بیشینه‌سازی امتیاز در بازی تصادفی match-3 با استفاده از یادگیری تقویتی عمیق

علی افروغ و مهدی رعایائی اردکانی*

دانشکدهٔ مهندسی برق و کامپیوتر، دانشگاه تربیت مدرس، تهران، ایران

چکیده

بازی‌های رایانه‌ای در سال‌های اخیر نقش مهمی در توسعهٔ هوش مصنوعی داشته‌اند. روش‌های گوناگون از جمله روش‌های مبتنی‌بر قوانین، جستجوی درختی و یادگیری ماشین (یادگیری نظارت‌شده و یادگیری تقویتی) برای ایجاد عامل‌های هوشمند در بازی‌های گوناگون توسعه یافته‌اند. از میان این پژوهش‌ها، می‌توان به پژوهش‌های Deep Blue در بازی شطرنج و AlphaGo در بازی Go اشاره کرد. AlphaGo اولین برنامهٔ رایانه‌ای است که یک بازی‌کن حرفه‌ای انسانی Go را شکست داد. همچنین، Deep Blue یک سامانهٔ رایانه‌ای حرفه‌ای شطرنج و نخستین برنامه است که در مقابل یک قهرمان جهان، برنده می‌شود. در این مقاله، ما بر روی بازی match-3 تمرکز داریم، که یک بازی محبوب در تلفن‌های همراه و شامل یک فضای حالت تصادفی بسیار بزرگ و تابع پاداش تصادفی است که یادگیری را دشوار می‌کند. در گذشته، پژوهش‌های زیادی در مورد بازی‌های گوناگون، از جمله match-3، انجام شده‌است. هدف اصلی این پژوهش‌ها به‌طور کلی بازی بهینه یا پیش‌بینی دشواری مراحل طراحی‌شده برای بازی‌کنان انسانی بوده‌است. پیش‌بینی دشواری مراحل به توسعه‌دهندگان بازی کمک می‌کند تا کیفیت بازی‌های خود را بهبود بخشند و تجربهٔ کاربری بهتری فراهم کنند. در این مقاله، یک عامل هوشمند بر اساس یادگیری تقویتی عمیق ارائه شده‌که هدف آن به بیشینه رساندن امتیاز در بازی match-3 است. یادگیری تقویتی یکی از شاخه‌های یادگیری ماشین است که عامل از طریق تجربیات خود از تعامل با محیط، سیاست بهینه را برای انتخاب اعمال در فضاهای گوناگون یاد می‌گیرد. در یادگیری تقویتی عمیق، الگوریتم‌های یادگیری تقویتی به‌همراه شبکه‌های عصبی عمیق استفاده می‌شوند. در روش پیشنهادی، سازوکارهای نگاشت گوناگونی برای فضای اعمال و فضای حالت استفاده شده‌است. همچنین، یک ساختار نوآورانه از شبکه‌های عصبی سفارشی‌سازی‌شده برای محیط بازی match-3 پیشنهاد شده‌است تا قابلیت یادگیری فضای حالت بزرگ را به‌دست‌آورد. نوآوری‌های این مقاله را می‌توان بدین شرح خلاصه کرد: روی‌کردی برای نگاشت از فضای اعمال به یک ماتریس دوبعدی ارائه شده که امکان جداکردن اعمال مجاز و غیرمجاز را تسهیل می‌کند. یک روش برای نگاشت از فضای حالت به ورودی شبهٔ عصبی عمیق طراحی شده که با کاهش عمق صافی‌های پیچشی، فضای ورودی را کاهش داده و این‌گونه فرایند یادگیری را بهبود می‌بخشد. همچنین، تابع پاداش از طریق جداکردن پاداش‌های تصادفی از پاداش‌های قطعی، فرایند یادگیری را پایدار کرده‌است. مقایسهٔ روش پیشنهادی با سایر روش‌های موجود، از جمله PPO، DQN، A3C، روش حریصانه و عوامل انسانی، نشان‌دهندهٔ عملکرد برتر روش پیشنهادی در بازی match-3 است.

واژگان کلیدی: یادگیری تقویتی عمیق، بازی تصادفی، match-3، فضای حالت بزرگ

Maximizing Score in Stochastic Match-3 Games Using Reinforcement Learning

Ali Afrougheh And Mehdy Roayaei Ardakany,*
Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

* نویسندهٔ عهده‌دار مکاتبات

* Corresponding author

• تاریخ ارسال مقاله: ۱۴۰۱/۸/۵ • تاریخ پذیرش: ۱۴۰۲/۹/۲۰ • تاریخ انتشار: ۱۴۰۲/۱۲/۲۹ • نوع مطالعه: کاربردی

Abstract

Computer games have played an important role in the development of artificial intelligence in recent years. Throughout the history of artificial intelligence, computer games have been a suitable test environment for evaluating new approaches and algorithms to artificial intelligence. Different methods, including rule-based methods, tree search methods, and machine learning methods (supervised learning and reinforcement learning) have been developed to create intelligent agents in different games. Games have been used as a suitable environment for trial and error, testing different artificial intelligence ideas and algorithms. Among these researches, we can mention the research of Deep Blue in the chess game and AlphaGo in the game Go. AlphaGo is the first computer program to defeat an expert human Go player. Also, Deep Blue is a chess-playing expert system is the first computer program to win a match, against a world champion.

In this paper, we focus on the match-3 game. The match-3 game is a popular game in cell phones, which consists of a very large random state space which makes learning difficult. It also has random reward function which makes learning unstable. Many researches have been done in the past on different games, including match-3. The aim of these researches has generally been to play optimally or to predict the difficulty of stages designed for human players. Predicting the difficulty of stages helps game developers to improve the quality of their games and provide a better experience for users. Based on the approach used, past works can be divided into three main categories including search-based methods, machine learning methods and heuristic methods.

In this paper, an intelligent agent based on deep reinforcement learning is presented, whose goal is to maximize the score in the match-3 game. Reinforcement learning is one of the approaches that has received a lot of attention recently. Reinforcement learning is one of the branches of machine learning in which the agent learns the optimal policy for choosing actions in different spaces through its experiences of interacting with the environment. In deep reinforcement learning, reinforcement learning algorithms are used along with deep neural networks.

In the proposed method, different mapping mechanisms for action space and state space are used. Also, a novel structure of neural network customized for the match-3 game environment has been proposed to achieve the ability to learn large state space. The contributions of this article can be summarized as follow. An approach for mapping the action space to a two-dimensional matrix is presented in which it is possible to easily separate valid and invalid actions. An approach has been designed to map the state space to the input of the deep neural network, which reduces the input space by reducing the depth of the convolutional filter and thus improves the learning process. The reward function has made the learning process stable by separating random rewards from deterministic rewards.

The comparison of the proposed method with other existing methods, including PPO, DQN, A3C, greedy method and human agents shows the superior performance of the proposed method in the match-3 game.

Keywords: deep reinforcement learning, random game, match-3, large state space

۱- مقدمه

در طول تاریخچهٔ هوش مصنوعی، بازی‌ها محیط آزمایشی مناسبی برای ارزیابی روی‌کردهای جدید هوش مصنوعی بوده‌اند. عامل‌های[1] هوشمند با روش‌های گوناگون از جمله روش‌های مبتنی‌بر قانون[2]، جست‌وجوی درخت و یادگیری ماشین[3] (یادگیری با نظارت[4] و یادگیری تقویتی[5]) توسعه داده‌شده‌اند. هدف این عامل‌ها بیشتر انجام بازی به شیوهٔ بهینه بوده‌است. برخی از پژوهش‌ها با روی‌کرد بهینه بازی‌کردن موفق به شکست انسان در بازی‌ها شده‌اند. از جملهٔ این پژوهش‌ها می‌توان به Deep Blue در بازی شطرنج [۱] و AlphaGo در بازی Go [۲] اشاره کرد. در روش Deep Blue از روی‌کرد جست‌وجوی درخت و همچنین، تابع تخمین ارزشی[6] که از تحلیل دانش بازی‌کنان ماهر شطرنج به‌دست‌آمده، و در پژوهش AlphaGo به‌صورت ترکیبی از جست‌وجو و همچنین یادگیری ماشین برای تخمین ارزش حالات استفاده شده‌است.

بازی‌کردن عمومی[7] یک زمینهٔ پژوهشی با هدف توسعهٔ عامل‌های هوشمندی است که تنها با دانستن قوانین بازی بتوانند بازی‌های گوناگونی را انجام دهند. یادگیری تقویتی یکی از روی‌کردهایی است که به‌تازگی در این زمینه توجه زیادی به آن شده و یکی از زیر شاخه‌های یادگیری ماشین است که در آن عامل از طریق تجربه‌های خود از تعامل با محیط[8]، سیاست[9] بهینه برای انتخاب اعمال[10] را یاد می‌گیرد. در یادگیری تقویتی

[1] Agent
[2] Rule-Based
[3] Machine Learning
[4] Supervised Learning
[5] Reinforcement Learning
[6] Value Approximation function
[7] General Game Playing
[8] Environment
[9] Policy
[10] Action

عمیق، الگوریتم‌های یادگیری تقویتی به همراه شبکه‌های عصبی استفاده می‌شوند. پژوهش انجام‌شده توسط مینه و همکاران بر روی بازی‌های آتاری نقطهٔ عطفی در یادگیری تقویتی عمیق بود و الگوریتم DQN[1] را پایه‌گذاری کرد [۳]. در آن پژوهش، DQN از تصویر بازی‌های آتاری به‌عنوان ویژگی‌های[2] محیط استفاده کرده‌است. در نتیجهٔ آن پژوهش، عامل در برخی بازی‌ها توانست به عمل‌کرد بهتری از بازی‌کنان انسانی برسد. پس‌از پژوهش DQN پژوهش‌های گوناگونی از جمله [۴] و [۵] برای بهبود عمل‌کرد و کارایی نمونه‌ها در الگوریتم DQN، و پژوهش [۶] برای موازی‌سازی الگوریتم DQN بر روی هسته‌های پردازنده انجام شدند. در پژوهش [۶] همچنین، الگوریتمی به نام A3C[3] که یک الگوریتم یادگیری تقویتی عمیق سیاست‌محور و بازیگر-منتقد[4] است، معرفی شد.

بازی match-3 یک بازی تک‌نفره و یکی از بازی‌های بسیار محبوب در گوشی‌های تلفن همراه است. این بازی دارای فضای حالت تصادفی[5] و پاداش[6] تصادفی است. فضای حالات در بازی match-3 نیز فضای بزرگی است که احتمال تکرار یک حالت را بسیار کم می‌کند. همچنین، تعداد اعمال در آن به‌نسبت بزرگ است. ویژگی‌های یادشده، بازی match-3 را به محیطی چالش‌برانگیز برای یادگیری تبدیل کرده و همچنین، توسعهٔ عاملی هوشمند برای انجام این بازی را دشوار می‌کند. در این پژوهش، عاملی با استفاده از یادگیری تقویتی عمیق پیشنهاد می‌شود که با استفادهٔ هوشمندانه از لایه‌های کانولوشنی[7] عملکرد مناسبی از خود ارائه می‌دهد.

نوآوری‌های این مقاله را می‌توان به‌اختصار به شکل زیر بیان کرد:

- روی‌کردی برای نگاشت فضای عمل به یک ماتریس دوبعدی ارائه شده که در آن بتوان به‌راحتی اعمال مجاز و غیرمجاز را از هم جدا کرد.
- روی‌کردی برای نگاشت فضای حالت به ورودی شبکهٔ عصبی عمیق طراحی شده که با کاهش عمق صافی کانولوشنی[8] موجب کاهش فضای ورودی و در نتیجه بهبود فرایند یادگیری می‌شود.
- تابع پاداش با جداسازی پاداش‌های تصادفی از پاداش‌های قطعی، موجب پایداری فرایند یادگیری شده‌است.

ساختار ادامهٔ مقاله به شکل زیر است:

در بخش ۲، پژوهش‌های پیشین بر روی بازی match-3 مرور می‌شود. بخش ۳، دربارهٔ مسئله و جزئیات آن است. پس‌از آن، و در بخش ۴، روش پیشنهادی به همراه جزئیات آن شرح داده‌می‌شوند. بخش ۵، نتایج ارزیابی روش پیشنهادی در مقایسه با دیگر روش‌ها را در برمی‌گیرد؛ و در نهایت، بخش ۶، حاوی جمع‌بندی و نظرات پیشنهادی برای پژوهش‌های بعدی است.

۲- مرور کارهای پیشین

پژوهش‌های گوناگونی در گذشته بر روی بازی‌ها از جمله بازی match-3 انجام شده‌است. هدف این پژوهش‌ها به‌طور کلی، بهینه بازی‌کردن یا پیش‌بینی سختی مراحل طراحی‌شده برای بازی‌کن انسانی بوده‌است. پیش‌بینی سختی مراحل به طراحان بازی کمک می‌کند تا کیفیت بازی‌های خود را ارتقا دهند و تجربهٔ بهتری برای کاربران فراهم کنند. کارهای گذشته را بر اساس روی‌کرد استفاده‌شده می‌توان به سه دستهٔ اصلی مبتنی‌بر جستجو، یادگیری ماشین و روش‌های ابتکاری[9] تقسیم‌بندی کرد.

روش‌های مبتنی‌بر جستجو، خود به دو دستهٔ MiniMax و درخت جستجوی مونت کارلو تقسیم می‌شوند. روش MiniMax روشی برای جستجوی بهترین عمل در بازی‌های دونفره است [۷] که بعدها تحت عنوان هرس آلفا-بتا[10] تکامل پیدا کرد [۸]. نخستین عامل هوش مصنوعی که توانست در بازی شطرنج، استاد بزرگ شطرنج را شکست دهد، از روش جستجوی MiniMax استفاده کرد [۱]. همچنین، در بازی چکرز[11] اولین موفقیت در مقابل انسان با روش MiniMax به‌دست‌آمده‌است [۹].

روش جستجوی درخت مونت کارلو یک روش جستجوی درختی است که توانایی جستجو را در عمق بسیاری بیشتری نسبت به MiniMax دارد. معیار روش جستجوی درخت مونت کارلو برای انشعاب یک شاخهٔ عمل، میزان امیدوارکننده[12]بودن آن شاخه است. با ابداع

[1] Deep Q Network
[2] Feature
[3] Asynchronous Advantage Actor Critic
[4] Actor-Critic
[5] Stochastic
[6] Reward
[7] Convolutional
[8] Convolutional filter

[9] Heuristic
[10] Alpha-Beta pruning
[11] Checkers
[12] Promising

و استفاده از این روش، پیشرفت چشم‌گیری در بازی Go حاصل شد [۱۰]، به‌طوری‌که توانایی شکست بهترین بازیکن انسانی در ابعاد کوچک به‌دست‌آمد، اما همچنان توانایی شکست بازی‌کنان برتر انسانی در ابعاد بزرگ به‌دست‌نیامده‌بود. همچنین، در سال ۲۰۱۷، از این روش برای پیش‌بینی نرخ موفقیت انسان[1] در مراحل بازی معروف Candy Crush[2] به‌وسیلهٔ شرکت سازندهٔ این بازی استفاده شد [۱۱]. در بهینه‌ترین حالت پیش‌بینی متوسط موفقیت انسان به میانگین اختلاف ۱.۵ درصد و انحراف معیار ۹.۹۷ درصد با آمار واقعی کاربران در پنجاه مرحلهٔ آزموده‌شده دست یافتند.

توابع ابتکاری کاربرد فراوانی در طراحی عامل‌های بازی دارند؛ به‌ویژه هنگامی‌که هدف تجاری باشد و نیازی به ساخت بهترین عامل نباشد. به‌طور مثال، می‌توان از ساخت حریف در بازی‌های دو یا چندنفره نام برد. هاردار و همکاران در پژوهشی در ارتباط با بازی match-3 با استفاده از توابع ابتکاری و حریصانه[3] عاملی طراحی کردند که هدفش کسب بیشترین امتیاز در تعداد حرکات معیّن بود. برای ارزیابی نتیجه از پنج بازی‌کن انسانی خواسته شده‌بود هرکدام ده بار با برنامه بازی کنند و روش پیشنهادی نیز صد بار اجرا شد. میانگین امتیاز روش پیشنهادی بسیار نزدیک به انسان بود [۱۲].

کارهای گذشته در یادگیری ماشین را می‌توان به دو دستهٔ یادگیری با نظارت و یادگیری تقویتی تقسیم کرد. یادگیری با نظارت در بازی‌ها مانند یک مسئلهٔ طبقه‌بندی عمل می‌کند؛ به این معنی که برچسب‌ها همان عمل‌ها هستند و عامل باید یک حالت بازی را به یک برچسب نگاشت کند.

همچنین، دلیل استفاده از یادگیری تقویتی در بازی‌ها (از جمله همین مقاله) را می‌توان در موارد زیر خلاصه کرد:

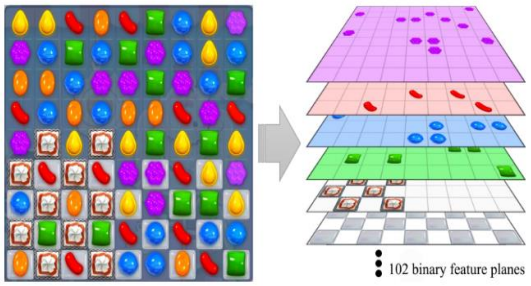
- **انعطاف‌پذیری و تطبیق‌پذیری**: روش‌های یادگیری تقویتی به‌عنوان روی‌کردی برای تصمیم‌گیری در محیط‌های پویا و تغییرات زمانی مناسب هستند. این روش‌ها، توانایی تطبیق و انطباق با تغییرات محیط را دارند و به‌عنوان یک عامل هوشمند، قادر به بهبود عمل‌کرد خود در مواجهه با چالش‌ها و شرایط گوناگون هستند.
- **کاربرد در محیط‌های تصادفی**: بسیاری از بازی‌ها، از جمله بازی‌های match-3 که در این مقاله بررسی شده، دارای جنبه‌های تصادفی هستند. یادگیری تقویتی از طریق جداسازی پاداش قطعی و تصادفی می‌تواند با محیط‌های پیچیده و تصادفی سازگار شود.
- **تعامل مستقیم با محیط**: در بازی‌ها، عامل هوشمند به‌طور مستقیم، با محیط تعامل دارد و اقدامات خود را بر اساس وضعیت فعلی و پاداش‌های دریافتی تصمیم‌گیری می‌کند. روش‌های یادگیری تقویتی می‌توانند به‌عنوان یک راه‌حل مناسب برای این نوع محیط‌های تعاملی عمل کنند.
- **پاداش مبتنی‌بر عمل**: یکی از ویژگی‌های منحصربه‌فرد یادگیری تقویتی استفاده از پاداش مستقیم برای انجام اقدامات است. در بازی‌ها، این ویژگی می‌تواند به‌راحتی پیاده‌سازی شود و از طریق تعریف تابع پاداش مناسب، عامل هوشمند به بهترین راه‌حل‌ها دسترسی پیدا می‌کند.
- **قابلیت یادگیری برخط**: روش‌های یادگیری تقویتی به‌طور معمول، به‌صورت برخط و تعاملی عمل می‌کنند. این به معنای آن است که عامل هوشمند در طول زمان بازی خود (سیاست یادگیری شده) را بهبود می‌دهد.

پورمونن در پژوهشی برای پیش‌بینی سختی مراحل در بازی match-3 از یادگیری عمیق با نظارت استفاده کرد [۱۳]. در این پژوهش از یک شبکهٔ عصبی عمیق کانولوشنی و مجموعه‌دادهٔ تولیدشده به‌وسیلهٔ عامل پژوهش [۱۲] استفاده شده‌است؛ به این صورت که شبکهٔ عصبی، لایه‌های دودویی[4] را از وجود رنگ‌ها در هر خانه به‌عنوان ورودی دریافت می‌کند و یک بردار از احتمال هر عمل را به‌عنوان خروجی می‌دهد. این پژوهش به دقت خوبی در پیش‌بینی حرکات نرسید؛ که از دلایل آن می‌توان به عدم‌توجه به هدف در هر مرحله اشاره کرد.

در ادامه، گادموندسان و همکاران [۱۴] از داده‌های بازی کاربران انسانی و یادگیری عمیق با نظارت برای پیش‌بینی حرکات انسانی استفاده کردند. در معماری شبکهٔ عصبی مورداستفاده در این پژوهش نیز از لایه‌های بی‌شمار کانولوشنی استفاده شد. در این پژوهش بیش از صد صفحهٔ دودویی از ویژگی‌های بازی به‌عنوان ورودی شبکهٔ عصبی استفاده شده‌است. تصویری از صفحات ورودی در شبکهٔ عصبی مورداستفاده در پژوهش

[1] Average Human Success Rate
[2] Candy Crush
[3] Greedy
[4] Binary

یادشده در (شکل- ۱ آمده‌است. این پژوهش توانست دقت خوبی در پیش‌بینی حرکات انسانی و در نهایت پیش‌بینی سختی مرحله برای انسان ارائه دهد. مشکل این روش لزوم وجود مجموعه‌داده و عدم‌امکان استفاده از آن برای عناصر جدید در بازی است که از آن داده‌ای پیش‌از انتشار آن برای کاربران در دسترس نباشد.



(شکل- ۱): تصویری از نحوه مدل‌سازی ورودی در پژوهش [۲۲]

(Figure-1):Input modeling of [22]

دستهٔ دیگر از روش‌های یادگیری ماشین که در بازی‌ها استفاده شده، یادگیری تقویتی است. پیشتر یادگیری تقویتی در بازی‌هایی مانند TD-Gammon [۱۵] و انواع گوناگون بازی‌های آتاری استفاده شده‌بود [۱۶]. یکی از مهم‌ترین این پژوهش‌ها توسط مینه و همکاران انجام شده و باعث معرفی روش DQN شد که نقطهٔ عطفی در یادگیری تقویتی به‌شمارمی‌آید [۱۷]. پس‌از تولد DQN، پژوهش‌های گوناگونی برای بهبود عملکرد این روش ارائه شد [۴، ۵].

شین و همکاران [۱۸] بر روی یک بازی match-3 با روی‌کرد یادگیری تقویتی و الگوریتم A2C[1] پژوهشی را انجام دادند که در آن پنج راهبرد گوناگون مهندسی‌شده به‌صورت دستی به‌عنوان عمل‌های عامل انتخاب شده‌اند. در حقیقت، عامل در این پژوهش با یادگیری تقویتی می‌آموزد در حالات گوناگون کدام راهبرد را انتخاب کند. پاداش در این پژوهش تنها در پایان بازی و درصورتی‌که عامل در حل مرحله موفق شده‌باشد یا نه، با ۱+ و ۱- مشخص می‌شود. این روش که از شبکهٔ عصبی عمیق استفاده می‌کند، صفحات گوناگونی را به‌عنوان ورودی می‌گیرد. در نتایج یادشده، این پژوهش توانسته عملکرد بهتری از عامل تصادفی داشته‌باشد. میانگین اختلاف با دادهٔ واقعی کاربران، در این پژوهش پنج درصد بوده‌است. اما این نتایج، باتوجه‌به مهندسی دستی راهبردها قابل تعمیم به محیط‌های دیگر نیست و همچنین، با افزوده‌شدن ویژگی‌های جدید به بازی ممکن است کارایی خود را ازدست‌دهند.

کمالدینوف و همکاران [۱۹] یک محیط match-3 توسعه دادند و سه الگوریتم یادگیری تقویتی متفاوت DQN، PPO[2] و A3C را در آن پیاده‌سازی کردند. این پژوهش از فیلترکردن اعمال غیرمجاز استفاده نکرده‌است و نتایج آن برای الگوریتم A3C که بهترین نتیجه را داشته، مشابه عمل‌کرد یک عامل تصادفی بوده و در باقی موارد عامل موفق نشده اعمال مجاز را یاد بگیرد.

ناپولیتانو و همکاران در پژوهش [۲۰] بر روی یک بازی match-3 پژوهش دیگری با روی‌کرد یادگیری تقویتی انجام دادند که در آن بر خلافِ بیشتر پژوهش‌های دیگر از شبکهٔ عصبی کاملاً متصل[3] و بدون لایهٔ کانولوشنی استفاده شد. ورودی به شکل یک بردار اعداد که برای هر خانه یک عدد متناظر با رنگ آن خانه و یک عدد ورودی که نمایانگر هدف فعلی است، الگو شده و در مجموع دارای اندازه ۸۲ است. نتایج حاصل‌شده بهتر از عامل تصادفی و هم‌تراز عاملی که به‌صورت حریصانه عمل می‌کند، بوده‌است.

همچنین، در سال‌های اخیر از یادگیری تقویتی عمیق در بازی‌های گوناگون نظیر آتاری [۲۳]، استارکرفت[4] [۲۴، ۲۵] و استراتگو[5] [۲۶] استفاده شده‌است.

جمع‌بندی کارهای گذشته را باتوجه‌به مسئلهٔ مطرح‌شده در این مقاله می‌توان به شکل زیر خلاصه کرد:

- در بیشتر مقالاتی که روی بازی‌های match-3 تمرکز کرده‌اند، هدف یادگیری نحوهٔ بازی عامل انسانی بوده و مسئله به‌صورت یادگیری با نظارت الگو شده‌است. درصورتی‌که در این مقاله، هدف بیشینه‌سازی امتیاز بوده و مسئله به‌صورت یک مسئلهٔ یادگیری تقویتی الگو شده‌است.
- یکی از مهم‌ترین چالش‌های بازی‌های سبک match-3 فضای حالت بسیار بزرگ است که یادگیری را برای هر عامل هوشمندی مشکل و زمان‌بر می‌کند. همچنین، فضای عمل تعریف‌شده برای این بازی نیز به نسبت مسائل دیگر یادگیری تقویتی، فضای بزرگی است.

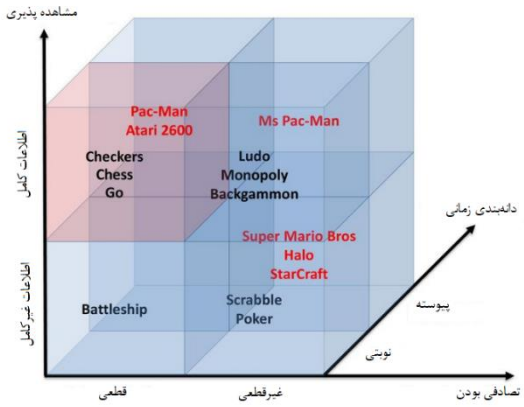
[1] Advantage Actor Critic

[2] Proximal Policy Optimization
[3] Fully-connected
[4] StarCraft
[5] Stratego

- دلیل دیگر پیچیدگی مسئلهٔ بررسی‌شونده در این مقاله، بحث تصادفی‌بودن بازی است که باعث اخلال در یادگیری عامل و ناپایداری[1] فرایند یادگیری می‌شود.

۳- تعریف مسئله

همان‌طورکه در (شکل- ۲ نشان داده شده، بازی‌ها را می‌توان بر حسب سه معیار «میزان مشاهده‌پذیری»[2]، «میزان تصادفی‌بودن»، و «دانه‌بندی زمانی»[3] به هشت دسته تقسیم کرد. در هر دسته به برخی از معروف‌ترین بازی‌های آن اشاره شده‌است. طبق این دسته‌بندی، بازی match-3 در دسته‌بندی غیرقطعی، با اطلاعات کامل و از نظر زمانی پیوسته قرار می‌گیرد. بازی‌های مبتنی‌بر match-3 قواعد پایهٔ یکسانی دارند، اما در بازی‌های گوناگون، قواعد ویژه‌ای می‌تواند به قواعد پایه اضافه شود.



(شکل- ۲): تقسیم‌بندی بازی‌ها

(Figure-2): Game Classification

بازی match-3 از یک جدول تشکیل شده که عناصری به رنگ‌های گوناگون داخل خانه‌های این جدول قرار دارند. تعداد حالات این بازی $m^{w\times h}$، h در آن، تعداد ستون‌ها، w تعداد سطرها و m تعداد رنگ‌هاست. در این پژوهش طول ۸، عرض ۸ و تعداد رنگ ۵ در نظر گرفته شده‌است. بنابراین، اندازهٔ فضای حالت مشاهده 5^{64} است که تعداد بسیار بالایی به‌شمارمی‌آید.

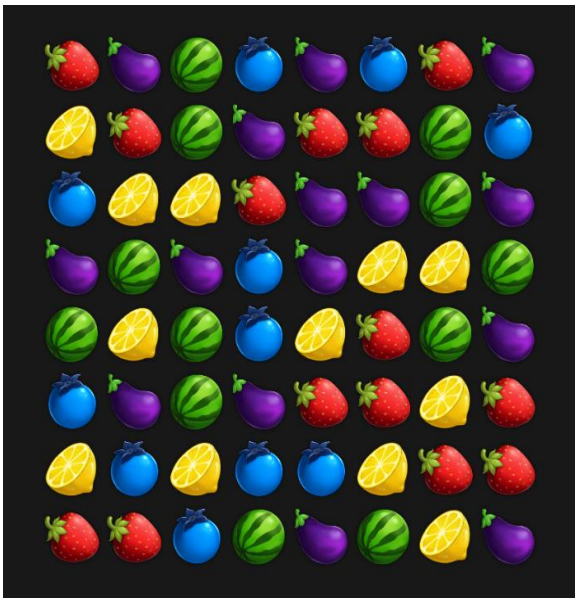
عمل در این بازی به این صورت است که بازی‌کن باید دو خانهٔ مجاور یکدیگر را جابه‌جا کند. اگر این جابه‌جایی منجر به این شود که سه عنصر هم‌رنگ یا بیشتر، در یک سطر یا ستون در مجاورت همدیگر قرار گیرند، این حرکت مجاز، و درغیراین‌صورت حرکت غیرمجاز است. وقتی بازی‌کن حرکت مجاز انجام دهد، مراحل زیر به‌ترتیب اتفاق می‌افتد:

۱) عناصر هم‌رنگی که در سه خانه یا بیشتر در مجاورت یکدیگر بوده‌اند (یا به‌اصطلاح جور[4] شده‌اند)، حذف می‌شوند.

۲) جاذبه در جدول اعمال‌شده و خانه‌های خالی پایینی با عناصر خانه‌های بالایی پر می‌شوند.

۳) در خانه‌هایی که در بالای صفحه خالی می‌مانند، عناصر به رنگ تصادفی ایجاد می‌شود.

مراحل اول تا سوم تا زمانی که دیگر هیچ جورشدنی در جدول اتفاق نیفتد، ادامه پیدا می‌کند. تعداد عمل بالقوه در هر حرکت برابر $(w-1)\times h + (h-1)\times w$ است. البته تمام حرکات مجاز نیست و حرکات مجاز بستگی به حالت فعلی جدول دارد. به‌این‌ترتیب، در این پژوهش اندازهٔ فضای عمل در نهایت، برابر با ۱۱۲ است. تصویری از صفحهٔ بازی match-3 در (شکل- ۳ آمده‌است.



(شکل- ۳): تصویری از محیط بازی match-3

(Figure- 3): A snapshot of match-3 Environment

امتیاز در بازی match-3 به شکل‌های گوناگونی تعریف می‌شود. یکی از رایج‌ترین تعاریف، تعداد عناصر حذف‌شده از جدول در نتیجهٔ هر عمل است که همین تعریف امتیاز در این پژوهش نیز استفاده شده‌است. هدف این پژوهش توسعهٔ یک عامل هوشمند مبتنی‌بر یادگیری تقویتی عمیق برای بیشینه‌سازی امتیاز در تعداد حرکات محدود و این مسئله یک مسئلهٔ NP-Hard است [۲۱].

[1] Instability
[2] Observability
[3] Time granularity
[4] Match

۴-روش پیشنهادی

در این بخش، روش پیشنهادی شامل الگوریتم به‌کاررفته برای یادگیری، ساختار شبکهٔ عصبی مورداستفاده، جزئیات نگاشت فضای حالت و اعمال و طراحی تابع پاداش بیان می‌شود.

۴-۱- الگوریتم

الگوریتم یادگیری استفاده‌شده در این پژوهش DQN است. تغییرات اندکی بر روی الگوریتم DQN داده شده تا توانایی فیلترکردن اعمال غیرمجاز را داشته‌باشد و عامل حین یادگیری تنها اعمال مجاز را انتخاب کند. در **Algorithm 1** الگوریتم مورداستفاده آمده‌است.

در خط یک، مقداردهی اولیهٔ بافر تکرار تجربه و شبکه‌های عصبی انجام شده‌است. در خطوط چهار تا شش، عامل بر اساس حالت فعلی، عملی را با استفاده از شبکهٔ عصبی انتخاب، روی محیط اعمال و در نهایت، پاداش و حالت بعدی را از محیط دریافت می‌کند. تجربهٔ دریافت‌شده در بافرِ تکرار تجربه ذخیره می‌شود.

در این الگوریتم در خط پنج امکان فیلترشدن عمل‌های ممکن برای عامل به الگوریتم پایهٔ DQN اضافه شده‌است؛ به این معنی که مجموعه عمل‌های عامل در هر مرحله ممکن است با مراحل دیگر متفاوت باشد و بسته به حالت فعلی، به‌طور صرف، عمل‌های مجاز به عامل برگردانده‌می‌شود تا بتواند از بین آن‌ها انتخاب انجام دهد.

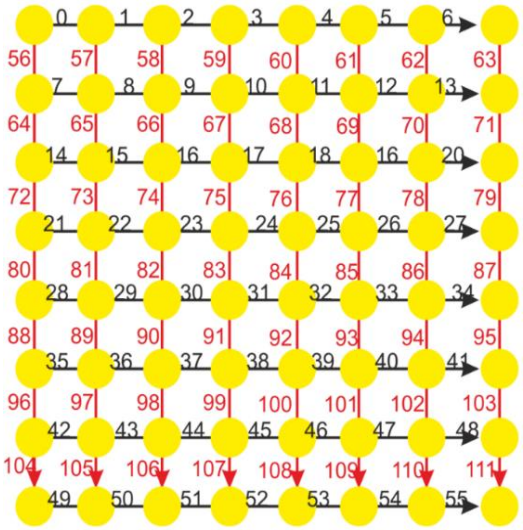
در خطوط هشت تا یازده بر اساس تجربهٔ انتخاب‌شده از بافر تکرار تجربه، شبکهٔ عصبی عامل به‌روزرسانی می‌شود؛ به‌طوری‌که خطای تخمین از ارزش فعلی عمل و حالت، در ادامه کاهش یابد.

همچنین، تفاوت دیگری که در این الگوریتم نسبت به الگوریتم پایه وجود دارد، استفاده از رابطهٔ (۱) در خط نه الگوریتم برای محاسبهٔ پاداش و تبدیل امتیازهای برگردانده‌شده توسط محیط به پاداشی است که تا حد ممکن اهمیت پاداش‌های قطعی را نسبت به پاداش‌های تصادفی برجسته کرده‌باشد.

همان‌طورکه در الگوریتم مشخص شده، از الگوریتم ϵ- حریصانه به‌عنوان روش انتخاب عمل استفاده می‌شود. در این روش، با احتمال $\epsilon - 1$ عملی با بیشترین ارزش انتخاب و با احتمال ϵ اعمال دیگر به‌صورت تصادفی انتخاب می‌شوند. همچنین، از بافر تکرار تجربه[1] برای ازبین‌بردن همبستگی[2] داده‌های ورودی شبکهٔ عصبی و از شبکهٔ هدف[3] برای حل مشکل متغیربودن تابع هدف و پایداری بیشتری یادگیری استفاده شده‌است. وزن‌های این شبکه در طول هر C تکرار ثابت می‌ماند و در پایان این C تکرار با وزن‌های شبکه اصلی به‌روزرسانی می‌شوند.

۴-۲- نگاشت اعمال

هر عمل، جابه‌جایی بالقوهٔ دو عنصر مجاور است. عملی مجاز است که در این جابه‌جایی یک جورشدن دست‌کم سه‌عنصری بسازد. در بازی تعریف‌شده در این مقاله، تفاوتی برای جابه‌جایی دو عنصر از چپ به راست یا راست به چپ وجود ندارد. به‌طور مشابه، تفاوتی نیز بین جابه‌جایی از بالا به پایین یا پایین به بالا نیست. برای هر عمل بالقوه یک شمارهٔ یکتا درنظرگرفته‌شده‌است. تصویر این نگاشت در (شکل- ۴ آمده‌است. همان‌طورکه در این شکل مشخص است، ۱۱۲ عمل بالقوه برای هر حالت بازی تعریف شده‌است.



(شکل- ۴): تصویر نگاشت اعمال به شمارهٔ یکتا

(Figure-4): Mapping of actions to unique numbers

۴-۳- نگاشت فضای حالت

برای نگاشت فضای حالت، شش صفحهٔ دودویی تعریف می‌کنیم. برای هر رنگ یک صفحهٔ دودویی تعریف شده که نشانگر وجود آن رنگ در آن خانه است. بنابراین، پنج صفحه از شش صفحه به این شکل، برای پنج رنگ گوناگون موجود تعریف شده‌است. مثالی از نگاشت یک‌رنگ در (شکل- ۵) آمده‌است.

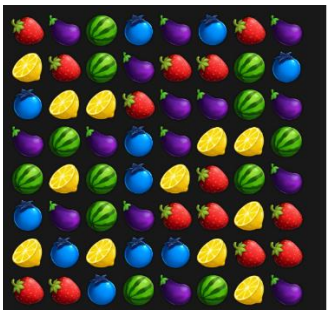
[1] Experience replay

[2] Correlation

[3] Target network

Algorithm 1: DQN with Action Masking

Initialize primary network Q_θ, target network $Q_{\theta'}$, replay buffer $\mathcal{D}$
for each iteration **do**
  for each environment step **do**
    Observe state s_t and select $a_t = \begin{cases} \text{a random action from } A(s) & w.p.\ \epsilon \\ argmax_{a \in A(s)} Q(s,a;\theta) & w.p.\ 1-\epsilon \end{cases}$
    Execute a_t and observe next state s_{t+1} , reward r_t and action mask $A(s_{t+1})$
    Store $(s_t, a_t, r_t, s_{t+1}, A(s_{t+1}))$ in replay buffer $\mathcal{D}$
  for each update step **do**
    Sample $e_t = (s_t, a_t, r_t, s_{t+1}, A(s_{t+1})) \sim \mathcal{D}$
    Compute target Q value: $Q^*(s_t, a_t) \approx r_t + \gamma \max_{a' \in A(s_{t+1})} Q(s_{t+1}, a'; \theta')$
    Perform gradient descent step on $\left(Q^*(s_t, a_t) - Q(s_t, a_t; \theta)\right)^2$
  Update target network parameters every C step: $\theta' \leftarrow \theta$

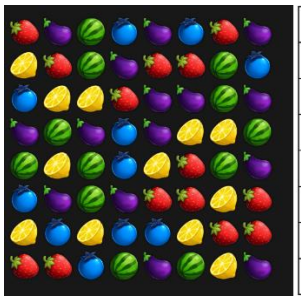


0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	1	1	0	0	0
0	0	1	0	0	0	0	0

(شکل- ۵): مثالی از نگاشت صفحهٔ متناظر با رنگ آبی

(Figure- 5): An example of page corresponding to blue color

صفحهٔ ششم نیز به‌صورت دودویی مشخص می‌کند آیا عنصر داخل هر خانه از این صفحه می‌تواند جزئی از یک حرکت مجاز باشد یا خیر. به بیان دیگر، آیا برای این خانه امکان جابه‌جایی برای جورشدن سه‌تایی وجود دارد یا خیر. تصویر نمونه‌ای از نگاشت صفحهٔ ششم در (شکل- ۶ آمده که شامل تصویری از محیط بازی در سمت چپ و ورودی متناظر در سمت راست است. در خانه‌هایی که می‌توانند جزء حرکات مجاز باشند، عدد یک و در خانه‌های دیگر عدد صفر قرار گرفته‌است. به‌این‌ترتیب، ورودی شبکهٔ عصبی یک ماتریس به ابعاد ۶×۸×۸ است.



0	0	0	0	1	0	1	0
1	0	0	1	1	0	1	0
1	0	0	1	0	0	1	1
0	1	1	0	1	0	1	1
0	1	0	0	1	0	1	0
0	0	0	1	0	1	1	1
0	1	1	1	0	1	1	0
0	0	1	0	0	0	0	0

(شکل- ۶): نحوهٔ نگاشت صفحهٔ ششم

(Figure-6): mapping of page #6

۴-۴- ساختار شبکهٔ عصبی

در این بخش ساختار طراحی‌شده برای شبکهٔ عصبی روش پیشنهادی شرح داده‌می‌شود. هنگامی‌که یک صافی کانولوشنی در ابعاد دوبعدی تعریف می‌شود، باتوجه‌به عمق ورودی، عمق می‌پذیرد؛ به‌عنوان مثال، اگر یک صافی سه در سه داشته‌باشیم، در این مسئله، باتوجه‌به ورودی که عمق شش دارد، صافی در واقع ۳×۳×۶ است.

در این مرحله، یک نوآوری نسبت به کارهای گذشته ایجاد شده‌است. به‌این‌صورت‌که در بازی match-3 وقتی الگویی برای یک رنگ برقرار باشد، برای رنگ دیگر نیز برقرار است؛ به‌عنوان مثال، تفاوتی ندارد یک حرکت که منجر به ساخت الگوی T و در نتیجه جورشدن پنج‌تایی شود، برای چه رنگی اتفاق افتاده‌باشد. استفاده از صافی‌های کانولوشنی به همان شکل که توضیح داده‌شد، به آنها در هر صفحه رنگ وزن‌های متفاوتی می‌دهد. در این صورت، الگوها برای هر رنگ باید جداگانه یادگیری شوند، که در این صورت، دقت یادگیری نیز پایین می‌آید. بنابراین، برای بهبود یادگیری، شبکه به دو بخش تقسیم می‌شود. در بخش نخست، بر روی لایهٔ دودویی هر رنگ به‌طور جداگانه اعمال می‌شود؛ سپس خروجی تمام لایه‌های رنگی در بخش نخست، به‌همراه لایهٔ ششم به‌عنوان ورودی به بخش دوم شبکهٔ عصبی وارد می‌شود که در نهایت، ارزش هر عمل را در خروجی تخمین می‌زند. جزئیات بخش نخست، شبکهٔ عصبی در (جدول- ۱ و جزئیات بخش دوم شبکهٔ عصبی در (جدول- ۲ آمده‌است.

(جدول- ۱): جزئیات بخش نخست ساختار شبکهٔ عصبی
(Table- 1): structure of part1 of NN

لایه	ابعاد	تعداد فیلترها	تابع فعال‌سازی	استراید[2]	پدینگ[1]
ورودی	۱×۸×۸	–	–	–	–
لایهٔ نخست	۵×۵	۲۴	ReLU	۱	۲×۲
لایهٔ دوم	۳×۳	۱۲	ReLU	۱	۱×۱
خروجی	۸×۸× ۱۲	–	–	–	–

(جدول- ۲): جزئیات بخش دوم ساختار شبکهٔ عصبی
(Table- 2): structure of part2 of NN

لایه	ابعاد	نوع	تابع فعال‌سازی
ورودی	۶۱×۸×۸	ورودی	–
نخست	۳۹۰۴	مسطح سازی[3]	–
دوم	۵۱۲	کاملاً متصل	ReLU
خروجی	۱۱۲	کاملاً متصل	–

۵-۴ تعریف تابع پاداش

هدف در این مسئله، بیشینه‌سازی امتیاز در بازی است. این امتیاز نیز به‌صورت تعداد عناصر حذف‌شده (جور شده) در هر عمل است. به‌این‌ترتیب، یک تعریف واضح برای تابع پاداش می‌تواند عناصر حذف‌شده در هر عمل، اما بخشی از پاداش دریافت‌شده ممکن است به دلیل تصادفی‌بودنِ محیط باشد. به این معنی که در هر حرکت، عناصر جدید به‌صورت تصادفی تولید می‌شوند و این عناصر تولیدشده ممکن است منجر به جورشدن عناصر و کسب امتیاز بیشتر شود. در بررسی‌های انجام‌شده مشخص شد که تصادفی‌بودن پاداش باعث ناپایداری در یادگیری شده و عامل توانایی یادگیری الگوهایی را که منجر به پاداش قطعی می‌شود، نخواهدداشت.

در نتیجه، تابع پاداش تعریف‌شده از دو بخش پاداش قطعی و تصادفی تشکیل شده‌است (رابطهٔ ۱). r_d پاداش دریافتی به‌صورت قطعی است که فارغ از رخدادهای تصادفی در بازی است. همچنین، r_s پاداش تصادفی است. برای پایدارسازی فرایند یادگیری، پاداش دریافتی به‌صورت قطعی وزن بیشتری نسبت به پاداش تصادفی داشته‌است. همچنین، در مجموع، امتیاز به‌دست‌آمده در ۰.۰۱ ضرب شده تا پاداش کوچک‌تر از یک و یادگیری پایدارتر باشد.

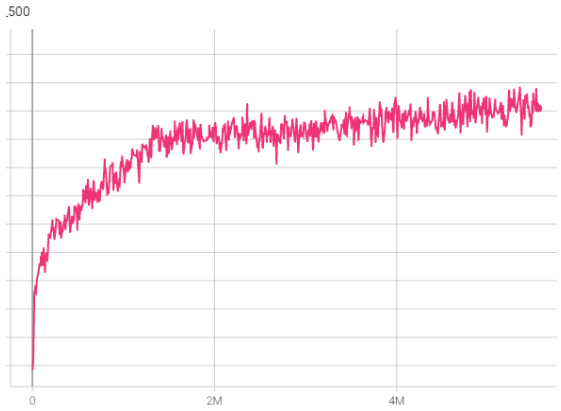
[1] Padding
[2] Stride
[3] Flattening Layer

$$r = (r_d + 0.05\, r_s) \times 0.01 \qquad (۱)$$

۵ - نتایج و ارزیابی

پیاده‌سازی این بازی در محیط بازی‌سازی Unity با زبان برنامه‌نویسی C# انجام شده، محیط پیاده‌سازی‌شده با استفاده از ابزار Unity-ML Agents تبدیل به محیطی برای یادگیری تقویتی و همچنین، قابل‌استفاده در زبان برنامه‌نویسی پایتون شده‌است. پیاده‌سازی شبکهٔ عصبی نیز با کتابخانه pytorch انجام گرفته و طول هر بازی در این محیط پانزده عمل یا حرکت تعریف شده‌است. همچنین، از بهینه‌ساز گرادیان کاهشی تصادفی[4] برای بهینه‌سازی شبکهٔ عصبی استفاده شده‌است. باتوجه‌به ارزیابی‌های انجام‌شده، نرخ کاهش یادگیری تقویتی (λ) برابر ۰.۴ و اندازهٔ دسته[5] ۶۴ درنظرگرفته‌شده‌است. نرخ اکتشاف (ϵ) در آغاز یادگیری برابر ۰.۴ قرار داده‌شده و به‌صورت خطی تا نرخ ۰.۰۵ کاهش می‌یابد. همچنین، باتوجه‌به ارزیابی‌های انجام‌شده، نرخ به‌روزرسانی شبکهٔ هدف (C) پانصد حرکت درنظرگرفته‌شده‌است.

روش پیشنهادی برای ۵/۵ میلیون حرکت اجرا و عملیات یادگیری انجام و مقدار پاداش دریافتی عامل حین یادگیری پایش شد. باتوجه‌به نوفه‌ای‌بودن پاداش دریافتی برای پایش از میانگین متحرک پانصد حرکت اخیر عامل استفاده شد. نمودار مقدار پاداش حین یادگیری در (شکل- ۷) آمده‌است.



(شکل- ۷): نمودار میانگین متحرک پاداش ۵۰۰ حرکت اخیر
(Figure- 7): Average of 500 recent rewards

برای ارزیابی عامل پیشنهادی، از عوامل دیگری استفاده شده‌است. این عوامل عبارتند از:

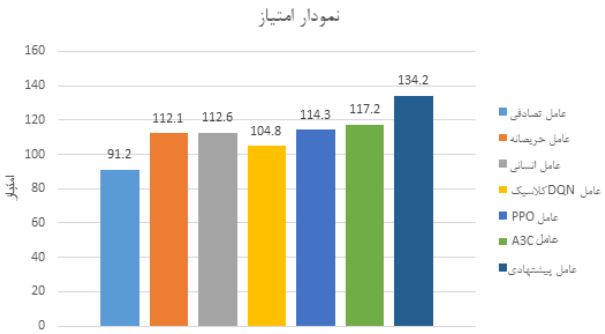
[4] Stochastic Gradient Descent
[5] Batch

- **عامل تصادفی:** این عامل ساده‌ترین نوع عامل است که به‌عنوان یک معیار در دیگر پژوهش‌ها از آن استفاده می‌شود. این عامل در هر حالت از بین اعمال مجاز به‌صورت تصادفی یک عمل انجام می‌دهد.
- **عامل حریصانه:** این عامل در هر حرکت، عملی که بیشترین امتیاز قابل‌مشاهده را داشته‌باشد، انجام می‌دهد.
- **عامل DQN کلاسیک:** عامل دیگری با الگوریتم یادگیری تقویتی عمیق DQN آموزش داده شده‌است. این عامل از ساختار شبکهٔ عصبی که در پژوهش [۲۲] نیز استفاده شده‌است، بهره می‌برد و مشابه عامل پیشنهادی ۵/۵ میلیون تعامل با محیط برای یادگیری داشته‌است.
- **عامل انسانی:** امتیاز عامل انسانی نیز برای سنجش در این پژوهش منظور می‌شود. این عامل برای هزار قسمت[1] بازی کرده‌است. البته میزان خبرگی آن‌ها را در بازی نمی‌توان سنجید و به‌همین‌دلیل، میانگین امتیاز عامل انسانی به‌عنوان نماینده بازی‌کن انسانی انتخاب می‌شود.
- **عامل PPO:** این عامل از الگوریتم یادگیری تقویتی بازیگر-منتقد PPO و همان ساختار مشابه شبکهٔ عصبی برای عامل DQN کلاسیک استفاده می‌کند. تفاوت ساختار شبکهٔ عصبی استفاده‌شده با عامل DQN کلاسیک در لایهٔ پایانی است. عامل PPO به‌عنوان یک عامل بازیگر-منتقد از دو جریان در پایان، یکی برای تخمین حالت-ارزش و دیگری برای احتمال هر عمل استفاده می‌کند.
- **عامل A3C:** این عامل از الگوریتم A3C (یکی از پرکاربردترین الگوریتم‌های یادگیری تقویتی بازیگر-منتقد) برای یادگیری و ساختار شبکهٔ عصبی برابر با عامل PPO استفاده می‌کند.

برای ارزیابی تمامی عوامل پس‌از تکمیل فرایند یادگیری، هر عامل هزار قسمت بازی match-3 را انجام داده‌است. منظور از هر قسمت بازی پانزده حرکت بازی است. در این هزار قسمت همهٔ عامل‌ها در محیط با دانهٔ[2] تصادفی یکسان عمل کرده‌اند تا تصادفی‌بودن محیط تأثیری در نتیجه نگذارد و همگی یک فضای محیط را بازی کنند.

نتایج ارزیابی میانگین امتیازات کسب‌شدهٔ عوامل در (شکل- ۸ آمده‌است. همان‌گونه‌که مشخص است، عامل حریصانه و انسانی بسیار مشابه و نزدیک به هم عمل کرده‌اند. عامل DQN کلاسیک عمل‌کرد بهتر نسبت به عامل تصادفی، اما ضعیف‌تر نسبت به عوامل دیگر داشته‌است. یکی از دلایل این عمل‌کرد ضعیف را می‌توان عدم توانایی شبکهٔ این عامل در یادگیری الگوی مناسب برای بازی دانست. عامل‌های PPO و A3C با اختلاف اندکی نسبت به عامل انسانی و حریصانه بهتر عمل کرده‌اند.

نتایج امتیازات کسب‌شده نشان از عمل‌کرد بهتر عامل پیشنهادی به نسبت دیگر عوامل در بازی match-3 دارد. دلایل این عمل‌کرد را می‌توان نگاشت مناسب فضای حالت و عمل، معماری شبکهٔ انتخاب‌شده و تابع پاداش تعریف‌شده دانست.



(شکل- ۸): نتایج ارزیابی امتیاز عامل‌های گوناگون

(Figure- 8): Evaluation Results of agents

یکی از ارزیابی‌های رایج در این گونه پژوهش‌ها، ارزیابی نسبی عوامل نسبت به عامل تصادفی است که در (جدول- ۳ آمده است. این ارزیابی بیان می‌کند که هر عامل نسبت به عامل تصادفی چند درصد بهتر یا بدتر عمل کرده‌است. همانطورکه در جدول مشخص است، عامل پیشنهادی در حدود ۴۷ درصد عملکرد بهتری نسبت به عامل تصادفی داشته که نسبت به سایر عامل‌ها بهترین عملکرد بود.

(جدول- ۳): مقایسهٔ نسبی امتیاز عامل‌ها نسبت به عامل تصادفی

(Table- 3): Relative comparison of agents compared to the random agent

عامل	حریصانه	انسان	DQN کلاسیک	PPO	A3C	عامل پیشنهادی
امتیاز نسبی	22.91%	23.46%	14.91%	25.32%	28.50%	**47.14%**

[1] Episode
[2] Seed

۶ - جمع‌بندی

در این پژوهش عاملی هوشمند با هدف بهینه بازی‌کردن بازی match-3 معرفی شد. این عامل با استفاده از شبکهٔ عصبی طراحی‌شده، نگاشت‌های فضای حالت و عمل ابتکاری، و تابع پاداش متفاوت، توانست عمل‌کرد بهتری در این بازی به نسبت دیگر عامل‌ها داشته‌باشد. استفاده از یادگیری تقویتی در محیطی که فضای حالت تصادفی و بزرگ و همچنین پاداش تصادفی دارند، از موضوعات موردتوجه و چالش‌برانگیز حوزهٔ یادگیری تقویتی است. بیشترِ موفقیت‌های یادگیری تقویتی در محیط‌های قطعی بوده‌است.

یکی از چالش‌های مهم مسئلهٔ موردبررسی، بحث پایداری الگوی پیشنهادی در محیطی است که دارای ویژگی تصادفی‌بودن است. در این بازی، به دلیل مشخص‌نبودن خانه‌های جدیدی که از بالا وارد صفحه بازی می‌شود، امتیاز دریافتی برای یک عمل نه‌تنها به خود آن عمل، بلکه تا حد زیادی به خانه‌های جدیدی که به‌طور تصادفی وارد بازی می‌شود، بستگی دارد. در بررسی‌های انجام‌شده مشخص شد که تصادفی‌بودن امتیاز باعث ناپایداری در یادگیری شده و عامل توانایی یادگیری الگوهایی را که منجر به پاداش قطعی می‌شود، نخواهدداشت. همچنین، دلیل دیگری که برای عدم پایداری روش پیشنهادی وجود دارد و ویژهٔ این مسئله هم نیست، وجود همبستگی بین داده‌های ورودی شبکهٔ عصبی است که به‌عنوان تجربه‌های تعامل عامل برای یادگیری به این شبکه داده می‌شود.

برای مقابله با دلیل نخست، چالش پایداری، تابع پاداش با جداسازی امتیازهای تصادفی از امتیازهای قطعی، موجب پایداری فرایند یادگیری شده‌است؛ به‌این‌صورت‌که برای پایدارسازی فرایند یادگیری، امتیاز دریافتی قطعی وزن بیشتری نسبت به امتیاز تصادفی داشته‌است. همچنین، در مجموع، امتیاز به‌دست‌آمده در ۰.۰۱ ضرب شده تا پاداش کوچک‌تر از یک و یادگیری پایدارتر باشد. استفاده از ضریب کوچک برای پاداش تصادفی موثر بوده، زیرا این روی‌کرد اثرات تصادفی‌بودن را در محیط کاهش می‌دهد و باعث افزایش پایداری در نتایج به‌دست‌آمده می‌شود. همچنین، برای مقابله با دلیل دوم این چالش، از روی‌کردی متداول در یادگیری تقویتی استفاده شده‌است. در واقع، از بافر تکرار تجربه برای ازبین‌بردن همبستگی داده‌های ورودی شبکهٔ عصبی و از شبکهٔ هدف برای حل مشکل متغیربودن تابع هدف و پایداری بیشتری فرایند یادگیری بهره گرفته شده‌است.

البته در کنار مزایای روش پیشنهادی، ضعف روش پیشنهادی، خاص‌شدن راه‌حل نسبت به دیگر عامل‌ها بابت ساختار شبکهٔ عصبی آن است که برای بازی match-3 تنظیم شده‌است.

در پژوهش‌های آینده می‌توان از نوع پیچیده‌تر محیط بازی match-3 که به‌عنوان مثال، حاوی شتاب‌دهنده[1] و موانع[2] است، استفاده کرد. از این پژوهش نیز می‌توان به‌عنوان معیاری برای عوامل بعدی که برای بازی match-3 توسعه داده‌می‌شود، استفاده کرد. استفاده از این شبکهٔ عصبی با الگوریتم‌های دیگر مانند PPO و A3C می‌تواند موضوع پژوهش‌های آینده باشد.

۷- مراجع 7-Refrence

[1] M. Campbell, A. J. Hoane, and F. H. Hsu, "Deep Blue," *Artificial intelligence*, vol. 134, no. 1–2, pp. 57–83, 2002.

[2] D. Silver et al., Mastering the game of Go with deep neural networks and tree search, *Nature*, *vol. 529*, no. 7587, pp. 484–489, 2016.

[3] V. Mnih et al., Human-level control through deep reinforcement learning, *Nature*, vol. 518, no.7540, pp. 529–533, 2015.

[4] H. Van Hasselt, A. Guez, and D. Silver, Deep Reinforcement Learning with Double Q-Learning, *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, Mar. 2016.

[5] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Frcitas, *Dueling Network Architectures for Deep Reinforcement Learning, in 33rd International Conference on Machine Learning, ICML 2016, 2016*, vol. 4, no. 9, pp. 2939–2947.

[6] V. Mnih et al., Asynchronous Methods for Deep Reinforcement Learning, in Proceedings of the 33rd International Conference on Machine Learning, 2016, vol. 48, pp. 1928–1937.

[7] J. v. Neumann, Zur Theorie der Gesellschaftsspiele, *Math. Ann.*, vol. 100, no. 1, pp. 295–320, Dec. 1928.

[8] D. Knuth, R. M. A., An analysis of alpha-beta pruning, *An analysis of alpha-beta pruning*, vol. 6, no. *4*, pp. 293–326, 1975.

[9] J. Schaeffer, R. Lake, P. Lu, M. B.A., Chinook

[1] Booster
[2] Obstacle

the world man-machine checkers champion, *AI magazine*, vol. 17(1), 1996.

[10] M. Enzenberger, M. Müller, B. Arneson, and R. Segal, FUEGO-An open-source framework for board games and go engine based on Monte Carlo tree search, *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 4, pp. 259–270, 2010.

[12] D. Hadar and O. Samuel, Crushing Candy Crush - An AI Project, Hebrew University of Jerusalem, 2015.

[13] E. R. Poromaa, Crushing Candy Crush, KTH Royal Inst. Technol., Stockholm, Sweden, 2017.

[14] S. Purmonen, Predicting game level difficulty using deep neural networks, KTH Royal Institute of Technology, Stockholm, Sweden, 2017.

[15] C. Tesau and G. Tesau, Temporal Difference Learning and TD-Gammon, *Communications of the ACM*, vol. 38, no. 3, pp. 58–68, 1995.

[16] V. Mnih et al., Playing Atari with Deep Reinforcement Learning, arXiv Prepr. arXiv1312.5602, 2013.

[17] V. Mnih et al., Human-level control through deep reinforcement learning, *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[18] Y. Shin, J. Kim, K. Jin, and Y. Bin Kim, Playtesting in Match 3 Game Using Strategic Plays via Reinforcement Learning, *IEEE Access*, vol. 8, pp. 51593–51600, 2020.

[19] I. Kamaldinov and I. Makarov, Deep reinforcement learning methods in match-3 game, *in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11832 LNCS, pp. 51–62, 2019.

[20] N. Napolitano, Testing match-3 video games with Deep Reinforcement Learning, arXiv, 2020.

[21] L. Gualà, S. Leucci, and E. Natale, Bejeweled, candy crush and other match-three games are (NP-)hard, *In 2014 IEEE Conference on Computational Intelligence and Games, CIG*, pp. 1–21, 2014.

[22] S. F. Gudmundsson et al., Human-Like Playtesting with Deep Learning, *in IEEE Conference on Computational Intelligence and Games, CIG*, 2018, vol. 2018.

[23] L. Kaiser, M. Babaeizadeh, P. Milos, et al., Model Based Reinforcement Learning for Atari. In International Conference on Learning Representations, 2019.

[24] O. Vinyals, I. Babuschkin, W. M. Wojciech Czarnecki, M. Mathieu, A. Dudzik, J. Chung, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning, *Nature 575*, no. 7782 (2019): 350-354.

[25] R. Z., Liu, Pang, Z. Y. Meng, W. Wang, Y. Yu, T., On efficient reinforcement learning for full-length game of StarCraft ii, *Journal of Artificial Intelligence Research*, 75, 213-260, 2022.

[26] J. Perolat, B. De Vylder, D. Hennes, E. Tarassov, E., F. Strub, V. de Boer, Mastering the game of Stratego with model-free multiagent reinforcement learning. Science, 378 (6623), 990-996, 2022.

علی افروغه، کارشناسی و کارشناسی ارشد خود را در سال‌های ۱۳۹۸ و ۱۴۰۱ در رشته مهندسی نرم‌افزار به‌ترتیب در دانشگاه‌های علم و فرهنگ و تربیت‌مدرس به اتمام رسانده است. زمینهٔ پژوهشی موردعلاقهٔ ایشان، کاربرد یادگیری تقویتی در بازی‌ها است.

نشانی رایانامه ایشان عبارت است از:

ali74afrougheh@gmail.com

مهدی رعایائی اردکانی، کارشناسی، کارشناسی ارشد و دکترای خود را به‌ترتیب در سال‌های ۱۳۸۷، ۱۳۸۹، ۱۳۹۵ در دانشگاه صنعتی امیرکبیر به اتمام رسانده‌است. ایشان هم‌اکنون استادیار دانشگاه تربیت مدرس است. علاقهٔ پژوهشی ایشان، یادگیری تقویتی، پردازش متن و تحلیل شبکه‌های اجتماعی است.

نشانی رایانامه ایشان عبارت است از:

mroayaei@modares.ac.ir