دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

به کارگیری روش یادگیری تقویتی برای مسیریابی پویا در شبکه به عنوان یک سیستم چندعاملی

پایان‌نامه کارشناسی ارشد- هوش مصنوعی

سعید مجیدی

استاد راهنما

دکتر مسعودرضا هاشمی

۱۳۸٦

دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

پایان‌نامه کارشناسی ارشد مهندسی کامپیوتر – هوش‌مصنوعی آقای سعید مجیدی

تحت عنوان

به کارگیری روش یادگیری تقویتی برای مسیریابی پویا در شبکه به عنوان یک سیستم چندعاملی

در تاریخ ۱۳۸۶/۲/۱ توسط کمیته زیر مورد بررسی و تصویب نهائی قرار گرفت .

۱- استاد راهنمای پایان‌نامه	دکتر مسعودرضا هاشمی
۲- استاد مشاور پایان‌نامه	دکتر مازیار پالهنگ
۳- استاد داور	دکتر جمال گلستانی
۴- استاد داور	دکتر جواد عسکری
سرپرست تحصیلات تکمیلی دانشکده	دکتر علی‌محمد دوست‌حسینی

من لم یشکر المخلوق لم یشکر الخالق

از دست و زبان که بر آید
کز عهده شکرش به در آید

حمد و سپاس بی‌پایان از آنّ خالق یکتاست که اندک داشته و اندوخته ناچیز خود را مرهون لطف و مرحمت بی‌کران او می‌باشم. پس از تواضع در برابر آستان حضرت دوست، لازم می‌دانم از خانواده عزیزم، بخصوص پدر و مادر مهربانم به خاطر زحمات بی‌دریغشان صمیمانه تشکر کنم.

از استاد ارجمند و بزرگوار، جناب آقای دکتر **مسعودرضا هاشمی**، استاد راهنمای پایان نامه، که در طی سال‌های اخیر افتخار شاگردی در مکتب پر خیر و برکت علم و اخلاق ایشان مایه مباهات بنده بوده و همواره از رهنمودهای ارزشمندشان بهره‌مند گردیده‌ام، تشکر و قدردانی می‌نمایم و از ایزد منان سلامتی و بهروزی را برای ایشان خواستارم.

ازاستاد مشاور پایان نامه، جناب آقای دکتر **مازیار پالهنگ** ، که بسیار صمیمانه و با صبر و حوصله فراوان تجربیات ارزشمند خود را در اختیار اینجانب قرار دارند تشکر می‌نمایم.

از اساتید محترم، جناب آقای دکتر گلستانی و جناب آقای دکتر عسکری که زحمت داوری این پایان‌نامه را بر عهده گرفتند و با راهنمایی‌های خود کمک ارزشمندی در تکمیل و بهبود این پایان نامه داشتند تشکر فراوان دارم.

همچنین از جناب آقای دکتر دوست حسینی، سرپرست محترم تحصیلات تکمیلی دانشکده و سرکار خانم نکویی به‌خاطر زحمات بی‌شائبه ایشان در طی این دوره کمال تشکر و قدردانی را به عمل می‌آورم.

از کلیه دوستان عزیز که دوران خوشی را در کنار آنها گذراندم و افتخار آشنایی با آنها را داشتم، تشکر می‌کنم و از خداوند متعال، سلامتی و بهروزی همه این عزیزان را خواستارم.

سعید مجیدی

بهار ۱۳۸۶

دانشگاه صنعتی اصفهان

این پایان‌نامه با حمایت مرکز **تحقیقات مخابرات ایران** (قرارداد
شماره ۵۰۰/۲۹۰۷/ت مورخ ۱۳۸۳/۳/۲۴) به انجام رسیده است.

تقدیم به

پدر و **مادر**

خوبم،

برادر و

خواهر

عزیزم،

و **او** که

هرچه داریم

از اوست.

فهرست مطالب

عنوان **صفحه**

فصل سوم: الگوریتم‌های مسیریابی برمبنای هوش مصنوعی

فصل چهارم: روش پیشنهادی

فصل پنجم: شبیه‌سازی و نتایج حاصل از آن

فصل ششم: نتیجه‌گیری و پیشنهادات

چکیده

شبکه‌های کامپیوتری نمونه مهمی از سیستم‌های پویای توزیع شده هستند که در زندگی روزانه نقش زیادی پیدا نموده‌اند. اهمیت استراتژیک و محدودیت‌های ذاتی این گونه سیستم‌ها منجر به نیاز به کنترل توزیع‌شده، به‌خصوص برای مسیریابی، می‌شود تا رفتار شبکه را قابل تطبیق با تغییرات در توپولوژی، ترافیک داده‌ها، سرویس‌ها و غیره نماید. اخیراً، محققین بر روی الگوریتم‌های مسیریابی جدیدتری به منظور فراهم آوردن میزان انطباق‌پذیری بیشتر با تغییر حالات شبکه تحقیق کرده‌اند که این الگوریتمها بر پایه روش‌های یادگیری ماشین بنا شده‌اند. یادگیری تقویتی یک روش یادگیری بدون سرپرست است که هدف از آن یادگیری یک سیاست، نگاشتی از مشاهدات به اعمال، بر مبنای بازخوردی است که از محیط دریافت می‌شود. این عمل یادگیری را می‌توان به صورت جستجوی مجموعه‌ای از سیاست‌ها نگریست که در هر لحظه در تعامل با محیط ارزیابی می‌شوند. در میان الگوریتم‌های مختلفی که برای یادگیری تقویتی تاکنون ارائه شده است الگوریتم یادگیری-Q دارای بیشترین کاربرد است. در این پایان‌نامه به شبکه به عنوان یک سیستم چندعاملی نگریسته شده است که هر گره آن نشان دهنده یک عامل می‌باشد. سپس بر پایه روش یادگیری تقویتی و با استفاده از الگوریتم یادگیری-Q هر یک از عامل‌ها شروع به یادگیری وضعیت شبکه می‌نماید تا در هر حالت شبکه بهترین عمل ممکن را از خود بروز دهد. در این روش وضعیت هر گره (عامل) به صورت تابعی از وضعیت گره‌های مجاور و خطوط ارتباطی مابین آن‌ها تعریف می‌شود و بدین صورت هر تغییر در وضعیت یک گره (عامل) در وضعیت و نحوه تصمیم‌گیری گره‌های (عاملین) مجاور آن تأثیرگذار است.

فصل اوّل

مقدمه

۱-۱ مقدمه

شبکه‌های کامپیوتری نمونه مهمی از سیستم‌های پویای توزیع‌شده هستند که در زندگی روزانه ما نقش زیادی پیدا کرده‌اند . اهمیت استراتژیک و محدودیت‌های ذاتی این‌گونه سیستم‌ها منجر به نیاز به کنترل توزیع‌شده، به‌خصوص برای مسیریابی، می‌شود تا رفتار شبکه را قابل تطبیق با تغییرات در توپولوژی، ترافیک داده‌ها، سرویس‌ها و غیره نماید. بنابراین، انجمن‌های کنترل و یادگیری ماشین همیشه علاقه‌مند به زمینه ارتباطات کامپیوتری بوده‌اند.

در دهه ۱۹۵۰ میلادی، Bellman و Ford روش برنامه‌ریزی پویا[1] را برای مسئله بهینه‌سازی مسیریابی در شبکه‌ها به‌کار بردند [۱,۲]. با اینکه الگوریتم مسیریابی Bellman-Ford یک کنترل توزیع‌شده را پیاده‌سازی می‌کند، با این‌حال میزان تطبیق پذیری محدودی را ارائه می‌نماید. بنابراین کارآیی آن در حالاتی که وضعیت شبکه با سرعت زیادی تغییر می‌کند کاهش پیدا می‌کند. در طول سالیان متمادی، افراد در جستجوی نسخه‌های تطبیق‌پذیرتری از الگوریتم اصلی Bellman-Ford بوده‌اند که به‌هرحال در این راه با مشکلات مختلفی نیز روبرو شدند. اخیراً، محققین بر روی الگوریتم‌های مسیریابی جدیدتری به منظور فراهم آوردن میزان انطباق‌پذیری بیشتر تحقیق کرده‌اند که این الگوریتمها بر پایه روش‌های یادگیری ماشین [۳] بنا شده‌اند. در این رابطه Nedzelinsk و Narendra راهکاری بر پایه اتوماتای یادگیر تصادفی[2] تولید کردند [۴]. در سال ۱۹۹۴، Boyan و Littman الگوریتم Q-routing [۵] را پیشنهاد کردند که اقتباسی از الگوریتم Bellman-Ford بود که از الگوریتم Q-Learning [۶]،که در مبحث یادگیری تقویتی[3] [۷، ۸] مطرح می‌شود، استفاده

---

[1] Dynamic Programming [2] Stochastic Learning Automata [3] Reinforcement Learning

می‌نمود. در سال ۱۹۹۸، Di Caro و Dorigo الگوریتم AntNet [۹] را ارائه نمودند که از متاهیوریستیک بهینه‌سازی اجتماع مورچه (ACO)[4] مشتق می‌شد و یک سیستم نمونه‌برداری توزیع‌شده مونت کارلو را به منظور یادگیری تصمیم‌های مسیریابی پیاده‌سازی می‌نمود.

علیرغم کارآیی بسیار خوبی که این الگوریتم‌ها و الگوریتم‌های تطبیق‌پذیرمشابه دیگر داشته‌اند، با این حال فن‌آوری‌های شبکه کنونی همچنان از الگوریتم‌های ایستای قبلی استفاده می‌کنند. الگوریتم‌های مسیریابی اینترنت همانند RIP و BGP از الگوریتم پایه Bellman-Ford مشتق شده‌اند. آن‌ها این قابلیت را دارند که با تغییرات توپولوژیکی نادری که ممکن است رخ دهد (همانند از کار افتادگی شبکه) کنار بیایند، اما در زمینه تطبیق‌پذیری ترافیک امکانی را فراهم نمی‌آورند. استراتژی اصلی برای مقابله با نوسانات ترافیکی و فراهم‌آوردن یک کیفیت سرویس گارانتی شده، محدود ساختن و مقید کردن بیش از اندازه منابع شبکه است که موجب می‌شود تا در عمل دیگر چندان ضرورتی برای وجود الگوریتم‌های مسیریابی کاملاً انطباق‌پذیر وجود نداشته باشد.

با این حال وضع موجود اکنون به‌سرعت در حال تغییر است. پیشرفت‌های حاصل در تکنولوژی‌های بی‌سیم، همانند Bluetooth و WiFi، درجه آزادی بیشتری را در هنگام ایجاد و یا تغییر یک شبکه داده‌ای فراهم آورده‌اند. همچنین معرفی مدل‌های ارتباطی و سرویس‌های کاربری جدید، همانند شبکه‌سازی Peer-to-Peer و Voice-over-IP، منجر به نیازهای جدید و در حال تغییری در زمینه ترافیک داده‌ها شده است. شبکه‌ها به سرعت در حال پویا و ناهمگون شدن هستند و از آنجاییکه این شبکه‌های جدید بیشتر بر مبنای کاربران هستند، یعنی ویژگی‌های آن‌ها توسط کاربران تعیین می شود و نه توسط یک قدرت مرکزی، لذا محدودسازی‌هایی که قبلاً ذکر شد نمی‌توانند چندان گزینه موثری باشند. انتظار می‌رود که این تحول و تکامل سرعت بیشتری به خود بگیرد و بنابراین نیاز به الگوریتم‌های کنترلی پویا و کارآمد در حال افزایش است. این الگوریتم‌ها می‌بایست وضعیت فعلی شبکه و کاربران را فرا گرفته و سیاست‌های تصمیم‌گیری خود را مطابق با آن سازند و حتی بتوانند پارامترهای درونی خود را تنظیم نمایند. حرکت گره‌های شبکه و تغییرات در الگوهای ترافیک داده‌ها که به خاطر پدید آمدن سرویس‌های جدید رخ می‌دهد، منجر به حالت‌های مختلف شبکه شده است که با خصوصیاتی همانند پهنای باند، اتصال و غیره تعریف می‌شوند. حالت شبکه ممکن است در طول زمان تغییر کند و یا حالات مختلفی همزمان در یک شبکه ناهمگون وجود داشته باشند. همچنین محدودیت‌ها و قیودی که توسط فن‌آوری‌های شبکه اعمال می‌شوند، پیچیدگی بیشتری را اضافه می‌کنند.

افزایش سرعت سخت‌افزار و دسترس بودن آن، اجازه می‌دهد تا کنترل‌کننده‌هایی را ایجاد کنیم که قابلیت و توانایی یادگیری داشته باشند. این کنترل‌کننده‌ها می‌توانند این عمل یادگیری را با استفاده از روش‌های

---

[4] Ant Colony Optimization

هوش‌مصنوعی همانند یادگیری تقویتی، شبکه‌های عصبی، الگوریتم‌های ژنتیک و غیره انجام دهند. این کنترل کننده‌های مبتنی بر هوش‌مصنوعی این توانایی را دارند که تعدادی از توابع و اعمال کنترلی موجود در شبکه‌های کامپیوتری و مخابراتی را پیاده‌سازی نمایند. این روش‌های جدید وفقی می‌بایست منجر به قابلیت انطباق بالا شوند که امکان برآورده‌سازی نیازهای سرویس، چه از منظر کاربران و چه از منظر متصدیان شبکه، را داشته باشند.

یادگیری تقویتی یک روش یادگیری بدون سرپرست است که هدف از آن یادگیری یک سیاست، نگاشتی از مشاهدات به اعمال، بر مبنای بازخوردی است که از محیط دریافت می‌شود. این عمل یادگیری را می‌توان به صورت جستجوی مجموعه‌ای از سیاست‌ها نگریست که در هر لحظه در تعامل با محیط ارزیابی می‌شوند. در میان الگوریتم‌های مختلفی که برای یادگیری تقویتی تاکنون ارائه شده است الگوریتم یادگیری-Q دارای بیشترین کاربرد است. با به‌کارگیری الگوریتم یادگیری-Q می‌توان یک راهکار مسیریابی خودمختار به صورت وفقی بدست آورد [۵]. با این‌حال یادگیری-Q یک روش متمرکز است که برای یادگیری یک سیستم تک عامل به‌کار می‌رود. در این پایان‌نامه به شبکه به عنوان یک سیستم چندعاملی[5] نگریسته شده است که هر گره آن نشان دهنده یک عامل می‌باشد. سپس بر پایه روش یادگیری تقویتی و با استفاده از الگوریتم یادگیری-Q هر یک از عامل‌ها شروع به یادگیری وضعیت شبکه می‌نماید تا در هر حالت شبکه بهترین عمل ممکن را از خود بروز دهد. در این روش وضعیت هر گره (عامل) به صورت تابعی از وضعیت گره‌های مجاور و خطوط ارتباطی مابین آن‌ها تعریف می‌شود و بدین صورت هر تغییر در وضعیت یک گره (عامل) در وضعیت و نحوه تصمیم‌گیری گره‌های (عاملین) مجاور آن تأثیرگذار است.

۱-۲ روند ارائه پایان‌نامه

در ادامه مطالب پایان‌نامه و در فصل دوم به معرفی روش یادگیری تقویتی می‌پردازیم. در این فصل مسئله یادگیری تقویتی و مدل آن معرفی می‌شود. سپس راهبردهای موجود برای این مسئله بیان می‌گردند و مزایا و مشکلات که این روش دارد بررسی می شود. همچنین روش‌های مختلف جستجو و اکتشاف که از آن‌ها در یک سیستم یادگیر تقویتی می‌توان بهره برد ارائه می‌شوند. در نهایت از بین الگوریتم‌های مورد استفاده در مسائل یادگیری تقویتی، از الگوریتم یادگیری-Q یاد می‌کنیم وبه بیان دقیق عملکرد آن و نحوه یادگیری توسط این الگوریتم می‌پردازیم.

در فصل سوم مسئله مسیریابی در شبکه را مطرح می‌کنیم و ویژگی‌های مختلفی که انواع الگوریتم‌های مسیریابی را از هم متمایز می‌نماید معرفی می‌کنیم. سپس به معرفی الگوریتم‌های ارائه شده برای مسیریابی که

---

[5] Multi Agent

از روش‌های هوش‌مصنوعی در آن‌ها استفاده شده است می‌پردازیم و ویژگی‌ها و خصوصیات هر یک را به تفکیک بیان می‌کنیم.

در فصل چهارم به معرفی روش ارائه شده می‌پردازیم. در این روش شبکه را به عنوان یک سیستم چند عاملی مورد بررسی قرار داده‌ایم که در آن هر گره معادل یک عامل می‌باشد. هر عامل توسط الگوریتم یادگیری Q- آموزش می‌بیند تا رفتار شبکه را یاد گرفته و بتواند یک رفتار هوشمندانه مطابق با وضعیت شبکه از خود بروز دهد. در این فصل همچنین به چگونگی شبیه‌سازی این سیستم پرداخته می‌شود و نتایج حاصل از آن ارائه می‌گردد.

در نهایت در فصل آخر نتیجه‌گیری و پیشنهاداتی که برای ادامه کار مناسب به نظر می‌رسند ارائه می‌شوند.

فصل دوم

یادگیری تقویتی

۲-۱ مقدمه

یادگیری تحت سرپرستی[1] یک روش عمومی در یادگیری ماشین است که در آن به یک سیستم مجموعه جفت‌های ورودی – خروجی ارائه شده و سیستم تلاش می‌کند تا تابعی از ورودی به خروجی را فرا گیرد. پس از یک دوره یادگیری، می‌توان داده ورودی را به سیستم ارائه نمود و سیستم تلاش می‌نماید تا خروجی متناسب با آن را تولید کند.

یادگیری تحت سرپرستی نیازمند تعدادی داده ورودی به منظور *آموزش*[2] سیستم است. با این حال رده‌ای از مسائل وجود دارند که خروجی مناسب که یک سیستم یادگیری تحت سرپرستی نیازمند آن است، برای آن‌ها موجود نیست. برای نمونه، در مسائل کنترل پویا (برای مثال کنترل ترافیک خطوط هوایی) ممکن است که پاسخ‌های صحیح زیادی وجود داشته باشد، اما این پاسخ‌ها به راحتی قابل دسترسی و بدست آوردن نباشند. این نوع از مسائل چندان قابل جوابگویی با استفاده از یادگیری تحت سرپرستی نیستند.

در *یادگیری تقویتی*[3] *(RL)*، سیستم تلاش می‌کند تا تقابلات خود با یک محیط پویا را از طریق خطا و آزمایش بهینه نماید. یادگیری تقویتی مدلی برای مسائلی از این قبیل فراهم می‌آورد.

سابقه یادگیری تقویتی (RL) به روزهای اولیه سایبرنتیک[4] و کار در زمینه‌های آمار، روانشناسی، علوم عصبی[5] و علوم کامپیوتر برمی‌گردد. در ۱۰ تا ۱۵ سال گذشته، این زمینه توجه بسیاری را در انجمن‌های هوش‌مصنوعی و یادگیری ماشین[6] با سرعت زیاد به خود جلب نموده است [۷]. هدف اولیه یادگیری تقویتی

---

[1] Supervised Learning
[2] Train
[3] Reinforcement Learning
[4] Cybernetic
[5] Neuroscience
[6] Machine Learning

برنامه‌ریزی عامل‌ها[7] با استفاده از تنبیه و تشویق است بدون آنکه ذکری از چگونگی انجام وظیفه آن‌ها شود. با این حال موانع محاسباتی دشواری برای رسیدن به این هدف وجود دارد.

یادگیری تقویتی از بسیاری از جهات با مسئله یادگیری تحت سرپرستی متفاوت است. مهم‌ترین تفاوت، آن است که در یادگیری تقویتی هیچ نوع زوج ورودی/خروجی ارائه نمی‌شود. به جای آن، پس از اتخاذ یک عمل، حالت بعدی و پاداش بلافصل به عامل ارائه می‌شود. برای عامل لازم است که تجربیات مفیدی در مورد حالات ممکن سیستم، اعمال، انتقال‌ها و پاداش‌ها جمع‌آوری نماید تا بتواند به بهترین حالت عمل نماید. تفاوت دیگر با یادگیری تحت سرپرستی آن است که عملکرد همزمان در اینجا بسیار مهم است: ارزیابی سیستم معمولاً همزمان با یادگیری صورت می‌گیرد.

یادگیری تقویتی مسئله‌ای است که یک عامل که می‌بایست رفتار خود را از طریق تعاملات آزمایش و خطا با یک محیط پویا فرا گیرد، با آن مواجه است. دو راهکار اصلی برای حل مسائل یادگیری تقویتی وجود دارد. روش اوّل آن است که در فضای رفتارها به منظور یافتن بهترین رفتار در محیط مورد نظر جستجو نماییم. این راهکار در الگوریتم‌های ژنتیک و برنامه‌ریزی ژنتیک به‌کار گرفته شده است [۱۰]. روش دوم به‌کارگیری تکنیک‌های آماری و روش‌های برنامه‌ریزی پویا[8] به منظور تخمینِ منفعتّ اتخاذ هر عمل در حالات مختلف محیط یادگیری است. این فصل به معرفی تکنیک‌های روش دوم می‌پردازد. چرا که این روش‌ها از مزایای ساختار ویژه مسائل یادگیری تقویتی استفاده می‌کنند که در سایر مسائل بهینه‌سازی به طور کل موجود نیست.

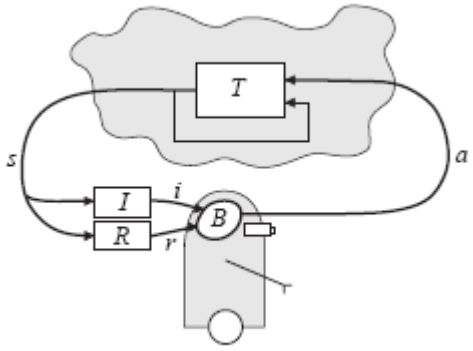
۲-۲ مدل یادگیری تقویتی

در مدل استاندارد یادگیری تقویتی، یک عامل از طریق ادراکات و اعمال خود با محیط در ارتباط است(شکل ۲-۱). در هر مرحله عامل به عنوان ورودی، i ، نشانه‌ای از حالت فعلی محیط، s ، را دریافت می‌کند. سپس عامل عمل خود، a ، را انتخاب نموده و به عنوان خروجی تولید می‌کند. این عمل حالت محیط را تغییر داده و مقدار این تغییر حالت از طریق اسکالر سیگنال تقویت، r ، به عامل منعکس می‌گردد. رفتار عامل (نحوه انتخاب یک عمل از میان اعمال ممکن) ، B، باید به‌گونه‌ای اعمال خود را انتخاب نماید که منجر به افزایش جمع مقادیر سیگنال تقویتیِ دریافتی در طول زمان طولانی اجرا شود. عامل می‌تواند این رفتار را در طول زمان از طریق آزمایش و خطای سیستماتیک که با الگوریتم‌های مختلفی قابل هدایت است، فرا گیرد.

---

[7] Agent

[8] Dynamic Programming



شکل 2-1: مدل استاندارد یادگیری تقویتی [۷]

مدل تشکیل شده است از :

- یک مجموعه گسسته از حالت‌های محیط، S
- یک مجموعه گسسته از اعمال عامل، A
- یک مجموعه از سیگنال‌های تقویت اسکالر؛ به طور معمول $\{۰ ، ۱\}$، و یا اعداد حقیقی.

دو تابع R و T به ترتیب تعیین کننده پاداش دریافتی عامل از انجام عمل a در حالت s و حالت بعدی که عامل در اثر این زوج حالت–عمل به آن منتقل می‌شود، هستند. شکل ۱ همچنین شامل یک تابع ورودی I نیز می‌شود که تعیین می‌کند که عامل چگونه حالات محیط را ببیند. فرض می‌شود که I یک تابع همانی باشد، بدین معنی که عامل دقیقاً چیزی را درک می‌کند که حالت محیط است.

یک روش شهودی برای فهمیدن ارتباط ما بین عامل و محیط آن در مثال زیر آمده است:

محیط: در حالت ٦٥ هستید. ٤ عمل ممکن دارید.

عامل: عمل ۲ را انجام می‌دهم.

محیط: شما ۷ واحد پاداش دریافت می‌کنید. اکنون در حالت ۱۵ هستید. ۲ عمل ممکن دارید.

عامل: عمل ۱ را انجام می‌دهم.

محیط: شما ٤– واحد پاداش دریافت می‌کنید. اکنون در حالت ۱۵ هستید. ۲ عمل ممکن دارید.

عامل: عمل ۲ را انجام می‌دهم.

محیط: شما ۵ واحد پاداش دریافت می‌کنید. اکنون در حالت ۱۵ هستید. ۲ عمل ممکن دارید.

⋮ ⋮

این مجموعه تراکنش‌های مابین عامل و محیط تا رسیدن عامل به حالت هدف و دستیابی به پاداش مناسب ادامه می‌یابد.

وظیفه عامل پیدا نمودن یک سیاست π است که حالات را به گونه ای به اعمال نگاشت می‌کند که یک مقیاس تقویت طولانی مدت را بیشینه نماید. به طور کلی انتظار می‌رود که محیط غیر قطعی[9] باشد، بدین

[9] Non Deterministic

معنی که اتخاذ یک عمل در یک حالت مشابه در دو موقعیت متفاوت ممکن است منجر به حالت بعدی و مقدار تقویت متفاوتی شود. این حالت در مثال فوق نیز اتفاق افتاده است: در حالت ٦٥ به‌کارگیری عمل ۲، در دو موقعیت متفاوت سیگنال تقویت و حالت بعدی متفاوتی را ایجاد کرده است. با این حال فرض می‌شود که محیط پایدار[10] باشد، بدین معنی که احتمالات انتقال حالت و یا دریافت یک سیگنال تقویت خاص در طول زمان تغییر نمی‌کند.

مسائل یادگیری تقویتی معمولاً به صورت *فرآیندهای تصمیم‌گیری مارکف* (MDP)[11] مدل می‌شوند. یک MDP تشکیل شده است از :

- یک مجموعه از حالات، S
- یک مجموعه از اعمال، A
- یک تابع پاداش $R: S \times A \to \Re$. $r = R(s,a)$ پاداش بلافصل مورد انتظار از انجام عمل a در حالت s است.
- یک تابع توزیع انتقال حالت: $T: S \times A \to \Pi(S)$ که در آن $\Pi(S)$ مجموعه‌ای از توزیع‌های تصادفی بر روی مجموعه S است. $T(s, a, s')$ به عنوان احتمال انتقال از حالت s به حالت s' با اتخاذ عمل a در نظر گرفته می‌شود.

سیستم ممکن است شامل یک سری حالت نهایی[12] باشد. یک حالت نهایی F در یک MDP به صورت حالتی بیان می‌شود که در آن هر عمل انتقال به حالت F دارای پاداش ۰ است. یک MDP جذب‌کننده[13] گونه‌ای است که در آن از هر حالت غیرنهایی بتوان سرانجام به یک حالت نهایی رسید.

یک مدل *مارکف* است اگر انتقال حالات تنها به حالت فعلی عامل بستگی داشته و مستقل از حالات و اعمال قبلی وی باشد.

۲-۳ تعریف سیاست بهینه

پاداش بلافصل یک عمل نتایج آتی آن عمل را در نظر نمی‌گیرد، اگرچه که ممکن است آن عمل اثر بزرگی بر روی عملکرد کلی سیستم داشته باشد. مسئله ارزیابی اعمال از طریق پاداش تأخیر یافته مربوط به *مسئله تخصیص اعتبار زمانی*[14] می‌شود. سیاست بهینه نیازمند آن است که علاوه بر نتیجه بلافصل عمل، نتایج آتی آن نیز در نظر گرفته شود.

---

[10] Stationary
[11] Markov Decision Process
[12] Terminal State
[13] Absorbing MDP
[14] Temporal Credit Assignment Problem

۲-۳-۱ مدل پاداش دراز مدت

لازم است که رفتار بهینه هر عامل تعریف شود. تعریف سیاست بهینه از ارزیابی عملکرد درازمدت عامل تشکیل می‌شود. تعدادی مدل استاندارد وجود دارد که برای تعریف این عملکرد به‌کار برده می‌شوند. این مدل‌ها تعیین می‌کنند که اثرات آتی عمل تا چه میزان باید در نظر گرفته شود، و آیا پاداش‌هایی که زودتر دریافت می‌شوند می‌بایست از ارزش بیشتری نسبت به پاداش‌هایی که دیرتر دریافت می‌شوند برخوردار باشند.

مدل *افق محدود*[15] به سادگی، میزان عملکرد یک عامل را به صورت جمع پاداش مورد انتظار در h مرحله بعد ارزش دهی می‌کند :

$$E\left(\sum_{t=0}^{h} r_t\right) \qquad (۱-۲)$$

که در آن r_t بیانگر پاداش دریافت شده در t مرحله بعد در آینده است. این مدل پاداشی را که در $h+1$ مرحله بعد و بعد از آن دریافت می‌شود در نظر نمی‌گیرد.

مدل افق نامحدود کاهش‌یافته[16] پاداش‌های دراز مدت یک عامل را در نظر می‌گیرد، اما این پاداش‌هایی که در آینده دریافت می شوند طبق یک فاکتور کاهش $0 < \gamma \leq 1$ به صورت هندسی کاهش می‌یابند:

$$E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \qquad (۲-۲)$$

این مدل به یک حاصل‌جمع همگرا شونده با توجه به پاداش نامحدود در آینده منجر می‌شود، در عین این که به پاداش‌های دریافت شده در آینده نزدیک تاکید بیشتری دارد.

۲-۳-۲ تابع ارزش بهینه و سیاست بهینه

در یک مدل با عملکرد بهینه‌ی از پیش تعریف شده، منظور از *مقدار بهینه*[17] یک حالت، عملکرد مورد انتظار یک عامل است در صورتی که از آن حالت شروع کرده و یک سیاست بهینه را اجرا کند. به عنوان مثال، با استفاده از مدل کاهش یافته و نشان دادن یک سیاست تصمیم‌گیری کامل با π، تعریف می‌کنیم:

$$V^*(s) = \max_{\pi} E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \qquad (۳-۲)$$

$V^*(s)$ مقدار بهینه s است. این تابع یکتا بوده و راه‌حل *معادلات Bellman* است:

$$V^*(s) = \max_{a}\left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V^*(s')\right) \qquad (۴-۲)$$

که در آن $T(s,a,s')$ و $R(s,a)$، *مدل سیستم* هستند.

---

[15] Finite Horizon [16] Infinite Horizon Discounted [17] Optimal Value

با توجه به رابطه (۲-۴) ، *سیاست بهینه* $\pi^*(s)$ به صورت زیر می‌تواند بیان شود:

$$\pi^*(s) = \arg\max_a \left(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V^*(s') \right) \qquad (۵-۲)$$

۲-۳-۳ الگوریتم‌های تکرار مقدار و تکرار سیاست

برای یافتن تابع مقدار بهینه و پس از آن سیاست بهینه، می‌توان از یک الگوریتم ساده تکراری که تکرار مقدار[۱۸] نام دارد استفاده نمود. در [11،12] نشان داده شده است که الگوریتم تکرار مقدار به مقادیر صحیح $V^*(s)$ همگرا خواهد شد. الگوریتم تکرار مقدار در جدول (۲-۱) نشان داده شده است.

جدول2-1 : الگوریتم تکرار مقدار

Value Iteration algorithm

initialize $V(s) \forall s \in S$ arbitrary
repeat
  for all $s \in S$ **do**
  for all $a \in A$ **do**
  $Q(s,a) \leftarrow R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V(s')$
  end for
  $V(s) \leftarrow \max_a Q(s,a)$
  end for
until policy is good enough

در برابرِ تکرار مقدار، که یک سیاست بهینه را به صورت غیر مستقیم با استفاده از تابع مقدار بهینه پیدا می‌کند، الگوریتم تکرار سیاست[۱۹] وجود دارد که به صورت مستقیم بر روی یک سیاست کار می‌کند. Puterman در [13] نشان داده است که الگوریتم تکرار سیاست در حداکثر تعداد نمایی تکرار خاتمه می‌یابد. الگوریتم تکرار سیاست در جدول (۲-۲) آورده شده است.

[18] Value Iteration [19] Policy Iteration

جدول2-2 : الگوریتم تکرار سیاست

Policy Iteration algorithm

choose an arbitrary policy
repeat
  for all $s \in S$ **do**
  $V(s) \leftarrow R(s,\pi(s)) + \gamma \sum_{s' \in S} T(s,\pi(s),s')\, V(s')$
  end for
  for all $s \in S$ **do**
  $\pi'(s) \leftarrow \arg\max\, (R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')\, V(s'))$
  end for
  $\pi \leftarrow \pi'$
until no further improvement is possible

۲-۴ راهبردهای یادگیری برای مسائل یادگیری تقویتی

همانگونه که در قسمت ۲-۳-۳ نشان داده شد، سیاست بهینه یک عامل را می توان با حل معادلات Bellman بدست آورد. اگر چه حل کامل این معادلات دستیابی به سیاست بهینه را تضمین می‌کند، با این حال روش‌های تخمینی وجود دارند که می‌توانند رفتارهای نزدیک به بهینه را بدون نیاز به محاسبه کامل تابع ارزش بهینه بدست آورند.

از این روش‌های تخمین به عنوان *راهبردهای یادگیری* یاد می‌شود. این راهبردها تعیین می‌کنند که هنگامی که عامل در حال یادگیری است، چه تجربیاتی در سیستم نگهداری شوند و اطلاعات جمع‌آوری شده از این تجربیات چگونه برای هدایت یادگیری و عملکرد بهتر به کار برده شوند.

۲-۴-۱ راهبردهای مدل-مبنا و مدل-آزاد

الگوریتم‌های یادگیری تقویتی را می‌توان به دو رده کلی تقسیم نمود: آنهایی که تلاش می‌کنند تا یک *مدل* از سیستم را فرا گیرند و آنهایی که نیاز به یادگیری مدل سیستم ندارند. هر دو رده روش های برمبنای مدل و بی‌نیاز از مدل قادر به پیدا نمودن سیاست بهینه هستند. روش‌های بی‌نیاز از مدل معمولاً نیاز به زمان محاسباتی کمتری در هر تکرار الگوریتم دارند، اما در این روش ها تعداد تکرار بیشتری برای رسیدن (نزدیک شدن) به سیاست بهینه لازم است.

در یک راهکار برمبنای مدل، سیستم تلاش می‌کند تا احتمالات انتقال حالت و پاداش‌ها را فرا گیرد. احتمال رفتن به حالت s' هنگامی که در حالت s هستیم وعمل a را انجام می‌دهیم را با $T(s,a,s')$ نشان می‌دهیم. پاداش انجام عمل a و رفتن به حالت s' در هنگامی که در حالت s هستیم با $R(s,a,s')$ نشان داده می‌شود. به زوج توابع T و R، *مدل تخمین زده* گفته می‌شود.