**یک عملگر نقاد پیشرفته برای یادگیری عامل در محیط**

**بازی فوتبال گوگل**

**شقایق صفاری1،\*، مرتضی دٌرّی‌گیو۲، فرزین یغمایی۳**

1. **دانشجوی دکتری، دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان**

[shaqayeq.saffari@semnan.ac.ir](mailto:shaqayeq.saffari@semnan.ac.ir)

1. **استادیار دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان**

[dorrigiv@semnan.ac.ir](mailto:dorrigiv@semnan.ac.ir)

1. **دانشیار دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان**

[f\_yaghmaee@semnan.ac.ir](mailto:f_yaghmaee@semnan.ac.ir)

چکیده

این مقاله یک الگوریتم یادگیری تقویتی عمیق را که عملگر نقاد[[1]](#footnote-1) پیشرفته نامیده می‌شود و بر اساس الگوریتم عملگر نقاد است، برای آموزش هوش‌مصنوعی بازی فوتبال ارائه می‌دهد. از آن‌جایی‌که گاهی در برخی حالات، برخی اقدامات لزومی ندارند که اعمال شوند، بنابراین این مدل به منظور دستیابی به تخمین ارزش دقیق و انتخاب صحیح عمل در فضایی پیوسته معرفی شده است. لذا وجود معیاری برای ارزیابی میزان اهمیت و مزیت هر عمل در هر حالت می‌تواند پارامتر ارزشمندی باشد. از این رو با توجه به مزایای مدل [[2]](#footnote-2)Dueling DQN و عملگر نقاد، مدل ارائه شده از ترکیب شبکه Dueling DQN در معماری عملگر نقاد استفاده کرده است. هدف، بهبود عملکرد الگوریتم عملگر نقاد معمولی در یک محیط پیچیده است. مطالعه موردی از اموزش هوش‌مصنوعی بازی، بازی فوتبال است. نتایج تایید می‌کند که الگوریتم بهبود یافته به طور موثر عملکرد را از نظر میانگین پاداش، نسبت به هر یک از دو مدل Dueling DQN و عملگر نقاد به صورت جداگانه بهبود می‌دهد. تجزیه و تحلیل در مورد چگونگی بهبود عملکرد مدل ارائه شده نیز، در بازی فوتبال ارائه شده است.

**کلمات کليدي: یادگیری تقویتی عمیق، بازی فوتبال، عملگر نقاد، Dueling DQN، عملگر نقاد پیشرفته**

**1- مقدمه**

یادگیری تقویتی در طیف وسیعی از زمينه‌ها از جمله بازی، روباتیک و پردازش زبان طبیعی به عملکرد چشمگیری دست یافته است. بیشتر دستاوردهای هیجان‌انگیز اخیر با ترکیب یادگیری عمیق و یادگیری تقویتی به دست آمده است که به عنوان یادگیری تقویتی عمیق شناخته می‌شود [1].

عوامل یادگیری تقویتی عمیق در بازی‌های مختلف از جمله آتاری،Go ، Poker و StarCraft به عملکرد چشمگیری دست یافته‌اند [2]. پژوهش‌هاي انجام گرفته در زمينه یادگیری تقویتی عمیق در بازی‌های ویدیویی پیچیده مانند StarCraftII و Dota2 به عملکرد فوق انسانی دست یافته‌اند. با این حال، سیستم‌های یادگیری تقویتی عمیق کنونی هنوز از چالش‌های پاداش‌های پراکنده، محیط‌های تصادفی و غیره رنج می‌برند [3]. آن‌ها اغلب با چند اشکال برای تحقیق همراه هستند. به عنوان مثال، حل آن‌ها برای الگوریتم‌های پیشرفته بسیار آسان است، یا نیاز به دسترسی به مقادیر زیادی از منابع محاسباتی دارند. در عین حال، ممکن است در برخي بازي‌ها مانند Go یا Chess قطعی باشند یا حتی ممکن است یک مدل شناخته شده از محیط وجود داشته باشد. به طور مشابه، بسیاری از محیط‌های یادگیری ذاتاً تنها با مدل‌سازی تعامل یک عامل با یک محیط ثابت یا بر یک جنبه واحد از یادگیری تقویتی، مانند کنترل مداوم تمرکز می‌کنند. همچنین، محیط‌های آموزشی ممکن است که مجوزهای محدودکننده داشته باشند. از این رو، محیط فوتبال گوگل نه تنها از نظر یادگیری چالش برانگیز است، بلکه از نظر دشواری قابل تنظیم بوده و برای پژوهش هم از نظر مجوز و هم از نظر منابع محاسباتی مورد نیاز قابل دسترسی است [4]. در جستجوی رسیدگی به این چالش‌ها، در پژوهش حاضر از یک بازی ویدیویی فوتبال، به عنوان مثال، محیط فوتبال گوگل به عنوان بستر آزمایشی استفاده شده است.

در حالی که یادگیری تقویتی می‌تواند مدل‌های یادگیری را برای یادگیری رفتار فعال کند، مدل‌ها می‌توانند به طور قابل توجهی از نظر محاسباتی هزينه بيشتري از کنترل‌کننده‌های کلاسیک یا الگوریتم‌های مبتنی بر اکتشاف باشند، به‌ویژه اگر چندین مدل مختلف نیاز به استقرار داشته باشند. برنامه‌های کاربردی گسترده‌تر یادگیری ماشین می‌توانند از منابع محاسباتی قدرتمند یا اختصاصی بهره ببرند. اما هزینه‌های استنتاج زمان اجرا، یکي از ملاحظات مهم برای بازی‌ها هنگام بودجه‌بندی برای هوش‌مصنوعی و تصوير كردن[[3]](#footnote-3) است. به‌ویژه زمانی که انتظار می‌رود که بازی‌ها بر روی مجموعه متنوعی از پیکربندی‌های سخت افزاری ممکن اجرا شوند. هزینه‌های محاسباتی بالا در نهایت می‌تواند مانع دسترسی به ابزارهای یادگیری تقویتی در توسعه و استقرار شود. معمولاً هزینه یک مدل یادگیری تقویتی عمیق متناسب با پیچیدگی مدل است. نتایج بررسی‌ها نشان داده است، برای کلاسی از الگوریتم‌های یادگیری تقویتی، عملگر نقاد، اندازه و هزینه زمان اجرای عملگرهای آموخته شده، اغلب می‌تواند به طور قابل توجهی کاهش یابد، در حالی که برابری عملکرد را با شبکه‌های بزرگ‌تر حفظ می‌كنند [5].

با توجه به ساز و كار پاداش، الگوریتم‌های یادگیری تقویتی یاد می‌گیرند که مقادیر وضعیت‌ها و اقدامات را تخمین بزنند و از مقادیر آموخته‌شده برای به‌روزرسانی بهینه‌سازی خط‌مشی استفاده کنند. روش‌های عملگر نقاد که در حال حاضر جزو رایج‌ترین کلاس‌های الگوریتم‌های یادگیری تقویتی عمیق بدون مدل هستند، با جداسازی توابع نشان‌دهنده عملگر (خط‌مشی یادگیری تقویتی) و نقاد (برآورد کننده تابع ارزش) مشخص می‌شوند. این موضوع به روش‌های عملگر نقاد اجازه می‌دهد، تا از نظر تئوری از بهره‌وری نمونه بهبود یافته حاصل از رویکردهای یادگیری تقویتی مبتنی بر ارزش، مانند یادگیری Q بهره‌مند شوند و در عین حال از توانایی استفاده از رویکردهای مبتنی بر خط‌مشی مانند گرادیان خط‌مشی، از امکان یادگیری در حوزه‌هایی با اقدامات مستمر نیز استفاده شوند [5]. از طرفی آموزش هوش‌مصنوعی، به دلیل محدودیت‌های الگوریتم عملگر نقاد معمولی به تعداد زیادی اجرای دور[[4]](#footnote-4) برای دستیابی به عملکرد رضایت‌بخش نیاز دارد [6, 7, 8]. به عنوان مثال، برآوردگر مزیت مقدار واحدی را برمی‌گرداند که منجر به دقت محدود می‌شود و انتخاب عمل تصادفی فاقد اثربخشی است.

در این مقاله، طراحی بهبود یافته الگوریتم عملگر نقاد، یعنی عملگر نقاد پیشرفته، برای همگرایی بهتر آموزش هوش‌مصنوعی پیشنهاد شده است. همچنين برای یادگیری خط‌مشی در انجام بازی، از یک معماری عملگر نقاد استفاده شد که با معماری Dueling DQN ادغام شده است. بدین صورت نقاد توانست دقت بیشتری در تخمین ارزش‌های هر حالت خاص و در نهایت به‌روزرسانی عملگر داشته باشد. چنین رویکردی برای بازی فوتبال اعمال می‌شود، تا تأیید شود که هوش‌مصنوعی می‌تواند سریع یاد بگیرد، بهتر بازی کند و با برنامه آموزشی مبتنی بر عملگر نقاد پیشرفته استحکام بیشتری داشته باشد. سهم اصلی این مقاله بهبود الگوریتم عملگر نقاد معمولی است که منجر به عملکرد بهتر در محیط‌های پیچیده می‌شود.

به منظور ارزیابی تأثیر هر یک از مدل‌ها و ادغام آن‌ها و سایر تنظیمات در بهبود عملکرد یادگیری خط‌مشی در انجام بازی، بازی انتها به انتها برای وظیفه کسب پاداش بیشتر که به طور پیش‌فرض از یک عملگر نقاد ساده و Duelling DQN برای یادگیری خط‌مشی استفاده می‌کرد، به‌صورت تکی و پس از ادغام دومدل، ارزیابی شد و در نهایت نتایج مقایسه شدند. نتایج نشان داد که روش پیشنهادی توانسته است، بهبود چشم‌گیری در عملکرد نهایی آموزش بازی ایجاد کند.

**2- پیشینه تحقیق**

در سال ۲۰۱۶، برنامه کامپيوتری توسعه یافته توسط تیم DeepMind گوگل، توانست فردی را که ۱۸ مرتبه قهرمان جهانی بازی Go شده بود، شکست دهد و قدرت هوش‌مصنوعی را آشکار کرد. در سال ۲۰۱۷، Alpha Go Zero الگوریتم یادگیری تقویتی عمیق را مورد استفاده قرار داد و نسل قبلی Go را در مدت کوتاهی شکست داده و قدرت یادگیری تقویتی عمیق را آشکار کرد [9].

با توسعه یادگیری تقویتی عمیق، روش استخراج ویژگی انتها به انتها، بر روش استخراج مرسوم ویژگی‌ها رجحان يافته است. به‌طوری که هوش‌مصنوعی می‌تواند کار بازی را بدون دخالت انسان انجام دهد. هوش‌مصنوعی مبتنی بر یادگیری تقویتی عمیق، حتی در برخی از بازی‌های آتاری از بازیکنان برتر انسانی پیشی گرفت [10]. با این حال، بازی‌های آتاری در مقایسه با کاربردهای عملی در زندگی واقعی، عموماً بسیار ساده هستند. از این رو تیم Deep Mind، خیلی زود محیط یادگیری StarCraftII را به عنوان یک بستر آزمایشی جدید برای مطالعات یادگیری تقویتی عمیق ارائه کرد. StarCraftII شامل ادراک و تخمین پویا، بازی با اطلاعات ناقص و مشکلات همکاری چندعاملی است که منجر به محیط آزمایشی عملی‌تر می‌شود، به‌نحوی که تصمیم‌گیری به موقعیت‌های واقعی بسیار نزدیک‌تر می‌شود. البته جدیدترین هوش‌مصنوعی StarCraftII در شرایط خاص توانسته است، برترین بازیکنان انسانی را شکست دهد [11].

در حوزه‌های بازی، شبکه Q عمیق قادر است، خط‌مشی‌های کنترلی را مستقیماً از ورودی‌های با ابعاد بالا در بازی‌های آتاری یاد بگیرد و AlphaGo قهرمان بازی Go جهان را شکست داده است[12]. در مرجع [2]، برای یادگیری نبردها در ViZDoom یک روش عملگر نقاد با منطقه اعتماد فاکتور کرونکر ارائه شده است. نتایج تجربی نیز، نشان داده است که روش‌های یادگیری تقویتی عمیق، عوامل مبارزه در این سناریوها را با موفقیت آموزش می‌دهند. علاوه بر این، عامل‌ها به طور قابل توجهی از عوامل [[5]](#footnote-5)A2C با یک حاشیه قابل توجه بهتر عمل می‌کنند. در مرجع [3] توزیع مزیت، تابع محدودیت نرمال و اکتشاف مبتنی بر اطمینان برای بهبود الگوریتم عملگر نقاد، ارائه شد. این رویکرد روی محیط بازی StarCraftII استفاده شد. نتایج تأیید می‌کنند که الگوریتم بهبودیافته به طور مؤثر عملکرد را از نظر نرخ هم‌گرایی، حداکثر پاداش، میانگین پاداش در هر ۱۰۰ قسمت و زمان رسیدن به یک پاداش خاص بهبود می‌بخشد. در مرجع [5] از الگوریتم عملگر نقاد در ۹ بازی مستقل استفاده شده است و برای بهبود عملکرد آن، با تفکیک معماری عملگر و نقاد از عملگرهای کوچکتر تا ۱ نورونی نیز استفاده کرده است که نشان داد با کاهش متوسط ۷۷ درصدی، منجر به کاهش تعداد وزن‌های شبکه نسبت به دیگر الگوریتم‌های عملگر نقاد شده است.

**3- راهکار پیشنهادی**

در ايا بخش ابتدا محیط بازی مورد استفاده و سپس هر یک از اجزای الگوریتم عملگر نقاد و روش DQN Duelling به اختصار شرح داده می‌شوند. سپس به شرح تنظیمات مدل ارائه شده پرداخته می‌شود.

**1-3- محیط بازی**

محیط بازی فوتبال گوگل در مرجع [4] ارائه شده است. در یادگیری تقویتی عمیق، حالت، عمل و پاداش مهم‌ترین عناصر هستند [3]. از این رو عملکرد هوش‌مصنوعی در بازی فوتبال با استفاده از این عناصر در عملگر نقاد پیشرفته بهبود یافته است.

* **حالت:** مجموعه کاملی از داده‌ها که پس از انجام اقدامات توسط محیط بازگردانده می‌شوند، به عنوان حالت تعریف می‌شود. از سوی دیگر، هر تغییر حالتی را که به عنوان ورودی به الگوریتم ارائه می‌شود، به عنوان مشاهدات یا نمایش تعریف می‌شود [3]. تعریف حالت شامل اطلاعاتی مانند موقعیت و مالکیت توپ، مختصات همه بازیکنان، بازیکن فعال، وضعیت بازی (سطح خستگی بازیکنان، کارت‌های زرد، امتیاز و غیره) و فریم پیکسل فعلی است [4]. در این محیط از نمایش نقشه کوچک استفاده شده است که می‌توان حالت‌ها را در چندین مرحله زمانی متوالی (مثلاً برای تعیین جهت توپ) روی هم قرار داد، یا از روی هم جدا کرد، یعنی فقط با مرحله زمانی فعلی مطابقت دارد.
* **نقشه کوچک:** نمایشی شامل چهار ماتریس ۷۲×۹۶ است که اطلاعات مربوط به تیم میزبان، تیم میهمان، توپ و بازیکن فعال را رمزگذاری می‌کند. رمزگذاری دودویی است و نشان می‌دهد که بازیکن یا توپ در مختصات مربوطه وجود دارد یا خیر.
* **عمل:** اقداماتی که برای یک نماینده (بازیکن) در دسترس است، در جدول ۱ نمایش داده شده است. این اقدامات شامل حرکات استاندارد (در ۸ جهت) و روش‌های مختلف ضربه زدن به توپ (پاس‌های کوتاه و بلند، شوت، و پاس‌های بلند است که در طول مسیر به راحتی قابل رهگیری نمی‌باشند) می‌باشد. همچنین، بازیکنان می‌توانند به سرعت بپرند (که بر میزان خستگی آن‌ها تاثیر می‌گذارد)، سعی کنند با تکل بلند برای توپ حائل ایجاد کنند و یا اگر توپ را در اختیار دارند، دریبل کنند.
* **پاداش:** موتور فوتبال شامل نوعی پاداش است که مربوط به پاداش طبیعی است. هر تیم یک جایزه ۱+ در هنگام به ثمر رساندن یک گل و یک پاداش ۱- در هنگام دریافت یک گل به تیم مقابل دریافت می‌کند. مشاهده پاداش در مراحل اولیه تمرین دشوار است، زیرا ممکن است به دنباله‌ای طولانی از رویدادهای متوالی، مانند غلبه بر دفاع یک حریف بالقوه قوی و گلزنی در برابر دروازه‌بان نیاز داشته باشد.

**جدول ۱- مجموعه عمل‌ها**

|  |  |  |  |
| --- | --- | --- | --- |
| راست | چپ | پایین | بالا |
| پایین- راست | چپ- پایین | بالا- راست | بالا- چپ |
| شوت | پاس طولانی | پاس بلند | پاس کوتاه |
| توقف- دریبل | دریبل | تکل | ایجاد حائل برای توپ |
| توقف کامل | توقف از دویدن سریع | بدوت حرکت | دویدن سریع |

**2-3- عملگر نقاد**

عملگر نقاد [12] با استفاده از یک شبکه عصبی که به ترتیب احتمالات عمل و ارزش انتقادی را تولید می‌کند، مدل‌سازی می‌شود. برای آموزش عامل با استفاده مدل عملگر نقاد مراحل زیر طی می‌شود:

عامل روی محیط اجرا می‌شود، تا داده‌های آموزشی در هر قسمت جمع‌آوری شود. بازده مورد انتظار در هر مرحله زمانی محاسبه می‌شود. ضرر برای مدل ترکیبی عملگر نقاد محاسبه می‌شود. شیب‌ها محاسبه شده و پارامترهای شبکه به روز می‌شود. سپس این مراحل تکرار می‌شود تا به معیار موفقیت درمدل مورد نظر برسد.

**1-2-3- جمع‌آوری داده‌های آموزشی**

همان‌طور که در یادگیری نظارت شده مطرح است، برای آموزش مدل عملگر نقاد، باید داده‌های آموزشی وجود داشته باشد. با این حال، برای جمع‌‌آوری چنین داده‌هایی، مدل باید در محیط اجرا شود. داده‌های آموزشی برای هر قسمت جمع‌‌آوری می‌شود. سپس در هر مرحله زمانی، گذر رو به جلو مدل بر روی حالت محیط اجرا می‌شود، تا احتمالات عمل و مقدار بر اساس پارامتر خط‌مشی فعلی که توسط وزن‌های مدل تعیین شده است، تولید شود. اقدام بعدی از احتمالات عمل تولید شده توسط مدل نمونه برداری می‌شود که سپس در محیط اعمال می‌شود و باعث ایجاد حالت و پاداش بعدی می‌شود.

**2-2-3- محاسبه بازده مورد انتظار**

دنباله پاداش‌ها برای هر مرحله زمانی ، جمع آوری شده در طول یک قسمت به دنباله‌ای از بازده‌های مورد انتظار تبدیل می‌شود که در آن مجموع پاداش‌ها از مرحله زمانی فعلی به گرفته می‌شود و هر پاداش با یک ضریب تخفیف به طور نمایی در حال زوال ، مطابق فرمول ۱ ضرب می‌شود:

(۱)

از آن‌جایی که است، به پاداش‌های دورتر از مرحله زمانی فعلی وزن کمتری می‌دهد. به طور شهودی، بازده مورد انتظار به سادگی نشان می‌دهد که پاداش در حال حاضر بهتر از پاداش‌های بعدی است. در مفهوم ریاضی، این عمل برای اطمینان از همگرایی مجموع پاداش‌ها است. برای تثبیت آموزش، توالی بازده حاصل نیز استاندارد شده است (یعنی صفر میانگین و انحراف استاندارد واحد).

**3-2-3- تابع ضرر عملگر نقاد**

از آن‌جایی که از یک مدل ترکیبی عملگر نقاد استفاده می‌شود، تابع ضرر انتخابی همانطور که در فرمول ۲ نشان داده شده است، ترکیبی از ضرر عملگر و نقاد برای آموزش است.

(۲)

**تابع ضرر عملگر:** ضرر عملگر بر اساس گرادیان‌های خط‌مشی با نقاد، به‌عنوان خط پایه وابسته به حالت است و با تخمین‌های تک نمونه (در هر قسمت) مطابق فرمول ۳ محاسبه می‌شود.

(۳)

در این فرمول تعداد گام‌های زمانی در هر قسمت است که می‌تواند در هر قسمت متفاوت باشد. پارامتر حالت در گام زمانی می‌باشد. پارامتر عمل انتخابی در مرحله زمانی برای حالت داده شده است. پارامتر نشان دهنده خط‌مشی (عملگر) است که با پارامتر بندی می‌شود. پارامتر نیز، تابع مقدار (نقاد) است که با پارامتربندی شده است. پارامتر بازده مورد انتظار برای یک حالت معین، یعنی جفت اقدام در مرحله زمانی می‌باشد. یک عبارت منفی نیز به مجموع اضافه می‌شود، زیرا ایده این است که با به حداقل رساندن زیان ترکیبی، احتمال اقداماتی که پاداش‌های بالاتری را به همراه دارند، به حداکثر رساند.

مقدار در فرمول مزیت نامیده می‌شود که نشان می‌دهد، چقدر به یک عمل در یک حالت خاص نسبت به یک عمل تصادفی انتخاب شده بر اساس خط مشی برای آن حالت برتری داده می‌شود.

در حالی که امکان حذف یک خط مبنا وجود دارد، این ممکن است منجر به واریانس بالا در طول اموزش شود و نکته خوب در مورد انتخاب منتقد به عنوان خط مبنا این است که آموزش تا زمانی است که تا حد ممکن به نزدیک باشد که منجر به واریانس کمتر می‌شود. علاوه بر این، بدون نقاد، الگوریتم تلاش می‌کند تا احتمالات اقدامات انجام شده در یک حالت خاص را بر اساس بازده مورد انتظار افزایش دهد که اگر احتمالات نسبی بین اقدامات ثابت بماند، ممکن است تفاوت زیادی ایجاد نکند.

به عنوان مثال، فرض کنید که دو اقدام برای یک حالت معین، بازده مورد انتظار یکسانی را به همراه داشته باشند. بدون نقاد، الگوریتم سعی می‌کند، احتمال این اقدامات را بر اساس هدف افزایش دهد. با نقاد ممکن است، معلوم شود که هیچ مزیتی وجود ندارد () و در نتیجه هیچ سودی در افزایش احتمالات اقدامات به دست نمی‌آید و الگوریتم گرادیان‌ها را صفر می‌کند.

**تابع ضرر نقاد:** آموزش نزدیک شدن به را می‌توان به عنوان یک مشکل رگرسیون با تابع ضرر مطابق فرمول ۴ تنظیم کرد:

(۴)

در این فرمول زیان Hober [13] است که نسبت به زیان مربعات خطا و داده‌های پرت حساسیت کمتری دارد.

**4-2-3- تعریف مرحله آموزش برای به‌روزرسانی پارامترها**

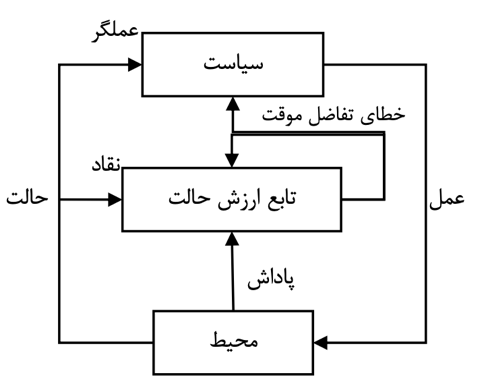
تمام مراحل بالا در یک مرحله آموزشی ترکیب می‌شوند، تا هر قسمت اجرا شود. تمام مراحل منتهی به تابع اجرا می‌شوند، تا تمایز خودکار را فعال کنند. این آموزش از بهینه‌سازAdam ]13[ برای اعمال گرادیان‌ها به پارامترهای مدل استفاده می‌کند. مجموع پاداش بدون تخفیف نیز، در مرحله پاداش‌ها محاسبه می‌شود. این مقدار بعداً برای ارزیابی اینکه آیا معیار موفقیت برآورده شده است، استفاده می‌شود. اعمال این مراحل در نهایت می‌تواند، منجر به افزایش سرعت چشم‌گیری در آموزش شود.

**3-3- عملگر نقاد پیشرفته**

همانطور که گفته شد، برای یادگیری خط‌مشی از یک معماری ترکیبی استفاده شد که عملگر نقاد و روش Duelling DQN و ترفندهای بهبود یادگیری با هم ترکیب شدند. در ادامه به توضیح بخش‌های مختلف مدل ارائه شده و همچنین نحوه تعامل اجزا با هم پرداخته می‌شود.

**1-3-3- عملگر نقاد**

روش‌های بهینه‌سازی خط‌مشی، به دو دسته عمده مبتنی بر خط‌مشی مثل گرادیان خط‌مشی و مبتنی بر ارزش مثل یادگیری Q تقسیم می‌شوند. اما هر دو این روش‌های عمده ایراداتی دارند. در دسته مبتنی بر خط‌مشی، سرعت یادگیری پایین است، چون به‌روزرسانی‌ها بعد از هر دور انجام می‌شود و زمان‌بر است. در دسته مبتنی بر ارزش، اگر محیط پیوسته باشد، همگرایی تضمینی ندارد، چون عملاً تعداد زیادی حالت و عمل وجود دارد و به محاسبه تعداد Q زیادی نیاز است. در عوض هر کدام مزیت خود را دارد. در دسته مبتنی بر خط‌مشی، به ‌راحتی می‌توان در یک محیط پیوسته عمل مناسب را انتخاب کرد. در دسته مبتنی بر ارزش با یک به‌روزرسانی به نتیجه می‌رسد. حال اگر این دو روش ترکیب شوند، در واقع می‌توانند مکمل هم باشند. در این حالت سرعت همگرایی بالاست. در این پژوهش موضوع انجام بازی فوتبال است که شامل ۲۰ عمل است. از آن‌جایی که ترکیب این اعمال و همچنین تعداد اهداف ممکن زیاد است و همین‌طور فضای حالات بزرگ است، لذا تصمیم گرفته شد که از معماری عملگر نقاد، جهت یادگیری خط‌مشی بهتر استفاده شود. اين معماري داراي دو بخش عملگر و نقاد بوده که بخش نقاد براي تقريب تابع ارزش و بخش عملگر براي تولید عمل استفاده مي‌شود. بخش نقاد مسئول پردازش پاداش‌هاي دريافتي از محیط و ارزيابي کیفیت خط‌مشی مورد استفاده توسط عامل است. بخش عملگر با بکارگیري اطلاعاتی از نقاد پارامترهاي خط‌مشی خود را به‌روز‌رساني مي‌کند. طول يادگیري در هر گام زماني نقاد، يک خطاي تفاضل موقت[[6]](#footnote-6) را مطابق فرمول ۱ که ذکر شده بود را تولید و بر اساس آن، يادگیري در عملگر و نقاد انجام مي‌شود. شکل ۱ معماری کلی یک عملگر نقاد را نشان می‌دهد.

****

**شکل ۱- معماری عملگر نقاد**

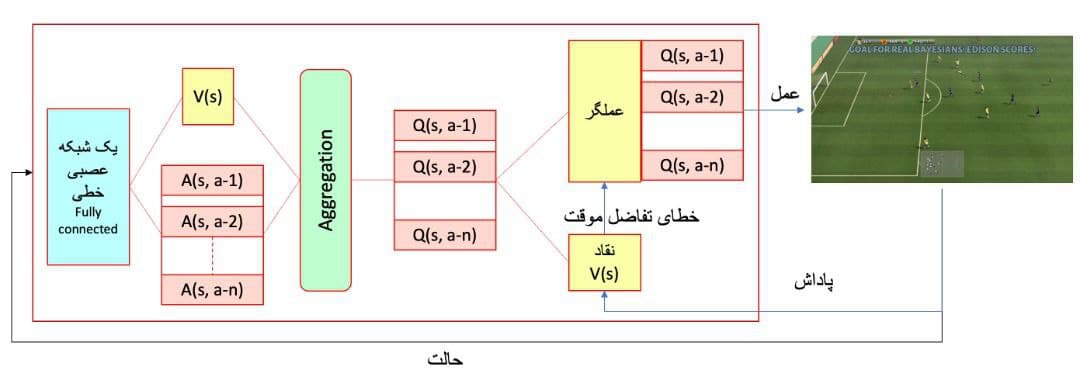
**2-3-3- معماری Duelling DQN**

گاهی در برخی حالات، برخی عمل‌ها لزومی ندارند که اعمال شوند. لذا وجود معیاری برای ارزیابی میزان اهمیت و مزیت هر عمل در هر حالت، می‌تواند پارامتر ارزشمندی باشد [14]. لذا در معماری Duelling DQN [15] بعد از یک بخش مشترک، دو شاخه جداگانه ایجاد می‌شود. مقدار ارزش یا مقدار حالت و میزان مزیت برای هر اقدامی مطابق فرمول ۵ محاسبه می‌شود. نهایتاً مقادیر ارزش اقدامات نسبت به ارزش حالت نرمال می‌شوند.

(۵)

**3-3-3- ترکیب برای ایجاد مدل عملگر نقاد پیشرفته**

ابتدا خطای چندگامه محاسبه می‌شود. سپس در معماری عملگر نقاد، الگوریتم Dueling DQN برای محاسبه و استفاده شد. در DQN Dueling خروجی دو شاخه مقدار و مزیت محاسبه می‌شود. خروجی این دو جریان با هم ترکیب و پس از عبور از لایه softmax [16] مقدار هر کدام از و ها به دست می‌آید. در هر گام زماني نقاد يک خطاي تفاضل موقت را تولید می‌کند و براساس آن، نقاد به‌روزرسانی می‌شود. همچنین نسبت به این خطا، میزان به‌روزرسانی عملگر نیز تعیین می‌شود. الگوریتم عملگر نقاد پیشرفته از چندین منظر بهبود یافته است. شبکه دارای استراتژی قوی‌تر بوده و ارزیابی شبکه دقیق‌تر است. شکل ۲ چارچوب تعامل بین الگوریتم عملگر نقاد پیشرفته و محیط بازی فوتبال گوگل را نشان می‌دهد.



**شکل ۲- تعامل بین هوش‌مصنوعی مبتنی بر الگوریتم عملگر نقاد پیشرفته و محیط فوتبال گوگل**

**4- ارزيابي نتایج**

آموزش عامل در محیط بازی فوتبال با مدل‌های عملگر نقاد، Duelling DQN و عملگر نقاد پیشرفته انجام شد. بر اساس پاداش میانگین مدل‌های گفته شده، میانگین امتیاز در چند تکرار متفاوت از دور‌ها و همچنین در دفعات متفاوتی از تعداد مراحل در هر دور مقایسه شدند. تمام پیاده‌‌سازی‌ها در محیط Colab[[7]](#footnote-7) و به زبان پایتون انجام شد. تنظیم پارامترهای اولیه در جدول ۲ آورده شده‌اند.

**جدول۲- پارامترهای تنظیم.**

|  |  |  |
| --- | --- | --- |
| پارامتر | مقدار | شرح |
| نرخ یادگیری | ۰۱/۰ |  |
| تعداد دورها | ۱۰۰۰ |  |
|  | ۱/۰ | فاکتور تخفیف |
| مراحل | ۱۰۰ | مراحل در هر دور |
| N | ۷۲×۹۶ | رزولوشن نقشه کوچک |
| بهینه‌ساز | Adam |  |

نتایج آموزش عامل با هر سه مدل در جدول ۳ نشان داده شده است. در مرحله اولیه آموزش، یعنی در قسمت‌های مشاهده، به‌روزرسانی شبکه خط‌مشی متوقف می‌شود. از آن‌جایی که آموزش شبکه خط‌مشی به شبکه نقاد بستگی دارد، شبکه نقاد ابتدا باید در این مشاهده آموزش داده شود. پاداش‌های حاصل از آموزش عامل با استفاده از هر سه مدل، همان‌طور که در جدول ۳ قابل مشاهده است، نشان می‌دهد که در اکثر موارد پاداش کسب شده توسط عامل در روش عملگر نقاد پیشرفته از دو روش دیگر بالاتر است. کلید کسب امتیاز بهتر به ترکیب دو مدل Dueling DQN و عملگر نقاد بستگی دارد. در حقیقت این مدل که پاداش مورد انتظار اخیر یک حالت را جمع‌آوری و محاسبه می‌کند، حاوی اطلاعات بیشتری است و به پاداش واقعی نزدیک‌تر از تخمين برآوردگر تک ارزشی در دو روش دیگری است که به صورت جداگانه استفاده می‌شوند. با این حال، نورون‌های بیشتری در شبکه نقاد نیاز است.

**جدول۳- میزان میانگین پاداش برای الگوریتم‌های عملگر نقاد، Dueling DQN و عملگر نقاد با Dueling DQN**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| مقادیر(پاداش‌ها بر حسب درصد می‌باشند) | | | | | | | | پارامتر |
| ۵۰۰۰۰ | ۲۰۰۰ | ۲۰۰۰ | ۱۰۰۰ | ۱۰۰۰ | ۱۰۰۰ | ۱۰۰۰ | ۲۰۰۰ | **تعداد دورها** |
| ۸/۰ | ۸/۰ | ۱/۰ | ۱/۰ | ۱/۰ | ۹۹/۰ | ۹۹/۰ | ۹۹/۰ | **(فاکتور تخفیف) 𝛾** |
| ۱۰۰ | ۱۰۰ | ۱۰۰ | ۱۰۰ | ۱۰۰۰ | ۱۰۰۰ | ۱۰۰ | ۱۰۰ | **تعداد مراحل در هر دور** |
| ۰ | ۰ | **۲/۹۴** | **۹/۸۳** | **۸/۶۷** | **۵/۳۶** | **۷/۸** | ۲/۰ | **پاداش عملگر نقاد با Dueling DQN** |
| ۰ | ۰ | ۰ | ۱ | ۰ | -۳ | ۱ | **۱۵** | **پاداش عملگر نقاد** |
| ۰ | ۰ | ۰ | ۸/۱ | -۹/۰ | -۶/۰ | ۰ | ۲/۰ | **Duelling DQNپاداش** |

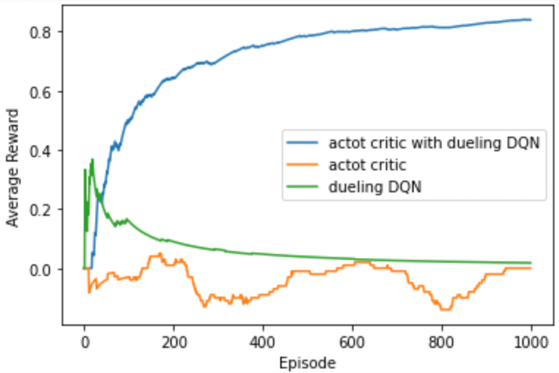
نتایج جدول ۳ نشان می‌دهد، سرعت همگرایی نه تنها به اینکه عامل یک امتیاز خوب کسب کند، بستگی دارد، بلکه به ارزشی که به هر حالت داده می‌شود نیز، بستگی دارد. اين نشان دهنده اطمینان عامل است. در عملگر نقاد پیشرفته، استفاده از Duelling DQN، عامل را در جهت درست تشویق و راهنمایی می‌کند. در نتیجه، عامل مبتنی بر عملگر نقاد پیشرفته بهتر از دو مدل دیگر در بازی همگرا می‌شود.

آزمایش‌ها با توجه به برخی تغییرات در تعداد نورون‌ها انجام شده است که نتایج تغییرات بر روی هر سه مدل در جدول ۴ قابل مشاهده است. نتایج نشان می‌دهد، بهترین تعداد نورون‌ها بر اساس میانگین پاداش‌ها به ازای بهترین مقداری که برای 𝛾 کسب شده بود، برای مدل عملگر نقاد پیشرفته ۱۲۸ نورون و برای عملگر نقاد ساده و Dueling DQN، ۱۶ نورون می‌باشد. حال اگر نمودار میانگین پاداش در هر سه الگوریتم با توجه به بهترین تنظیمات برای هر روش با کمک جدول ۳ و ۴ رسم شوند، نمودار میزان میانگین پاداش هر سه الگوریتم به ازای پیشروی در تعداد دورها در شکل ۳ قابل مشاهده است.

**جدول ۴- میزان میانگین پاداش برای الگوریتم‌های عملگر نقاد، Dueling DQN و عملگر نقاد با Dueling DQN به ازای تغییرات در**

**تعداد نورون‌ها**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **پارامتر** | **مقدار(بهترین) در هر ۱۰۰۰ دور(پاداش‌ها بر حسب درصد)** | | | | |
| **تعداد واحد های پنهان** | ۱۶ | ۳۲ | ۶۴ | ۱۲۸ | ۵۱۲ |
| **پاداش عملگر نقاد - بهترین(۹۹/۰= 𝛾)** | **۱۳** | ۱ | ۱ | ۱ | ۰ |
| **پاداش عملگر نقاد با Dueling DQN - بهترین(۱/۰= 𝛾)** | ۸ | ۵۶/۸ | ۴/۲۳ | ۹**/۸۳** | ۲ |
| **پاداش شبکه Dueling DQN - بهترین(۱/۰= 𝛾)** | **۸/۱** | ۱/۰- | ۵/۰ | ۹/۰ | ۵/۰ |



**شکل۳- مقایسه میانگین پاداش دو الگوریتم عملگر نقاد و Dueling DQN با الگوریتم عملگر نقاد با Dueling DQN**

در شکل ۳ به خوبی مشاهده می‌شود که میانگین پاداش در مدل بهبود یافته توسط معماری ارائه شده در این مقاله، نسبت به دو الگوریتم دیگر افزایش قابل توجهی داشته است. این افزایش پاداش نشان می‌دهد که در عملگر نقاد، نقاد سعی بر بهبود یادگیری دارد، اما عملگر مانع بهبود عملکرد می‌شود و روند صعودی نزولی متناوبی را ایجاد کرده است. همچنین الگوریتم Dueling DQN رفته‌رفته دچار نزول میانگین پاداش می‌شود و نتایج خوبی را در آموزش عامل موجب نمی‌شود. اما روش عملگر نقاد پیشرفته با استفاده از مزایای هر دو الگورتم دیگر، منجر به نتایج بسیار بهتری شده است و می‌توان گفت میانگین پاداش به ازای ۱۰۰۰ دور، حداقل ۴۵ درصد افزایش یافته است.

**5- نتيجه­گيري**

وجود معیاری برای ارزیابی میزان اهمیت و مزیت هر عمل در هر حالت می‌تواند پارامتر ارزشمندی باشد. همچنین با توجه به مزایای مدل Dueling DQN و عملگر نقاد، مدل ارائه شده از ترکیب مدل Dueling DQN در معماری عملگر نقاد استفاده شده است. هدف بهبود عملکرد الگوریتم عملگر نقاد معمولی در یک محیط پیچیده است. مطالعه موردی آموزش عامل در محیط بازی فوتبال است. آموزش عامل در محیط بازی فوتبال با مدل‌های عملگر نقاد، Duelling DQN و عملگر نقاد پیشرفته انجام شد و بر اساس پاداش میانگین مدل‌های گفته شده، میانگین امتیاز در چند تکرار متفاوت از دور‌ها و همچنین در دفعات متفاوتی از تعداد مراحل در هر دور مقایسه شدند. نتایج تأیید کرد که الگوریتم بهبود یافته به طور مؤثر عملکرد را بیش از چهار برابر از نظر میانگین پاداش، نسبت به هر یک از دو مدل Dueling DQN و عملگر نقاد بهبود می‌دهد. همچنین می‌توان فهمید که مدل بهیود یافته به خوبی در محیط‌های پیچیده و پیوسته که پویایی بالایی دارند نیز، می‌تواند مدل مناسبی برای آموزش عامل‌ها باشد.

**مراجع**

*1. K. Shao, Z. Tang, D. Zhu, Y,. Li, N. and D. Zhao, “A survey of deep reinforcement learning in video games,” arXiv preprint arXiv:1912.10944, 2019.*

*2. K. Shao, Z. Tang, D. Zhu, Y,. Li, N. and D. Zhao, “Learning Battles in ViZDoom via Deep Reinforcement Learning,” IEEE Conference on Computational Intelligence and Games(CIG), pp. 1-4, 2018.*

*3. Z. Zha., X. Tang, B. Wang, “An Advanced Actor-Critic Algorithm for Training Video Game AI,” ApplicationsIn International Conference on Neural Computing for Advanced Applications(NCAA),v Singapore, vol. 1265, 2020.*

*4. K. Kurach, A. Raichuk, P. Stańczyk, M. Zając, O. Bachem, L. Espeholt, Riquelme, C. Vincent, D. Michalski, M. Bousquet, and S. Gelly, “Google research football: A novel reinforcement learning environment,” In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 04, pp. 4501-4510, April 2020.*

*5. S. Mysore, B. Mabsout , R Mancuso,., & Saenko, K, “Honey, I Shrunk The Actor: A Case Study on Preserving Performance ith Smaller Actors in Actor-Critic RL,” arXiv preprint arXiv:2102.11893, 2021.*

*6. Y. Wang, H. He, and X. Tan, “Truly proximal policy optimization,” In Uncertainty in Artificial Intelligence, pp. 113-122, August 2020.*

*7. V. Mnih, A.P Badia, M. Mirza, A. Graves, T. Lillicrap, T.Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” In International conference on machine learning, pp. 1928-1937, June 2016.*

*8. Y. Wu, E. Mansimov, R.B. Grosse, S. Liao, and J. Ba, “Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation,” Advances in neural information processing systems, Vol. 30, pp. 5279-5288, 2017.*

*9. K. Tomoda, & K .Hasebe,“Playing Geister by Estimating Hidden Information with Deep Reinforcement Learning,” In2021 IEEE Conference on Games(CoG), pp. 01-04, Aug 2021.*

*10. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, “Playing atari with deep reinforcement learning,” arXiv preprint arXiv:1312.5602, Dec 2013.*

*11. Vinyals, O., et al.: StarCraft II: a new challenge for reinforcement learning. arXiv preprint arXiv:1708.04782, 2017.*

*12. G. Lample, and S.C. Devendra, “Playing FPS games with deep reinforcement learning,” In Thirty-First AAAI Conference on Artificial Intelligence. 2017.*

*13. Z. Zhang, “Improved adam optimizer for deep neural networks,” 2018 IEEE/ACM 26th International Symposium on Quality of Service(IWQoS), 2018.*

*14. V. Hasselt, H. Guez, D. Silver, “Deep reinforcement learning with double q-learning,” In: Thirtieth AAAI Conference on Artificial Intelligence, Vol. 30, No. 1, 2016.*

*15. M. Sewak, “Deep q network (dqn), double dqn, and dueling dqn,” InDeep Reinforcement Learning, Springer, Singapor, pp. 95-108, 2019.*

*16. M. Xiaoteng, Li. Xia, Z. Qianchuan, “Air-combat strategy using Deep Q-Learning,” 2018 Chinese Automation Congress(CAC), IEEE, 2018.*

1. ۱ Actor-Critic [↑](#footnote-ref-1)
2. ۲ Dueling Deep Q Network [↑](#footnote-ref-2)
3. Render [↑](#footnote-ref-3)
4. 2 Episode [↑](#footnote-ref-4)
5. ۱ Advantage Actor Critic(A2C) [↑](#footnote-ref-5)
6. ۱Temporal Difference(TD) Error [↑](#footnote-ref-6)
7. ۱ Colab: https://colab.research.google.com/ [↑](#footnote-ref-7)