Iranian Journal of Wargaming
AJA Command and Staff University
http://www.ijwg.ir/
Online ISSN: 2676-7708
Print ISSN: 2676-5284

An Approach to First-Person Shooter Games Using Deep Reinforcement Learning

Jafar Saadati Razian[1], Akbar Asgharzadeh-Bonab[2]✉, Alireza Mohamadi[3], Saber Jabarzadeh[4]

1. Lecturer, Imam Ali Officer University, Tehran, Iran. E-mail: jsfarmandeh@gmail.com
2. Corresponding Author, Department of science and technology studies, AJA Command and Staff University, Tehran, Iran. Email: a.asgharzadeh@Urmia.ac.ir
3. Department of science and technology studies, AJA Command and Staff University,Tehran, Iran. Email: arm3stars@gmail.com
4. Department of science and technology studies, AJA Command and Staff University,Tehran, Iran. Email: saber.jabarzadeh@gmail.com

Article Info

Article type:
Research Article

Article history:
Received 23 April 2024
Received in revised form 2 July 2024
Accepted 29 july 2024

Keywords:
Deep learning, AI War Game, Deep reinforcement learning,

ABSTRACT

Objective: The main objective of this research is to investigate and improve the performance of Double Deep Q-Learning Network models in first-person shooter games with a focus on intelligent competition.

Methodology: In this research, Deep Q-Network (DQN) and Deep Double Deep Q-Network (DDQN) models are used for the game War. First, the DQN and DDQN models are evaluated, and then their performance is improved using the Prioritized Experience Replay (PER) method. Three experimental game environments are used to evaluate and assess the models.

Findings: The findings of this research show that the proposed Double Deep Q-Network architecture with the Prioritized Experience Replay method has better performance than other proposed algorithms in this field.

Conclusion: The use of the Prioritized Experience Replay method in reinforcement learning has significant advantages that lead to improved performance of the artificial intelligence agent. This method, by utilizing high-quality data and experiences, focuses specifically on more informative experiences, thus significantly increasing sampling efficiency.

Cite this article: Saadati Razian, J. Asgharzadeh-Bonab, A. mohamadi, A. & jabarzadeh, S. (2024). An Approach to First-Person Shooter Games Using Deep Reinforcement Learning. Iranian Journal of Wargaming, 6 (13), 61- 92.

DOI: 10.22034/ijwg.2024.453676.1083

Publisher: Command and Staff University

دوفصلنامه علمی بازی جنگ
دانشگاه فرماندهی و ستاد آجا
http://www.ijwg.ir/
شاپای چاپی: ۵۲۸۴-۲۶۷۶ شاپای الکترونیکی: ۷۷۰۸-۲۶۷۶

رویکرد نوین دربازی‌های اول‌شخص تیرانداز با استفاده از یادگیری تقویتی عمیق

جعفر سعادتی رازیان [1] اکبر اصغرزاده [2]✉ علیرضا محمدی [3] صابر جبارزاده [4]

1. مدرس دانشگاه افسری امام علی (ع)، تهران، ایران. رایانامه: jsfarmandeh@gmail.com
2. گروه مطالعات علم و فناوری، دانشگاه فرماندهی و ستاد آجا، تهران، ایران، رایانامه: a.asgharzadeh@urmia.ac.ir
3. گروه مطالعات علم و فناوری، دانشگاه فرماندهی و ستاد آجا، تهران، ایران، رایانامه: arm3stars@gmail.com
4. گروه مطالعات علم و فناوری، دانشگاه فرماندهی و ستاد آجا، تهران، ایران، رایانامه: saberjbr@gmail.com

اطلاعات مقاله

نوع مقاله:
مقالهٔ پژوهشی

تاریخ دریافت:
1403/02/04

تاریخ بازنگری:
1403/04/12

تاریخ پذیرش:
1403/05/08

کلیدواژه‌ها:
بازی جنگ هوش مصنوعی، یادگیری عمیق، یادگیری تقویتی عمیق، بازی اول‌شخص تیرانداز.

چکیده

هدف: هدف اصلی بررسی و بهبود عملکرد مدل‌های Double Deep Q-Learning Network دربازی‌های اول‌شخص تیرانداز با تمرکز بر رقابت هوشمند است.

روش: در این پژوهش، از مدل‌های Deep Q-Network (DQN) و Deep Double Deep Q-Netwok (DDQN) برای بازی جنگ استفاده شده است. ابتدا، مدل‌های DQN و DDQN مورد بررسی قرار گرفته و سپس با روش بازپخش تجربه اولویت‌دار[1]، عملکرد آن بهبود داده شده است. از سه محیط بازی آزمایشی برای بررسی و ارزیابی مدل‌ها استفاده شده است.

یافته‌ها: یافته‌های این پژوهش نشان می‌دهد که معماری پیشنهادی Double Deep Q-Network با روش بازپخش تجربه اولویت‌دار، عملکرد بهتری نسبت به سایر الگوریتم‌های پیشنهادی در این زمینه داشته است.

نتیجه‌گیری: استفاده از روش بازپخش تجربه اولویت‌دار در یادگیری تقویتی، مزایای قابل‌توجهی را به همراه دارد که منجر به ارتقای عملکرد عامل هوش مصنوعی می‌شود. این روش با بهره‌گیری از داده‌ها و تجربیات باکیفیت، به‌طور هدفمند بر روی تجربیات آموزنده‌تر تمرکز می‌کند و بدین ترتیب، کارایی نمونه‌برداری را به‌طور قابل‌توجهی افزایش می‌دهد.

استناد: سعادتی رازیان، جعفر؛ اصغرزاده بناب، اکبر؛ محمدی، علیرضا و جبارزاده، صابر (1403). رویکرد نوین در بازی‌های اول‌شخص تیرانداز با استفاده از یادگیری تقویتی عمیق. دو فصلنامه بازی جنگ، 6 (13)، 61- 92.

DOI: 10.22034/ijwg.2024.453676.1083

ناشر: دانشگاه فرماندهی و ستاد ارتش جمهوری اسلامی ایران

[1] Prioritized Experience Replay (PER)

مقدمه

بازی جنگ ابزاری قدرتمند برای نیروهای مسلح در زمینه‌های مختلف است. به‌طورکلی، می‌توان کاربردهای آن را در سه دستهٔ آموزش، برنامه‌ریزی و آمادگی و تحقیق و توسعه دسته‌بندی کرد.

در آموزش، بازی جنگ به سربازان و فرماندهان کمک می‌کند تا در شرایط جنگی شبیه‌سازی‌شده، مهارت‌های تصمیم‌گیری خود را ارتقا داده و تاکتیک‌ها و استراتژی‌های جدید را آزمایش کنند. همچنین، از این ابزار می‌توان برای آموزش کارکنان در مورد تجهیزات و فناوری‌های جدید و ارتقای آمادگی رزمی آن‌ها استفاده کرد.

در برنامه‌ریزی و آمادگی، بازی جنگ به نیروهای نظامی کمک می‌کند تا سناریوهای احتمالی جنگ را پیش‌بینی کرده و برنامه‌های عملیاتی برای مقابله با آن‌ها تدوین کنند. همچنین، از این ابزار می‌توان برای شناسایی نقاط ضعف و قوت در توانایی‌های نظامی و تدوین برنامه‌های بهبود آن‌ها و همچنین تمرین یگان‌های نظامی برای عملیات خاص استفاده کرد.

در خصوص تحقیق و توسعه، بازی جنگ ابزاری برای آزمایش و ارزیابی فناوری‌های جدید نظامی و مطالعه رفتار دشمن است.

هوش مصنوعی ابزاری قدرتمند برای نیروهای مسلح در خصوص بازی جنگ است. استفاده از هوش مصنوعی در این بازی مزایای متعددی دارد، ازجمله:

- افزایش واقع‌گرایی: هوش مصنوعی می‌تواند محیط‌های جنگی شبیه‌سازی‌شدهٔ واقع‌گرایانه‌تری ایجاد کند که رفتار و تاکتیک‌های دشمن را به‌طور دقیق‌تری شبیه‌سازی می‌کند.
- افزایش چالش: هوش مصنوعی می‌تواند با تطبیق تاکتیک‌های بازیکنان، چالش‌های آموزشی را به‌طور پویا افزایش دهد.
- افزایش انعطاف‌پذیری: هوش مصنوعی می‌تواند سناریوهای جنگی مختلفی را با سطوح مختلف دشواری ایجاد کند.
- کاهش هزینه‌ها: هوش مصنوعی می‌تواند به کاهش هزینه‌های آموزش نظامی کمک کند.

- افزایش ایمنی: هوش مصنوعی می‌تواند به افزایش ایمنی آموزش نظامی کمک کند.

با توجه به موارد ذکرشده، استفاده از هوش مصنوعی در بازی جنگ مزایای متعددی دارد و به همین دلیل، استفاده از آن در ارتش‌های سراسر جهان رواج پیدا کرده و امروزه استفاده از هوش مصنوعی در بازی جنگ ارتش، گامی مهم در جهت ارتقای توانایی‌های نظامی و حفظ امنیت ملی است.

در این مقاله، با بررسی یادگیری عمیق[1]، از روش‌های یادگیری ماشین[2]، یادگیری تقویتی عمیق[3] را شرح می‌دهیم و در ادامه با معرفی یادگیری تقویتی دوگانه عمیق[4] و کارکرد آن در حل مسائل پیچیده با رویکرد شبکه‌های عصبی[5] و همچنین معرفی بازپخش تجربیات اولویت‌بندی شده[6]، به معرفی الگوریتم یادگیری تقویتی دوگانه عمیق با بازپخش تجربیات اولویت‌بندی شده[7] می‌پردازیم. به‌طورکلی قصد داریم در این پژوهش به بررسی امکانات و کارایی یادگیری تقویتی عمیق دربازی‌های ویدئویی بپردازیم و نشان دهیم که چگونه این تکنولوژی می‌تواند در بهبود عملکرد عامل‌های بازی و تصمیم‌گیری‌های آن‌ها مؤثر باشد.

مبانی نظری و بیشینه‌های پژوهش

مبانی نظری

بازی تک‌تیرانداز اول‌شخص نوعی بازی جنگ است که در آن بازیکن از دیدگاه اول‌شخص یک تک‌تیرانداز را کنترل می‌کند. این بازی معمولاً بر روی مخفی‌کاری و دقت تمرکز دارد، زیرا بازیکن باید دشمنان را از راه دور و بدون جلب‌توجه از بین ببرد.

ویژگی‌های مهم بازی تک‌تیرانداز اول‌شخص عبارت‌اند از:

- دیدگاه اول‌شخص: بازیکن از طریق چشمان تک‌تیرانداز بازی را مشاهده می‌کند.

---

[1] Deep Learning
[2] Machine Learning
[3] Deep Reinforcement Learning
[4] Double Deep Reinforcement Learning
[5] Neural Network [5]
[6] Prioritized Experience Replay
[7] DDQN with Prioritized Experience Replay

- مخفی‌کاری: بازیکن باید از محیط برای پنهان‌شدن از دشمنان خود استفاده کند.
- دقت: بازیکن باید برای شلیک دقیق به دشمنان از دوربین تک‌تیرانداز استفاده کند.
- مأموریت‌های مختلف: بازیکن باید در مأموریت‌های مختلفی مانند ترور، نجات گروگان و خنثی‌سازی بمب شرکت کند.

ترکیب هوش مصنوعی دربازی‌های تک‌تیرانداز اول‌شخص می‌تواند نقش مهمی ایفا کند و کارکردهای آن را به شرح زیر افزایش دهد:

- ایجاد دشمنان باهوش: هوش مصنوعی می‌تواند دشمنانی را ایجاد کند که رفتار واقع‌بینانه‌تری داشته باشند و به‌طور مؤثرتری به بازیکن پاسخ دهند. دشمنان باهوش می‌توانند چالش‌برانگیزتر و جذاب‌تر باشند و تجربه بازی را هیجان‌انگیزتر کنند.
- تنظیم سختی بازی: هوش مصنوعی می‌تواند برای تنظیم سختی بازی به کار رود. با افزایش سطح هوش مصنوعی، دشمنان قوی‌تر و باهوش‌تر می‌شوند و بازی چالش‌برانگیزتر می‌شود.
- ایجاد تنوع در بازی: هوش مصنوعی می‌تواند برای ایجاد تنوع در بازی به کار رود. با استفاده از هوش مصنوعی می‌توان دشمنان مختلف با توانایی‌های مختلف ایجاد کرد. این تنوع می‌تواند باعث شود که بازی جذاب‌تر و سرگرم‌کننده‌تر باشد.
- ایجاد محیط‌های پویا: هوش مصنوعی می‌تواند برای ایجاد محیط‌های پویا به کار رود. با استفاده از هوش مصنوعی می‌توان محیط‌هایی را ایجاد کرد که در آن دشمنان به‌طور تصادفی حرکت می‌کنند و به اتفاقات مختلف واکنش نشان می‌دهند. این پویایی می‌تواند باعث شود که بازی جذاب‌تر و غیرقابل‌پیش‌بینی‌تر باشد.
- ارائهٔ تجربیات شخصی‌سازی‌شده: هوش مصنوعی می‌تواند برای ارائهٔ تجربیات شخصی‌سازی‌شده به کار رود. با استفاده از هوش مصنوعی می‌توان بازی را به‌گونه‌ای تنظیم کرد که با سبک بازی بازیکن مطابقت داشته باشد.

پیشینه‌های پژوهش

- ولادیمیر نیح[1] و همکاران (2013) در پژوهشی، اولین مدل یادگیری عمیق را معرفی کرده‌اند که می‌تواند مستقیماً از اطلاعات دریافتی با حجم زیاد (مثل تصاویر خام) با استفاده از یادگیری تقویتی، کنترل را یاد بگیرد. مدل آن‌ها یک شبکه عصبی کانولوشنال است که با نسخه‌ای از یادگیری Q آموزش داده شده است. ورودی این شبکه، پیکسل‌های خام تصویر و خروجی آن، تابعی از ارزش است که پاداش‌های آینده را تخمین می‌زند. آن‌ها روش خود را روی هفت بازی آتاری از یک محیط یادگیری آرکید اعمال کردند، بدون اینکه هیچ تغییری در ساختار یا الگوریتم یادگیری بدهند. نتایج نشان داد که این مدل در شش بازی از تمام روش‌های قبلی بهتر عمل می‌کند و حتی در سه بازی، عملکرد آن از یک انسان متخصص هم فراتر می‌رود. اهمیت این پژوهش در این است که نشان می‌دهد یادگیری عمیق می‌تواند برای یادگیری تقویتی با داده‌هایی با حجم زیاد بسیار مؤثر باشد. این روش به مدل اجازه می‌دهد تا روابط پیچیده را مستقیماً از داده‌های خام یاد بگیرد.

- ولادیمیر نیح و همکاران (2015) در پژوهشی دیگر در خصوص یادگیری تقویتی عمیق و تصمیمات در سطح انسانی، با استفاده از پیشرفت‌های اخیر در آموزش شبکه‌های عصبی عمیق، یک عامل مصنوعی جدید به نام شبکه Q عمیق معرفی می‌کنند. این شبکه می‌تواند مستقیماً از اطلاعات حسی با حجم زیاد، با استفاده از یادگیری تقویتی سرتاسر، کنترل بهینه را یاد بگیرد. شبکه کیو عمیق از یک شبکه عصبی کانولوشنال برای استخراج ویژگی‌های معنی‌دار از تصاویر دریافتی استفاده می‌کند. سپس از این ویژگی‌ها برای تخمین ارزش هر عمل (میزان پاداشی که در آینده به دست خواهد آمد) استفاده می‌کند. در نهایت، از این

[1] Volodymyr Mnih

تخمین‌ها برای انتخاب بهترین عمل در هر موقعیت استفاده می‌شود. طی نتایج حاصله از آزمایش بر روی مجموعه بازی‌های آتاری ۲۶۰۰ نتایج نشان داد که شبکه کیو عمیق می‌تواند عملکرد تمام الگوریتم‌های قبلی را در این بازی‌ها پشت سر بگذارد و در مجموعه‌ای از ۴۹ بازی، به سطح عملکرد یک تست‌کنندهٔ حرفه‌ای بازی‌های آتاری برسد.

- گیلیوم لمپل[1] و همکاران (2016) در پژوهشی به بررسی چگونگی به‌کارگیری یادگیری تقویتی برای کنترل عامل‌های هوش مصنوعی در بازی‌های تیراندازی اول‌شخص پرداخته‌اند و یک عامل هوش مصنوعی مبتنی بر شبکه عصبی عمیق و الگوریتم Q-learning را توسعه داده‌اند که توانایی یادگیری و بهبود عملکرد در محیط‌های بازی FPS را دارد. آن‌ها از تکنیک‌های مختلفی مانند تجربهٔ بازپخش و سیاست حریصانه استفاده کرده‌اند تا عامل بتواند تصمیمات بهتری بگیرد. نتایج این پژوهش نشان می‌دهد که عامل هوش مصنوعی توسعه‌یافته، قادر است به‌طور مؤثری تاکتیک‌های مختلف بازی را یاد بگیرد و در محیط‌های پیچیده عملکرد خوبی داشته باشد. عملکرد عامل با گذشت زمان بهبودیافته و توانسته است با توجه به تجربیات گذشتهٔ خود، تصمیمات بهتری بگیرد.

- اشفق صالحین[2] (2024) در پژوهشی یک معماری پیشرفته از یادگیری تقویتی عمیق برای آموزش عامل‌های شبکه عصبی در بازی‌های آتاری ارائه داد که عامل با داشتن تنها تصاویر پیکسلی بازی، فضای عمل و اطلاعات پاداش، سیستم قادر به آموزش عامل‌ها برای بازی‌کردن در هر بازی آتاری است. در ابتدا، این سیستم از تکنیک‌های پیشرفته‌ای مانند شبکه‌های Q عمیق برای آموزش عامل‌های کارآمد استفاده می‌کند. جهت توسعه، شبکه‌های عصبی قابل شکل‌دهی به‌عنوان

---

[1] Guillaume Lample
[2] Ashfaq Salehin

عامل استفاده شده‌اند و مورد تجزیه و تحلیل قرار گرفته است. پیاده‌سازی قابلیت شکل‌پذیری بر اساس پس‌انتشار خطا و قانون بهینه‌سازی هبی[1] استفاده شده است. شبکه‌های عصبی شکل‌پذیر ویژگی‌های برجسته‌ای مانند یادگیری مادام‌العمر پس از آموزش اولیه دارند که آن‌ها را در محیط‌های یادگیری تطبیقی بسیار مناسب می‌سازد.

- گونچالو کوریدو[2] و همکاران (2023) در پژوهشی یک عامل جدید یادگیری تقویتی بدون مدل پیشنهاد داده‌اند که قادر به یادگیری راه حل برای یک وظیفه ناشناخته با دسترسی تنها به یک قسمت از مشاهدات ورودی است. آن‌ها از مفاهیم توجه بصری و درک فعال الهام گرفته‌اند که ویژگی‌هایی از انسان‌ها است و سعی کرده‌اند آن‌ها را با ایجاد یک مکانیزم توجه سخت[3] به عامل منتقل کنند. در این مکانیزم، مدل ابتدا تصمیم می‌گیرد که به کدام ناحیه از تصویر ورودی باید نگاه کند و فقط پس از آن به پیکسل‌های آن ناحیه دسترسی دارد. در معماری پیشنهادی، یک مدل موجود به نام مدل توجه مکرر[4] را اقتباس کرده و آن را با الگوریتم بهینه‌سازی سیاست مجاور ترکیب کرده‌اند. این تحلیل در بازی‌های Atari، Pong و SpaceInvaders که دارای فضای عمل گسسته هستند و در CarRacing که دارای فضای عمل پیوسته است، صورت می‌گیرد. علاوه بر ارزیابی عملکرد، حرکت توجه مدل خود را تحلیل کرده‌اند و آن را با آنچه می‌تواند نمونه‌ای از رفتار انسانی باشد، مقایسه کرده‌اند.

---

[1] Hebbian Plasticity
[2] Gonçalo Querido
[3] Hard Attention Mechanism
[4] Recurrent Attention Model

- ککین وانگ[1] و همکاران (2022) در این پژوهش، الگوریتم یادگیری تقویتی عمیق به نام شبکه‌های چند عملی[2] را معرفی می‌کنند که در فضاهای گسسته با ابعاد بالا عملکرد مناسبی دارد. در این روش فضای عمل N بعدی به N مؤلفه یک‌بعدی، به نام زیر عمل‌ها، تجزیه می‌شود و برای هر زیر عمل یک شبکه عصبی ارزش ایجاد می‌شود. سپس، شبکه‌های چند عملی از یادگیری تفاضل زمانی استفاده می‌کند تا شبکه‌ها را به‌صورت هم‌زمان آموزش دهد که ساده‌تر از آموزش یک شبکه تک برای خروجی عمل بزرگ مستقیم است. ارزیابی نتایج روش پیشنهادی بر روی سه محیط بازی نشان می‌دهد که شبکه‌های چند عملی بهتر و سریع‌تر از روش یادگیری تقویتی عمیق است.

یادگیری تقویتی عمیق[3]

یادگیری تقویتی عمیق زیرشاخه‌ای از یادگیری ماشین است که یادگیری تقویتی[4] و یادگیری عمیق[5] را با هم ترکیب می‌کند. یادگیری تقویتی به یک عامل (مانند یک ربات یا یک برنامه رایانه‌ای) اجازه می‌دهد از طریق آزمون‌وخطا در یک محیط، رفتار خود را یاد بگیرد. عامل با انجام اقداماتی در محیط، پاداش یا تنبیه دریافت می‌کند و به‌مرورزمان یاد می‌گیرد که چه اقداماتی منجر به پاداش بیشتر می‌شود.

یادگیری عمیق نوعی یادگیری ماشین است که از شبکه‌های عصبی مصنوعی برای یادگیری از داده‌ها استفاده می‌کند. شبکه‌های عصبی مصنوعی می‌توانند الگوهای پیچیده را در داده‌ها تشخیص دهند و از این الگوها برای پیش‌بینی یا تصمیم‌گیری استفاده کنند. یادگیری تقویتی عمیق از قدرت یادگیری عمیق برای بهبود عملکرد الگوریتم‌های یادگیری تقویتی استفاده می‌کند. شبکه‌های عصبی مصنوعی می‌توانند برای تخمین ارزش اقدامات مختلف در یک محیط، یا برای یادگیری سیاست‌های رفتاری بهینه برای عامل

---

[1] Keqin Wang
[2] Multi-Action Networks
[3] Deep Reinforcement Learning
[4] Reinforcement Learning
[5] Deep Learning

استفاده شوند. در یادگیری تقویتی عمیق، عامل در هر تکرار وضعیت محیط s_t را مشاهده می‌کند و تصمیم می‌گیرد که عمل a_t را با توجه به سیاست π انجام دهد و پس از آن مقدار پاداش r_t را مشاهده و بررسی می‌کند. هدف عامل آن است که سیاستی را پیدا کند که مقدار پاداش‌ها را به حداکثر برساند:

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$$

که مقدار γ یا همان فاکتور تخفیف به‌صورت زیر است:

$$\gamma \in [0,1]$$

که در آن T حداکثر تکرار مجاز بازی است. تابع Q برای سیاست π با توجه به انجام عمل a در وضعیت s به‌صورت زیر تعریف می‌شود:

$$Q^{\pi}(s,a) = E[Rt \mid st = s, a_t = a]$$

در این حالات خیلی معمول است که از یک تابع تخمین برای تقریب مقدار Q استفاده گردد که راهکار آن استفاده از شبکه Q عمیق است و آن از پارامتر θ استفاده می‌کند و هدف آن تخمین تابع Q بر اساس سیاست فعلی است که به سیاست بهینه Q^* نزدیک است و مقدار حداکثر آن در نظر گرفته می‌شود:

$$Q^*(s,a) = \max_{\pi} E[Rt \mid st = s, a_t = a] = \max_{\pi} Q^{\pi}(s,a)$$

به‌عبارت‌دیگر، هدف پیدا کردن θ به صورتی است که:

$$Q_\theta(s,a) \approx Q^*(s,a)$$

و معادله تابع Q بهینه یا Q^* بر اساس معادله بهینه بلمن به‌صورت زیر قابل بازنویسی است:

$$Q^*(s,a) = E\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right]$$

که اگر $Q_\theta \approx Q^*$ در نظر گرفته شود، داریم:

$$L_t(\theta_t) = E_{s,a,r,s'}\left[\left(y_t - Q_{\theta_t}(s,a)\right)^2\right]$$

که در رابطه ذکرشده، t قدم مرحله فعلی است و مقدار y_t به‌صورت زیر محاسبه می‌شود:

$$y_t = r + \gamma \max_{a'} Q_{\theta_t}(s', a')$$

و مقدار y_t که ثابت است، درنتیجه گرادیان زیر را خواهیم داشت:

$$\nabla_{\theta_t} L_t(\theta_t) = E_{s,a,r,s'}\left[\left(y_t - Q_\theta(s,a)\right) \nabla_{\theta_t} Q_{\theta_t}(s,a)\right]$$

که می‌توان از تقریب زیر نیز استفاده کرد:

$$\nabla_{\theta_t} L_t(\theta_t) \approx \left(y_t - Q_\theta(s,a)\right) \nabla_{\theta_t} Q_{\theta_t}(s,a)$$

اگرچه معادله بالا به‌صورت تقریبی محاسبه می‌گردد؛ ولی نتایج بررسی‌های گذشته نشان داده که عملکرد مناسبی داشته است.

در هر مرحله آموزش، عمل بعدی با استفاده از استراتژی ε حریصانه تولید می‌شود، به این صورت که با احتمال ε عمل بعدی به‌صورت تصادفی انتخاب می‌شود. معمول است که در ابتدا مقدار ε=1 در نظر گرفته شود و سپس از مقدار آن کاسته شود.

یادگیری تقویتی دوگانه

الگوریتم یادگیری Q به‌عنوان یکی از الگوریتم‌های محبوب نقشی اساسی در آموزش و توانمندسازی عاملان هوشمند ایفا می‌کند و به اثبات کارایی خود در حل مسائل مختلف رسیده است. بااین‌حال، این الگوریتم خالی از ایراد نیست و گاه دچار برآورد بیش از حد ارزش اقدامات می‌شود.

یکی از مشکلات کلیدی در الگوریتم یادگیری Q، سوء برآورد[1] است. در این پدیده، ارزش واقعی اقدامات بیش از حد تخمین زده می‌شود. اولین بار ترون[2] و شوارتز[3] (۱۹۹۳) به بررسی این موضوع پرداختند. آن‌ها نشان دادند که اگر مقدار خطای موجود در ارزش اقدامات تصادفی باشد، هر هدف نهایی (ارزش به‌دست‌آمده) می‌تواند تا مقدار مشخصی بیش از حد برآورد شود. علاوه بر این، آن‌ها با ارائهٔ یک مثال عملی نشان دادند که این

[1] Overestimate
[2] Thrun
[3] Schwartz

سوء برآوردها حتی می‌توانند به انتخاب سیاست‌های نامطلوب در بلندمدت منجر شوند. همچنین، آن‌ها وجود این سوء برآوردها را در مسائل کوچک و با استفاده از تقریب تابعی[۱] به اثبات رساندند. بعدها، فن هاسلت[۲] (۲۰۱۰) استدلال کرد که حتی با استفاده از نمایش جدولی[۳] نیز، وجود نویز در محیط می‌تواند منجر به سوء برآورد شود. او برای حل این مشکل، الگوریتم یادگیری دوگانه Q را پیشنهاد داد.

با توجه به بررسی‌های انجام‌شده، می‌توان بیان کرد که هر نوع خطایی در تخمین می‌تواند منجر به یک سوگیری به سمت بالا شود، صرف‌نظر از اینکه این خطا ناشی از نویز محیطی، تقریب تابعی، عدم ثبات (غیر ایستا بودن محیط)، یا هر عامل دیگری باشد. این موضوع اهمیت ویژه‌ای دارد، زیرا در عمل، هر روشی به دلیل ناشناخته بودن مقادیر واقعی در ابتدای یادگیری، با مقداری عدم دقت مواجه خواهد شد.

در اینجا، یادگیری دوگانه Q[۴] به‌عنوان راه‌حلی نوین پا به عرصه می‌گذارد. این الگوریتم با اتکا به دو تخمین‌گر جداگانه برای ارزش Q، به‌طور هوشمندانه اقدام به انتخاب و ارزیابی اقدامات می‌کند و بدین ترتیب، مشکل برآورد بیش از حد ارزش اقدامات در یادگیری Q را مرتفع می‌سازد.

مزیت کلیدی یادگیری دوگانه Q، کاهش سوگیری و ارتقای ثبات در عملکرد عامل است. این امر، انتخاب‌های دقیق‌تر و نتایج بهینه‌تر را به ارمغان می‌آورد. علاوه بر این، یادگیری دوگانه Q، مزایای دوگانه Q را نیز به‌طور کامل حفظ می‌کند. به‌عنوان‌مثال، این الگوریتم نیازی به مدل صریح از محیط ندارد و به‌خوبی با محیط‌های تصادفی سازگار می‌شود. درمجموع، یادگیری دوگانه Q گامی فراتر در مسیر یادگیری تقویتی به شمار می‌رود و با ارائه روشی دقیق‌تر و کارآمدتر برای تخمین ارزش اقدامات، به ارتقای عملکرد عاملان هوشمند در حل مسائل مختلف یاری می‌رساند.

با توجه به توضیحات ارائه‌شده، دو تابع ارزش با اختصاص تصادفی هر تجربه برای به‌روزرسانی یکی از آن‌ها انجام می‌شود؛ بنابراین، دو مجموعه وزن، θ و θ' وجود دارد. برای هر به‌روزرسانی، از یک مجموعه وزن برای تعیین بهترین سیاست (استراتژی) و از مجموعه دیگر برای تعیین ارزش آن استفاده می‌شود.

---

[4]Function Approximation
[5]Van Hasselt
[6]Tabular Representation
[7]Double Q-Learning

$$Y_t^Q = R_{t+1} + \gamma Q(S_{t+1}, argmaxQ(S_{t+1}, a; \theta_t); \theta_t)$$

و خطای یادگیری دوگانه Q به‌صورت زیر محاسبه می‌شود:

$$Y_t^{DoubleQ} = R_{t+1} + \gamma Q(S_{t+1}, argmaxQ(S_{t+1}, a; \theta_t); \theta_t')$$

انتخاب اقدام در این الگوریتم، مشابه یادگیری Q، بر اساس وزن‌های آنلاین θ_t انجام می‌شود. به این معنی که ارزش بهترین سیاست[1] با توجه به مقادیر فعلی که توسط θ_t تعریف می‌شوند، تخمین زده می‌شود؛ اما نکته کلیدی در یادگیری دوگانه Q، نحوهٔ ارزیابی این سیاست است. در اینجا، از مجموعه دوم وزن‌ها، θ_t' برای ارزیابی منصفانه و بی‌طرفانه ارزش سیاست استفاده می‌شود. این امر به‌منظور جلوگیری از برآورد بیش از حد ارزش‌ها انجام می‌شود که در یادگیری Q رخ می‌دهد. مجموعه دوم وزن‌ها، یعنی θ_t' نیز به‌صورت تقارنی و با جابه‌جایی نقش‌های θ و θ_t' به‌روزرسانی می‌شوند. این جابه‌جایی نقش‌ها به‌منظور حفظ تعادل و جلوگیری از غلبهٔ یک مجموعهٔ وزن بر دیگری انجام می‌شود.

درمجموع، یادگیری دوگانه Q با تفکیک انتخاب و ارزیابی اقدامات و استفاده از دو مجموعهٔ وزن مستقل، به تخمین دقیق‌تر ارزش سیاست و درنهایت، عملکرد بهتر عامل در حل مسائل مختلف دست می‌یابد.

یادگیری تقویتی دوگانه عمیق

همان‌طور که در بخش قبلی گفته شد، تخمین بیش از حد (سوء برآورد) در یادگیری Q مشکل‌ساز است. الگوریتم یادگیری دوگانه Q راه‌حلی برای کاهش این سوء برآورد ارائه می‌دهد که ایدهٔ اصلی آن، تجزیه عملگر argmax به دو بخش مجزا است: انتخاب عمل و ارزیابی اقدام. بااینکه این دو بخش کاملاً مستقل از هم نیستند، اما یادگیری تقویتی عمیق به‌عنوان گزینه‌ای مناسب برای تابع ارزش دوم عمل می‌کند، بدون اینکه نیاز به معرفی شبکه‌های اضافی باشد؛ بنابراین، در یادگیری Q دوگانهٔ عمیق[2]، انتخاب بهترین اقدام بر اساس شبکه به‌صورت آنلاین انجام می‌شود، اما برای تخمین ارزش آن اقدام از شبکه هدف استفاده می‌گردد. به‌عبارت‌دیگر، در به‌روزرسانی الگوریتم دوگانه عمیق، هدف نهایی شبکه

---

[1] Policy

[1] Double Deep Q-learning Network

یادگیری تقویتی عمیق، با مقداری جدید جایگزین می‌شود که با استفاده از شبکه هدف محاسبه‌شده است.

پس با توجه به توضیحات ارائه‌شده، مقدار $Y_t^{DoubleQ}$ در یادگیری دوگانه با مقدار $Y_t^{DoubleDQNt}$ جایگزین می‌گردد، بنابراین داریم:

$$Y_t^{DoubleQ} = R_{t+1} + \gamma Q(S_{t+1}, argmaxQ(S_{t+1}, a; \theta_t); \theta_t^-)$$

در مقایسه با الگوریتم اصلی یادگیری دوگانه Q، در یادگیری دوگانه عمیق Q برای ارزیابی سیاست حریصانه جاری (بهترین اقدام در لحظه با توجه به دانش فعلی)، وزن‌های شبکه دوم θ_t' با وزن‌های شبکه هدف θ^-_t جایگزین می‌شوند. به‌روزرسانی شبکه هدف همچنان بدون تغییر نسبت به DQN باقی می‌ماند و به‌صورت یک کپی دوره‌ای از شبکه آنلاین انجام می‌شود.

بازپخش تجربه اولویت‌دار در یادگیری تقویتی عمیق

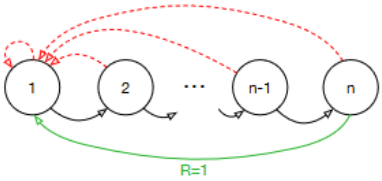
طی بررسی به‌عمل‌آمده توسط تام کوال[1] و همکاران (2016) در زمینهٔ یادگیری تقویتی، بازپخش تجربه به عاملان این امکان را می‌دهد که تجربیات گذشته را به خاطر بسپارند و دوباره از آن‌ها به نحو مناسب‌تری استفاده کنند. در الگوریتم‌هایی که قبلاً به آن‌ها اشاره شد، گذارهای تجربی به‌طور یکنواخت از حافظهٔ بازپخش نمونه‌برداری و بدون توجه به اهمیت آن‌ها پخش یا به کار گرفته می‌شدند.

ما از اولویت‌بندی بازپخش تجربه در شبکه‌های Q عمیق استفاده می‌کنیم که می‌تواند در بسیاری از بازی‌های آتاری به عملکردی در سطح انسان دست یابد. DQN با اولویت‌بندی

[2]Tom Schaul

بازپخش تجربه به سطح جدیدی از دانش دست می‌یابد و در اکثر بازی‌ها، عملکرد بهتری نسبت به DQN با بازپخش یکنواخت ارائه می‌دهد.

برای درک مزیت اولویت‌بندی، محیطی مصنوعی به نام صخره‌نوردی کورکورانه[1] (شکل زیر) را فرض نمایید؛ این محیط نمونه‌ای از چالش اکتشاف (جستجو برای یافتن مسیر) در شرایطی است که پاداش‌ها نادر هستند.



تصویر 1 نمایش محیط مصنوعی صخره‌نوردی کورکورانه

این محیط که تنها n حالت دارد، مستلزم تعداد تصاعدی گام‌های تصادفی تا رسیدن به اولین پاداش غیر صفر است. به‌عبارت‌دیگر، احتمال رسیدن به پاداش از طریق یک دنباله تصادفی از اقدامات، برابر با 2^{-n} است. علاوه بر این، مرتبط‌ترین گذارها[2] (از موفقیت‌های نادر) در میان انبوهی از موارد شکست تکراری پنهان‌شده‌اند، شبیه به رباتی دوپا که بارها می‌افتد تا اینکه راه رفتن را یاد بگیرد. هر دو عامل به‌روزرسانی‌های یادگیری Q را روی گذارهایی انجام می‌دهند که از یک حافظهٔ بازپخش مشترک استخراج‌شده‌اند. عامل اول، گذارها را به‌طور یکنواخت و تصادفی بازپخش می‌کند، درحالی‌که عامل دوم از یک پیشگو[3] برای اولویت‌بندی گذارها استفاده می‌کند. این پیشگو به‌صورت حریصانه[4] گذارهایی را انتخاب می‌کند که حداکثر کاهش را بعد از به‌روزرسانی پارامترها در وضعیت فعلی به همراه داشته باشند.

بررسی‌ها نشان می‌دهد که انتخاب گذارها با ترتیب مناسب می‌تواند منجر به سرعت‌های تصاعدی نسبت به انتخاب تصادفی یکنواخت شود. پیش‌بینی به این صورت قطعاً غیرواقعی

---
[1] Blind Cliffwalk
[2] Transitions
[3] Oracle
[4] Greedy

است، اما این شکاف بزرگ، ما را به جستجو برای رویکردی عملی سوق می‌دهد که بازپخش تصادفی یکنواخت را بهبود بخشد.

روش‌های مختلفی برای اولویت‌بندی وجود دارد، در ادامه به دو نمونه می‌پردازیم.

- اولویت‌بندی با استفاده از خطای TD:

بخش اصلی بازپخش اولویت‌دار، معیاری است که اهمیت هر گذار[1] را با آن می‌سنجیم. یک معیار ایده‌آل، میزان یادگیری عامل تقویتی از یک گذار در وضعیت فعلی آن (پیشرفت مورد انتظار یادگیری) است. درحالی‌که دسترسی مستقیم به این معیار وجود ندارد، خطای TD نشان می‌دهد که چقدر یک گذار غافلگیرکننده یا غیرمنتظره است، به‌طور خاص، مقدار ارزش چقدر با برآورد بوت‌استرپ[2] گام بعدی آن فاصله دارد. این معیار به‌ویژه برای الگوریتم‌های یادگیری تقویتی آنلاین و افزایشی مانند سارسا[3] یا یادگیری Q که از قبل خطای TD را محاسبه کرده و پارامترها را متناسب با δ به‌روزرسانی می‌کنند، مناسب است. خطای TD گاهی اوقات می‌تواند برآورد ضعیفی باشد، به‌عنوان‌مثال زمانی که پاداش‌ها نویز داشته باشند. برای نشان دادن اثربخشی اولویت‌بندی بازپخش با استفاده از خطای TD، عملکرد مبنای یکنواخت و پیشگو در محیط صخره‌نوردی کورکورانه را با الگوریتم اولویت‌بندی حریصانه بر اساس خطای TD مقایسه می‌کنیم. این الگوریتم، آخرین خطای TD برخورد شده را به همراه هر گذار در حافظهٔ بازپخش ذخیره می‌کند. گذاری با بیشترین مقدار مطلق خطای TD، از حافظه بازپخش انتخاب و مجدداً پخش می‌شود. یک به‌روزرسانی یادگیری Q روی این گذار اعمال می‌شود که وزن‌ها را متناسب با خطای TD به‌روز می‌کند. گذارهای جدید بدون خطای TD شناخته‌شده وارد می‌شوند، بنابراین ما برای اطمینان از اینکه حداقل یک‌بار همه تجربیات دیده‌شده‌اند، آن‌ها را در اولویت حداکثر قرار می‌دهیم. آزمایش‌های انجام‌شده حاکی از آن است که

---

[1] Transition
[2] Bootstrap
[3] SARSA

این الگوریتم منجر به کاهش قابل‌توجه تعداد تلاش برای حل صخره‌نوردی کورکورانه شده است.

- اولویت‌بندی تصادفی:

اولویت‌بندی حریصانه بر اساس خطای TD باوجود کارکرد مناسب، چند چالش دارد، مورد اول این است که برای جلوگیری از جستجوی پرهزینه در کل حافظه بازپخش، خطای TD فقط برای گذارهایی که دوباره پخش می‌شوند، به‌روزرسانی می‌شود. درنتیجه، گذارهایی که در اولین بازدید خطای TD کمی دارند، ممکن است برای مدت طولانی دوباره پخش نشوند. علاوه بر آن، این روش به جهش‌های نویز (مثلاً زمانی که پاداش‌ها تصادفی هستند) حساس است که می‌تواند با بوت‌استرپ کردن تشدید شود، جایی که خطاهای تقریب به‌عنوان منبع دیگری از نویز ظاهر می‌شوند. درنهایت، اولویت‌بندی حریصانه بر روی زیرمجموعهٔ کوچکی از تجربیات تمرکز می‌کند، خطاها به‌آرامی کاهش می‌یابند، به‌ویژه هنگام استفاده از تقریب تابعی، به این معنی که گذارهایی با خطای اولیهٔ بالا به‌طور مکرر دوباره پخش می‌شوند. این فقدان تنوع باعث می‌شود سیستم مستعد برازش بیش از حد شود.

برای غلبه بر مشکلات ذکرشده، ما یک روش نمونه‌برداری تصادفی را معرفی می‌کنیم که بین اولویت‌بندی کاملاً حریصانه و نمونه‌برداری تصادفی یکنواخت واسطه‌گری می‌کند. ما اطمینان حاصل می‌کنیم که احتمال نمونه‌برداری شدن یک گذار، با اولویت آن گذار رابطه یکنواخت صعودی داشته باشد، درحالی‌که تضمین می‌کنیم حتی برای گذار با کمترین اولویت، احتمال غیر صفر وجود داشته باشد. به‌طور مشخص، ما احتمال نمونه‌برداری از گذار i را به‌صورت زیر تعریف می‌کنیم:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$$

که در آن $p_i > 0$ اولویت گذار i است. ضریب α میزان استفاده از اولویت‌بندی را تعیین می‌کند، به‌طوری‌که α برابر با صفر حالت یکنواخت است. اولین وضعیتی که

در نظر می‌گیریم اولویت‌بندی مستقیم و متناسب است به‌صورت زیر محاسبه می‌گردد:

$$p_i = |\delta_i| + \epsilon$$

مقدار ϵ یک مقدار ثابت مثبت کوچک است که مانع از این می‌شود که گذارها بعد از صفر شدن خطای آن‌ها، دیگر مورد بازبینی قرار نگیرند.
دومین وضعیت، اولویت‌بندی غیرمستقیم مبتنی بر رتبه است که در آن:

$$p_i = \frac{1}{rank(i)}$$

است که مقدار $rank(i)$ رتبه‌ی گذار i است زمانی که حافظهٔ بازپخش بر اساس $|\delta_i|$ مرتب‌سازی شود. در این حالت، P به یک توزیع توانی با نماینده α تبدیل می‌شود. هردوی این توزیع‌ها با $|\delta|$ صعودی یکنواخت هستند، اما احتمال دومی قوی‌تر است، زیرا نسبت به داده‌های پرت[1] حساسیت کمتری دارد.

بازپخش تجربه اولویت‌دار در یادگیری تقویتی دوگانهٔ عمیق

شبکه Q عمیق و شبکه Q دوگانه عمیق، هر دو الگوریتم یادگیری تقویتی هستند، اما تفاوت‌های کلیدی باهم دارند. مهم‌ترین تفاوت آن‌ها در نحوهٔ تخمین مقدار ارزش برای عمل‌ها (همان Q-Value) است. DQN از یک شبکه واحد برای تخمین هر دو مقدار Q فعلی و هدف استفاده می‌کند که می‌تواند منجر به بیش‌برآورد این مقادیر شود. DDQN برای حل این مشکل از دو شبکه جداگانه استفاده می‌کند: یک شبکه برای مقدار Q هدف و یک شبکه برای مقدار Q فعلی. این کار به کاهش پدیده بیش‌برآورد در DQN کمک می‌کند.
تفاوت دیگر در نحوه به‌روزرسانی مقدار Q هدف است. DQN از یک روش ساده مبتنی بر شبکه فعلی برای به‌روزرسانی مقدار Q هدف استفاده می‌کند و در DDQN برای انتخاب بهترین عمل در حالت بعدی از شبکه هدف و برای محاسبه مقدار Q آن عمل، از شبکه فعلی استفاده می‌کند. این روش به کاهش بیش‌برآورد در DDQN کمک می‌کند. پس DDQN بر اساس معماری DQN ساخته‌شده است، اما با معرفی رویکرد یادگیری دوگانه

[1] Outliers

Q که از دو شبکه استفاده می‌کند، باعث تخمین دقیق‌تر مقدار Q و رفع مشکل بیش‌برآورد در DQN می‌شود. این تغییر، پایداری و همگرایی فرایند یادگیری را در سناریوهای یادگیری تقویتی بهبود می‌بخشد.

حال رویکرد جدیدی که معرفی می‌شود، بر مبنای روش بازپخش تجربیات اولویت‌بندی شده در شبکه DDQN است. می‌توان با استفاده از رویکرد بازپخش تجربیات اولویت‌بندی شده، کارایی DDQN را با تغییر نحوهٔ نمونه‌برداری از تجربیات ذخیره‌شده، بهبود بخشید. در اینجا یک مرور کلی در مورد چگونگی انجام این کار ارائه‌شده است:

1. ذخیره‌سازی تجربیات: مشابه با DDQN استاندارد، همچنان هر تجربه (شامل وضعیت محیط، عمل انجام‌شده، پاداش دریافتی و وضعیت بعدی محیط) را در یک حافظه ذخیره‌سازی می‌کنید، درحالی‌که عامل با محیط تعامل دارد.
2. اولویت‌بندی تجربیات: به‌جای نمونه‌برداری تصادفی از تجربیات، به هر تجربه یک اولویت اختصاص می‌دهید. این اولویت می‌تواند بر اساس بزرگی خطای اختلاف‌زمانی (خطای TD) باشد که نشان می‌دهد این تجربه چقدر غافلگیرکننده یا غیرمنتظره بوده است. تجربیات با خطای TD بالاتر، مهم‌تر در نظر گرفته می‌شوند و بنابراین احتمال نمونه‌برداری از آن‌ها بیشتر است.
3. نمونه‌برداری از تجربیات: هنگام به‌روزرسانی وزن‌های شبکه، بر اساس اولویت آن‌ها از حافظه ذخیره‌سازی نمونه‌برداری می‌کنید. این کار با استفاده از تکنیکی به نام اولویت‌بندی متناسب انجام می‌شود، جایی که احتمال نمونه‌برداری از یک تجربهٔ خاص با اولویت آن نسبت مستقیم دارد.
4. به‌روزرسانی وزن‌ها: سپس از این تجربیات نمونه‌برداری شده برای به‌روزرسانی وزن‌های شبکه استفاده می‌کنید. برای جبران توزیع نامتقارن تجربیات، از وزن‌های نمونه‌گیری بااهمیت[1] برای تنظیم به‌روزرسانی‌ها استفاده می‌کنید.

[1]Importance Weights Sampling

5. تنظیم اولویت‌ها: پس از هر به‌روزرسانی وزن، خطای TD را برای تجربیات نمونه‌برداری شده دوباره محاسبه کرده و اولویت آن‌ها را در حافظه ذخیره‌سازی بر اساس آن تنظیم می‌کنید.
6. به‌روزرسانی شبکه هدف: همان‌طور که در DDQN استاندارد وجود دارد، به‌طور دوره‌ای شبکه هدف را با وزن‌های شبکه اصلی به‌روز می‌کنید.

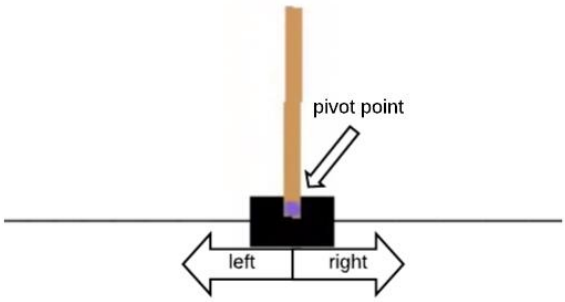
این رویکرد به عامل یادگیرنده اجازه می‌دهد تا روی تجربیات مهم‌تر تمرکز کند که می‌تواند منجر به یادگیری کارآمدتری شود. همچنین بایستی به خاطر سپرد، درحالی‌که بازپخش اولویت‌دار تجربه می‌تواند کارایی یادگیری را بهبود بخشد، به دلیل نیاز به نگهداری و نمونه‌گیری از صف اولویت‌دار، قاعدتاً پیچیدگی و بار محاسباتی بیشتری را نیز به همراه دارد؛ بنابراین، باید توازنی بین کارایی و پیچیدگی در نظر گرفته شود.

ویژگی‌های بازی

برای آموزش و ارزیابی الگوریتم پیشنهادی یادگیری تقویتی دوگانه عمیق با بازپخش تجربیات اولویت، از سه محیط بازی ارائه‌شده توسط OpenAI استفاده می‌کنیم که در ادامه، به شرح هر یک می‌پردازیم:

1- محیط بازی Cart Pole:

محیط بازی Cart Pole یک محیط آموزش و ارزیابی الگوریتم‌های یادگیری تقویتی است. در این محیط، یک میله بلند و نازک روی یک چرخ‌دستی دوچرخه قرارگرفته است و هدف عامل یادگیرنده، کنترل چرخ‌دستی با اعمال نیرو به سمت چپ یا راست است، به‌گونه‌ای که میله در حالت عمودی (تعادل) باقی بماند.



تصویر ۲: محیط بازی Cart Pole

در این محیط، دو نوع عمل وجود دارد:

الف) نیروی سمت چپ: چرخ‌دستی را به سمت چپ هل می‌دهد.

ب) نیروی سمت راست: چرخ‌دستی را به سمت راست هل می‌دهد.

شدت نیرو ثابت است و در هر بار اعمال عمل، یکسان خواهد بود. چهار مشاهده در هر گام زمانی ارائه می‌شود:

1- موقعیت چرخ‌دستی: موقعیت افقی چرخ‌دستی در طول محور x.

2- سرعت چرخ‌دستی: سرعت افقی چرخ‌دستی.

3- زاویهٔ میله: زاویهٔ میله نسبت به خط عمود (صفر درجه نشان‌دهنده حالت عمودی است).

4- سرعت زاویه‌ای میله: سرعت چرخش میله حول محور عمودی.

در این محیط، پاداش به ازای هر گام زمانی که میله در حالت عمودی باقی بماند، 1 واحد در نظر گرفته می‌شود. اگر میله واژگون شود یا چرخ‌دستی بیش از حد به سمت چپ یا راست حرکت کند، بازی به پایان می‌رسد و پاداش صفر دریافت می‌شود، همچنین بازی در دو حالت خاص به پایان می‌رسد:

1- واژگونی میله: زمانی که زاویهٔ میله از 15 درجه بیشتر یا کمتر شود، میله واژگون شده تلقی می‌شود و بازی به پایان می‌رسد.

2- حرکت بیش از حد چرخ‌دستی: اگر موقعیت افقی چرخ‌دستی از 2.4 واحد بیشتر یا کم‌تر شود، چرخ‌دستی بیش از حد حرکت کرده و بازی به پایان می‌رسد.

محیط Cart Pole به دلیل وجود ناپایداری و پویایی غیرخطی، چالش‌های متعددی را برای الگوریتم‌های یادگیری تقویتی ارائه می‌دهد. برای حل این چالش‌ها، الگوریتم باید بتواند دینامیک محیط را به‌طور دقیق مدل‌سازی کند و استراتژی‌های کنترلی کارآمدی را برای حفظ تعادل میله در شرایط مختلف اعمال کند.

2- محیط بازی Acrobat:

محیط بازی Acrobot همانند بازی Cart Pole، یک محیط برای آموزش و ارزیابی الگوریتم‌های یادگیری تقویتی به کار می‌رود. در این بازی، بایستی دو بازوی بلند و نازک متصل‌به‌هم که از یک نقطه ثابت به سقف وصل شده‌اند، کنترل شوند. هدف عامل یادگیرنده، حرکت دادن این دو بازو به‌گونه‌ای است که نوک بازوی دوم به یک نقطه هدف مشخص در محیط برسد. در این بازی، سه نوع عمل وجود دارد:

1. نیروی چرخش مفصل اول: به مفصل اول بین دو بازو گشتاور اعمال می‌کند.
2. نیروی چرخش مفصل دوم: به مفصل دوم بین دو بازو گشتاور اعمال می‌کند.
3. بدون عمل: هیچ گشتاوری به هیچ‌یک از مفاصل اعمال نمی‌شود.

شدت گشتاورها ثابت است و در هر عمل، یکسان خواهد بود. شش مشاهده در هر گام زمانی ارائه می‌شود:

1. موقعیت زاویه‌ای مفصل اول: زاویه مفصل اول بین دو بازو.
2. سرعت زاویه‌ای مفصل اول: سرعت چرخش مفصل اول.
3. موقعیت زاویه‌ای مفصل دوم: زاویه مفصل دوم بین دو بازو.
4. سرعت زاویه‌ای مفصل دوم: سرعت چرخش مفصل دوم.
5. موقعیت افقی نوک بازوی دوم: موقعیت افقی نوک بازوی دوم در محور x.

6. سرعت افقی نوک بازوی دوم: سرعت افقی نوک بازوی دوم.

در این بازی، پاداش به ازای هر گام زمانی که نوک بازوی دوم به نقطه هدف نزدیک‌تر شود، 1 واحد در نظر گرفته می‌شود. اگر نوک بازوی دوم به نقطه هدف برسد، پاداش 100 واحد دریافت خواهد شد. همچنین بازی در دو حالت به پایان می‌رسد:

1. سقوط: زمانی که هر یک از زوایای مفاصل از 3.66 رادیان بیشتر یا کمتر شود، سقوط کرده و بازی به پایان می‌رسد.
2. زمان بیش از حد: اگر قسمت زمانی بیش از 1000 گام طول بکشد، بازی به پایان می‌رسد.

الگوریتم حل بازی باید بتواند استراتژی‌های کنترلی کارآمدی را برای حرکت دادن Acrobot به سمت نقطه هدف و حفظ تعادل آن در طول مسیر باوجود فضای حالت با ابعاد بالا اعمال کند.

3-محیط بازی Alien:

محیط بازی Alien یک محیط از بازی‌های سری آتاری 2600[1] است که برای آموزش و ارزیابی الگوریتم‌های یادگیری تقویتی به‌صورت گسترده کاربرد دارد. در این بازی، عامل در یک هزارتو[2]، مانند سفینه فضایی گیر افتاده و سه موجود فضایی در حال تعقیب وی هستند. هدف عامل یادگیرنده، نابودی تخم‌های موجودات فضایی است. عامل برای این کار یک شعله‌افکن دارد و می‌تواند از آن برای دور کردن موجودات فضایی در مواقع خطرناک استفاده کند. همچنین گاه‌به‌گاه می‌تواند یک قدرت ماورایی دریافت کند که به عامل این امکان را می‌دهد که برای مدت کوتاهی موجودات فضایی را از بین ببرد.

---

[1] Atari 2600
[2] Maze

تصویر ۳: محیط بازی Alien

محیط بازی Alien و بازی‌های تیراندازی اول‌شخص هر دو تجربه‌ای پویا به مخاطب ارائه می‌دهند که به مهارت، دقت، استراتژی و حرکات سریع نیاز دارند. در هر دو نوع بازی، باید با دشمنان مقابله کرده و برای رسیدن به اهداف اقدام مناسب را انجام داد. شباهت‌های کلیدی بین دو بازی به‌صورت زیر است:

1. چالش: هر دو نوع بازی به مهارت و دقت بالایی نیاز دارند. عامل در Alien، باید تخم‌ها و موجودات فضایی را با دقت هدف قرار دهد و از آن‌ها دوری کند و دربازی‌های FPS، باید به‌سرعت دشمنان را شناسایی و هدف قرار دهد و درعین‌حال از تیراندازی آن‌ها نیز در امان بماند.

2. پویایی: هر دو نوع بازی پویا و سریع هستند. در Alien، عامل باید دائماً در حال حرکت باشد و محیط را اسکن کند تا از تخم‌ها و موجودات فضایی جدید آگاه شود و دربازی‌های FPS، باید دائماً به دنبال دشمنان جدید باشد و به‌موقع به آن‌ها واکنش نشان دهد.

3. استراتژی: هر دو نوع بازی به استراتژی و برنامه‌ریزی نیاز دارند. در Alien، عامل باید تصمیم بگیرد که چه زمانی و کجا شلیک کند و چگونه از موجودات فضایی دوری کند و دربازی‌های FPS، باید تصمیم بگیرد که از چه سلاحی استفاده کند، چگونه به دشمنان نزدیک شود و چه زمانی پنهان شود.

4. مهارت‌های حرکتی: هر دو نوع بازی به مهارت‌های حرکتی خوب نیاز دارند. در Alien، عامل باید بتواند به‌سرعت و با دقت شخصیت خود را کنترل کند و دربازی‌های FPS، باید بتواند به‌سرعت هدف بگیرد و شلیک کند.

در این بازی، شش نوع عمل وجود دارد:

- ثابت: هیچ عملی انجام نمی‌دهید.
- آتش: با شعله‌افکن به سمت جلو آتش پرتاب می‌کند.
- حرکت به سمت بالا: به سمت بالا حرکت می‌کند.
- حرکت به سمت راست: به سمت راست حرکت می‌کند.
- حرکت به سمت چپ: به سمت چپ حرکت می‌کند.
- حرکت به سمت پایین: به سمت پایین حرکت می‌کند.

نحوه امتیازدهی پاداش‌ها به عامل به‌صورت زیر است:

- 1 امتیاز برای هر تخم موجود فضایی که نابود می‌کند.
- 5 امتیاز برای هر موجود فضایی که با شعله‌افکن از بین می‌برد.
- 10 امتیاز برای هر موجود فضایی که باقدرت ماورایی از بین می‌برد.
- 1 امتیاز منفی برای هر بار که موجود فضایی عامل را لمس می‌کند.

و بازی زمانی به پایان می‌رسد که:

- تمام تخم‌های موجودات فضایی نابود شود.
- عامل توسط موجودات فضایی کشته شود.
- عامل امتیاز مشخصی (100000 امتیاز) کسب کند.

آموزش

آموزش با بازپخش تجربه اولویت‌دار(PER) ، طی مراحل زیر انجام می‌شود:

1- مقداردهی ظرفیت حافظه بازپخش:

حافظهٔ بازپخش در PER کمی با حافظهٔ بازپخش استاندارد متفاوت است. این حافظه نه‌تنها تجربیات عامل را ذخیره می‌کند، بلکه به هر تجربه اولویتی اختصاص می‌دهد که برای تعیین احتمال نمونه‌برداری از آن تجربه استفاده می‌شود. اولویت هر تجربه پس از هر مرحله یادگیری به‌روز می‌شود.

2- مقداردهی تابع مقدار - عمل Q را با وزن‌های تصادفی:

تابع مقدار - عمل Q به‌عنوان یک شبکه عصبی عمیق، یک حالت و یک عمل را به‌عنوان ورودی می‌گیرد و مقدار Q مربوطه را خروجی می‌دهد. وزن‌های DNN در ابتدای آموزش به‌صورت تصادفی مقداردهی می‌شوند.

3- برای هر اپیزود:

یک اپیزود توالی از حالت‌ها، اعمال و پاداش‌ها است که با یک حالت نهایی پایان می‌یابد. برای هر اپیزود، توالی را مقداردهی و حالت اولیه پیش‌پردازش می‌شود و خروجی آن وارد شبکه عصبی عمیق می‌شود تا مقادیر Q برای اعمال احتمالی به دست آید، سپس برای هر اپیزود، مراحل زیر تکرار می‌شود:

3-1. انتخاب یک عمل: با استفاده از سیاست حریصانه-تصادفی، یا یک عمل تصادفی انتخاب می‌شود، یا عملی با بالاترین مقدار Q انتخاب می‌شود. ε در ابتدا روی مقدار بالایی (۱.۰) تنظیم می‌شود و به‌تدریج کاهش می‌یابد تا به عامل اجازه دهد ابتدا محیط را کاوش کند و سپس از دانش خود بهره‌برداری کند.

3-2. اجرای عمل و مشاهده پاداش و حالت بعدی: عمل انتخاب‌شده در محیط انجام می‌شود و عامل پاداش و حالت بعدی را دریافت می‌کند.

3-3. ذخیره تجربه در حافظه بازپخش: تجربه (حالت، عمل، پاداش، حالت بعدی، پایان) در حافظه بازپخش ذخیره می‌شود.

3-4. نمونه‌برداری تصادفی از حافظه بازپخش: مجموعه‌ای کوچک از تجربیات به‌صورت تصادفی از حافظه بازپخش نمونه‌برداری می‌شوند. در PER، تجربیات به‌طور یکنواخت نمونه‌برداری نمی‌شوند و یک تجربه با احتمالی متناسب با اولویت آن نمونه‌برداری می‌شود، پس تجربیات با اولویت بالا بیشتر احتمال دارد نمونه‌برداری شوند.

3-5. محاسبه اهداف یادگیری Q برای هر نمونه: برای هر تجربه نمونه‌برداری شده، هدف یادگیری Q برای حالت-عمل جاری تحت سیاست فعلی محاسبه می‌شود.

3-6. آموزش DNN با استفاده از اهداف محاسبه‌شده: وزن‌های DNN با کاهش تفاوت بین اهداف یادگیری Q و مقادیر Q پیش‌بینی‌شده به‌روز می‌شوند. این کار گرادیان نزولی[1] انجام می‌شود.

3-7. به‌روزرسانی اولویت‌های تجربیات نمونه‌برداری شده: پس از هر مرحلهٔ یادگیری، اولویت‌های تجربیات نمونه‌برداری شده، به‌روز می‌شوند. اولویت هر تجربه روی مقدار

[1]Gradiant decent

خطای TD تنظیم می‌شود که تفاوت بین یادگیری هدف Q و مقدار Q پیش‌بینی‌شده است.

4- پایان آموزش:

هنگامی‌که شرایط خاصی مانند حداکثر تعداد اپیزود یا حداقل مقدار ε برآورده شود، آموزش پایان می‌یابد.

تجزیه‌وتحلیل داده‌ها

هایپرپارامترها

برای آموزش تمامی محیط‌های ذکرشده، از پارامترها به‌صورت جدول زیر استفاده نموده‌ایم:

جدول (۱) هایپرپارامترها

ردیف	نام پارامتر	مقدار
1	فاکتور تخفیف	0/99
2	تعداد لایه‌های مخفی شبکه عصبی	256
3	نرخ یادگیری (در شروع)	0/00015
4	حداکثر تکرار یادگیری	300000
5	طول حافظه بازپخش	200000
6	ضریب α محاسبه PER	0/6

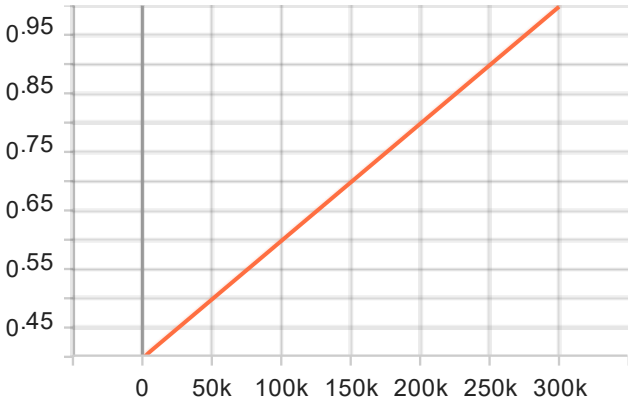
همچنین نمونه‌برداری از حافظه بازپخش، با جای‌گذاری نیست.

سناریو

در هر سه محیط بازی عنوان‌شده، دو الگوریتم DQN PER و DDQN PER را اجرا نمودیم.

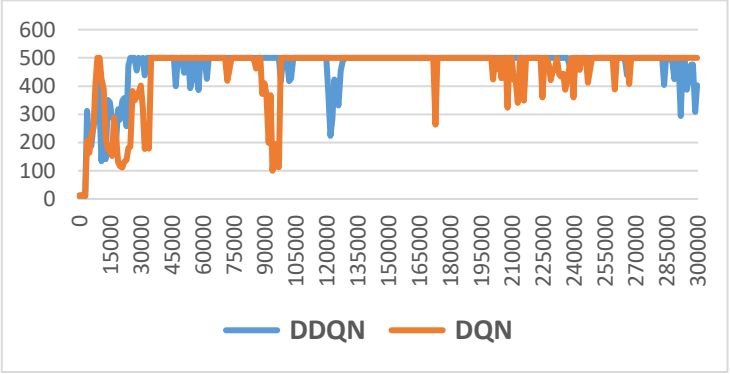
نتایج و تجزیه‌وتحلیل

درروند آموزش هر دو الگوریتم، مقدار ضریب α به‌صورت زیر از صفر تا یک تغییر می‌دهیم:



نمودار ۱: تغییرات مقدار α طی روند آموزش

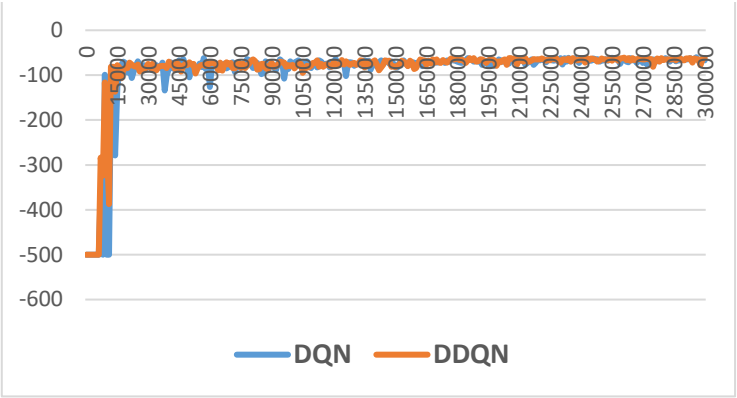
تغییرات پاداش دریافتی در محیط Cart Pole به‌صورت زیر است:



نمودار ۲: تغییرات مقدار پاداش دریافتی طی روند آموزش محیط Cart Pole

با توجه به نمودار، در محیط Cart Pole مقادیر پاداش دریافتی، به وضعیت مناسبی طی هر دو روش رسیده است؛ ولی در روش DDQN PER مقدار نویز در تغییرات پاداش، کمتر است.

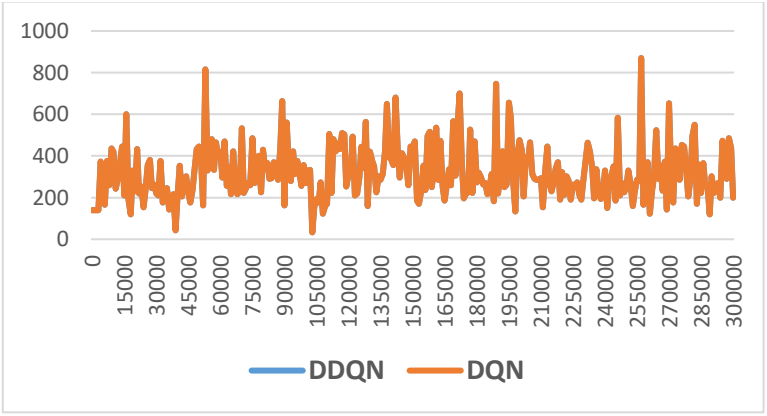
تغییرات پاداش دریافتی در محیط Acrobat به‌صورت زیر است:



نمودار ۳: تغییرات مقدار پاداش دریافتی طی روند آموزش محیط Acrobat

با توجه به نمودار در محیط Acrobat، دو الگوریتم یادشده عملکرد مشابهی دارند، بایستی این نکته را در نظر داشت که در محیط Acrobat نحوه امتیازدهی محیط متفاوت است و درواقع الگوریتم‌های یادشده پاداش دریافتی اولیه را بهبود داده‌اند و پاداش در حالت کلی منفی بوده است.

همچنین تغییرات پاداش در محیط Alien به‌صورت زیر است:



نمودار ۴: تغییرات مقدار پاداش دریافتی طی روند آموزش محیط Alien

در محیط Alien، الگوریتم DQN PER با مشکل گیرکردن در نقطهٔ بهینه محلی[1] روبرو شده است و در واقع روند آموزش آن به دلیل عدم تغییر پاداش، با خطا روبرو شده است.

[1]Local Optimum

این مشکل به دلیل کاوش ناکافی و یا نوسانات زیاد در به‌روزرسانی Q-valueها و در نتیجه باعث ناپایداری الگوریتم و در نهایت سبب گیرکردن در بهینه محلی شده است. در مقابل الگوریتم DDQN PER پیشنهادی در هر اپیزود از بازی، عملکرد مناسبی در پاداش دریافتی داشته است و توانسته است امتیازات به‌مراتب مناسبی در اپیزودهای بازی کسب کند. بایستی توجه داشت که الگوریتم DDQN PER به‌طور کامل از گیرکردن در بهینه محلی مصون نیست؛ ولی با کاهش بیش‌برآورد و نمونه‌برداری مؤثرتر به کاهش احتمال این مشکل کمک می‌کند. همچنین بایستی توجه داشت که ساختار بازی Alien با بازی‌های دیگر به دلیل وجود رقبای هوشمند، متفاوت است و لزوماً روند دریافت پاداش، صعودی نخواهد بود.

نتیجه‌گیری و پیشنهادها

در این مقاله، ما یک رویکرد جدید برای بازی هوشمند ارائه نمودیم که می‌تواند در بازی جنگ و علی‌الخصوص در بازی اول‌شخص تیرانداز نیز کارکرد مناسبی از خود ارائه دهد. درواقع با رویکرد تمرکز بر تجربیات باارزش گذشته و همچنین نمونه‌برداری هدفمند از تجربیات، بهره‌وری افزایش پیدا می‌کند و عملکرد نیز بهبود می‌یابد. همچنین رویکرد پیشنهادی مشکل رایج در الگوریتم‌های یادگیری تقویتی یعنی ناپایداری یادگیری را تا حد مناسبی حل کرده است. برای تحقیقات آینده، می‌توان رویکرد پیشنهادی را با معماری‌های دوئل جهت بهبود آن دربازی‌های همه علیه همه[1] ترکیب کرد.

قدردانی

از همه اســتادان و پژوهشــگرانی که با ارائه نظرات ارزشــمند خود، در ارتقای کیفیت این مقاله ما را یاری کرده‌اند؛ تقدیر و تشکر می‌کنیم.

[1] Deathmatch

منابع

- Mnih, V., Kavukcuoglu, K. (2013). Playing Atari with Deep Reinforcement Learning. NIPS Deep Learning Workshop.
- Mnih, V., Kavukcuoglu, K. (2015). Human-level control through deep reinforcement learning. [Paper presented at the...]
- Lample, G., Chaplot, D. S., & Anandkumar, A. (2016). Playing FPS games with deep reinforcement learning. arXiv preprint arXiv:1609.05521.
- Salehin, A. (2024). Learning To Play Atari Games Using Dueling Q-Learning and Hebbian Plasticity. arXiv preprint arXiv: 2405.13960.
- Querido, G., Sardinha,A. , S. Melo, A. (2023). Learning to Perceive in Deep Model-Free Reinforcement Learning. arXiv preprint arXiv: 2301.03730.
- Wang, K., Bartsch, A., Barati Farimani, A. (2022). Multi-Action Networks Learning. arXiv preprint arXiv: 2209.09329.
- Schaul, T., Quan, J. (2016). Prioritized Experience Replay. Stanford, California, USA. (Note: This reference lacks a publisher. If you have one, please include it.)
- Mnih, V., Kavukcuoglu, K. (2015). Human-level control through deep reinforcement learning. [Paper presented at the...]
- Van Hasselt, H., Guez, A., Silver, D. (2016). Deep Reinforcement Learning with Double Q-learning. arXiv preprint arXiv:1509.06461.
- Mnih, V., Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755.
- Bellemare, M., et al. (2012). The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research, 47, 103-143.
- Foerster, J., et al. (2016). Learning to communicate to solve riddles with deep distributed recurrent q-networks. arXiv preprint arXiv:1602.02672.

- Hausknecht, M., & Stone, P. (2015). Deep recurrent q-learning for partially observable mdps. arXiv preprint arXiv:1507.06527.

- McPartland, M., & Gallagher, M. (2008). Learning to be a bot: Reinforcement learning in shooter games. In Proceedings of the AIIDE Conference (pp. 108-113). AAAI Press.

- Tastan, B., & Sukthankar, G. R. (2011). Learning policies for first person shooter games using inverse reinforcement learning. In Proceedings of the AIIDE Conference (pp. 114-119). AAAI Press.

- van Hasselt, H., Guez, A., & Silver, D. (2016). Deep Reinforcement Learning with Double Q-learning. arXiv preprint arXiv:1509.06461. (This is a duplicate of a previous reference. You can remove one.)

- Wang, Z., de Freitas, N., & Lanctot, M. (2015). Dueling network architectures for deep reinforcement learning. arXiv preprint arXiv:1511.06581.