



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



گزارش تمرین شماره ۵

درس یادگیری تعاملی

پاییز ۱۴۰۱

نام و نام خانوادگی

مهیار ملکی

شماره دانشجویی

۸۱۰۱۰۰۴۷۶

فهرست

سوالات تحلیلی	۳
سوال ۱ -	۳
سوال ۲ -	۳
سوال ۳ -	۴
سوالات پیاده‌سازی	۵
بخش اول - آشنایی با محیط مسئله	۵
استیت‌ها	۵
اکشن‌ها	۵
پاداش	۶
بخش دوم - الگوریتم Deep Q-Learning و تسک merge	۷
بخش سوم - انتقال تجربه با استفاده از transfer learning	۱۰
بخش چهارم - الگوریتم DQN با استفاده از image observation (امتیازی)	۱۲
روند اجرای کد پیاده‌سازی	۱۳


سوالات تحلیلی

سوال ۱ –


هدف الگوریتم‌های RL پیدا کردن سیاستی است که منجر به بیشینه شدن پاداش شود. تا پیش از این سیاست بهینه، با استفاده از "ارزش" حالات و اعمال تخمین زده می‌شد. اما با استفاده از الگوریتم policy gradient سیاست بهینه بصورت مستقیم از روی حالات و اعمال زده می‌شود.

در واقع ما سیاست را با استفاده از تابع F ، parametrize کرده‌ایم و حال باید پارامترهای این تابع را تخمین بزنیم. برای تخمین این پارامترها می‌توانیم یک معیار fitness برای هر حالت در نظر بگیریم (discounted return یا differential return) و این معیار را با استفاده از روش gradient ascent بیشینه کنیم.

سوال ۲ –


مزایا: 

در مواردی که محیط پیوسته باشد، یعنی حالات یا اعمال یا هر دو پیوسته باشند، با استفاده از الگوریتم‌های Deep RL می‌توان به صورت مستقیم و بدون نیاز به استفاده از کرنل‌ها، ارزش هر حالت-عمل را تخمین زد.


معایب: 

روش‌های Deep RL حجم محاسباتی بالایی دارند، همچنین برای آموزش شبکه‌های عمیق داده‌های زیادی مورد نیاز است. لذا این روش‌ها دارای sample efficiency پایینی هستند. همچنین در این روش‌ها تضمینی برای همگرایی وجود ندارد.

سوال ۳ -

استفاده بهینه‌تر از تجارب در حین آموزش: 

استفاده از بافر تجارب کمک می‌کند تا در حین آموزش هر تجربه را (حالت فعلی، عمل انتخاب شده، پاداش گرفته شده، حالت بعدی) ذخیره‌سازی کرده تا در ادامه بتوانیم از آنها استفاده کنیم. در واقع این امکان را می‌دهد که بتوانیم از هر تجربه به دفعات استفاده کنیم.

کاهش کوریلیشن بین تجربه‌ها: 

استفاده از نمونه‌های تصادفی بافر تجارب، باعث می‌شود تا کوریلیشن بین مشاهدات از بین برود و در نتیجه از نوسان و واگرایی ارزش عمل‌ها جلوگیری شود.

سوالات پیاده‌سازی

بخش اول – آشنایی با محیط مسئله

استیت‌ها

در این مسئله استیت‌ها به صورت یک ماتریس می‌باشد که ستون‌های آن معرف حضور یا عدم حضور عامل، مختصات x و y مکان آن و سرعت عامل در این جهت‌ها بوده و هر سطر نیز مربوط به یکی اتومبیل‌های حاضر در محیط است. البته ویژگی‌های دیگری مرتبط به زاویه حرکت عامل و موارد دیگر نیز قابل تعریف می‌باشد.

	Presence	X	Y	V_x	V_y	...
Vehicle 1						
...						
Vehicle n						

چنان چه از ماهیت این ویژگی‌ها (مکان و سرعت) نیز مشخص است، استیت‌های این محیط پیوسته می‌باشند.

لازم به ذکر است که در این مسئله، استیت محیط می‌تواند به صورت تصویر سیاه‌سفید یا زمان تصادف یا Occupancy grid نیز تعریف شود.

اکشن‌ها

سه حالت برای اکشن‌ها در این محیط قابل انتخاب است:

- حالت پیوسته: در این حالت، اکشن‌های ما به دو انتخاب شتاب و زاویه فرمان تبدیل می‌شود که هر کدام از این کمیت‌ها پیوسته می‌باشند.
- حالت گسسته: این حالت در واقع گسسته شده اکشن‌های حالت پیوسته است.
- حالت فراگسسته: در این حالت که ساده‌تر از دو حالت قبلی نیز می‌باشد، تنها ۵ اکشن گسسته داریم که در هر لحظه تنها یکی از آنها به صورت صفر و یکی قابل اجرا است. این اکشن‌ها به ترتیب عبارتند از: انتقال به لاین چپ، بی‌حرکت، انتقال به لاین راست، افزایش سرعت، کاهش سرعت

پاداش

در اکثر مسائل محیط highway پاداش به دو عامل وابسته است:

۱. پیشروی سریعتر در جاده

۲. جلوگیری از تصادف

لذا پاداش عامل مطابق فرمول زیر محاسبه می‌شود:

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

به عنوان مثال در محیط merge پاداش‌های زیر تعریف شده‌است:

- پاداش تصادف : 1-
- پاداش حرکت در لاین راست : 0.1+
- پاداش سرعت بالا : 0.2+
- پاداش سرعت در زمان ادغام لاین جدید : 0.5-
- پاداش تغییر لاین : 0.05-

- نکته: محیط طراحی شده است که تمام پاداش‌ها بین 0 و 1 نرمالایز شوند. از پاداش منفی نیز جلوگیری می‌شود زیرا ممکن است باعث شود تا در زمانی که هیچ خط سیر بهتری وجود ندارد، عامل ترجیح دهد با عملی که منجر به تصادف می‌شود اپیزود را به اتمام برساند ولی پاداش آن منفی نشود.

بخش دوم – الگوریتم Deep Q-Learning و تسک merge

در این قسمت الگوریتم Deep Q-learning برای تسک merge پیاده‌سازی شده است.

سه حالت مختلف در این بخش بررسی شده‌اند:

۱. استفاده از state ها و شبکه fully-connected :

در این حالت چنان چه در نمودارهای پاداش و هزینه شکل ۱ و ۲ قابل مشاهده است، هزینه شبکه در حین آموزش کاهش یافته و عامل به خوبی آموزش دیده و پس از ۲۰۰۰ اپیزود به محدود پاداش ۱۴ همگرا شده است.

۲. استفاده از observation ها و شبکه convolutional (امتیازی)

در این حالت در مقایسه با حالت قبلی، همگرایی پاداش عامل با سرعت کمتری اتفاق افتاده است، در واقع میزان حسرت در این حالت به طور قابل توجهی بیشتر شده است. هنگام تغییر مشاهدات محیط به تصویر، دو پارامتر قابل تغییر وجود دارد، یکی تعداد تصاویر و دیگری اندازه تصاویر. با افزایش تعداد تصاویر در هر استیت تعدادی تصویر از حالت قبلی محیط نیز برگردانده می‌شود و در واقع اطلاعات بیشتری از استیت ایجنت از جمله سرعت آن به دست می‌آید، همچنین با افزایش اندازه تصاویر، گستره بزرگتری از محیط قابل مشاهده می‌شود.

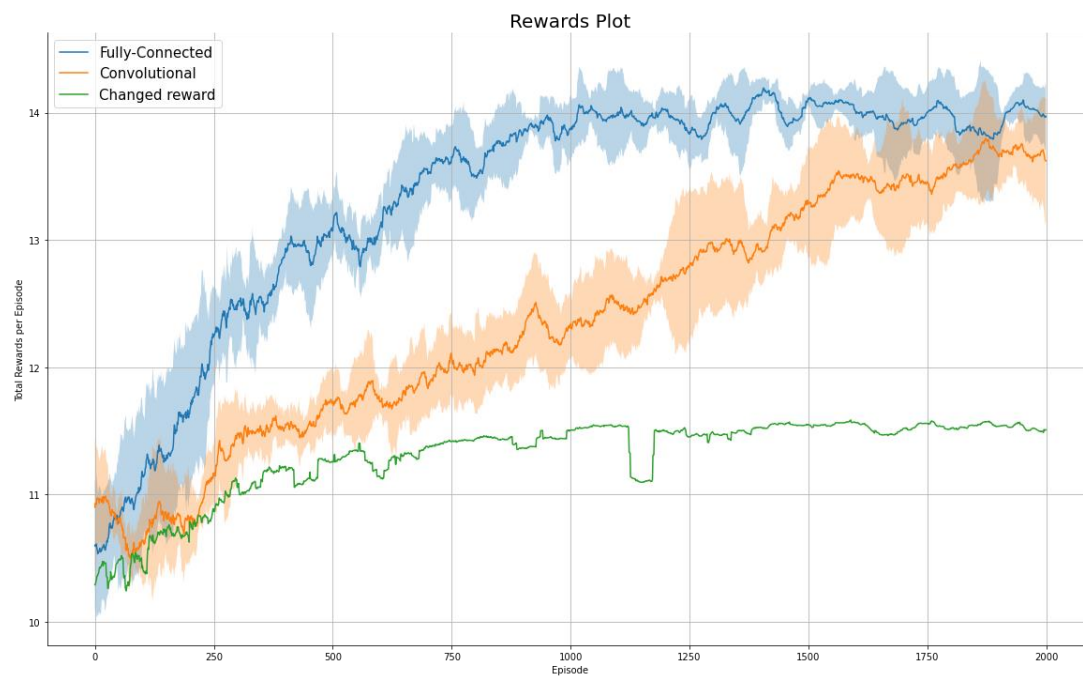
در این قسمت ما تعداد تصاویر را 1 عدد و اندازه تصاویر را نیز 128 در 64 که ناحیه محدودی از محیط را شامل می‌شود، در نظر گرفته ایم. لذا نسبت به حالت قبلی اطلاعات خیلی کمتری از استیت عامل از محیط می‌گیریم و به نظر می‌رسد همین امر باعث عملکرد بدتر آموزش عامل و افزایش حسرت شده است.

۳. استفاده از محیط تغییر یافته (پاداش سرعت از ۰.۲ به ۰.۵ افزایش یافت)

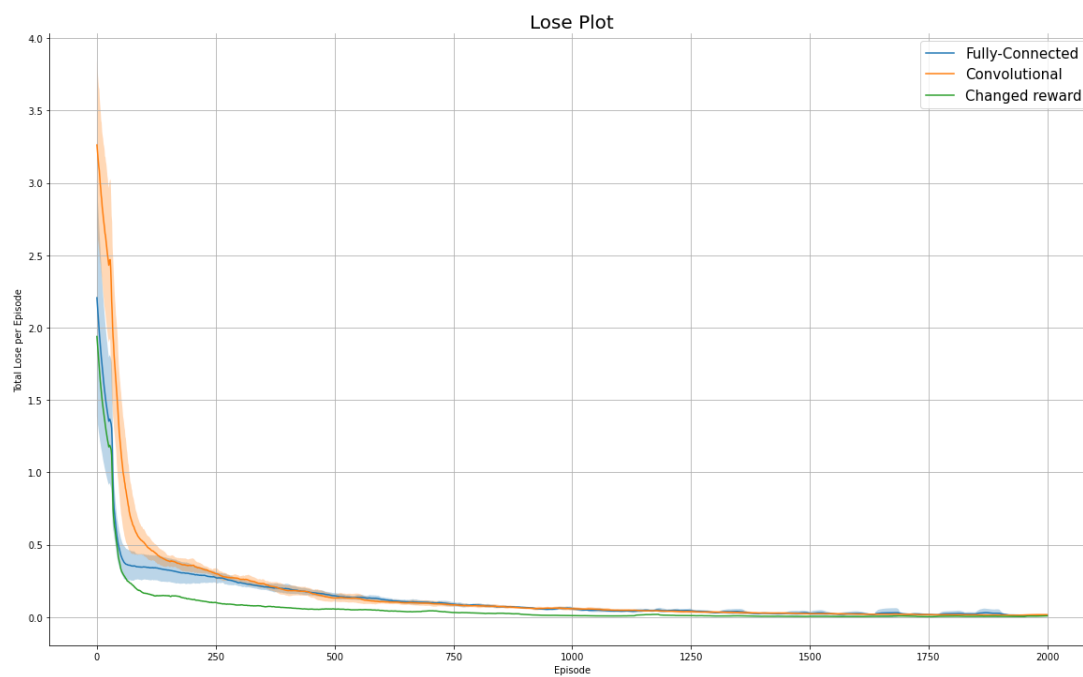
پس از تست عامل در حالات قبلی مشاهده می‌شود که عمل در مسیری مستقیم حرکت کرده و هنگام ادغام شدن لاین‌ها سرعتش را کاهش می‌دهد. این عملکرد به نظر منطقی می‌رسد، زیرا در کانفیگ پیش فرض محیط، سرعت عامل هنگام ادغام پاداش منفی 0.5 را دارد. اما به هر حال برای این که مطابق نمونه گیتهاب مسئله، عامل هنگام ادغام تغییر لاین دهد، پاداش سرعت را افزایش دادیم تا به جای کاهش سرعت، تغییر لاین رخ دهد. لازم به ذکر است که در این حالت پاداش بهینه تغییر می‌کند. این حالت تنها با یک تکرار انجام شده بازه اطمینانی در شکل برای آن وجود ندارد.

جدول ۱- پارامترهای آموزش تسک merge

Gamma	0.9
Learning rate	5e-4
Maximum size of buffer	10000
Maximum epsilon	1
Minimum epsilon	0.005
Decay rate of epsilon	0.005
Number of episodes	2000
Sample batch size	32
Repeats	5
Network structure	Fully-connected : (3 linear-hidden-layer 128 - 256 - 128) Convolutional : (3 conv-hidden-layer 16 - 32 -32)
Network optimizer	Adam
Network loss func.	MSE



شکل ۱- نمودار پاداش تسک Merge برای حالات مختلف



شکل ۲- نمودار هزینه تسک Merge برای حالات مختلف

بخش سوم – انتقال تجربه با استفاده از transfer learning

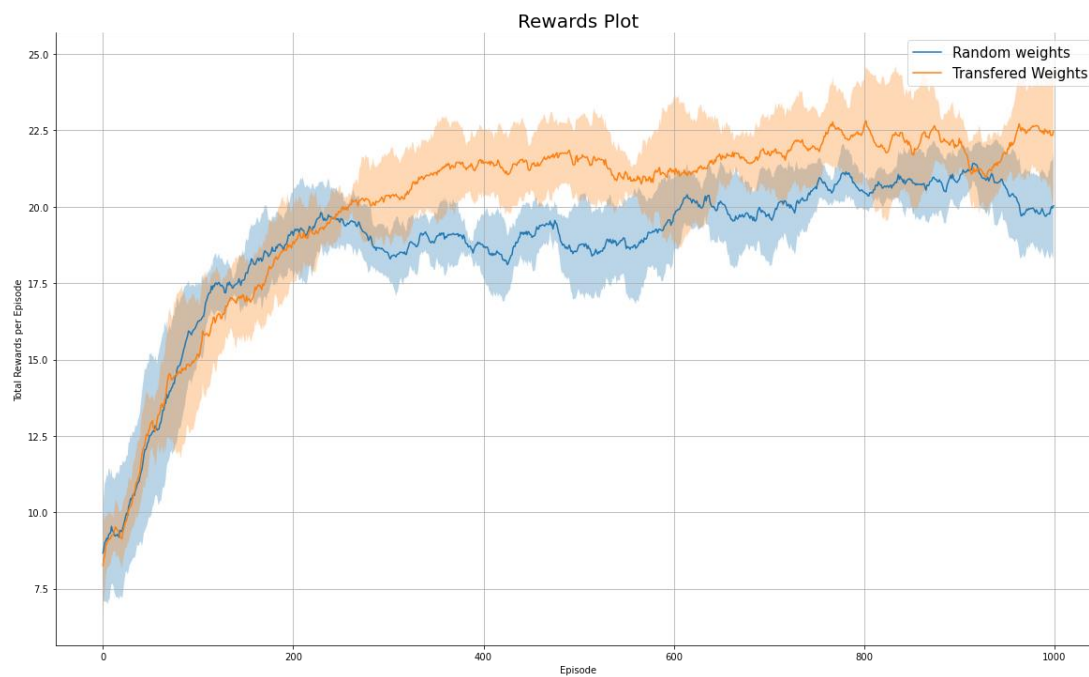
در این قسمت تاثیر انتقال تجربه با استفاده از روش transfer learning برای تسک highway-fast بررسی می‌شود.

همانطور که در نمودارهای پاداش و هزینه شکل‌های ۳ و ۴ قابل مشاهده است، با استفاده از روش transfer learning به پاداش بیشتری همگرا شده و همچنین میزان هزینه نیز سریعتر کاهش یافته و یادگیری سریعتر صورت گرفته است.

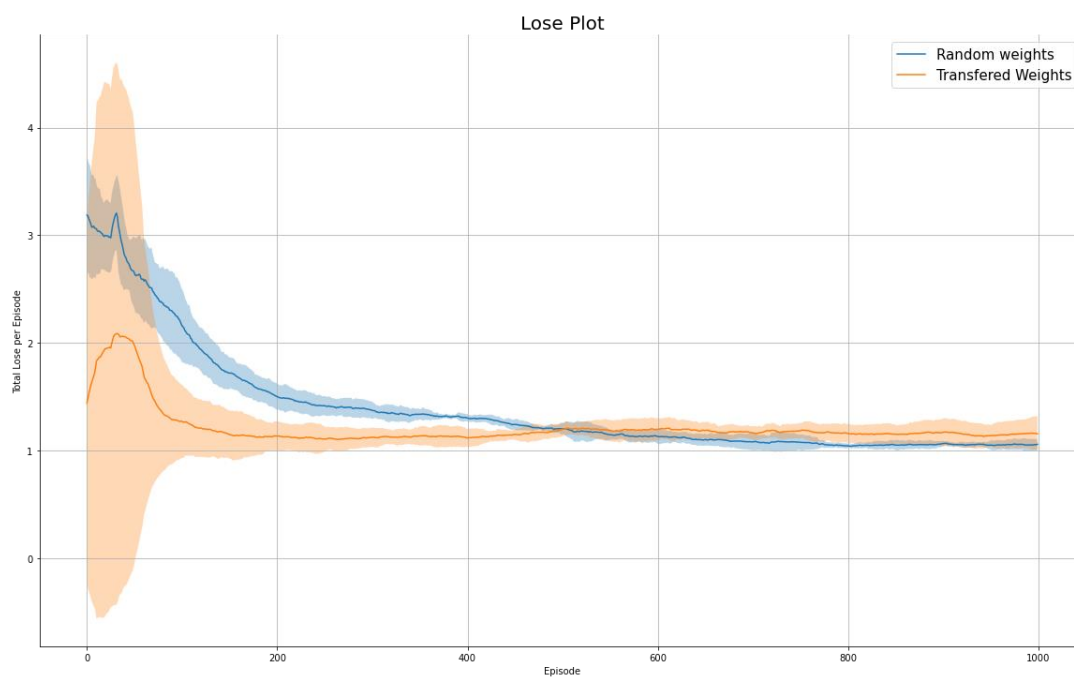
علت افزایش سرعت، این می‌باشد که با استفاده از روش transfer learning به جای این که از مقادیر تصادفی برای مقدار اولیه وزن‌ها استفاده کنیم، از تخمینی نزدیکتر به مقادیر وزن‌های تسک اصلی استفاده می‌کنیم و این باعث می‌شود تا با اپیزودهای کمتری به همگرایی برسیم.

جدول ۲- پارامترهای آموزش تسک highway-fast

Gamma	0.9
Learning rate	5e-4
Maximum size of buffer	10000
Maximum epsilon	1
Minimum epsilon	0.005
Decay rate of epsilon	0.01
Number of episodes	1000
Sample batch size	32
Repeats	5
Network structure	Fully-connected : (3 linear-hidden-layer 128 - 256 - 128)
Network optimizer	Adam
Network loss func.	MSE



شکل ۳- نمودار پاداش تسک highway-fast



شکل ۴- نمودار هزینه تسک highway-fast

بخش چهارم – الگوریتم DQN با استفاده از image observation (امتیازی)

نتایج حاصل از آموزش شبکه در این قسمت، در بخش دوم قابل مشاهده و بررسی شد.

- آیا اختلاف به صورت significant است؟

برای مقایسه دو روش از تکرارهای مختلف میانگین گرفته و از آزمون t-test استفاده می‌کنیم. در نهایت مقدار $pvalue=3.2632640388392195e-159$ بدست آمده که با در نظر گرفتن $\alpha=0.05$ نتیجه می‌گیریم که دو روش اختلاف معناداری داشته و روش convolution با استفاده از image observation عملکرد بدتری را حاصل شده است.

- تفاوت observation و state :

State اطلاعاتی را شامل می‌شود که محیط را توصیف می‌کند. در واقع state تمام اطلاعاتی است که برای تصمیم‌گیری نیاز می‌شود. برای مثال برای تسک‌های این تمرین state شامل تمام متغیرهای دینامیکی محیط و ماشین‌ها مانند سرعت، موقعیت و ... می‌شود. داشتن اطلاعات کامل محیط این اطمینان را می‌دهد که سیستم به صورت MDP می‌باشد، به این معنی که حالت فعلی محیط، مستقل از عمل انجام شده، اطلاعات کاملی درباره توزیع احتمالی تمام حالات آینده به دست می‌دهد.

observation فقط بخشی از اطلاعات state است که عامل از محیط دریافت می‌کند و لزوماً تمامی اطلاعات لازم برای تصمیم‌گیری را شامل نمی‌شود. داشتن اطلاعات ناقص از محیط باعث می‌شود که نتوان از خاصیت سیستم‌های MDP که در بالا گفته شد، استفاده کرد.

- راه‌حل‌های ممکن برای رفع مشکلات observation :

۱. افزایش تعداد مشاهدات، به عنوان مثال در تسک convolutional این تمرین با افزایش stack size می‌توان تصاویری از حالات قبلی محیط را نیز وارد مشاهده فعلی کرد.
۲. استفاده از الگوریتم‌های مبتنی بر مدل که با پیش‌بینی حالات و پاداش‌ها، این امکان را می‌دهد تا از اطلاعات کمتری استفاده کرد.

روند اجرای کد پیاده‌سازی

تمام کدها و پیاده‌سازی‌ها در فایل `HW5.ipynb` قرار دارد. تنها لازم است که تمام سلول‌ها به ترتیب از ابتدا اجرا شوند تا نتایج و نمودارها به دست آیند.