پردیس علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

بررسی روش‌های به‌کارگیری یادگیری تقویتی در بازارهای مالی

نگارنده:
پریناز زارعی

استاد راهنما:
دکتر هدیه ساجدی

پایان‌نامه برای دریافت درجه کارشناسی
در رشته آمار

مرداد ۱۴۰۱

چکیده

در این پروژه به بررسی رویکردی نوآورانه مبتنی بر یادگیری تقویتی برای حل مسئله معاملات الگوریتمی تعیین معامله‌ی بهینه در هر لحظه از یک فعالیت معاملاتی در بازار سهام پرداخته می‌شود. این رویکرد از سیاست معاملاتی DRL[1] جدیدی برای بیشینه کردن شاخص عملکرد نسبت شارپ در طیف گسترده‌ای از بازارهای سهام استفاده می‌کند. رویکرد DRL جدید که «معاملات الگوریتمی کیو-شبکه مبتنی بر یادگیری عمیق» $TDQN$ [2] نامیده می‌شود، الهام گرفته‌شده از الگوریتم محبوب DQN [3] و حاصل تطابق آن با مسئله معاملات الگوریتمی خاص است. آموزش عامل یادگیری تقویتی (RL) حاصله مبتنی بر تولید مسیرهایی مصنوعی از مجموعه محدودی از داده‌های تاریخی بازار سهام است. به منظور ارزیابی عینی عملکرد استراتژی‌های معاملاتی، به بررسی یک روش جدید و دقیق‌تر برای ارزیابی علمکرد می‌پردازیم. به دنبال این رویکرد ارزیابی عملکرد جدید، نتایج امیدوارکننده‌ای برای الگوریتم $TDQN$ گزارش شده‌است.

کلیدواژه‌ها: یادگیری تقویتی عمیق، نسبت شارپ، مسیرهای مصنوعی، سیاست معاملاتی.

[1] Deep Reinforcement Learning
[2] Trading Deep Q-Network
[3] Deep Q-Network

فهرست مطالب

فصل ۱

مقدمه

علاقه به هوش مصنوعی در چند سال گذشته با سرعت بالایی رشد کرده است و هر ساله مقالات متعددی در این حوزه منتشر می‌شوند. ازجمله دلایل اصلی این علاقه رو به رشد، موفقیت‌های چشمگیر تکنیک‌های یادگیری عمیق مبتنی بر مدل‌های ریاضی شبکه عصبی عمیقی (DNN)[1] هستند که مستقیماً از ساختار مغز انسان الهام گرفته شده‌اند.

این تکنیک‌های خاص امروزه نقشی اساسی در زمینه‌های کاربردی بسیاری مانند تشخیص گفتار، طبقه‌بندی تصویر یا پردازش زبان طبیعی دارند. اخیراً، جامعه پژوهشی به حوزه یادگیری تقویتی عمیق DRL نیز به‌موازات یادگیری عمیق علاقه‌مند شده است. خانواده تکنیک‌های DRL با فرآیند یادگیری عاملی هوشمند مرتبط‌اند که:

- در تعامل متوالی با محیطی ناشناخته باشد.
- هدفش بیشینه نمودن پاداش‌های تجمعی باشد.
- از تکنیک‌های یادگیری عمیق برای تعمیم اطلاعات به‌دست‌آمده از تعامل با محیط استفاده کند.

موفقیت‌های متعدد اخیر تکنیک‌های DRL حاکی از توانایی برجسته آن‌ها در حل مسائل پیچیده تصمیم‌گیری متوالی است. امروزه، صنعت نوظهور فناوری مالی، که به اختصار فین‌تک [2] نامیده می‌شود، به‌سرعت در حال رشد است. هدف فین‌تک واضح است: بهره‌گیری گسترده از فناوری در جهت نوآوری و بهبود فعالیت‌های مالی. انتظار می‌رود صنعت «فین‌تک» نحوه رسیدگی به بسیاری از مشکلات مرتبط با تصمیم‌گیری بخش مالی، ازجمله مشکلات مربوط به معامله‌گری، سرمایه‌گذاری، مدیریت ریسک، کشف تقلب و مشاوره مالی را در چند سال آینده متحول سازد. چون این مسائل تصمیم‌گیری معمولاً ماهیتی متوالی داشته و بسیار تصادفی هستند و در محیطی نیمه مشاهده‌پذیر و

[1] Deep Neural Network
[2] FinTech

با عواملی احتمالا متضاد مطرح می‌شوند، بسیار پیچیده هستند. به‌ویژه، معاملات الگوریتمی، که بخش کلیدی صنعت فین‌تک هستند، چالش‌های جالبی را موجب می‌شوند. معاملات الگوریتمی که معاملات کمّی نیز خوانده می‌شوند، به معنای استفاده از رایانه و مجموعه خاصی از قوانین ریاضی در معاملات است. هدف اصلی این پروژه بررسی موضوعات مربوطه برای پاسخ به این سوال است: چگونه می‌توان سیاست (الگوریتم) معاملاتی جدیدی مبتنی بر تکنیک‌های هوش مصنوعی طراحی کرد که بتواند با استراتژی‌های معاملاتی الگوریتمی محبوب پرکاربرد رقابت کند؟ برای پاسخ به این سوال، راه‌حل یادگیری تقویتی عمیق جدیدی برای حل مسئله معاملات الگوریتمی تعیین موقعیت معاملاتی بهینه (در بلندمدت یا کوتاه‌مدت) طی معامله‌ای تجاری در بازار سهام بررسی و تحلیل شده‌است. الگوریتم پیشنهادی که در این پروژه بررسی شده، الهام گرفته شده و حاصل تطابق مسئله تصمیم‌گیری متوالی موردنظر با الگوریتم محبوب کیو-شبکه عمیق (DQN) است. اینکه محیط معاملاتی موردنظرمان ویژگی‌های بسیار متفاوتی (نظیر شدیداً تصادفی بودن و مشاهده‌پذیری بسیار ضعیف) نسبت به ویژگی‌هایی که قبلاً توسط رویکردهای یادگیری تقویتی با موفقیت حل شده‌اند، دارد، حاکی از اهمیت هر چه بیشتر تلاش برای پاسخ دادن به سوال تحقیق دارد. ساختار پروژه به شرح زیر است. ابتدا، مطالعات علمی درباره معاملات الگوریتمی و پیشرفت‌های اصلی مبتنی بر هوش مصنوعی به طور مختصر بررسی شده، سپس مسئله معاملات الگوریتمی موردنظر بیان می‌شود. همچنین در این بخش توضیحاتی درباره‌ی ارتباط معاملات الگوریتمی با رویکرد یادگیری تقویتی (RL) مطرح می‌شود. سپس طراحی کامل استراتژی معاملاتی $TDQN$ مبتنی بر مفاهیم یادگیری تقویتی عمیق توضیح داده می‌شود. متعاقباً، روش جدیدی برای ارزیابی عینی عملکرد استراتژی‌های معاملاتی شرح داده شده و بررسی می‌شود. در نهایت در بخش آخر، نتایج به‌دست‌آمده از استراتژی معاملاتی $TDQN$ ارائه و بحث شده‌است.

۱.۰.۱ مروری بر مطالعات گذشته

در شروع مرور کوتاهی بر مطالعات گذشته، بر دو واقعیت تاکید می‌کنیم. اول این‌که، بسیاری از آثار علمی معتبر در زمینه معاملات الگوریتمی در دسترس عموم نیستند. درواقع همان‌طور که لی (۲۰۱۷) توضیح می‌دهد، چون پای مقدار زیادی پول در میان است، بعید است که شرکت‌های خصوصی فین‌تک آخرین نتایج تحقیقات خود را منتشر کنند. دوم این‌که، باید بپذیریم که مقایسه منصفانه بین استراتژی‌های معاملاتی، به دلیل فقدان چارچوبی مشترک و تثبیت شده برای ارزیابی صحیح عملکرد آن‌ها، کاری دشوار است. به همین دلیل، هر یک از محققان، چارچوبی کلی برای مقایسات خود با انحرافات آشکار تعریف کرده‌اند. مشکل مهم دیگر مربوط به هزینه‌های معاملات است که تعاریف متفاوتی برای آن ارائه شده‌اند و یا حتی در مواردی تعریف نشده‌اند. قبل از هر چیز لازم است اشاره کنیم بیش‌تر مطالعات روی معاملات الگوریتمی، تکنیک‌هایی هستند که توسط ریاضیدانان، اقتصاددانان و معامله‌گرانی که از هوش مصنوعی استفاده نمی‌کنند، ارائه شده‌اند، مانند استراتژی‌های رایج معاملات کلاسیک شامل استراتژی‌های بازگشت به میانگین و

استراتژی معامله در جهت روند هستند که در چان (۲۰۰۹)، چان (۲۰۱۳) و نارنگ (۲۰۰۹) به تفصیل توضیح داده شده‌اند. بیش‌تر مطالعاتی که از تکنیک‌های یادگیری ماشین ML در زمینه معاملات الگوریتمی استفاده می‌کنند بر پیش‌بینی متمرکز هستند. اگر از قبل نسبت به تکامل بازار مالی اطمینان معقولی داشته باشیم، می‌توان به راحتی تصمیمات معاملاتی را بهینه کرد. مطالعات قبلی با استفاده از تکنیک‌های یادگیری عمیق به نتایج خوبی رسیده‌اند، ازجمله، آروالو (۲۰۱۶) که یک استراتژی معاملاتی مبتنی بر DNN [۳] معرفی کردند، و به‌ویژه بائو و همکاران (۲۰۱۷) که از تبدیل موجک، رمزگذارهای خودکار پشته‌ای و حافظه کوتاه‌مدت_بلندمدت $LSTM$ [۴] استفاده کرده‌اند. محققان بسیاری نیز از تکنیک‌های RL برای حل مسئله معاملات الگوریتمی استفاده کرده‌اند. برای مثال، مودی و سافل (۲۰۰۱) یک الگوریتم یادگیری عمیق بازگشتی برای کشف سیاست‌های سرمایه‌گذاری جدید بدون نیاز به ساخت مدل‌های پیش‌بینی معرفی کردند، دمپستر و لیمانز (۲۰۰۶) از RL تطبیقی برای معامله در بازارهای ارز استفاده کردند و ...

[3] Deep neural network
[4] Long short-term memory

فصل ۲

بیان رسمی مشکل معاملات الگوریتمی

در این بخش مسئله معاملات الگوریتمی تصمیم‌گیری متوالی مورد مطالعه به تفصیل ارائه و بررسی شده‌است. علاوه بر این، رسمی سازی دقیق این موضوع خاص نیز مطرح شده است.

۱.۲ معاملات الگوریتمی

معاملات الگوریتمی، که به آن معاملات کمّی نیز می‌گویند را می‌توان زیرشاخه‌ای از امور مالی و رویکردی برای تصمیم‌گیری خودکار در معاملات بر اساس مجموعه‌ای از قوانین ریاضی محاسبه شده توسط یک ماشین، در نظر گرفت. اگرچه تعاریف دیگری نیز در این زمینه وجود دارند، ولی از همین تعریف رایج پذیرفته شده در این پروژه بهره گرفته شده‌است. درواقع در برخی از متون بین تصمیمات معاملاتی (معاملات کمّی) و اجرای واقعی معاملات (معاملات الگوریتمی) تمایز قائل‌اند. به‌منظور قابل تعمیم بودن نتایج در پروژه‌ی حاضر، معاملات الگوریتمی و معاملات کمّی در تعریف کل فرآیند معاملات خودکار، مترادف در نظر گرفته شده‌اند. سودمندی تجارت الگوریتمی در بازارها کاملا ثابت شده‌است و مزیت اصلی آن بهبود قابل توجه نقدینگی است. استفاده از استراتژی‌های معاملاتی الگوریتمی در بازارهای زیادی مناسب است. ظهور اخیر ارزهای دیجیتال، مانند بیت کوین، امکانات جالب جدیدی را فراهم کرده است. پیش‌بینی می‌شود الگوریتم‌های توسعه یافته که در ادامه بررسی خواهیم کرد در بازارهای متعددی قابل اجرا باشد با این حال، تمرکز بیش‌تر الگوریتم‌های معرفی شده بر روی بازار سهام است اگر چه اشاره‌ی کوچکی به سایر بازارها نیز می‌کنیم.

در واقع یک فعالیت تجاری را می‌توان به صورت مدیریت یک سبد سهام در نظر گرفت، که این سبد، مجموعه‌ای از دارایی‌ها شامل سهام متنوع، اوراق قرضه، کالاها، ارزها و ... است.

در این پروژه سبد سهام در نظر گرفته شده شامل یک سهام واحد همراه با وجه نقد نماینده است. ارزش سهام v_t که تشکیل شده از ارزش نقدی نماینده‌ی تجاری v_t^c و ارزش سهام مشترک v_t^s که

به طور مداوم در طول زمان t تغییر می‌کند. همچنین لازم به ذکر است عملیات خرید و فروش به صورت نقدی و مبادلات سهام است. عامل معاملاتی از طریق دفتر سفارش، که شامل کل مجموعه سفارشات خرید (پیشنهادها) و سفارشات فروش (درخواست‌ها) است، با بازار سهام تعامل می‌کند. نمونه‌ای از دفتر سفارش ساده در جدول ۱.۲ نشان داده شده‌است. هر سفارش نشان دهنده تمایل شرکت‌کننده در بازار برای معامله است و از قیمت p و مقدار سفارش q و نوع معامله‌ی s (اعم از پیشنهاد خرید یا درخواست فروش) تشکیل شده‌است. برای انجام معامله، لازم است یک پیشنهاد خرید و درخواست فروش با یکدیگر جفت شوند و این رویداد فقط زمانی رخ می‌دهد که $p_{\max}^{\text{bid}} \geq p_{\min}^{ask}$ که در آن $p_{\max}^{\text{bid}}$ $\left(p_{min}^{ask}\right)$ به ترتیب حداکثر (حداقل) قیمت یک سفارش خرید (فروش) است. سپس عامل معامله‌گر کار سختی برای تولید معامله‌ای همراه با سود دارد: این‌که چه چیزی را، چه زمانی، چگونه، با چه قیمتی و چه مقدار معامله کند. این مسئله تصمیم‌گیری متوالی پیچیده معاملات الگوریتمی موضوعی است که در این پروژه بررسی خواهد شد.

جدول ۱.۲: نمونه ای از یک دفتر سفارش ساده

p Price	q Quantity	s Side
۱۰۷	۳۰۰۰	Ask
۱۰۶	۱۵۰۰	Ask
۱۰۵	۵۰۰	Ask
۹۵	۱۰۰۰	Bid
۹۴	۲۰۰۰	Bid
۹۳	۴۰۰۰	Bid

۱.۱.۲ گسسته‌سازی محور زمان

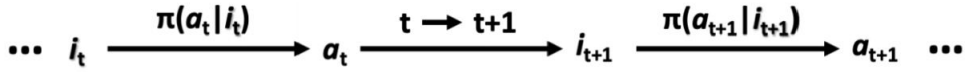
به دلیل آن‌که تصمیمات معاملاتی را می‌توان در هر زمانی صادر کرد، پس فعالیت معاملاتی فرآیندی پیوسته است. برای مطالعه مسئله معاملات الگوریتمی شرح داده شده در این پروژه، محور زمان معاملات به فرم گسسته درآمده‌است. جدول زمانی معاملات به تعداد بالایی از مراحل زمانی معامله‌های گسسته با گام‌های t همراه با طول ثابت Δt تقسیم می‌شود. در این پروژه برای شفافیت بیش‌تر، عملیات افزایش (کاهش) $t + 1(t - 1)$ برای مدل‌سازی انتقال گسسته از t به $t + \Delta t(t - \Delta t)$ استفاده می‌شود.
مدت زمان Δt ارتباط نزدیکی با بسامد معاملاتی موردنظر عامل معاملات دارد (بازه‌های معاملاتی خیلی کوتاه، بازه‌های زمانی کم‌تر از ۲۴ ساعت، روزانه، ماهانه و غیره) این عملیات گسسته‌سازی ناگریز محدودیتی را با توجه به این بسامد معاملاتی تحمیل می‌کند. در واقع، چون بازه‌ی زمانی Δt بین دو گام زمانی به علت محدودیت‌های فنی نمی‌تواند به هر میزان دلخواه کوچکی انتخاب شود، حداکثر بسامد معاملاتی در دسترس را برابر با $\frac{1}{\Delta t}$ در نظر می‌گیرند. در این پروژه، محدودیت

گفته شده با توجه به این این‌که بسامد معاملاتی مورد نظر روزانه است، برآورده می‌شود، یعنی عامل معاملاتی هر روز فقط یک‌بار تصمیم جدید می‌گیرد.

۲.۱.۲ راهبرد معاملاتی

رویکرد معاملات الگوریتمی مبتنی بر قانون است، به این معنا که تصمیمات معاملاتی بر اساس مجموعه‌ای از قوانین یا یک ترفند معاملاتی اتخاذ می‌شوند. از نظر فنی، یک ترفند معاملاتی را می‌توان به عنوان یک سیاست برنامه‌ریزی شده $\pi\left(a_t \mid i_t\right)$ در نظر گرفت. همچنین به صورت قطعی یا تصادفی با توجه به اطلاعات موجود از عامل معاملاتی درگام زمانی t اقدام به انجام معامله‌ای می‌کند. به‌علاوه، همان‌طور که در شکل ۱.۲ می‌بینید، مشخصه کلیدی استراتژی معاملاتی جنبه ترتیبی آن است. هر عامل استراتژی معاملاتی خود را به‌صورت متوالی و طی مراحل زیر اجرا می‌کند:

- مرحله اول: به روز رسانی اطلاعات بازار موجود i_t
- مرحله دوم: اجرای سیاست $\pi\left(a_t \mid i_t\right)$ برای انجام عمل a_t
- مرحله سوم: اجرای اقدام معاملاتی تعیین شده a_t
- مرحله چهارم. به گام زمانی t یک واحد اضافه کنید: $t \rightarrow t+1$، سپس به مرحله‌ی اول برگردید.



شکل ۱.۲: تصویری از اجرای استراتژی معاملاتی

در بخش بعدی، مسئله تصمیم‌گیری متوالی معاملات الگوریتمی، که شباهت‌هایی با سایر موضوعاتی دارد که با موفقیت توسط جامعه‌ی یادگیری تقویتی حل شده‌است، به عنوان یک مسئله یادگیری تقویتی مطرح شده‌است.

۳.۱.۲ بیان رسمی مسئله یادگیری تقویتی

همانطور که در شکل۲.۲ نشان داده شده‌است، مسئله یادگیری تقویتی مربوط به تعامل متوالی یک عامل با محیط‌اش است. در هر گام زمانی t عامل یادگیری تقویتی ابتدا محیط RL [1] محیط داخلی

[1]Reinforcement Learning

s_t را بررسی می‌کند، و یک مشاهده‌ی o_t را برمی‌گرداند. و پس از آن بنا بر سیاست $\pi(a_t \mid h_t)$ یادگیری تقویتی، عمل a_t را انجام می‌دهد. در این رابطه‌ی h_t پیشینه‌ی عامل یادگیری تقویتی، پاداشی به اندازه‌ی r_t در ازای عمل انجام شده دریافت می‌کند. پیشینه‌ی عامل در این یادگیری تقویتی را می‌توان به این صورت نوشت: $h_t = \{(o_\tau, a_\tau, r_\tau) \mid \tau = 0, 1, \ldots, t\}$.

تکنیک‌های یادگیری تقویتی به طراحی خط‌مشی‌های π برای به حداکثر رساندن یک معیار بهینگی هستند. این معیار مستقیماً به پاداش‌های فوری r_t مشاهده شده در یک افق زمانی مشخص بستگی دارد. از نظر ریاضی، سیاست بهینه π^* به صورت زیر بیان می‌شود:

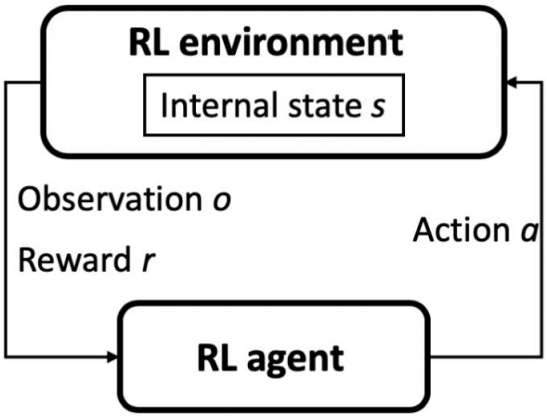
$$\pi^* = \underset{\pi}{\operatorname{argmax}} \mathbb{E}[R \mid \pi] \tag{۱.۲}$$

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \tag{۲.۲}$$

که در آن پارامتر γ ضریب تنزیل است که در بازه‌ی $\gamma \in [0, 1]$ قرار دارد. همچنین γ اهمیت پاداش‌های آینده را تعیین می‌کند. برای مثال اگر $\gamma = 0$، گفته می‌شود عامل یادگیری نزدیک‌بین است، زیرا فقط پاداش‌های فعلی را در نظر می‌گیرد و به طور کامل پاداش‌های آینده را کنار می‌گذارد. وقتی γ افزایش می‌یابد، عامل یادگیری تقویتی گرایش بیش‌تری به بازه‌ی طولانی مدت دارد. در حالت شدید وقتی $\gamma = 1$ است، عامل یادگیری هر نوع پاداشی را به طور مساوی در نظر می‌گیرد. واضح است که این پارامتر کلیدی باید با توجه به رفتار مورد نظر تنظیم شود.

۲.۲ مشاهدات یادگیری تقویتی

محیط یادگیری تقویتی در مسئله معاملات الگوریتمی، کل دنیای معاملاتی پیچیده حول عامل یادگیری تقویتی است. درواقع، این محیط معاملاتی را می‌توان محیطی انتزاعی شامل مکانیسم‌های معاملاتی همراه با تک‌تک اطلاعاتی که می‌تواند بر فعالیت معاملاتی عامل تأثیر بگذارد، در نظر گرفت. چالش اصلی مسئله معاملات الگوریتمی مشاهده‌پذیری بسیار ضعیف این محیط است. درواقع، اطلاعات زیادی، نظیر اطلاعات محرمانه برخی شرکت‌ها و استراتژی‌های سایر شرکت‌کنندگان در بازار، به سادگی از دید عامل معاملاتی پنهان می‌شوند. در واقع اطلاعات در دسترس عامل یادگیری تقویتی در مقایسه با پیچیدگی محیط بسیار محدود است. به علاوه این اطلاعات می‌تواند شامل اشکال مختلفی اعم از کمّی و کیفی داشته باشد. پردازش صحیح چنین اطلاعاتی و بیان مجدد آنها با استفاده از ارقام کمّی مربوطه، ضمن به حداقل رساندن جهت‌گیری‌های ذهنی، حیاتی است. در نهایت، باید بر پیچیدگی قابل توجه همبستگی زمانی نیز فائق آمد. بنابراین،



شکل ۲.۲: بلوک‌های اصلی ساختمان یادگیری تقویتی

اطلاعات بازیابی شده توسط عامل یادگیری تقویتی در هر گام زمانی باید به طور متوالی، به صورت دنباله‌ای از اطلاعات (و نه به صورت جداگانه) در نظر گرفته شود.
در هر مرحله از زمان معامله‌ی t عامل یادگیری تقویتی وضعیت داخلی بازار سهام $s_t \in \mathcal{S}$ را مشاهده می‌کند. این اطلاعات محدودی که توسط عامل یادگیری در این محیط معاملاتی جمع‌آوری شده است، با $o_t \in \mathcal{O}$ نشان داده می‌شود. در حالت ایده‌ال، فضای مشاهده‌ی $\mathcal{O}$ باید شامل تمام اطلاعاتی باشد که می‌تواند بر قیمت‌های بازار تاثیر بگذارد. به دلیل جنبه ترتیبی مسئله معاملات الگوریتمی، یک مشاهده‌ی o_t باید به عنوان دنباله‌ای از هر دو اطلاعات جمع‌آوری شده در مراحل قبلی τ (تاریخچه) و اطلاعات جدید در دسترس در مرحله‌ی زمانی t در نظر گرفته شود. در این پروژه، مشاهدات عامل یادگیری تقویتی می‌تواند به صورت ریاضی زیر بیان شوند:

$$o_t = \{S(t'), D(t'), T(t'), I(t'), M(t'), N(t'), E(t')\}_{t'=t-\tau}^{t} \qquad (۳.۲)$$

که در این معادله:

- $S(t)$ اطلاعات وضعیت عامل یادگیری در گام زمانی t (موقعیت معاملات فعلی، تعداد سهام متعلق به عامل یادگیری و وجه نقد موجود)
- $D(t)$ اطلاعاتی است که توسط عامل یادگیری در گام زمانی t مربوطه به داده‌های $OHLCV$ [۲] که بازار سهام مشخص می‌کند، جمع‌آوری می‌شود. $D(t)$ را می‌توان به صورت زیر بیان کرد:

$$D(t) = \{p_t^O, p_t^H, p_t^L, p_t^C, V_t\} \qquad (۴.۲)$$

[2] Open-High-Low-Close-Volume

که در معادله‌ی ۴ داریم:

- p_t^O قیمت باز شدن بازار سهام در بازه زمانی $[t-\Delta t, t]$ است.
- p_t^H بالاترین قیمت بازار سهام در بازه زمانی $[t-\Delta t, t]$ است.
- p_t^L کم‌ترین قیمت بازار سهام در بازه‌ی زمانی $[t-\Delta t, t]$ است.
- p_t^C قیمت بسته شدن بازار سهام در بازه زمانی $[t-\Delta t, t]$ است.
- همچنین V_t حجم کل سهام مبادله شده در بازه‌ی زمانی $[t-\Delta t, t]$ است.

همچنین داریم:

- $T(t)$ اطلاعات عامل یادگیری در مورد معامله در زمان t است. (تاریخ، روز هفته، زمان)
- $I(t)$ اطلاعات عامل درباره چند شاخص فنی مربوط به بازار سهام در گام زمانی t است. شاخص‌های فنی زیادی وجود دارد که درک‌مان را از پدیده‌های مالی متنوع مانند میانگین متحرک همگرایی - واگرایی [۳] $(MACD)$ شاخص قدرت نسبی [۴] (RSI) و یا شاخص میانگین جهت‌دار [۵] (ADX) و ...
- $M(t)$ اطلاعات کلان اقتصادی در اختیار عامل یادگیری را در مرحله‌ی زمانی t جمع‌آوری می‌کند. بسیاری از شاخص‌های اقتصادی کلان وجود دارند که می‌توانند به صورت بالقوه برای پیش‌بینی بازارها مفید باشند، مانند نرخ بهره یا نرخ ارز.
- $N(t)$ اطلاعات خبری جمع‌آوری شده توسط توسط عامل یادگیری در زمان t را نشان می‌دهد. این داده‌های خبری را می‌توان از منابع مختلفی مانند رسانه‌های اجتماعی (توییتر، فیس‌بوک، لینکدین)، روزنامه‌ها، مجلات خاص و ... استخراج کرد. سپس می‌توان از مدل‌های تحلیل پیچیده برای استخراج ارقام کمی معنادار استفاده کرد.
- $E(t)$ هر گونه اطلاعات مفید اضافی که در مرحله‌ی زمانی t در اختیار عامل معاملاتی است. برای مثال: استراتژی معاملاتی شرکت‌کنندگان در بازار، اطلاعات محرمانه شرکت‌ها، رفتارهای مشابه در بازار سهام، توصیه‌های کارشناسان و ...

[3] Moving Average Convergence-Divergence
[4] Relative Strength Index
[5] Average Directional Index

کاهش فضای مشاهده: یکی از مفروضات در این پروژه، این است که عامل یادگیری فقط اطلاعات مربوط به داده‌های کلاسیک $OHLCV$ یعنی $D(t)$ و اطلاعات وضعیت حالت یعنی $S(t)$ را در نظر می‌گیرد. با این مفروضات، فضای مشاهده کاهش یافته یادگیری تقویتی، o_t را می‌توان به صورت زیر نوشت:

$$o_t = \left\{ \{p_{t'}^O, p_{t'}^H, p_{t'}^L, p_{t'}^C, V_{t'}\}_{t'=t-\tau}^{t}, P_t \right\} \qquad (۵.۲)$$

که در آن P_t وضعیت معاملاتی عامل یادگیری تقویتی در گام زمانی t (صرف‌نظر از بلند یا کوتاه بودن بازه معاملاتی) است.

۳.۲ اقدامات یادگیری تقویتی

عامل یادگیری تقویتی در هر مرحله‌ی زمانی t، یک عمل معاملاتی $a_t \in \mathcal{A}$ حاصل از سیاست $\pi(a_t \mid h_t)$ را اجرا می‌کند. در واقع، عامل معاملات باید به چندین سوالی پاسخ دهد: چگونه و چقدر معامله کند؟ چنین تصمیماتی را می‌توان با مقدار سهام خریداری شده توسط عامل معاملاتی در زمان t که با $Q_t \in \mathbb{Z}$ نشان داده می‌شود، مدل کرد. بنابراین اقدامات یادگیری تقویتی را می‌توان به صورت زیر بیان کرد:

$$a_t = Q_t \qquad (۶.۲)$$

با توجه به مقدار Q_t یکی از سه حالت زیر می‌تواند رخ بدهد:

- $Q_t > 0$: عامل یادگیری در بازار سهام، با ارائه پیشنهاد خرید در دفتر سفارش، سهام خریداری می‌کند.
- $Q_t < 0$: عامل یادگیری با ارسال درخواست فروش جدید در دفتر سفارش، در بازار سهام می‌فروشد.
- $Q_t = 0$: نه سهمی خریداری می‌شود و نه سهمی فروخته می‌شود.

در واقع اقدامات واقعی در محدوده یک فعالیت معاملاتی، سفارش‌هایی هستند که در دفترچه سفارش‌ها ثبت می‌شوند. فرض بر این است که عامل یادگیری با ماژولی خارجی که مسئول ترکیب این اقدامات واقعی با توجه به مقدار Q_t است، ارتباط برقرار می‌کند. این ماژول، سیستم اجرای معاملات است. لازم به ذکر است که استراتژی‌های اجرایی متعددی را می‌توان با توجه به هدف کلی معاملات در نظر گرفت.
اقدامات معاملاتی بر هر دو مؤلفه ارزش‌های نقدی و سهام، تأثیر می‌گذارند. با این فرض که معامله

در قیمتی نزدیک به قیمت بسته شدن بازار $p_t \simeq p_t^C$ انجام می‌شود. این دو مؤلفه با معادلات زیر به‌روزرسانی می‌شود:

$$v_{t+1}^c = v_t^c - Q_t p_t \tag{۷.۲}$$

$$v_{t+1}^s = \underbrace{(n_t + Q_t)}_{n_{t+1}} p_{t+1} \tag{۸.۲}$$

که در آن، $n_t \in \mathbb{Z}$ تعداد سهام متعلق به عامل معاملاتی در گام زمانی t است. در چارچوب این پروژه، مقدار n_t می‌تواند مقادیر منفی نیز بگیرد. باید توجه کنیم منظور از مقدار منفی سهام صرفاً استقراض و فروش سهام و در عین حال، تعهد به بازپرداخت سهام به تأمین‌کننده سهام در آینده است. چنین فرآیندی جالب نیز هست، زیرا امکانات جدیدی را برای عامل معاملاتی در نظر می‌گیرد.

دو محدودیت مهم برای مقدار سهام معامله شده Q_t در نظر گرفته شده است:

- مورد اول: بر خلاف ارزش سهام v_t^s که می‌تواند مثبت یا منفی باشد، ارزش نقدی v_t^c باید در هر گام زمانی t باید مثبت باشد. درواقع این محدودیت کران بالایی برای تعداد سهامی که عامل معاملاتی قادر به خریدش است، در نظر می‌گیرد. این کران بالا به‌راحتی از معادله ۷.۲ به دست می‌آید.
- مورد دوم: اگر عامل معاملاتی متحمل ضررهای قابل‌توجهی شود، ریسکی مرتبط با ناتوانی بازپرداخت به وام‌دهنده سهام نیز وجود خواهد داشت.

برای جلوگیری از وقوع چنین وضعیتی، در مواردی که تعداد سهام در دسترس منفی است، برای اینکه عامل بتواند به وضعیت خنثی بازگردد (یعنی $n_t = 0$)، باید به‌اندازه کافی بزرگ در نظر گرفته شود. حداکثر تغییر نسبی در قیمت‌ها، که برحسب درصد ($\%$) بیان و با $\epsilon \in \mathbb{R}^+$ نشان داده می‌شود، قبل از شروع فعالیت معاملاتی توسط عامل یادگیری تقویتی تعیین می‌شود. این پارامتر مربوط به حداکثر تکامل روزانه بازار توسط عامل در کل افق معاملاتی است. به عبارت دیگر، مادامی که تغییرات بازار کمتر از مقدار این پارامتر باشد، عامل معاملاتی همیشه می‌تواند بدهی خود به وام‌دهنده سهم بپردازد. بنابراین، محدودیت‌هایی اعمال شده بر اقدامات یادگیری تقویتی در مرحله زمانی t را می‌توان به‌صورت ریاضیاتی زیر بیان کرد:

$$v_{t+1}^c \geq 0 \tag{۹.۲}$$

$$v_{t+1}^c \geq -n_{t+1} p_t (1 + \epsilon) \tag{۱۰.۲}$$

با این فرض که شرط زیر برآورده شود:

$$\left|\frac{p_{t+1}-p_t}{p_t}\right| \leq \epsilon \qquad (۱۱.۲)$$

ملاحظات هزینه‌های معاملاتی:
در ابتدا باید توجه کنیم مدل ارائه شده توسط معادله ۷.۲ نادرست است و منجر به نتایجی غیرواقعی خواهد شد. درواقع، نباید هزینه‌های معاملاتی را در شبیه‌سازی فعالیت‌های معاملاتی نادیده گرفت. این نادیده‌انگاری عموماً گمراه‌کننده است، زیرا یک استراتژی معاملاتی که سودآوری زیادی در مدل شبیه‌سازی دارد، به‌ویژه وقتی بسامد معاملات بالا است، ممکن است در معاملات واقعی، به دلیل این‌که هزینه‌های معاملاتی نادیده گرفته شده، ضررهای زیادی را موجب شود. هزینه‌های معاملاتی را می‌توان به دو دسته تقسیم کرد:

- هزینه‌ی اول: هزینه‌های آشکاری که ناشی از هزینه‌های مبادله و مالیات است.
- هزینه‌ی دوم: هزینه‌های ضمنی یا هزینه‌های افت قیمت که از سه عنصر اصلی تشکیل شده و مرتبط با برخی پویایی‌های محیط معاملاتی است.

هزینه‌های مختلف افت قیمت در ادامه شرح داده می‌شود:

- هزینه‌های پخش: این هزینه‌ها ناشی از تفاوت بین حداقل قیمت پیشنهادی فروش p_{min}^{ask} و حداکثر قیمت پیشنهادی خرید $p_{\max}^{\text{bid}}$ است که هزینه‌های پخش [۶] نامیده می‌شود. چون پردازش دقیق وضعیت کامل دفتر سفارش معمولاً بسیار پیچیده است یا حتی این دفتر در دسترس نیست، تصمیمات معاملاتی عمدتاً بر اساس این قیمت متوسط محاسبه می‌شوند: $p^{\text{mid}} = \left(p_{\max}^{\text{bid}} + p_{\min}^{ask}\right)/2$ با این حال، معامله خرید (فروش) پیشنهاد شده در قیمت p^{mid} به ناچار در قیمت p با قید زیر انجام می‌شود:

$$.p \geq p_{\min}^{ask} \left(p \leq p_{\max}^{\text{bid}}\right)$$

این‌گونه هزینه‌ها به‌ویژه وقتی نقدینگی بازار سهام در مقایسه با حجم سهام معامله‌شده پایین باشد، قابل‌توجه‌تر هستند.

- هزینه‌های تاثیر بازار: این هزینه‌ها ناشی از تاثیر اقدامات معامله‌گر بر بازار است. هر معامله‌ای (اعم از خرید و فروش) تاثیر بالقوه‌ای بر قیمت دارد. زمانی که نقدینگی بازار سهام نسبت به حجم معاملات سهام کمتر است، اهمیت این پدیده بیشتر نیز می‌شود.

[6] Spread costs

- هزینه‌های زمان‌بندی: با توجه به تغییر پیوسته قیمت بازار، این هزینه‌ها مربوط به زمان موردنیاز برای انجام فیزیکی معامله پس از قطعی شدن تصمیم انجام معامله است. اولین علت این هزینه‌ها، تاخیر اجتناب‌ناپذیر ناشی از ثبت سفارش‌ها در دفتر سفارشات بازار است. دومین علت، تأخیرهای عمدی سیستم اجرای معاملات است. برای مثال، معاملات بزرگ را می‌توان به چند معامله کوچک‌تر تقسیم کرد تا تأثیرش در طول زمان پخش‌شده و هزینه‌های تاثیر بازار محدودتری داشته باشد. مدل‌سازی دقیق هزینه‌های معاملاتی برای بازتولید واقعی پویایی محیط واقعی معاملات ضروری است. اگرچه ملاحظه هزینه‌های آشکار نسبتاً آسان است، ولی مدل‌سازی صحیح هزینه‌های افت قیمت کاری بسیار پیچیده‌ای است. پروژه‌ی حال حاضر برای ادغام این دو هزینه در محیط یادگیری تقویتی از یک راهکار ابتکاری استفاده می‌کند. وقتی معامله‌ای انجام می‌شود، مقدار سرمایه مشخصی معادل C درصد از کل پول سرمایه‌گذاری شده از دست می‌رود. مقدار پارامتر C در شبیه‌سازی‌ها به‌طور واقع‌گرایانه‌ای برابر با 0.1% انتخاب شده است. هزینه‌های معاملات در عمل مستقیماً از وجه نقد عامل معاملات برداشت می‌شود. با توجه به الگوریتم مبتکرانه‌ای که قبلا شرح داده شد، برای مدل‌سازی هزینه‌های معاملاتی، می‌توان معادله ۷.۲ را با اضافه کردن جمله‌ای اصلاحی به‌صورت زیر بازنویسی کرد:

$$v_{t+1}^c = v_t^c - Q_t p_t - \underbrace{C\,|Q_t|\,p_t}_{\text{costs Trading}} \qquad (۱۲.۲)$$

بعلاوه، هزینه‌های معاملاتی باید به‌درستی در نامساوی ۱۰.۲ ملاحظه شوند. در واقع، ضمن ملاحظه هزینه‌های معاملاتی، مقدار نقدینگی v_t^c باید آن‌قدر بزرگ باشد که در صورت رخداد حداکثر تغییرات بازار، بتوان به وضعیت خنثی بازگشت $(n_t = 0)$. در نتیجه، معادله ۱۰.۲ به صورت زیر بازنویسی می‌شود:

$$v_{t+1}^c \geq -n_{t+1} p_t (1+\epsilon)(1+C) \qquad (۱۳.۲)$$

در انتها، فضای عمل یادگیری تقویتی یعنی $\mathcal{A}$ را می‌توان به صورت مجموعه‌ای گسسته از مقادیر قابل قبول برای مقدار سهام معامله شده یعنی Q_t تعریف کرد. همچنین لازم به ذکر است فضای A را به فرم ریاضی می‌توان به صورت زیر شرح داد:

$$\mathcal{A} = \left\{ Q_t \in \mathbb{Z} \cap \left[\underline{Q_t}, \overline{Q_t}\right] \right\} \qquad (۱۴.۲)$$

که در آن داریم:

$$\overline{Q_t} = \frac{v_t^c}{p_t(1+C)}$$

$$\underline{Q_t} = \begin{cases} \frac{\Delta_t}{p_t\epsilon(1+C)} & \text{if } \Delta_t \geq 0 \\ \frac{\Delta_t}{p_t(2C+\epsilon(1+C))} & \text{if } \Delta_t < 0 \end{cases}$$
با: $\Delta_t = -v_t^c - n_t p_t(1+\epsilon)(1+C).$

کاهش فضای عمل

در چارچوب موضوعاتی که تا به الان بررسی شده، فضای عمل $\mathcal{A}$ به‌منظور کاهش پیچیدگی مسئله معاملات الگوریتمی، کاهش می‌یابد. فضای عمل کاهش یافته تنها از دو عمل یادگیری تقویتی تشکیل شده است و می‌تواند به صورت ریاضیاتی زیر بیان شود:

$$a_t = Q_t \in \left\{ Q_t^{\text{Long}}, Q_t^{\text{Short}} \right\} \qquad (۱۵.۲)$$

اولین عمل یادگیری یعنی Q_t^{Long} با تبدیل هرچه بیشتر نقدینگی v_t^c به ارزش سهام v_t^s تعداد سهام متعلق به عامل معاملاتی را بیشینه می‌کند، که به‌صورت ریاضیاتی زیر بیان می‌شود:

$$Q_t^{\text{Long}} = \begin{cases} \left\lfloor \frac{v_t^c}{p_t(1+C)} \right\rfloor & \text{if } a_{t-1} \neq Q_{t-1}^{\text{Long}}, \\ 0 & \text{otherwise.} \end{cases} \qquad (۱۶.۲)$$

عمل Q_t^{Long} همیشه معتبر است، زیرا به وضوح در فضای عمل اصلی $\mathcal{A}$ که با معادله‌ی ۱۴.۲ تعریف می‌شود، گنجانده شده‌است. در نتیجه، عامل معاملاتی، مالک این تعداد سهام خواهد بود: $N_t^{\text{Long}} = n_t + Q_t^{\text{Long}}$. در مقابل، دومین اقدام یادگیری تقویتی که با Q_t^{Short} مشخص شده است، ارزش سهم v_t^s را به ارزش نقدی v_t^c تبدیل می‌کند، به طوری که تعداد سهام یادگیری تقویتی برابر با $-N_t^{\text{Long}}$ می‌شود. این عملیات را می‌توان به صورت ریاضیاتی زیر بیان کرد:

$$\widehat{Q_t^{\text{Short}}} = \begin{cases} -2n_t - \left\lfloor \frac{v_t^c}{p_t(1+C)} \right\rfloor & \text{if } a_{t-1} \neq Q_{t-1}^{Short}, \\ 0 & \text{otherwise.} \end{cases} \qquad (۱۷.۲)$$

با این حال، وقتی قیمت به طور قابل توجهی افزایش می‌یابد، عمل $\widehat{Q_t^{\text{Short}}}$ ممکن است کران پایین فضای عمل $\mathcal{A}$ یعنی $\underline{Q_t}$ را نقض کند. در نهایت، دومین عمل یادگیری تقویتی یعنی Q_t^{Short} به صورت زیر بیان می‌شود:

$$Q_t^{Short} = \max\left\{ \widehat{Q_t^{\text{Short}}}, \underline{Q_t} \right\} \qquad (۱۸.۲)$$

در انتها، لازم به ذکر است که دو عمل یادگیری تقویتی کاهش‌یافته درواقع مربوط به وضعیت معاملاتی بعدی عامل یعنی P_{t+1} هستند. در واقع اولین عمل، Q_t^{Long}، باعث می‌شود وضعیت

معاملاتی طولانی می‌شود، زیرا تعداد سهام متعلق به عامل، مثبت است. در مقابل عمل دوم، Q_t^{Short}، همیشه باعث منفی شدن تعداد سهام می‌شود، که این موضوع در امور مالی به‌طورکلی با عنوان وضعیت معاملاتی کوتاه‌مدت شناخته می‌شود.

پاداش یادگیری تقویتی: استراتژی بازده روزانه، انتخابی طبیعی برای پاداش یادگیری تقویتی در بررسی مسئله معاملات الگوریتمی است. استفاده از بازدهی‌های مثبت، که نشانه‌ای از سودآور بودن استراتژی هستند، به لحاظ شهودی منطقی است. بعلاوه، مزیت این کمیت مستقل بودن از تعداد سهامی n_t است که در حال حاضر در تملک عامل است. مزیت دیگر این انتخاب، عدم نیاز به کار کردن با سیستم پیچیده پاداش پراکنده است. پاداش‌های یادگیری تقویتی را می‌توان به‌صورت ریاضیاتی زیر بیان کرد:

$$r_t = \frac{v_{t+1} - v_t}{v_t} \qquad (۱۹.۲)$$

۴.۲ بررسی عینی و واقع‌گرایانه

ارزیابی عینی عملکرد استراتژی معاملاتی، با توجه به تعدد عوامل کمّی و کیفی که باید در نظر گرفت، کاری دشوار است. درواقع، انتظار نداریم یک استراتژی معاملاتی با عملکرد خوب حتماً سود ایجاد کند، بلکه انتظار داریم ریسک مرتبط با فعالیت معاملاتی را به‌طور مؤثر کاهش دهد. توازن بین این دو هدف به مشخصات عامل معاملاتی و میزان ریسک‌پذیری آن بستگی دارد. به لحاظ شهودی واضح است که بیشینه کردن سود حاصل از یک استراتژی معاملاتی، لازم ولی ناکافی است. بلکه، هدف اصلی استراتژی معاملاتی، حداکثر کردن نسبت شارپ است. این نسبت، شاخصی عملکردی است که کاربرد گسترده‌ای در زمینه‌های مالی و معاملات الگوریتمی دارد. این شاخص هم سود تولیدشده و هم ریسک مرتبط با فعالیت معاملاتی را در نظر می‌گیرد و به همین دلیل، محصول برای ارزیابی عملکرد مناسب است. بیان ریاضیاتی نسبت شارپ S_r به‌صورت زیر است:

$$S_r = \frac{\mathbb{E}\left[R_s - R_f\right]}{\sigma_r} = \frac{\mathbb{E}\left[R_s - R_f\right]}{\sqrt{\mathrm{var}\left[R_s - R_f\right]}} \simeq \frac{\mathbb{E}\left[R_s\right]}{\sqrt{\mathrm{var}\left[R_s\right]}} \qquad (۲۰.۲)$$

که در این معادله داریم:

- R_s بازدهی استراتژی معاملاتی در یک دوره زمانی معین است که سودآوری آن را مدل‌سازی می‌کند.
- R_f بازده بدون ریسک یا بازده مورد انتظار از یک سرمایه‌گذاری کاملا مطمئن است. (مقدار ریسک ناچیز است)

- σ_r انحراف استاندارد بازده مازاد استراتژی معاملاتی $R_s - R_f$ است، که ریسک‌پذیری آن را مدل‌سازی می‌کند.

در عمل برای محاسبه نسبت شارپ S_r ابتدا بازده روزانه حاصله از استراتژی معاملاتی با استفاده از فرمول زیر محاسبه می‌شود:

$$\rho_t = (v_t - v_{t-1}) / v_{t-1}$$

سپس نسبت بین میانگین بازده و انحراف معیار بررسی می‌شود. در نهایت، نسبت شارپ سالانه از ضرب این مقدار در جذر تعداد روزهای معاملاتی در یک سال به دست می‌آید.

همچنین در حالت ایده‌آل، استراتژی معاملاتی با عملکرد خوب باید عملکرد قابل قبولی در بازارهای مختلف با الگوهای متفاوت داشته باشد. برای مثال، استراتژی معاملاتی باید عملکرد درستی در بازارهایی با روندهای افزایشی و کاهشی قوی قیمت با سطوح نوسانات مختلف داشته باشد. بنابراین توسعه استراتژی معاملاتی جدید بر اساس تکنیک‌های یادگیری تقویتی عمیق برای بیشینه کردن میانگین نسبت شارپ محاسبه‌شده روی کل مجموعه بازارهای سهام موجود دارای اهمیت است.

هرچند هدف نهایی حداکثر کردن نسبت شارپ است، ولی الگوریتم یادگیری تقویتی عمیق استفاده شده در این پروژه در حقیقت مجموع تخفیف مورد انتظار پاداش‌ها (بازده‌های روزانه) در یک افق زمانی نامحدود را بیشینه می‌کند. این معیار بهینه‌سازی، که دقیقاً منجر به بیشینه شدن سود نمی‌شود ولی بسیار به جواب بهینه نزدیک می‌شود، را درواقع می‌توان نوعی ساده‌سازی معیار نسبت شارپ در نظر گرفت.

فصل ۳

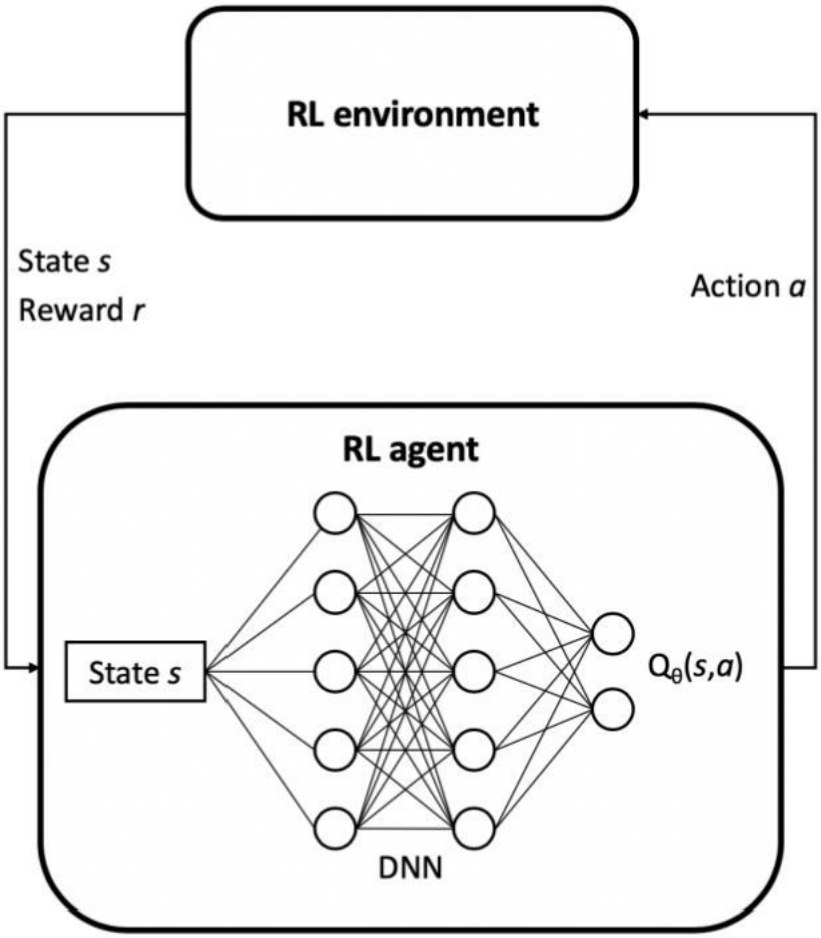
طراحی الگوریتم یادگیری تقویتی عمیق

در این بخش، یک الگوریتم یادگیری تقویتی عمیق جدید برای حل مسئله معاملات الگوریتمی، که قبلاً معرفی شده بود، طراحی می‌شود. استراتژی معاملاتی حاصله، که الگوریتم کیو- شبکه معاملاتی عمیق [۱] ($TDQN$) نامیده می‌شود، نتیجه الهام از الگوریتم موفق کیو-شبکه عمیق [۲] (DQN) که توسط امنیح و همکاران (۲۰۱۳) ارائه شده است، و تطبیق قابل توجه آن با مسئله تصمیم‌گیری مورد نظر مطالعه است. برای آموزش عامل یادگیری تقویتی، مسیرهایی مصنوعی از مجموعه محدودی از داده‌های تاریخی بازار سهام تولید می‌شوند.

۱.۰.۳ الگوریتم کیو-شبکه‌ی عمیق

الگوریتم کیو-شبکه عمیق (DQN) یک الگوریتم یادگیری تقویتی عمیق قادر به یادگیری موفقیت‌آمیز سیاست‌های کنترلی از روی ورودی‌های حسی با ابعاد بالا است. این الگوریتم به‌نوعی جایگزین الگوریتم محبوب کیو-یادگیری ارائه شده در واتکینز و دایان (۱۹۹۲) است. گفته می‌شود این الگوریتم یادگیری تقویتی عمیق مستقل از مدل است، یعنی الزامی به وجود مدلی کامل از محیط وجود ندارد و وجود مسیرها کافی است. این الگوریتم متعلق به خانواده الگوریتم‌های کیو-یادگیری و مبتنی بر یادگیری تقریبی تابع مقدار وضعیت-عمل است که با DNN [۳] نشان داده می‌شود. در چنین زمینه‌ای، یادگیری کیو-تابع معادل یادگیری پارامترهای θ این شبکه‌ی عصبی عمیق است. در نهایت گفته می‌شود که این الگوریتم کیو-شبکه‌ی عمیق مستقل از سیاست است، زیرا با استفاده از تجربیات قبلی $e_t = (s_t, a_t, r_t, s_{t+1})$ جمع‌آوری‌شده در هر نقطه از آموزش استفاده می‌کند.

[1] Trading Deep Q-Network algorithm
[2] Deep Q-Network algorithm
[3] Deep Neural Network



شکل ۱.۳: نمایش الگوریتم DQN

الگوریتم DQN در شکل بالا به طور خلاصه نشان داده شده و توضیحات مفصلی دیگری درباره آن ارائه نشده اما مطالعات زیاد دیگری درباره این الگوریتم وجود دارد، ازجمله: وان هاسلت و همکاران (۲۰۱۵)، وانگ و همکاران (۲۰۱۵)، شاول و همکاران (۲۰۱۶)، بلمار و همکاران (۲۰۱۷)، فورتوناتو و همکاران (۲۰۱۸) و هسل و همکاران (۲۰۱۷).

۲.۰.۳ تولید مسیرهای مصنوعی

مدل کاملی برای محیط $\mathcal{E}$ در چارچوب مسئله معاملات الگوریتمی وجود ندارد. آموزش الگوریتم کیو- شبکه معاملاتی عمیق کاملاً مبتنی بر تولید مسیرهای مصنوعی از مجموعه محدودی از داده‌های روزانه OHLCV [۴] تاریخی بازار سهام است. یک مسیر τ دنباله‌ای از مشاهدات $o_t \in \mathcal{O}$، اعمال $a_t \in \mathcal{A}$ و پاداش r_t مربوط به یک عامل یادگیری تقویتی برای تعداد معینی T گام زمانی معاملاتی t است.

$$\tau = (\{o_0, a_0, r_0\}, \{o_1, a_1, r_1\}, \ldots, \{o_T, a_T, r_T\})$$

اگر چه محیط $\mathcal{E}$ در ابتدا ناشناخته است، ولی یکی از مسیرهای واقعی منطبق با رفتار تاریخی بازار سهام، یعنی حالت خاص غیرفعال بودن عامل یادگیری تقویتی، وجود دارد. این مسیر اصلی متشکل از قیمت‌ها و حجم‌های تاریخی همراه با اعمال طولانی اجرا شده توسط عامل یادگیری تقویتی، بدون اینکه پولی در اختیارش باشد، است و در حقیقت نشان می‌دهد که هیچ سهمی معامله نشده است.
برای شبیه‌سازی تعاملات با محیط $\mathcal{E}$ ، مسیرهای فعال جدیدی برای مسئله معاملات الگوریتمی به طور مصنوعی و بر اساس مسیر واقعی منحصربه‌فرد فوق ایجاد می‌شوند. رفتار تاریخی بازار سهام به بیان ساده تحت تأثیر اعمال جدید عامل معاملاتی نیست. مسیرهای مصنوعی تولید شده به بیان ساده متشکل از توالی مشاهدات واقعی تاریخی مرتبط با توالی‌های مختلف اعمال معاملاتی عامل یادگیری تقویتی است. برای اینکه چنین اقدامی به لحاظ علمی قابل قبول باشد و شبیه‌سازی‌های واقع‌بینانه‌ای را نتیجه دهد، عامل معاملاتی نباید قادر به اثرگذاری بر رفتار بازار سهام باشد. این فرض معمولاً در مواردی برقرار است که تعداد سهام معامله شده توسط عامل معاملاتی نسبت به نقدشوندگی بازار سهام کم باشد.
علاوه بر تولید مسیرهای مصنوعی توضیح داده شده در بالا، ترفندی نیز برای بهبود اندک توانایی بررسی عامل یادگیری تقویتی استفاده شده است. این ترفند مبتنی بر این واقعیت است که فضای عمل کاهش‌یافته $\mathcal{A}$ تنها از دو عمل تشکیل شده است: عمل طولانی $\left(Q_t^{\text{Long}}\right)$ و عمل کوتاه $\left(Q_t^{\text{Short}}\right)$. در هر گام زمان معاملاتی t، عمل انتخاب شده a_t در محیط معاملاتی $\mathcal{E}$ و عمل مخالف آن a_t^- دقیقا روی این محیط $\mathcal{E}^-$ اجرا می‌شود. اگر چه استفاده از این ترفند، توازن چالش‌برانگیز بین اکتشاف و استفاده را به‌طور کامل حل نمی‌کند، ولی عامل یادگیری تقویتی را به ازای اندکی هزینه‌ی محاسباتی بیش‌تر، قادر به کاوش پیوسته می‌کند.

۳.۰.۳ بهبودها و تغییرات متنوع

الگوریتم کیو-شبکه‌ی عمیق (DQN) نقطه شروع ارائه و بررسی استراتژی معاملاتی یادگیری تقویتی عمیق جدید بود، ولی در ادامه این الگوریتم به میزان قابل توجهی با مسئله تصمیم‌گیری

[4] Open, High, Low, Close and Volume

معاملات الگوریتمی خاص موردنظر مطابقت داده شد. بهبودها و تغییرات متنوعی، که عمدتاً بر اساس شبیه‌سازی‌های متعدد انجام شده هستند، در ادامه خلاصه شده‌اند:

- مورد اول: معماری شبکه‌ی عصبی عمیق اولین تفاوت الگوریتم پیشنهادی ارائه شده با الگوریتم کیو_شبکه‌ی عمیق (DQN) کلاسیک، معماری شبکه‌ی عصبی عمیق (DNN) است که تابع مقدار_عمل $Q(s,a)$ را تقریب می‌زند. به دلیل ماهیت متفاوت ورودی‌ها (یعنی سری‌های زمانی به‌جای تصاویر خام)، به جای شبکه عصبی پیچشی (CNN) از یک شبکه‌ی عصبی عمیق (DNN) پیش‌خور کلاسیک به همراه برخی توابع فعال‌سازی واحد خطی یک‌سوشده لیکی ($LeakyReLU$) استفاده شده است.

- مورد دوم: کیو_ شبکه عمیق (DQN) دوگانه
مشکل الگوریتم (DQN) برآوردهایی بیش‌تر از مقدار واقعی است، که این خوش‌بینی بیش‌ازحد، عملکرد الگوریتم را تضعیف می‌کند. وان هاسلت و همکاران (۲۰۱۵)، برای کاهش تأثیر این پدیده نامطلوب، الگوریتم دوگانه‌ی (DQN) را ارائه کردند که مبتنی بر تجزیه حداکثر عملیات هدف در هر دو زمینه‌ی انتخاب عمل و ارزیابی عمل است.

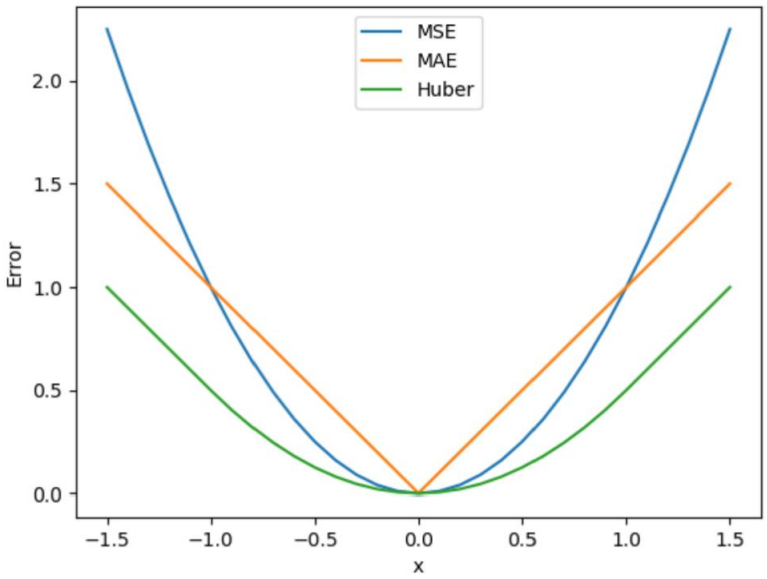
- مورد سوم: بهینه‌ساز $ADAM$
الگوریتم کلاسیک DQN بهینه‌ساز $RMSProp$ را اجرا می‌کند. بااین‌حال به‌طور تجربی ثابت شده است، بهینه‌ساز $ADAM$، که در کینگما و با (۲۰۱۵) معرفی شد، هم پایداری آموزش و هم سرعت همگرایی الگوریتم DRL را بهبود می‌بخشد.

- مورد چهارم: تابع زیان هوبر [۵]
اگرچه الگوریتم کلاسیک DQN نیز تابع زیان خطای میانگین مربعات (MSE) را پیاده‌سازی می‌کند، ولی تابع زیان هوبر به‌طور تجربی پایداری مرحله آموزش را بهبود می‌بخشد. چنین مشاهداتی این‌گونه توضیح داده می‌شود که تابع زیان MSE جریمه‌های قابل‌توجهی برای خطاهای بزرگ در نظر می‌گیرد که اگرچه به طورکلی مطلوب است ولی اثر جانبی منفی نامطلوبی بر الگوریتم DQN دارد. زیرا فرض بر آن است که DNN مقادیری را پیش‌بینی می‌کند که به ورودی خود بستگی دارند. مقدار این DNN نباید در یک به‌روزرسانی آموزشی خیلی زیاد تغییر کند، زیرا تغییرات خیلی زیاد منجر به تغییر قابل‌توجهی در هدف نیز می‌شود که آن‌هم به‌نوبه خود می‌تواند منجر به خطای بزرگ‌تر شود. در حالت ایده آل، به‌روزرسانی DNN باید به شیوه‌ای کندتر و پایدارتر انجام شود. از سوی دیگر، مشکل میانگین خطای مطلق (MAE) در نقطه‌ی صفر مشتق‌پذیر نیست. با استفاده از تابع ضرر هوبر H در زیر

[5] Huber loss

می‌توان توازن نسبتا مناسبی بین این دو ضرر ایجاد کرد:

$$H(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \leq 1 \\ |x| - \frac{1}{2} & \text{otherwise} \end{cases} \qquad (۱.۳)$$



شکل ۲.۳: مقایسه توابع ضرر هوبر، MSE و MAE

- مورد پنجم: محدود کردن شیب
این تکنیک در الگوریتم $TDQN$ برای حل مشکل افزایش بیش از حد شیب که منجر به ناپایداری‌های قابل توجهی در طول آموزش DNN می‌شود، استفاده می‌شود.

- مورد ششم: مقداردهی اولیه ژاویر [۶]
الگوریتم کلاسیک DQN به بیان ساده وزن‌های DNN را به‌طور تصادفی مقداردهی اولیه می‌کند، ولی روش مقداردهی اولیه ژاویر برای بهبود همگرایی الگوریتم پیاده‌سازی می‌شود. درواقع فرض می‌شود وزن‌های اولیه به‌گونه‌ای تنظیم شده‌اند که واریانس گرادیان در سراسر لایه‌های DNN ثابت می‌ماند.

[6] Xavier

- مورد هفتم: لایه‌های نرمال‌سازی دسته‌ای
این تکنیک یادگیری عمیق که توسط آیوف و زیگدی (۲۰۱۵) معرفی شده است، لایه ورودی را با تنظیم و مقیاس‌بندی توابع فعال‌سازی، نرمال می‌کند. این تکنیک مزایای بسیاری دارد ازجمله، مرحله آموزش را سریع‌تر و قوی‌تر می‌کند و همچنین تعمیم نتایج را بهبود می‌بخشد.

- مورد هشتم: تکنیک‌های تنظیم

چون در اولین آزمایش‌ها با استراتژی معاملاتی یادگیری تقویتی عمیق، تمایل زیادی به بیش‌برازش مشاهده شد، از سه تکنیک تنظیم استفاده کرده‌ایم:

  - اول: حذف تصادفی
  - دوم: تنظیم $L2$
  - سوم: توقف اولیه

- مورد نه ام: پیش‌پردازش و نرمال‌سازی حلقه آموزشی الگوریتم $TDQN$ با هر دو عملیات پیش‌پردازش و نرمال‌سازی مشاهدات یادگیری عمیق یعنی o_t انجام می‌شود. اولاً، چون سیگنال ناخواسته [۷] با بسامد بالا موجود در داده‌های معاملاتی به‌طور تجربی برای کاهش تعمیم‌پذیری الگوریتم در نظر گرفته شده‌است، یک عملیات فیلترینگ پایین‌ـگذر اجرا می‌شود. بااین‌حال، چنین عملیات پیش‌پردازشی هزینه‌بر است، زیرا برخی الگوهای معاملاتی بالقوه مفید را تغییر داده یا حتی از بین می‌برد و منجر به ایجاد تأخیری غیرقابل‌چشم‌پوشی می‌شود. ثانیاً، داده‌های حاصله به منظور انتقال اطلاعات معنی‌دارتر حرکات بازار تبدیل می‌شوند. معمولا از تغییرات روزانه قیمت‌ها به‌جای قیمت‌های خام استفاده می‌شود. ثالثاً، داده‌های باقی‌مانده نرمال‌سازی می‌شوند.

تکنیک‌های داده‌افزایی یک چالش کلیدی مسئله معاملات الگوریتمی، تعداد محدود داده‌های موجود است که معمولا کیفیت پایینی نیز دارند. برای حل این مشکل بزرگ از چند تکنیک داده‌افزایی استفاده شده‌است که عبارتند از تغییر سیگنال، فیلترینگ سیگنال، و اضافه کردن سیگنال ناخواسته مصنوعی. داده‌های معاملاتی جدیدی به طور مصنوعی با استفاده از این تکنیک‌های داده‌افزایی تولید می‌شوند که اندکی متفاوت‌اند ولی پدیده‌های مالی یکسانی را نتیجه می‌دهند. در نهایت، الگوریتم استراتژی معاملاتی $TDQN$ به تفصیل در قالب الگوریتم۱ نشان داده شده‌است.

[7] Noise

الگوریتم ۱ الگوریتم TDQN

Initialise the experience replay memory M of capacity C.
Initialise the main DNN weights (Xavier initialisation).
Initialise the target DNN weights $\theta^{-1} = \theta$.
for episode = 1 **to** $\mathbb{N}$ **do**
Acquire the initial observation o_1 from the environment $\mathcal{E}$ and preprocess it.
for $t = 1$ **to** T **do**
With probability ϵ, select a random action a_t from $\mathcal{A}$.
Otherwise, select $a_t = arg\max_{a\in\mathcal{A}} Q(o_t, a; \theta)$.
Copy the environment $\mathcal{E}^- = \mathcal{E}$.
Interact with the environment$\mathcal{E}$(action a_t) and get the new observation o_{t+1}
and reward r_t.
Perform the same operation on $\mathcal{E}^-$ with the opposite action a_t^-, getting o_{t+1}^-
and r_t^-.
Preprocess both new observations o_{t+1} and o_{t+1}^-.
Store both experiences $e_t = (o_t, a_t, r_t, o_{t+1})$ and $e_t^- = (o_t, a_t^-, r_t^-, o_{t+1}^-)$ in M.
if $t\%T' = 0$ **then**
Randomly sample from M minibatch of N_e experiences $e_i = (o_i, a_i, r_i, o_{i+1})$.
Set $y_i = \begin{cases} r_i & \text{if the state} s_{i+1} \text{ is terminal,} \\ r_i + \gamma Q(o_{i+1}, arg\max_{a\in\mathcal{A}} Q(o_{i+1}, a; \theta); \theta^-) & \text{otherwise.} \end{cases}$
Compute and clip the gradients based on the Huber loss $H(y_i, Q(o_i, a_i; \theta))$.
Optimise the main DNN parameters θ based on these clipped gradients.
Update the target DNN parameters $\theta^- = \theta$ every N^- steps.
end if
Annenl the ϵ-Greedy exploration parameter ϵ.
end for
end for

فصل ۴

ارزیابی عملکرد

استفاده از یک رویکرد ارزیابی دقیق عملکرد برای تولید نتایج معنادار حیاتی است. همان‌طور که قبلا اشاره شد، اهمیت این روش با توجه به فقدان واقعی روشی برای ارزیابی عملکرد مناسب در زمینه معاملات الگوریتمی بسیار زیاد است. در این بخش، روشی جدید و قابل‌اعتمادتر برای ارزیابی عینی عملکرد استراتژی‌های معاملات الگوریتمی، مثل الگوریتم $TDQN$، ارائه می‌شود.

۱.۰.۴ محک زدن روش پیشنهادی

در مطالعات گذشته، عملکرد استراتژی‌های معاملاتی با استفاده از ابزاری واحد (مثل بازار سهام یا سایر ابزارها) در دوره زمانی مشخصی ارزیابی و بررسی شده‌است. با این حال، تجزیه و تحلیل حاصل از چنین رویکرد کاملاً قابل‌اعتماد نیست، زیرا می‌توان داده‌های معاملاتی را به‌گونه‌ای انتخاب کرد که استراتژی معاملاتی سودآور به نظر برسد، حتی اگر در واقعیت چنین نباشد. برای جلوگیری از چنین اشتباهاتی، در حالت ایده‌آل، باید عملکرد رویکرد را روی ابزارهای متعددی با الگوهای مختلف ارزیابی کرد. در پروژه‌ی حاضر با هدف تولید نتایجی قابل اعتماد، رویکرد پیشنهادی مطرح شده بر روی نمونه‌ای حاوی ۳۰ سهم با ویژگی‌های متنوع (از نظر بخش، مناطق، نوسانات، نقدینگی و غیره) بررسی می‌شود. این نمونه متنوع در جدول ۱.۴ نشان داده شده‌است. برای جلوگیری از هرگونه سردرگمی، مرجع رسمی هر سهم (تیکر) [1] در پرانتز مشخص شده است. برای جلوگیری از هرگونه ابهام در مورد پروتکل‌های آموزش و ارزیابی، لازم به ذکر است که به هر سهم موجود در تست، استراتژی معاملاتی جدیدی آموزش داده می‌شود. با این حال، برای حفظ کلیت مطالعه، هیچ‌یک از فراپارامترهای الگوریتم در کل نمونه آزمون تغییر نمی‌کنند. افق زمانی معاملات بررسی شده هشت سال قبل انتخاب شده تا به خوبی معرف شرایط فعلی بازار باشد. ممکن است فرض کنید این دوره زمانی کوتاه محدودتر از آن است که بتواند کل مجموعه پدیده‌های مالی

[1] ticker

را منعکس کند. برای مثال، بحران مالی سال ۲۰۰۸ را کنار می‌گذاریم، هرچند که ارزیابی استواری استراتژی‌های معاملاتی مرتبط با چنین رویداد خارق‌العاده‌ای می‌تواند جالب باشد. بااین‌حال، این انتخاب ناشی از این واقعیت بود که احتمال تغییرات قابل‌توجه در رژیم بازار در افق معاملاتی کوتاه‌تر کمتر است و این امر ممکن است آسیبی جدی به ثبات آموزشی استراتژی‌های معاملاتی وارد کند. در نهایت، افق معاملاتی هشت‌ساله را به دو مجموعه آموزشی و آزمون به شرح زیر تقسیم کرده‌ایم:

- مجموعه آموزشی:

$$.1390/10/11 \rightarrow 1396/10/10$$

- مجموعه آزمون:

$$.1396/10/11 \rightarrow 1398/10/10$$

یک مجموعه اعتبار سنجی نیز به‌عنوان زیرمجموعه‌ای از مجموعه آموزشی برای تنظیم فراپارامترهای متعدد الگوریتم $TDQN$ در نظر گرفته شده است. باید توجه کنیم که پارامترهای DNN سیاست یادگیری تقویتی یعنی θ طی اجرای استراتژی معاملاتی در کل مجموعه آزمایشی ثابت هستند، یعنی تجربیات جدید به‌دست‌آمده برای آموزش بیشتر ارزش‌گذاری نمی‌شوند. با این حال، بررسی خلاف این وضعیت می‌تواند موضوع پژوهشی جالبی برای تحقیقات آتی باشد. در پایان این زیربخش، لازم به ذکر است که نمونه آزمایش پیشنهادی می‌تواند با تنوع‌بخشی به میزان بیشتری نیز بهبود داد. بعلاوه واضح است که با گنجاندن سهامی دیگر با وضعیت و دارایی‌های مالی متفاوت می‌تواند به بهبود نمونه آزمایشی کمک کند. نکته جالب دیگر ملاحظه دوره‌های زمانی مختلف آموزش/آزمایش ضمن حذف تغییرات قابل توجه رژیم بازار است، هرچند که در پروژه‌ی حاضر از این ایده به دلیل زمان مهمی که از قبل برای تولید نتایج نمونه آزمایش پیشنهادی لازم بود، استفاده نشده است.

۲.۰.۴ مقایسه استراتژی‌های معاملاتی

برای ارزیابی صحیح نقاط قوت و ضعف الگوریتم $TDQN$، نتایج حاصل از روش پیشنهادی خود را با نتایج چند استراتژی‌های معاملات الگوریتمی دیگر مقایسه کردیم. برای این منظور، فقط استراتژی‌های معاملاتی کلاسیکی که معمولاً در عمل استفاده می‌شوند را در نظر گرفتیم و مثلا استراتژی‌های مبتنی بر تکنیک‌های یادگیری عمیق یا سایر روش‌های پیشرفته را کنار گذاشتیم. هرچند که الگوریتم $TDQN$ یک استراتژی معاملاتی فعال است، ولی هر دو استراتژی غیرفعال و فعال را هم در نظر گرفتیم. برای منصفانه بودن مقایسه‌ها، فضاهای ورودی و خروجی استراتژی‌ها ($\mathcal{O}$ و $\mathcal{A}$) یکسان انتخاب شد. فهرست زیر خلاصه‌ای از استراتژی‌های معیار انتخاب شده را نشان می‌دهد:

بخش	منطقه		
	آمریکا	اروپا	آسیا
شاخص معاملاتی	Dow Jones(DIA) S&P ۵۰۰ (SPY)	FTSE ۱۰۰(EZU)	Nikkei ۲۲۵ (EWJ)
فناوری	Apple (AAPL) Google (GOOGL) Facebook (FB) Microsoft (MSFT)	Sony (۶۷۵۸.T) Nokia (NOK) Philips (PHIA.AS) Siemens (SIE.DE)	Baidu (BIDU) Tencent (۰۷۰۰.HK) Alibaba (BABA)
خدمات مالی	JPMorgan Chase (JPM)	HSBC (HSBC)	CCB (۰۹۳۹.HK)
انرژی	ExxonMobil (XOM)	Shell (RDSA.AS)	PetroChina (PTR)
خودروسازی	Tesla (TSLA)	Volkswagen (VOW۳.DE)	Toyota (۷۲۰۳.T)
غذایی	Cola Coca (KO)	InBev AB (ABI.BR)	Kirin (۲۵۰۳.T)

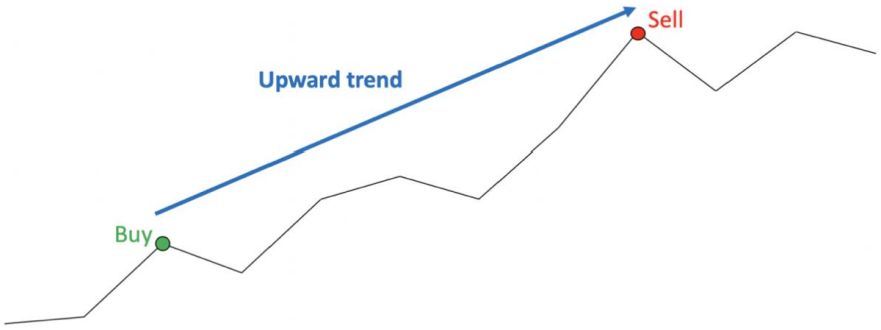
جدول ۱.۴: نمونه آزمون ارزیابی عملکرد

- خرید و نگهداری $(B\&H)$ [۲]
- فروش و نگهداری $(S\&H)$ [۳]
- معامله در جهت روند با توجه به میانگین متحرک (TF)
- بازگشت به میانگین با توجه به میانگین متحرک (MR)

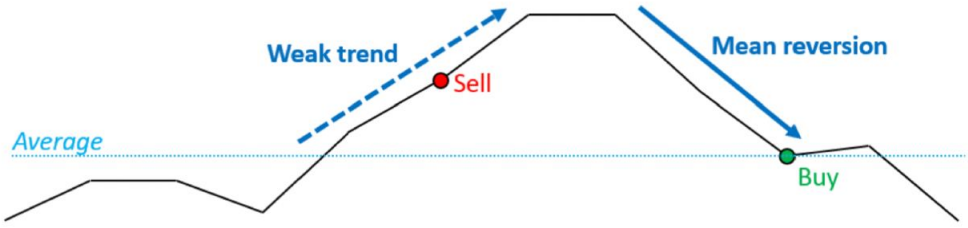
دو استراتژی معاملاتی معیار اول یعنی ($B\&H$ و $S\&H$) اصطلاحاً منفعل هستند، یعنی هیچ تغییری در وضعیت معاملاتی در افق معاملات رخ نمی‌دهد. در مقابل، دو استراتژی معیار بعدی یعنی (TF و MR) اصطلاحاً فعال هستند، یعنی تغییرات متعددی در وضعیت معاملاتی طی افق معاملات ایجاد می‌شود. از یک‌سو، از یک استراتژی معامله‌گری در جهت روند برای شناسایی و پیگیری روندهای مهم بازار، مطابق با شکل ۱.۴، استفاده شده‌است. از طرف دیگر، از یک استراتژی بازگشت به میانگین مبتنی بر تمایل بازار سهام برای بازگشت به میانگین قیمت قبلی خود در غیاب سایر روندهای آشکار استفاده کرده‌ایم (به شکل ۲.۴ مراجعه شود). از نظر طراحی، استراتژی معامله‌گری در جهت روند نسبت به استراتژی بازگشت به میانگین، معمولا سود بیشتری را ایجاد می‌کند، هرچند که عکس آن نیز می‌تواند صادق باشد. دلیلش این واقعیت است

[2] Buy&Hold
[3] Sale&Hold

که این دو خانواده استراتژی‌های معاملاتی دارای وضعیت‌های متضادی هستند: استراتژی بازگشت به میانگین همیشه منکر روندها است و عکس روند حرکت می‌کند، درحالی‌که استراتژی معامله‌گری در جهت روند، حرکات روند را دنبال می‌کند. برای مثال، یک بازار سهام برای بازگشت به میانگین قیمت قبلی خود در غیاب روندهای واضح. استراتژی پیروی از روند عموماً زمانی سود می‌آورد که استراتژی بازگشت متوسط سود نداشته باشد، برعکس نیز صادق است. این به دلیل این واقعیت است که این دو خانواده از استراتژی‌های معاملاتی موقعیت‌های متضادی را اتخاذ می‌کنند: یک استراتژی بازگشت متوسط همیشه روندها را انکار می‌کند و برخلاف آن حرکت می‌کند درحالی‌که استراتژی معامله‌گری در جهت روند، روند را دنبال می‌کند.



شکل ۱.۴: تصویر یک‌روند معمولی به دنبال استراتژی معاملاتی



شکل ۲.۴: تصویری از یک استراتژی معاملاتی بازگشت متوسط معمولی

۳.۰.۴ ارزیابی کمّی عملکرد

ارزیابی عملکرد کمی به معنای تعریف یک یا چند شاخص عملکرد برای تعیین کمی عملکرد استراتژی‌های معاملات الگوریتمی است. چون هدف اصلی استراتژی‌های معاملاتی سودآوری است،

در ارزیابی عملکرد آن‌ها باید به مقدار پول به‌دست‌آمده از آنها توجه کرد. با این حال، استدلال فوق ریسک مرتبط با فعالیت معاملاتی، که باید به‌طور مؤثری کاهش یابد، را در نظر نمی‌گیرد. درمجموع، پس از آنکه چندین بار متضرر شویم، متوجه می‌شویم که یک استراتژی معاملاتی با سود کم ولی پایدار بهتر از یک استراتژی معاملاتی با سود زیاد ولی بسیار ناپایدار است. در نهایت، انتخاب استراتژی معاملاتی به خود سرمایه‌گذار و تمایلش به کسب درآمد بیشتر در ازای پذیرش ریسک بیشتر بستگی دارد.

شرح شاخص	شاخص عملکرد
پول به‌دست‌آمده یا ازدست‌رفته در پایان فعالیت معاملاتی	نسبت شارپ
بازده فعالیت معاملاتی در مقایسه با ریسک‌پذیری آن	سود و زیان
بازده سالانه تولیدشده طی فعالیت معاملاتی	بازده سالانه
مدل‌سازی ریسک فعالیت معاملاتی	تغییرات سالیانه
درصد معاملات برنده طی فعالیت معاملاتی	نسبت سوددهی
نسبت سود و زیان متوسط فعالیت معاملاتی	نسبت سود و زیان
مشابه نسبت شارپ است، فقط با این تفاوت که ریسک منفی جریمه می‌شود	نسبت سورتینو
بیشترین ضرر از اوج به پایین در طی فعالیت معاملاتی	حداکثر افت سرمایه
مدت‌زمان حداکثر افت سرمایه فعالیت معاملاتی	مدت‌زمان حداکثر افت سرمایه

جدول ۲.۴: شاخص‌های کمی ارزیابی عملکرد

شاخص عملکرد مختلفی برای ارزیابی دقیق عملکرد استراتژی معاملاتی انتخاب شدند که مهم‌ترین آن‌ها نسبت شارپ است. این شاخص عملکرد، که کاربرد گسترده‌ای در زمینه معاملات الگوریتمی دارد، سودآوری و ریسک را با هم ترکیب می‌کند و به همین دلیل بسیار آگاهی‌بخش است. علاوه بر نسبت شارپ، از چند شاخص عملکرد دیگر نیز برای فراهم کردن بینش بیشتر استفاده کرده‌ایم. جدول ۲.۴ کل مجموعه شاخص‌های عملکردی استفاده شده برای کمّی سازی عملکرد استراتژی معاملاتی را نشان می‌دهد.

نمایش گرافیکی رفتار استراتژی معاملاتی در کنار محاسبه این شاخص‌های عملکردی متعدد جالب توجه است. ترسیم تغییرات قیمت بازار سهام p_t و سبد سهام v_t در کنار اعمال معاملاتی صادر شده توسط استراتژی معاملاتی برای تحلیل دقیق سیاست معاملاتی مناسب به نظر می‌رسد. بعلاوه، چنین نمایشی می‌تواند بینش بیشتری در مورد عملکرد و همچنین نقاط ضعف و قوت استراتژی تحلیل شده ارائه کند.

فصل ۵

ارائه و تحلیل چند مثال (سهام شرکت اپل و تسلا)

در این بخش، به بررسی ارزیابی استراتژی معاملاتی $TDQN$ بر اساس روش ارزیابی عملکردی که قبلا توضیح دادیم، می‌پردازیم. اولاً، تجزیه‌وتحلیل دقیقی روی موردی که نتایج خوبی را ارائه می‌کند و همچنین موردی که نتایجش کاهش‌یافته‌اند، انجام می‌دهیم. با این کار نقاط ضعف و قوت و محدودیت‌های الگوریتم $TDQN$ را تشخیص می‌دهیم. سپس، عملکرد استراتژی معاملاتی DRL را روی کل نمونه آزمون خلاصه و تحلیل می‌کنیم. در نهایت، توضیحات بیشتری درباره پارامتر ضریب تنزیل، تأثیر هزینه‌های معاملاتی و چالش‌های اصلی الگوریتم $TDQN$ ارائه شده است.

۱.۰.۵ نتایج مورد قبول سهام اپل

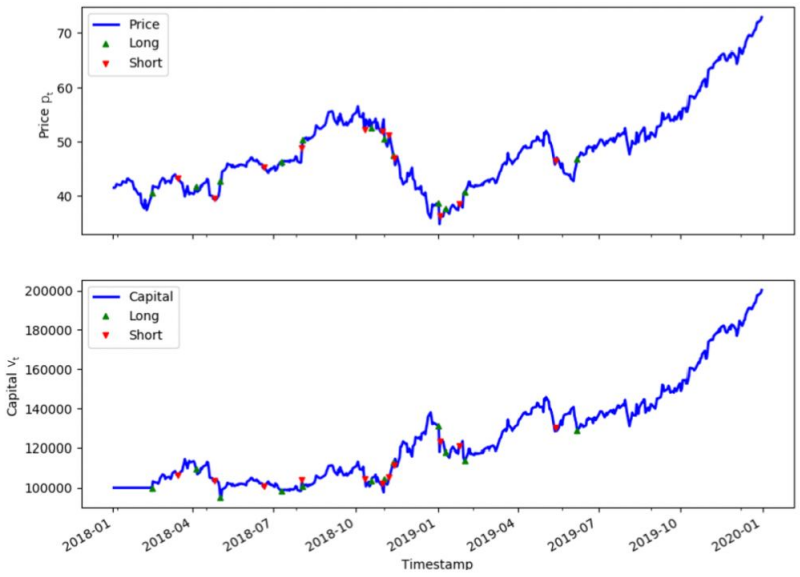
اولین بررسی دقیق اجرای استراتژی معاملاتی $TDQN$ روی سهام اپل انجام شد که نتایج امیدوارکننده‌ای حاصل شد. واریانس الگوریتم $TDQN$ همچون بسیاری الگوریتم‌های یادگیری تقویتی عمیق دیگر غیرقابل‌اغماض است. انجام چند آزمایش آموزشی با شرایط اولیه دقیقاً یکسان، قطعاً استراتژی‌های معاملاتی با تفاوت اندکی را برای متغیر نتیجه می‌دهند. در نتیجه، از این به بعد، هم بررسی اجرای معمولی الگوریتم $TDQN$ و هم بررسی عملکرد مورد انتظار از آن ارائه می‌شوند. روش اجرا: ابتدا، جدول ۱.۵ عملکرد حاصل از هر استراتژی معاملاتی بررسی شده را با این فرض که مقدار اولیه پول برابر با ۱۰۰۰۰۰ دلار است، نشان می‌دهد. الگوریتم $TDQN$ نتایج خوبی را هم از نظر درآمد و هم از نظر کاهش ریسک نتیجه می‌دهد و عملکردش از سایر استراتژی‌های معاملاتی فعال و غیرفعال معیار به‌وضوح بهتر است. دوم اینکه، شکل ۱.۵ نمودار تغییرات ارزش سبد عامل یادگیری تقویتی یعنی v_t و قیمت بازار سهام p_t را به همراه عمل a_t حاصل از الگوریتم $TDQN$

TDQN	MR	TF	S&H	B&H	شاخص عملکرد
1.484	−0.609	1.178	−1.593	1.239	نسبت شارپ
100288	−34630	68738	−80023	79823	سود و زیان(برحسب دلار)
32.81	−19.09	25.97	−100.00	28.86	بازده سالیانه (بر حسب درصد)
25.69	28.33	24.86	44.39	26.62	نوسانات سالیانه (بر حسب درصد)
52.17	56.67	42.31	0.00	100	نسبت سودآوری (بر حسب درصد)
2.958	0.492	3.182	0.00	∞	نسبت سود و زیان
1.841	−0.812	1.802	−2.203	1.558	نسبت سورتینو
17.31	51.12	14.89	82.48	38.51	حداکثر کاهش (بر حسب درصد)
25	204	20	250	62	حداکثر مدت زمان برداشت (برحسب روز)

جدول ۱.۵: ارزیابی عملکرد سهام اپل

نشان می‌دهد. مشاهده می‌شود که استراتژی معاملاتی یادگیری تقویتی عمیق قادر به تشخیص دقیق و استفاده از روندهای اصلی است، ولی با افزایش نوسانات و طی تغییرات رفتاری بازار عملکردش تضعیف می‌شود. همچنین مشاهده می‌شود که عامل معاملاتی معمولاً کمی از روندهای بازار عقب است، یعنی الگوریتم $TDQN$ آموخته است که در این سهم خاص، باید بیشتر واکنشی باشد تا پیش‌گستر. انتظار چنین رفتاری را از چنین فضای مشاهده محدودی $\mathcal{O}$، که دلایل جهت‌گیری‌های آتی بازار (مثل اعلام محصول جدید، گزارشات مالی، اقتصاد کلان و غیره) را در نظر نمی‌گیرد، می‌توان داشت. بااین‌حال، سیاست‌های آموخته شده صرفاً واکنشی نیستند. درواقع، مشاهده شد که ممکن است عامل یادگیری تقویتی تصمیم بگیرد که وضعیت معاملاتی خود را قبل از تغییر روند با توجه به افزایش نوسانات تغییر دهد، بنابراین پیش‌بینی می‌کند و فعال است.

عملکرد مورد انتظار: به‌منظور تخمین عملکرد مورد انتظار و همچنین واریانس الگوریتم $TDQN$، یک عامل معاملاتی یادگیری تقویتی چندین بار آموزش داده می‌شود. شکل ۲.۵ میانگین عملکرد الگوریتم $TDQN$ مجموعه‌های آموزشی و آزمایشی با توجه به تعداد قسمت‌های آموزشی (برای بیش‌از ۵۰ تکرار) نشان می‌دهد. این عملکرد مورد انتظار با عملکرد حاصله طی اجرای معمولی الگوریتم قابل‌مقایسه است. همچنین به نظر می‌رسد که تمایل بیش‌برازش عامل یادگیری تقویتی در این بازار خاص به‌درستی مدیریت شده است. توجه داشته باشید که اینکه عملکرد مجموعه آزمایش موقتاً برتر از عملکرد مجموعه آموزشی است، اشتباهی حاصل نشده است، بلکه به بیان ساده نشان دهنده بازاری آسان‌تر و سودآورتر برای دوره معاملاتی مجموعه آزمایشی برای سهام اپل است. این مثال به‌خوبی یکی از دشواری‌های عمده مسئله معاملات الگوریتمی را نشان می‌دهد: مجموعه‌های آموزشی و آزمایشی توزیع‌های یکسانی ندارند. درواقع، توزیع بازده روزانه پیوسته در حال تغییر است، که منجر به پیچیدگی آموزش استراتژی معاملاتی یادگیری تقویتی عمیق و همچنین پیچیدگی ارزیابی عملکرد این استراتژی می‌شود.

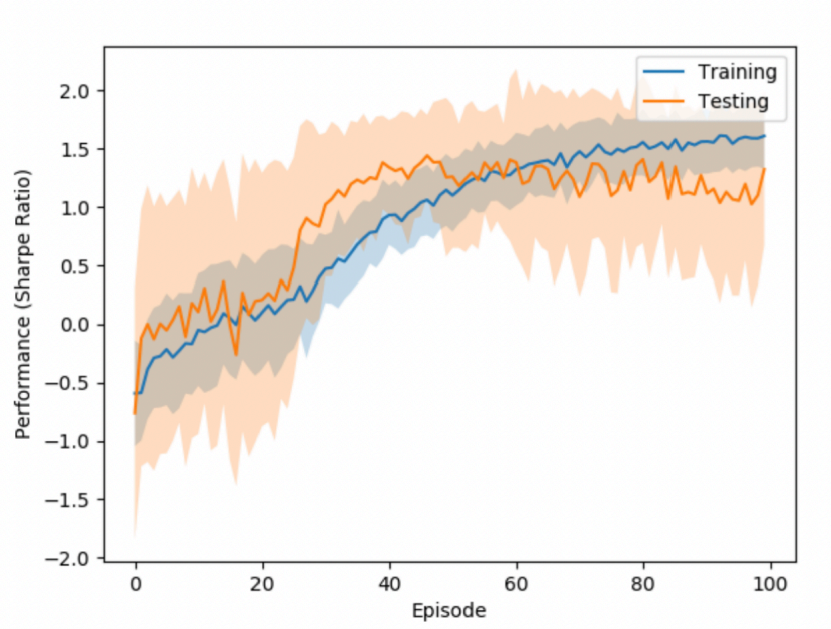


شکل ۱.۵: اجرای الگوریتم $TDQN$ برای سهام اپل (مجموعه آزمایشی)

۲.۰.۵ نتایج کاهش‌یافته سهام تسلا

تجزیه و تحلیل دقیق مشابهی مانند مثال سهام اپل روی سهام تسلا انجام شد و ویژگی‌های کاملا متفاوتی در مقایسه با سهام اپل (مانند نوسانات شدید) به دست آمد. برخلاف عملکرد امیدوارکننده‌ای که برای سهام اپل به دست آمد، سهام تسلا به‌ویژه برای برجسته کردن محدودیت‌های الگوریتم $TDQN$ انتخاب شده است.

اجرای معمولی: مشابه تجزیه وتحلیل حالت سهام اپل، جدول ۲.۵ عملکرد مربوط به استراتژی معاملاتی بررسی شده را نشان می‌دهد (با این فرض که مقدار اولیه پول برابر با ۱۰۰۰۰۰ دلار است). نتایج کاهش‌یافته حاصله از استراتژی‌های فعال معیار نشان می‌دهد که معامله سهام تسلا بسیار به دلیل نوسانات زیادش تا حدود زیادی دشوار است. اگرچه نسبت شارپ الگوریتم $TDQN$ مثبت است، ولی تقریباً هیچ سودی ایجاد نمی‌شود. بعلاوه، سطح ریسک این فعالیت معاملاتی واقعاً قابل‌قبول نیست. برای مثال، مدت‌زمان حداکثر افت سرمایه بسیار زیاد است، که این امر بر استرس اپراتور مسئول استراتژی معاملاتی می‌افزاید. شکل ۳.۵ تغییرات قیمت بازار سهام p_t و ارزش سبد عامل یادگیری تقویتی را همراه با اعمال a_t حاصله از الگوریتم $TDQN$ نشان می‌دهد و مؤید این مشاهدات است. بعلاوه، به‌وضوح مشاهده می‌شود که به دلیل نوسانات شدید سهام تسلا، استفاده از فرکانس معاملاتی بالاتر (تغییر در وضعیت‌های معاملاتی، که مطابق با وضعیتی $a_t \neq a_{t-1}$ است)،



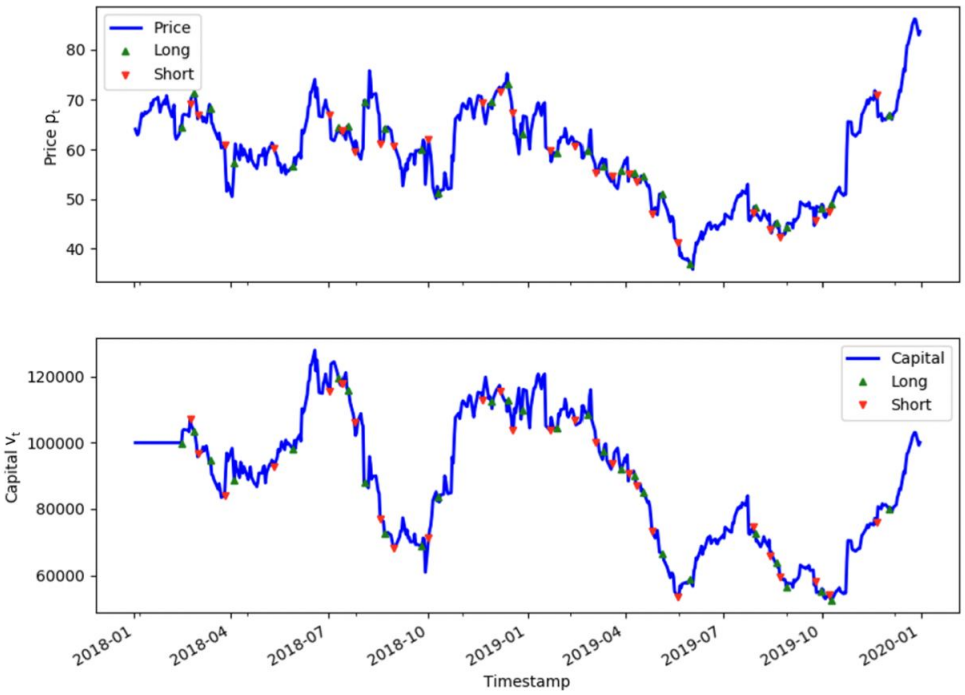
شکل ۲.۵: عملکرد مورد انتظار الگوریتم $TDQN$ برای سهام اپل

با وجود افزایش قابل توجه هزینه‌های معامله‌گری لازم است و این موضوع ریسک استراتژی معاملاتی یادگیری تقویتی عمیق را نیز بیشتر می‌کند.

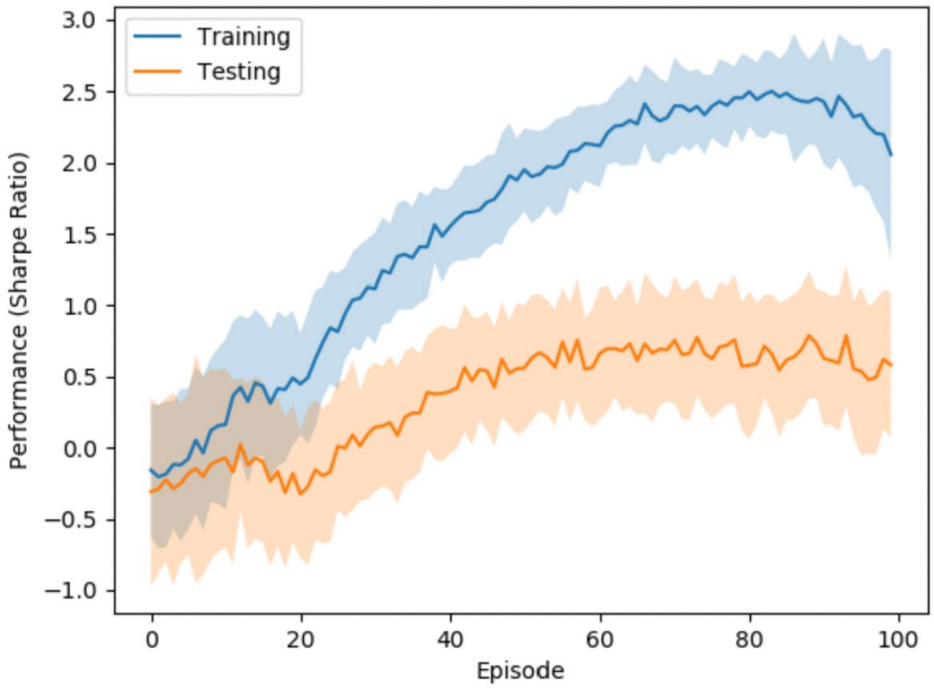
عملکرد مورد انتظار: در شکل ۴.۵، عملکرد مورد انتظار الگوریتم $TDQN$ برای مجموعه‌های آموزشی و آزمایشی به صورت تابعی از تعداد قسمت‌های آموزشی (در حالت بیش‌از ۵۰ تکرار) ترسیم شده است. مشخص است که این عملکرد مورد انتظار به طور قابل‌توجهی بهتر از عملکرد حاصله از اجرای معمولی تجزیه و تحلیل شده قبلی است، بنابراین نمی‌توان آن را نماینده واقعی رفتار میانگین در نظر گرفت. این حقیقت نشانگر محدودیتی کلیدی در الگوریتم $TDQN$ است: واریانس زیاد ممکن است منجر به انتخاب سیاست‌های ضعیف نسبت به عملکرد مورد انتظار شود. هرچند تکنیک‌های تنظیم متعددی پیاده‌سازی شده‌اند، ولی عملکرد بسیار بالاتر حاصله از مجموعه آموزشی نشان می‌دهد که الگوریتم یادگیری تقویتی عمیق در این مورد خاص، دچار بیش‌برازش شده است. پدیده بیش‌برازش را می‌توان تا حدودی با توجه به محدودیت فضای مشاهده $\mathcal{O}$ برای شناخت مؤثر سهام تسلا توضیح داد. حتی اگر پدیده بیش‌برازش در این مورد خاص خیلی مضر به نظر نرسد، ممکن است منجر به عملکرد ضعیفی در سایر سهم‌ها شود.

TDQN	MR	TF	S&H	B&H	شاخص عملکرد
0.261	0.358	−0.987	−0.154	0.508	نسبت شارپ
98	8600	−73301	−29847	29847	سود و زیان (برحسب درصد)
12.80	19.02	−100.00	−7.38	24.11	بازده سالیانه (بر حسب درصد)
52.09	58.05	52.70	46.11	53.14	نوسانات سالیانه (بر حسب درصد)
38.18	67.65	34.38	0.00	۱۰۰	نسبت سودآوری (بر حسب درصد)
1.621	0.496	0.534	0.00	∞	نسبت سود و زیان
0.359	0.539	−1.229	−0.205	0.741	نسبت سورتینو
58.95	65.31	79.91	54.09	52.83	حداکثر کاهش (بر حسب درصد)
331	159	229	144	205	حداکثر مدت زمان برداشت (برحسب روز)

جدول ۲.۵: ارزیابی عملکرد سهام تسلا



شکل ۳.۵: اجرای الگوریتم TDQN برای سهام تسلا (مجموعه آزمایشی)



شکل ۴.۵: عملکرد مورد انتظار الگوریتم $TDQN$ برای سهام تسلا

۳.۰.۵ نتایج عمومی نمونه آزمون

همان‌طور که قبلاً نیز اشاره شد، به‌منظور دستیابی به نتایجی قوی‌تر و قابل‌اعتمادتر، الگوریتم $TDQN$ را روی نمونه آزمون معرفی شده در بخش ۵-۱ بررسی و ارزیابی کردیم. جدول ۳.۵ حاوی نتایج نسبت شارپ مورد انتظار برای حاصله از الگوریتم $TDQN$ و استراتژی معاملاتی معیار روی کل مجموعه سهام موجود در نمونه آزمون است. نتایج عملکردی حاصله از استراتژی‌های معاملاتی معیار مؤید اهمیت تمایز بین استراتژی‌های غیرفعال (مثل $B\&H$ و $S\&H$) و استراتژی‌های فعال (TF و MR) است. درواقع، دومین خانواده استراتژی‌های معاملاتی، پتانسیل بیشتری دارند به‌شرط آنکه ریسک غیرقابل‌اغماض دیگری نیز بپذیرند: سفته‌بازی مستمر. چون بازارهای سهام عمدتاً صعودی بودند (یعنی قیمت p_t عمدتاً در حال افزایش بوده است) و برخی بی‌ثباتی‌ها طی دوره معاملات مجموعه آزمایشی رخ دادند، بهتر بودن عملکرد استراتژی خرید و نگهداری در مقایسه با سایر استراتژی‌های معاملاتی معیار عجیب نیست. درواقع، نه معامله‌گری در جهت روند و نه استراتژی بازگشت به میانگین، نتایج نسبتاً رضایت‌بخشی روی نمونه آزمون ارائه نکردند. در نتیجه، دشواری معامله‌گری فعال در چنین شرایط بازاری واضح است. در توجیه این عملکرد ضعیف‌تر می‌توان گفت که چنین استراتژی‌هایی معمولاً برای استفاده در الگوهای مالی خاصی مناسب هستند و چون همه‌منظوره نیستند اغلب عملکرد متوسط خوبی روی مجموعه بزرگی از سهام با ویژگی‌های متنوع ندارند. بعلاوه، چنین استراتژی‌هایی معمولاً فرکانس معاملاتی بالاتری دارند و بیشتر تحت

تأثیر هزینه‌های معاملاتی هستند (چون مشابه با شرایط پروژه‌ی حاضر، مدت‌زمان محاسبه میانگین متحرک نسبتاً کوتاه است). الگوریتم $TDQN$ با استفاده از استراتژی معاملاتی نوآورانه به نتایج امیدوارکننده‌ای روی نمونه آزمون رسید به‌طوری که عملکردش به‌طور متوسط بهتر از استراتژی‌های معاملاتی فعال معیار است. بااین‌حال، عملکرد استراتژی معاملاتی یادگیری تقویتی عمیق در این بازارهای صعودی خاص، که برای این استراتژی ساده غیرفعال مطلوب هستند، به‌ندرت بهتر از استراتژی خرید و نگهداری است. جالب اینکه عملکرد الگوریتم $TDQN$ در چند سهم مشابه با یا بسیار نزدیک به عملکرد استراتژی‌های معاملاتی غیرفعال ($B\&H$ و $S\&H$) است. در توضیح این اتفاق باید به این حقیقت اشاره کنیم که استراتژی یادگیری تقویتی عمیق به‌گونه‌ای مؤثر می‌آموزد که وقتی عدم اطمینان ناشی از معاملات فعال افزایش می‌یابد، باید از استراتژی معاملاتی منفعل استفاده کند. لازم به ذکر است که الگوریتم $TDQN$ نه از نوع معامله‌گری در جهت روند است و نه یک استراتژی معاملاتی بازگشت به میانگین است، زیرا هردوی این الگوهای مالی را می‌توان در عمل به‌گونه‌ای مؤثر مدیریت کرد. بنابراین، مزیت اصلی استراتژی معاملاتی یادگیری تقویتی عمیق مطمئناً تطبیق‌پذیری و توانایی آن در مدیریت کارآمد بازارهایی مختلف با ویژگی‌هایی متنوعی است.

۴.۰.۵ بحث و بررسی ضریب تنزیل

همان‌طور که در بخش ۳-۴ مطرح شد، ضریب تخفیف γ اهمیت پاداش‌های آتی را نشان می‌دهد. تنظیم مناسب این پارامتر در مقاله حاضر با توجه به عدم اطمینان قابل‌توجه آتی بی‌اهمیت نیست. ازیک‌طرف، به‌منظور اجتناب از هزینه‌های معاملاتی قابل‌توجه ناشی از فرکانس معاملاتی بیش‌ازحد بالا باید سیاست معاملاتی موردنظرمان بلندمدت باشد، یعنی: $(\gamma \to 1)$. از سوی دیگر، اهمیت بیش‌ازحد به آینده بازار سهامی غیرقطعی، عاقلانه نیست، یعنی: $(\gamma \to 0)$. بنابراین، توازنی شهودی برای پارامتر ضریب تنزیل وجود دارد. استدلال فوق با آزمایش‌های متعددی که برای تنظیم پارامتر γ انجام شده است تأیید می‌شود. درواقع، مشاهده شد که مقدار بهینه‌ای برای ضریب تنزیل وجود دارد که نه خیلی کوچک است و نه خیلی بزرگ. بعلاوه، این آزمایش‌ها پیوند پنهان بین ضریب تنزیل و بسامد معاملات را در اثر هزینه‌های معاملاتی آشکار کردند. از دیدگاه عامل یادگیری تقویتی، این هزینه‌ها مانعی برای غلبه بر عدم تغییر در وضعیت معاملاتی به دلیل کاهش فوری (و اغلب منفی) پاداش دریافتی است. درواقع، این حقیقت را مدل‌سازی می‌کند که عامل تجاری برای غلبه بر ریسک اضافی هزینه‌های معاملاتی باید نسبت به آینده به‌اندازه کافی مطمئن باشد. ضریب تنزیل تعیین‌کننده اهمیت اختصاص داده شده به آینده است به‌طوری‌که مقدار کوچک پارامتر γ منجر به کاهش تمایل عامل یادگیری تقویتی برای تغییر وضعیت معاملاتی خود می‌شود، که آن‌هم به‌نوبه خود بسامد معاملاتی الگوریتم $TDQN$ را کاهش می‌دهد.

	نسبت شارپ				
سهام	$B\&H$	$S\&H$	TF	MR	$TDQN$
$DowJones(DIA)$	0.684	−0.636	−0.325	−0.214	0.684
$S\&P500(SPY)$	0.834	−0.833	−0.309	−0.376	0.834
$NASDAQ100-(QQQ)$	0.845	−0.806	0.264	0.060	0.845
$FTSE100(EZU)$	0.088	0.026	−0.404	−0.030	0.103
$Nikkei225(EWJ)$	0.128	−0.025	−1.649	0.418	0.019
$Google(GOOGL)$	0.570	−0.370	0.125	0.555	0.227
$Apple(AAPL)$	1.239	−1.593	1.178	−0.609	1.424
$Facebook(FB)$	0.371	−0.078	0.248	−0.168	0.151
$Amazon(AMZN$	0.559	−0.187	0.161	−1.193	0.419
$Microsoft(MSFT)$	1.364	−1.390	−0.041	−0.416	0.987
$Twitter(TWTR)$	0.189	0.314	−0.271	−0.422	0.238
$Nokia(NOK)$	−0.408	0.565	1.088	1.314	−0.094
$Philips(PHIA.AS)$	1.062	−0.672	−0.167	−0.599	0.675
$Baidu(BIDU)$	−0.699	0.866	−1.209	0.167	0.080
$Alibaba(BABA)$	0.357	−0.139	−0.068	0.293	0.021
$Tencent(0700.HK)$	−0.013	0.309	0.179	−0.466	−0.198
$Sony(6758.T)$	0.794	−0.655	−0.352	0.415	0.424
$JPMorganChase(JPM)$	0.713	−0.743	−1.325	−0.004	0.722
$HSBC(HSBC)$	−0.518	0.725	−1.061	0.447	0.011
$CCB(0939.HK)$	0.026	0.165	−1.163	−0.388	0.202
$ExxonMobil(XOM)$	0.055	0.132	−0.386	−0.673	0.098
$Shell(RDSA.AS)$	0.488	−0.238	−0.043	0.742	0.425
$PetroChina(PTR)$	−0.376	0.514	−0.821	−0.238	0.156
$Tesla(TSLA)$	0.508	−0.154	−0.987	−0.358	0.621
$Volkswagen(VOW3.DE)$	0.384	−0.208	−0.361	0.601	0.216
$Toyota(7203.T)$	0.352	−0.242	−1.108	−0.378	0.304
$CocaCola(KO)$	1.031	−0.871	−0.236	−0.394	1.068
$ABInBev(ABI.BR)$	−0.058	0.275	0.036	−1.313	0.187
$Kirin(2503.T)$	0.106	0.156	−1.441	0.313	0.852
میانگین	0.369	−0.202	−0.331	−0.056	0.404

جدول ۳.۵: ارزیابی عملکرد برای کل نمونه آزمون

۵.۰.۵ بحث و بررسی هزینه‌های معاملاتی

تاثیر تجزیه‌وتحلیل هزینه‌های معاملاتی بر رفتار و عملکرد استراتژی معاملاتی حیاتی است، زیرا چنین هزینه‌هایی حاکی از وجود ریسک بیشتر و نیاز به تلاش برای کاهش این ریسک است. هزینه‌های معاملاتی دلیل اصلی مطالعه راه‌حل‌های یادگیری تقویتی عمیق به‌جای تکنیک‌های پیش‌بینی صرف مبتنی بر معماری‌های یادگیری عمیق است. همان‌طور که در بخش‌های قبلی دیدیم، صورت‌گرایی [1] یادگیری تقویتی ملاحظه مستقیم این هزینه‌های اضافی در فرآیند تصمیم‌گیری را ممکن می‌سازد. همچنین سیاست بهینه با توجه به ارزش هزینه‌های معاملاتی آموخته می‌شود. بالعکس، رویکردی پیش‌بینی‌کننده صرف فقط پیش‌بینی‌هایی درباره جهت یا قیمت‌های آتی بازار ارائه می‌دهد و هیچ نشانه‌ای درباره استراتژی معاملاتی مناسب با در نظر گرفتن هزینه‌های معاملاتی ارائه نمی‌کند. اگرچه رویکرد دوم انعطاف‌پذیری بیشتری دارد و استراتژی‌های معاملاتی ناشی از آن مطمئناً عملکرد خوبی دارند، ولی از نظر طراحی کارآمدی کمتری دارند. برای نمایش توانایی الگوریتم $TDQN$ در انطباق خودکار و کارآمد با هزینه‌های معاملاتی مختلف، ضمن ثابت نگه‌داشتن سایر پارامترها، شکل ۵.۵ رفتار استراتژی معاملاتی یادگیری تقویتی عمیق را برای سه مقدار هزینه مختلف نشان می‌دهد. به وضوح مشاهده می‌شود که با افزایش هزینه‌های معاملاتی، همان‌طور که انتظار می‌رود الگوریتم $TDQN$ بسامد معاملاتش را به‌طور مؤثر کاهش می‌دهد. وقتی این هزینه‌ها خیلی زیاد باشند، الگوریتم یادگیری تقویتی عمیق خیلی ساده انجام معاملات را متوقف می‌کند و رویکردی منفعل (نظیر استراتژی‌های خرید و نگهداری یا فروش و نگه‌داری) اتخاذ می‌کند.

۶.۰.۵ چالش‌های اصلی

امروزه، از روش‌های اصلی یادگیری تقویتی عمیق برای حل مشکلات واقعی مربوط به محیط‌هایی خاص با ویژگی‌های خاص مانند بازی‌ها استفاده می‌شود (برای مثال الگوریتم معروف آلفاگو که توسط سیلور و همکاران (۲۰۱۶) که در گوگل دیپ‌مایند ارائه شده است). در این مقاله، مسئله معاملات الگوریتمی روی محیطی کاملا متفاوت با پیچیدگی و عدم قطعیت قابل توجه مطالعه شده است. مطمئناً چالش‌های متعددی طی تحقیق پیرامون الگوریتم $TDQN$ رخ دادند که در ادامه عمده‌ترین آنها لیست می‌شود. اولاً، مشاهده‌پذیری بسیار ضعیف محیط معاملاتی، عملکرد الگوریتم $TDQN$ را به‌طور قابل‌توجهی تضعیف می‌کند. درواقع، اطلاعات در دسترس عامل یادگیری تقویتی برای توضیح دقیق پدیده‌های مالی رخ داده طی آموزش، که برای یادگیری کارآمد معاملات ضروری است، کافی نیست. ثانیاً، اگرچه توزیع بازده روزانه همواره در حال تغییر است، ولی اگر گذشته به‌اندازه کافی مبین آینده نباشد، الگوریتم $TDQN$ نتایج خوبی ارائه نخواهد داد. این امر باعث می‌شود استراتژی معاملاتی یادگیری تقویتی عمیق حساسیت ویژه‌ای به تغییرات قابل‌توجه رژیم بازار

[1] formalism

داشته باشد. ثالثاً، مدیریت بیش‌برازش الگوریتم $TDQN$ لازمه دستیابی به یک استراتژی معاملاتی قابل اعتماد است. با توجه به‌شدت مشکل بیش برازش در تکنیک‌های رایج یادگیری تقویتی عمیق به پروتکل‌های ارزیابی دقیق‌تری در یادگیری عمیق نیاز است. استفاده از تکنیک‌های یادگیری تقویتی عمیق در طیف وسیع‌تری از برنامه‌های واقعی زندگی مستلزم تحقیقات بیشتری روی این موضوع خاص است. در نهایت، واریانس زیاد الگوریتم‌های یادگیری تقویتی عمیق مانند DQN، استفاده موفق از این الگوریتم‌ها در حل مسائل خاص، به ویژه وقتی که مجموعه‌های آموزشی و آزمایشی تفاوت قابل‌توجهی دارند، را دشوار می‌سازد. این یکی از محدودیت کلیدی الگوریتم $TDQN$ است که قبلاً در بررسی سهام تسلا متوجه‌اش شده بودیم.

(a) Trading costs: 0% (b) Trading costs: 0.1% (c) Trading costs: 0.2%

شکل ۵.۵: تاثیر هزینه‌های معاملاتی روی الگوریتم $TDQN$ در بررسی سهام شرکت اپل

فصل ۶

نتیجه‌گیری

در پروژه‌ی حاضر الگوریتم کیو-شبکه یادگیری عمیق که یک روش حل یادگیری تقویتی عمیق برای مسئله معاملات الگوریتمی تعیین وضعیت معاملاتی بهینه در هر نقطه از زمان طی معامله در بازارهای سهام است، را بررسی کردیم. این استراتژی معاملاتی نوآورانه با ارزیابی دقیق عملکرد، نتایجی امیدوارکننده و به‌طور متوسط بهتری نسبت به استراتژی‌های معاملاتی معیار ارائه می‌کند. بعلاوه، الگوریتم $TDQN$ در مقایسه با رویکردهای کلاسیک‌تر، مزایای متعددی مانند تطبیق‌پذیری و استواری قابل‌توجه نسبت به هزینه‌های معاملاتی مختلف، دارد. بعلاوه، مزیت رویکرد مبتنی بر داده پیشنهادی، عدم نیاز به تعریف پیچیده قوانینی صریح و مناسب برای بازارهای مالی خاص است. با این حال، هنوز هم می‌توان عملکرد الگوریتم $TDQN$ را از نظر تعمیم و تکرارپذیری بهبود داد. چندین موضوع تحقیق برای ارتقای روش‌های یادگیری تقویتی عمیق مانند استفاده از لایه‌های $LSTM$ [1] در شبکه عصبی عمیق، که به پردازش بهتر داده‌های سری زمانی مالی کمک می‌کند (هاوسکنشت و استون (۲۰۱۵))، پیشنهاد شده است. همچنین موضوع مقایسه الگوریتم $TDQN$ با الگوریتم‌های یادگیری تقویتی عمیق بهینه‌سازی سیاست مانند الگوریتم بهینه‌سازی سیاست پروگزیمال PPO نیز می‌تواند جالب باشد. همچنین بیان رسمی مسئله معاملات الگوریتمی در یک محیط یادگیری تقویتی به صورت پیشرفته‌تر نیز می‌تواند راه‌گشا باشد. ابتدا، فضای مشاهده $\mathcal{O}$ باید برای افزایش قابلیت مشاهده محیط معاملاتی گسترش یابد. به طور مشابه، می‌توان برخی محدودیت‌های فضای عمل $\mathcal{A}$ را به‌منظور فعال شدن فرصت‌های معاملاتی جدید حذف کرد. دوم اینکه، می‌توان از مهندسی پاداش یادگیری تقویتی پیشرفته برای کاهش شکاف بین دو تابع هدف یادگیری عمیق و حداکثر سازی نسبت شارپ استفاده کرد. در نهایت، بررسی موضوع استفاده از توزیع‌ها به‌جای مقادیر مورد انتظار در الگوریتم $TDQN$ برای ملاحظه مفهوم ریسک و مدیریت بهتر عدم قطعیت نیز می‌تواند جالب باشد.

[1] Long short-term memory

فصل ۷

ضمیمه

۱.۷ استخراج فضای عمل $\mathcal{A}$

قضیه ۱.۷. فضای عمل یادگیری تقویتی $\mathcal{A}$ دارای یک کران بالا مانند $\overline{Q_t}$ است، بطوریکه داریم:

$$\overline{Q_t} = \frac{v_t^c}{p_t(1+C)}$$

اثبات. کران بالای فضای عمل یادگیری تقویتی $\mathcal{A}$ از این موضوع بدست می‌آید که ارزش نقدی v_t^c باید در کل معاملات مثبت باقی بماند. ایجاد این فرض که $v_t^c \geq 0$، تعداد سهام Q_t معامله شده توسط عامل یادگیری تقویتی در مرحله زمانی t باید چنان باشد که $v_{t+1}^c \geq 0$. با توجه به این شرط و همچنین با توجه به معادله‌ی شماره‌ی ۱۲ که داشتیم:

$$v_{t+1}^c = v_t^c - Q_t p_t - \underbrace{C\,|Q_t|\,p_t}_{\text{costs Trading}}$$

برای به روز رسانی ارزش نقدی داریم که:

$$v_t^c - Q_t p_t - C\,|Q_t|\,p_t \geq 0$$

همچنین دو حالت با توجه به مقدار Q_t ایجاد می‌شود:

- حالت اول: $Q_t < 0$
  عبارت قبلی به

$$\overline{v_t^c - Q_t p_t + C} Q_t p_t \geq 0$$
$$\Leftrightarrow Q_t \leq \frac{v_t^c}{p_t(1-C)}$$

تبدیل می‌شود. عبارت سمت راست نابرابری به دلیل این فرضیه که: $v_t^c \geq 0$، همواره مثبت است. همچنین در این حالت چون $Q_t < 0$ منفی است، شرط برآورده می‌شود.

- حالت دوم: $Q_t \geq 0$
این‌بار عبارت قبلی به

$$\begin{gathered} \overline{v_t^c - Q_t p_t + C} Q_t p_t \geq 0 \\ \Leftrightarrow Q_t \leq \frac{v_t^c}{p_t(1+C)} \end{gathered}$$

تبدیل می‌شود. این شرط کران بالایی (مثبت) را برای فضای عمل یادگیری تقویتی $\mathcal{A}$ نشان می‌دهد.

□

قضیه ۲.۷. فضای عمل یادگیری تقویتی $\mathcal{A}$ دارای یک کران پایین مانند $\underline{Q_t}$ است، بطوریکه:

$$\underline{Q_t} = \begin{cases} \frac{\Delta_t}{p_t \epsilon(1+C)} & \text{if } \Delta_t \geq 0 \\ \frac{\Delta_t}{p_t(2C+\epsilon(1+C))} & \text{if } \Delta_t < 0 \end{cases}$$

که در آن: $\Delta_t = -v_t^c - n_t p_t (1+\epsilon)(1+C)$.

اثبات. کران پایین فضای عمل یادگیری تقویتی $\mathcal{A}$ از این موضوع بدست می‌آید که ارزش نقدی v_t^c می‌باست برای بازگشت به موقعیت خنثی در کل افق معاملاتی $(n_t = 0)$ کافی باشد. ایجاد این فرضیه که این شرط در مرحله زمانی t برآورده می شود، تعداد سهام Q_t معامله شده توسط عامل یادگیری تقویتی باید به نحوی باشد که این شرط در زمان بعدی $t+1$ هم صادق بماند. با در نظر گرفتن موضوع گفته شده و وارد کردن این موضوع در معادله‌ی $v_{t+1}^c = v_t^c - Q_t p_t - \underbrace{C\left|Q_t\right| p_t}_{\text{costs Trading}}$ داریم:

$$v_t^c - Q_t p_t - C\left|Q_t\right| p_t \geq -\left(n_t + Q_t\right) p_t (1+C)(1+\epsilon)$$

دو حالت با توجه به مقدار Q_t ایجاد می‌شود:

- حالت اول: $Q_t \geq 0$
عبارت قبلی به

$$\begin{gathered} \overline{v_t^c - Q_t p_t - C} Q_t p_t \geq -\left(n_t + Q_t\right) p_t (1+C)(1+\epsilon) \\ \Leftrightarrow Q_t \geq \frac{-v_t^c - n_t p_t (1+C)(1+\epsilon)}{p_t \epsilon (1+C)} \Leftrightarrow v_t^c \geq -n_t p_t (1+C)(1+\epsilon) - Q_t p_t \epsilon (1+C) \\ \Leftrightarrow Q_t \geq \frac{-v_t^c - n_t p_t (1+C)(1+\epsilon)}{p_t \epsilon (1+C)} \end{gathered}$$

تبدیل می‌شود. عبارت سمت راست نابرابری نشان دهنده اولین کران پایین برای فضای عمل یادگیری تقویتی است.

- حالت دوم: $Q_t < 0$
در این حالت نیز عبارت قبلی به

$$\begin{aligned} &\overline{v_t^c - Q_t p_t + C} Q_t p_t \geq -\left(n_t + Q_t\right) p_t (1+C)(1+\epsilon) \\ &\Leftrightarrow v_t^c \geq -n_t p_t (1+C)(1+\epsilon) - Q_t p_t (2C + \epsilon + \epsilon C) \\ &\Leftrightarrow Q_t \geq \tfrac{-v_t^c - n_t p_t (1+C)(1+\epsilon)}{p_t (2C + \epsilon(1+C))} \end{aligned}$$

تبدیل می‌شود.

عبارت سمت راست نابرابری نشان دهنده دومین کران پایین برای فضای عمل یادگیری تقویتی $\mathcal{A}$ است. □

هر دو کران پایین به دست آمده در روابط بالا، در کسرهایشان دارای صورت یکسانی هستند که با Δ_t نمایش داده شده‌است. این مقدار نشان دهنده‌ی تفاوت بین حداکثر هزینه‌ی مفروض برای بازگشت به موقعیت خنثی همراه با ارزش نقدی فعلی عامل، v_t^c در زمان بعدی $t+1$ است. این عبارت بررسی می‌کند که اگر عامل در مرحله‌ی زمان فعلی هیچ کاری انجام ندهد، آیا می‌تواند بدهی خود را در بدترین حالت فرضی در زمان بعدی پرداخت کند؟ برای پاسخ به این سوال با توجه به علامت Δ_t دو حالت رخ می‌دهد:

- $\Delta_t < 0$
عامل معاملات در این شرایط مشکلی برای پرداخت بدهی خود ندارد. این شرط زمانی برقرار است که همواره تعداد مثبت سهام $(n_t \geq 0)$ موجود باشد.

- $\Delta_t \geq 0$
ممکن است عامل معاملاتی در شرایطی که قبلا توضیح داده شد، در پرداخت بدهی مشکل داشته باشد، محدودترین حد پایینی از این دو به شرح زیر است:

$$\underline{Q_t} = \tfrac{\Delta_t}{p_t \epsilon (1+C)}$$

کتاب‌نامه

[1] Ralf Herbrich, Thore Graepel, Masashi Sugiyama, Statistical reinforcement learning, Modern machine learning, Chapman & Hall/CRC, Taylor & Francis Group, 2015

[2] J. Carapuco, R. Neves, N. Horta, Reinforcement Learning applied to Forex Trading. Appl. Soft Comput., 73:783-794, 2018.

[3] Yue Deng, Feng Bao, Youkong Kong, Zhiquan Ren, Qionghai Dai, Deep direct reinforcement learning for financial signal representation and trading, Institute of Electrical and Electronics Engineers Trans, Neural Network and Learning System, 653-664 (2017)

[4] Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). A Brief Survey of Deep Reinforcement Learning. CoRR, abs/1708.05866.

[5] Bailey, D. H., Borwein, J. M., de Prado, M. L., and Zhu, Q. J. (2014). Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance. Notice of the American Mathematical Society, pages 458-471.

[6] Fortunato, M., Azar, M. G., Piot, B., Menick, J., Hessel, M., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. (2018). Noisy Networks for Exploration. CoRR, abs/1706.10295.

[7] Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. (2017). Deep Direct Reinforcement Learning for Financial Signal Representa- tion and Trading. IEEE Transactions on Neural Networks and Learning Systems, 28:653-664.

[8] Sutton, R. S. and Barto, A. G. (2018). Reinforcement Learning: An Introduction. The MIT Press, second edition.

[9] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-Level Control through Deep Reinforcement Learning. Nature, 518:529-533.

[10] van Hasselt, H. P., Guez, A., and Silver, D. (2015). Deep Reinforcement Learning with Double Q-Learning. CoRR, abs/1509.06461.

Abstract

This scientific research paper investigates an innovative approach based on reinforcement learning (RL) to solve the algorithmic trading problem of determining the optimal trading position at any point in time during a trading activity in the stock market. It proposes a novel DRL trading policy so as to maximise the resulting Sharpe ratio performance indicator on a broad range of stock markets. Denominated the Trading Deep Q-Network algorithm (TDQN), this new DRL approach is inspired from the popular DQN algorithm and significantly adapted to the specific algorithmic trading problem at hand. The training of the resulting reinforcement learning (RL) agent is entirely based on the generation of artificial trajectories from a limited set of stock market historical data. In order to objectively assess the performance of trading strategies, the research paper also proposes a novel, more rigorous performance assessment methodology. Following this new performance assessment approach, promising results are reported for the TDQN algorithm.

College of Science
School of Mathematics, Statistics, and Computer Science

An Application of Reinforcement Learning to Algorithmic Trading

Parinaz Zarei

Supervisor:
Dr. Hedieh Sajedi

A thesis submitted to Graduate Studies Office
in partial fulfillment of the requirements for the degree of
B.Sc. in
Applied Statistics

July 2022