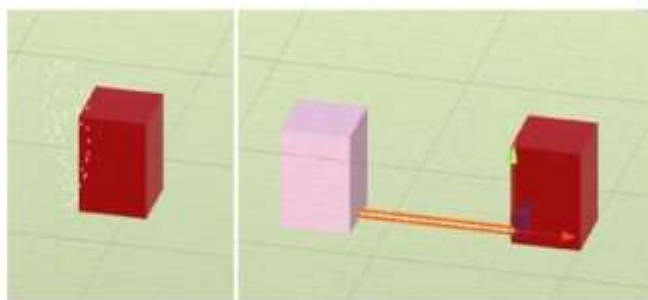
+ ۳D

حمله شگفت آور: این حمله یک حمله "شگفت آور" را در جهت ماوس پر می کند و با فشار دادن دکمه سمت راست ماوس شروع می شود. این حمله به مدت ۳ ثانیه به حریف دشمن خیره می شود و ضربه را انجام می دهد

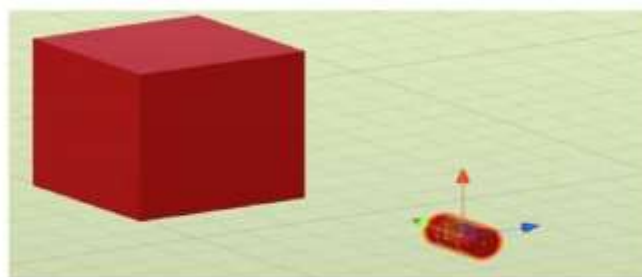
حمله سنگین: این حمله یک گلوله سنگین را در جهت ماوس می کشد و با نگه داشتن دکمه سمت چپ ماوس به مدت ۲ ثانیه باعث می شود. این گلوله به ۱۵ پرتابه اصلی آسیب می رساند. این ۲۰ ثانیه خنک کننده است.

۴.مدلسازی

ما با توسعه محیط بازی و شخصیت اصلی شروع می کنیم. Unity راحتی ایجاد زمین و دیگر مدل های اساسی را فوراً ارائه می دهد، بنابراین ما می توانیم شروع به تست کردن مکانیک را از سر بگیریم. ابتدا، ما یک شیء سه بعدی اساسی را بر روی زمین ایجاد می کنیم. بعد، سعی می کنیم اسکرپت های محلی را برای انواع مختلف اجرا کنیم Max ۳D یک برنامه کاربردی است که برای وارد کردن و رندر کردن مدل های سه بعدی ضروری است که بعدها فرم های ساده ما مانند مکعب و هواپیما را جایگزین می کند. شکل های ۱، ۲ و ۳ بیانگر این مدلسازی است.



شکل ۱: خط مستقیم و حرکت



شکل ۲: تصویربرداری تیراندازی



شکل ۳: جایگزینی شکل با مدل های سه بعدی

۵. شبکه عصبی

شبکه عصبی مصنوعی^{*} یک الگوریتم بسیار محاسباتی محاسباتی است که مفهوم ساده آن را از ساختار شبکه‌های عصبی زیستی[†] تعیین شده در آناتومی گرفته شده است. ANNها همچنین می‌توانند به عنوان "ساختارهای عصبی مصنوعی" یا سیستم‌های پردازش توزیع موازی یا ابزار ارتباط شناخته شوند. ANN خواستار یک گروه عظیمی از واحدها می‌شود که ممکن است در برخی از سبک‌ها به هر نوع متفاوت متصل شوند و این به امضای گفتمان شفاهی بین جریانه‌های واحد تبدیل شود. این ابزارها یا واحدها نیز به عنوان گره‌ها یا نورون‌ها شناخته می‌شوند که با یکدیگر موازی هستند. هر نورون از طریق اتصال چندین لینک به نورون‌های مختلف متصل می‌شود [۷].

هر مجموعه در ارتباط با یک وزن مربوطه است که اطلاعات را به صورت تقریبی مربوط به نشانه ورودی ادامه می‌دهد. این است که حداکثر آمار مفید برای هر نورون ظرفیت شناخت و مشخص کردن یک مشکل خاص را دارد، زیرا وزن به طور کلی با سیگنال این که در سراسر آن ارسال یا ارسال می‌شود، مطابقت دارد. هر نورون در حالت داخلی قرار دارد، که به عنوان سیگنال فعال شدن آن اشاره شده است. هشدار خروجی، که بعد از جمع آوری سیگنال‌های ورودی و قانون فعال سازی می‌تواند منتقل شود، در آن صورت علاوه بر واحد‌ها معرفی می‌شوند.

۶. نورون زیستی

در موجودات زنده، سلول‌های عصبی مسئول انتقال اطلاعات حساس در سراسر بدن هستند. تا 10^{11} نورون در بدن انسان تا 10^{15} اتصال به آنها وجود دارد. این تقریب خوبی است برای درک اینکه چگونه تصمیم‌گیری انسان را پیچیده می‌تواند باشد.

۴ بخش از نورون زیستی وجود دارد. توضیح کار هر یک به ما کمک می‌کند تا عملکرد آن را بهتر درک کنیم.

* Artificial Neural Network(ANN)

† Biological Neural Network(BNN)

دندریته‌ها* - اینها ساختارهای موضوعی هستند که مسئول دریافت اطلاعات از عصب های همسایه هستند. آنها مانند "گوش" نوروں عمل می کنند

سوما† - این بدن نوروں است. اطلاعات دریافت شده توسط دندریته‌ها در اینجا پردازش می شود.

آکسون‡ - بخش مرکزی طولانی از نوروں است که فرم سوما پردازش اطلاعات را به سیناپس حمل می - کند.

سیناپسها§ - محل تماس دو عصب است که اطلاعات را به دندریته‌ها از دیگر نوروں ها ارسال می کند.

جدول ۱ اصطلاحات معادل بین BNN و ANN را نشان می دهد.

جدول ۱: مقایسات اصطلاحات BNN در مقابل ANN.

شبکه عصبی مصنوعی (ANN)	شبکه عصبی زیستی (BNN)
ورودی	Dendrites
گره	Soma
خروجی	Axon
وزنه	Synapse

جدول ۲ مقایسه‌های بین ANN و BNN برای وظایف مختلفی است که آنها انجام می دهند.

جدول ۲: مقایسه BNN در مقابل تابع ANN.

شاخص	BNN	ANN
در حال پردازش	به طور موازی، آهسته اما کارآمدتر از ANN	خیلی موازی، سریع اما با کیفیت پایین تر از BNN
اندازه	10^{11} نوروں و 10^{15} اتصالات	10^2 تا 10^4 گره (بیشتر به نوع برنامه و طراح شبکه بستگی دارد)
یادگیری	آنها می توانند داده های گسترده را بیابند.	در نهایت، داده های ساخت یافته و فرمت شده برای درک اطلاعات مبهم مورد نیاز است
تحمل خطا	عملکرد می تواند حتی	برای عملکرد فوق العاده طراحی شده است، بنابراین می

* Dendrites

† Soma

‡ Axon

§ Synapses

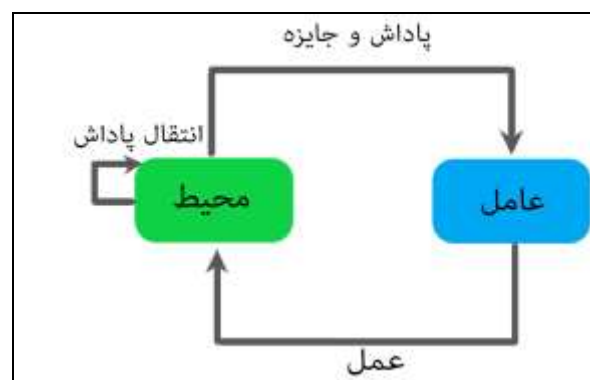
تواند هر نوع چالش را تحمل کند	با یک دقیقه صدمه ببیند	
این داده ها را در مکان های حافظه مداوم ذخیره می کند	این داده ها را به شکل سیناپس ذخیره می کند	گنجایش انبار

۷. طراحی و معماری یادگیری

همانطور که قبلا ذکر شد، بازیها اطلاعاتی ندارند، زیرا هیچ تکنیک یادگیری ندارند. برای مقابله با آن، یادگیری ماشین است که در آن مکانیسم برای AI برای گردآوری داده ها و تبدیل آن به دانش استفاده می شود. این مکانیسم از سیستم به سیستم متفاوت نیاز دارد و با دقت ساخته شده زیرا AI اطلاعات درست و قابل پیش بینی از آن را جمع آوری می کند. در مورد این بازی ها، AI توسط پاداش و مجازات نردبان هدایت می شود که به بهترین تصمیم در یک سناریوی داده می شود. یادگیری تقویت یک تکنیک مناسب برای چنین مدل یادگیری است.

در روش یادگیری تقویت، AI مجموعه ای از دستورالعمل ها را برای آنچه انجام می شود ارائه نمی دهد. در عوض به سادگی می گویند درست است یا اشتباه و بر اساس آن AI بازی را یاد می گیرد. سپس AI اقدامات تصادفی را در تلاش برای یادگیری از محدوده خود انجام می دهد، در حالیکه هر چیز درست، یک پاداش را به دست می آورد و هر کاری اشتباهی را مجازات می کند. انجام این کار به طور مکرر زمان کافی داده می شود که AI به یادگیری اقدامات انجام شده که احتمالا منجر به پاداش مثبت می شود، منجر شود.

Unity، متشکل از عامل های یادگیری ماشین* که به سادگی یک برنامه پایتون است که اجازه می دهد تا AI از تقویت یادگیری عمیق استفاده کند. این یک فرم پیشرفته یادگیری ماشین است که AI در مورد محیط خود از اعمالشان یاد می گیرد و تغییرات را می آموزد. رفتار آنها براساس آن است. این باعث می شود که بازی دشواری دینامیکی داشته باشد که در آن بیشتر عامل بازی[†] را بازی می کند و کارآمدتر می شود. شکل ۴ معرف چرخه آموزش تقویت یادگیری است.



شکل ۴: چرخه آموزش تقویت یادگیری

محیط یادگیری پیکربندی شده برای ML-agents شامل ۳ اشیاء اولیه است:

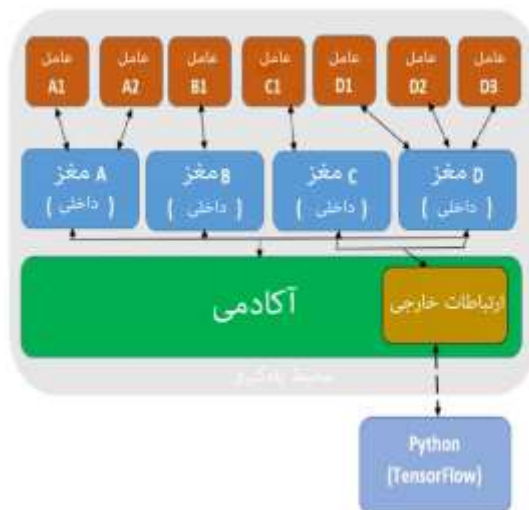
* Machine Learning Agents(ML-Agents)

[†] Game

عامل^{*}: همه عوامل دارای مجموعه ای منحصر به فرد از حالت ها، مشاهدات و اقدامات هستند، هر کدام از پاداش ها با رویداد مرتبط با آنها.

مغز[†]: مغز در این مورد (بسیار شبیه یک مغز انسان تصمیم می گیرد که چه اقداماتی برای سناریوهای خاص انجام شود.

آکادمی[‡]: این آیتم حاوی تمام مغز در محیط است. با این حال، در مورد AI تنها لازم نیست.



شکل ۵: نمودار نشان دهنده معماری یادگیری است.

در اینجا یادگیری تقویتی برای ساخت یک عصب استفاده می شود. همانطور که می دانیم شبکه های عصبی با استفاده از نورون ها ایجاد می شوند که در مجموع لایه ها را نشان می دهد. لایه داخلی با محیط ارتباط برقرار می کند و اطلاعات ورودی را جمع آوری می کند. از سوی دیگر، لایه خروجی مسئول ذخیره سازی خروجی برای مجموعه ای از ورودی ها می باشد. در نهایت، لایه پنهان در وسط این دو است و به سادگی یک لایه اضافی از ورودی ساندویچ برای ایجاد نتایج بیشتر استفاده میشود. این به چند پارامتر اجازه می دهد تا یک شبکه پیچیده ای از نتایج را ایجاد کنند که به همان اندازه که بتوان آن را به هوشمند تر شدن نزدیک کرد. نورونهای هر لایه متصل می شوند به طوری که خروجی هر لایه داده شده به ورودی لایه بعدی تبدیل می شود. آنها اغلب وزن دارند که درجه ارتباط بین دو نورون دو لایه را مشخص می کند. فرآیند یادگیری با داشتن عوامل انجام اقدامات در بازی اتفاق می افتد و سعی می کند بازیکن انسانی را ضرب و شتم کند. نتایج این اقدامات به نماینده همراه با پاداش و مجازات مربوط به آن اعمال می شود.

محیط یادگیری شامل "آکادمی" است که حاوی اطلاعاتی در مورد پارامترهایی است که در طول آموزش (عکس ها، عکس ها دریافت ...) به عنوان یک اسکریپت مورد استفاده قرار می گیرد. سپس این پارامترها به مغز منتقل می شود - یک نهاد هدایت نماینده با یک مدل آموزشی. سپس یک عامل به مغز متصل شده است، که به ما می گوید چه اقداماتی را انجام

* Agent

† Brain

‡ Academy

می‌دهد و اطلاعات برای این عمل دوباره برای تکمیل چرخه یادگیری ذخیره می‌شود. آموزش با استفاده از عملیات اضافی بنام اسکریپت تانسور جریان پایتون* است که مغز را به یک کتابخانه خارجی برای یادگیری متصل می‌شود. هنگامی که یک تکرار آموزش کامل است، تانسور جریان یک مدل جدید را ایجاد می‌کند که در واقع یک مغز جدید برای عامل است که از دستورات استفاده می‌کند. این به عنوان یک کنترل نسخه به عنوان پیشرفت AI عمل می‌کند. تابع پاداش نسبتاً ساده است، گلوله را با گلوله هدف قرار می‌دهد تا زمانی که زمان به پایان برسد. از سوی دیگر، برای مجازات AI برای دریافت عکس از بازیکن انسانی و در نهایت مرگ، یک مجازات وجود دارد. پس از بسیاری از تلاش‌ها، AI با استفاده بیشتر از مکانیک بازی، مانند استفاده از گلوله‌های انسانی و تیراندازی انسان با گلوله‌های برتر، بهبود یافت. همچنین دقت و صحت را بهبود می‌بخشد و قادر است پیش‌بینی کننده‌ی تصحیح یک انسان باشد. تابع پاداش به خوبی کار می‌کند که در نهایت به شناسایی عناصر بازی غیر ضروری کمک کرد. به عنوان مثال، اگر AI در یک نقطه باشد که بتواند به راحتی بدون استفاده از یک ویژگی خاص در بازی به راحتی از یک حریف انسانی بکاهد، این ویژگی می‌تواند کاملاً برطرف شود زیرا الگوریتم اعتقاد ندارد که استفاده از آن موجب افزایش شانس برنده شدن است [۸].



شکل ۶: مجازات و پاداش نردبان

۸. نتایج

پس از مجموع ۱۵۰ بازی در مقابل AI مبتنی بر شبکه سنتی و عصبی، طی تعداد دفعاتی که AI موفق به شکست یک بازیکن انسانی شده است نشان دهنده این است که روندی به طور کاملاً افزایشی برای شبکه عصبی و کاهشی در مورد AI استاندارد است. این به دلیل این واقعیت است که در حالی که AI شبکه عصبی از خروجی‌های بازی قبلی به عنوان ورودی استفاده می‌کند و بر آن افزوده می‌شود، AI استاندارد همان استراتژی را دوباره و دوباره تکرار می‌کند و در طول زمان قابل پیش‌بینی می‌شود. براساس قابلیت توانایی‌های AI که توسط یک بازیکن اخراج شده است معرف این است که افزایش کلی دفعاتی که توسط AI استاندارد دریافت می‌شود و در مورد AI شبکه عصبی کاهش می‌یابد. همین

* Tensor Flow Python

توضیح نیز در این مورد خوب است. در حالی که شبکه عصبی رشد می کند و قادر به پیش بینی عکس العمل های انسانی است. AI استاندارد براساس مسیرش بهبود نمی یابد و مسیر رمز شده آن را دنبال میکند که بسیار قابل پیش بینی و راحت تر برای ضربه زدن است.

بعضی از محدودیت هایی که سیستم شبکه عصبی مواجه می شود، به شرح زیر است:

نیاز به مجموعه داده های عظیم: شبکه های عصبی قادر به پیدا کردن راه حل مشکل پیچیده به همان شیوه انسان هستند، با این حال، آنها نیاز به مجموعه داده های بسیار زیادی دارند. انسان در حال حاضر حل این مشکلات برای ساختن شبکه خود را، با سرعت داراست

راه حل های غیر قابل اعتماد زمانی رخ می دهد که غیر منتظره است: AI مبتنی بر شبکه عصبی معمولا موثر است. با این حال، آنها گاهی اوقات نتایج غیر قابل اعتماد را تولید می کنند. این اتفاق می افتد در موارد خاص که در آن مشکل به طور کامل به AI جدید است و برای آن در مجموعه داده های آن ثبت نشده است.

داده های کامل: برای ایجاد AI مبتنی بر شبکه عصبی قابل اعتماد، نیاز به داشتن سیستم بسیار خوب است که برای آن ساخته شده است. اگر سیستم دارای متغیرهایی باشد که مقادیر هیچ ارتباطی ندارند، شبکه عصبی نیز راه حل هایی را ایجاد می کند که قادر به کار در درازمدت نیستند. شبکه برای دسترسی به تمام متغیرها و عوامل موثر بر آن متغیرها برای دستیابی به موفقیت نیاز دارد.

۹. نتیجه گیری و کارهای آینده

این مطالعه نشان می دهد که AI مبتنی بر شبکه عصبی قادر به تولید راه حل برای مشکلات پیچیده برای سیستم های مربوط به انسان است AI استاندارد بیشتر و بیشتر قابل پیش بینی بود در حالی که ماشین یادگیری AI تنها با هر بازی به منظور ضربه زدن به یک انسان بهبود یافته است. یادگیری عمیق و شبکه عصبی در حال تکامل هستند و از آن به بعد از این مرحله بیشتر و بیشتر پیشرفت می کنند. این یک گام بزرگ در توسعه هوش مصنوعی واقعی است که می تواند تقریبا به عنوان یک انسان واقعی هوشمند باشد.

این روش همچنین می تواند در بسیاری از سیستم عامل های دیگر استفاده شود که می تواند بازار سهام را پیش بینی کند، در تشخیص پزشکی بر اساس علائم و سوابق گذشته کمک کند. هنوز فضای زیادی برای رشد در این حوزه وجود دارد و می تواند یک نیروی محرک آینده نسل بعدی کامپیوترها به شمار می رود.

۱۰. مراجع

۱. Justesen, N., Mahlmann, T., Risi, S., & Togelius, J. (۲۰۱۸). Playing Multi action Adversarial Games: Online Evolutionary Planning Versus Tree Search. IEEE Transactions on Games, ۱۰(۳), ۲۸۱-۲۹۱.

۲. Frutos-Pascual, M., & Zapirain, B. G. (۲۰۱۷). Review of the use of AI techniques in serious games: Decision making and machine learning. IEEE Transactions on Computational Intelligence and AI in Games, ۹(۲), ۱۳۳-۱۵۲.

٣. Sehrawat, A., & Raj, G. (٢٠١٨, June). Intelligent PC Games: Comparison of Neural Network Based AI against Pre-Scripted AI. In ٢٠١٨ International Conference on Advances in Computing and Communication Engineering (ICACCE) (pp. ٣٧٨-٣٨٣). IEEE.
٤. Joppen, T., Moneke, M. U., Schröder, N., Wirth, C., & Fürnkranz, J. (٢٠١٨). Informed Hybrid Game Tree Search for General Video Game Playing. *IEEE Transactions on Games*, ١٠(١), ٧٨-٩٠.
٥. Pell, B. (١٩٩٣). Strategy generation and evaluation for meta-game playing (Doctoral dissertation, University of Cambridge).
٦. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (٢٠١٦). Mastering the game of Go with deep neural networks and tree search. *nature*, ٥٢٩(٧٥٨٧), ٤٨٤.
٧. Risi, S., & Togelius, J. (٢٠١٧). Neuroevolution in games: State of the art and open challenges. *IEEE Transactions on Computational Intelligence and AI in Games*, ٩(١), ٢٥-٤١.
٨. Justesen, N., & Risi, S. (٢٠١٧). Learning macromanagement in starcraft from replays using deep learning. *arXiv preprint arXiv:١٧٠٧.٠٣٧٤٣*.