دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

گزارش تمرین شماره 5

درس یادگیری تعاملی

پاییز 1401

امیر حسین بیرژندی

...

810198367

...

فهرست

چکیده

با توجه به پیشرفت روز افزون مدلهای یادگیری عمیق و همچنین نتایج قابل توجه ادغام این مدلها با کاربردهای یادیگیری تقویتی، تمرین پنجم به بررسی این الگوریتمها و مدلها خواهد پرداخت. حل کردن مسائل یادگیری تقویتی با استفاده از شبکههای عصبی پیشینهای قدیمی دارد و با توجه به پیشرفت سختافزارهای محاسباتی در دو دهه اخیر، سرعت توسعه مدلهای عمیق برای مسائل یادگیری تقویتی افزایش قابل مالحظهای داشته است. استفاده از شبکههای عصبی این امکان را به ما میدهد که از مسئله را با استفاده از یک مدل end to end حل کنیم. با توجه به این نکته کسب مهارت کارکردن با مدلهای یادگیری عمیق و حل مسائل یادگیری تقویتی با استفاده از این مدلها از مهارتهای ضروری در زمینه یادگیری تقویتی میباشد.

سوال 1 - سوالات تحلیلی

هدف پالیسی گردینت: در خیلی از الگوریتم های یادگیری تقویتی که به آن‌ها action-value methods گفته می‌شود سعی بر تخمین ارزش هر اکشن می‌شود اما در پالیسی گردینت با پارامتری کردن سیاست، ما مستقیماً سعی در یادگیری سیاست بهینه می‌کنیم. با این کار با دریافت پاداش های متفاوت برعکس action-value methods احتمال رخداد یک اکشن بسیار نرم تغییر می‌کند. همچنین در الگوریتم های action-value methods به دلیل وجود e-greedy هیچگاه به یک سیاست deterministic همگرا نمی شویم اما در پالیسی گردینت به دلیل پارامتری کردن سیاست همگرایی خیلی خوبی به سیاست حریضانه خواهیم داشت.

فواید و معایب Deep RL:

در بسیاری از کاربرد های دنیای واقعی استیت‌ها و یا اکشن‌ها فضای پیوسته‌ای دارند و استفاده از الگوریتم های کلاسیک یادگیری تعاملی مانند Q-Learning که برای تمامی حالات و اکشن ها یک جدول ارزش تهیه می‌کند عملاً غیر ممکن است و نیاز به حافظه بسیار بزرگی دارد در نتیجه در عوض یک شبکه عصبی آموزش می‌دهیم که بتواند پارامتر ها را به گونه‌ای انتخاب کند که ارزش استیت-اکشن‌های متفاوت را بدرستی به ما خروجی دهد.

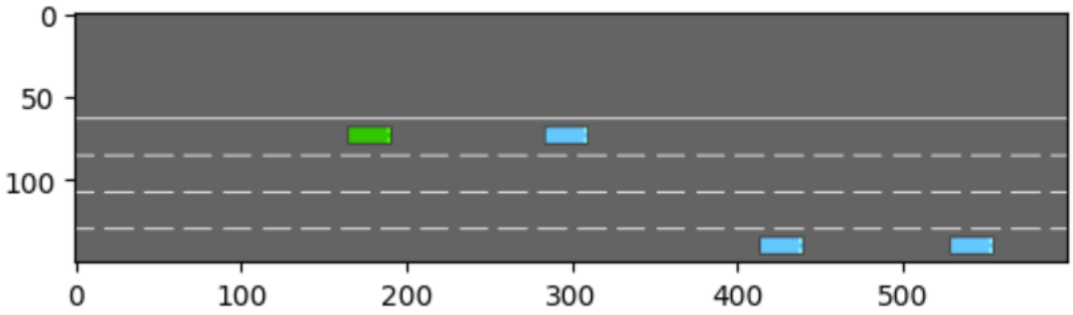
اما در مقابل این سود بزرگ Deep RL می توان گفت پیچیدگی الگوریتم های آن یک عیب به شمار می رود. به عبارتی هم از منظر پیاده‌سازی الگوریتم و انتخاب درست پارامتر ها(تعداد لایه ها و ...) و هم زمان آموزش این الگوریتم ها بار محاسباتی زیادی دارند.

بافر تجارب:

در الگوریتم های Deep Rl اگر قرار بود پس از هر حرکت شبکه را آپدیت کنیم به دلیل وابستگی حرکات پشت سر هم در محیط، فرآیند ما هم از نظر بازدهی و هم از نظر کیفیت آموزش دچار اختلال می‌شود. در نتیجه برای رفع این مشکل در هر مرحله (s,a,s',r) در یک جدول ذخیره شده و هر بار برای آموزش شبکه با سمپل برداری رندوم از آن جدول شبکه را آموزش می‌دهیم. با این تغییر که به آن Experience replay گفته می‌شود در واقع وابستگی بین اعمال و نتایج پشت سر هم را از بین می‌بریم و نتایج بسیار بهتری در همگرایی به سیاست بهینه کسب خواهیم کرد.

سوال 2 - سوال پیاده‌سازی

بخش 1: آشنایی با محیط



محیط را در مود highway-v0 اجرا کرده و تصویر بالا یک فریم از این محیط می‌باشد. در تصویر زیر ابعاد استیت این مسئله، تعداد اکشن ها و یک نمونه از استیت دریافتی در محیط را مشاهده می‌کنیم.

```
State shape:  (5, 5)

Number of actions:  5

State sample:
 [[-0.9938485  -0.09121667  1.985699   -0.95928794 -0.25251576]
 [ 0.21950413 -0.10494459 -0.5317607   0.3748983   0.29245466]
 [-0.91219074 -1.4008539  -2.8870678  -0.70187443  0.46165502]
 [-0.06739108  0.86131966 -2.315493   -2.1946537   2.9830003 ]
 [ 0.33089188 -1.2700201  -0.39070916  0.12549862  0.75004834]]
```

استیت این محیط آرایه ای است که در ردیف اول آن اطلاعاتی در مورد ماشین سبز و یا همان ماشین راننده است و در ردیف های دیگر اطلاعات در مورد وسایل نقلیه دیگر موجود است. منظور از اطلاعات مکان، حضور و ویژگی های دیگر می‌باشد.

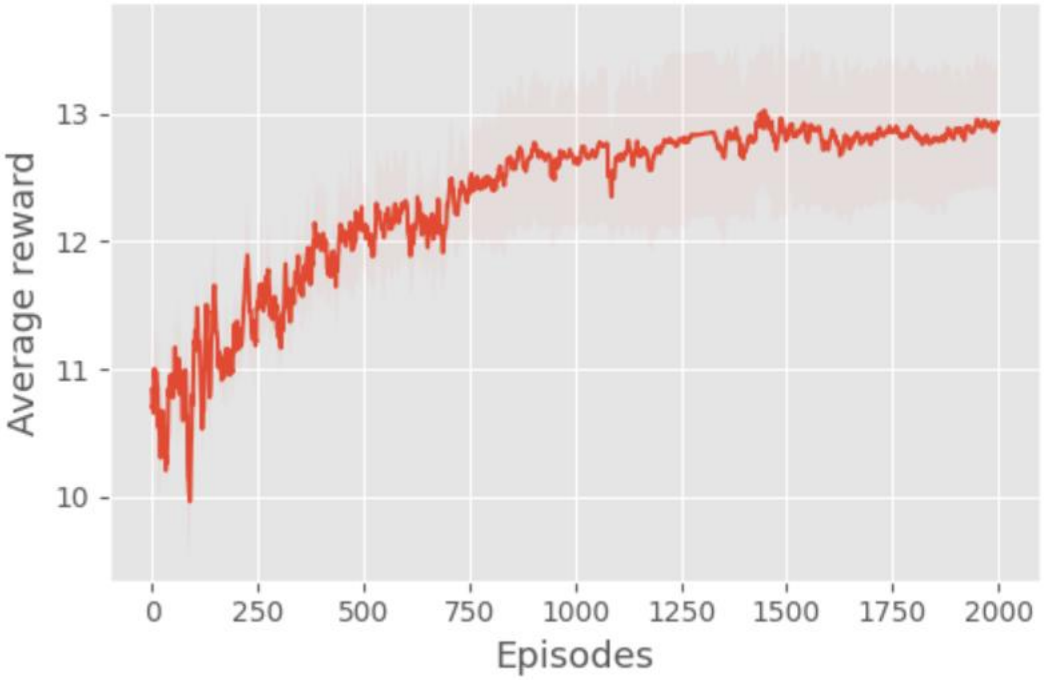
اکشن های این محیط می‌تواند هم به صورت گسسته و هم به صورت پیوسته باشد و بسته به کانفیگ محیط می‌توان این تغییر را ایجاد کرد. در حالت گسسته که ما آن را حل کردیم اکشن ها به صورت {خط چپ، سرعت ثابت، خط راست، سریع‌تر، آروم تر} می باشد.

پاداش در این مسئله به صورت عبارتی شامل یک عامل سرعت و یک عامل تصادف می‌باشد که در تصویر رابطه آن را مشاهده می‌کنیم.

$$R(s,a) = a\frac{v - v_{\min}}{v_{\max} - v_{\min}} - b\,\text{collision}$$

بخش 2: الگوریتم حل

پیاده سازی Deep Q-Learning



در تصویر بالا عامل DQN در محیط merge-v0 با 5 تکرار و 2000 اپیزود ران شده است. همانظور که مشاهده می‌کنید در مرور زمان پاداش میانگین ما افزایش یافته است.

پارامتر های مورد استفاده:

Repeats	5
Episodes	2000
Buffer size	1000
Batch size	64
Gamma	0.99
Tau	0.001
alpha	5e-4
Update freq	4

بخش 3: انتقال تجارب با استفاده از Transfer Learning

در این بخش پس از آموزش عاملی که در محیط merge-v0 زندگی کرده است، وزن های نهایی شبکه آموزش دیده شده را ذخیره می‌کنیم. سپس آن وزن ها را برای عاملی که قرار است در محیط highway-fast-v0 زندگی کند استفاده می‌کنیم و تاثیر آن را مشاهده خواهیم کرد.

مشاهده می‌کنیم که سرعت یادگیری به شدت افزایش یافته است. دلیل این موضوع می تواند شبیه بودن دو محیط باشد یعنی از آنجایی که در هر دو محیط هدف عدم برخورد با ماشین های دیگر است اگر در یک محیط در حالتی هستیم که ماشینی جلویمان است استیت دریافتی بسیار مشابه استیتی است که در محیط دیگر با همین شرایط دریافت می‌کنیم. به همین دلیل وزن های بدست آمده از یک محیط دیگر بسیار می‌تواند ما را در همگرایی سریع تر یاری کند. همانطور که در شکل بالا مشاهده می‌کنیم میانگین پاداش عاملی که از صفر شروع کرده نیز در حال افزایش می باشد اما شیب تغییرات آن بسیار کند است.

منابع

[1] https://github.com/eleurent/highway-env

[2] https://highway-env.readthedocs.io/_/downloads/en/stable/pdf/

[3] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.