گزارش سمینار درس یادگیری ماشین
یادگیری تقویتی در محیط‌های پیوسته در چند کاربرد

داریوش حسن‌پور آده

۹۳۰۸۱۶۴

۱ مقدمه

در حالت کلی یادگیری تقویتی شامل ۳ قسمت می‌باشد: عامل، محیط و اعمال که عامل همان موجودیتی می‌باشد که تصمیم می‌گیرد در محیط چه عملی را از میان اعمال تعریف شده برای خود به اجرا در آورد. محیط نیز در بازخورد به اعمالی که عامل انجام می‌دهد واکنش می‌دهد که این واکنش به عنوان امتیاز در یادگیری تقویتی تعریف شده است. هدف کلی از این زمینه تحقیق در مورد روش‌هایی که بتواند عامل با توجه به پاداش های دریافتی سیاست بهینه‌ی مرتبط با محیط را یاد بگیرد. در اکثر مسائل دنیای واقعی دنیا ربات(عامل) در محیطی پیوسته با موقعیت‌ها و اعمال پیوسته فعالیت می‌کند برای همین روش‌های متعارف ارائه شده برای یادگیری تقویتی در این‌گونه محیط‌های پیوسته قابل پیاده‌سازی نیست؛ بنابراین در این گزارش به معرفی تعدادی از کاربردهای پیوسته‌ی یادگیری تقویتی و روش‌هایی که برای این کاربردها ارائه شده است که بتوان یادگیری تقویتی را در مسائل پیوسته استفاده کرد. در این گزارش بنابه اینکه در دستور کار سمینار آورده شده است که باید دو مقاله بررسی شود به معرفی دو روشی که در مقالات آورده شده است پرداخته و به تحلیل نتایج حاصل از این روش‌ها می‌پردازیم و در نهایت از بحث‌های آورده شده نتیجه‌گیری‌ای خواهد شد.

کلیدواژه‌ها: یادگیری تقویتی، محیط‌ها و اعمال پیوسته

فهرست مطالب

فهرست تصاویر

فهرست جداول

۲ تعریف مساله

همان‌طور که قبلا گفته شد کلیه‌ی الگوریتم‌های یادگیری تقویتی بر مبنای «موقعیت، عمل، پاداش» عمل می‌کنند بدین‌گونه که عامل در موقعیتی خاص تصمیم می‌گیرد که یکی از اعمال ممکن و تعریف شده برایش را اجرا کند می‌کند و براساس آن عمل یک بازخوردی از محیط دریافت می‌کند که معمولا این بازخورد به صورت پاداش در نظر گرفته می‌شود[۱]. در تعریف محیط‌های پیوسته در یادگیری تقویتی پیوسته، به محیط‌هایی گفته می شود که یکی از یا هردوی «موقعیت» و «اعمال» شامل مقادیر پیوسته باشند. به عوان مثال می‌توان یادگیری ربات چهارپره را در نظر گرفت که موقعیت مکانی و زاویه‌ای و سرعت‌های خطی و زاویه‌ای ربات شامل مقادیر پیوسته می‌باشد که که موقعیت ربات در هرلحظه تعیین می‌کنند و اعمالی که چهارپره می‌تواند انجام دهد شامل سرعت هایی است که می‌تواند در هر لحظه به پره‌های خود بدهد می‌باشد. همان‌طور که می‌بینیم در این مثال هردوی موقعیت و اعمال شامل مقادیر پیوسته می‌باشند، به همین خاطر روش‌های مرسوم یادگیری تقویتی نمی‌توانند برای این مورد استفاده شود زیرا که به‌تعداد نامتناهی موقعیت و اعمال داریم و روش‌های عادی نمی‌تواند به ازای تعداد نامتنهایی موقعیت و اعمال سیاست بهینه را یادگیری بگیرد.

۳ انگیزه

در مورد مشکلات کاربردهای دنیای واقعی با یادگیری تقویتی می‌توان گفت که، یک اینکه برای یادگیری یک سیاست بهینه حداقل چند هزار دور ربات باید اجرا شود که در دنیای واقعی معمولا این مساله ممکن نیست زیرا که هر اجرای ربات نیازمند زمان است، و این زمان هرچند اندک(حتی در حتی ۲ـ۱ ثانیه) به ازای چندهزار بار اجرا عملا قابل اجرا نیست، دوم اینکه چون در ابتدای کار یادگیری اکتشاف بیشتر رخ می‌دهد بنابراین ممکن است ربات در طی این اکتشاف‌ها با اجرای اعمال خطرناک به خود یا دیگران صدمه بزند و همچنین میزان زمان کالیبره کردن سنسورها بعد از اجرای هر دوره[۲] زمان‌گیر و دردسرساز می‌باشد، به خاطر این دلایل از یادگیری تقویتی در ربات‌های واقعی معمولا استفاده نمی‌شوند.

علاوه‌بر مشکلات که ذکر شد که چرا یادگیری تقویتی در حالت کلی در رباتیک کاربرد ندارند مشکلاتی دیگر نیز وجود دارند که چرا یادگیری تقویتی برای محیط‌های پیوسته کابرد ندارد می‌توان به ۳ مورد کلی «زمان، حافظه و عدم تضمین هم‌گرایی» اشاره کرد، علت دو مورد اول بدیهی است، زیرا که زمانی که با محیط‌های پیوسته درگیر می‌شویم یکی یا هردوی «موقعیت یا اعمال» به تعداد نامتناهی خواهیم داشت که بعد از اجرا الگوریتم خیلی طول نمی‌کشد که حافظه پر می‌شود، حتی اگر حافظه هم پر نشود چون تعداد نامتناهی است زمان یادگیری سیاست بهینه نیز نامتناهی می‌شود. علت مورد سوم را نیز می‌توان به صورت نشان داده، در اثبات هم‌گرایی یادگیری تقویتی آمده است که اگر بینهایت بار هر یکی از موقعیت‌ـعمل‌ها را مشاهده کنیم در نهایت می‌توانیم به سیاست بهینه برسیم، ولی زمانی که تعداد موقعیت‌ـعمل نامتناهی باشد، دیگر نمی‌توان یک تعداد نامتناهی را به تعداد دفعات نامتناهی مشاهده کنیم که بتوانیم سیاست بهینه را یاد بگیرد.

[۱] دقت شود که در دنیای واقعی، محیط نمی‌توانند پاداشی به عامل بدهد، مگر اینکه توسط آموزگار این پاداش ارائه شود(که در حالت کلی آموزگار و محیط دو مفهوم مجزا هستند).
[۲]

در مورد راه‌حل‌های احتمالی برای یادگیری تقویتی در محیط‌های پیوسته می‌توان به موارد «گسسته‌سازی»، «مدل‌سازی و یادگیری مدل»، «کاهش ابعاد»، «یادگیری توزیع‌شده(ماژولار)» اشاره کرد که بسته به کاربردهای مختلف می‌تواند استراتژی‌های حل مساله‌ی متفاوتی را برگزید.

۴ یادگیری تقویتی پیوسته

در این بخش به دو مورد از کاربردهای یادگیری تقویتی در محیط‌های پیوسته می‌پردازیم.

۱.۴ یادگیری دریبل ربات‌ها در مسابقات Robocop

در این کاربرد[۱] هدف یادگیری کنترل ربات‌های انسان‌نما می‌باشد که مدت زمان در اختیار داشتن توپ توسط ربات حداکثر شود. مساله را به این‌گونه تعریف کرده‌اند موقعیت ربات به تعداد نامتناهی و پیوسته که توسط یک بردار از اعداد پیوسته نمایش داده می‌شود و اعمال به تعداد متناهی به ازای هر موقعیت، هرعمل شامل یک بردار از اعداد حقیقی از پارامتر می‌باشد و هدف یادگیری این مساله می‌باشد که به ازای یک موقعیت خاص چه عملی را باید انجام دهد و سپس به ازای هر عملی مقادیر پارامترهای کنترلی ربات[۳] به چه میزان باشد. در این مقاله یک معماریی برای یادگیری در محیط‌هایی با مقادیر موقعیت‌های پیوسته و تعداد اعمال محدود ولی با مقادیر پیوسته، ارائه داده شده است.

در شکل ۱ معماری رایجی که برای یادگیری تقویتی تعریف شده است، آمده است. که در این شکل میزان خطای TD Error برای بهبود سیاست و تابع مقدار(جهت وفق‌پذیر – در برخی کاربردها) می‌رود که مقدار TD Error در ۲ آمده است.

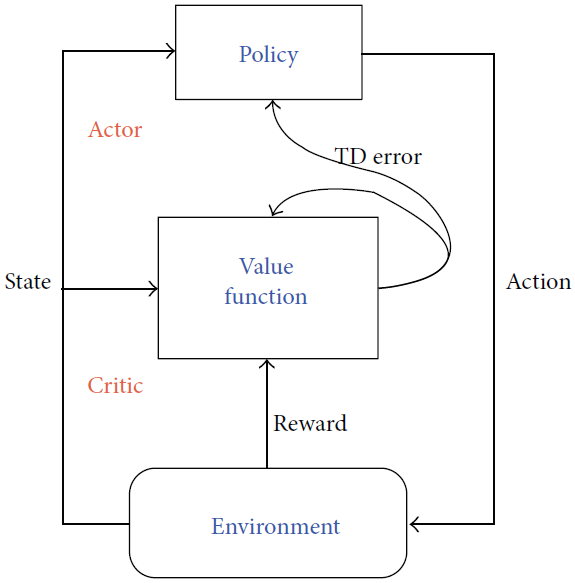
$$V_{t+1}(s) \leftarrow V_t(s) + \alpha\delta \quad (1)$$

$$\delta = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \quad (2)$$

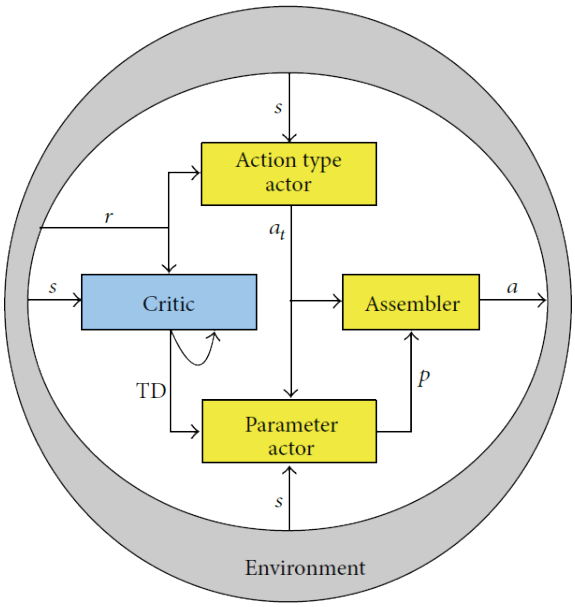
در این مقاله معماریی نوین که معرفی شده است در شکل ۲ آمده است. همان‌طور که در این معماری نشان داده شده است یک یادگیری بروی اینکه در هر موقعیت چه کاری باید انجام شود و در سپس پرامترهای مربوط به آن عمل چگونه باید باشند و در نهایت با ترکیب خروجی‌های این دو یادگیر عمل مربوطه به محیط اعمال می‌شود. همان‌طور که در شکل ۳ نشان داده شده است، از تخمین تابع جهت یادگیری نوع حرکت و مقادیر پارامترهای آن حرکات و منتقد استفاده شده است.

در شکل ۳ به ازای یک موقعیت برای انتخاب مناسب‌ترین عمل، به تعداد اعمال که تخمین تابع داریم، عملی را انتخاب می‌کنیم که مقدار تخمین زننده‌ی آن از بقیه بیشتر باشد، بعد از انتخاب عمل به سراغ تعیین پارامترهای آن عمل می‌رویم که در آنجا نیز به تعداد پارامترهای هر عمل تخمین‌زننده داریم. این تخمین‌زننده

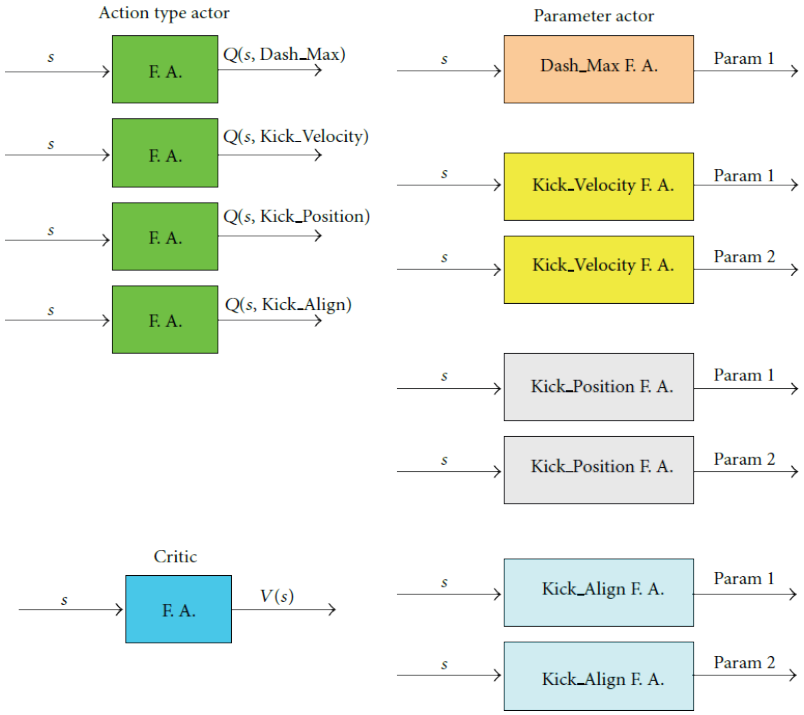
[۳] کمیت سیگنال‌هایی که موتورهای ربات فرستاده می‌شود.



شکل۱: معماری رایج برای یادگیری تقویتی



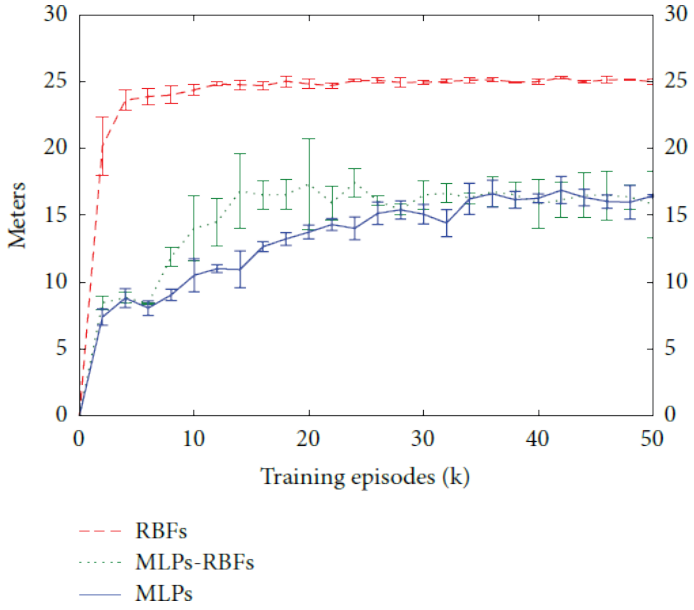
شکل۲: معماری معرفی شده توسط[۱]



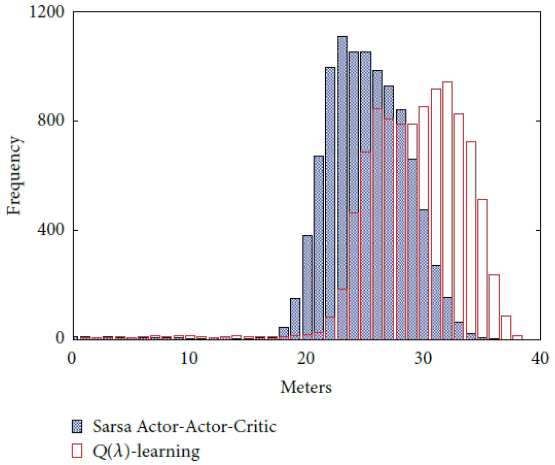
شکل۳: نحوه‌ی استفاده از تخمین تابع جهت یادگیری نوع حرکت و مقادیر پارامترهای آن حرکات و منتقد

ها به ازای پاداشی که از محیط می‌گیرید باید بتوانند خود را بهبود ببخشند. در آزمایش‌های این مقاله از ۳ تخمین‌زننده استفاده شده است، که عملکرد هریک از این تخمین‌زننده در شکل ۴ آمده است. همان‌طور که می بینیم تخمین‌زننده‌ی RBF در کل بیشتر از بقیه توانسته است که توپ را در اختیار خود داشته باشد. در شکل ۵ نیز معماری معرفی شده با الگوریتم $Q(\lambda)$ مقایسه شده است، همان‌طور که از شکل سمت راست می‌بینیم با اینکه در حالت کلی الگوریتم $Q(\lambda)$ عملکرد بهتری نسبت به معماری معرفی شده داشته است ولی برای این عملکرد «اندکی» بهتر[۴] باید زمانی نزدیک به ۲۴ ساعت صرف کنیم در حالی که معماری معرفی فقط در ۱۰ دقیقه توانسته سیاست خوبی را یاد بگیرد.

[۴]مقادیر میانگین و حداکثر مسافت اختیار توپ را در شکل سمت راست دو روش را مقایسه کنید!



شکل۴: عملکرد هریک از ۳ تخمین‌زننده برحسب فاصله‌ای که ربات توانسته است که توپ را در کنترل خود داشته باشد



	SARSA A^2C	$Q(\lambda)$-learning
Algorithm type	Actor-Critic	$Q(\lambda)$-learning
Function approx.	RBFs	CMACs
States	Continuous	Continuous
Actions	Continuous	Discrete
Total learning time	10 minutes	24 hours 30 minutes
Average distance	25.45 meters	29.21 meters
Maximum distance	36.23 meters	39.0 meters

شکل۵: مقایسه‌ی معماری معرفی شده با الگوریتم $Q(\lambda)$

Algorithm 1 SMC-learning algorithm

for all $s \in \mathcal{S}$ **do**
  Initialize $\mathcal{A}(s)$ by drawing N samples from $\pi^0(a|s)$
  Initialize $\mathcal{W}(s)$ with uniform values: $w_i = 1/N$
end for
for each time step t **do**
  Action Selection
  Given the current state s_t, the actor selects action a_t from $\mathcal{A}(s_t)$ according to $\pi^t(a|s) = \sum_{i=1}^{N} w_i \cdot \delta(a - a_i)$
  Critic Update
  Given the reward r_t and the utility of next state s_{t+1}, the critic updates the action value $Q(s_t, a_t)$
  Actor Update
  Given the action-value function, the actor updates the importance weights
  if the weights have a high variance **then**
    the set $\mathcal{A}(s_t)$ is resampled
  end if
end for

شکل۶: الگوریتم ارائه شده برای کاربرد یادگیری قایق‌رانی

۲.۴ یادگیری قایق‌رانی

در این کاربرد [۲] مفاهیم بنیادی همانند کاربرد قبلی می‌باشد، در این کاربرد نیز فرض بر این است که تعداد اعمال محدودی با مقادیر پیوسته داریم. همان‌طور که در ۳ تا ۵ آمده است، در هر موقعیت تعدادی محدود از اعمال وجود دارد که احتمال هریک از آن‌ها از توزیعی همانند π تبعیت می‌کنند که در این توزیع احتمال انتخاب عمل iام برابر با w_i می‌باشد. تابع δ همان تابع دلتای دیراک می‌باشد که در صورتی که $a = a_i$ باشد مقدار ۱ و در غیر این صورت مقدار ۰ را به عنوان خروجی می‌دهد.

$$\mathcal{A}(s) = \{a_1, a_2, \ldots, a_N\} \tag{۳}$$

$$a_i \sim \pi^*(a|s) \tag{۴}$$

$$\pi^*(a|s) \simeq \sum_{i=1}^{N} w_i \cdot \delta(a - a_i) \tag{۵}$$

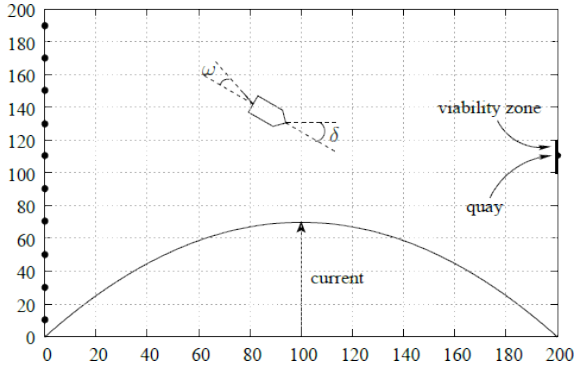
$$a_i \in \mathcal{A}(s), w_i \in \mathcal{W}(s)$$

در ابتدا احتمال انتخاب هر عمل در از توزیع یکنواخت پیروی می‌کند و به مرور احتمال انتخاب‌ها متناسب به بازخوردی که از محیط می‌گیرد ممکن است تغییر کنند. الگوریتم ارائه شده همانند الگوریتم‌های معمول یادگیری تقویتی می‌باشد فقط با این تفاوت که هنگام انتخاب عمل در هر موقعیت، عملی را انتخاب می‌کند که دارای بیشترین وزن(w_i) باشد و برای بروزرسانی وزن‌ها از رابطه‌ی ۶ استفاده می‌کنیم. الگوریتم ارائه شده در شکل ۶ آمده است.

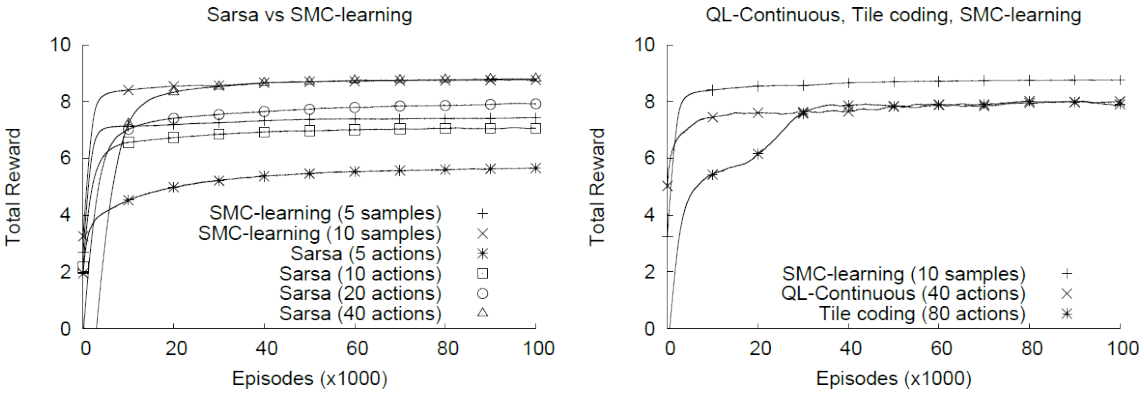
$$w_i^{t+1} = w_i^t \frac{e^{\frac{\Delta Q^{t+1}(s,a_i)}{\tau}}}{\sum_{j=1}^{N} w_j e^{\frac{\Delta Q^{t+1}(s,a_j)}{\tau}}} \tag{۶}$$

$$\Delta Q^{t+1}(s, a_i) = Q^{t+1}(s, a_i) - Q^t(s, a_i)$$

مساله‌ای که در این مقاله مطرح شده است، مساله‌ی یادگیری قایق‌رانی است، همان‌طور که در شکل ۷ می‌بینیم،



شکل۷: تعریف مساله‌ی کاربرد یادگیری قایق‌رانی



شکل۸: مقایسه‌ی الگوریتم ارائه شده(شکل ۶) با الگوریتم‌های SARSA و Q

قایق توسط زاویه‌ی فرمان و اختلاف نوک قایق با افق هدف مشخص می‌شود، همچنین به عنوان اغتشاش یک جریان آبی نیز در رودخانه در جریان است، هدف این است که قایق در ناحیه‌ی مشخص شده بتواند لنگر بیاندازد. همان‌طور که در شکل ۸ می‌بینیم، الگوریتم ارائه شده توانسته زودتر از الگوریتم SARSA همگرا شود و نسبت به الگوریتم Q پاداش‌های بیشتری توانسته کسب کند.

۵ نتیجه‌گیری

در این گزارش به بررسی دلایلی که چرا الگوریتم‌های معمول یادگیری تقویتی نمی‌توانند برای کاربردهای دنیای واقعی استفاده شوند، پرداختیم؛ سپس به معرفی دو کاربرد متفاوت از کاربردهای دنیای واقعی پرداختیم و همان‌طور که دیدیم بسته به نوع مساله باید تغییرات/روش‌هایی ارایه داد که متناسب با مساله باشد و با مقادیر پیوسته‌ی موقعیت و اعمال بتواند اجرای خوبی داشته باشد – روش‌های معمول توانایی کار با مسایل با مقادیر پیوسته را ندارند.

۶ مراجع

[1] V.Uc-Cetina, “A novel reinforcement learning architecture for continuous state and action spaces,” *Advances in Artificial Intelligence*, vol.2013, p.7, 2013.

[2] A.Lazaric, M.Restelli, and A.Bonarini, “Reinforcement learning in continuous action spaces through sequential monte carlo methods,” in *Advances in neural information processing systems*, pp.833–840, 2007.