رانندگی خودکار در محیط بزرگراه مبتنی بر یادگیری سیاست با استفاده از روش‌های یادگیری تقویتی توزیعی

مهدی ملائی[۱] عبدالله امیرخانی[۲]

۱- دانش آموخته کارشناسی ارشد- دانشکده مهندسی خودرو- دانشگاه علم و صنعت ایران- تهران- ایران

۲- استادیار- دانشکده مهندسی خودرو- دانشگاه علم و صنعت ایران - تهران- ایران

amirkhani@iust.ac.ir

چکیده: این مقاله به ارائه یک روش یادگیری مبتنی بر یادگیری تقویتی جهت طراحی یک ناظر به منظور رانندگی خودکار در محیط بزرگراه می‌پردازد. با توجه به تصادفی بودن شرایط رانندگی در بزرگراه و همچنین درنظر گرفتن شرایط واقعی تر رانندگی، از مزایای یادگیری تقویتی توزیعی عمیق بهره گرفته شده است. در این مقاله برای اولین بار جهت یادگیری سیاست‌های رانندگی استفاده از روش‌های یادگیری تقویتی توزیعی تابع کمی تمام پارامتری شده (FQF) و شبکه کمی ضمنی (IQN) پیشنهاد شده است. برای آموزش عامل، استفاده از داده‌های دوربین، لیدار و ترکیب آن دو پیشنهاد شده است. به منظور استفاده از ترکیب دو نوع داده، ساختار شبکه چند ورودی را به خدمت گرفته ایم. همچنین برای ارزیابی روش‌های پیشنهاد شده، از شبیه ساز رانندگی در بزرگراه که در نرم افزار Unity توسعه یافته است، استفاده شده است. تحقق خودرو خودران در شبیه ساز مورد نظر به کمک سیستم‌های کمک راننده صورت پذیرفته است. افزون بر این، ارزیابی عامل براساس یادگیری سیاست رانندگی که قادر به انتخاب عمل صحیح برای هدایت خودور باشد نیز انجام شده است. در راستای ارزیابی بهتر روش‌ها دو معیار تغییرات سرعت و تغییرات لاین را برای یادگیری سیاست رانندگی بررسی کرده‌ایم. نتایج بدست آمده از مقاله با روش‌هایی نظیر شبکه Q عمیق (DQN)، شبکه Q عمیق رگرسیون کمی (QR-DQN) که پیش تر ارائه شده بود مقایسه گردید. نتایج بدست آمده نشان دهنده آن است که الگوریتم‌های پیشنهادی توانایی یادگیری سیاست‌های مناسب رانندگی در محیط بزرگراه را دارند. همچنین روش FQF عملکرد بهتری نسبت به IQN و سایر روش‌هایی که در گذشته پیاده سازی شده‌اند از خود نشان می‌دهد.

واژه‌های کلیدی: یادگیری تقویتی توزیعی، خودرو خودران، سیستم‌های کمک راننده

نوع مقاله: پژوهشی

DOI :10.52547/jiaeee.19.2.207

تاریخ ارسال مقاله: ۱۳۹۹/۱۲/۲۷

تاریخ پذیرش مشروط مقاله: ۱۴۰۰/۰۵/۰۵

تاریخ پذیرش مقاله: ۱۴۰۰/۰۶/۱۶

نام نویسنده‌ی مسئول: دکتر عبدالله امیرخانی

نشانی نویسنده‌ی مسئول: تهران - میدان رسالت - خیابان هنگام - خیابان دانشگاه - دانشگاه علم و صنعت ایران – دانشکده مهندسی خودرو

۱- مقدمه

اشتباهات انسانی در رانندگی، همواره یکی از تاثیر گذارترین عوامل موجود در تصادفات و سوانح جاده‌ای بوده است. به همین دلیل در چند دهه اخیر، توجه و مطالعات پژوهشگران و صنایع خودروسازی جهت کاهش و یا رفع این اشتباهات به اوج خود رسید. سیستم‌های کمک راننده نتیجه تحقیق و توسعه در این حوزه هستند [۱]. این سیستم‌ها با هدف کمک رسانی در شرایط خاص و بحرانی طراحی و ساخته می‌شوند. سیستم‌های کمک راننده با اطلاعاتی که از حسگرها و دوربین‌ها بدست می‌آورند گاهی از طریق اعلان خطر و گاهی با مداخله و تصمیم گیری مستقیم موجب کاهش تصادفات در رانندگی و یا به حداقل رساندن خسارات می‌شوند. بسیاری از سیستم‌های کمک راننده در حال حاضر تجاری شده و در خودروها به کار گرفته می‌شوند که از آن جمله می‌توان به سیستم ترمز ضد قفل (ABS[۱])، کروز کنترل تطبیقی (ACC[۲])، سیستم حفظ لاین حرکت و مشابه این‌ها اشاره کرد [۲-۴].

پیاده سازی و اجرای سیستم‌های کمک راننده با توجه به شرایط واقعی جاده‌ها نیازمند روش‌های یادگیری هستند که کاری زمان بر و دشوار است. از سویی دیگر به کار گیری این سیستم ها در کنار هم نیز می‌تواند منجر به تحقق یک خودرو خودران شود. خودروهای خودران همواره به عنوان یکی از مهم ترین عوامل در بهبود ایمنی در حمل و نقل [۵]، کاهش محدودیت‌های ترافیکی [۶] و بهینه سازی مصرف انرژی [۷] یاد می‌شوند. این خودروها نسبت به سیستم‌های کمک راننده نیازمند روش‌های یادگیری با سطوح بالاتر جهت تصمیم گیری‌های پیچیده‌تر در شرایط جاده‌ای مختلف هستند. از این رو پیاده سازی آن‌ها از دشواری بیشتری رنج می‌برد [۸]. در این مقاله به ارائه برخی از روش‌های یادگیری تقویتی عمیق جهت کنترل یک وسیله نقلیه با استفاده از سیستم‌های کمک راننده تجاری شده و موجود در خودروها نظیر سیستم کروز کنترل و سیستم کنترل تغییر لاین در محیط بزرگراه پرداخته‌ایم.

یادگیری تقویتی در کنار یادگیری تحت نظارت و یادگیری غیر نظارتی، یکی از الگوهای اساسی در یادگیری ماشین است. این روش یادگیری مبتنی بر تعامل عامل با محیطی که در آن قرار دارد صورت می‌گیرد و اقدامات در آن بصورت پی در پی و بر اساس پردازش تصمیم گیری مارکوف (MDP[۳]) و با ملاک پاداش و مجازات انجام می‌شود [۹]. هدف در یادگیری تقویتی این است که عامل با توجه به محیطی که در آن قرار دارد به روش آزمون و خطا سعی در حل یک مسئله داشته باشد با این منظور که میزان پاداشی که به آن تعلق می‌گیرد بیشینه شود [۱۰، ۱۱]. یادگیری تقویتی عمیق، با ادغام یادگیری تقویتی و یادگیری عمیق ایجاد می‌شود که یکی از موثرترین روش‌های یادگیری و همچنین شبیه ترین روش به یادگیری انسان است [۱۲، ۱۳]. یادگیری تقویتی عمیق و پیشرفت‌های صورت پذیرفته در این حوزه، توجهات زیادی را در سال‌های اخیر به خود جلب کرده است و بصورت چشم گیری در زمینه‌های مختلف نظیر بازی‌های رایانه‌ای [۱۴، ۱۵]، حفظ سلامت [۱۶، ۱۷]، رباتیک [۱۸، ۱۹] و پردازش تصویر [۲۰، ۲۱] تاکنون به کار گرفته شده است. با همه این توصیفات، الگوریتم‌های یادگیری تقویتی عمیق در حالت کلی در محیط‌هایی که اثری از احتمال و تصادف حضور دارد چندان خوب عمل نمی‌کند. زیرا در این روش یادگیری میزان پاداشی که به یک عمل تخصیص داده می‌شود مقداری معین است درحالیکه شرایط محیط عامل یک محیط تصادفی و احتمالاتی است [۲۲]. برای حل این مشکل الگوریتم‌های یادگیری تقویتی توزیعی ارائه شدند. این الگوریتم‌ها پاداشی که به یک عمل آینده تخصیص داده می‌شود را بصورت یک توزیع در نظر می‌گیرند. بنابراین امکان لحاظ کردن فضای احتمالی و تصادفی در این الگوریتم فراهم می‌شود [۲۳].

ما در این مقاله، به منظور شبیه سازی حرکت یک خودرو در محیط بزرگراه، از شبیه ساز ارائه شده در [۲۴] استفاده کردیم. در این شبیه ساز، خودرو در محیط بزرگراه در حال حرکت است و هدف از فرآیند آموزش این است که خودرو مسافت بزرگراه را با سرعت مناسب یعنی زمان کمتر طی کند و همچنین به منظور حفظ رفاه سرنشینان و ایمنی، کمترین میزان تغییرات لاین را نیز داشته باشد. جهت تحقق یک خودرو خودران در این شبیه ساز از سیستم‌های کمک راننده‌ای که بصورت تجاری در خودروها به کار گرفته می‌شود نظیر ACC، سیستم حفظ لاین حرکت و ترمز اضطراری خودکار (AEB[۴]) استفاده شده است. همچنین علاوه بر سیستم‌های نامبرده، به منظور تقویت ادراک از محیط اطراف یک دوربین در جلو و نیز یک لیدار در خودرو تعبیه شده است. با توجه به اینکه اتفاقات در محیط بزرگراه بصورت تصادفی و احتمالاتی رخ می‌دهند فرآیند آموزش و ارزیابی عامل به کمک الگوریتم یادگیری تقویتی توزیعی عمیق انجام می‌پذیرد. بدین منظور برخی از روش‌های موجود در این حوزه به کار گرفته شده است. روش کنترل خودرو نیز به این صورت است که بجای تنظیم میزان سرعت و تغییر درجه چرخ به کمک فرمان در طول بزرگراه، از حالت‌های حفظ لاین حرکت، تغییر لاین حرکت و ACC جهت تنظیم شتاب خودرو استفاده شده است. همچنین با استفاده از روش‌های یادگیری عمیق جهت فشرده سازی داده‌ها و ترکیبشان با هم که حاصل از تصاویر دوربین موجود در جلوی خودرو و اطلاعات لیدار است، ساختار شبکه‌ای چند ورودی نیز طراحی شده است.

۲- مرور ادبیات

مطالعات در حوزه کنترل وسایل نقلیه به دو دسته کلی تقسیم می‌گردد [۸]. دسته اول کنترل با روش‌های کلاسیک و دسته دوم کنترل به کمک الگوریتم‌های یادگیری ماشین است. در سال‌های اخیر یادگیری ماشین توجهات زیادی را در حوزه خودروهای خودران به خود جلب کرده است. یادگیری تقویتی یکی از مهم ترین و

شاخص‌ترین روش‌هایی است که در زمینه کنترل آن‌ها کاربرد دارد [۲۵]. با توجه به این موضوع که با رفتار غیر طبیعی یک وسیله نقلیه حین رانندگی ممکن است رفتار سایر خودروها نیز بصورت غیرقابل پیش بینی در آید، در [۲۶] یک تابع پاداش پیش گویانه بر اساس خطای پیش بینی یک شبکه عمیق پیش بینی کننده ارائه شده است که قادر به مدل کردن گذر محیط اطراف است.

در [۲۷] وراس[۵] و موسی[۶] با توجه به چالش‌های موجود در سیستم‌های حمل و نقل هوشمند، به مطالعه کاربرد یادگیری عمیق در آن‌ها پرداختند. با توجه به این واقعیت که مدل‌های یادگیری عمیق نقشی مهم در یادگیری تقویتی دارند آن‌ها موفق شدند به کمک شبکه‌های عصبی مصنوعی غیر خطی بر چالش‌های معمول در زمینه کاربردهای مبتنی بر داده حمل و نقل هوشمند غلبه کنند. در مرجع [۲۸] نویسندگان با استفاده از چهارچوب کنترل پشت به پشت موفق به آموزش یک شبکه عصبی پیچشی (CNN[۷]) با استفاده از یادگیری تحت نظارت شدند که قادر به یادگیری سیاست هدایت فرمان بود. با این حال مدل آن‌ها تنها قادر بود تا با حفظ لاین تطبیق یابد. در مرجع [۲۹] به منظور تجزیه مسئله رانندگی خودران به مسائل تشخیص خودرو، تشخیص لاین روشی مبتنی بر CNN ارائه شد که در آن سیستم پیشنهادی به کمک یک دیتاست بزرگراه دنیای واقعی مورد ارزیابی قرار می‌گرفت. برای سناریو رانندگی با زمان طولانی، نویسندگان [۳۰] از مزایای یادگیری تقویتی بهره گرفتند. همچنین از روش‌های گرادیان سیاست به منظور حفظ مسیر ایجاد شده بهره گرفتند که قادر به تضمین ایمنی بود. در مرجع [۳۱] نویسندگانبه منظور بهبود سیاست‌های رانندگی چند بعدی با استفاده از شبکه‌های Q، یک تابع پاداش را به چند تابع پاداش تجزیه کردند. در [۳۲] به منظور دسترسی به احتمال رسیدن به هدف در هر زوج حالت-عمل استفاده از یک چک کننده مدل احتمالی ارائه شد. روش ارائه شده قادر بود تا با تعریف یک آستانه توسط کاربر روی احتمال، موفق به شناسایی عمل ایمن شود.

نویسندگان مرجع [۳۳] با بهره گیری از مزایای شبکه Q عمیق و نقد کننده عمل قطعی عمیق (DDAC[۸]) جهت مشخص کردن مقدار مناسب ترمز، شتاب و زاویه فرمان توانستند در آموزش الگوریتم یادگیری تقویتی موفق باشند. اگرچه روش آن‌ها برای کنترل سطح پایین یک خودرو خودران مناسب بود اما در تضمین ایمنی آن موفق نبودند. برای حل مشکل ذکر شده در [۳۴] روشی مبتنی بر ادغام یادگیری تقویتی و رشته پتانسیل ارائه شد. نویسندگان توانستند روشی برای رانندگی پیشنهاد کنند که در آن فرآیند یادگیری رانندگی عامل ابتدا با استفاده از روش گرادیان سیاست قطعی عمیق (DDPG[۹]) در یک محیط بدون مانع انجام می‌شد. سپس بعد از این مرحله و اینکه عامل موفق به یادگیری رانندگی در یک محیط بدون مانع شد، پتانسیل رفع مانع ایجاد شده و تحت یک سیاستی برای جلوگیری از برخورد به عامل افزوده می‌شود.

جهت یادگیری زاویه فرمان و مقدار شتاب با هدف جلوگیری از برخورد با موانع نویسندگان [۳۵] روشی با اعمال DDPG پیشنهاد کردند. در مرجع [۳۶]، یک رویکرد کنترلی جهت آموزش چگونگی رانندگی به یک خودرو خودران مورد مطالعه قرار گرفت. رویکرد پیشنهادی یادگیری Q عمیق با تجربیات فیلتر شده (DQFE[۱۰]) نام دارد. در [۳۷]، به کمک تصاویر RGB جمع آوری شده توسط دوربین خودرو، روش های یادگیری تقویتی در شبیه ساز ترافیکی CARLA مورد تجزیه و تحلیل قرار گرفتند. همچنین در مرجع [۳۸]، نویسندگان یک رویکرد یادگیری تقویتی عمیق چند عامله پیشنهاد کردند. در این مطالعه وسایل نقلیه خودران با هماهنگی یاد می‌گیرند که چگونه بتوانند در سناریو بزرگراه رفتار کنند.

۳- یادگیری تقویتی توزیعی

الگوریتم یادگیری تقویتی در حالت کلی به بررسی تعامل یک عامل با محیط می‌پردازد و هدف آن یادگیری عامل به گونه‌ای است که مجموع پاداشی که به ازای عمل‌هایش بدست می‌آورد بیشینه شود [۲۲]. انواع روش‌های یادگیری تقویتی به سه دسته کلی تقسیم می‌شوند که عبارتند از [۳۹]: ۱- مبتنی بر مقدار ۲- مبتنی بر سیاست ۳- نقد عمل. دسته اول تلاش بر مدلسازی امید بازگشت کل دارد. ابتدا در آن یک تابع مقدار تصادفی انتخاب شده و هر مرحله تابع مقدار جدید به خود می‌گیرد و این روند تا زمانیکه به یک مقدار بهینه دست یابد ادامه می‌یابد [۲۲]. در رویکرد دوم، هدف یادگیری مجموعه‌ای از پارامترهاست به جای فضای حالت، همین امر سبب می‌شود تا از نظر حافظه و حجم محاسبات بهتر باشد زیرا از تشکیل مدل پردازش مارکوف (MDM[۱۱]) و تابع مقدار اجتناب می‌کند [۴۰]. در دسته سوم، یک شبکه عصبی عمل کننده حالت‌ها را انتخاب کرده و یک شبکه عصبی نقاد مقادیر را پیش بینی می‌کند [۴۱]. این روش از خواص هردو روش قبل بهره می‌برد. در این مقاله رویکرد ما استفاده از یادگیری تقویتی مبتنی بر مقدار است.

یکی از چالش‌ها در این حوزه مبتنی بر رویکرد پیش بینی کمیت است که به عنوان مقدار یاد می‌شود. در واقع، مقدار به میانگین مجموع پاداش کاسته شده گفته می‌شود که از ابتدای آموزش تا پایان اپیزودی که در حال انجام است مورد بررسی قرار می‌گیرد. همانطور که اشاره شد، چالشی که در این روش وجود دارد این است که اگر تعداد حالت‌ها افزایش یابد، تابع مقدار غیرخطی می‌شود [۲۲]. از سویی دیگر شرایط رانندگی در محیط بزرگراه یک وضعیت کاملا تصادفی است و از عدم قطعیت بالایی برخوردار است. به همین دلیل اختصاص یک مقدار اسکالر به پاداش مرتبط با یک عمل کار دشواری است [۲۴]. از این رو به سراغ الگوریتم یادگیری تقویتی توزیعی رفتیم که امکان تخمین زدن میزان پاداش آینده بصورت یک توزیع را داشته باشد.

۳-۱- شبکه Q عمیق (DQN[۱۲])

در بخش‌های قبل برخی از مفاهیم یادگیری تقویتی بیان و برخی از مزیای آن معرفی شد. با این وجود، وقتی که افزایش تعداد حالت‌ها برای ظرفیت حافظه مشکل ساز شود، چندان موثر نیست. علاوه بر این در دنیای واقعی عامل با حالت‌های پیوسته سر و کار دارد نه گسسته که نیازمند مسائل کنترل پیوسته است. با توجه به تعداد حالت‌ها و کنترل پیوسته که موجب پیچیدگی محیطی می‌شود که عامل در آن فعالیت می‌کند، یک شبکه عصبی عمیق بجای جدول Q یادگیری تقویتی قرار می‌گیرد که نگاشتی از حالت‌ها به عمل‌ها دارد [۴۲]. در DQN انتخاب هایپر پارامترها، معماری شبکه و یادگیری در طول فرآیند آموزش صورت می‌پذیرد. همچنین DQN امکان جستجو و کسب دانش از محیط‌های بدون ساختار را فراهم می‌کند که موجب رفتار در سطح انسان می‌شود [۴۳].

پیکسل‌های تصویردر روش DQN بعنوان مشاهده درنظر گرفته می‌شوند. این پیسکل‌ها به یک شبکه عصبی عمیق اعمال شده و مقادیر Q متناسب برای هر عمل بدست می‌آید. همچنین با روش ε-greedy عمل مناسب انتخاب می‌شود. با انتخاب یک عمل، مشاهده و پاداش مرحله بعد نیز مشخص می‌شود [۴۲]. در نتیجه این مقادیر بعنوان تجربه شخصی در حافظه ذخیره می‌شوند که بصورت (O_t, A_t, R_t, O_{t+1}) نشان داده می‌شود که به ترتیب نشان دهنده مشاهده، عمل، پاداش آینده و مشاهده آینده می‌باشند. در DQN براساس یادگیری نظارتی، تابع زیان بصورت مربع اختلاف میان هدف و مقدار پیش بینی شده تعیین می‌شود. همچنین با توجه به گذر عامل از یک حالت به حالت بعدی و عملی که انجام می‌دهد و پاداشی که متناسب با آن کسب می‌کند، به روز رسانی وزن‌های شبکه با هدف کاهش زیان، انجام می‌شود. برای محاسبه مقدار پیش بینی شده و مقدار هدف، در طول فرآیند یادگیری از دو شبکه Q جداگانه استفاده می‌شود که یکی شبکه Q محلی و دیگری شبکه Q هدف نامیده می‌شود [۴۳]. تابع زیان در DQN بصورت رابطه (۱) تعریف می‌شود.

$$L(\theta) = ((R_t + \gamma \max Q(O_{t+1}, A_{t+1}; \theta^-) - Q(O_t, A_t; \theta))^2 \qquad (۱)$$

که در رابطه بالا، θ پارامتر شبکه عصبی و θ^- پارامتر شبکه عصبی هدف است. γ، عامل تنزیل نامیده می‌شود که میزان مجموع مورد انتظار پاداش آینده در نظر گرفته شده را نشان می دهد.

۳-۲- رقابت شبکه Q عمیق دوتایی[۱۳]

الگوریتم شبکه Q عمیق دوتایی به مشکل بیش برآورد کردن در DQN می‌پردازد که ناشی از حضور مقدار بیشینه Q برای حالت بعدی در به روز رسانی یادگیری Q است [۴۴]. عملگر ماکزیمم روی مقادیر Q بیشینگی بایاس را موجب می‌شود که ممکن است منجر به عملکرد نامناسب عامل در محیط‌های خاص شود. یک مقدار Q هدف مشابه رابطه (۲) را در نظر بگیرید:

$$Q^*(s,a) = E_{s' \sim \varepsilon}[r + \gamma \max_{a'} Q^*(s',a') \mid s,a] \qquad (۲)$$

گرفتن بیشینه مقادیر بیش برآورد شده بطور تلویحی تخمین بیشینه مقدار است. همین بیش برآورد موجب بیشینه بایاس در یادگیری می‌شود [۴۴]. راه حل این مشکل استفاده از دو تخمین زن مقدار Q است که جدا از هم هستند و برای به روز رسانی یکدیگر به کار گرفته می‌شوند. با کمک این تخمین زن‌های جداگانه می‌توان تخمین مقادیر Q عمل‌های انتخاب شده توسط تخمین زن‌های مخالف را بدون بایاس کرد. بنابراین می‌توان از بیشینگی بایاس به کمک جداکردن به روز رسانی از تخمین بایاس شده، اجتناب کرد [۴۵]. DQN دوتایی برای مشخص کردن مقدار Q هدف از معادله (۳) استفاده می‌کند.

$$Q^*(s,a) = E_{s' \sim \varepsilon}[r + \gamma \max_{a'} Q^*(s',a') \mid s,a] \qquad (۳)$$

که θ_t و θ^- به ترتیب، نشان دهنده پارامترهای شبکه اولیه و هدف در لحظه t هستند. همانطور که مشخص است عمل بهینه از طریق شبکه اولیه انتخاب شده و فرآیند ارزیابی و تخمین مقدار Q به کمک شبکه هدف صورت می‌پذیرد که بصورت رابطه (۴) نمایش داده می‌شود.

$$Q_{target} = r_{t+1} + \gamma Q(s_{t+1}, a^*; \theta_t^-) \ , \ a^* = \arg\max Q(s_{t+1}, a; \theta_t) \qquad (۴)$$

با انجام این دو کار بصورت هم زمان فرآیند انتخاب عمل‌ها و ارزیابی آن‌ها توسط شبکه‌های متفاوت صورت می‌پذیرد [۴۵].

در روش دوئل DQN دو تخمین متفاوت وجود دارد، اولا باید بررسی شود حالتی که عامل در آن قرار دارد چقدر خوب است (تخمین مقدار حالت داده شده) و ثانیا عمل انجام شده در یک حالت چقدر خوب است (تخمین مزیت عمل در یک حالت) [۳۹]. بنابراین $Q(s,a)$ را می‌توان به دو مولفه $V(s)$ و $A(s,a)$ تجزیه کرد که به ترتیب نشان دهنده ارزش قرار داشتن در حالت s و مزیت عمل انجام شده a در حالت s است. پس خواهیم داشت:

$$Q(s,a) = V(s) + A(s,a) \qquad (۵)$$

این دو جریان را به کمک یک لایه تجمیع برای ایجاد تخمین مقادیر Q هر عمل a در حالت s ترکیب می‌کنیم. نتیجتاً می‌توان به کمک روش DQN دوتایی این مقادیر Q را به منظور نزدیک‌تر شدن به مقدار هدف آموزش داد. با جداسازی مقدار و مزیت، شبکه مقدار تخمین

دقیق تری از مقدار و مزیت‌ها بدست می‌آورد و عمل فرآیند یادگیری بهبود می‌یابد [۴۶].

C51 -۳-۳

همانطور که گفته شد روش‌های یادگیری تقویتی توزیعی مبتنی بر مقدار سعی در مدل کردن انتظار کل بازگشت دارد. الگوریتم C51 یکی از روش‌های یادگیری تقویتی توزیعی است در سال ۲۰۱۷ که توسط بلمار[۱۴] و همکارانش ارائه شد [۴۷]. این الگوریتم در ابتدا یک گام تصویر ابتکاری را اجرا کرده و سپس عمل کمینه کردن واگرایی kl میان تخمین و به روزرسانی بلمن تصویر شده را انجام می‌دهد و در نهایت احتمال توزیع را به عنوان خروجی تحویل می‌دهد. در C51 احتمال بازگشت بین دو مجموعه گسسته از مقادیر ثابت محدود است و احتمال هر مقدار از طریق تعاملی که عامل با محیط انجام می‌دهد یاد گرفته می‌شود [۴۸].

نقاط روی محور افقی توزیع در C51 نشان‌دهنده ساپورت‌ها هستند که همان مجموع پاداش مورد انتظار آینده را نشان می‌دهند. همچنین نقاط روی محور عمودی نشان دهنده میزان احتمال برای هر ساپورت است. مقدار ساپورت به کمک رابطه (۶) محاسبه می‌شود [۴۷]:

$$\Delta z := \frac{V_{\max} - V_{\min}}{N-1} \qquad (۶)$$

که V_{min} و V_{max} نشان دهنده میزان بیشینه و کمینه مقدار است و همچنین N نیز تعداد ساپورت می‌باشد. همانطور که مشخص است مقدار ساپورت در این روش یک مقدار ثابت است و بصورت زیر بدست می‌آید [۴۷]:

$$z_i = V_{\min} + i\Delta z, \ for\ i = 0,...,N \qquad (۷)$$

خروجی شبکه، احتمال هر ساپورت به ازای همه عمل‌ها است. مقادیر Q بصورت زیر محاسبه می‌شود [۴۷]:

$$Q(s_{t+1}, a) = \sum_{i=1}^{N} z_i\, p_i(s_{t+1}, a) \qquad (۸)$$

مشابه روش DQN، اینجا نیز از تجربیات گذشته و روش شبکه هدف استفاده می‌شود. شبکه توزیع را به عنوان خروجی پیش بینی می‌کند در نتیجه هدف آموزش نیز باید یک توزیع بدست آید نه یک اسکالر [۴۷]. بنابراین به روز رسانی بلمن برای هر ساپورت بصورت زیر انجام می‌شود:

$$\mathrm{Tz}_i = [r_t + \gamma_t z_i]_{V_{\min}}^{V_{\max}}\ ,\ i = 1,...,N \qquad (۹)$$

T_{Z_i} نشان دهنده به روز رسانی بلمن است.

پس از محاسبه به روز رسانی بلمن، تصویر از T_{Z_i} به ساپورت Z_i انجام می‌شود. بنابراین احتمال توزیع هدف با توجه به هر ساپورت بدست می‌آید. در نهایت زیان C51 به کمک زیان آنتروپی متقاطع بصورت زیر محاسبه می‌شود [۴۸].

$$-\sum_i m_i \log p_i(s_t, a_t) \qquad (۱۰)$$

m_i نشان دهنده هر ساپورت است به ازای $i=1,...,N$.

QR-DQN[۱۵] شبکه Q عمیق رگرسیون کمی -۴-۳

QR-DQN از این جهت با C51 تفاوت دارد که مقادیر کمی را بصورت مستقیم یاد می‌گیرد [۴۹]. این روش به کمک رگرسیون کمی، بازگشت کمی را در کسرهای کمی یکنواخت و ثابت محاسبه می‌کند. همچنین موجب کمینه شدن زیان هابر میان توزیع به روز شده بلمن و توزیع بازگشت فعلی می‌شود. مشابه روش C51، QR-DQN نیز مقدار یا احتمال تابع کمی را در مکان‌های ثابت شده تقریب می‌زند و همچنین به جای تخمین یک اسکالر برای هر زوج حالت و عمل یک توزیع مقادیر در نظر می‌گیرد. با داشتن توزیع، سیاست بهبود می‌یابد نسبت به حالتی که تنها میانگین مقادیر را داشته باشیم.

در C51، ۵۱ مکان ثابت برای مقدار توزیع در نظر گرفته می‌شود و احتمالات این نقاط یادگرفته می‌شود درحالیکه در روش QR-DQN با درنظر گرفتن احتمالات ثابت، این پارامترسازی جابجا می‌شود [۵۰]. با تغییر تعداد لایه‌های خروجی در DQN به $|A|*N$ به QR-DQN نزدیک‌تر می‌شویم، که N یک هایپر پارامتری است که تعداد مقادیر کمی هدف را می‌دهد و A فضای عمل است.

$$Z_\theta(x,a) := \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i}(x,a) \qquad (۱۱)$$

Z_θ مقدار Q معین است. همچنین با تغییر تابع زیان با زیان هوبر کمی، QR-DQN حاصل می‌شود. بنابراین با اعمال این تغییرات و با در نظر گرفتن آستانه k در زیان هوبر کمی روابط (۱۲) و (۱۳) بدست خواهد آمد [۴۹].

$$\sum_{i=1}^{N} E_j[\rho_{\tau_i}^k(T\theta_j - \theta_i(O_t, A_t))] \qquad (۱۲)$$

$$\rho_{\tau_i}^k(u) = \begin{cases} \frac{1}{2}u^2\left|\tau - \delta_{\{u<0\}}\right| & if\ |u| \le k \\ k\left(|u| - \frac{1}{2}k\left|\tau - \delta_{\{u<0\}}\right|\right), & otherwise \end{cases} \qquad (۱۳)$$

در روابط بالا θ ساپورت توزیع، τ مرکز مقادیر توزیع شده تجمیعی و u، تفاضل زمانی است که از معادلات (۱۴) و (۱۵) محاسبه می شوند [۴۹].

$$u = T\theta_j - \theta_i(s,a) \quad (۱۴)$$

$$\tau_i = \frac{2(i-1)+1}{2N}, i = 1,\dots,N \quad (۱۵)$$

۳-۵- شبکه‌های کمی ضمنی (IQN[۱۶])

همانطور که در مباحث قبل توضیح داده شد هدف یادگیری توزیعی آموختن توزیع بازگشت‌های تصادفی است که عامل از محیط خود دریافت می‌کند. الگوریتم‌های معمول یادگیری تقویتی نظیر DQN، بجای یادگیری توزیع مقدار، میانگین در کل توزیع را بصورت مستقیم تخمین می‌زنند. الگوریتم C51 نسبت به DQN بهبودهایی را از خود نشان داد اما از برخی از محدودیت‌ها رنج می‌برد [۵۱]. به منظور بهبود محدودیت‌های C51، الگوریتم QR-DQN ارائه شد. در این روش از رگرسیون کمی برای کمینه کردن معیار واسرشتاین استفاده می‌شود. الگوریتم IQN با توجه به ایده QR-DQN ارائه شده است که بهبودهای قابل توجهی هم داشته است [۵۲].

دو تفاوت میان الگوریتم IQN و QR-DQN وجود دارد که عبارتند از: ۱- IQN مقادیر τ (توزیع یکنواخت) را از طریق برخی توابع تفاضلی تخمین می‌زند، بطور مثال شبکه عصبی ۲- نمونه برداری متفاوت از τ توزیع‌های مستقل پیوسته [۵۱]. با تقریب تابع کمی Z داریم:

$$Z_\tau(s,a) \approx f(\psi(s), \phi(\tau))_a \quad (۱۶)$$

ψ لایه‌ای است که حالت‌ها را رمز گذاری می‌کند، ϕ تقریب مقادیر نمونه برداری شده از τ است و f حاصل ضرب عضو به عضو، ψ و ϕ است یعنی $(\psi \odot \phi)$. برای $\phi(\tau)$ داریم [۵۱]:

$$\phi_j(\tau) := \mathrm{ReLU}\left(\sum_{i=0}^{n-1} \cos(\pi i \tau)\omega_{ij} + b_j\right) \quad (۱۷)$$

w و b پارامترهای شبکه هستند.

رابطه زیر تابع مقدار-عمل را نشان می‌دهد که براساس سیاست π و عمل a در حالت s مشخص می‌شود.

$$Q_\pi(s,a) = E_\pi[G_t \mid S_t = s, A_t = a] \quad (۱۸)$$

تابع Q در IQN بوسیله θ مشخص می‌شود و از طریق رابطه زیر بدست می‌آید [۵۱]:

$$Q(s,a;\theta) = E_{\tau\sim U([0,1])}[Z_\tau(s,a;\theta)] \approx \frac{1}{N}\sum_{i=1}^{N} Z_{\tau_i}(s,a;\theta) \quad (۱۹)$$

این الگوریتم با گرفتن نمونه‌ای از توزیع بازگشت $Z(s,a;\theta)$، مقدار-عمل را از طریق رابطه بالا پیدا می‌کند. نتایج بدست آمده در [۵۱] نشان از بهبود عملکرد در بازی‌های Atari-57 نسبت به روش‌هایی که پیش تر توضیح داده شد، دارد. شکل (۱) تفاوت میان الگوریتم IQN و DQN را نشان می‌دهد.

۳-۶- تابع کمی تمام پارامتری شده (FQF[۱۷])

در الگوریتم IQN اگر تعداد مقادیر کمی و ظرفیت شبکه به بی نهایت میل کند، شبکه قادر به تخمین تابع کمی خواهد بود. اما این موضوع در عمل ممکن نیست ازین رو کارایی نمونه‌ها باید برای تعداد محدودی از کسرهای کمی مورد بررسی قرار گیرد [۵۳]. از آنجا که در IQN نمونه برداری معمولا بجای یادگیری تابع کمی تقریبی به یادگیری شبکه کمی تلویحی می‌پردازد بنابراین احتمالات نمونه برداری لزوماً منجر به تخمین بهتر تابع کمی نسبت به احتمالات ثابت نمی‌شود. از این رو در الگوریتم FQF، کسرهای کمی و مقادیر کمی مربوطه بر خلاف روش‌های IQN و QR-DQN پارامتری می‌شوند. FQF شامل دو شبکه می‌شود: ۱- شبکه پیشنهادی کسر که برای هر زوج عمل-حالت مجموعه‌ای از کسرهای کمی ایجاد می‌کند، ۲- شبکه مقدار کمی که احتمالات را به مقادیر کمی نگاشت می‌دهد [۵۳].

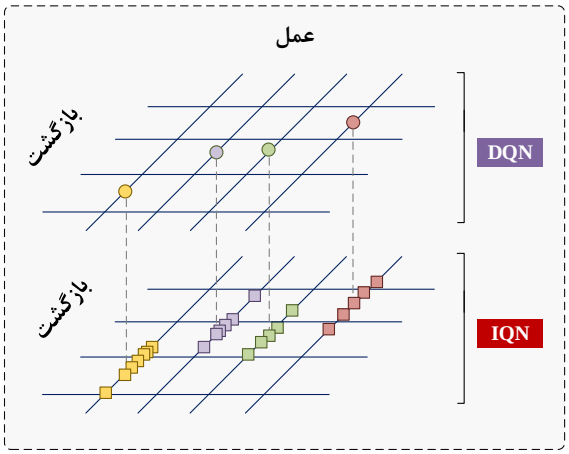
با ایجاد کسرهای کمی توسط شبکه پیشنهادی کسر، امکان یادگیری مقدار شبکه کمی فراهم می‌شود. همچنین کسرهای خود تنظیم شده موجب بهبود تخمین توزیع واقعی می‌شوند. در FQF، به منظور پیش بینی کسر کمی، N مقدار کمی قابل تنظیم برای N کسر قابل تنظیم تخمین زده می‌شود. در این روش توزیع بازگشت به کمک ترکیب وزن داری از N تابع ضربه تخمین زده می‌شود [۵۳]:

$$Z_\theta(s,a) := \sum_{i=0}^{N-1} (\tau_{i+1} - \tau_i)\delta_{\theta_i(s,a)} \quad (۲۰)$$

که در آن δ_z ضربه در $z \in R$ و $\tau_0, \dots, \tau_N$ کسر قابل تنظیم را نشان می‌دهد.

شبکه پیشنهادی کسر از طریق کمینه کردن فاصله واسرشتین میان مقدار واقعی و تخمینی توزیع آموزش می‌بینند. شبکه مقدار کمی نیز از طریق زیان هوبر کمی ساخته می‌شود مشابه آنچه که در IQN داشتیم. نهایتاً مقادیر Q در روش FQF به کمک رابطه (۲۱) محاسبه می‌گردد.

$$Q(s,a) = \sum_{i=0}^{N-1} (\tau_{i+1} - \tau_i) F^{-1}_{Z,\omega_2}(\hat{\tau}_i) \quad (۲۱)$$



شکل (۱): تفاوت میان الگوریتم DQN و IQN

که F^{-1}_{Z,w_2} نشان دهنده تابع کمی واقعی با پارامتر w_2 است.

معکوس تابع توزیع تجمعی، F^{-1}_{Z}، بصورت زیر تعریف می‌شود که همان تابع کمی است [۵۳]:

$$F_Z^{-1}(p) := inf\{x \in R : p \le F_Z(z)\} \quad (۲۲)$$

p نشان دهنده کسر کمی است.

در رابطه (۲۱) برای $\hat{\tau}$ نیز داریم [۵۳]:

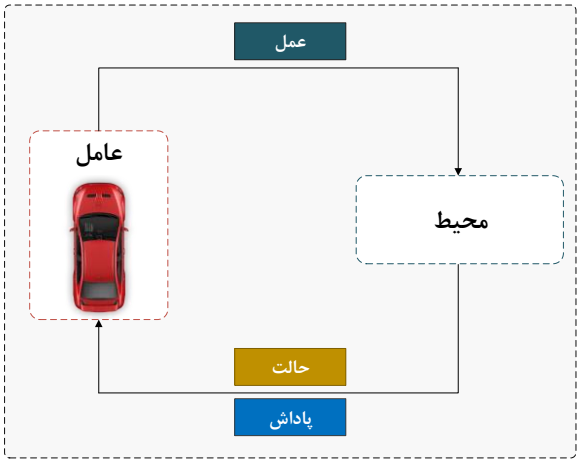
$$\hat{\tau}_i = \frac{\tau_i + \tau_{i+1}}{2} \quad (۲۳)$$

که $\tau_1=0$ و $\tau_N=1$.

۴- تعریف مسئله

پردازش تصمیم مارکوف (MDP) یک چارچوب محاسباتی است که قادر به حل مسائل در حوزه یادگیری تقویتی بوده و از یک تاپل $<S,A,T,R,\gamma>$ تشکیل شده است که S,A,T,R,γ به ترتیب نشان دهنده مجموعه حالت‌ها، مجموعه عمل‌ها، یک مدل گذر، تابع پاداش و عامل تنزیل می‌باشد [۵۴]. شکل (۲) خلاصه‌ای از MDP است. با توجه به آن عامل و محیط در لحظات متفاوت با هم در تعامل قرار دارند، در هر لحظه از زمان عامل اطلاعاتی را درباره حالت محیط بدست می‌آورد و براساس آن یک عملی را انتخاب می‌کند همچنین به ازای عملی که انجام می‌دهد پاداش را دریافت می‌کند. هدف MDP آموزش یک عامل برای دستیابی به سیاستی است که می‌تواند منجر به بازگشت بیشینه پاداش‌های تجمعی در پی مجموعه‌ای از عمل‌ها در یک یا برخی از حالت‌ها شود.

در مقاله حاضر، از محیط شبیه ساز که در [۲۴] ارائه شد، استفاده شده است. فرآیند یادگیری عامل نیز باتوجه به MDP صورت می‌گیرد که در آن وسیله نقلیه میزبان با سایر خودروها و محیط بزرگراه در تعامل قرار دارد.



شکل (۲): مدل پردازش تصمیم مارکوف

جدول (۱): پارامترهای پاداش

عمل	پاداش
تغییر لاین (η_c)	-۰/۲۵
برخورد (r_c)	-۱۰
سبقت (r_{ov})	۰/۵

در ادامه به شرح بخش‌های مختلف مسئله نظیر فضای حالت S، فضای عمل A و تابع پاداش R جهت یادگیری سیاست رانندگی خواهیم پرداخت.

حالت‌های مشاهده از طریق تصاویر دوربین در جلوی خودرو و لیدار بدست می‌آید. فاصله از هر مانعی از طریق لیدار به ازای هر درجه چرخش محاسبه می‌شود. با توجه به سیستم‌های کمک راننده‌ای که در این خودرو به کار رفته است برخی از عمل‌ها برای حرکت آن درنظر گرفته شده است. برای حرکت در راستای طولی خودرو با توجه به سیستم ACC سه نوع عمل تعریف شده است: ۱- سرعت کروز کنترل با $v+v_{CC}$ که v_{CC} مقدار سرعتی است که باید به سرعت اصلی خودرو اضافه شود که مقدار آن $5\ km/h$ است. این سرعت به دلیل کاهش مدت زمان طی مسیر حرکت در نظر گرفته می‌شود. ۲- سرعت کروز کنترل با v، یعنی وسیله نقلیه سرعت فعلی خود را حفظ کند. ۳- سرعت کروز کنترل با $v-v_{CC}$ که برای کاهش سرعت خودرو لحاظ می‌شود. برای حرکت در راستای عرضی خودرو نیز ۳ عمل با توجه موقعیت آن درنظر گرفته شده است: ۱- تغییر لاین به چپ ۲- حفظ لاین ۳- تغییر لاین به راست. با این وجود، با توجه به اینکه حرکت خودرو باید در دو راستای طولی و عرضی صورت گیرد در مجموع ۵ عمل به وسیله نقلیه اعمال می‌شود که عبارتند از:

- بدون عمل
- افزایش شتاب
- کاهش شتاب
- تغییر لاین به راست
- تغییر لاین به چپ

در یادگیری تقویتی برای هر عمل یک پاداشی در نظر گرفته می‌شود و همانطور که گفته شد هدف MDP یافتن سیاستی است که موجب بیشینه شدن مقادیر مورد انتظار پاداش آینده شود. ازین رو، نیازمند به طراحی پاداش جهت یادگرفتن سیاست مناسب رانندگی هستیم. با توجه به حرکت خودرو در بزرگراه سه هدف ضروری باید لحاظ شود: ۱- یافتن سیاستی که موجب حرکت خودرو با بشینه میزان سرعت شود، ۲- حرکت کردن خودرو بدون برخورد با موانع در مسیر، ۳- کاهش تغییرات لاین در طول حرکت. در نتیجه با توجه به اهدافی که بیان شد یک تابع پاداشی بصورت زیر طراحی شده است [۲۴]:

$$r_{total}(v) = r_v(v) + r_c + r_{lc} + r_{ot} \quad (۲۴)$$

که r_V پاداشی است که به سرعت تعلق می‌گیرد، η_C جریمه تغییر لاین است، r_C جریمه برخورد با موانع است و r_{ot} پاداشی است که به ازای هر سبقت گرفتن از دیگر خودروها به عامل تعلق می‌گیرد. پاداش سرعت، r_V، بصورت زیر تعریف می‌شود:

$$r_v(v) = \frac{v - v_{\min}}{v_{\max} - v_{\min}} r_{v,\max} \quad (۲۵)$$

که v نشان دهنده سرعت فعلی خودرو، $v_{\max}$ بیشینه سرعتی است که برای خودرو درنظر گرفته شده است و مقدار آن km/h ۸۰ است و $v_{\min}$ کمینه سرعتی است که برای خودرو لحاظ شده است و مقدار آن نیز km/h ۴۰ است. $r_{v,\max}$ نیز نشان دهنده ضریب پاداشی است که به سرعت عامل تعلق می‌گیرد که مقدار آن ۱ لحاظ می‌شود. تمامی مقادیر گفته شده در جدول (۱) مشخص شده است.

با توجه به آنچه که در [۲۴] ارائه شد، در کنار بررسی روش های گفته شده مبتنی بر داده‌های دوربین و داده‌های لیدار، از ساختار شبکه چند ورودی نیز بهره گرفته شده است. ساختار شبکه پیشنهاد شده در [۲۴] به منظور آموزش در شکل (۳) نشان داده شده است. هدف از آموزش وسیله نقلیه، یافتن سیاست $\pi(a|o)$ است که از فضای مشاهده حالت O باید به بهترین حرکت بعدی در فضای عمل A در یک محیط رانندگی احتمالاتی نگاشت دهد. به منظور دریافت ویژگی‌های مفید از داده‌های لیدار و دوربین توسط شبکه، شبکه باید اطلاعات مکانی-زمانی را از حسگرهای گفته شده بدست آورد.

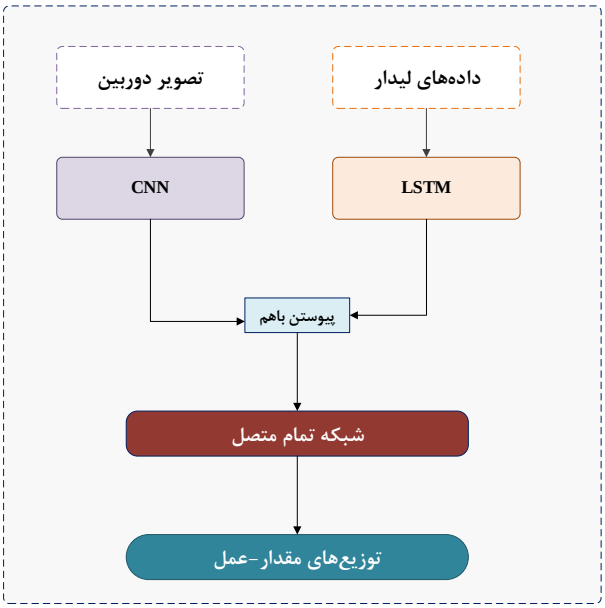
جدول (۲): هایپرپارامترهای شبکه

هایپر پارامترها	فعالساز	نوع	داده
اندازه پچ = (۸×۸) تعداد فیلترها = ۳۲	ReLU	پیچش	داده دوربین
اندازه پچ = (۴×۴) تعداد فیلترها = ۶۴			
اندازه پچ = (۳×۳) تعداد فیلترها = ۶۴			
گام های زمان = ۴ تعداد حالت‌های سلول = ۲۵	-	LSTM	داده لیدار
تعداد واحدها = ۵۱۲	ReLU	تمام متصل	داده‌های بهم پیوست شده

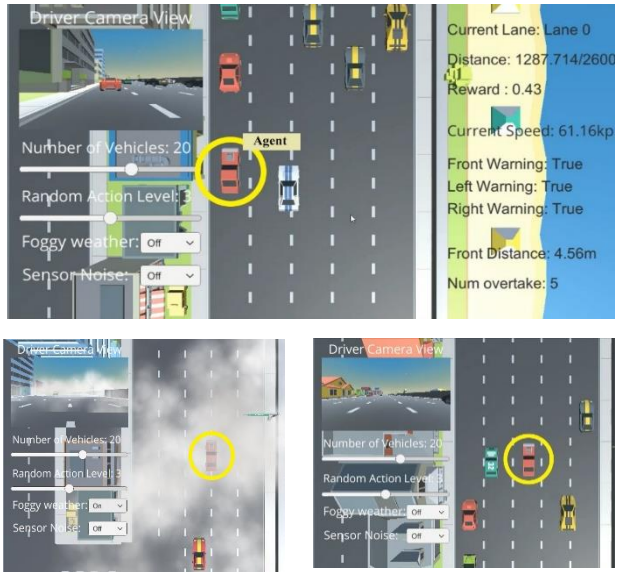
از این رو از یک شبکه عصبی پیچشی به منظور استخراج ویژگی‌های مکانی از تصاویر دوربین و نیز حافظه کوتاه مدت (LSTM[۱۸]) جهت استخراج ویژگی‌های زمانی از داده‌های لیدار استفاده شده است. خروجی داده‌های LSTM برداری با تعداد ۳۶۵ عضو است. همچنین خروجی CNN نیز یک بردار ویژگی می‌باشد. داده‌های حاصل شده از این دو شبکه بهم متصل شده و در قالب یک بردار ورودی به یک شبکه تمام متصل اعمال می‌شوند. در واقع بردار LSTM در امتداد بردار ویژگی ناشی از CNN به این شبکه وارد می‌شود. اطلاعات شبکه و هایپرپارامترها در جدول (۲) مشخص شده است. در شاخه LSTM داده های لایه دار وارد یک بخش پیش پردازش شده و بر اساس الگوریتم های تعریف شده در این بخش نرمالیزه می‌شوند، به این معنا که تابع توزیع آن‌ها به شکل یک تابع توزیع استاندارد گاوسی تبدیل می‌شود و یک بردار ویژگی با ابعاد ۱ در ۳۶۵ بدست می آید، سپس وارد سلول‌های شبکه عصبی بازگشتی (RNN[۱۹]) می شود که شامل ۲۷ عدد است. این ۲۷ سلول خود یک سلول LSTM را تشکیل می دهند. از سویی دیگر تصاویر ورودی نیز وارد شاخه CNN می‌شوند، CNN شامل سه لایه پیچشی است که به ترتیب به ابعاد ۳۲، ۶۴ و ۶۴ می باشند. خروجی هرکدام از این لایه ها وارد مسطح کننده شده تا خروجی شبکه های پیچشی صاف شده و مستقیم وارد شبکه تمام متصل شوند. همچنین از سه لایه انکدر استفاده شده تا منجر به کاهش بعد در ویژگی های استخراج شده از لایه‌های پیچشی شوند. نتیجه این شاخه نیز یک بردار ویژگی است که با بردار ویژگی حاصل از شاخه LSTM بهم متصل شده و وارد دسته بندی کننده می‌شوند که همان شبکه تمام متصل است. برای آموزش شبکه‌ها و تعیین وزن شبکه انتخابی نیز از روش خاویر[۲۰] و همچنین به به منظور بهینه کردن پارامترهای آموزش از الگوریتم بهینه ساز آدام[۲۱] استفاده شده است.

علاوه برآنچه که گفته شد از آنجاییکه شبکه باید انتخاب عمل را باتوجه به احتمال ذاتی که در محیط وجود دارد انجام دهد، از چارچوب یادگیری تقویتی توزیعی استفاده شده است. از طریق

لایه‌های تمام متصل توزیع مورد نظر برای الگوریتم‌های بیان شده ایجاد می‌شود.



شکل (۳): ساختار شبکه چند ورودی



شکل (۴): نمونه‌هایی از محیط شبیه ساز

با توجه به مقادیر Q تخمین زده شده در روابط الگوریتم‌ها، بهترین عمل a^*، آن است که بیشترین مقدار Q را دارد و از میان مقادیر Q محدود در فضای عمل انتخاب می‌شود [۲۴]، یعنی:

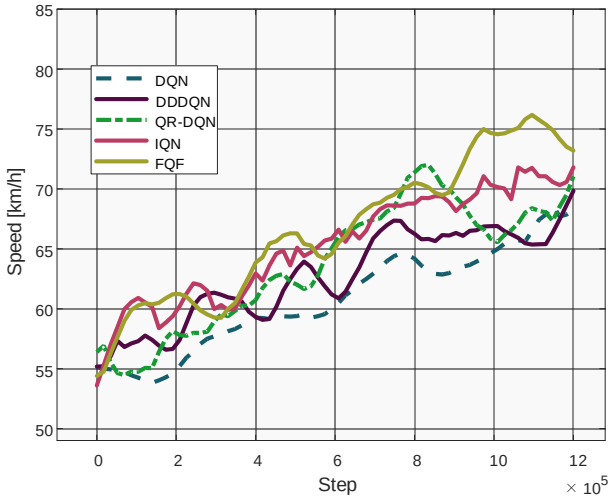
$$a^* = \arg\max_a Q(o,a) \qquad (۲۶)$$

۵- شبیه سازی و نتایج

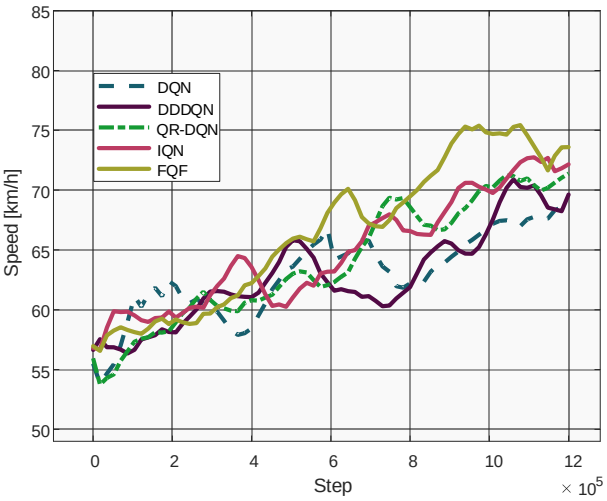
همانطور که پیش تر به آن اشاره شد شبیه سازی این مقاله با توجه به شبیه ساز ارائه شده در [۲۴] انجام شده است. این شبیه ساز در نرم افزار Unity ساخته شده است که در آن یک عامل یا وسیله نقلیه میزبان در یک بزرگراه شروع به رانندگی می‌کند. بزرگراه شامل ۵ لاین جاده است و حرکت سایر خودروها شامل تغییر لاین و سبقت نیز با این فرض که با یکدیگر برخوردی ندارند، بصورت تصادفی انجام می‌شود. نمونه‌هایی از شبیه ساز در شکل (۴) نشان داده شده است.

آموزش و ارزیابی عامل در محیط بزرگراه با استفاده از داده‌های لیدار، داده‌های دوربین و ترکیبی از هردوی آن‌ها انجام شده است. در حالت سوم همانطور که در قسمت قبل نیز توضیح داده شد، با استفاده از یک CNN و LSTM اطلاعات مفید از دوربین و لیدار استخراج و ترکیب شده و به عنوان ورودی به شبکه تمام متصل وارد می‌شوند و مقادیر Q بدست می‌آید. آموزش عامل نیز به کمک روش‌های یادگیری تقویتی توزیعی که معرفی نیز شد انجام شده است.

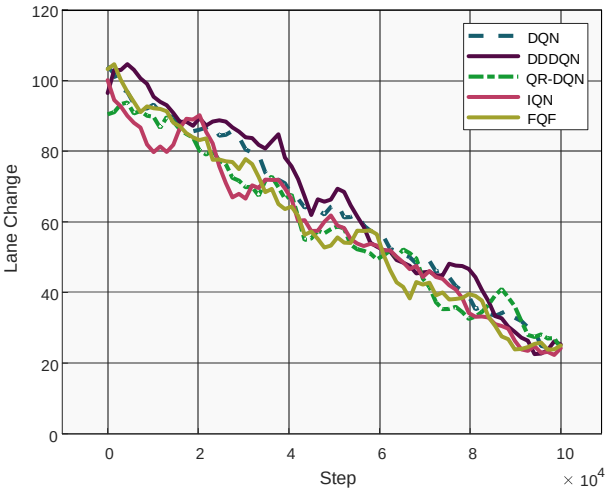
شکل (۵) تا شکل (۷) تغییرات سرعت در طی آموزش را نشان می‌دهند. این تغییرات به ازای گام‌های مختلف آموزش بوده و در ۵ اپیزود انجام شده است همچنین آموزش عامل به ترتیب بر اساس داده‌های بدست آمده از دوربین به تنهایی، لیدار به تنهایی و ترکیبی از دوربین و لیدار در شکل‌های (۵) تا (۷) رسم شده است.



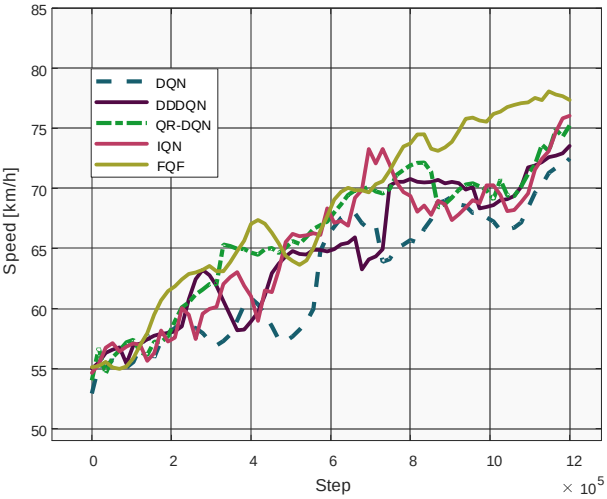
شکل (۵): نرخ تغییرات سرعت طی فرآیند آموزش مبتنی بر داده‌های تصویر



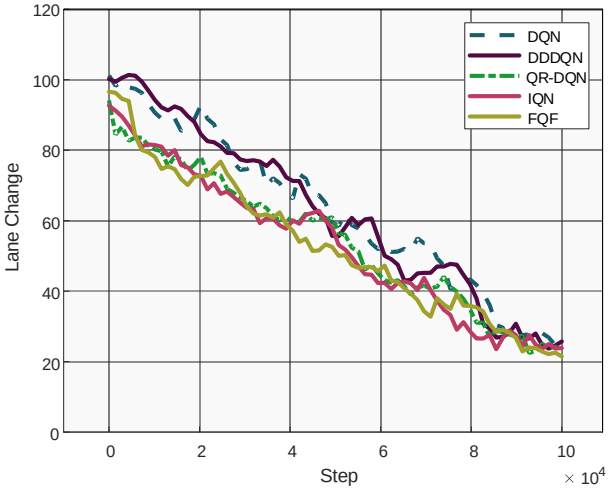
شکل (۶): نرخ تغییرات سرعت طی فرآیند آموزش مبتنی بر داده‌های لیدار



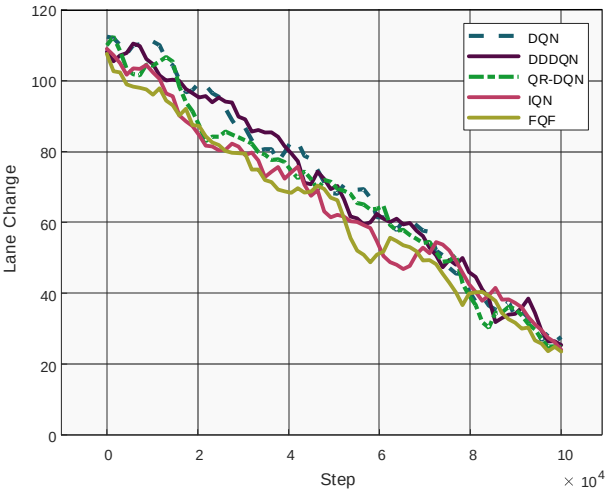
شکل (۹): نرخ تغییرات لاین طی فرآیند آموزش مبتنی بر داده‌های لیدار



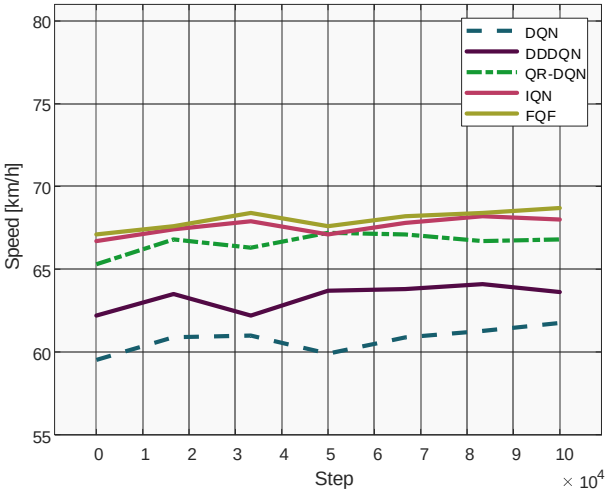
شکل (۷): نرخ تغییرات سرعت طی فرآیند آموزش مبتنی بر داده‌های چند ورودی



شکل (۱۰): نرخ تغییرات لاین طی فرآیند آموزش مبتنی بر داده‌های چند ورودی



شکل (۸): نرخ تغییرات لاین طی فرآیند آموزش مبتنی بر داده‌های تصویر



شکل (۱۱): میانگین تغییرات سرعت طی فرآیند استنتاج مبتنی بر داده‌های تصویر

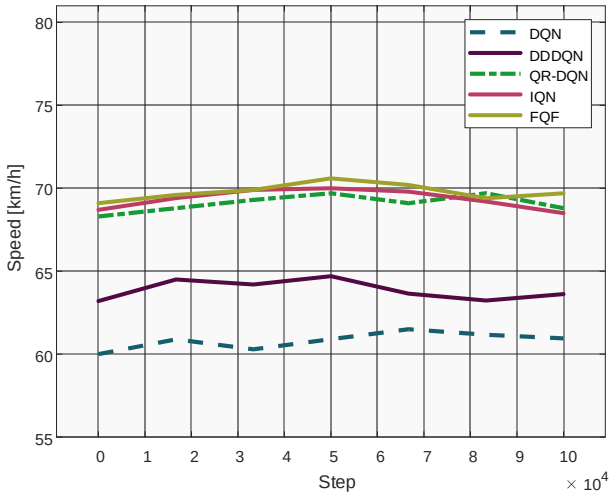
همانطور که مشخص است در همه شکل‌ها و براساس همه روش‌ها با پیشرفت آموزش سرعت خودرو در حال افزایش است. این موضوع یکی از اهداف ما نیز بود که در تعریف تابع پاداشی به آن اشاره کردیم.

شکل (۸) تا شکل (۱۰) نیز تغییرات لاین در حین آموزش را بر اساس گام‌های مختلف آموزش نشان می‌دهند. این اشکال نیز برحسب اطلاعاتی (دوربین، لیدار، چند ورودی) که برای آموزش شبکه از آن استفاده شده است ترسیم شده است. همانطور که مشخص است برای تغییرات لاین نیز روند کاهشی در طول آموزش برای همه روش‌ها مشاهده می‌شود که این نیز یکی از اهداف ما در آموزش عامل بود.

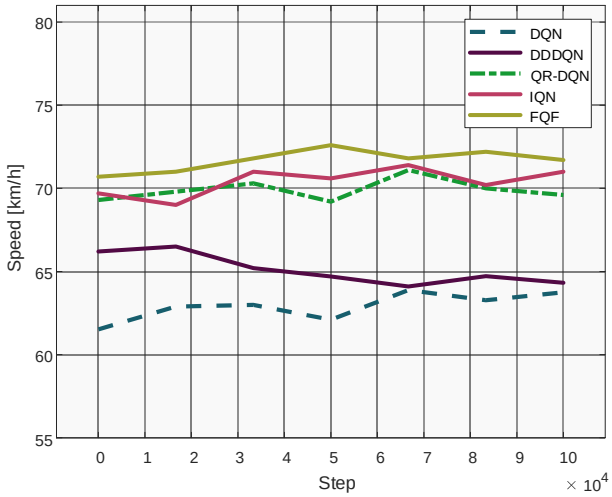
پس از آموزش، فرآیند استنتاج انجام می‌شود. شکل (۱۱) تا شکل (۱۳) به ترتیب نشان دهنده استنتاج با توجه به داده‌های دوربین، لیدار و چند ورودی برای تغییرات سرعت با توجه روش‌های مختلف DRL می‌باشد. در استنتاج به منظور سخت تر کردن شرایط رانندگی، نویز و حالت مه آلود نیز به شبیه ساز افزوده می‌شود. مقادیر میانگین بدست آمده از روش‌های مختلف DRL در استنتاج در جدول (۳) مشخص شده است. میانگین سرعت برای روش FQF با توجه به داده‌های دوربین ۶۸/۰۰ و برای داده‌های لیدار ۶۹/۷۸ و برای حالت چند ورودی این مقدار ۷۱/۶۸ می‌باشد. همانطور که مشخص است بیشترین میزان برای میانگین سرعت مرتبط با روش FQF است. این امر نشان دهنده این است که این الگوریتم توانایی بهتری نسبت به سایر در تصمیم گیری در سطح بالاتر را داشته با توجه به نتایج اشاره شده این مقدار برای روش FQF با داده‌های چند ورودی از سایر بیشتر است. بنابراین مشخص است که عملکرد شبکه با استفاده از داده‌های چند ورودی نتایج بهتری نسبت به حالت تک ورودی دارد.

شکل (۱۴) تا شکل (۱۶) نیز میانگین تغییرات لاین را با توجه به تغییر گام برای فرآیند استنتاج نشان می‌دهد. مجدداً با توجه به مقادیر ارائه شده در جدول (۳) مشاهده می‌شود که برای فاکتور تغییر لاین نیز روش FQF نسبت به سایر روش‌های DRL عملکرد بهتری را از خود نشان می‌دهد.

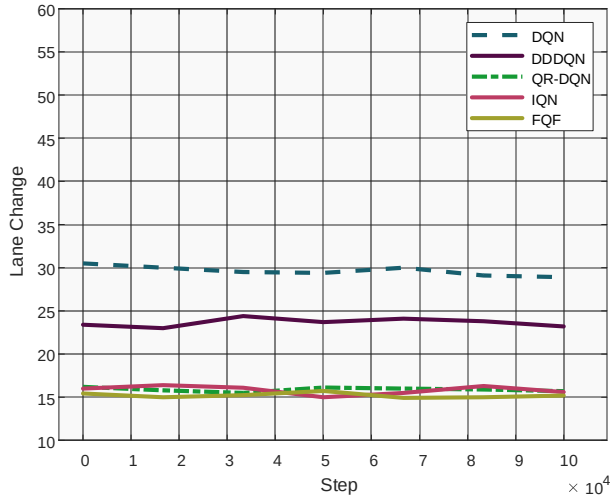
هرچند روش‌های IQN و QR-DQN نیز عملکرد نزدیک به روش FQF از خود نشان می‌دهند. این میزان برای الگوریتم FQF با توجه به داده‌های دوربین، ۱۴/۹۲، برای داده‌های لیدار مقدار ۴۳/۲۵ و در حالت چند ورودی نیز مقدار ۱۵/۰۱ می‌باشد. مشخص است که عملکرد تغییر لاین برای همه روش‌های مربوط به شبکه با اطلاعات دوربین نسبت به دو حالت دیگر بهتر است به این علت که در این حالت خودرو با سرعت به مراتب کمتری نسبت به دو حالت دیگر حرکت می‌کند و همین امر سبب می‌شود تا کمترین تغییرات لاین را نیز داشته باشد. عملکرد روش‌های مختلف با توجه به داده‌های لیدار نسبت به روش‌های مشابه در دو حالت دیگر به مراتب بدتر است، علت آن هم این است که نویزی که در استنتاج به شبیه ساز اضافه شد اثرات مخرب بیشتری بر روی لیدار نسبت به سایر داشته است و نهایتاً موجب عملکرد بدتر در این حالت شده است.



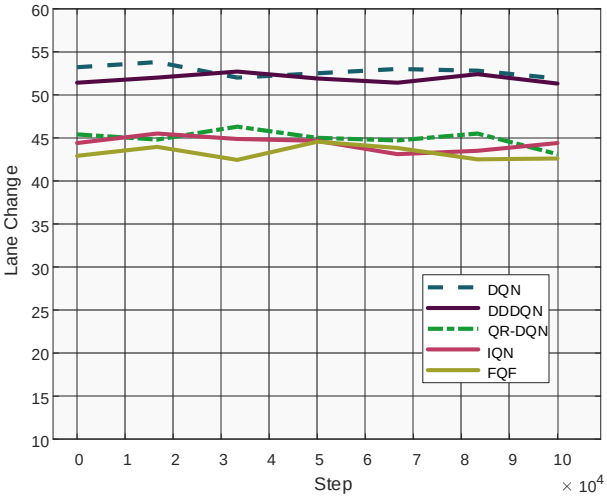
شکل (۱۲): میانگین تغییرات سرعت طی فرآیند استنتاج مبتنی بر داده‌های لیدار



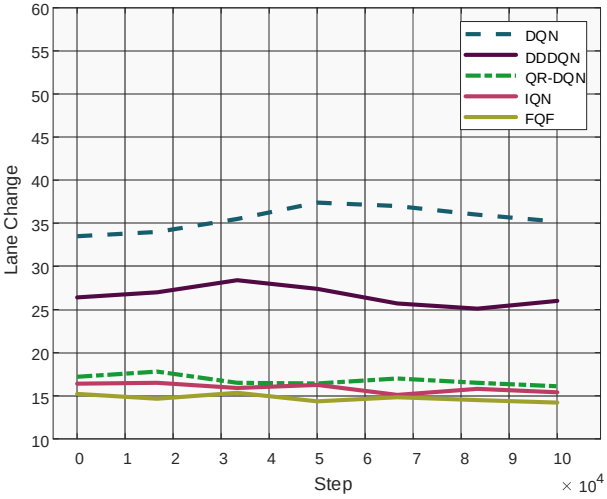
شکل (۱۳): میانگین تغییرات سرعت طی فرآیند استنتاج مبتنی بر داده‌های چند ورودی



شکل (۱۴): میانگین تغییرات لاین طی فرآیند استنتاج مبتنی بر داده‌های تصویر



شکل (۱۵): میانگین تغییرات لاین طی فرآیند استنتاج مبتنی بر داده‌های لیدار



شکل (۱۶): میانگین تغییرات لاین طی فرآیند استنتاج مبتنی بر داده‌های چند ورودی

جدول (۳): میانگین تغییرات سرعت و لاین طی فرآیند استنتاج

الگوریتم‌های یادگیری تقویتی توزیعی	میانگین سرعت (km/h)	میانگین تغییرات لاین
DQN (Image)	۶۰/۷۴	۲۹/۶۲
DQN(LIDAR)	۶۰/۸۱	۵۲/۷۴
DQN(Multi-Input) [۲۱]	۶۲/۹۱	۳۵/۵۱
DDDQN(Image)	۶۳/۳۰	۲۳/۶۵
DDDQN(LIDAR)	۶۳/۸۷	۵۱/۸۷
DDDQN(Multi-Input) [۲۱]	۶۵/۱۰	۲۶/۵۷
QR-DQN(Image) [۲۱]	۶۶/۶۰	۱۵/۴۸
QR-DQN(LIDAR) [۲۱]	۶۹/۱۰	۴۴/۹۷
QR-DQN(Multi-Input) [۲۱]	۶۹/۹۰	۱۶/۷۸
IQN(Image)	۶۷/۵۸	۱۵/۴۲
IQN(LIDAR)	۶۹/۳۵	۴۴/۳۴
IQN(Multi-Input)	۷۱/۴۱	۱۵/۹۰
FQF(Image)	۶۸/۰۰	۱۴/۹۲
FQF(LIDAR)	۶۹/۷۸	۴۳/۲۵
FQF(Multi-Input)	۷۱/۶۸	۱۵/۰۱

۶- نتیجه‌گیری

در این مقاله یادگیری سیاست‌های رانندگی به کمک الگوریتم‌های یادگیری تقویتی مورد بررسی قرار گرفت. بدین منظور از یک شبیه ساز برای رانندگی در بزرگراه بهره گرفته شد. وسیله نقلیه در این شبیه سازی به سه سیستم کمک راننده مجهز شده و عمل‌ها برای رانندگی براساس امکانات آن‌ها انتخاب شد. از آنجاییکه رانندگی در محیط بزرگراه شرایط احتمالی و حالت‌های تصادفی زیادی را شامل می‌شود الگوریتم‌های یادگیری تقویتی توزیعی عمیق به کار گرفته شدند. هدف یافتن سیاستی بود که بتواند منجر به حرکت در بزرگراه با سرعت مناسب و کمترین تغییر در تعداد لاین باشد. البته فاکتور سبقت گرفتن نیز به منظور ارزیابی بهتر کیفیت روش‌های گفته شده مورد بررسی قرار گرفت. نتایج بدست آمده حاکی از آن بود که الگوریتم‌های توزیعی نتایج بهتری نسبت به الگوریتم یادگیری تقویتی کلی دارند. از میان روش‌های پیشنهادی، روش FQF تخمین بهتری از مقدار واقعی توزیع داشت و نتایج بهتری از خود نسبت به سایر روش‌ها در فاکتورهای شبیه سازی شده نشان داد. بعنوان یک کار آینده می‌توان تعداد بیشتری از سیستم‌های کمک راننده را برای سناریو پیچیده تری جهت رانندگی خودرو طراحی کرده و از الگوریتم‌های یادگیری تقویتی توزیعی به منظور یادگیری سیاست درست رانندگی استفاده کرد.

مراجع

[1] F. Jiménez, J. E. Naranjo, J. J. Anaya, F. García, A. Ponz, and J. M. Armingol, "Advanced driver assistance system for road environments to improve safety and efficiency," Transp. Res. procedia, vol. 14, pp. 2245–2254, 2016.

[2] M. Flad, L. Fröhlich, and S. Hohmann, "Cooperative shared control driver assistance systems based on motion primitives and differential games," IEEE Trans. Human-Machine Syst., vol. 47, no. 5, pp. 711–722, 2017.

[3] C. Braunagel, W. Rosenstiel, and E. Kasneci, "Ready for take-over? A new driver assistance system for an automated classification of driver take-over readiness," IEEE Intell. Transp. Syst. Mag., vol. 9, no. 4, pp. 10–22, 2017.

[4] H. M. Fahmy, M. A. Abd El Ghany, and G. Baumann, "Vehicle risk assessment and control for lane-keeping and collision avoidance at low-speed and high-speed scenarios," IEEE Trans. Veh. Technol., vol. 67, no. 6, pp. 4806–4818, 2018.

[5] J. Hawkins and K. Nurul Habib, "Integrated models of land use and transportation for the autonomous vehicle revolution," Transp. Rev., vol. 39, no. 1, pp. 66–83, 2019.

[6] D. Miculescu and S. Karaman, "Polling-systems-based autonomous vehicle coordination in traffic intersections with no traffic signals," IEEE Trans. Automat. Contr., vol. 65, no. 2, pp. 680–694, 2019.

[7] F. Liu, F. Zhao, Z. Liu, and H. Hao, "Can autonomous vehicle reduce greenhouse gas emissions? A country-level evaluation," Energy Policy, vol. 132, pp. 462–473, 2019.
[8] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," IEEE Trans. Intell. Transp. Syst., 2020.
[9] A. Noori, E. Kalhor, M. ali sadrnia, and S. Saboori Rad, "Controlling the Cancer Cells in a Nonlinear Model of Melanoma by Considering the Uncertainty Using Q-learning Algorithm Under the Case Based Reasoning Policy TT - کنترل جمعیت سلول‌های سرطانی در مدل غیرخطی سرطان ملانوما با لحاظ عدم قطعیت با استفاده از الگوریتم یادگیری Q مورد بر مبتنی استدلال سیاست تحت(CBR)," jiaeee, vol. 17, no. 3, pp. 25–37, Sep. 2020.
[10] Z. Wang and T. Hong, "Reinforcement learning for building controls: The opportunities and challenges," Appl. Energy, vol. 269, p. 115036, 2020.
[11] A. Younesi, H. Shayeghi, A. Akbari, and Y. Hashemi, "Design of PSS3B stabilizer using KH Algorithm and Q-Learning for damping Low-frequency Oscillations in SMIB TT - طراحی پایدارساز PSS3B الگوریتم بر اساس KH و برای میراسازی نوسانات فرکانس پایین سیستم قدرت Q-learning تک‌ماشینه," jiaeee, vol. 14, no. 3, pp. 69–77, Dec. 2017.
[12] M. Botvinick, J. X. Wang, W. Dabney, K. J. Miller, and Z. Kurth-Nelson, "Deep reinforcement learning and its neuroscientific implications," Neuron, 2020.
[13] M. Aslani and M. Saadi Mesgari, "Developing Continuous Reinforcement Learning in Distributed Spatial Problems (Case Study: Adaptive Traffic Control) TT - توسعه یادگیری تقویتی پیوسته در مسائل مکانی توزیع یافته (مورد مطالعاتی: کنترل هوشمند چراغ های راهنمایی)," jiaeee, vol. 17, no. 3, pp. 63–78, Sep. 2020.
[14] Y. Zhan, S. Guo, P. Li, and J. Zhang, "A deep reinforcement learning based offloading game in edge computing," IEEE Trans. Comput., vol. 69, no. 6, pp. 883–893, 2020.
[15] C. Han, L. Huo, X. Tong, H. Wang, and X. Liu, "Spatial Anti-Jamming Scheme for Internet of Satellites Based on the Deep Reinforcement Learning and Stackelberg Game," IEEE Trans. Veh. Technol., vol. 69, no. 5, pp. 5331–5342, 2020.
[16] O. Gottesman et al., "Guidelines for reinforcement learning in healthcare," Nat. Med., vol. 25, no. 1, pp. 16–18, 2019.
[17] M. Baucum, A. Khojandi, and R. Vasudevan, "Improving Deep Reinforcement Learning with Transitional Variational Autoencoders: A Healthcare Application," IEEE J. Biomed. Heal. Informatics, 2020.
[18] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, "Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation," Rob. Auton. Syst., vol. 112, pp. 72–83, 2019.
[19] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 610–617, 2019.
[20] N. Xu et al., "Multi-level policy and reward-based deep reinforcement learning framework for image captioning," IEEE Trans. Multimed., vol. 22, no. 5, pp. 1372–1383, 2019.
[21] B. Uzkent, C. Yeh, and S. Ermon, "Efficient object detection in large images using deep reinforcement learning," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 1824–1833.
[22] W. Dabney et al., "A distributional code for value in dopamine-based reinforcement learning," Nature, vol. 577, no. 7792, pp. 671–675, 2020.
[23] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," IEEE J. Sel. Areas Commun., vol. 38, no. 2, pp. 334–349, 2019.
[24] K. Min, H. Kim, and K. Huh, "Deep Distributional Reinforcement Learning Based High-Level Driving Policy Determination," IEEE Trans. Intell. Veh., vol. 4, no. 3, pp. 416–424, 2019, doi: 10.1109/TIV.2019.2919467.
[25] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," Rob. Auton. Syst., vol. 114, pp. 1–18, 2019.
[26] D. Hayashi, Y. Xu, T. Bando, and K. Takeda, "A Predictive Reward Function for Human-Like Driving Based on a Transition Model of Surrounding Environment," in 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 7618–7624.
[27] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 8, pp. 3152–3168, 2019.
[28] M. Bojarski et al., "End to end learning for self-driving cars," arXiv Prepr. arXiv1604.07316, 2016.
[29] B. Huval et al., "An empirical evaluation of deep learning on highway driving," arXiv Prepr. arXiv1504.01716, 2015.
[30] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," arXiv Prepr. arXiv1610.03295, 2016.
[31] W. Yuan, M. Yang, Y. He, C. Wang, and B. Wang, "Multi-Reward Architecture based Reinforcement Learning for Highway Driving Policies," in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 3810–3815.
[32] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Safe reinforcement learning with scene decomposition for navigating complex urban environments," in 2019 IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 1469–1476.
[33] A. E. L. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," Electron. Imaging, vol. 2017, no. 19, pp. 70–76, 2017.
[34] X. Xiong, J. Wang, F. Zhang, and K. Li, "Combining deep reinforcement learning and safety based control for autonomous driving," arXiv Prepr. arXiv1612.00147, 2016.
[35] X. Zong, G. Xu, G. Yu, H. Su, and C. Hu, "Obstacle avoidance for self-driving vehicle with reinforcement learning," SAE Int. J. Passeng. Cars-Electronic Electr. Syst., vol. 11, no. 1, pp. 28–38, 2018.
[36] W. Xia, H. Li, and B. Li, "A control strategy of autonomous vehicles based on deep reinforcement learning," in 2016 9th International Symposium on Computational Intelligence and Design (ISCID), 2016, vol. 2, pp. 198–201.
[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms,"

arXiv Prepr. arXiv1707.06347, 2017.
[38] C. Yu et al., "Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 2, pp. 735–748, 2019.
[39] Z. Zhu and H. Zhao, "A Survey of Deep RL and IL for Autonomous Driving Policy Learning," arXiv Prepr. arXiv2101.01993, 2021.
[40] J. Weng, X. Jiang, W.-L. Zheng, and J. Yuan, "Early action recognition with category exclusion using policy-based reinforcement learning," IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 12, pp. 4626–4638, 2020.
[41] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach," IEEE Trans. Wirel. Commun., vol. 17, no. 1, pp. 680–692, 2017.
[42] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.
[43] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing," IEEE Trans. Veh. Technol., vol. 68, no. 8, pp. 8050–8062, 2019.
[44] V.-H. Bui, A. Hussain, and H.-M. Kim, "Double deep Q-learning-based distributed operation of battery energy storage system considering uncertainties," IEEE Trans. Smart Grid, vol. 11, no. 1, pp. 457–469, 2019.
[45] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep Q-learning model for energy-efficient edge scheduling," IEEE Trans. Serv. Comput., vol. 12, no. 5, pp. 739–749, 2018.
[46] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in International conference on machine learning, 2016, pp. 1995–2003.
[47] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in International Conference on Machine Learning, 2017, pp. 449–458.
[48] M. Rowland, M. Bellemare, W. Dabney, R. Munos, and Y. W. Teh, "An analysis of categorical distributional reinforcement learning," in International Conference on Artificial Intelligence and Statistics, 2018, pp. 29–37.
[49] W. Dabney, M. Rowland, M. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in Proceedings of the AAAI Conference on Artificial Intelligence, 2018, vol. 32, no. 1.
[50] G. Ostrovski, W. Dabney, and R. Munos, "Autoregressive quantile networks for generative modeling," in International Conference on Machine Learning, 2018, pp. 3936–3945.
[51] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in International conference on machine learning, 2018, pp. 1096–1105.
[52] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models," IEEE Trans. neural networks Learn. Syst., vol. 31, no. 7, pp. 2469–2489, 2019.
[53] D. Yang, L. Zhao, Z. Lin, T. Qin, J. Bian, and T. Liu, "Fully parameterized quantile function for distributional reinforcement learning," arXiv Prepr. arXiv1911.02140, 2019.
[54] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A Decision-Making Strategy for Vehicle Autonomous Braking in Emergency via Deep Reinforcement Learning," IEEE Trans. Veh. Technol., vol. 69, no. 6, pp. 5876–5888, 2020, doi: 10.1109/TVT.2020.2986005.

زیرنویس‌ها

[1] Anti-lock Braking System
[2] Adaptive Cruise Control
[3] Markov Decision Process
[4] Automatic Emergency Braking
[5] Veras
[6] Moussa
[7] Convolutional Neural Network
[8] Deep Deterministic Actor Critic
[9] Deep Deterministic Policy Gradient
[10] Deep Q-Learning with Filtered Experiences
[11] Markov Decision Model
[12] Deep Q Network
[13] Dueling Double Deep Q Network
[14] Bellemare
[15] Quantile Regression Deep Q Network
[16] Implicit Quantile Network
[17] Fully Parameterized Quantile Function
[18] Long Short-Term Memory
[19] Recurrent Neural Network
[20] Xavier
[21] Adam