زن زندگی آزادی

دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر

گزارش تمرین شماره پنج
درس یادگیری تعاملی
پاییز ۱۴۰۱

نام و نام خانوادگی
سیاوش رزمی
شماره دانشجویی
۸۱۰۱۰۰۳۵۲

فهرست

چکیده

در این تمرین به بررسی الگوریتم های حالت پیوسته به طور مشخص DQN و Policy Gradient می‌پردازیم و سعی میکنیم که تسک های محیط highway کتابخانه gym را با استفاده از پیاده‌سازی این الگوریتم ها حل کنیم، همچنین با اعمال تغییراتی از قبیل Transfer Learning و استفاده از شبکه Convolutional به جای Fully Connected جهت دریافت مستقیم تصویر محیط تأثیر آن‌ها را در عمل‌کرد مدل بررسی میکنیم.

سؤالات تحلیلی

۱- در الگوریتم Policy Gradient هدف ما تخمین سیاست بهینه به صورت مستقیم از روی محیط و بدون استفاده از مقادیر واسط مانند V و Q است، در این روش با استفاده از متوسط پاداشی که از محیط می‌گیریم و الگوریتم Gradient میتوانیم به سیاست بهینه که باعث بیشینه شدن میزان پاداش می‌شود دست پیدا کنیم، در این الگوریتم ما با استفاده از یک تخمین‌گر (Function Approximator) اقدام به تخمین احتمال انتخاب هر اکشن در هر state میزنیم و سپس با استفاده از پاداش های بدست آمده پارامتر های تابع را به روز رسانی می‌کنیم.

۲- مزایا: با استفاده از یادگیری عمیق میتوان مسائلی با فضای حالت با ابعاد بالاتر را حل کرد به این معنا که با استفاده از DeepRL میتوان بسیاری از مسائل پیچیده‌تر را دانش اولیه کمتر حل نمود دلیل این امر توانایی بالای مدل های یادگیری عمیق در مدل سازی روابط پیچیده‌تر در داده است.

معایب: جهت آموزش مدل های عمیق نیاز به مقدار بسیار زیادی از داده برچسب خورده و قدرت پردازشی است که ممکن است در بسیاری از مسائل تهیه آن بسیار سخت و یا غیر ممکن باشد زیرا در مسائل RL ما برای یادگیری از مقادیر عددی پاداش جهت یادگیری استفاده میکنیم که ممکن است دارای نویز و Sparse و تأخیر باشد و این مسأله یادگیری برای شبکه عمیق را دشوار میکند همچنین مدل های یادگیری عمیق معمولاً پیش‌فرض را بر i.i.d بودن نمونه‌های دریافتی در نظر می‌گیرند در حالیکه مشاهده های پشت سر هم در مدل های RL معمولاً دارای همبستگی زمانی هستند، مدل های یادگیری عمیق همچنین به دلیل پیچیدگی بالا و پارامتر های بسیار زیاد همچنین نسبت به Overfitting نیز بسیار آسیب‌پذیر تر خواهند بود.

۳ – با استفاده از بافر تجارب میتوان دو مشکل بزرگ از مدل هایی بدون بافر را حل کرد:

۱- به روز رسانی هایی که به شدت با همدیگر همبستگی دارند که باعث نقض پیشفرض i.i.d بودن سمپل ها در بسیاری از الگوریتم های Gradient می‌شوند.

۲- از یاد بردن تجاربی که ممکن است نادر باشند و در آینده به کار بیایند.

با استفاده از بافر تجارب میتوان با ترکیب تجارب قدیمی و جدید همبستگی زمانی نمونه هارا از بین برده و همچنین از تجارب نادر چندین مرتبه برای به روز رسانی استفاده نمود.

سوال ۱ - آشنایی با محیط مسئله

۱- در تسک highway-v0 یک عامل به شکل یک وسیله نقلیه در اتوبان در حال حرکت است و هدف مسأله این است که بیشترین سرعت بدون برخورد به بقیه ماشین‌ها حرکت کند، همچنین حرکت در لاین سمت راست اتوبان پاداش مضاعف دارد.

اکشن ها:

در محیط highway چهار مدل مختلف تعریف اکشن موجود است که با تغییر دادن تنظیمات محیط امکان انتخاب آن وجود دارد:

۱- اکشن پیوسته (Continuous Action): در این حالت محیط به عامل اجازه میدهد که به شکل مستقیم زاویه فرمان و میزان گاز را کنترل کند.

۲- اکشن گسسته (Discrete Action): در این حالت اکشن های پیوسته حالت قبلی به شکل یکنواخت گسسته شده‌اند.

۳- متا-اکشن گسسته (Discrete Meta Action): در این حالت عامل با یک لایه سطح بالا اکشن ها تعامل میکند به این شکل که وسیله به شکل خودکار در حال حرکت است و اکشن های موجود صرفاً تغییر لاین به سمت چپ و راست و بیشتر و کمتر کردن سرعت و همچنین عدم تغییر حالت فعلی هستند، برخی از این ۵ اکشن ممکن است در حالت‌های مختلف موجود نباشد، به طور مثال در زمانی که در لاین سمت راست هستیم امکان تغییر لاین به سمت راست وجود نخواهد داشت و در این وضعیت این اکشن مشابه عدم تغییر حالت فعلی عمل خواهد کرد.

کنترل فرمان و سرعت هرکدام میتواند در تنظیمات غیر فعال شود به این معنا که کنترل آن به شکل خودکار توسط محیط انجام خواهد شد.

۴- حالت کنترل دستی(manual Control): در این حالت کنترل وسیله توسط کاربر با دکمه های کیبورد قابل انجام است.

حالت ها:

مشابه اکشن ها، حالت‌ها نیز به شکل‌های مختلف قابل تغییر هستند:

۱- حالت سینماتیک (Kinematics): در این حالت یک ماتریس به ابعاد VxF به عنوان حالت برگردانده می‌شود که در آن V وسایل نقیله نزدیک و F ویژگی‌های آنهاست، ویژگی‌های وسیله نقلیه نیز در تنظیمات قابل تغییر است از جمله این ویژگی‌های میتوان به موارد زیر اشاره کرد:

عدم وجود وسیله نقلیه که توسط ۰ نمایش داده می‌شود، فاصله از عامل، سرعت در هر جهت، جهت به رادیان و...

۲- تصویر سیاه سفید (GrayScale Image): این حالت یک تصویر سیاه سفید از محیط بر می‌گرداند.

۳- ماتریس تصرف (Occupancy grid): در این حالت یک ماتریس به ابعاد WxHxF برگردانده می‌شود که WxH یک جدول بندی جهت گسسته کردن محیط است، هر خانه از این جدول توسط F ویژگی قابل توصیف است.

۴- فاصله زمانی تا برخورد (Time to Collision): در این حالت یک ماتریس VxHxL باز گردانده می‌شود که در آن V سرعت وسیله نقلیه عامل به شکل گسسته، L تعداد لاین های اتوبان و H زمان برخورد گسسته به شکل one-hot encode است، در این حالت به ازای سرعت های مختلف عامل زمان برخورد با بقیه وسیله های نقلیه تخمین زده می‌شود.

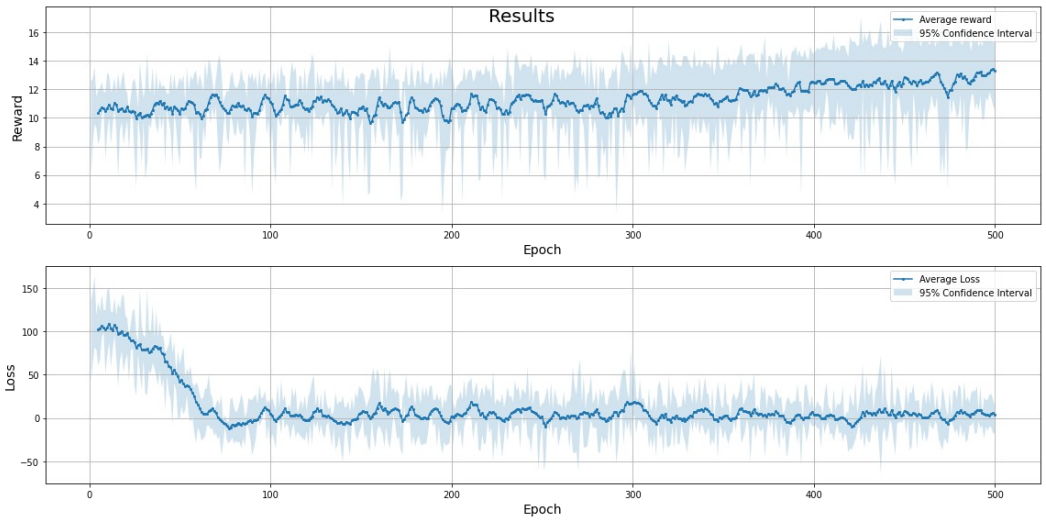
اکشن ها در این مسأله پیوسته هستند اما در حالت‌های گسسته و متا-گسسته با تقسیم‌بندی اندازه ها به مقادیر گسسته سعی در گسسته سازی محیط شده است، حالت‌های ما نیز ترکیبی از موقعیت، جهت حرکت، سرعت وسیله نقلیه عامل و مابقی وسایل است که در حالت‌های تصویر سیاه سفید، سینماتیک و در حالت ماتریس تصرف به شکل پیوسته است (با وجود گسسته سازی سلول‌های محیط اما مقادیر ویژگی‌ها به حالت پیوسته اعلام می‌شود) و مشابه حالت سینماتیک است، در فاصله زمانی تا برخورد اما مقادیر به شکل کاملاً گسسته اعلام می‌شود، در هر صورت ما برای حل این مسأله از Function Approximation استفاده میکنیم زیرا حتی در صورتی که فضا را گسسته در نظر بگیریم تعداد حالت‌ها بسیار زیاد و حل با الگوریتم های گسسته بسیار دشوار خواهد بود.

سوال ۲ - الگوریتم حل

۱- در این پیاده‌سازی از حالت meta-discrete action و kinematics که حالت‌های پیشفرض محیط هستند استفاده شد، الگوریتم Policy Gradient جهت پیاده‌سازی انتخاب گردید، همچنین جهت افزایش پایداری Baseline نیز به الگوریتم اضافه کردیم. جهت پیاده‌سازی شبکه policy از ۲ لایه مخفی با اندازه ۶۴ استفاده شد که با گرفتن ماتریس حالت، احتمال انتخاب هر کدام را با کمک تابع softmax در خروجی تحویل می‌دهد، یک شبکه با همین ساختار نیز جهت تخمین ارزش حالت‌ها پیاده‌سازی شد که با گرفتن ماتریس حالت، ارزش هر حالت را به شکل یک عدد در خروجی می‌دهد. در هر اپیاک با استفاده از تابع generate_episode عامل یک اپیزود را تا انتها انجام می‌دهد و در نهایت تمام حالت‌ها، اکشن ها و پاداش هارا به همراه احتمال انتخاب هر اکشن را به عنوان خروجی بازمیگرداند، سپس با استفاده از تابع compute_discounted_reward مقدار G را در هر حالت S محاسبه می‌کنیم و همچنین با استفاده از شبکه value_net که در ابتدا تعریف کرده‌ایم ارزش هر کدام از حالت‌های مشاهده شده را تخمین میزنیم سپس با مقادیر G به دست آمده خطای شبکه تخمین را به دست آورده و وزن های آن را به روز رسانی می‌کنیم، از مقدار تخمینی ارزش هر حالت به عنوان baseline استفاده می‌کنیم و در هر بار به روز رسانی شبکه سیاست عامل مقدار تخمین زده را از مقدار G کم می‌کنیم، این کار باعث می‌شود مقادیر Return به دست آمده پایدارتر شده و جستجوی ما در فضای بهینه سازی سیاست به شکل روان تر و سریع‌تری انجام گیرد سپس با استفاده از مقادیر خطای بدست آمده طبق فرمول زیر مقادیر وزن هارا به روزرسانی میکنیم:

$$\nabla_\Theta J(\pi) = \mathbb{E}_{\tau\sim\pi}[\nabla_\Theta log\pi(a_t|s_t)R(\tau)]$$

در نهایت با ۵ مرتبه اجرای الگوریتم به تعداد ۵۰۰ اپیاک نمودار پاداش و خطا به شکل زیر بدست آمد:



شکل ۱-۲: نتایج الگوریتم Policy Gradient در تسک merge (مقادیر میانگین smooth شده اند)

همانطور که قابل مشاهده است مقادیر خطا پس از حدود ۱۰۰ ایپاک به حدود صفر رسیده است، البته در برخی موارد نیز مقدار خطا منفی شده است که این به نظر به دلیل استفاده از Baseline و خطای شبکه Baseline در تخمین صحیح مقادیر ارزش‌ها است، مقادیر پاداش در نهایت به حدود ۱۴.۹ می‌رسید که به نظر می‌آید بیشترین مقدار برای این تسک است.

* ویدیوی عمل‌کرد عامل نیز به همراه گزارش در فولدر Videos ارسال شده است.

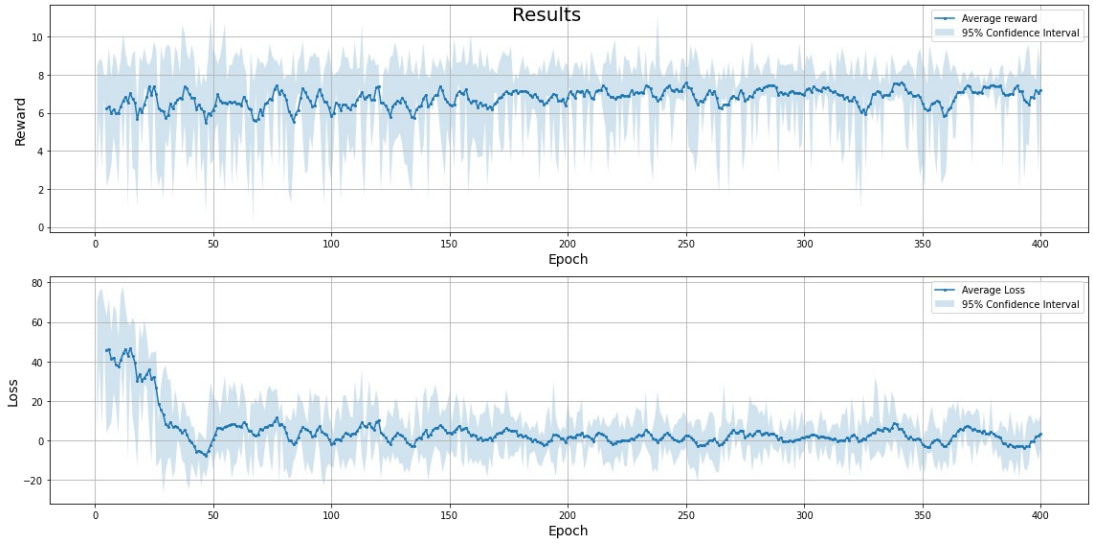
۲- پارامتر های مورد استفاده:

جدول ۱-۲: هایپرپارامتر های مدل

Index	Name	Value
۱	Hidden layer neurons	۶۴
۲	Activation function	Relu
۳	Learning rate	۰.۰۰۱
۴	Discount Factor	۰.۹۹
۵	Environment related	Default
۶	Epochs	۵۰۰
۷	Optimizer	Adam

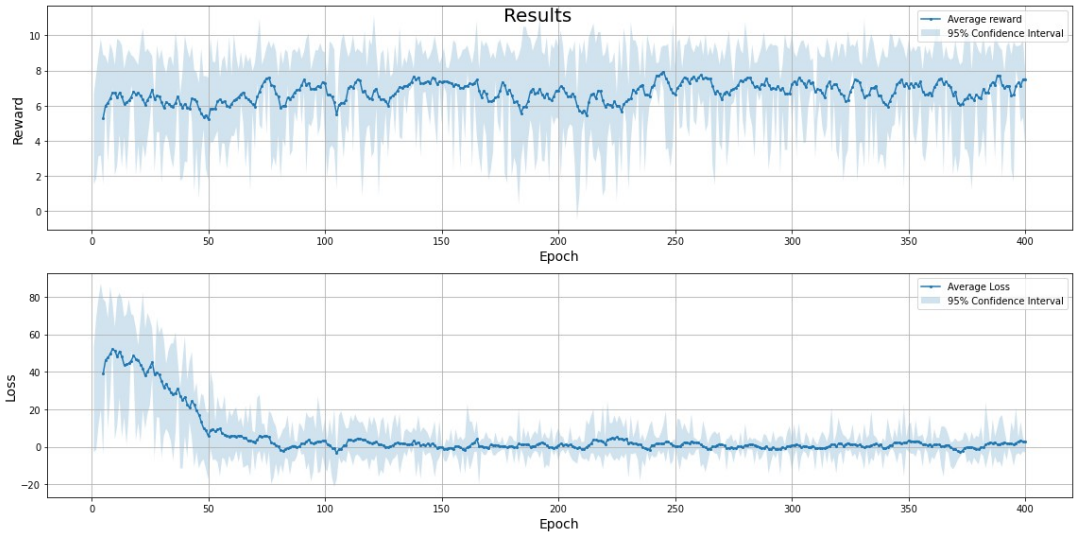
سوال ۳ - انتقال تجربه با استفاده از Transfer Learning

۱- برای استفاده از محیط highway-v0 به دلیل نامتناهی بودن این محیط، از Trajectory هایی با اندازه ۱۰ استفاده شد در ابتدا شبکه با استفاده از وزن دهی تصادفی آموزش داده شد:



شکل ۱-۳: نتایج مدل Policy Gradient در محیط Highway با وزن دهی تصادفی

۲- سپس با استفاده از وزن های مرحله قبل Transfer Learning انجام شد:



شکل ۲-۳: نتایج مدل Policy Gradient در محیط Highway با Transfer Learning

با بررسی نمودار مشخص است که در حالت Transfer learning مدل حدوداً هم‌زمان به دقت صفر رسیده است اما نوسان میزان خطا در این حالت بسیار کمتر است، از لحاظ میزان پاداش نیز سریع‌تر به مقدار نهایی همگرا شده است، بنابرین میتوان نتیجه گرفت که استفاده از وزن های آموزش داده شده در تسک های دیگر در سرعت همگرایی تأثیر مثبت خواهد داشت.

سوال ۴ -امتیازی

- برای پیاده‌سازی این بخش از تمرین قسمت دوم یعنی پیاده‌سازی الگوریتم Policy Gradient با استفاده از Image Observation انتخاب شد.

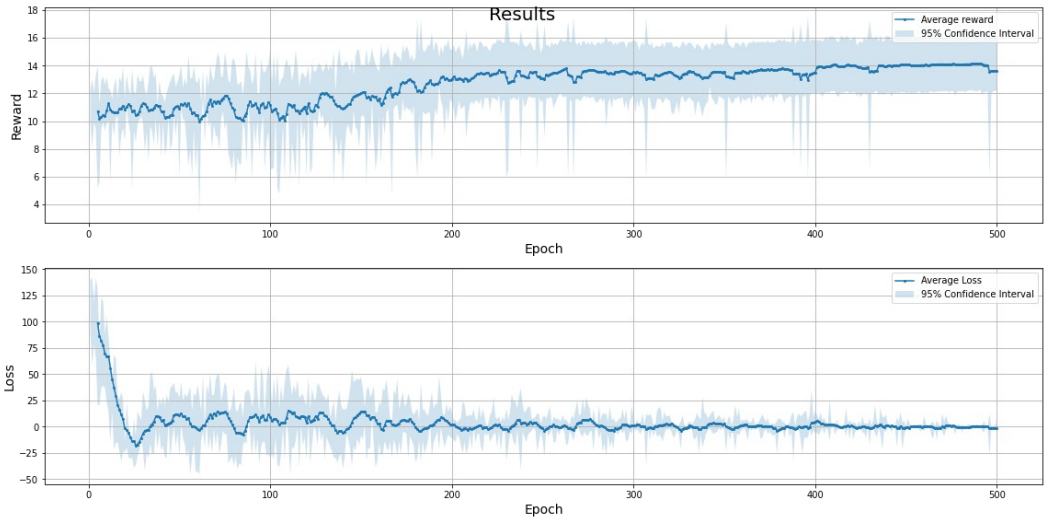
تفاوت عمده Observation و state در نحوه Representation محیط است به این معنا که در state ما یک سری ویژگی‌های Hand Crafted شده که با استفاده از دانش اولیه خودمان از محیط است را از محیط استخراج کرده و جهت تخمین سیاست به مدل تخمین گر میدهیم، اما در Observation ما مستقیماً تجربه ایی که از محیط به دست می‌آوریم را به شکل خام به مدل میدهیم و مدل با استفاده از فیلتر های پردازش تصویر مانند Convolution سعی به Representation Learning میکند، به این معنا که مدل تلاش میکند ویژگی‌هایی از محیط استخراج کند که بیشترین کمک به تخمین صحیح مقدار خروجی را بکند، چالش هایی که این روش می‌تواند داشته باشد ۱- روش‌های یادگیری عمیق نیاز به مقادیر بسیار زیادی داده برچسب خورده دارند الگوریتم های RL بر خلاف این با سیگنال های رقمی آموزش داده می‌شوند که اغلب نویزی، Sparse و دارای تأخیر هستند ۲- اکثر الگوریتم های یادگیری عمیق فرض را بر این میگیرند که نمونه‌های داده دارای شرط i.i.d هستند، در مسائل RL اما اغلب نمونه‌های پشت سر هم همبستگی بسیار زیادی با یکدیگر دارند. برای حل مشکل همبستگی میتوان از مکانسیم Exprience Replay استفاده کرد که با ذخیره کردن تجربه‌ها و نمونه برداری تصادفی از آن‌ها همبستگی بین این نمونه‌های پشت سر همدیگر را تا مقداری از بین می‌برد.

۲- پیاده سازی:

برای ساخت شبکه CNN از شبکه قسمت‌های قبلی تمرین استفاده کردیم و لایه اول آن را با یک لایه فیلتر Convolution جایگزین کردیم، سپس خروجی این لایه به لایه فعال ساز و Maxpool رفته و در نهایت به لایه خطی داده می‌شود و سپس احتمال انتخاب هر اکشن به عنوان خروجی داده می‌شود، شبکه Baseline نیز با همین تغییرات ساخته شد.

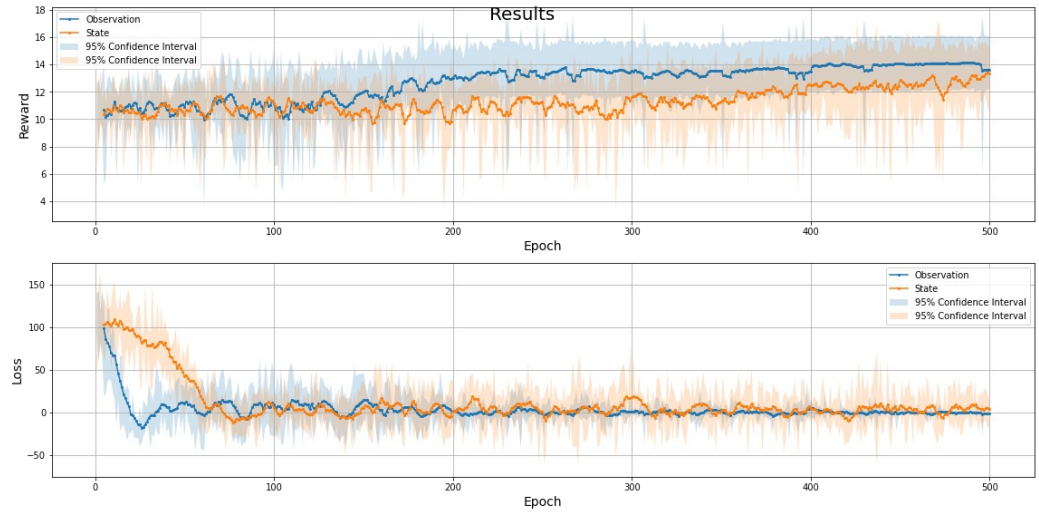
در تنظیمات محیط نیز به جای Kinematics از Gray Scale استفاده کردیم و سایز stack را نیز ۳ در نظر گرفتیم بنابرین در هر مرتبه ۳ تصویر از محیط به عنوان state برگردانده می‌شود که با داده شدن به شبکه احتمال انتخاب هر اکشن دریافت می‌شود.

نتایج آموزش شبکه به شرح زیر است:



شکل ۱-۴: نتایج الگوریتم Policy Gradient با استفاده از شبکه CNN

همانطور که از نمودار مشخص است الگوریتم پس حدود ۲۰۰ ایپاک به بیشترین پاداش خود و کمترین میزان خطا می‌رسد، حال برای مقایسه عمل‌کرد با حالت عادی نمودار خطا و پاداش به همراه حالت عادی نیز رسم شد:



شکل ۲-۴: مقایسه خروجی شبکه Policy Gradient عادی و CNN

با توجه به نمودار مدل عادی بسیار کندتر به خطای صفر رسیده است و همچنین متوسط پاداش به دست آمده آن در مراحل آخر آموزش از حالت CNN کمتر است.

حال برای مقایسه دقیق‌تر از آزمون t-test یکطرفه برای مقایسه عمل‌کرد دو مدل در ۵۰ ایپاک آخر آموزش جهت بررسی فرض زیر استفاده می‌کنیم:

فرض صفر: میانگین پاداش مدل عادی از مدل CNN بالاتر یا مساوی است.

فرض جایگزین: میانگین پاداش مدل CNN از مدل عادی بالاتر است.

که فرض صفر با مقدار p-value برابر با 1.83e-11 با قاطعیت رد می‌شود.

جدول ۱-۴: هایپرپارامتر های مدل

Index	Name	Value
۱	Hidden layer neurons	۳۳۶۰
۲	Convolution Kernel Size	۴
۳	Stride	۲
۴	MaxPool Kernel Size	۴
۵	Activation function	Relu
۶	Learning rate	۰.۰۰۱
۷	Discount Factor	۰.۹۹
۸	Environment related	Default
۹	Epochs	۵۰۰
۱۰	Optimizer	Adam

منابع

[۱] A. Joy, “Pros and cons of reinforcement learning,” *Pythonista Planet*, 31-Mar-2019. .

[۲] J. Peters, “Policy gradient methods,” *Scholarpedia J.*, vol. 5, no. 11, p. 3698, 2010

[۳] What is the downside of deep reinforcement learning? When shouldn’t it be “used?,” *Quora*. [Online]. Available: https://www.quora.com/What-is-the-downside-of-deep-reinforcement-learning-When-shouldnt-it-be-used. [Accessed: 03-Feb-2023] .

[۴] S. Ulyanin, “Breaking down Richard Sutton’s policy gradient with PyTorch and lunar lander,” *Towards Data Science*, 16-Oct-2019. [Online]. Available: https://towardsdatascience.com/breaking-down-richard-suttons-policy-gradient-9768602cb63b. [Accessed: 03-Feb-2023] .

[۵] L. Weng, “Policy gradient algorithms,” *Github.io*, 08-Apr-2018 .

[۶] Observations — highway-env documentation,” *Readthedocs.io*. [Online]. “Available: https://highway-env.readthedocs.io/en/latest/observations/index.html. [Accessed: 03-Feb-2023].