DOI: 10.22122/jrrs.v15i1.3446

Published by Vesnu Publications

بهبود یادگیری تقویتی عمیق با برنامه درسی در فضای بازی

محمدرضا محمدنژاد[۱]، مرتضی دری‌گیو[۲]، فرزین یغمایی[۳]

مقاله پژوهشی

چکیده

مقدمه: یادگیری تقویتی عمیق با برنامه درسی (Curriculum learning)، شیوه‌ای از آموزش عامل هوشمند است که ابتدا عمل‌های ساده و سپس عمل‌های سخت به عامل آموزش داده می‌شود تا عامل هوشمند بتواند عمل‌های پیچیده در فضای گسترده بازی را بهتر آموزش ببیند.

مواد و روش‌ها: در مطالعه حاضر، از یادگیری تقویتی عمیق با برنامه درسی برای آموزش عامل هوشمند در فضای بازی غار اژدها استفاده گردید. آموزش برنامه درسی از فعالیت‌های ساده شروع شد و به تدریج به فعالیت‌های سخت‌تر رسید. به کمک بهینه‌سازی نزدیک خط‌مشی، عوامل هوشمند در محیط‌های متفاوت یکی در محیطی با برنامه درسی و دیگری در محیط بدون برنامه درسی آموزش داده شد. سپس هر دو در محیطی یکسان شروع به بازی کردند.

یافته‌ها: یافته‌ها حاکی از بهبود کیفیت عامل هوشمند با برنامه درسی نسبت به عامل هوشمند یادگیری تقویتی عمیق بدون برنامه درسی بود.

نتیجه‌گیری: استفاده از یادگیری تقویتی با برنامه درسی، باعث افزایش سرعت و کیفیت آموزش عامل هوشمند در محیط‌های بازی پیچیده بازی‌های استراتژیک می‌شود.

کلید واژه‌ها: عامل هوشمند، یادگیری تقویتی عمیق با برنامه درسی، یادگیری ماشین، شبکه عصبی

ارجاع: محمدنژاد محمدرضا، دری‌گیو مرتضی، یغمایی فرزین. **بهبود یادگیری تقویتی عمیق با برنامه درسی در فضای بازی.** پژوهش در علوم توانبخشی ۱۳۹۸؛ ۱۵ (۱): ۵۷-۵۰

تاریخ دریافت: ۱۳۹۷/۱۱/۲۰ تاریخ پذیرش: ۱۳۹۷/۱۲/۲۰ تاریخ چاپ: ۱۳۹۸/۱/۱۵

مقدمه

هوش مصنوعی باکیفیت در سال‌های اخیر، تأثیر بسزایی در فروش بازی‌های کامپیوتری داشته است، اما در بازی‌ها با محیط پیچیده، هوش مصنوعی بازی بسیار عقب‌تر از هوش انسانی می‌باشد (۱). پیاده‌سازی رفتارهای هوش مصنوعی به صورت دستی، زمانبر و دشوار است و استفاده از تکنیک‌های یادگیری ماشین (Machine learning)، جایگزین امیدوارکننده‌ای برای تولید رفتار هوشمند به شمار می‌رود. یادگیری تقویتی عمیق (Deep reinforcement learning یا DRL) که بر یادگیری از روی آزمون و خطا تمرکز می‌کند، نوعی آموزش هوش مصنوعی می‌باشد که کاربرد آن در فضای بازی موفقیت‌آمیز بوده است (۲).

هنگامی که عمل‌های بازی بسیار پیچیده و پاداش عمل‌ها تنک (Spare) باشند، یادگیری تقویتی برای همگرا شدن آموزش زمان زیادی می‌برد و آموزش عامل به صورت مستقیم در محیط پیچیده بازی، عملکرد ضعیفی دارد (۳). به منظور بهبود آموزش عامل در پژوهش حاضر، از یادگیری تقویتی عمیق با برنامه درسی استفاده شد که عامل را با دنباله‌ای از محیط‌هایی که به تدریج سخت‌تر می‌شوند، آموزش می‌دهد. یادگیری با برنامه آموزشی به نوعی از یادگیری گفته می‌شود که عامل، آموزش را با مثال‌های ساده‌ای از کار شروع می‌کند و سپس به تدریج دشواری کار افزایش داده می‌شود (۴). این نوع آموزش در زمینه‌های مختلفی مانند طبقه‌بندی تصویر اعمال می‌شود و عملکرد مناسبی نسبت به سایر رویکردهای یادگیری ماشین دارد (۵).

بعضی از عمل‌ها در محیط بازی به قدری بزرگ هستند که می‌توان آن را به چند عمل فرعی (Subtask) با درجات متفاوت دشواری تقسیم کرد و در ادامه پس از تجزیه عمل، با برنامه درسی آن را آموزش داد. بدین ترتیب، عامل در هر مرحله با یک مسأله ساده مواجه است، آن را می‌آموزد و سپس وارد مرحله بعدی آموزش می‌شود (۳). در واقع، این همان روشی است که انسان کاری را یاد می‌گیرد (برای مثال برای عمل راه رفتن). ابتدا غلت می‌زند، سپس خزیدن و ایستادن را تجربه می‌کند و در نهایت، راه رفتن را می‌آموزد و با این نوع از

۱- دانشجوی دکتری تخصصی، گروه هوش مصنوعی، دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران
۲- استادیار، گروه مهندسی نرم‌افزار، دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران
۳- دانشیار، گروه مهندسی نرم‌افزار، دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران
نویسنده مسؤول: مرتضی دری‌گیو؛ استادیار، گروه مهندسی نرم‌افزار، دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران
Email: dorrigiv@semnan.ac.ir

آموزش الگوریتم یادگیری، می‌تواند از مفاهیم اساسی که از مرحله قبل آموخته است، بهره ببرد تا عمل‌های سطح بالاتر (High level) را راحت‌تر آموزش ببیند.

یادگیری تقویتی عمیق، استفاده از شبکه عصبی (Neural network) به عنوان تابع تخمین زننده یادگیری تقویتی است (۶). ایده ترکیب شبکه عصبی و یادگیری تقویتی، تاریخچه طولانی دارد و توسط Tesauro در اوایل سال ۱۹۹۰ با استفاده از شبکه عصبی در بازی به عنوان تابع تخمین زننده توسعه یافت و در سطح بهترین بازیکن انسانی نمایش داده شد (۷). شبکه عصبی مدت زمان طولانی در سیستم‌های شناسایی و کنترل استفاده می‌شد (۸). لازم به ذکر است که پس از گذشت دو دهه از نتایج Tesauro (۷)، یادگیری تقویتی با تابع تخمین زننده غیر خطی هنوز تا حدودی مبهم مانده است. انفجار در استفاده از یادگیری تقویتی عمیق، بعد از موفقیت در نتایج پژوهش Mnih و همکاران بود که نشان داد کامپیوتر می‌تواند با ورودی تصویر، بازی‌های آتاری را یاد بگیرد (۹). پس از آن تحقیقات جالب بسیاری در این زمینه شروع شد.

در یادگیری با برنامه درسی، نمونه‌های آموزش باید بر اساس دشواری، به چندین مجموعه تقسیم‌بندی شوند که این موضوع به دانش در حوزه تحقیق نیاز دارد (۴). روش یادگیری «کودک گام» در مطالعه Bengio و همکاران پیشنهاد شد که یادگیری با برنامه درسی را بهبود می‌بخشد. با این وجود، یادگیری با برنامه درسی و یادگیری کودک گام، نیاز به پیش‌پردازش نمونه‌های آموزش به صورت دستی قبل از شروع آموزش دارد (۴).

در پژوهش دیگری از شبکه عصبی Convolutional Neural Network (یا CNN) به همراه استراتژی یادگیری با برنامه درسی در طبقه‌بندی عکس‌های ماموگرافی استفاده گردید (۱۰). به طور خاص، ابتدا طبقه‌بندی (Classifier) بر روی تصاویر ضایعات در ماموگرافی‌ها آموزش داده شد و سپس با استفاده از ویژگی‌های آموخته شده، یک مدل مبتنی بر اسکن ارایه گردید. همچنین، از یادگیری با مرحله درسی و الگوریتم مورد استفاده در AlphaGo استفاده شد (۱۱) تا بتواند بازی Gomoku را در سطح انسانی اجرا کند (۱۲). استفاده از یادگیری تقویتی Actor-Critic با برنامه درسی در محیط بازی، در بازی اول شخص تیراندازی (First person shooter) نتایج امیدوارکننده‌ای داشت؛ به طوری که عامل هوشمند توانست در محیط سه بعدی بازی، از تاکتیک‌های جدیدی استفاده کند (۱۳). هدف از انجام مطالعه حاضر، استفاده از یادگیری تقویتی با برنامه درسی در محیط بازی‌های استراتژیک بود.

مواد و روش‌ها

تکنیک یادگیری عمیق، نتایج بسیار امیدوارکننده‌ای را در شناسایی شیء (۱۴) و تشخیص گفتار (۱۵) نشان داده است. یکی از چالش‌ها در یادگیری تقویتی، نداشتن دسترسی آماری به تابع در حال بهینه‌سازی می‌باشد (مجموع پاداش پیش‌بینی شده عامل). هدف عامل، وابسته به مدل پویایی است که گاهی در فضای مسأله ناشناخته است. همچنین، داده‌های ورودی به الگوریتم، به رفتار عامل در محیط وابستگی دارد و بنابراین، نمی‌توان الگوریتمی ارایه داد که بهبود یکنواختی داشته باشد. در مسایل پیچیده‌تر، گاهی به جای یک تابع، تعدادی تابع تقریبی مختلف وجود دارد. آموزش عامل یادگیری تقویتی عمیق با برنامه درسی، می‌تواند در آموزش عمل‌های پیچیده به عامل کمک کند.

یک برنامه درسی می‌تواند دنباله‌ای وزن‌دار از درس‌های آموزش باشد. ابتدا وزن نمونه‌های آموزش به گونه‌ای است که درس‌ها را آسان‌تر می‌کند. به عبارت دیگر، نمونه‌هایی آموزش داده می‌شوند که ساده‌ترین مفاهیم را در خود دارند و می‌توان آن‌ها را به راحتی یاد گرفت. در پایان دنباله، وزن نمونه‌ها یکنواخت است و آموزش با تمام نمونه‌ها انجام می‌شود. فرض بر این است که z یک متغیر تصادفی از مجموعه نمونه‌های آموزش [جفت (x,y) برای یادگیری با نظارت] و P (z) توزیع نمونه آموزش باشد که عامل در نهایت، باید تابعی از آن را یاد بگیرد. همچنین، اگر $0 \le W_\lambda(z) \le 1$ وزنی باشد که برای نمونه z در مرحله λ برنامه درسی اعمال می‌شود، با $0 \le \lambda \le 1$ و $W_\lambda(z) = 1$، رابطه ۱ برقرار است.

$$\forall z \cdot Q_\lambda(z) \propto W_\lambda(z) P(z) \qquad \text{رابطه ۱}$$

و اگر $\int Q_\lambda(z)dz = 1$ باشد، رابطه ۲ برقرار است.

$$\forall z \cdot Q_1(z) = P(z) \qquad \text{رابطه ۲}$$

در یک دنباله افزایش یکنواخت که مقدار λ از صفر تا ۱ افزایش می‌یابد، باید آنتروپی افزایش پیدا کند تا تنوع نمونه‌های آموزش نیز افزایش یابد و وزن نمونه‌ها با افزوده شدن به مجموعه تمرین زیاد شود. در هر فضای مسأله، تعریف آسانی و سختی نمونه آموزش متفاوت است (۴).

بهینه‌سازی خط‌مشی نزدیک مبدأ (Proximal policy optimization یا PPO)

الگوریتم مورد استفاده در یادگیری تقویتی عمیق، بهینه‌سازی خط‌مشی نزدیک مبدأ (۱۵) می‌باشد. این روش تغییر یافته و اصلاح شده روش ناحیه اطمینان است و هدف هر دو روش، بیشینه کردن تابع جایگزین با محدودیت در مقدار به‌روزرسانی خط‌مشی است. PPO از هدف تقطیع شده (Clipped objective) استفاده می‌کند تا به صورت ابتکاری واگرایی Kullback-Leibler divergence (KL-divergence) را محدود نماید.

در رابطه ۳، $\rho_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ و ε هایپرپارامترها هستند. زمانی که A_t مثبت باشد، تابع با $1+\varepsilon$ و زمانی که A_t منفی باشد، تابع با $1-\varepsilon$ تقطیع می‌شود. علاوه بر این، L^{PPO} تغییراتی که موجب بهبود هدف می‌شود را در نظر نمی‌گیرد و تغییراتی که باعث بدتر شدن هدف می‌شود را در نظر می‌گیرد (۱۵).

$$\max_\theta L^{PPO}(\theta),\ L^{PPO}(\theta) = E[\min(\rho_t A_t, clip(\rho_t, 1-\varepsilon, 1+\varepsilon) A_t)] \qquad \text{رابطه ۳}$$

بستر آزمایش

غار اژدها: غار اژدها یک بازی دو بعدی استراتژیک می‌باشد که با استفاده از موتور بازی Unity و توسط یکی از نویسندگان این مقاله ساخته شد و در فروشگاه نرم‌افزار «بازار» قابل دریافت است (۱۶). در این بازی ۴ اژدهای متفاوت با توان‌های مختلف به مقابله با ۹ دشمن خود می‌پردازند تا از غارشان دفاع کنند. هر اژدها جان و جادوهای (Spell) مربوط به خودش را دارد که از این بین تنها می‌تواند شش جادو را در دِک (دسته جادوها) خود جایگذاری کند. بازی ۵ مسیر دارد که در آن‌ها دشمنان به سمت غار حرکت می‌کنند و اژدها با جابه‌جایی و حرکت در این مسیرها، جادوی خود را به سمت آن‌ها پرتاب می‌کند و آن‌ها را از بین می‌برد. بازیکن زمانی که بتواند ۸۰ ثانیه جلوی دشمنان ایستادگی کند، پیروز بازی خواهد شد. در شکل ۱ نمایی از محیط بازی غار اژدها مشاهده می‌شود.

شکل ۱. نمایی از محیط بازی غار اژدها

توانایی‌های اژدها: اژدها در بازی می‌تواند اعمال خاصی را با داشتن نیروهایی انجام دهد (به طور مثال جان دشمنان خود را کم کند یا جان خود را زیاد کند و...) که این نیروها در بازی جادو نامگذاری شده‌اند. هر اژدها می‌تواند در هر لحظه از زمان از یک جادو استفاده کند. به عبارت دیگر، شمای کلی سیستم اژدها مانند شکل ۲ است. هر جادو ویژگی‌های متفاوتی دارد. ویژگی مشترک جادوها داشتن زمان آماده‌سازی است؛ به گونه‌ای که بعد از استفاده از هر جادو، مدتی برای آماده‌سازی و استفاده مجدد زمان خواهد برد. لیست کامل جادوهای اژدها در جدول ۱ نشان داده شده است.

شکل ۲. نمای کلی جادوی‌های اژدها

نیروهای مختلف هر اژدها تأثیر متفاوتی را بر محیط بازی (از جمله دشمنان و اژدها) می‌گذارد. به عنوان مثال، استفاده از نیرو می‌تواند به دشمنان آسیب برساند یا سرعت حرکت آن‌ها را آرام کند و یا میزان جان و قدرت اژدها را افزایش دهد و یا برج‌های دفاعی در بازی به وجود آورد. هر بازیکن در بازی می‌تواند کاراکتر خود را در مسیرهای مختلف جابه‌جا کند. همان‌طور که در شکل ۳ نمایی از مسیرهای پنج‌گانه مشاهده می‌شود، هر نقشه بازی شامل ۵ مسیر افقی است که بازیکن با لمس هر کدام از مسیرها می‌تواند بدون گرفتن زمان به آن مسیر برود. دشمنان از هر مسیر به صورت تصادفی به سمت بازیکن حرکت می‌کنند و اژدها برای استفاده از بیشتر نیروهایش باید در مسیری که دشمن مورد نظر در آن قرار دارد، حرکت کند. حضور در مسیری که دشمن در آن وجود دارد، از قوانین پایه‌ای بازی است.

شکل ۳. نمایی از مسیرهای پنج‌گانه

دشمنان: دشمنان در ۹ مدل مختلف با اژدها روبه‌رو می‌شوند و در ۵ مسیر مختلف به صورت تصادفی به سمت اژدها حرکت خواهند کرد و در میزان آسیب در ثانیه، سرعت حرکت و میزان جان با یکدیگر تفاوت دارند. دشمنان مشاهده اصلی بازیکن در حین بازی هستند. در جدول ۲، دشمنان و ویژگی‌هایشان مشخص شده است.

رقابت برخط: در بخش برخط (Online) بازی که بازیکن از مرحله ۳۰ به آن وارد می‌شود، دو اژدها با یکدیگر به رقابت می‌پردازند و هر کدام که بهتر از غار خود دفاع کند، برنده این بخش خواهد بود. دو کاراکتر اژدها که یکی توسط عامل هوشمند و دیگری توسط عامل انسانی کنترل می‌شوند، در مقابل هم قرار می‌گیرند. اعمال قابل انجام توسط بازیکنان شامل حرکت عمودی در خطوط و اجرای عملی موسوم به نیرو می‌باشد. در این بین، ماشین باید تصمیم بگیرد که چطور در مسیرها جابه‌جا شود و هر جادو را به کدام مدل از دشمنان خود پرتاب کند تا هم رفتاری واقعی و انسان‌نما داشته باشد و هم حرکاتش معقولانه به نظر برسد.

جدول ۱. جادوهای بازی غار اژدها

نام جادو	تصویر	میزان آسیب	میزان اکسیر مصرفی	مدت زمان آماده‌سازی
توپ آتشین		به اندازه ۱۲۰ درصد از قدرت اژدها به دشمن آسیب می‌رساند.	۲۰	۱/۵
گدازه		به اندازه ۱۸۰ درصد از قدرت اژدها به دشمن آسیب می‌رساند.	۴۰	۵
انفجار آتشین		به اندازه ۱۵۰ درصد از قدرت اژدها به دشمن آسیب می‌رساند.	۵۰	۳۰
آتشدان		۱۰۰۰ عدد از ضربات مرگبار دو نیروی توپ آتشین و گدازه را در خود ذخیره می‌کند و بعد از برخورد با دشمن به همان اندازه به دشمن آسیب می‌رساند.	۸۰	۰
طوفان اژدها		بعد از برخورد با دشمن، دشمن را می‌کشد و به اندازه ۱۰ درصد از جانش به دشمنان پشت سر آسیب می‌رساند.	۸۰	۳۰
برج آتش		اژدها ۵ برج آتش ظاهر می‌کند که برای مدت ۱۵ ثانیه از غار اژدها محافظت می‌کنند.	۱۲۰	۱۲۰
حضور ذهن		زمان آماده‌سازی تمام نیروهای اژدها را صفر می‌کند.	۶۰	۹۰

جدول ۲. دشمنان در بازی غار اژدها

نام دشمنان در بازی	تصویر	میزان جان	میزان آسیب در ثانیه	سرعت حرکت در ثانیه
موذی		۶۰۰	۴۱	۲۰
سپردار کوچک		۱۵۰۰	۷۱	۲۴
شکارچی		۱۰۵۰	۱۰۱	۸
نیزه‌دار		۱۲۰۰	۱۰۱	۱۳
منجنیق		۹۰۰	۲۵۱	۳
سپردار بزرگ		۲۴۰۰	۱۳۱	۱۵
مزدور		۱۳۵۰	۹۱	۲۸
گنده‌بگ		۳۰۰۰	۱۲۱	۱۳
جادوگر		۹۰۰	۱۲۱	۱۲

یافته‌ها

الگوریتم آموزش عامل: عامل در فضای بازی در حال مشاهده و تصمیم‌گیری می‌باشد. وظیفه عامل، استفاده از جادوهای اژدها در مقابل دشمن است. آموزش عامل برای یک نوع از اژدها انجام شده و برای ساده‌سازی آموزش اژدها با ۴ نیرو در نظر گرفته شده است. مسیر موفقیت در یادگیری تقویتی واضح و مشخص نیست و الگوریتم متغیرها و اجزای در حال تغییر زیادی دارد که موجب می‌شود رفع خطا کردن را مشکل سازد و تلاش زیادی برای تنظیم این اجزا به منظور رسیدن به نتایج خوب مورد نیاز است.

PPO توازن بین سهولت اجرا، پیچیدگی نمونه و سهولت تنظیم و تلاش برای محاسبه یک به‌روزرسانی در هر مرحله که تابع هزینه را به حداقل می‌رساند؛ در حالی که اطمینان از انحراف از خط‌مشی قبلی نسبتاً کوچک می‌باشد. به همین دلیل از این الگوریتم برای آموزش عامل استفاده شده است. همچنین، از برنامه درسی برای آموزش عامل استفاده گردید؛ به گونه‌ای که عامل ابتدا عمل‌های ساده را آموزش ببیند و سپس عمل‌های سخت‌تر به عامل آموزش داده شود.

مشاهدات: بردار مشاهده عامل شامل ۷۹ ویژگی است. مشاهدات عبارتند از در کدام مسیر دشمن حضور دارد؟، برای هر ۴ جادوی اژدها: آیا نیرو آماده استفاده است؟، میزان آسیب نیرو نرمال شده در بازه [۰، ۱] (بیشترین میزانی که جادو می‌تواند اعمال کند، ۲۰۰۰ است و همه جادوها میزان آسیبشان در بازه صفر و ۱ بر اساس بیشترین آسیب نرمال شده‌اند)، برای ۶ دشمن نزدیک به اژدها: میزان جان دشمن نرمال شده در بازه [۰، ۱]، نوع دشمن (۹ دشمن متفاوت در بازی وجود دارد)، آیا دشمن در حال ضربه زدن است یا خیر؟ عامل، وظیفه استفاده از جادوهای اژدها را بر عهده دارد، اعمال عبارت از استفاده از نیروی ۱، استفاده از نیروی ۲، استفاده از نیروی ۳، استفاده از نیروی ۴، استفاده نکردن از نیرو بود.

تابع پاداش: تابع پاداش به گونه‌ای طراحی شده است که عامل به سمت ضربه زدن به دشمنان سوق داده شود. همچنین، از هدف کلی سیستم که پیروز شدن بازی است، پاداش خواهد گرفت. الگوریتم ۱ تابع پاداش را نشان می‌دهد.

الگوریتم ۱: تابع پاداش

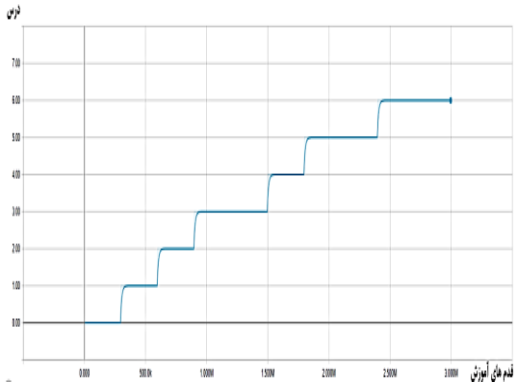
```
if        Player Cast Spell
if !any Enemy Exis in Same Lane
                    Punish()
if Player Select Not Ready Spell
                    Punish()
if Die
                    Punish()
if Win
                    Punish()
if Spell hit Enemy
                    Punish()
if Enemy Attacking
                    Punish()
```

اگر بازیکن پیروز شود یا جادو باعث کاهش جان دشمن شود، پاداش می‌گیرد. اگر بازیکن در زمان استفاده از جادو در مسیر دشمن قرار نداشته باشد، مجازات می‌شود. علاوه بر این، اگر بازیکن ببازد یا دشمن در حال آسیب زدن باشد، عامل مجازات می‌شود.

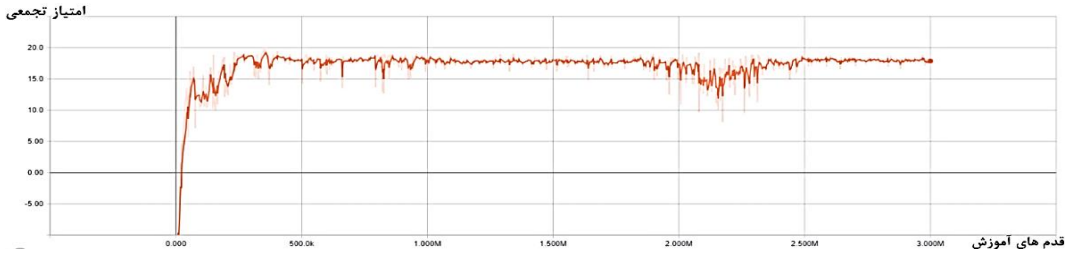
طراحی برنامه درسی: ابتدا عامل در محیط ساده آموزش داده می‌شود. منظور از محیط ساده در بازی غار اژدها، مواجه شدن با دشمنان محدودتر است. عامل باید نحوه مقابله با ۹ دشمن متفاوت خود را آموزش ببیند. به عبارت دیگر، باید نگاشت مناسبی بین هر دشمن و جادوهایش داشته باشد و به دست آوردن این دانش که در برابر هر دشمن چه جادویی استفاده شود، عامل اصلی در برتری یک بازیکن است. به عنوان مثال، جادوی طوفان اژدها می‌تواند اولین دشمن در مسیر را نابود کند. اگر این جادو در مقابل دشمنی که بیشترین جان را دارد استفاده شود، بهترین بهره‌وری از آن جادو است و اگر در برابر ضعیف‌ترین دشمن استفاده شود، ممکن است موجب باخت بازیکن شود.

آموختن این امر مستلزم آن است که عامل ابتدا نحوه مقابله با دشمن ضعیف‌تر را بیاموزد. این دانش سطح پایین عامل را به سمت بهینه محلی حرکت می‌دهد و مسیر آموزش را به سمت یادگیری عمل‌های سطح بالاتر تسهیل می‌کند. آموزش به میزان ۳ میلیون قدم و طراحی مراحل درسی به صورت زیر اعمال شد. ابتدا دشمنان به ترتیب قدرت مرتب شدند و ۷ برنامه درسی طراحی گردید. طراحی این مراحل به گونه‌ای بود که عامل قدم به قدم با دشمنان دارای نیروهای متفاوت مقابله می‌کرد و با وزن‌های آموخته از برنامه درسی قبلی با دشمنان جدید روبه‌رو می‌شد.

در شکل ۴ مراحل افزایش مرحله درسی در قدم‌های آموزش نشان داده شده است.



شکل ۴. مراحل افزایش مرحله درسی در قدم‌های آموزش



شکل ۵. پاداش تجمعی در حین آموزش بدون برنامه درسی

۱۰ درصد ابتدای قدم‌های آموزش: مواجه شدن عامل با دو نوع از دشمنان (دشمنان ضعیف‌تر)
۱۰ درصد دوم آموزش: مواجه شدن عامل با سه نوع از دشمنان
۱۰ درصد سوم آموزش: مواجه شدن عامل با چهار نوع از دشمنان
۲۰ درصد چهارم آموزش: مواجه شدن عامل با پنج نوع از دشمنان
۱۰ درصد پنجم آموزش: مواجه شدن عامل با شش نوع از دشمنان
۲۰ درصد ششم آموزش: مواجه شدن عامل با هفت نوع از دشمنان
۲۰ درصد نهایی آموزش: مواجه شدن عامل با نه نوع از دشمنان

آموزش عامل هوشمند: در شکل‌های ۵ و ۶ میزان پاداش تجمعی عامل در حین آموزش نشان داده شده است. دو عامل با شاخص‌ها و الگوریتم‌های یکسان آموزش دیده‌اند (یک عامل در فضای پیچیده بازی و دیگری با برنامه درسی). همان‌گونه که مشاهده می‌شود، عامل بدون برنامه درسی توانست در پایان آموزش ۱۸ امتیاز از بازی به دست آورد که این موضوع برای عامل با برنامه درسی به ۲۶ امتیاز افزایش یافته است. نتایج نشان داد که این روش باعث افزایش سرعت و کیفیت یادگیری عامل می‌گردد. در واقع، عامل با برنامه درسی توانست امتیاز بیشتری از فضای بازی کسب کند و حرکت سریع‌تری به سمت بهینه محلی داشته باشد.

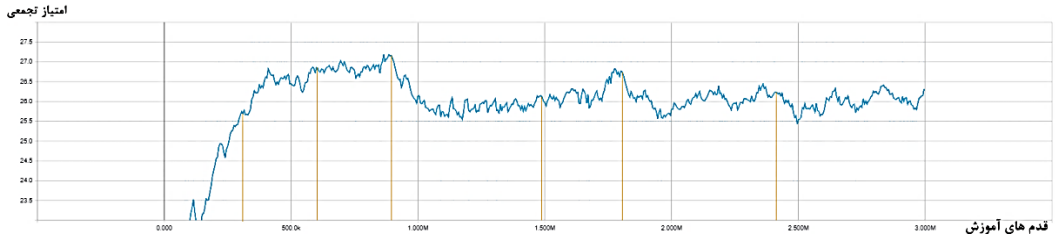
بحث

پژوهش حاضر با هدف بررسی موفقیت یادگیری تقویتی عمیق با برنامه درسی برای آموزش عامل هوشمند در محیط بازی غار اژدها انجام شد. به نظر می‌رسد که این روش باعث افزایش سرعت و کیفیت یادگیری عامل گردید. در واقع، عامل با برنامه درسی توانست امتیاز بیشتری از فضای بازی کسب کند و حرکت سریع‌تری به سمت بهینه محلی داشته باشد. در قدم‌های آموزش، زمانی که برنامه درسی جدید به فضای آموزش اضافه می‌شود، امتیاز عامل افت می‌کند؛ چون عامل، شناختی از مسأله جدید ندارد. با استفاده از روش یادگیری تقویتی عمیق با برنامه درسی، عامل خیلی سریع با استفاده از دانش به دست آورده در مرحله قبل، عمل جدید را می‌آموزد؛ در حالی که بدون برنامه درسی، عامل دشواری زیادی در یادگیری عمل‌های مورد آموزش دارد.

موضوع مطالعه حاضر تا حدودی نو و بدیع می‌باشد و در مقایسه با موضوعات دیگر، کمتر به آن پرداخته شده است. به عنوان نمونه، Sukhbaatar و همکاران چارچوبی برای یادگیری خودکار برنامه درسی از طریق بازی نامتقارن با خود (Asymmetric Self-Play) پیشنهاد کردند. دو عامل به نام‌های آلیس و باب با اهداف مختلف، وظیفه یکسانی داشتند. آلیس، باب را برای اجرای عملی به چالش می‌کشد و باب تلاش می‌کند تا آن را هرچه سریع‌تر انجام دهد. بدین ترتیب، تعامل بین آلیس و باب به طور خودکار برنامه درسی شامل وظایف سخت‌تر و چالش‌برانگیز ایجاد می‌کرد (۱۷).

Justesen و همکاران از یادگیری تقویتی در طراحی محتوای بازی استفاده کردند و نشان دادند که یادگیری تقویتی در بعضی از بازی‌های خاص می‌تواند بیش‌برازش (Overfit) شود، اما به کمک برنامه درسی می‌توان باعث تعمیم خط‌مشی آموزشی شد تا بتواند محتوای بازی را در سطح انسانی طراحی کند (۱۸).

در این پژوهش به کمک بهینه‌سازی نزدیک خط‌مشی، در بازی اول شخص تیراندازی، دو عامل هوشمند در محیط‌های متفاوت آموزش داده شد؛ یکی در محیطی با برنامه درسی و دیگری در محیط بدون برنامه درسی. سپس هر دو در محیط یکسانی شروع به بازی کردند که نتایج حاکی از کیفیت بهتر عامل با برنامه درسی از نظر امتیاز به دست آورده بود. در این موارد به طور عمده آزمایش بر بازی‌هایی با محیط‌های ثابت و یا در محیط‌هایی غیر از بازی متمرکز بوده است؛ در حالی که پژوهش حاضر توسط تیم تحقیقاتی بر روی بازی‌های استراتژیک انجام شد که محیط پیچیده‌تری نسبت به بازی با محیط ثابت داشت.



شکل ۶. میزان پاداش تجمعی عامل یادگیری با مراحل درسی (خط‌های نارنجی زمان روبه‌رو شدن عامل با برنامه درسی جدید است)

محدودیت‌ها

سیستم ضعیف کامپیوتری، از جمله محدودیت‌های مطالعه حاضر بود که مدت آموزش عامل به اندازه ۳ میلیون قدم اکتفا شد. از آن‌جایی که پژوهش حاضر اولین مطالعه‌ای بود که بر روی بازی غار اژدها انجام گرفته است، امکان مقایسه با کارهای تحقیقی دیگر وجود نداشت.

پیشنهادها

پیشنهاد می‌شود پژوهش حاضر در سیستم کامپیوتری قوی‌تر، تعداد قدم‌های بیشتر و آموزش عامل هوشمند انجام شود و در روال آموزش با گرفتن حد امتیاز مطلوب، از هر مرحله درسی عبور کند و این امتیاز مطلوب، معیار یاد گرفتن آن عمل در آن درس باشد. بنابراین، برنامه درسی با رسیدن عامل به امتیاز تغییر کند. در مطالعه حاضر، نحوه افزایش برنامه درسی با تعداد قدم‌های آموزش است که شاید در بعضی از درس‌ها معیار مناسبی نباشد. علاوه بر این، می‌توان از ایده مطرح شده در این تحقیق در فضای بازی‌های استراتژیک دیگر نیز استفاده نمود.

نتیجه‌گیری

با توجه به نتایج به دست آمده، فرض اول مبنی بر این که یادگیری تقویتی عمیق با برنامه درسی در فضای پیچیده بازی‌های استراتژیک به عامل هوشمند کمک می‌کند تا تعمیم (Generalize) بهتر و سریع‌تری در یادگیری داشته باشد، تأیید شد.

تشکر و قدردانی

بدین وسیله از شرکت سرگرمی‌سازان آسمان امید برای در اختیار قرار دادن بازی مورد نظر تشکر و قدردانی به عمل می‌آید.

مقاله حاضر از میان مقالات ارسال شده به دبیرخانه پنجمین کنفرانس بین‌المللی «بازی‌های رایانه‌ای؛ فرصت‌ها و چالش‌ها» با نگاه ویژه به بازی‌های درمانی (بهمن ماه ۱۳۹۸، اصفهان)، از سوی هیأت تحریریه مجله پژوهش در علوم توان‌بخشی مورد تقدیر قرار گرفت. بدین وسیله نویسندگان از پژوهشگاه فضای مجازی مرکز ملی فضای مجازی به جهت حمایت از انتشار این مقاله قدردانی به عمل می‌آورند. همچنین، از مرکز نوآوری صنایع سرگرمی دانشگاه اصفهان که در جمع‌آوری داده‌ها و به ثمر رسیدن این پروژه نقش مهمی داشتند، سپاسگزاری می‌گردد.

نقش نویسندگان

محمدرضا محمدنژاد، طراحی و ایده‌پردازی مطالعه، خدمات پشتیبانی و اجرایی و علمی مطالعه، تحلیل و تفسیر نتایج، تنظیم دست‌نوشته، ارزیابی تخصصی دست‌نوشته از نظر مفاهیم علمی، تأیید دست‌نوشته نهایی جهت ارسال به دفتر مجله، مسؤولیت حفظ یکپارچگی فرایند انجام مطالعه از آغاز تا انتشار و پاسخگویی به نظرات داوران، مرتضی دری‌گیو، تحلیل و تفسیر نتایج، تنظیم دست‌نوشته، ارزیابی تخصصی دست‌نوشته از نظر مفاهیم علمی، تأیید دست‌نوشته نهایی جهت ارسال به دفتر مجله، مسؤولیت حفظ یکپارچگی فرایند انجام مطالعه از آغاز تا انتشار و پاسخگویی به نظرات داوران، فرزین یغمایی، تحلیل و تفسیر نتایج، تنظیم دست‌نوشته، ارزیابی تخصصی دست‌نوشته از نظر مفاهیم علمی، تأیید دست‌نوشته نهایی جهت ارسال به دفتر مجله و مسؤولیت حفظ یکپارچگی فرایند انجام مطالعه از آغاز تا انتشار و پاسخگویی به نظرات داوران، را بر عهده داشتند.

منابع مالی

پژوهش حاضر تحت حمایت مالی دانشگاه سمنان انجام گردید. این دانشگاه در جمع‌آوری داده‌ها، تحلیل و گزارش آن‌ها، تنظیم دست‌نوشته و تأیید نهایی مقاله برای انتشار اعمال نظر نداشته است. بررسی و انتشار تحقیق حاضر در مجله پژوهش در علوم توان‌بخشی، با حمایت مالی پژوهشگاه فضای مجازی مرکز ملی فضای مجازی، حامی پنجمین همایش بین‌المللی بازی‌های کامپیوتری با رویکرد بازی‌های درمانی صورت گرفت. این پژوهشگاه در طراحی، تدوین و گزارش این مطالعه نقشی نداشت.

تعارض منافع

نویسندگان دارای تعارض منافع نمی‌باشند.

References

1. Arulraj JP. Adaptive agent generation using machine learning for dynamic difficulty adjustment. Proceedings of the 2010 International Conference on Computer and Communication Technology (ICCCT). 2019 Sep 17-19; Allahabad, Uttar Pradesh, India. p. 746-51.
2. Mohammadnejad M, Yaghmaee F. Design of Intelligent agent with deep reinforcement learning in game enviroment. Proceedings of the 4th National and 2nd International Conference on Computer Games, Challenge and Opportunities; 2019 Feb 21; Kashan, Iran. p. 1-16. [In Persian].
3. Wu Y, Tian Y. Training Agent for First-Person Shooter Game with Actor-Critic Curriculum Learning. Proceedings of the International Conference on Learning Representations, ICLR 2017; 2017 Apr 24-26; Toulon, France. p. 1-10.
4. Bengio Y, Louradour J, Collobert R, Weston J. Curriculum learning. Proceedings of the 26th Annual International Conference on Machine Learning (ICML 2009); 2009 Jun 14-18; Montreal, Canada. p. 41-8.
5. Gong C, Tao D, Maybank SJ, Liu W, Kang G, Yang J. Multi-modal curriculum learning for semi-supervised image classification. IEEE T Image Process 2016; 25(7): 3249-60.
6. Francois-Lavet V, Henderson P, Islam R, Bellemare MG, Pineau J. An introduction to deep reinforcement learning. Foundations and Trends in Machine Learning 2018; 11(3-4): 219-354.
7. Tesauro G. Temporal difference learning and TD-Gammon. Communications of the ACM 1995; 38(3): 58-68.

8. Narendra KS, Parthasarathy K. Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks 1990; 1(1): 4-27.

9. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M. Playing Atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602. 2013.

10. Lotter W, Sorensen G, Cox D. A Multi-scale CNN and Curriculum Learning Strategy for Mammogram Classification. Cham, Switzerland: Springer International Publishing; 2017 p. 169-77.

11. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature 2016; 529(7587): 484-9.

12. Xie Z, Fu X, Yu J. AlphaGomoku: An AlphaGo-based Gomoku Artificial Intelligence using Curriculum Learning. arXiv, abs/1809.10595. 2018

13. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM 2012; 60 (6): 1097–1105.

14. Dahl GE, Yu D, Deng L, Acero A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing 2012; 20(1): 30-42.

15. Tuan YL, Zhang J, Li Y, Lee HY. Proximal policy optimization and its dynamic version for sequence generation. arXiv: 1808.07982. 2018.

16. Mohammadnejad M. Dragon Cave, a strategy game [Online]. [cited 2020 Feb 20]; Available from: URL: https://cafebazaar.ir/app/ir.sinsin.DragonCave.v_0/?l=en, developed by M. Mohammadnejad

17. Sukhbaatar S, Lin Z, Kostrikov I, Synnaeve G, Szlam A, Fergus R. Intrinsic motivation and automatic curricula via asymmetric self-play. 2018. Proceedings of the 6th International Conference on Learning Representations, ICLR 2018; 2018 Apr 30-May 3; Vancouver, Canada.

18. Justesen N, Torrado RR, Bontrager P, Khalifa A, Togelius J, Risi S. Illuminating Generalization in Deep Reinforcement Learning through Procedural Level Generation. arXiv: 1806.10729 [cs.LG]. 2018.

DOI: 10.22122/jrrs.v15i1.3446

Published by Vesnu Publications

Improvement of Deep Reinforcement Learning Using Curriculum in Game Environment

Mohammadreza Mohammadnejad[1], Morteza Dorri-Giv[2], Farzin Yaghmaee[3]

Original Article

Abstract

Introduction: Training deep curriculum learning is a kind of smart agent training in which, first the simple acts, and then, the difficult acts are trained to smart agent. In this study, we proposed a new framework for training deep curriculum learning to defense-based game in particular Dragon Cave.

Materials and Methods: Deep reinforcement learning approach with curriculum learning was used to train an intelligent agent in the game Dragon Cave. Curriculum learning paradigm started from simple tasks, and then gradually tried harder ones. Using Proximal Policy Optimization, the intelligent agents were trained in various environments, once in a curriculum-learning environment, and once in an environment without curriculum learning. Then, they started the game in the same environment.

Results: The improvement of the agent was observed with deep curriculum reinforcement learning.

Conclusion: It seems that the deep curriculum reinforcement learning increases the rate and the quality of intelligent agent training in complex environment of strategic games.

Keywords: Intelligent agent, Deep reinforcement with curriculum learning, Machine learning, Neural network

Citation: Mohammadnejad M, Dorri-Giv M, Yaghmaee F. **Improvement of Deep Reinforcement Learning Using Curriculum in Game Environment.** J Res Rehabil Sci 2019; 15(1): 50-7.

Received: 09.02.2019 Accepted: 11.03.2019 Published: 04.04.2019

1- PhD Student, Department of Artificial Intelligence, School of Electrical and Computer Engineering, University of Semnan, Semnana, Iran
2- Assistant Professor, Department of Software Engineering, School of Electrical and Computer Engineering, University of Semnan, Semnana, Iran
3- Assistant Professor, Department of Software Engineering, School of Electrical and Computer Engineering, University of Semnan, Semnana, Iran
Corresponding Author: Morteza Dorri-Giv; Assistant Professor, Department of Software Engineering, School of Electrical and Computer Engineering, University of Semnan, Semnana, Iran; Email: dorrigiv@semnan.ac.ir