



دانشگاه صنعتی شریف
دانشکده‌ی علوم ریاضی

پایان‌نامه‌ی کارشناسی ارشد
رشته علوم کامپیوتر

عنوان:

دسته بندی الگوریتم های یادگیری تقویتی

نگارش:

حسین یوسفی زاده

استاد راهنما:

دکتر دانشگر

دی ۱۳۹۹

سلام افلا

به نام خدا
دانشگاه صنعتی شریف
دانشکده‌ی علوم ریاضی

پایان‌نامه‌ی کارشناسی ارشد

عنوان: دسته بندی الگوریتم های یادگیری تقویتی
نگارش: حسین یوسفی زاده

کمیته‌ی ممتحنین

امضاء: استاد راهنما: دکتر دانشگر

امضاء: استاد مشاور: استاد مشاور

امضاء: استاد مدعو: استاد ممتحن

تاریخ:

چکیده

تئوری یادگیری تقویتی (RL) به تدریج به یکی از فعال ترین حوزه های تحقیقاتی در یادگیری ماشین و هوش مصنوعی تبدیل شده، که ریشه در دیدگاه های روانشناختی و علوم اعصاب درباره رفتار حیوانات و انسان دارد. یادگیری تقویتی تلاش می کند به این سوال پاسخ دهد: چه کار کنیم که بیشترین پاداش یا کمترین هزینه نصیبمان شود؟ اینکه چگونه عوامل هوشمند می توانند کنترل خود را روی محیط بهینه کنند نیز در حوزه یادگیری تقویتی قرار می گیرد. عوامل هوشمند با کاری دشوار روبرو می شوند: آنها باید مدل های کارآمدی از محیط را با استفاده از ورودی های حسی بدست آورند و از این مدل ها برای تعمیم تجربه گذشته به موقعیت های جدید استفاده کنند. به نظر می رسد انسان و سایر حیوانات این مشکل را از طریق ترکیب هماهنگ یادگیری تقویتی و سیستم های حسی سلسله مراتبی حل می کنند. در این پایان نامه، به بررسی روش های مبتنی بر یادگیری تقویتی در حالت گسسته می پردازیم و الگوریتم های مهم آن را خواهیم دید. نهایتاً چند کاربرد دیگر را بررسی خواهیم کرد.

کلیدواژه ها: یادگیری تقویتی

فهرست مطالب

۸	۱ مقدمه
۹	۱-۱ تعریف مسئله
۹	۲-۱ اهمیت موضوع
۹	۳-۱ فضای حالت‌ها و عمل‌ها
۹	۴-۱ خط مشی
۹	۵-۱ سیگنال پاداش
۱۰	۶-۱ تابع ارزش
۱۰	۷-۱ محیط
۱۰	۸-۱ مسئله اکتشاف و بهره برداری
۱۰	۹-۱ اهداف تحقیق
۱۰	۱۰-۱ ساختار پایان‌نامه
۱۱	۲ مفاهیم اولیه یادگیری تقویتی
۱۲	۱-۲ دینامیک عامل-محیط
۱۲	۲-۲ فرایند تصمیم‌گیری مارکوف
۱۴	۳-۲ عناصر اصلی یادگیری تقویتی

۱۴ ۲-۳-۱ خطمشی
۱۴ ۲-۳-۲ سیگنال پاداش
۱۵ ۲-۳-۳ عایدی و تابع ارزش
۱۶ ۲-۳-۴ محیط
۱۶ ۲-۴ برنامه‌ریزی پویا
۱۶ ۲-۴-۱ خطمشی و تابع ارزش بهینه
۱۷ ۲-۴-۲ معادله بلمن
۱۸ ۲-۴-۳ بهینگی و معادله بهینگی بلمن
۱۸ ۲-۴-۴ برنامه‌ریزی پویا
۱۹ ۲-۴-۵ بهبود خطمشی
۲۰ ۲-۴-۶ الگوریتم خطمشی تکراری
۲۰ ۲-۴-۷ الگوریتم Iteration Value
۲۱ ۲-۴-۸ درهم تنیدگی PE و PI و الگوریتم های GPI
۲۲ ۲-۵ Q-learning

۳ نتایج اخیر ۲۳

۲۳ ۳-۱ روش‌های مبتنی بر مدل و بدون مدل
۲۴ ۳-۲ روش‌های بدون مدل
۲۴ ۳-۲-۱ روش‌های مبتنی بر ارزش
۲۷ ۳-۲-۲ روش‌های مبتنی بر خطمشی
۳۴ ۳-۲-۳ مقایسه روش بهینه سازی خطمشی و Q-learning
۳۴ ۳-۳ روش‌های مبتنی بر مدل
۳۴ ۳-۳-۱ روش مدل جهان

فصل ۱

مقدمه

یک روش یادگیری ماشین تعریف می شود که مربوط به نحوه اقدام یک عامل هوشمند^۱ در محیط براساس هدفی مشخص است. یادگیری تقویتی عبارت است از یادگیری اینکه عامل هوشمند چه کاری باید انجام دهد (نحوه انتخاب اقدامات برحسب موقعیت) تا به حداکثر پاداش برسد. این روش تمامی مسائل یک عامل هدفمند را صریحا در تعامل با یک محیط نامشخص بررسی می کند. به عامل هوشمند گفته نمی شود که چه کارهایی را انجام دهد، اما در عوض باید کشف کند که کدام اقدامات، بیشترین پاداش را به همراه دارد. در جالب ترین و چالش برانگیزترین موارد، اقدامات ممکن دارد نه تنها بر پاداش فوری بلکه در وضعیت بعدی محیط، و از طریق آن، بر کلیه پاداش های بعدی تأثیر بگذارد. این دو ویژگی (جستجوی آزمون و خطا و پاداش تأخیری) دو ویژگی مهم تمیز دهنده یادگیری تقویتی از روش های متداول یادگیری ماشین هستند. اقدامات عامل هوشمند، می تواند بر وضعیت آینده محیط تأثیر بگذارد یادگیری تقویتی یک رویکرد محاسباتی برای درک و خودکار کردن یادگیری و تصمیم گیری هدفمند است. یادگیری تقویتی، یادگیری از طریق تعامل است که چگونه می توان برای رسیدن به یک هدف رفتار کرد. عامل یادگیری تقویتی و محیط، در طی مراحل زمانی گسسته یا پیوسته با یکدیگر تعامل دارند.

^۱ Agent

۱-۱ تعریف مسئله

مسئله‌ی یادگیری تقویتی در اصل یک مسئله بهینه سازی است. هدف اصلی مسئله، به حداکثر رساندن پاداشی است که از محیط دریافت می شود

تعریف دقیق تر این مسئله را در فصل دوم خواهیم دید.

۲-۱ اهمیت موضوع

یادگیری تقویتی در بسیاری از رشته ها مانند نظریه بازی، نظریه کنترل، تحقیق در عملیات، نظریه اطلاعات، بهینه سازی مبتنی بر شبیه سازی، سیستم های چند عاملی، هوش انبوه و آمار مورد مطالعه قرار می گیرد. در ادبیات تحقیق و کنترل عملیات، یادگیری تقویتی را برنامه ریزی تقریبی پویا^۲ یا برنامه ریزی عصبی پویا^۳ می نامند. مسائل مورد بررسی در یادگیری تقویتی در نظریه کنترل بهینه^۴ نیز مورد بررسی قرار گرفته است، که بیشتر مربوط به وجود و توصیف راه حل های بهینه و الگوریتم های محاسبه دقیق آنهاست، و کمتر مربوط به یادگیری یا تقریب، به ویژه در غیاب یک مدل ریاضی از محیط. در اقتصاد و نظریه بازی، ممکن است از یادگیری تقویتی برای توضیح چگونگی ایجاد تعادل، استفاده شود.

۳-۱ فضای حالت ها و عمل ها

۴-۱ خط مشی

۵-۱ سیگنال پاداش

In reinforcement learning, the agent of goal or purpose is formalized in terms of the learning agent. the to environment the from passing reward, the called signal, special a of

^۲Approximate Dynamic Programming

^۳Neuro-dynamic Programming

^۴Optimal Control Theory

agent's the Informally, R_t number, simple a is reward the step, time each At
 maximiz- means This receives. it reward of amount total the maximize to is goal
 clearly can We run. long the in reward cumulative but reward, immediate not ing
 hypothesis reward the as idea informal this state

همه ی آنچه به عنوان هدف مدنظر داریم می تواند به صورت بیشینه سازی مقدار میانگین یک سیگنال
 عددی بیان شود.

۱-۶ تابع ارزش

۱-۷ محیط

۱-۸ مسئله اکتشاف و بهره برداری

یکی از چالش هایی که در یادگیری تقویتی برخلاف سایر روش های یادگیری وجود دارد ، رقابت بین
 اکتشاف و بهره برداری است. برای به دست آوردن پاداش زیاد ، یک عامل یادگیری تقویتی باید کارهایی
 را ترجیح دهد که در گذشته انجام داده و در تولید پاداش موثرتر بوده است. اما برای کشف چنین اعمالی،
 باید اقداماتی را امتحان کند که قبلاً انتخاب نکرده است. این عامل برای به دست آوردن پاداش مجبور
 است از آنچه قبلاً تجربه کرده است بهره برداری کند، اما همچنین برای انتخاب اقدامات بهتر در آینده
 باید به کاوش بپردازد. اکتشاف و بهره برداری هیچکدام به تنهایی در رسیدن به هدف، کارا نیست.

۱-۹ اهداف تحقیق

۱-۱۰ ساختار پایان نامه

این پایان نامه شامل — فصل است. در فصل ...

فصل ۲

مفاهیم اولیه یادگیری تقویتی

کنترل بهینه^۱ در اواخر دهه ۱۹۵۰ برای توصیف مسئله طراحی یک کنترل‌گر برای به حداقل رساندن اندازه‌گیری رفتار سیستم دینامیکی^۲ در طول زمان مورد استفاده قرار گرفت. یکی از رویکردهای این مسئله در اواسط دهه ۱۹۵۰ توسط ریچارد بلمن^۳ و دیگران از طریق گسترش نظریه قرن نوزدهم همیلتون^۴ و جاکوبی^۵ توسعه یافت. این رویکرد از مفاهیم حالت یک سیستم دینامیکی و یک تابع ارزش^۶ برای تعریف یک معادله تابعی استفاده می‌کند؛ که اکنون معادله بلمن^۷ نامیده می‌شود. مجموعه روش‌های حل مسائل کنترل بهینه به کمک معادله بلمن به عنوان برنامه‌ریزی پویا شناخته می‌شود. همچنین بلمن نسخه گسسته از مسئله کنترل بهینه را که تحت عنوان فرایندهای تصمیم‌گیری مارکوف^۸ شناخته می‌شود، معرفی کرد. رونالد هوارد (۱۹۶۰) روش Policy Iteration را برای MDP ها طراحی کرد. همه این‌ها عناصر اساسی در تئوری و الگوریتم‌های یادگیری تقویتی^۹ مدرن هستند. در این فصل، با فرایندهای تصمیم‌گیری مارکوف آشنا خواهیم شد و چهار عنصر اصلی یادگیری تقویتی، یعنی خط‌مشی^{۱۰}، سیگنال

Optimal control^۱

Dynamical system^۲

Richard Bellman^۳

Hamilton^۴

Jacobi^۵

Value function^۶

Bellman equation^۷

Markov decision process(MDP)^۸

Reinforcement learning (RL)^۹

Policy^{۱۰}

پاداش^{۱۱}، تابع ارزش و محیط^{۱۲} را دقیقاً تعریف خواهیم کرد. همچنین برخی روش‌های کلاسیک در یادگیری تقویتی را معرفی می‌کنیم.

۲-۱ دینامیک عامل-محیط

حوزه یادگیری تقویتی دو بازیگر اصلی دارد: عامل و محیط. موجود تصمیم‌گیرنده و آموزنده، عامل یادگیری یا به اختصار، عامل نامیده می‌شود. قسمتی که عامل با آن تعامل دارد، شامل هر چیز خارج از عامل، محیط نامیده می‌شود. در ادبیات کنترل بهینه، معمولاً به جای واژه‌های عامل و محیط، از کنترل‌کننده و سیستم کنترل شده استفاده می‌شود. عامل و محیط به طور مداوم با یکدیگر ارتباط برقرار می‌کنند، عامل انتخاب می‌کند که چه اقدامی انجام دهد و محیط، به این اقدامات پاسخ می‌دهد و موقعیت جدیدی را به عامل ارائه می‌دهد. محیط همچنین مقادیر عددی ویژه‌ای به نام پاداش به عامل برمیگرداند، که عامل به دنبال به حداکثر رساندن آن است.

به طور خاص، عامل و محیط در یک توالی زمانی گسسته تعامل می‌کنند $t = 0, 1, 2, 3, \dots$ در هر مرحله t ، عامل وضعیت محیط $S_t \in \mathcal{S}$ را دریافت می‌کند، و براساس آن یک عمل $A_t \in \mathcal{A}$ را انتخاب می‌کند. در گام بعدی، عامل به عنوان نتیجه عمل خود، یک پاداش عددی $R_{t+1} \in \mathcal{R}$ دریافت می‌کند و خود را در حالت جدید S_{t+1} می‌یابد. دینامیک عامل-محیط را می‌توان به شکل یک دنباله از حالت‌ها، عمل‌ها و پاداش‌ها به شکل زیر نمایش داد:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots$$

۲-۲ فرایند تصمیم‌گیری مارکوف

یادگیری تقویتی از چارچوب رسمی فرایندهای تصمیم‌گیری مارکوف (MDP) برای تعریف تعامل بین یک عامل یادگیری و محیط آن توسط حالت‌ها، اقدامات و پاداش استفاده می‌کند. مدل MDP یک مدل کلاسیک از تصمیم‌گیری متوالی است، جایی که اقدامات نه تنها بر پاداش‌های فوری، بلکه بر موقعیت‌ها و حالت‌های بعدی و به تبع آن بر پاداش‌های آینده تأثیر می‌گذارد. MDP یک فرم ایده‌آل ریاضی از

^{۱۱} Reward signal

^{۱۲} Environment

مسئله یادگیری تقویتی است که برای آن تئوری‌های دقیقی بیان شده‌است. MDP متناهی، یک MDP با مجموعه حالت‌های محدود است. بیشتر نظریه‌های فعلی یادگیری تقویتی، محدود به MDP متناهی است، اما روش‌ها و ایده‌ها به طور کلی بیان می‌شوند.

تعریف ۱-۲ (فرایند تصمیم‌گیری مارکوف) فرایند تصمیم‌گیری مارکوف

یک ۴ تایی،

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P} \rangle$$

است که

- \mathcal{S} بیانگر مجموعه تمام حالت‌هاست،
- \mathcal{A} بیانگر مجموعه تمام عمل‌هاست،
- $\mathcal{R} \subseteq \mathbb{R}$ بیانگر مجموعه پاداش‌هاست،
- \mathcal{P} هسته احتمال انتقال^{۱۳} $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S} \times \mathcal{R})$ تابعی است که دینامیک MDP را مشخص می‌کند.

هسته احتمال انتقال یا تابع انتقال \mathcal{P} ، هر دوتایی حالت-عمل (s, a) ، که $s \in \mathcal{S}$ و $a \in \mathcal{A}$ ، را به یک توزیع احتمال روی دوتایی‌هایی به شکل (s', r) نسبت می‌دهد که s' بیانگر حالت بعدی و r بیانگر پاداش این انتقال است. به ازای هر دو حالت $s, s' \in \mathcal{S}$ و هر عمل $a \in \mathcal{A}$ و هر پاداش $r \in \mathcal{R}$ احتمال رسیدن به حالت s' و دریافت پاداش r با انتخاب عمل a در حالت s ، یک عدد حقیقی عضو $[0, 1]$ است که آن را به شکل $p(s', r | s, a)$ نمایش می‌دهیم:

$$p(s', r | s, a) \triangleq \Pr\{S_t = s', R_t = r | S_{t-1} = s, A_{t-1} = a\}$$

نامگذاری فرایند تصمیم‌گیری مارکوف اشاره به این موضوع دارد که این سیستم‌ها دارای ویژگی مارکوف هستند، بدین معنا که تابع انتقال تنها به حالت فعلی سیستم و آخرین عمل وابسته است، و نسبت به حالت‌ها و اعمال قبل از آن مستقل است.

^{۱۳}Probability transition kernel

۳-۲ عناصر اصلی یادگیری تقویتی

۱-۳-۲ خط‌مشی

خط‌مشی^{۱۴} نحوه رفتار عامل یادگیری را در یک زمان خاص، مشخص می‌کند و هسته اصلی رفتار یک عامل یادگیری تقویتی است. خط‌مشی به تنهایی برای تعیین رفتار کافی است. به عنوان یک تعریف غیر دقیق، خط‌مشی، نگاشتی از حالت‌های مدل شده از محیط به اقداماتی است که باید در آن حالت انجام شود. خط‌مشی ممکن است یک عملکرد ساده یا جدول جستجو باشد، یا ممکن است شامل محاسبات پیچیده‌ای مانند فرآیند جستجو باشد؛ همچنین خط‌مشی‌ها ممکن است تصادفی باشند.

تعریف ۲-۲ (خط‌مشی احتمالاتی ثابت) یک خط‌مشی احتمالاتی ثابت^{۱۵} (یا به طور خلاصه خط‌مشی ثابت) $\pi : S \rightarrow \Pi(A)$ حالت‌ها را به توزیع احتمال روی فضای عمل می‌نگارد. به طور خلاصه احتمال انتخاب عمل a از حالت s را با $\pi(a|s)$ نشان می‌دهیم.

می‌گوییم خط‌مشی π در یک MDP دنبال می‌شود هرگاه

$$A_t \sim \pi(\cdot | X_t), \quad t \in \mathbb{N}.$$

۲-۳-۲ سیگنال پاداش

یک سیگنال پاداش هدف را در یک مسئله یادگیری تقویتی تعریف می‌کند. در هر گام، محیط یک عدد حقیقی به نام پاداش برای عامل یادگیری تقویتی ارسال می‌کند. تنها هدف عامل، به حداکثر رساندن کل پاداش دریافتی در طولانی زمان است. سیگنال پاداش اتفاقات خوب و بد برای عامل را مشخص می‌کند و مبنای اصلی تغییر خط‌مشی است. سیگنال پاداش می‌تواند تابعی تصادفی از وضعیت محیط و اقدام انجام شده باشد.

^{۱۴} Policy

^{۱۵} Stationary probabilistic policy

۲-۳-۳ عایدی و تابع ارزش

سیگنال پاداش نشان می‌دهد که انجام چه عملی در هر گام خوب است، در حالی که تابع ارزش مشخص می‌کند که کدام خط‌مشی در طولانی مدت بهتر است. در واقع تابع ارزش نشانگر مطلوبیت طولانی مدت حالت‌ها پس از در نظر گرفتن حالت‌هایی است که احتمالاً در پی خواهند داشت. ارزش حالت s ، مجموع میزان پاداشی است که عامل می‌تواند انتظار داشته باشد با شروع از s در آینده کسب کند. پاداش‌ها به یک معنا اولیه هستند، در حالی که ارزش‌ها، به عنوان پیش‌بینی پاداش‌ها، ثانویه هستند. بدون پاداش هیچ ارزشی وجود ندارد و تنها هدف تخمین ارزش‌ها، دستیابی به پاداش بیشتر است. با این وجود، این تابع ارزش است که هنگام تصمیم‌گیری و ارزیابی بیشتر به آن توجه می‌کنیم. تعیین ارزش بسیار دشوارتر از تعیین پاداش است ارزش‌ها باید از توالی مشاهداتی که یک عامل در طول عمر خود انجام می‌دهد، تخمین زده و مجدد برآورد شوند. مهمترین مولفه بیشتر الگوریتم‌های یادگیری تقویتی که در نظر می‌گیریم، روشی برای تخمین کارآمد تابع ارزش است.

تعریف ۲-۳ عایدی تخفیف دار آینده^{۱۶} یا به اختصار، عایدی، در زمان t به شکل

$$G_t \triangleq \sum_{t'=t}^T \gamma^{t'-t} R_{t'}$$

تعریف می‌شود که T زمانی است که اپیزود به اتمام می‌رسد. اگر مسئله مستمر باشد آنگاه $T = \infty$

از تعریف بالا نتیجه می‌شود

$$\begin{aligned} G_t &= R_t + \sum_{t'=t+1}^T \gamma^{t'-t} R_{t'} \\ &= R_t + \gamma \sum_{t'=t+1}^T \gamma^{t'-(t+1)} R_{t'} \\ &= R_t + \gamma G_{t+1}. \end{aligned} \quad (۱-۲)$$

تعریف ۲-۴ (تابع ارزش حالت) ارزش حالت s تحت خط‌مشی π یا $v_\pi(s)$ به شکل امید ریاضی عایدی، با شروع از s و دنبال کردن خط‌مشی π تعریف می‌شود.

$$v_\pi(s) = \mathbb{E}_\pi [G_t | S_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right]$$

^{۱۶} Future discounted Return

تابع v_π را تابع ارزش حالت ^{۱۷} مربوط به خط‌مشی π می‌نامیم.

تعریف ۲-۵ (تابع ارزش عمل) ارزش عمل a در حالت s تحت خط‌مشی π یا $q_\pi(s, a)$ به شکل امیدریاضی عایدی، با شروع از s و انتخاب عمل a و سپس دنبال کردن خط‌مشی π تعریف می‌شود.

$$q_\pi(s, a) = \mathbb{E}_\pi [G_t | S_t = s, A_t = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right]$$

تابع q_π را تابع ارزش عمل ^{۱۸} مربوط به خط‌مشی π می‌نامیم.

۲-۳-۴ محیط

مدل محیط رفتار محیط را تقلید می‌کند، یا به طور کلی‌تر، اجازه می‌دهد تا در مورد نحوه رفتار محیط، پیشبینی کارآمدی داشته‌باشیم. از مدل‌ها برای برنامه‌ریزی و انتخاب در روند تصمیم‌گیری با در نظر گرفتن شرایط احتمالی آینده بدون تجربه واقعی آنها استفاده می‌شود. روش‌هایی که برای حل مشکلات یادگیری تقویتی از مدل‌ها و برنامه‌ریزی‌ها استفاده می‌کنند، روش‌های مبتنی بر مدل نامیده می‌شوند. این روش‌ها در مقابل روش‌های بدون مدل هستند که همگی آزمون و خطا هستند.

۲-۴ برنامه‌ریزی پویا

۲-۴-۱ خط‌مشی و تابع ارزش بهینه

برای MDP های متناهی، می‌توانیم خط‌مشی بهینه ^{۱۹} را به صورت زیر تعریف کنیم

تعریف ۲-۶ می‌گوییم خط‌مشی π بهتر از خط‌مشی π' است و می‌نویسیم $\pi \geq \pi'$ هرگاه برای هر $s \in S$

$$v_\pi(s) \geq v_{\pi'}(s).$$

می‌توان نشان داد که حداقل یک خط‌مشی وجود دارد که بهتر یا مساوی هر خط‌مشی دیگری باشد

^{۱۷} State Value Function

^{۱۸} Action Value Function

^{۱۹} Optimal policy

[۱]. به چنین خط‌مشی‌ای **خط‌مشی بهینه** گفته می‌شود. ممکن است بیش از یک خط‌مشی بهینه وجود داشته باشد ولی تابع ارزش متناظر با همه خط‌مشی‌های بهینه یکسان است و برابر با تابع ارزش بهینه است که با نماد v_* نمایش داده شده و به شکل زیر تعریف می‌شود. برای هر $s \in S$

$$v_*(s) \triangleq \max_{\pi} v_{\pi}(s)$$

همچنین تمام خط‌مشی‌های بهینه تابع عمل-ارزش مشترکی دارند که آن را با نماد q_* نمایش می‌دهیم و به شکل زیر تعریف می‌شود. برای هر دوتایی حالت عمل (s, a) که $s \in S$ و $a \in A(s)$

$$q_*(s, a) \triangleq \max_{\pi} q_{\pi}(s, a)$$

می‌توانیم q_* را برحسب v_* به شکل زیر بنویسیم

$$q_*(s, a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \quad (2-2)$$

۲-۴-۲ معادله بلمن

ویژگی اساسی توابع ارزش که در طول یادگیری تقویتی و برنامه‌ریزی پویا استفاده می‌شوند صدق کردن در روابط بازگشتی است. معادلات بلمن رابطه‌ای بین ارزش یک حالت و ارزش‌های حالت‌های بعدی آن را بیان می‌کند

$$\begin{aligned} v_{\pi}(s) &\triangleq \mathbb{E}_{\pi}[G_t | S_t = s] \\ &= \mathbb{E}_{\pi}[R_{t+1} + \gamma G_{t+1} | S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) [r + \gamma \mathbb{E}_{\pi}[G_{t+1} | S_{t+1} = s']] \\ &= \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma v_{\pi}(s')] \quad \forall s \in (S) \end{aligned} \quad (3-2)$$

۲-۴-۳ بهینگی و معادله بهینگی بلمن

$$\begin{aligned}
v_*(s) &= \max_{a \in \mathbb{A}(s)} q_{\pi_*}(s, a) \\
&= \max_a \mathbb{E}_{\pi_*}[G_t | S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \\
&= \max_a \sum_{s', r} p(s', r | s, a) [r + \gamma v_*(s')]. \tag{۴-۲}
\end{aligned}$$

$$\begin{aligned}
q_*(s, a) &= \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') | S_t = s, A_t = a] \\
&= \sum_{s', r} p(s', r | s, a) [r + \gamma \max_{a'} q_*(s', a')]
\end{aligned}$$

۲-۴-۴ برنامه‌ریزی پویا

برنامه‌ریزی پویا^{۲۰} مجموعه‌ای از الگوریتم‌ها است که برای محاسبه خط‌مشی بهینه استفاده می‌شود. الگوریتم‌های کلاسیک برنامه‌ریزی پویا به دلیل فرض مدل کاملی از محیط و همچنین هزینه محاسباتی زیادشان، به لحاظ عملی چندان قابل استفاده نیستند اما به لحاظ نظری مهم هستند.

ایده اصلی DP و به طور کلی یادگیری تقویتی، استفاده از تابع ارزش حالت یا عمل برای سازماندهی یک الگوریتم جستجو برای خط‌مشی بهینه است. هدف، تخمین تابع ارزش بهینه، v_* یا q_* است که در معادلات بهینگی بلمن صدق می‌کند:

$$v_*(s) = \max_a \sum_{s', r} p(s', r | s, a) [r + \gamma v_*(s')] \tag{۵-۲}$$

$$q_*(s, a) = \sum_{s', r} p(s', r | s, a) [r + \gamma \max_{a'} q_*(s', a')] \tag{۶-۲}$$

اگر دینامیک محیط کاملاً مشخص باشد معادلات ۲-۵ و ۲-۶ به ترتیب یک دستگاه معادلات خطی با $|S|$ است که راه‌حل آن سراسر است. برای اهداف ما، روش‌های تکراری مناسب‌ترین روش‌ها هستند.

^{۲۰}Dynamic Programing(DP)

یک دنباله از توابع ارزش تقریبی v_0, v_1, v_2, \dots را در نظر بگیرید که هرکدام نگاشتی از S به \mathbb{R} هستند. تقریب اولیه v_0 به طور دلخواه انتخاب می‌شود (به جز در حالت های پایانی که باید صفر باشد). هر تقریب موفق به عنوان یک قانون به روز رسانی به وسیله معادله بلمن برای v_π بدست می‌آید.

$$\begin{aligned} v_{k+1}(s) &\triangleq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) | S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma v_k(s')] \end{aligned} \quad (۷-۲)$$

برای هر $s \in S$. بر اساس معادله بلمن می‌توان درستی این تساوی را برای v_π نوشت و واضح است که $v_k = v_\pi$ نقطه ثابتی برای این قانون بروز رسانی است. در واقع، می‌توان نشان داد که در حالت کلی دنباله $\{v_k\}$ وفتی که $k \rightarrow \infty$ و تحت همان شرایط که وجود v_π را تضمین می‌کند، به مقدار v_π همگرا می‌شود. این الگوریتم را ارزیابی خط‌مشی تکراری^{۲۱} می‌نامند.

۲-۴-۵ بهبود خط‌مشی

دلیل ما برای محاسبه تابع ارزش یک خط‌مشی کمک به یافتن خط‌مشی‌های بهتر است. فرض کنید ما تابع ارزش v_π را برای یک خط‌مشی تعیین‌گرایانه دلخواه π تعیین کرده‌ایم. برای یک حالت s ما می‌خواهیم بدانیم که آیا باید خط‌مشی را برای انتخاب قطعی یک عمل a تغییر دهیم یا خیر. ما می‌دانیم که پیروی از خط‌مشی فعلی از حالت s چقدر خوب است، اما آیا تغییر به سیاست جدید بهتر است یا بدتر؟ یکی از راه‌های پاسخ به این سوال در نظر گرفتن انتخاب عمل a در حالت s و پس از آن پیروی از خط‌مشی موجود (π) است. ارزش این شیوه رفتار این است که ؟؟؟؟؟

قضیه ۱-۲ (قضیه بهبود خط‌مشی) فرض کنید π و π' دو خط‌مشی معین باشند که برای هر $s \in S$

$$q_\pi(s, \pi'(s)) \geq \pi'(s)$$

$$v_{\pi'}(s) \geq v_\pi(s) \text{ در این صورت}$$

درستی قضیه بالا را می‌توان با استفاده از تعاریف به روشنی بررسی کرد

$$\begin{aligned}
v_{\pi}(s) &\leq q_{\pi}(s, \pi'(s)) = \mathbb{E} [R_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = s, A_t = \pi'(a)] \\
&= \mathbb{E}_{\pi'} [R_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = s] \\
&\leq \mathbb{E}_{\pi'} [R_{t+1} + \gamma q_{\pi}(S_{t+1}, \pi'(S_{t+1})) | S_t = s] \\
&= \mathbb{E}_{\pi'} [R_{t+1} + \gamma \mathbb{E}_{\pi'} [R_{t+2} + \gamma v_{\pi}(S_{t+2}) | S_t = s]] \\
&\leq \mathbb{E}_{\pi'} [R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_{\pi}(S_{t+3}) | S_t = s] \\
&\vdots \\
&\leq \mathbb{E}_{\pi'} [R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots | S_t = s] \\
&\leq v_{\pi'}(s)
\end{aligned}$$

۲-۴-۶ الگوریتم خط‌مشی تکراری

در الگوریتم خط‌مشی تکراری^{۲۲} هنگامی که یک خط‌مشی π با استفاده از v_{π} بهبود یافته است تا به خط‌مشی بهتری π_0 برسیم، سپس می‌توانیم v_{π} را محاسبه کرده و مجدداً آن را بهبود بخشیم تا $\pi_{0.0}$ بهتر داشته باشیم. بنابراین ما می‌توانیم دنباله‌ای از خط‌مشی‌ها و تابع ارزش‌هایی بدست آوریم که به صورت یکنوا در حال بهبود هستند:

$$\pi_0 \longrightarrow v_{\pi_0} \longrightarrow \pi_1 \longrightarrow v_{\pi_1} \longrightarrow \pi_2 \longrightarrow \dots \longrightarrow \pi_* \longrightarrow v_*$$

۲-۴-۷ الگوریتم Iteration Value

یک اشکال در تکرار خط‌مشی این است که هر یک از تکرارهای آن شامل ارزیابی سیاست است که ممکن است باشد به خودی خود یک محاسبه تکراری طولانی مدت است که نیاز به رفت و برگشت‌های متعدد از طریق مجموعه حالت دارد. اگر سیاست ارزیابی بصورت تکراری انجام می‌شود، سپس همگرایی دقیقاً به v_{π} فقط در حد مجاز اتفاق می‌افتد. باید منتظر بمانیم برای همگرایی دقیق، یا می‌توانیم کوتاه‌تر از آن متوقف شویم؟ مثالی که در شکل ۱۰۴ وجود دارد قطعاً این را نشان می‌دهد کوتاه کردن ارزیابی

^{۲۲} Policy iteration

سیاست ممکن است باشد. در آن مثال، تکرار ارزیابی سیاست فراتر از آن است سه مورد اول هیچ تاثیری در سیاست حریص مربوطه ندارند. در واقع، مرحله ارزیابی سیاست تکرار سیاست می تواند از چند طریق بدون ضرر کوتاه شود همگرایی تضمین سیاست را تضمین می کند. یک مورد خاص مهم، ارزیابی سیاسی است فقط پس از یک بار جابجایی متوقف می شود (یک به روزرسانی از هر حالت). این الگوریتم تکرار مقدار نامیده می شود. آی تی را می توان به عنوان یک عملیات به روزرسانی ساده نوشت که ترکیبی از بهبود سیاست و مراحل ارزیابی سیاست کوتاه شده: $\text{policy to drawback One}$ itself may which evaluation, policy involves iterations its of each that is iteration state the through sweeps multiple requiring computation iterative protracted a be occurs $\forall \pi$ to exactly convergence then iteratively, done is evaluation policy If set. of short stop we can or convergence, exact for wait we Must limit. the in only to possible be may it that suggests certainly ۱.۴ Figure in example The that? beyond iterations evaluation policy example, that In evaluation. policy truncate the fact, In policy. greedy corresponding the on effect no have three first the without ways several in truncated be can iteration policy of step evaluation policy case special important One iteration. policy of guarantees convergence the losing each of update (one sweep one just after stopped is evaluation policy when is particularly a as written be can It iteration. value called is algorithm This state). truncated and improvement policy the combines that operation update simple steps: evaluation policy

to converge to shown be can v_k sequence the v_0 , arbitrary For S . $\forall s$ all for $v[s]$. of existence the guarantee that conditions same the under $v[s]$

۲-۴-۸ درهم تنیدگی PE و PI و الگوریتم های GPI

in- they that is far so discussed have we that methods DP the to drawback major A sweeps require they is, that MDP, the of set state entire the over operations volve pro- be can sweep single a even then large, very is set state the If set. state the of

expensive hibitively

not are that algorithms DP iterative in-place are algorithms DP Asynchronous
update algorithms These set. state the of sweeps systematic of terms in organized
states other of values whatever using whatsoever, order any in states of values the
times several updated be may states some of values The available. be to happen
however, correctly, converge To once. updated are others of values the before
states: the all of values the update to continue must algorithm asynchronous an
DP Asynchronous computation. the in point some after state any ignore can't it
update. to states selecting in flexibility great allow algorithms

۵-۲ Q-learning

ایده اصلی در روش Q-learning، تخمین تابع مقدار عمل $Q^*(s, a)$ با استفاده از معادله بلمن به عنوان یک به روزرسانی تکراری،

$$Q_{i+1}(s, a) = \mathbb{E}[r + \gamma \max_{a'} Q_i(s', a') | s, a]$$

چنین الگوریتم های تکرار مقداری به تابع عمل-ارزش بهینه همگرا می شوند، $Q_i \rightarrow Q^*$ وقتی $i \rightarrow \infty$
در عمل، این رویکرد کلی کاملاً غیر عملی است، زیرا تابع عمل-ارزش برای هر دنباله، به طور
جداگانه و بدون هیچ گونه تعمیم برآورد می شود. در عوض، معمولاً از یک تخمین گر توابع (مثل شبکه
عصبی) برای تخمین تابع عمل-ارزش استفاده می شود، در فصل سوم با این روش بیشتر آشنا خواهیم
شد.

فصل ۳

نتایج اخیر

۳-۱ روش‌های مبتنی بر مدل و بدون مدل

یکی از مهمترین نقاط انشعاب در الگوریتم RL این است که آیا عامل به یک مدل از محیط دسترسی دارد یا توانایی آموختن مدلی از محیط را دارد؟ منظور از مدل محیط، تابعی است که انتقال و پاداش هر حالت-عمل را پیش‌بینی می‌کند.

نقطه قوت داشتن مدل این است که به عامل اجازه می‌دهد با برآورد قبلی، طیف وسیعی از گزینه‌های ممکن را پیش‌بینی کند و به صراحت در مورد گزینه‌های خود تصمیم بگیرد. سپس عامل می‌تواند نتایج حاصل از برنامه ریزی قبلی را در قالب یک خط‌مشی بیاموزد یک نمونه مشهور از این روش AlphaZero است. در عمل، چنانچه دستیابی به مدلی از محیط امکان‌پذیر و عملی باشد، معمولاً از روش‌های مبتنی بر مدل استفاده می‌شود. زیرا می‌تواند باعث بهبود قابل توجه‌ای در کارایی نمونه نسبت به روش‌های بدون مدل شود.

اصلی‌ترین نقطه ضعف این روش‌ها این است که معمولاً یک مدل کامل از محیط در دسترس عامل نیست و مدل کاملاً از طریق تجربه یاد گرفته می‌شود.

الگوریتم‌هایی که از یک مدل استفاده می‌کنند، روش‌های مبتنی بر مدل و آنهایی که از چنین مدلی استفاده نمی‌کنند، بدون مدل نامیده می‌شوند. روش‌های بدون مدل از دستاوردهای بالقوه روش‌های مبتنی بر مدل چشم‌پوشی می‌کنند، اما پیاده‌سازی و تنظیم آنها آسان‌تر است. به همین خاطر، روش‌های

بدون مدل از محبوبیت بیشتری برخوردار بوده و به طور گسترده‌تری توسعه و آزمایش شده‌اند.

۳-۲ روش‌های بدون مدل

۳-۲-۱ روش‌های مبتنی بر ارزش

در روش‌های یادگیری تقویتی بدون مدل مبتنی بر ارزش تابع ارزش عمل با استفاده از یک تخمین‌گیر تابع ^۱، مانند شبکه عصبی، نشان داده می‌شود. فرض کنید $Q(s, a; \theta)$ یک تابع ارزش عمل تقریبی با پارامتر θ باشد الگوریتم‌های مختلفی برای بروزرسانی θ وجود دارد الگوریتم $Q-learning$ یکی از نمونه‌های چنین الگوریتمی است که هدف آن تقریب مستقیم تابع ارزش عمل-ارزش بهینه $Q^*(s, a) \approx Q(s, a; \theta)$ است. در $Q-learning$ یک مرحله‌ای، پارامترهای θ از تابع ارزش عمل-ارزش با به حداقل رساندن تابع هزینه به شکل مرحله به مرحله آموخته می‌شوند، به شکلی که تابع هزینه i ام به شکل

$$L_i(\theta_i) = \mathbb{E} \left(r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i) \right)^2$$

تعریف می‌شود که s' حالتی است که بعد از حالت s دیده می‌شود.

روش‌های Q-learning

خانواده روش‌های Q-learning تلاش می‌کنند مستقیماً تابع ارزش عمل-حالت بهینه $Q^*(s, a)$ را تخمین بزنند. آن‌ها به طور معمول از یک تابع هدف مبتنی بر معادله بلمن استفاده می‌کنند. این بهینه‌سازی تقریباً همیشه به صورت off-policy انجام می‌شود، به این معنی که هر بروزرسانی می‌تواند از داده‌های جمع‌آوری شده در هر نقطه استفاده کند، بدون در نظر گرفتن نحوه انتخاب عامل برای کشف محیط در هنگام بدست آوردن داده‌ها. خط‌مشی مربوطه از طریق ارتباط بین Q^* و π^* بدست می‌آید: عامل بعد از یادگرفتن تابع $Q_\theta(s, a)$ به طوریکه $Q_\theta(s, a) \approx Q^*(s, a)$ می‌تواند عمل بهینه در حالت s را به صورت زیر محاسبه کند

$$a(s) = \arg \max_a Q_\theta(s, a).$$

¹Function approximator

از جمله الگوریتم‌های Q-learning می‌توان به موارد زیر اشاره کرد:

- روش کلاسیک DQN که حوزه یادگیری تقویتی ژرف^۲ را عمیقاً ارتقا بخشید.
- روش C5۱ که توزیعی روی عایدی را می‌آموزد که امیدریاضی آن Q^* است.

روش DQN

معمولاً برای تقریب زدن توابع ارزش در یادگیری تقویتی، از یک تابع خطی استفاده می‌شود. اما گاهی اوقات از یک تقریب عملکرد غیر خطی به جای آن، مانند یک شبکه عصبی استفاده می‌شود. شبکه‌های عصبی با عنوان شبکه Q ^۳ شناخته می‌شوند. شبکه Q را می‌توان با کمینه ساختن دنباله‌ای از توابع هزینه

به شکل $L_1(\theta_1), L_2(\theta_2), L_3(\theta_3), \dots$

آموزش داد؛ به طوری که

$$L_i(\theta_i) = \mathbb{E} \left[(y_i - Q(s, a; \theta_i))^2 \right]$$

که

$$y_i = \mathbb{E}[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a]$$

با مشتق گرفتن از تابع هزینه نسبت به پارامترهای θ_i خواهیم داشت

$$\nabla_{\theta_i} L_i \theta_i = \mathbb{E} \left[(r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i)) \nabla_{\theta_i} Q(s, a; \theta_i) \right]$$

به جای محاسبه امیدریاضی کامل در گرادیان فوق، غالباً از نظر محاسباتی، بهینه سازی تابع هزینه با نزول گرادیان تصادفی^۴ راه حل بهتری است. اگر در هر مقطع زمانی، وزن‌ها به روزرسانی شود و

^۲Deep reinforcement learning

^۳Q-Network

^۴stochastic gradient descend

امیدریاضی با یک نمونه از توزیع خط‌مشی رفتار^۵ جایگزین شود، الگوریتم Q-learning به شکل زیر خواهد بود.

الگوریتم ۱ الگوریتم Q-learning با replay Experience

- ۱: حافظه D replay را مقدار دهی اولیه کن
- ۲: تابع عمل-ارزش Q را با وزن‌های تصادفی مقداردهی اولیه کن
- ۳: برای هر اپیزود $1 \dots M$:
- ۴: دنباله $d_1 = \{S_1\}$ و کدینگ $\phi_1 = \phi(d_1)$ را مقداردهی اولیه کن
- ۵: برای $t = 1 \dots T$:
- ۶: با احتمال ϵ یک عمل تصادفی a_t را انتخاب کن، در غیر این صورت $a_t = \max_a Q^*(\phi(d_t), a; \theta)$ را انتخاب کن
- ۷: عمل a_t را انجام بده و حالت S_{t+1} و پاداش R_t را مشاهده کن
- ۸: قرار بده $d_{t+1} = d_t, a_t, S_{t+1}$ و $\phi_{t+1} = \phi(d_{t+1})$
- ۹: تجربه $(\phi_t, A_t, R_t, \phi_{t+1})$ را در D ذخیره کن
- ۱۰: یک نمونه تصادفی از تجربه‌های $(\phi(j), A_j, R_j, \phi_{j+1})$ از انبار تجربیات D انتخاب کن
- ۱۱: قرار بده
$$y_j = \begin{cases} r_j & \phi_{j+1} \text{ terminal} \\ r_j & \text{otherwise} \end{cases}$$
- ۱۲: یک گام از نزول گرادیان را برای تابع هزینه $(y_j - Q(\phi_j, a_j; \theta))^2$ انجام بده

توجه داشته باشید که این یک الگوریتم بدون مدل است: این کار وظیفه یادگیری تقویتی را مستقیماً با استفاده از نمونه‌های شبیه ساز E بدون ساختن صریح تخمین E حل می‌کند. در مورد استراتژی حریص $a: (s, Q \max_a = a)$ ، یاد می‌گیرد، در حالی که توزیع رفتاری را دنبال می‌کند که کاوش کافی در فضای دولت را تضمین می‌کند. در عمل، توزیع رفتار اغلب توسط یک استراتژی Greedy انتخاب می‌شود که استراتژی حریصانه را با احتمال ۱ دنبال می‌کند و یک اقدام تصادفی با احتمال ϵ

⁵behavior policy

روش C۵۱

روش DDPG

۲-۲-۳ روش‌های مبتنی بر خط‌مشی

در این قسمت روش‌هایی را در نظر می‌گیریم که به جای استفاده از تابع عمل-ارزش یا حالت-ارزش برای دستیابی به خط‌مشی بهینه، یک خط‌مشی پارامتریزه^۶ شده را می‌آموزد که می‌تواند اقدامات را بدون استفاده از یک تابع ارزش انتخاب کند. یک تابع ارزش ممکن است همچنان برای یادگیری پارامترهای خط‌مشی استفاده شود، اما برای انتخاب عمل مورد نیاز نیست. ما از نماد $\theta \in \mathbb{R}^{d'}$ برای بردار پارامتر خط‌مشی استفاده می‌کنیم. برخلاف روش‌های مبتنی بر ارزش، روش‌های مبتنی بر خط‌مشی مستقیماً تابع خط‌مشی $\pi_\theta(a|s)$ را تخمین می‌زنند و پارامترهای θ را با استفاده از صعود گرادیان^۷ روی یک مقیاس عملکرد^۸ $J(\pi_\theta)$ یا به طور مستقیم و یا با بیشینه‌سازی تخمین‌های محلی از $J(\pi_\theta)$ بروزرسانی می‌کند. این روش تقریباً همیشه به صورت on-policy عمل می‌کند.

همانطور که در ادامه خواهیم دید، می‌توان از توابع مختلفی برای مقیاس عملکرد J استفاده نمود. یک انتخاب بدیهی $J(\pi_\theta) = \mathbb{E}[R_t]$ است. هدف این روش‌ها بیشینه کردن تابع J است. در عبارت

$$\theta_{t+1} = \theta_t + \alpha \nabla \hat{J}(\theta_t)$$

$\nabla \hat{J}(\theta_t)$ ، تخمینی احتمالاتی است که امیدریاضی آن گرادیان مقیاس عملکرد J را نسبت به پارامترهای خط‌مشی θ_t تخمین می‌زنند.

به روش‌هایی که چنین الگویی را برای محاسبه خط‌مشی بهینه دنبال می‌کنند، روش‌های گرادیان خط‌مشی^۹ می‌گوییم. دسته‌ای از روش‌های گرادیان خط‌مشی وجود دارند که تلاش می‌کنند تخمینی از تابع ارزش را نیز محاسبه کنند. به چنین روش‌هایی، بازیگر-منتقد (Actor-Critic) گفته می‌شود که بازیگر (Actor) اشاره به خط‌مشی آموخته شده و منتقد (Critic) اشاره به تابع ارزش آموخته شده (معمولاً یک تابع حالت-ارزش) دارد.

^۶Parameterized^۷Gradient ascent^۸Performance Measure^۹Policy Gradient

الگوریتم ۲ الگوریتم DDPG

۱: پارامترهای θ_μ و θ_Q به ترتیب مربوط به بازیگر $\mu(s; \theta_\mu)$ و منتقد $Q(s, a; \theta_Q)$ را مقداردهی اولیه کن.

۲: پارامترهای توابع هدف μ' و Q' را با وزنهای $\theta_\mu \longleftrightarrow \theta_{\mu'}$ و $\theta_Q \longleftarrow \theta_{Q'}$ مقداردهی اولیه کن

۳: حافظه تکرارها R را بساز برای هر اپیزود $1 \dots M$

۴: یک تابع نویز تصادفی \mathbb{N} بساز

۵: حالت اولیه S_1 را مشاهده کن برای $t = 1 \dots T$

۶: عمل $a_t = \mu(s_t; \theta_\mu) + \mathbb{N}_t$ را بر اساس خطمشی فعلی و نویز اکتشاف، انتخاب کن و حالت بعدی S_{t+1} و پاداش R_t را مشاهده کن.

۷: تجربه (s_t, a_t, r_t, s_{t+1}) را در انبار تجربه R ذخیره کن

۸: یک نمونه به اندازه N از تجربههای (s_i, a_i, r_i, s_{i+1}) از انبار تجربه R انتخاب کن

۹: وزنهای منتقد θ_Q را با در نظر گرفتن تابع هزینه $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i; \theta_Q))^2$ به روزرسانی کن

۱۰: وزنهای بازیگر θ را با استفاده از گرادیان خطمشی نمونه

$$\nabla_{\theta_\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a; \theta_Q)|_{s=s_i, a=\mu(s_i)} \nabla_\mu \mu(s; \theta_\mu)|_{S_i}$$

به روزرسانی کن

۱۱: وزنهای توابع هدف را به شکل

$$\theta_{Q'} = \tau \theta_Q + (1 - \tau) \theta_{Q'} \theta_{\mu'} = \tau \theta_\mu + (1 - \tau) \theta_{\mu'}$$

به روزرسانی کن

۱۳:

۱۴:

قضیه ۱-۳ (گرادیان خط‌مشی)

$$\nabla J(\pi_\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s, a) \nabla_\theta \pi_\theta(a|s)$$

که μ یک توزیع احتمال روی S است که متناسب با تعداد دفعاتی است که حالت s با دنبال کردن خط‌مشی π_θ تکرار می‌شود.

می‌توان نشان داد [۱]

$$\nabla J(\pi_\theta) = \mathbb{E}_\pi \left[R_t \frac{\nabla_\theta \pi_\theta(a|S_t)}{\pi_\theta(a|S_t)} \right]$$

بنابراین در هر گام $\left[R_t \frac{\nabla_\theta \pi_\theta(a|S_t)}{\pi_\theta(a|S_t)} \right]$ یک تخمین‌گر نااریب از $\nabla J(\pi_\theta)$ خواهد بود. پس می‌توان در هر گام θ را به شکل زیر بروزرسانی کرد

$$\theta_{t+1} = \theta_t + \alpha R_t \frac{\nabla_\theta \pi_\theta(a|S_t)}{\pi_\theta(a|S_t)} = \theta_t + \alpha R_t \nabla_\theta \log \pi_\theta(a|S_t)$$

چند نمونه از روش‌های بهینه‌سازی خط‌مشی به شرح زیر است.
روش‌های Actor-Critic که الگوریتم ascent Gradient را مستقیماً برای بیشینه‌سازی $J(\pi_\theta)$ به کار می‌برند.

روش Optimization Policy Proximal که!!!!!!!

روش‌های Actor-Critic

یک نمونه از روش‌های بهینه‌سازی خط‌مشی، خانواده REINFORCE از الگوریتم‌های یادگیری تقویتی است [۲]. الگوریتم استاندارد REINFORCE پارامترهای θ را در جهت $\nabla_\theta \log \pi(a_t|s_t)$: R_t بروزرسانی می‌کند که یک تخمین نااریب از $\nabla_\theta \mathbb{E}[R_t]$ است. می‌توان با تفریق یک تابع آموخته شده روی حالت‌ها، $b_t(s_t)$ از R_t ، واریانس این تخمین را کاهش داد بطوری‌که نااریب باقی بماند. به چنین تابعی پایه گفته می‌شود. در نتیجه گرادیان به شکل $\nabla_\theta \log \pi(a_t|s_t; \theta)(R_t - b_t(s_t))$ خواهد بود. معمولاً از تخمینی آموخته‌شده از تابع ارزش به عنوان پایه استفاده می‌شود، $b_t(s_t) \approx v_\pi(s_t)$ ، که منجر به تخمینی با واریانس بسیار کوچک‌تر از گرادیان خط‌مشی می‌شود؛ در حالیکه تخمین نااریب

باقی می ماند و در نتیجه عملیات یادگیری با سرعت بیشتری انجام می شود. این روش می تواند به شکل معماری بازیگر-منتقد^{۱۰} تعبیر شود که خط مشی π بازیگر و پایه b_t منتقد است. اگر از تخمین یک تابع حالت-ارزش به عنوان پایه استفاده کنیم $b_t(s) = v_{\pi_{\theta_t}}(s)$ عبارت $R_t - b_t(s)$ می تواند به شکل تخمینی از مزیت^{۱۱} عمل a_t در حالت s_t یا $A(a_t, s_t) = Q(a_t, s_t) - v(s_t)$ تعبیر شود. چراکه R_t تخمینی از $Q_{\pi}(a_t, s_t)$ و b_t تخمینی از $v_{\pi}(s_t)$ است. در این صورت به این روش Actor-Critic Advantage یا A۲C گفته می شود.

الگوریتم ۳ الگوریتم Advantage Actor – Critic [۱]

- ورودی: یک پارامتریزه سازی مشتق پذیر از خط مشی $\pi_{\theta}(a|s)$
- ورودی: یک پارامتریزه سازی مشتق پذیر از تابع حالت-ارزش $v_{\omega}(s)$
- ۱: پارامترهای خط مشی $\theta \in \mathbb{R}^{d'}$ و تابع حالت-ارزش $\omega \in \mathbb{R}^d$ را مقداردهی اولیه کن
 - ۲: تکرار کن:
 - ۳: حالت اولیه S را بساز
 - ۴: $I \rightarrow 1$
 - ۵: تا وقتی S حالت نهایی نیست:
 - ۶: $A \sim \pi_{\theta}(\cdot|S)$
 - ۷: عمل A را انجام بده و حالت S' و پاداش R را مشاهده کن
 - ۸: $R + \gamma v_{\omega}(S') - v_{\omega}(S) \rightarrow \delta$
 - ۹: $\omega + \alpha^{\omega} I \delta \nabla_{\omega} v_{\omega}(S) \rightarrow \omega$
 - ۱۰: $\theta + \alpha^{\theta} I \delta \nabla_{\theta} \ln \pi_{\theta}(A|S) \rightarrow \theta$
 - ۱۱: $\gamma I \rightarrow I$
 - ۱۲: $S' \rightarrow S$

¹⁰ Actor-Critic¹¹ Advantage

روش TRPO

تعریف ۳-۱ اگر $\rho_\pi : S \rightarrow \mathbb{R}$ تابع فرکانس تخفیف‌دار دیده شدن حالت‌ها باشد. یعنی

$$\rho_\pi(s) = P(S_0 = s) + \gamma P(S_1 = s) + \gamma^2 P(S_2 = s) + \dots$$

که دنباله S_0, S_1, S_2 خط‌مشی π را دنبال می‌کند.

اگر π و π' دو خط‌مشی باشند و $J(\pi) = \mathbb{E}_\pi[R_0]$ می‌توان نشان داد [۳]

$$J(\pi') = J(\pi) + \sum_s \rho_{\pi'}(s) \sum_a \pi'(a|s) A_\pi(s, a)$$

که $A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s)$ تابع مزیت عمل a در حالت s باشد. وابستگی پیچیده $\rho'_\pi(s)$ در طرف راست تساوی به π' بهینه‌سازی مستقیم را مشکل می‌کند. برای حل این مشکل شولمن و همکارانش مقیاس عملکرد دیگری، $L_\pi(\pi')$ ، را معرفی می‌کند و نشان می‌دهد که اگر π و π' به اندازه کافی به یکدیگر نزدیک باشند، افزایش $L_\pi(\pi')$ همواره با افزایش J همراه خواهد بود [۳].!!!!!!!!!!!!!!!!!!!!!!

$$L_\pi(\pi') = J(\pi) + \sum_s \rho_\pi(s) \sum_a \pi'(a|s) A_\pi(s, a)$$

در عبارت بالا L_π از تابع فرکانس ρ_π به جای $\rho_{\pi'}$ استفاده می‌کند.

قضیه ۳-۲ فرض کنید $\alpha = D_{TV}^m ax(\pi_{old}, \pi_{new})$ باشد که

$$D_{TV}^m ax(\pi, \pi') = \max_s D_{TV}(\pi(\cdot|s) || \pi'(\cdot|s))$$

و $D_{TV}(p||q)$ دیورژانس *total variation* بین دو بردار p و q باشد

$$D_{TV}(p||q) = \frac{1}{2} \sum_i |p_i - q_i|$$

در این صورت

$$J(\pi_{new}) \geq L_{\pi_{old}}(\pi_{new}) - \frac{\epsilon \gamma}{(1 - \gamma)^2}$$

که $\epsilon = \max_{s,a} |A_\pi(s, a)|$

با توجه به قضیه فوق و نامعادله $D_{TV}(p||q)^2 \leq D_{KL}(p||q)$ [۳] که $D_{KL}(p||q)$ برابر با دیورژانس KL دو بردار p و q است. می توان نتیجه گرفت

$$J(\pi') \geq L_{\pi}(\pi') - C D_{KL}^m ax(\pi, \pi')$$

که

$$C = \frac{2\epsilon\gamma}{(1-\gamma)^2}$$

رابطه بالا نشان می دهد که می توان یک دنباله صعودی از خطمشی ها داشت به طوری که

$$J(\pi_0) \leq J(\pi_1) \leq J(\pi_2) \leq \dots$$

. چرا که فرض کنید $M_i(\pi) = L_{\pi_i}(\pi) - C D_{KL}^m ax(\pi_i, \pi)$ در این صورت

$$J(\pi_{i+1}) \geq M_i(\pi_{i+1})$$

$$J(\pi_i) = M_i(\pi_i)$$

بنابراین

$$J(\pi_{i+1}) - J(\pi_i) \geq M_i(\pi_{i+1}) - M_i(\pi_i)$$

می توان نتیجه گرفت با بیشینه کردن M_i در هر گام می توان اطمینان حاصل کرد که مقیاس عملکرد واقعی J غیرنزولی خواهد بود.

اگر \hat{A}_t تخمین مزیت $A_{\pi_t}(S_t, A_t)$ باشد که در گام t محاسبه می شود، می توان نشان داد که در روش TRPO مقیاس عملکرد L_{π} در هر گام به شکل

$$\mathbb{E}_t \left[\frac{\pi_{\theta_{new}}(A_t|S_t)}{\pi_{\theta_{old}}(A_t|S_t)} \hat{A}_t \right]$$

خواهد بود.

روش PPO

در روش TRPO دیدیم که بیشینه سازی مقیاس عملکرد L_{π} ساده تر از مقیاس عملکرد J است ولی در عوض الگوریتم صعود گرادیان تنها مجاز به اعمال تغییرات کوچک در خطمشی است. یک راه دیگر

الگوریتم ۴ *PolicyIteration* با مقیاس عملکرد L_π

- ۱: خط‌مشی π_0 را مقداردهی اولیه کن
 - ۲: برای $i = 0, 1, \dots$ تکرار کن:
 - ۳: همه مزیت‌های $A_{\pi_i}(s, a)$ را محاسبه کن
 - ۴: $\arg \max_{\pi} [L_{\pi_i}(\pi) - C D_{KL}^m ax(\pi_i, \pi)] \rightarrow \pi_{i+1} =$ که
- $$C = (4\epsilon\gamma)/(1 - \gamma)^2$$

و

$$L_{\pi_i}(\pi) = J(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$$

برای کنترل تغییرات خط‌مشی استفاده از تابع clip است. در روش TRPO دیدیم که تابع مقیاس عملکرد، در گام t به شکل زیر است

$$L_{\pi_{old}}(\pi_{new}) = \mathbb{E}_{\approx} \left[\frac{\pi_{\theta_{new}}(A_t|S_t)}{\pi_{\theta_{old}}(A_t|S_t)} A_{\pi}(S_t, A_t) \right]$$

فرض کنید $r_t(\theta_{new})$ نسبت احتمالات $\frac{\pi_{\theta_{new}}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ باشد. بنابراین

$$L_{\pi_{old}}(\pi_{new}) = \mathbb{E}_t \left[\frac{\pi_{\theta_{new}}(A_t|S_t)}{\pi_{\theta_{old}}(A_t|S_t)} A_{\pi}(S_t, A_t) \right] = \mathbb{E}_t [r_t(\theta_{new}) A_{\pi_{old}}(S_t, A_t)]$$

. مقیاس عملکرد $L^{CLIP}(\theta)$ را به شکل زیر تعریف می‌کنیم

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t) \right]$$

که ϵ یک ابرپارامتر^{۱۲} مثلاً $\epsilon = 0.2$ است. اولین عبارت داخل min همان مقیاس عملکرد روش TRPO است. در عبارت دوم مقادیر بزرگتر از $1 + \epsilon$ یا کوچکتر از $1 - \epsilon$ در $r_t(\theta)$ به ترتیب به $1 + \epsilon$ و $1 - \epsilon$ تغییر پیدا کرده اند تا تغییرات بزرگ خط‌مشی را کنترل کنند. نهایتاً عبارت کوچکتر از میان این دو انتخاب خواهد شد. بنابراین اگر مقیاس عملکرد clip نشده (عبارت اول) کوچکتر یا مساوی با

¹²Hyperparameter

حالت clip شده (عبارت دوم) باشد، L^{CLIP} دقیقاً همان مقیاس عملکرد L_{π} خواهد بود. در غیر این صورت مقیاس عملکرد clip شده انتخاب می‌شود تا از تغییرات بزرگ خط‌مشی جلوگیری شود.

۳-۲-۳ مقایسه روش بهینه سازی خط‌مشی و Q-learning

نقطه قوت اصلی روش‌های بهینه سازی خط‌مشی، اصولی بودن آنهاست، به این معنا که شما مستقیماً چیزی که می‌خواهید را بهینه سازی می‌کنید. در نتیجه این روش‌ها قابل اتکا و باثبات هستند. در مقابل، روش‌های Q-learning با یادگیری تابع Q، مقیاس عملکرد را به طور غیر مستقیم بهینه می‌کند. حالت‌های زیادی برای این نوع یادگیری وجود دارد که به شکست منتهی می‌شود، بنابراین این روش‌ها ثبات کمتری دارند. [۱] اما، روش‌های Q-learning این مزیت را دارند که در هنگام کار، به طور قابل ملاحظه‌ای کارآمد هستند، زیرا آنها می‌توانند از داده‌ها به طور موثرتری نسبت به تکنیک‌های بهینه سازی خط‌مشی استفاده کنند.

تعامل بین بهینه سازی خط‌مشی و Q-learning. بهینه سازی خط‌مشی و Q-learning ناسازگار نیستند (و به نظر می‌رسد تحت برخی شرایط، معادل آن باشد)، و طیف وسیعی از الگوریتم‌ها وجود دارد که بین دو حد این طیف زندگی می‌کنند. الگوریتم‌هایی که در این طیف زندگی می‌کنند قادرند با دقت بین نقاط قوت و ضعف طرفین معامله کنند. مثالها شامل

DDPG، الگوریتمی است که همزمان با استفاده از هر یک برای بهبود دیگری، یک خط‌مشی قطعی و یک تابع Q را یاد می‌گیرد، و SAC، نوعی که از خط‌مشی‌های تصادفی، تنظیم آنتروپی^{۱۳} و چند ترند دیگر برای تثبیت یادگیری و کسب امتیاز بالاتر از DDPG در معیارهای استاندارد استفاده می‌کند.

۳-۳ روش‌های مبتنی بر مدل

۱-۳-۳ روش مدل جهان

مدل جهانی می‌تواند به سرعت و به روشی بدون نظارت آموزش ببیند تا یک بازنمایی از محیط را بیاموزد. سپس با استفاده از ویژگی‌های استخراج شده از مدل جهان به عنوان ورودی به یک عامل، می‌توان یک خط‌مشی ساده و فشرده را آموخت که می‌تواند وظیفه مورد نیاز را حل کند. حتی می‌توانیم عامل را کاملاً

¹³entropy regularization

در داخل محیط رویایی خود که توسط مدل جهانی آن ایجاد شده، آموزش دهیم و این خط‌مشی آموخته شده را به محیط واقعی انتقال دهیم.

در بسیاری از مسائل یادگیری تقویتی، یک عامل مصنوعی نیز از داشتن نمایش خوبی از حالات گذشته و حال و مدل پیش بینی خوبی از آینده، ترجیحاً یک پیش بینی قدرتمند سود می‌برد. مثلاً مدل پیاده سازی شده بر روی یک شبکه عصبی بازگشتی.^{۱۴}

اکثر رویکردهای مبتنی بر مدل موجود در یادگیری تقویتی، مدلی از محیط را یاد می‌گیرند، اما همچنان در محیط واقعی آموزش می‌بینند. در این روش، ما همچنین می‌توانیم یک محیط مصنوعی را کاملاً جایگزین محیط واقعی، کنیم و خط‌مشی عامل خود را فقط در داخل محیط تولید شده توسط مدل جهان آموزش دهیم و نهایتاً خط‌مشی آموخته شده را به محیط واقعی انتقال دهیم.

¹⁴Recurrent Neural Network

پیوست آ

مطالب تکمیلی

پیوست‌های خود را در صورت وجود می‌توانید در این قسمت قرار دهید.

مراجع

- [1] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [2] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.
- [3] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

واژه‌نامه

الف

pallet پالت	heuristic ابتکاری
robustness پایداری	worth ارزش
support پشتیبان	satisfiability ارضاپذیری
convex hull پوسته‌ی محدب	strategy استراتژی
upper envelope پوش بالایی	coalition ائتلاف
covering پوششی	

ب

projective transformation تبدیل تصویری	loading بارگذاری
equilibrium تعادل	game بازی
relaxation تعدیل	label برچسب
intersection تقاطع	linear programming برنامه‌ریزی خطی
partition تقسیم‌بندی	integer programming برنامه‌ریزی صحیح
evolutionary تکاملی	packing بسته‌بندی
distributed توزیع‌شده	best response بهترین پاسخ
	maximum بیشینه

ج

brute-force جست‌وجوی جامع
Depth-First Search جست‌وجوی عمق‌اول

پ

س	bin جعبه
constructive ساختی	
pay off, utility سود	چ
	sink چاله
ش	
quasi-polynomial شبه‌چندجمله‌ای	ح
quasi-concave شبه‌مقعر	حرکت action
ص	خ
formal صوری	selfish خودخواهانه
	clique خوشه
ع	د
rational عاقل	دودویی binary
agent-based عامل-محور	dual دوگان
action عمل	bimatrix دو ماتریسی
غ	ر
missing غائب	vertex رأس
decentralized غیرمتمرکز	behaviour رفتار
degenerate غیرمعمول	coloring رنگ‌آمیزی
ق	ز
transferable قابل انتقال	scheduling زمان‌بندی
lexicographically قاموسی	biology زیست‌شناسی
strong قوی	
ک	

art gallery نگارخانه‌ی هنر	minimum کمینه
gaurd نگهبان	
profile نمایه	م
round-robin نوبتی	مجموع زیرمجموعه‌ها subset sum
	مجموعه set
و	محور pivot
facet وجه	مختلط mixed
	مخفی hidden
ه	مستوی affine
price of anarchy (POA) هزینه‌ی آشوب	مسطح planar
social cost هزینه‌ی اجتماعی	منطقی reasonable
price of stability (POS) هزینه‌ی پایداری	موازی parallel
	ن
ی	نتیجه‌ی نهایی outcome
edge یال	نش Nash
isomorphism یکریختی	نقطه‌ثابت fixed point

Abstract

We present a standard template for typesetting theses in Persian. The template is based on the X_YPersian package for the L^AT_EX typesetting system. This write-up shows a sample usage of this template.

Keywords: Thesis, Typesetting, Template, X_YPersian



Sharif University of Technology

Department of Computer Engineering

M.Sc. Thesis

A Standard Template for Typesetting Theses in Persian

By:

Hamid Zarrabi-Zadeh

Supervisor:

Dr. Supervisor

September 2017