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


بسم الله الرحمن الرحیم

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

استاد راهنما: دکتر دانشگر امضاء:

استاد مشاور: استاد مشاور امضاء:

استاد مدعو: استاد ممتحن امضاء:

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 می‌پردازیم و الگوریتم های مهم آن را خواهیم دید. نهایتا چند کاربرد دیگر را بررسی خواهیم کرد.

کلیدواژه‌ها: یادگیری تقویتی

فهرست مطالب

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 درعوض 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 تمییز دهنده یادگیری تقویتی از روش های متداول یادگیری ماشین هستند. اقدامات عامل هوشمند، می‌تواند 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

۱ـ۱ 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 دقیق‌تر این مسئله را در فصل دوم خواهیم دید.

۱ـ۲ 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.

۱ـ۳ فضای حالت‌ها و عمل‌ها

۱ـ۴ خط مشی

۱ـ۵ سیگنال پاداش

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

[۲] Approximate Dynamic Programming
[۳] Neuro-dynamic Programming
[۴] Optimal Control Theory

agent's the Informally، R. ؟ Rt 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 مدنظرداریم می تواند به صورت بیشینه سازی مقدار میانگین یک سیگنال عددی بیان شود.

۱ـ۶ تابع ارزش

۱ـ۷ محیط

۱ـ۸ مسئله اکتشاف و بهره برداری

یکی از چالش هایی که در یادگیری تقویتی برخلاف سایر روش‌های 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.

۱ـ۹ اهداف تحقیق

۱ـ۱۰ ساختار پایان‌نامه

این پایان‌نامه شامل ـــ فصل است. در فصل ...

فصل ۲

مفاهیم اولیه یادگیری تقویتی

اصطلاح کنترل بهینه در اواخر دهه ۱۹۵۰ برای توصیف مسئله طراحی یک کنترل گر برای به حداقل رساندن اندازه گیری رفتار سیستم پویا در طول زمان مورد استفاده قرار گرفت. یکی از رویکردهای این مسئله در اواسط دهه ۱۹۵۰ توسط ریچارد بلمن و دیگران از طریق گسترش نظریه قرن نوزدهم همیلتون و جاکوبی توسعه یافت. این رویکرد از مفاهیم حالت یک سیستم پویا و یک تابع ارزش برای تعریف یک معادله تابعی استفاده می کند، که اکنون با نام معادله بلمن نامیده می شود. کلاس روش های حل مسائل کنترل بهینه با حل این معادله به عنوان برنامه نویسی پویا شناخته می شود. همچنین نسخه تصادفی گسسته از مسئله کنترل بهینه را که تحت عنوان فرایندهای تصمیم گیری مارکوف (MDP) شناخته می شود ، معرفی کرد و رونالد هوارد (۱۹۶۰) روش Policy Iteration را برای MDP ها طراحی کرد. همه اینها عناصر اساسی در تئوری و الگوریتم های یادگیری تقویتی مدرن هستند. در این فصل، یا فرایندهای تصمیم گیری مارکوف آشنا خواهیم شد و چهار عنصر اصلی یادگیری تقویتی، یعنی خط مشی، سیگنال پاداش، تابع ارزش و محیط را دقیقا تعریف خواهیم کرد. همچنین برخی روش های کلاسیک در یادگیری تقویتی را معرفی می کنیم.

۲ـ۱ دینامیک عامل ـ محیط

حوزه یادگیری تقویتی دو بازیگر اصلی دارد: عامل و محیط. موجود تصمیم گیرنده و آموزنده، عامل یادگیری یا به اختصار، عامل نامیده می شود. قسمتی که عامل با آن تعامل دارد ، شامل هر چیز خارج

از عامل ، محیط نامیده می شود. در ادبیات کنترل بهینه، معمولا به جای واژه های عامل و محیط، از کنترل کننده و سیستم کنترل شده استفاده می شود. عامل و محیط به طور مداوم با یکدیگر ارتباط برقرار می کنند ، عامل انتخاب می کند که چه اقدامی انجام دهد و محیط، به این اقدامات پاسخ می دهد و موقعیت جدیدی را به عامل ارائه می دهد. محیط همچنین مقادیر عددی ویژه ای 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 می کنند $t = ۰, ۱, ۲, ۳, ...$ در هر مرحله t ، عامل وضعیت محیط $S_t \in \mathcal{S}$ را دریافت می کند، و بر اساس آن یک عمل $A_t \in \mathcal{A}$ را انتخاب می کند. در گام بعدی، عامل به عنوان نتیجه عمل خود، یک پاداش عددی $R_{t+۱} \in \mathcal{R}$ دریافت می کند و خود را در حالت جدید $S_{t+۱}$ می یابد. دینامیک عامل ـ محیط را می توان به شکل یک دنباله از حالت ها، عمل ها و پاداش ها به شکل زیر نمایش داد:

$$S_۰, A_۰, R_۱, S_۱, A_۱, R_۲, S_۲, A_۲, R_۳, ...$$

۲ـ۲ فرایند تصمیم‌گیری مارکوف

یادگیری تقویتی از چارچوب رسمی فرایندهای تصمیم گیری مارکوف (MDP) برای تعریف تعامل بین یک عامل یادگیری و محیط آن توسط حالت ها ، اقدامات و پاداش استفاده می کند. مدل MDP یک مدل کلاسیک از تصمیم گیری متوالی است ، جایی که اقدامات نه تنها بر پاداش های فوری ، بلکه همچنین بر موقعیت‌ها و حالت‌های بعدی و از طریق آن پاداش های آینده تأثیر می‌گذارد. MDP یک فرم ایده آل ریاضی از مسئله یادگیری تقویتی است که می توان برای آن تئوری‌های دقیقی بیان کرد یک MDP متناهی ، یک MDP با مجموعه حالت‌های محدود است. بیشتر نظریه‌های فعلی یادگیری تقویتی محدود به MDP متناهی است ، اما روش ها و ایده ها به طور کلی بیان می شوند.

تعریف ۲ـ۱ (فرایند تصمیم گیری مارکوف) *یک فرایند تصمیم گیری مارکوف MDP ، ۴ تایی*

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

است که

- $\mathcal{S}$ *بیانگر مجموعه تمام* ***حالت****هاست*

- $\mathcal{A}$ *بیانگر مجموعه تمام* ***عمل****هاست*
- $\mathcal{R} \subseteq \mathbb{R}$ *بیانگر مجموعه پاداش هاست*
- $\mathcal{P}$ ***هسته احتمال انتقال*** $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S} \times \mathcal{R})$ *تابعی است که دینامیک MDP را مشخص می کند.*

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 دوتایی هایی به شکل (s', r) نسبت می دهد 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 بین صفر و یک (با احتساب خود صفر و یک) است که آن را به شکل $p(s', r|s, a)$ نمایش می دهیم:

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

نامگذاری فرایند تصمیم گیری 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 اند و نسبت به حالت‌ها و اعمال قبل از آن مستقل هستند.

۲_۳ خط مشی

خط مشی نحوه رفتار عامل را در یک زمان خاص، مشخص می کند. خط مشی، هسته اصلی یک عامل یادگیری تقویت کننده است به این معنا که به تنهایی برای تعیین رفتار کافی است. به طور تقریبی ، یک خط مشی، نگاشت از حالت های مدل شده از محیط به اقدامی است که باید در آن حالت انجام شود. خط مشی ممکن است یک عملکرد ساده یا جدول جستجو باشد ، یا ممکن است شامل محاسبات پیچیده‌ای مانند فرآیند جستجو باشد خط مشی ها ممکن است تصادفی باشند.

تعریف ۲_۲ (خط‌مشی احتمالاتی ثابت) *یک خط‌مشی احتمالاتی ثابت (یا به طور خلاصه خط‌مشی ثابت)* $\pi : S \to \Pi(A)$ *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

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

۲ـ۴ سیگنال پاداش

یک سیگنال پاداش هدف را در یک مسئله یادگیری تقویت کننده تعریف می کند. در هر مرحله ، محیط یک عدد حقیقی به نام پاداش برای عامل یادگیری تقویتی ارسال می کند. تنها هدف عامل، به حداکثر رساندن کل پاداش دریافتی در طولانی مدت است. سیگنال پاداش مشخص می کند که اتفاقات خوب و بد برای عامل چیست. این مبنای اصلی تغییر خط مشی است. سیگنال پاداش می‌تواند تابعی تصادفی از وضعیت محیط و اقدام انجام شده باشند.

۲ـ۵ عایدی و تابع ارزش

در حالی که سیگنال پاداش نشان می دهد که چه عملی خوب است، یک تابع ارزش مشخص می‌کند که چه چیزی در طولانی مدت خوب است. این نشانگر مطلوبیت طولانی مدت حالت‌ها پس از در نظر گرفتن حالت‌هایی است که احتمالاً در پی خواهند داشت. ارزش یک حالت، کل میزان پاداشی است که عامل می تواند انتظار داشته باشد در آینده کسب کند، اگر از آن حالت شروع کند. پاداش ها به یک معنا اولیه هستند، در حالی که ارزش ها، به عنوان پیش بینی پاداش ها، ثانویه هستند. بدون پاداش هیچ مقداری وجود ندارد و تنها هدف از برآورد ارزش ها، دستیابی به پاداش بیشتر است. با این وجود ، این تابع ارزش است که هنگام تصمیم گیری و ارزیابی بیشتر به آن توجه می کنیم. تعیین ارزش بسیار دشوارتر از تعیین پاداش است ارزش‌ها باید از توالی مشاهداتی که یک عامل در طول عمر خود انجام می دهد ، تخمین زده و مجدداً برآورد شوند. مهمترین مولفه تقریباً همه الگوریتم‌های یادگیری تقویتی که در نظر می گیریم، روشی برای تخمین کارآمد تابع ارزش است.

تعریف ۲ـ۳ (عایدی) ***عایدی تخفیف دار آینده*** [۱] *یا به اختصار، عایدی، در زمان t به شکل*

[۱] Future Expected Return

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

تعریف می شود که T زمانی است که اپیزود به اتمام می رسد. اگر مسئله ادامه دار باشد آنگاه $T = \infty$

از تعریف بالا نتیجه می شود

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

تعریف ۲ـ۴ (تابع ارزش حالت) *ارزش حالت s تحت خط مشی π یا $v_\pi(s)$ به شکل امیدریاضی عایدی، با شروع از s و دنبال کردن خط مشی π تعریف می شود.*

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

تابع v_π را تابع ارزش حالت [۲] مربوط به خط مشی π می نامیم.

تعریف ۲ـ۵ (تابع ارزش عمل) *ارزش عمل a در حالت s تحت خط مشی π یا $q_\pi(s,a)$ به شکل امیدریاضی عایدی، با شروع از s و انتخاب عمل a و سپس دنبال کردن خط مشی π تعریف می شود.*

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

تابع q_π را تابع ارزش عمل [۳] مربوط به خط مشی π می نامیم.

۲ـ۶ محیط

مدل محیط چیزی است که رفتار محیط را تقلید می کند، یا به طور کلی تر، اجازه می دهد تا در مورد نحوه رفتار محیط، پیشبینی کارامدی داشته باشیم. از مدلها برای برنامه ریزی استفاده می شود، یعنی هر نوع تصمیم گیری در مورد روند کار با در نظر گرفتن شرایط احتمالی آینده قبل از تجربه واقعی آنها. روش‌هایی که برای حل مشکلات یادگیری تقویتی از مدلها و برنامه ریزی‌ها استفاده می کنند، روشهای مبتنی بر مدل نامیده می شوند ، در مقابل روشهای بدون مدل هستند که همگی آزمایش و خطا هستند.

[۲] State Value Function
[۳] Action Value Function

۲ـ۷ خط‌مشی و تابع ارزش بهینه

برای MDP های متناهی، می‌توانیم خط‌مشی بهینه را به صورت زیر تعریف کنیم

تعریف ۲ـ۶ *می‌گوییم خط‌مشی π **بهتر از** خط‌مشی π' است و می‌نویسیم $\pi \geqslant \pi'$ هرگاه برای هر $s \in \mathcal{S}$*

$$v_\pi(s) \geqslant v_{\pi'}(s)$$

می‌توان نشان داد که حداقل یک خط‌مشی وجود دارد که بهتر از هر مشی دیگری باشد [۱] . به چنین خط‌مشی‌ای **خط‌مشی بهینه** گفته می‌شود. ممکن است بیش از یک خط‌مشی بهینه وجود داشته باشد ولی تابع ارزش متناظر با همه خط‌مشی های 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 عمل‌ـارزش 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

$$q_*(s,a) = \mathbb{E}[R_{t+۱} + \gamma v_*(S_{t+۱})|S_t = s, A_t = a]$$

۲ـ۸ معادله بلمن

ویژگی اساسی توابع ارزشی که در طول یادگیری تقویتی و برنامه نویسی پویا استفاده می شود این است که آنها در روابطی بازگشتی صدق می‌کنند. معادلات بلمن رابطه ای بین ارزش یک حالت و ارزش‌های حالت‌های بعدی آن را بیان می کند. با استفاده مستقیم از تعریف v_π خواهیم داشت به ازای هر $s \in \mathcal{S}$

$$v_\pi(s) \triangleq \mathbb{E}_\pi[G_t|S_t = s] \tag{۷}$$
$$= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s]$$
$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \left[r + \gamma \mathbb{E}_\pi[G_{t+1}|S_{t+1} = s']\right]$$
$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')]$$

(۱ـ۲)

به معادله ۲ـ۸، معادله بلمن گفته می شود. معادله بلمن، ارزش هر حالت را بر اساس ارزش حالت های بعدی آن بیان می کند. چنین رابطه ای را می توان برای تابع عملـارزش نیز به دست آورد که با نام معادله بلمن برای تابع عملـارزش شناخته می شود. با استفاده از تعریف تابع عملـارزش داریم

$$q_\pi(s, a) \triangleq \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$$
$$= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a]$$
$$= \sum_{s'} \sum_r p(s', r|s, a)r + \sum_{a'}$$

۲ـ۹ بهینگی و معادله بهینگی بلمن

$$v_*(s) = \max_{a \in \mathbb{A}(s)} q_{\pi_*}(s, a) = \max_a \mathbb{E}_{\pi_*}[G_t|S_t = s, A_t = a]$$
$$= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a]$$
$$= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1})|S_t$$
$$= s, A_t = a] = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_۸(s')].$$

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

۲ـ۱۰ برنامه ریزی پویا

اصطلاح *برنامه نویسی پویا* [۴] به مجموعه ای از الگوریتم ها گفته می شود که می تواند برای محاسبه خط مشی بهینه استفاده شود، در صورتی که یک مدل کامل از محیط موجود باشد. الگوریتم های کلاسیک برنامه ریزی 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 خط مشی بهینه است. هدف، تخمین تابع ارزش بهینه، v_* یا q_* است که در معادلات بهینگی بلمن صدق می کند:

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

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

اگر دینامیک محیط کاملاً مشخص باشد معادلات ۲ـ۱۰ و ۲ـ۱۰ به ترتیب یک دستگاه معادلات خطی با |𝕊| است همزمان خطی معادلات در | S | ناشناخته ها (vπ ،s (s. در اصل ، راه حل آن ساده ، اگر خسته کننده باشد ، محاسبه برای اهداف ما ، روشهای حل تکراری مناسب ترین هستند. یک دنباله را در نظر بگیرید توابع مقدار تقریبی v۰ ، v۱ ، v۲ ،. . . ، هر نقشه برداری S + به R (اعداد واقعی). اولیه تقریب ، v۰ ، خودسرانه انتخاب می شود (با این تفاوت که در صورت وجود ، به حالت ترمینال باید مقدار ۰ داده شود) ، و هر تقریب پی در پی با استفاده از معادله بلمن برای vπ (۴.۴) به عنوان یک به روزرسانی بدست می آید قانون: If the dynamics environment's are equations linear simultaneous |S| of system a is (۴.۴) then known، completely straightforward، a is solution its principle، In S). s vπ(s)، (the unknowns |S| in most are methods solution iterative purposes، our For computation. tedious، if

[۴] Dynamic Programing

،. . . v۲، v۱، v۰، functions value approximate of sequence a Consider suitable. is v۰، approximation، initial The numbers). real (the R to + S mapping each ،(۰ value given be must any، if state، terminal the that (except arbitrarily chosen for equation Bellman the using by obtained is approximation successive each and rule: update an as (۴.۴) vπ

$$v_{k+۱}(s) \triangleq \mathbb{E}_\pi[R_{t+۱} + \gamma v_k(S_{t+۱})|S_t = s] = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_k(s')]$$

the because rule update this for point fixed a is vπ = vk Clearly، S. ۲ s all for sequence the Indeed، case. this in equality of us assures vπ for equation Bellman condi- same the under ∞ ۲ k as vπ to converge to general in shown be can vk policy iterative called is algorithm This vπ. of existence the guarantee that tions evaluation.

۲ـ۱۱ بهبود خط مشی

bet- find help to is policy a for function value the computing for reason Our arbitrary an for vπ function value the determined have we Suppose policies. ter not or whether know to like would we s state some For π. policy deterministic We π(s). =۶ a action an choose deterministically to policy the change should we vπ(s)—but is s—that from policy current the follow to is it good how know be it would

question this answer to way One policy؟ new the to change to worse or better The π. policy، existing the following thereafter and s in a selecting consider to is is behaving of way this of value

قضیه‌ی ۲ـ۱ (قضیه‌ی بهبود خط مشی) *فرض کنید π و π' دو خط مشی معین باشند که برای هر*

$s \in S$

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

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

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

$$v_\pi(s) \leqslant q_\pi(s, \pi'(s)) = \mathbb{E}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = \pi'(a)\right]$$
$$= \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma v_\pi(s_{t+1}) | S_t = s\right]$$
$$\leqslant \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) | S_t = s]$$
$$= \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}_{\pi'}[R_{t+2} + \gamma v_\pi(S_{t+2})] | S_t = s]$$
$$\leqslant \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(s_{t+3}) | S_t = s]$$
$$\vdots$$
$$\leqslant \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ldots | S_t = s]$$
$$= v_{\pi'}(s)$$

۱۲ـ۲ الگوریتم Iteration Policy

we ، ۰ π policy، better a yield to vπ using improved been has π، policy، a Once can We .۰۰ π better even an yield to again it improve and vπ۰ compute then can functions: value and policies improving monotonically of sequence a obtain thus

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

۱۳ـ۲ الگوریتم Iteration Value

eval- policy involves iterations its of each that is iteration policy to drawback One multiple requiring computation iterative protracted a be itself may which uation،

con- then iteratively، done is evaluation policy If set. state the through sweeps conver- exact for wait we Must limit. the in only occurs vπ to exactly vergence suggests certainly ۱.۴ Figure in example The that؟ of short stop we can or gence، policy example، that In evaluation. policy truncate to possible be may it that correspond- the on effect no have three first the beyond iterations evaluation be can iteration policy of step evaluation policy the fact، In policy. greedy ing policy of guarantees convergence the losing without ways several in truncated after stopped is evaluation policy when is case special important One iteration. iter- value called is algorithm This state). each of update (one sweep one just combines that operation update simple particularly a as written be can It ation. steps: evaluation policy truncated and improvement policy the

to converge to shown be can vk sequence the v۰، arbitrary For S. ۴ s all for v۴. of existence the guarantee that conditions same the under v۴

۲ـ۱۴ 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

۲ـ۱۵ 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 ـ ارزش بهینه همگرا می شوند، $Q_i \longrightarrow Q^*$ وقتی $i \longrightarrow \infty$

در عمل ، این رویکرد کلی کاملا غیر عملی است ، زیرا تابع عمل ـ 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 ـ ارزش استفاده می شود، در فصل سوم با این روش بیشتر آشنا خواهیم شد.

فصل ۳

نتایج اخیر

در این فصل به معرفی برخی از روش های مدرن در یادگیری تقویتی می پردازیم. روش های معرفی شده در این فصل به دو دسته کلی مبتنی بر مدل و بدون مدل تقسیم می شوند. تفاوت اصلی این روش های در این است که آیا در الگوریتم یادگیری از تابع انتقال MDP یا تخمینی از آن استفاده می شود یا خیر. در روش های بدون مدل، عامل هیچ اطلاعی از دینامیک محیط ندارد و تنها از طریق تجربه می تواند بیاموزد. در مقابل روش های مبتنی بر مدل دسترسی کامل یا تقریبی به تابع انتقال را مفروض می گیرند. البته برخی از روش های مبتنی بر مدل می توانند از طریق تجربه تخمینی از دینامیک محیط را یاد بگیرند.

۳ـ۱ روش‌های مبتنی بر مدل و بدون مدل

یکی از مهمترین نقاط انشعاب در الگوریتم‌های 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 پیش بینی می کند.

نکته مثبت اصلی در داشتن مدل این است که به عامل اجازه می دهد با تفکر از قبل ، ببیند چه اتفاقی برای طیف وسیعی از گزینه های ممکن رخ می دهد و به صراحت در مورد گزینه های خود تصمیم بگیرد. سپس عامل می تواند نتایج حاصل از برنامه ریزی قبلی را در قالب یک خط مشی بیاموزد یک

نمونه مشهور از این روش AlphaZero است. در عمل، اگر دستیابی به مدلی از محیط امکان پذیر و عملی باشد، معمولا از روش های مبتنی بر مدل استفاده می شود. زیرا می تواند باعث بهبود قابل توجه کارایی نمونه نسبت به روش های بدون مدل شوند.

اصلی ترین نقطه ضعف این روش ها این است که معمولا یک مدل کامل از محیط در دسترس عامل نیست و عامل، باید مدل را کاملاً از طریق تجربه یاد بگیرد.

الگوریتم هایی که از یک مدل استفاده می کنند ، روش های مبتنی بر مدل و آنهایی که از چنین مدلی استفاده نمی کنند، بدون مدل نامیده می شوند ، نامیده می شوند. در حالی که روش های بدون مدل از دستاوردهای بالقوه در بهره وری نمونه با استفاده از مدل چشم پوشی می کنند ، اما پیاده سازی و تنظیم آنها آسان تر است. به همین خاطر، روش های بدون مدل از محبوبیت بیشتری برخوردار بوده و به طور گسترده تری نسبت به روش های مبتنی بر مدل توسعه و آزمایش شده اند.

۳ـ۲ روش های بدون مدل

۳ـ۳ روش های مبتنی بر ارزش

در روشهای یادگیری تقویتی بدون مدل مبتنی بر ارزش تابع ارزش عمل با استفاده از یک function approximator، مانند شبکه عصبی ، نشان داده می شود. فرض کنید $Q(s,a;\theta)$ یک تابع ارزش عمل تقریبی با پارامتر θ باشد الگوریتم های مختلفی برای بروزرسانی θ وجود دارد الگوریتم Q-learning یکی از نمونه های چنین الگوریتمی است که هدف آن تقریب مستقیم تابع عمل_ارزش بهینه $Q^*(s,a) \approx Q(s,a:\theta)$ است

در Q-learning یک مرحله ای، پارامترهای θ از تابع عمل_ارزش با به حداقل رساندن تابع هزینه به شکل مرحله به مرحله آموخته می شوند، به شکلی که تابع هزینه ام-i 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 می شود که s' حالتی است که بعد از حالت s دیده می شود.

۳ـ۴ روش های مبتنی بر خط مشی

در این قسمت روشهایی را در نظر می گیریم که به جای استفاده از تابع عمل‌ـارزش یا حالت‌ـارزش برای دستیابی به خط مشی بهینه، یک خط مشی پارامتریزه [۱] شده را می آموزد که می تواند اقدامات را بدون استفاده از یک تابع ارزش، انتخاب کند. یک تابع ارزش ممکن است همچنان برای یادگیری پارامترهای خط مشی استفاده شود ، اما برای انتخاب اقدام مورد نیاز نیست. ما از نماد $\theta \in \mathbb{R}^{d'}$

برای بردار پارامتر خط مشی استفاده می کنیم. بر خلاف روشهای مبتنی بر ارزش ، روشهای مبتنی بر خط مشی مستقیماً تابع خط مشی $\pi_\theta(a|s)$ را تخمین می زنند و پارامترهای θ را با استفاده از **صعود گرادیان** روی یک مقیاس عملکرد [۲] $J(\pi_\theta)$

یا به طور مستقیم و یا با بیشینه سازی تخمین های محلی از $J(\pi_\theta)$ بروزرسانی می کند. این روش تقریبا همیشه به صورت on-policy عمل می کنند.

همانطور که در ادامه خواهیم دید، می توان از توابع مختلفی برای مقیاس عملکرد J استفاده نمود. یک انتخاب بدیهی $J(\pi_\theta) = \mathbb{E}[R_t]$ است. این روش های تلاش می کنند تابع J را بیشینه کنند

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

که $\nabla \hat{J}(\theta_t)$

تخمینی احتمالاتی است که که امیدریاضی آن گرادیان مقیاس عملکرد J را نسبت به پارامترهای خط مشی θ_t تخمین می زند.

به روش هایی که چنین الگویی را برای محاسبه خط مشی بهینه دنبال می کنند، روش های **گرادیان خط مشی** [۳] می گوییم. دسته ای از روش های گرادیان خط مشی وجود دارند که تلاش می کنند تخمینی از تابع ارزش را نیز محاسبه کنند. به چنین روش هایی روش های، **بازیگر‌ـ منتقد** (Actor-Critic) گفته می شود که **بازیگر** (Actor) اشاره به خط مشی آموخته شده و **منتقد** (Critic) اشاره به تابع ارزش آموخته شده (معمولا یک تابع حالت‌ـارزش) دارد.

[۱] Parameterized
[۲] Performance Measure
[۳] Policy Gradient

قضیه‌ی ۳ـ۱ (گرادیان خط مشی)

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

که μ یک توزیع احتمال روی S است که متناسب با تعداد دفعاتی است که حالت s با دنبال کردن خط مشی π_θ تکرار می شود.

می توان نشان داد [؟]

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

بنابراین در هر گام $\left[R_t \frac{\nabla_\theta \pi_\theta(a|S_t)}{\pi_\theta(a|S_t)} \right]$ یک تخمین گر ناریب از $\nabla J(\pi_\theta)$ خواهد بود. پس می توان 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 روش های بهینه سازی خط مشی به شرح زیر است.
روش های Actor-Critic که الگوریتم ascent Gradient را مستقیما برای بیشینه سازی $J(\pi\theta)$ به کار می برند.

روش Optimization Policy Proximal که

۳ـ۵ روش های Actor-Critic

یک نمونه از روش های بهینه سازی خط مشی، خانواده REINFORCE از الگوریتم های یادگیری تقویتی است. [۲]

الگوریتم استاندارد REINFORCE پارامترهای θ را در جهت

$\nabla_\theta log\ \pi(a_t|s_t:\theta)R_t$ بروزرسانی می کند که یک تخمین ناریب از $\nabla_\theta \mathbb{E}[R_t]$ است. می توان با کم کردن یک تابع آموخته شده روی حالت ها $b_t(s_t)$ از R_t ، واریانس این تخمین را کاهش داد بطوریکه ناریب باقی بماند. به چنین تابعی **پایه** گفته می شود. نتیجتا گرادیان به شکل $\nabla_\theta\ log\ \pi(a_t|s_t;\theta)(R_t -$

$b_t(s_t)$) خواهد بود.

معمولا از یک تخمین آموخته شده از تابع ارزش به عنوان پایه استفاده می شود $b_t(s_t) \approx V^{\pi}(s_t)$

که منجر به تخمینی با واریانس بسیار کوچک تر از گرادیان خط مشی می شود در حالیکه تخمین، ناااریب باقی می ماند و نتیجتا عملیات یادگیری با سرعت بیشتری انجام می شود. این روش می تواند به شکل معماری **بازیگر_منتقد** [۴] 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** [۵] 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 $AdvantageActor - Critic$ یا $A۲C$ گفته می شود.

[۱]

۳_۶ روش TRPO

تعریف ۳_۱ $\rho_{\pi} : S \longrightarrow \mathbb{R}$ *تابع فرکانس تخفیف دار دیده شدن حالت ها باشد. یعنی* $\rho_{\pi}(s) = P(S_۰ = s) + \gamma P(S_۱ = s) + \gamma^{۲} P(S_۲ = s) + ...$

که دنباله $S_۰, S_۱, S_۲$ *خط مشی* π *را دنبال می کند.*

اگر π و π' دو خط مشی باشند و $J(\pi) = \mathbb{E}_{\pi}[R_۰]$ می توان نشان داد [؟]

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

[۴] Actor-Critic
[۵] Advantage

الگوریتم ۱ الگوریتم $Advantage\ Actor-Critic$

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: $۱ \longrightarrow I$

۵: تا وقتی S حالت نهایی نیست::

۶: $A \sim \pi_\theta(.|S)$

۷: عمل A را انجام بده و حالت S' و پاداش R را مشاهده کن

۸: $R + \gamma v_\omega(S') - v_\omega(S) \longrightarrow \delta$

۹: $\omega + \alpha^\omega I \delta \nabla_\omega v_\omega(S) \longrightarrow \omega$

۱۰: $\theta + \alpha^\theta I \delta \nabla_\theta \, ln \, \pi_\theta(A|S) \longrightarrow \theta$

۱۱: $\gamma I \longrightarrow I$

۱۲: $S' \longrightarrow S$

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' بهینه سازی مستقیم را مشکل می کند.

برای حل این مشکل [؟] مقیاس عملکرد دیگری $L_\pi(\pi')$ را معرفی می کند و نشان می دهد که اگر π و π' به اندازه کافی به یکدیگر نزدیک باشند، افزایش $L_\pi(\pi')$ همواره با افزایش J همراه خواهد بود.

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

توجه کنید که L_π از تابع فرکانس ρ_π به جای $\rho_{\pi'}$ استفاده می کند.

قضیه‌ی ۳ـ۲ *فرض کنید* $\alpha = D^m_{TV}ax(\pi_{old}, \pi_{new})$ *باشد که*

$$D^m_{TV}ax(\pi, \pi') = \max_s D_{TV}(\pi(.|s)||\pi'(.|s)$$

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

$$D_{TV}(p||q) = \frac{۱}{۲} \sum_i |p_i - q_i|$$

در این صورت

$$J(\pi_{new}) \geqslant L_{\pi_{old}}(\pi_{new}) - \frac{۴\epsilon\gamma}{(۱-\gamma)^۲}$$

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

با توجه به قضیه فوق و نامعادله $D_{TV}(p||q)^۲ \leqslant D_{KL}(p||q)$ [؟] که $D_{KL}(p||q)$ برابر با دیورژانس KL دو بردار p و q است. می توان نتیجه گرفت

$$J(\pi') \geqslant L_\pi(\pi') - C\ D^m_{KL}ax(\pi, \pi')$$

که

$$C = \frac{۴\epsilon\gamma}{(۱-\gamma)^۲}$$

رابطه بالا نشان می دهد که می توان یک دنباله صعودی از خط مشی ها داشت به طوری که $J(\pi_۰) \leqslant J(\pi_۱) \leqslant J(\pi_۲) \leqslant ...$. چرا که فرض کنید $M_i(\pi) = L_{\pi_i}(\pi) - C\ D^m_{KL}ax(\pi_i, \pi)$ در این صورت $J(\pi_{i+۱}) \geqslant M_i(\pi i + ۱)$

$J(\pi_i) = M_i(\pi_i)$ بنابراین $J(\pi_{i+۱}) - J(\pi_i) \geqslant M_i(\pi_{i+۱}) - M_i(\pi_i)$ بنابراین با بیشینه کردن M_i در هر گام می توان اطمینان حاصل کرد که مقیاس عملکرد واقعی J غیرنزولی خواهد بود.

الگوریتم ۲ الگوریتم $PolicyIteration$ با مقیاس عملکرد L_π

۱: خط مشی $\pi_۰$ را مقداردهی اولیه کن

۲: برای $i = ۰, ۱, ...$ تکرار کن:

۳: همه ی مزیت های $A_{\pi_i}(s, a)$ را محاسبه کن

۴: $arg\max_\pi[L_{\pi_i}(\pi) - C\ D^m_{KL}ax(\pi_i, \pi)] \longrightarrow \pi_{i+۱} =$ که $C = (۴\epsilon\gamma)/(۱ - \gamma)^۲$ و $L_{\pi_i}(\pi) = J(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$

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.

۳ـ۷ روش PPO

در روش TRPO دیدیم که بیشینه سازی مقیاس عمکلرد L_π ساده تر از مقیاس عملکرد J است ولی در عوض الگوریتم ascent Gradient تنها مجاز به اعمال تغییرات کوچک در خط مشی است. یک راه دیگر برای کنترل تغییرات خط مشی 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 $r_t(\theta_{new}) = \frac{\pi_{\theta_{new}}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ باشد. بنابراین

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

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

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), ۱ - \epsilon, ۱ + \epsilon)\hat{A}_t \right]$$

که ϵ یک ابرپارامتر [۶] مثلا $\epsilon = ۰/۲$ است. اولین عبارت داخل min همان مقیاس عملکرد روش TRPO است. در عبارت دوم مقادیر بزرگتر از $۱ + \epsilon$ یا کوچکتر از $۱ - \epsilon$ در $r_t(\theta)$ به ترتیب به $۱ + \epsilon$ و $۱ - \epsilon$ تغییر پیدا کرده اند تا تغییرات بزرگ خط مشی را کنترل کنند. نهایتا عبارت کوچکتر از میان این دو انتخاب خواهد شد. بنابراین اگر مقیاس عملکرد clip نشده (عبارت اول) کوچکتر یا مساوی با حالت clip شده (عبارت دوم) باشد، L^{CLIP} دقیقا همان مقیاس عملکرد L_π خواهد بود. در غیر این صورت مقیاس عملکرد clip شده انتخاب می شود تا از تغییرات بزرگ خط مشی جلوگیری شود.

۳ـ۸ روش های Q-learning

خانواده روش های Q-learning تلاش می کنند مستقیما تابع ارزش عمل_حالت بهینه $Q^*(s, a)$ را تخمین بزنند. آنها به طور معمول از یک تابع هدف مبتنی بر معادله بلمن استفاده می کنند. این بهینه سازی تقریباً همیشه به صورت off-policy انجام می شود، به این معنی که هر به روزرسانی می تواند از داده های جمع آوری شده در هر نقطه استفاده کند ، بدون درنظرگرفتن نحوه انتخاب عامل برای کشف محیط در هنگام بدست آوردن داده ها. خط مشی مربوطه از طریق ارتباط بین Q^* و π^* بدست می آید: عامل بعد از یادگرفتن تابع $Q_\theta(s, a)$ به طوریکه $Q_\theta(s, a) \approx Q^*(s, a)$ می تواند عمل بهینه در حالت s را به به صورت زیر محاسبه کند

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

[۶] Hyperparameter

.

از جمله الگوریتم های Q-learning می توان به موارد زیر اشاره کرد

روش کلاسیک DQN که حوزه یادگیری تقویتی ژرف را عمیقا ارتقا بخشید

روش C۵۱ که توزیعی روی عایدی را می آموزد که امیدریاضی آن Q^* است

۳ـ۹ روش DQN

معمولا برای تقریب زدن توابع ارزش در یادگیری تقویتی، از یک تابع خطی استفاده می شود. اما گاهی اوقات از یک تقریب عملکرد غیر خطی به جای آن ، مانند یک شبکه عصبی استفاده می شود. شبکه های عصبی با عنوان شبکه Q[۷] شناخته می شوند. شبکه Q را می توان با کمینه ساختن دنباله ای از توابع هزینه به شکل $L_۱(\theta_۱), L_۲(\theta_۲), L_۳(\theta_۳), ...$

آموزش داد؛ به طوری که

$$L_i(\theta_i) = \mathbb{E}\left[(y_i - Q(s,a;\theta_i))^۲\right]$$

که

$$y_i = \mathbb{E}[r + \gamma \max_{a'} Q(s',a';\theta_{i-۱})|s,a]$$

.

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

$$\nabla_{\theta_i} L_i \theta_i = \mathbb{E}\left[\left(r + \gamma \max a' Q(s',a';\theta_{i-۱}) - Q(s,a;\theta_i)\right) \nabla_{\theta_i} Q(s,a;\theta_i)\right]$$

به جای محاسبه امیدریاضی کامل در گرادیان فوق ، غالباً از نظر محاسباتی، بهینه سازی تابع هزینه با نزول گرادیان تصادفی [۸] راه حل بهتری است. اگر در هر مقطع زمانی، وزن ها به روزرسانی شود و

[۷] Q-Network
[۸] stochastic gradient descend

امیدریاضی با یک نمونه از توزیع خط مشی رفتار [۹] جایگزین شود، الگوریتم Q-learning به شکل زیر خواهد بود.

الگوریتم ۳ الگوریتم Q-learning با replay Experience

حافظه replay D را مقدار دهی اولیه کن
تابع عمل_ارزش Q را با وزن های تصادفی مقداردهی اولیه کن
برای هر اپیزود $M...۱$:
دنباله $d_۱ = \{S_۱\}$ و کدینگ $\phi_۱ = \phi(d_۱)$ را مقداردهی اولیه کن
برای $t = ۱...T$:
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
عمل a_t را انجام بده و حالت $S_{t+۱}$ و پاداش R_t را مشاهده کن
قرار بده $d_{t+۱} = d_t, a_t, S_{t+۱}$ و $\phi_{t+۱} = \phi(d_{t+۱})$
تجربه $(\phi_t, A_t, R_t, \phi_{t+۱})$ را در D ذخیره کن
یک نمونه تصادفی از تجربه های $(\phi(j), A_j, R_j, \phi_{j+۱})$ از انبار تجربیات D انتخاب کن
قرار بده $y_j = \begin{cases} \mathrm{r}_j & \phi_{j+1}\ terminal \\ \mathrm{r}_j & \text{otherwise} \end{cases}$
یک گام از نزول گرادیان را برای تابع هزینه $(y_j - Q(\phi_j, a_j; \theta))^۲$ انجام بده

توجه داشته باشید که این یک الگوریتم بدون مدل است: این کار وظیفه یادگیری تقویتی را مستقیماً با استفاده از نمونه های شبیه ساز E بدون ساختن صریح تخمین .E حل می کند. در مورد استراتژی حریص a = maxa Q (s ، a؛) یاد می گیرد ، در حالی که توزیع رفتاری را دنبال می کند که کاوش کافی در فضای دولت را تضمین می کند. در عمل ، توزیع رفتار اغلب توسط یک استراتژی Greed انتخاب می شود که استراتژی حریصانه را با احتمال ۱ دنبال می کند ــ و یک اقدام تصادفی با احتمال ϵ

[۹] behavior policy

۳ـ۱۰ روش C۵۱

۳ـ۱۱ مقایسه روش بهینه سازی خط مشی و Q-learning

نقطه قوت اصلی روش های بهینه سازی خط مشی، اصولی بودن آنهاست ، به این معنا که شما مستقیماً چیزی که می خواهید را بهینه سازی می کنید. در نتیجه این روش ها قابل اتکا و باثبات هستند. در مقابل ، روشهای Q-learning با یادگیری تابع Q، مقیاس عملکرد را به طور غیر مستقیم بهینه می کند. حالت های زیادی برای این نوع یادگیری وجود دارد که به شکست منتهی می شود، بنابراین این روش ها ثبات کمتری دارند. [۱] اما ، روش های Q-learning این مزیت را دارند که در هنگام کار ، به طور قابل ملاحظه ای کارآمد هستند ، زیرا آنها می توانند از داده ها به طور موثرتری نسبت به تکنیک های بهینه سازی خط مشی استفاده کنند.

تعامل بین بهینه سازی خط مشی و .Q-learning بهینه سازی خط مشی و Q-learning ناسازگار نیستند (و به نظر می رسد تحت برخی شرایط ، معادل آن باشد) ، و طیف وسیعی از الگوریتم ها وجود دارد که بین دو حد این طیف زندگی می کنند. الگوریتم هایی که در این طیف زندگی می کنند 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 خط مشی قطعی و یک تابع Q را یاد می گیرد ، و SAC ، نوعی که از خط مشی های تصادفی ، تنظیم آنتروپی [۱۰] و چند ترفند دیگر برای تثبیت یادگیری و کسب امتیاز بالاتر از DDPG در معیارهای استاندارد استفاده می کند.

۳ـ۱۲ روش DDPG

success recent the from insights with approach actor-critic the combine we Here was it DQN، to Prior .(۲۰۱۵ ؛۲۰۱۳ al.، et (Mnih (DQN) Network Q Deep of function non-linear large، using functions value learning that believed generally functions value learn to able is DQN unstable. and difficult was approximators

[۱۰] entropy regularization

inno- two to due way robust and stable a in approximators function such using buffer replay a from samples with off-policy trained is network the .۱ vations: tar- a with trained is network the .۲ samples: between correlations minimize to In backups. difference temporal during targets consistent give to network Q get (Ioffe normalization batch with along ideas، same the of use make we work this learning. deep in advance recent a ،(۲۰۱۵ Szegedy،

۳ـ۱۳ روش SAC

۳ـ۱۴ روش های مبتنی بر مدل

of clusters easy-to-define of number small a aren't there RL، model-free Unlike models. using of ways orthogonal many are there RL: model-based for methods the case، each In exhaustive. from far is list the but examples، few a give We'll learned. or given be either may model

repre- explicitly never approach basic most The Planning. Pure Background: model-predictive like techniques planning pure uses instead، and policy، the sents envi- the observes agent the time each MPC، In actions. select to (MPC) control where model، the to respect with optimal is which plan a computes it ronment، the after time of window fixed some over take to actions all describes plan the planning the by considered be may horizon the beyond rewards (Future present. executes then agent The function.) value learned a of use the through algorithm a computes It it. of rest the discards immediately and plan، the of action first the using avoid to environment، the with interact to prepares it time each plan new horizon. planning shorter-than-desired a with plan a from action an

الگوریتم ۴ 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 وزن های $\theta_\mu \longleftrightarrow \theta_{\mu'}$ و $\theta_Q \longleftarrow \theta_{Q'}$ مقداردهی اولیه کن

۳: حافظه تکرارها R را بساز برای هر اپیزود M...۱

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

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

۶: عمل $a_t = \mu(s_t;\theta_\mu) + \mathbb{N}_t$ را بر اساس خط مشی 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 تجربه های (s_i, a_i, r_i, s_{i+1}) از انبار تجربه R انتخاب کن

۹: وزن های منتقد θ_Q را با درنظر گرفتن تابع هزینه $L = \frac{1}{N}\sum_i (y_i - Q(S_i, A_i;\theta_Q))^2$ به روزرسانی کن

۱۰: وزن های بازیگر θ را با استفاده از گرادیان خط مشی 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: وزن های توابع هدف را به شکل

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

به روز رسانی کن

۱۳:

۱۴:

some on models environment learned with MPC explores work MBMF The follow- straightforward A Iteration. Expert RL. deep for tasks benchmark standard the of representation explicit an learning and using involves planning pure to on

$\pi_\theta(a|s).The agent uses a planning algorithm (like Monte Carlo Tree Search) in the model, generating candidate acti$

play to networks neural deep train to approach this uses algorithm ExIt The for Augmentation Data approach. this of example another is AlphaZero Hex. Q- or policy a train to algorithm RL model-free a Use Methods. Model-Free updating in ones fictitious with experiences real augment (۱ either but function، agent. the updating for experience fictitous only use (۲ or agent، the fictitious with experiences real augmenting of example an for MBVE See experience fictitious purely using of example an for Models World See ones. Planning Embedding dream." the in "training call they which agent، the train to directly procedure planning the embeds approach Another Policies. into Loops information side become plans complete that subroutine—so a as policy a into model- standard any with policy the of output the training policy—while the for to learn can policy the framework، this in that is concept key The algorithm. free problem، a of less bias model makes This plans. the use to when and how choose learn simply can policy the states، some in planning for bad is model the if because it. ignore to

imagina- of style this with endowed being agents of example an for I۲A See tion.

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] T. Degris, M. White, and R. S. Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

واژه‌نامه

الف

ابتکاری ... heuristic
ارزش ... worth
ارضاپذیری ... satisfiability
استراتژی ... strategy
ائتلاف ... coalition

ب

بارگذاری ... loading
بازی ... game
برچسب ... label
برنامه‌ریزی خطی ... linear programming
برنامه‌ریزی صحیح ... integer programming
بسته‌بندی ... packing
بهترین پاسخ ... best response
بیشینه ... maximum

پ

پالت ... pallet
پایداری ... robustness
پشتیبان ... support
پوسته‌ی محدب ... convex hull
پوش بالایی ... upper envelope
پوششی ... covering

ت

تبدیل تصویری ... projective transformation
تعادل ... equlibrium
تعدیل ... relaxation
تقاطع ... intersection
تقسیم‌بندی ... partition
تکاملی ... evolutionary
توزیع‌شده ... distributed

ج

جست‌وجوی جامع ... brute-force
جست‌وجوی عمق‌اول ... Depth-First Search

جعبه ..... bin

چ

چاله ..... sink

ح

حرکت ..... action

خ

خودخواهانه ..... selfish
خوشه ..... clique

د

دودویی ..... binary
دوگان ..... dual
دو ماتریسی ..... bimatrix

ر

رأس ..... vertex
رفتار ..... behaviour
رنگ‌آمیزی ..... coloring

ز

زمان‌بندی ..... scheduling
زیست‌شناسی ..... biology

س

ساختی ..... constructive
سود ..... pay off, utility

ش

شبه‌چندجمله‌ای ..... quasi-polynomial
شبه‌مقعر ..... quasi-concave

ص

صوری ..... formal

ع

عاقل ..... rational
عامل‌ـ‌محور ..... agent-based
عمل ..... action

غ

غائب ..... missing
غیرمتمرکز ..... decentralized
غیرمعمول ..... degenerate

ق

قابل انتقال ..... transferable
قاموسی ..... lexicographically
قوی ..... strong

ک

کمینه ..... minimum

م

مجموع زیرمجموعه‌ها ..... subset sum
مجموعه ..... set
محور ..... pivot
مختلط ..... mixed
مخفی ..... hidden
مستوی ..... affine
مسطح ..... planar
منطقی ..... reasonable
موازی ..... parallel

ن

نتیجه‌ی نهایی ..... outcome
نش ..... Nash
نقطه‌ثابت ..... fixed point
نگارخانه‌ی هنر ..... art gallery
نگهبان ..... gaurd
نمایه ..... profile
نوبتی ..... round-robin

و

وجه ..... facet

هـ

هزینه‌ی آشوب ..... price of anarchy (POA)
هزینه‌ی اجتماعی ..... social cost
هزینه‌ی پایداری ..... price of stability (POS)

ی

یال ..... edge
یکریختی ..... isomorphism

Abstract

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

Keywords: Thesis, Typesetting, Template, XePersian

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