دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر

گزارش تمرین شماره چهارم
درس یادگیری تعاملی
پاییز ۱۴۰۱

نام و نام خانوادگی
سیاوش رزمی
شماره دانشجویی
۸۱۰۱۰۰۳۵۲

فهرست

چکیده

در این تمرین برخلاف تمرین‌های گذشته قصد داریم به حل مسأله با فرض ناشناخته بودن محیط یا به اصطلاح مسائل Model-free بپردازیم، در بخش اول به بررسی برخی از مسائل تحلیلی و شناخت الگوریتم های model-free می‌پردازیم و در بخش دوم با استفاده از این الگوریتم ها در محیط Taxi کتابخانه‌ی gym سعی در حل این مسأله میکنیم.

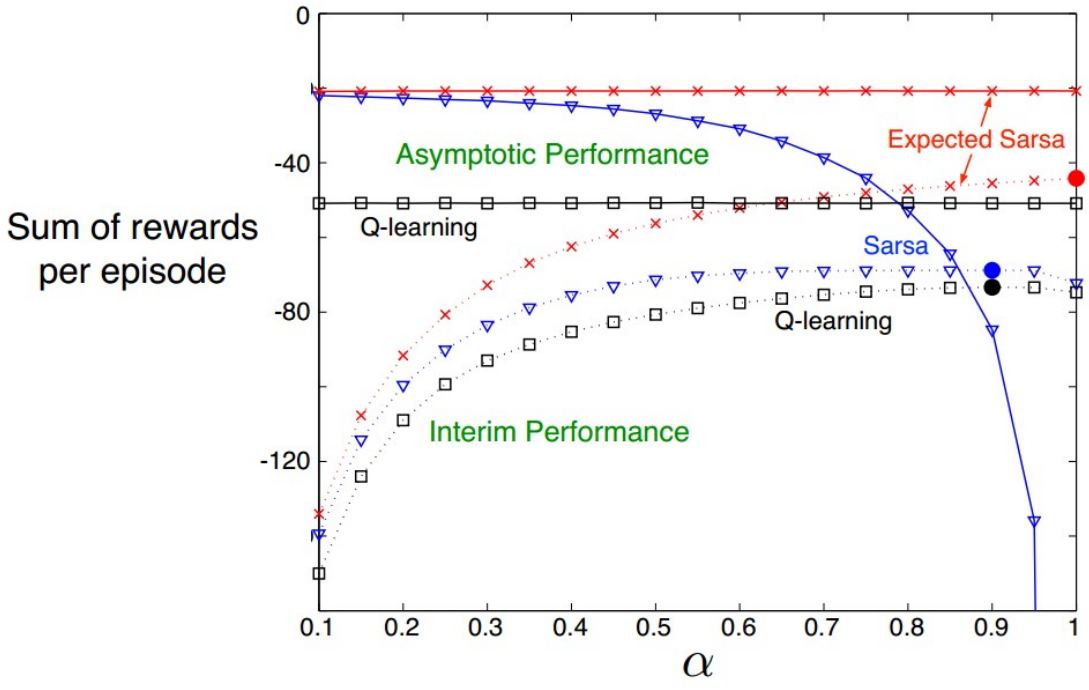
سوال ۱ –

۱

۱.۱. به طور کلی الگوریتم Expected Sarsa به دلیل متوسط گرفتن از تمامی مقادیر ارزش اکشن حالت‌ها نسبت به الگوریتم Sarsa دارای پیچیدگی محاسباتی بیشتری است اما انجام این کار باعث می‌شود که واریانس تخمین در این الگوریتم کمتر شده و همچنین Sampling efficiency بالاتر برود، بنابرین الگوریتم Expected Sarsa با تعداد sample های کمتری میتولند به سیاست بهینه برسد و دارای میزان Regret کمتری است اما میزان محاسبه و زمان مصرفی آن بیشتر است.

۱.۲. در مورد تفاوت سیاست بهینه و اپسیلون بهینه به نظر می‌رسد روند همگرایی در این دو الگوریتم در این دو نوع سیاست تقریباً یکسان است اما میزان Regret به دست آمده در الگوریتم بهینه بیشتر خواهد بود به این دلیل که میزان Reward بهینه به شکل واضح از الگوریتم اپسیلون بهینه بیشتر است.

۱.۳. در زمانی که محیط ما کاملاً Deterministic باشد الگوریتم Expected sarsa نسبت به تغییرات نرخ یادگیری حساسیت بسیار کمی دارد و با افزایش نرخ یادگیری در کوتاه مدت بازدهی آن افزایش پیدا کرده و در بلند مدت نیز تغییرات نرخ یادگیری تأثیر به سزایی در همگرایی آن ندارد، اما بازدهی الگوریتم Sarsa با نزدیک شدن نرخ یادگیری به ۱ کاهش پیدا میکند و در بلند مدت تنها با کاهش نرخ یادگیری به مقدار زیاد امکان همگرایی دارد، بنابرین میتوان گفت در الگوریتم Sarsa بایستی پس از مدتی نرخ یادگیری را کاهش داد تا الگوریتم امکان همگرایی داشته باشد، اما در الگوریتم Expected sarsa تغییرات نرخ یادگیری تأثیر به سزایی در عمل‌کرد الگوریتم ندارد و نرخ یادگیری در الگوریتم Sarsa بایستی سریع‌تر کاهش پیدا کند.



شکل ۱-۱: تأثیر تغییرات نرخ یادگیری در الگوریتم های Sarsa و Expected Sarsa

اما در مورد نرخ کاهش اپسیلون، به دلیل اینکه الگوریتم Expected Sarsa دارای Sample efficiency بالاتری نسبت به الگوریتم Sarsa است و تعداد سمپل کمتری نیاز دارد تا بتواند سیاست بهینه را پیدا کند به نظر می‌رسد در زمان کوتاه تری میتواند به سمت الگوریتم حریصانه تر حرکت کند و نرخ کاهش اپسیلون برای این الگوریتم بایستی بیشتر از الگوریتم Sarsa باشد.

۲

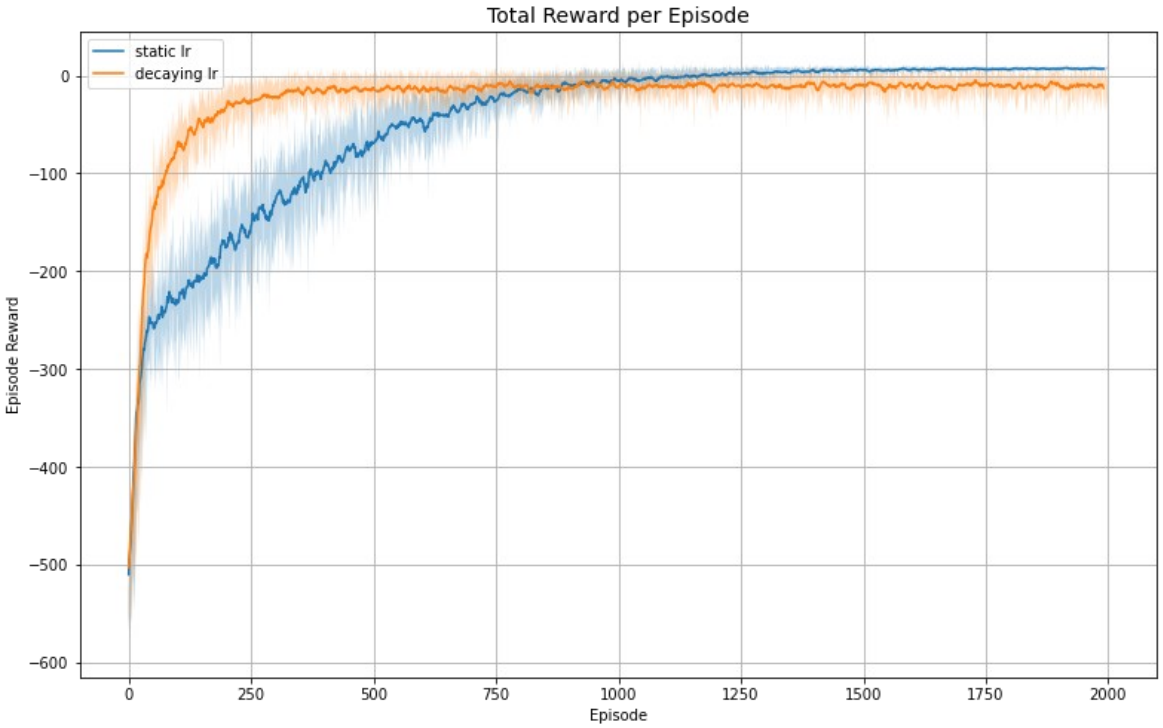
طبق خاصیت error reduction property در الگوریتم های n-step return، تخمین آن‌ها از v_π در مرحله n در بدترین حالت از تخمین v_{t+n-1} بهتر است، درواقع:

$$\underset{\mathbf{s}}{Max}\,|\,\mathbb{E}[G_{t:t+n}|S_t = s] - v_\pi(s)| \leqq \gamma^n \underset{\mathbf{s}}{Max}\,|V_{t+n-1}(s) - v_\pi(s)|$$

بدین معنا که در بدترین حالت هم تخمین این الگوریتم از γ^n برابر ماکزیمم تفاوت بین تخمین V_{t+n-1} و v_π کمتر مساوی است.

سوال ۲ -

۱.

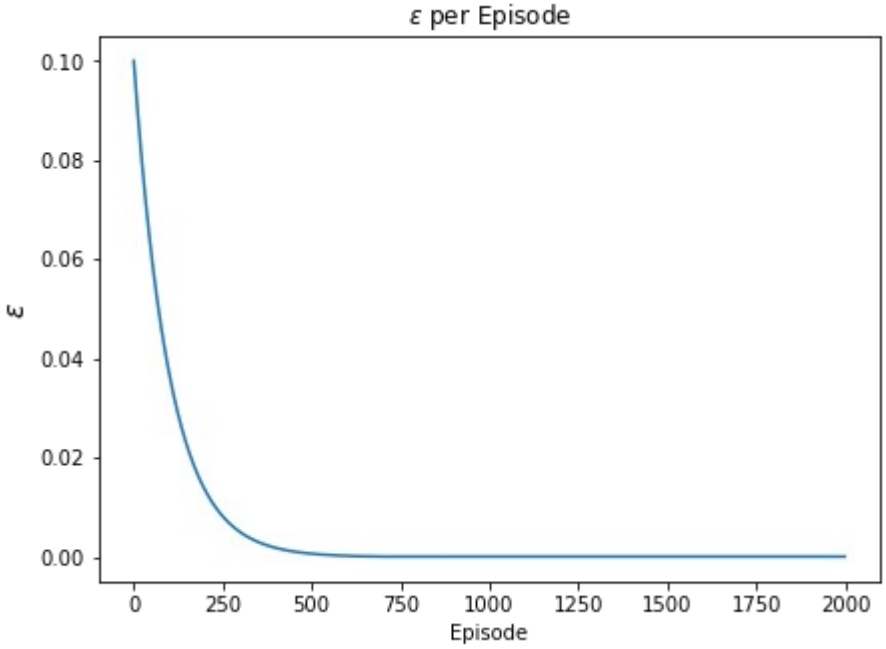


شکل ۱-۲: نتایج مقایسه الگوریتم Q learning با استفاده از نرخ یادگیری ثابت و کاهشی

همانطور که از شکل ۱-۲ قابل مشاهده است، الگوریتم با نرخ یادگیری کاهشی بسیار سریع‌تر به همگرایی می‌رسد و با توجه به بازه اطمینان ۹۵ درصد (هاله رنگی در اطراف مقدار میانگین) این اختلاف معنا دار است، اما میزان پاداش نهایی در الگوریتم با نرخ یادگیری ثابت در نهایت کمی بیشتر از حالت کاهشی است.

* تصاویر عمل‌کرد عامل در ۲۰ مرحله اجرا برای تمامی مسائل در فولدر pictures به همراه گزارش ارسال شده است.

مقدار اپسیلون در این سؤال و تمامی سؤال‌های بعدی به شکل نمایی کاهش پیدا میکند، بدین صورت که یک تابع نمایی با نرخ کاهش ۰.۰۰۱ از ۱ شروع و تا ۰.۰۰۱ کاهش پیدا میکند سپس با ضرب این مقدار در مقدار اپسیلون مشخص شده در صورت سؤال (۰.۱) مقدار نهایی اپسیلون در هر اپیزود به دست می‌آید.

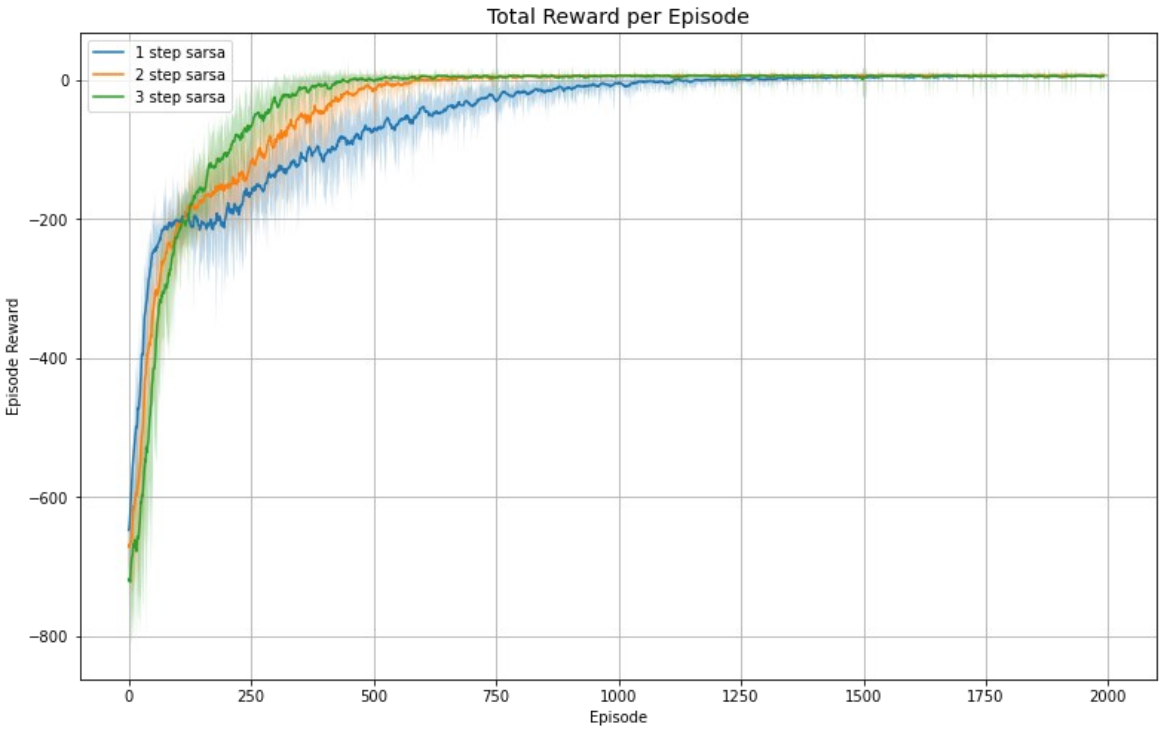


شکل ۲-۲: نحوه کاهش اپسیلون در هر اپیزود

.۲

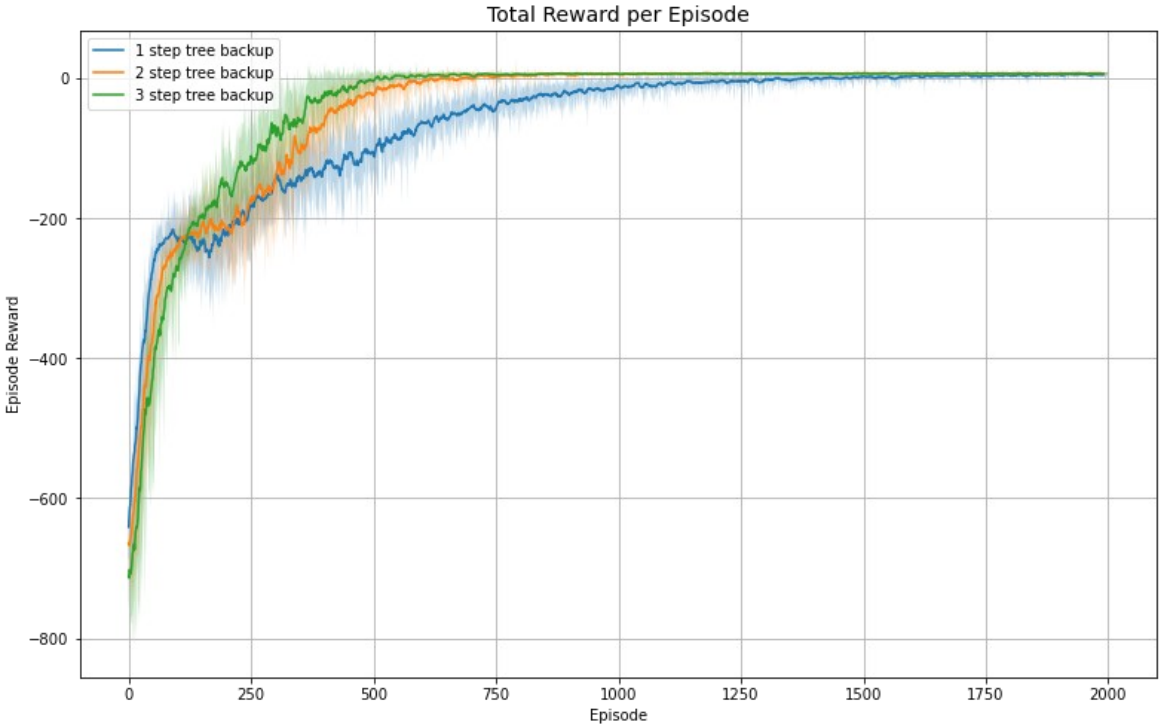
در محیط معرفی شده تاکسی میتواند در ۲۵ خانه، مسافر در ۵ محل (چهار نقطه رنگی به همراه خود تاکسی) و ۴ مقصد است بنابرین به طور ۵۰۰ حالت مجزا در این مسأله تعریف می‌شود، حالت‌های غیر قابل دسترسی مربوط به مواقعی است که محل اولیه و مقصد مسافر یکسان باشد که برابر ۱۰۰ حالت می‌شود، اما در ۴ حالت از این ۱۰۰ حالت تاکسی هم در محل مقصد است که درواقع با یک اکشن به انتهای اپیزود می‌رسیم بنابرین در کل ۴۰۴ حالت در این محیط قابل دسترسی و ۹۶ حالت غیر قابل دسترسی می‌باشد. و برای شناسایی این حالت‌ها کافیست که محل مسافر و مقصد و محل تاکسی را با شروط ذکر شده بررسی کنیم تا دسترسی پذیر یا ناپذیر بودن آن حالت را به دست آوریم.

۳. هر دو الگوریتم n step sarsa و n step tree backup به ازای ۳ مقدار ۱و۲و۳ پیاده شده و نتایج زیر از اجرای آن‌ها حاصل شد:



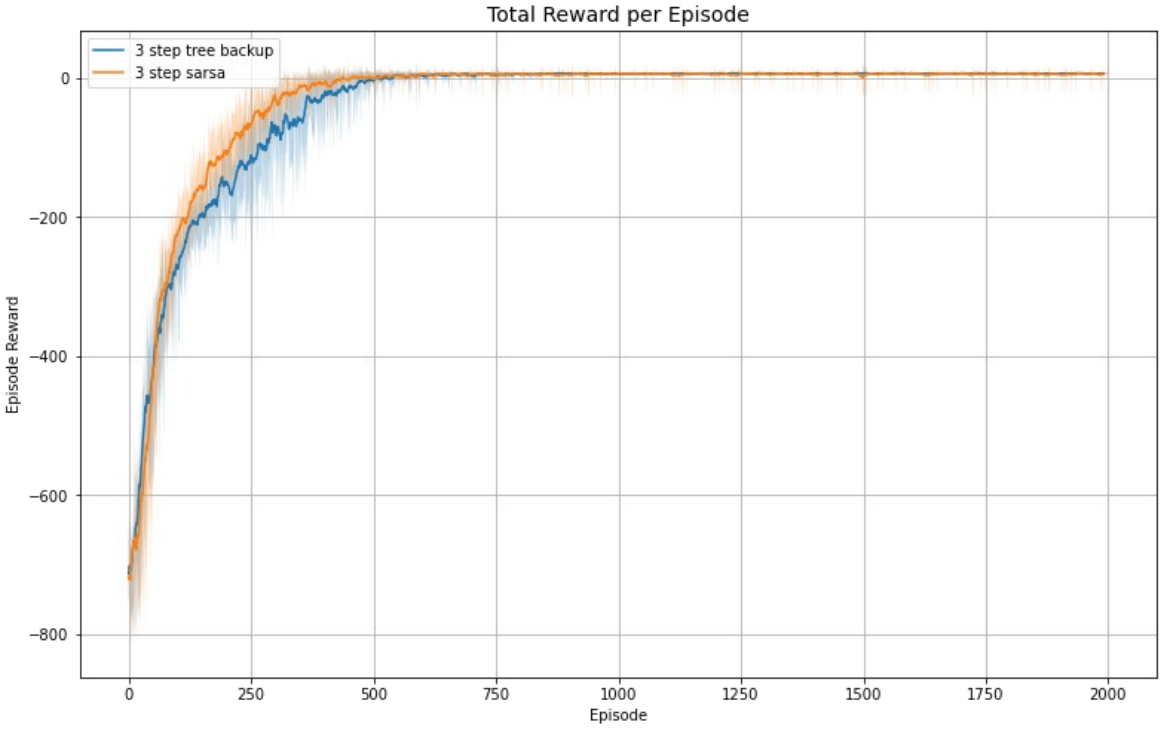
شکل ۳-۲: نتیجه الگوریتم Sarsa برای سه مقدار مختلف

همانطور که از نتایج مشخص است مقدار n برابر با ۳ دارای کمترین میزان Regret در بین ۳ نوع است هر الگوریتم تا حدود ۵۰ اپیزود اول عمل‌کرد مشابهی دارند اما از آنجا به بعد با افزایش میزان n عمل‌کرد الگوریتم بهتر شده و در نهایت هر سه الگوریتم به یک مقدار همگرا شده‌اند با توجه به بازه اطمینان الگوریتم ها در n برابر با ۳ به شکل معنا داری از n برابر با ۱ حسرت کمتری دارد .



شکل ۴-۲: نتیجه الگوریتم tree backup به ازای n های مختلف

الگوریتم tree backup نیز عمل‌کردی حدوداً مشابه الگوریتم sarsa داشت و با افزایش n میزان حسرت آن کاهش پیدا میکند و در نهایت به یک مقدار یکسان همگرا می‌شوند، همچنین با مقایسه عمل‌کرد بهترین نسخه از هر کدام از الگوریتم ها به نظر می‌آید که نمیتوان گفت دو الگوریتم تفاوت معنا داری در عمل‌کرد خود دارند زیرا بازه اطمینان هر دو تقریباً با همدیگر همپوشانی دارد.



شکل ۵-۲: مقایسه عمل‌کرد بهترین نسخه Sarsa و Tree Backup

دلیل اینکه با افزایش میزان n سرعت همگرایی به سیاست بهینه در این الگوریتم ها بیشتر می‌شود این است که با زیادتر کردن مقدار n برای محاسبه ارزش هر اکشن حالت به جای استفاده از ارزش تخمینی از پاداش هایی که عامل به شکل واقعی از محیط کسب کرده استفاده میکنیم در نتیجه تخمین ما نسبت به حالت‌های بانn کمتر دقیق‌تر خواهد بود و Q به دست آمده به Q بهینه سریعتر نزدیک می‌شود.

۴.۱.

الگوریتم On policy Monte carlo دو بار به ازای اپسیلون ثابت ۰.۱ و اپسیلون کاهشی ۰.۵ اجرا شد که طبق نتایج به دست آمده از شکل ۶-۲ در حالت اپسیلون ثابت الگوریتم بسیار سریع‌تر همگرا می‌شود اما به سیاست بهینه همگرا نشده و مقدار sub optimal دارد.

۶-۲: نتیجه الگوریتم On policy monte carlo به ازای مقادیر اپسیلون ثابت و کاهشی

۵.

سرعت یادگیری این الگوریتم نسبت به الگوریتم های قبلی کندتر است و دلیل آن نیز Sample inefficient بودن الگوریتم Monte carlo نسبت به الگوریتم های دیگر و همچنین نیاز به انجام کل اپیزود جهت به روز رسانی مقادیر Q است.

یکی از روش‌های افزایش سرعت الگوریتم پیاده‌سازی Batch sample learning است، به این معنا که از مشاهدات گذشته عامل در محیط چندین مرتبه جهت به روز رسانی استفاده کنیم.

منابع

References

[۱] R. S. Sutton and A. G. Barto, *An Reinforcement Learning: Introduction*. Mit .Press, 2012