Machines Learning and Networks Neural

ترجمه فصل ۱۲ - برنامه نویسی پویا

نویسنده:

امین خانی
aminkhani@std.kashanu.ac.ir

مریم رضوانی
mrymrzvni@gmail.com

فروردین ۱۴۰۲

فهرست مطالب

فصل ۱۲

برنامه نویسی پویا - Dynamic Programming

مرور کلی

سه هدف این فصل:

۱. برای بحث در مورد توسعه برنامه نویسی پویا به عنوان مبنای ریاضی برنامه ریزی یک دوره عمل چند مرحله ای توسط عاملی که در یک محیط تصادفی کار می کند.

۲. برای بیان یک اشتقاق مستقیم از یادگیری تقویتی بعنوان یک شکل تقریبی از برنامه نویسی پویا.

۳. ارائه روش های غیر مستقیم برنامه نویسی پویا تقریبی برای مقابله با نفرین ابعاد.

این فصل بصورت زیر طبقه بندی شده است:

۱. بخش ۱.۱۲ ، بخش مقدماتی، برانگیختن انگیزه مطالعه برنامه نویسی پویا است، بحث دررابطه با فرایندهای تصمیم گیری مارکوف که در بخش ۲.۱۲ انجام شده است.

۲. بخش ۳.۱۲ از طریق ۵.۱۲ مطرح کردن نظریه برنامه نویسی پویا بلمن و دو روش مرتبط: تکرار سیاست و تکرار ارزش.

۳. بخش ۶.۱۲ منطق پشت تقریب مبتنی بر یادگیری مستقیم را توصیف می کند.برنامه نویسی پویا بدان وسیله منجربه توسعه یادگیری تفاوت های زمانی و یادگیری Q می شود.

۴. بخش ۹.۱۲ منطق پشت تقریب مبتنی بر یادگیری غیرمستقیم را توصیف می کند. برنامه نویسی پویا برای مقابله با مشکل نفرین ابعاد. بدان وسیله منجر به بررسی ارزیابی سیاست حداقل مربعات و بازگویی ارزش تقریبی ارئه شده به ترتیب دربخش ۱۰.۱۲ و ۱۱.۱۲ ارائه شده است.

این فصل با خلاصه و بحث در بخش ۱۲.۱۲ به پایان می رسد.

۱.۱۲ مقدمه

در فصل مقدماتی، ما دو الگوی اصلی یادگیری را شناسایی کرده ایم. یادگیری همراه با معلم و یادگیری بدون معلم. الگوی یادگیری بدون معلم به یادگیری خود سازمان یافته (بدون نظارت) و یادگیری تقویتی تقسیم می شود. اشکال مختلف یادگیری با معلم، یا یادگیری تحت نظارت، در فصل های ۱ تا ۶ و اشکال مختلف یادگیری بدون نظارت در فصل های ۹ تا ۱۱ مورد

بحث قرار گرفت. یادگیری نیمه نظارتی در فصل ۷ مورد بحث قرار گرفت. در این فصل، یادگیری تقویتی را مورد بحث قرار می دهیم.

یادگیری تحت نظارت یک مشکل یادگیری «شناختی» است که تحت نظارت یک معلم انجام می شود. این متکی به در دسترس بودن مجموعه ای کافی از نمونه های ورودی_خروجی است که نماینده محیط عملیاتی هستند. در مقابل، یادگیری تقویتی یک مشکل یادگیری «رفتاری» است. این از طریق تعامل بین یک عامل و محیط آن انجام می شود، که در آن عامل یا تصمیم گیرنده به دنبال دستیابی به یک هدف خاص علیرغم وجود عدم قطعیت است (بارتو و همکاران، ۱۹۸۳؛ ساتون و بارتو، ۱۹۹۸). این واقعیت که این تعامل بدون معلم انجام می‌شود، یادگیری تقویتی را به ویژه برای موقعیت‌های پویا جذاب می‌کند، جایی که جمع‌آوری مجموعه‌ای رضایت‌بخش از نمونه‌های خروجی ورودی پرهزینه یا دشوار (اگر نه غیرممکن) است.

دو رویکرد برای مطالعه یادگیری تقویتی وجود دارد که به طور خلاصه به شرح زیر است:

۱. رویکرد کلاسیک، که در آن یادگیری از طریق فرآیند تنبیه و پاداش، با هدف دستیابی به یک رفتار بسیار ماهرانه صورت می‌گیرد.

۲. رویکرد مدرن، که بر پایه یک تکنیک ریاضی معروف به برنامه‌نویسی پویا استوار است تا با در نظر گرفتن مراحل احتمالی آینده بدون تجربه واقعی، در مورد یک دوره عمل تصمیم بگیرد. در اینجا تاکید بر برنامه ریزی است.

بحث ما بر یادگیری تقویتی مدرن متمرکز است. برنامه نویسی پویا تکنیکی است که به موقعیت هایی می پردازد که در آن تصمیمات به صورت مرحله ای گرفته می شوند و نتیجه هر تصمیم قبل از اتخاذ تصمیم بعدی تا حدی قابل پیش بینی است. یکی از جنبه‌های کلیدی چنین موقعیت‌هایی این است که نمی‌توان تصمیم‌ها را به تنهایی اتخاذ کرد. در عوض، میل به هزینه کم در حال حاضر باید در برابر نامطلوب بودن هزینه بالا در آینده متعادل شود. این یک مشکل تخصیص اعتبار است، زیرا اعتبار یا سرزنش باید به هر یک از مجموعه ای از تصمیمات متقابل اختصاص داده شود. برای برنامه‌ریزی بهینه، لازم است که بین هزینه‌های فوری و آتی یک معاوضه کارآمد وجود داشته باشد. چنین مبادله ای در واقع با فرمالیسم برنامه نویسی پویا تسخیر شده است. به طور خاص، برنامه نویسی پویا به مشکل اساسی زیر می پردازد:

چگونه یک عامل یا تصمیم گیرنده می تواند عملکرد بلندمدت خود را در یک محیط تصادفی بهبود بخشد، در حالی که دستیابی به این بهبود ممکن است نیاز به قربانی کردن عملکرد کوتاه مدت داشته باشد؟

برنامه نویسی پویا بلمن راه حلی بهینه برای این مشکل اساسی به شیوه ای ظریف و اصولی ارائه می دهد.

در هنر مدل سازی ریاضی، چالش ایجاد تعادل مناسب بین دو نهاد است، یکی عملی و دیگری نظری. به ترتیب، این دو نهاد هستند

- توصیف واقع بینانه یک مسئله داده شده و
- قدرت روش های تحلیلی و محاسباتی برای اعمال مشکل.

در برنامه نویسی پویا، موضوع مورد توجه ویژه تصمیم گیری توسط عاملی است که در یک محیط تصادفی عمل می کند. برای پرداختن به این موضوع، مدل خود را حول فرآیندهای تصمیم مارکوف می سازیم. با توجه به وضعیت اولیه یک سیستم پویا، یک فرآیند تصمیم گیری مارکوف مبنای ریاضی را برای انتخاب دنباله ای از تصمیمات فراهم می کند که بازده از یک فرآیند تصمیم گیری مرحله N را به حداکثر می رساند. آنچه که ما توضیح دادیم ماهیت برنامه نویسی پویا بلمن است. بنابراین مناسب است که مطالعه برنامه نویسی پویا را با بحث در مورد فرآیندهای تصمیم مارکوف آغاز کنیم.

۲.۱۲ فرآیند تصمیم گیری مارکوف

عامل یا تصمیم گیرنده ای را در نظر بگیرید که با محیط خود به روشی که در شکل ۱.۱۲ نشان داده شده است در تعامل است. عامل مطابق با یک فرآیند تصمیم گیری مارکویی زمان محدود عمل می کند که به شرح زیر مشخص می شود:

- محیط به صورت احتمالی تکامل می‌یابد و مجموعه‌ای محدود از حالات گسسته را اشغال می‌کند. با این حال، ایالت حاوی آمارهای گذشته نیست، حتی اگر این آمارها می تواند برای عامل مفید باشد.
- برای هر مرحله محیطی، مجموعه محدودی از اقدامات احتمالی وجود دارد که ممکن است توسط عامل انجام شود.
- هر بار که نماینده اقدامی انجام می دهد، هزینه خاصی متحمل می شود.
- ایالت ها مشاهده می شوند، اقداماتی انجام می شود و هزینه ها در زمان های مجزا متحمل می شوند.

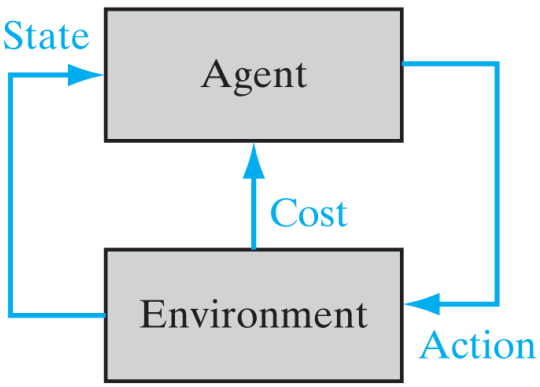
در چارچوب بحث حاضر، تعریف زیر را معرفی می کنیم:

وضعیت محیط خلاصه ای از کل تجربه گذشته یک عامل است که از تعامل آن با محیط به دست می آید، به طوری که اطلاعات لازم برای عامل برای پیش بینی رفتار آینده محیط در آن خلاصه موجود است.

حالت در مرحله زمانی n با متغیر تصادفی X_n و وضعیت واقعی در مرحله زمانی n با i_n نشان داده می شود. مجموعه حالت های محدود با x نشان داده می شود. یکی از جنبه های شگفت انگیز برنامه نویسی پویا این است که کاربرد آن به ماهیت وضعیت بسیار کمی بستگی دارد. بنابراین ممکن است بدون هیچ گونه فرضی در مورد ساختار فضای حالت پیش برویم. همچنین توجه داشته باشید که پیچیدگی الگوریتم پویاـ برنامه ریزی در بعد فضای حالت درجه دوم و در بعد فضای عمل خطی است.

برای مثال، برای حالت i، مجموعه اقدامات موجود (یعنی ورودی‌های اعمال شده توسط عامل به محیط) با نشان داده می‌شود، که در آن زیرنویس دوم k در عمل a_{ik} که توسط عامل انجام می‌شود، صرفاً در دسترس بودن بیش از یک عمل ممکن را نشان می‌دهد. زمانی که محیط در حالت i است. انتقال محیط از حالت i به حالت جدید j به عنوان مثال به دلیل عمل a_{ik} ماهیت احتمالی دارد. اما مهمتر از همه، احتمال انتقال از حالت i به حالت j کاملاً به وضعیت فعلی i و عملکرد مربوطه a_{ik} بستگی دارد. این ویژگی مارکوف است که در فصل ۱۱ مورد بحث قرار گرفت. این ویژگی بسیار مهم است زیرا به این معنی است که جریان فعلی وضعیت محیط اطلاعات لازم را برای عامل فراهم می کند تا تصمیم بگیرد چه اقدامی انجام دهد.

متغیر تصادفی که بیانگر عمل انجام شده توسط عامل در مرحله n زمان است با A_n نشان داده می شود. اجازه دهید $p_{ij}(a)$ احتمال انتقال از حالت i به حالت j را نشان دهد



شکل ۱.۱۲: بلوک دیاگرام یک عامل در تعامل با محیط خود.

اقدام انجام شده در مرحله زمانی n، جایی که $A_n = a$. به موجب فرض مارکف در مورد پویایی حالت، ما داریم

$$p_{ij}(a) = P(X_{n+1} = j | X_n = i, A_n = a) \tag{۱.۱۲}$$

۳.۱۲ معیار بهینه بودن معادله بلمن

متن تست

۴.۱۲ تکرار سیاست

۵.۱۲ تکرار ارزش

۶.۱۲ برنامه نویسی دینامیک تقریبی: روش های مستقیم

۷.۱۲ یادگیری تفاوت زمانی

۸.۱۲ یادگیری-Q

۹.۱۲ برنامه نویسی دینامیک تقریبی: روش های غیر مستقیم

۱۰.۱۲ ارزیابی سیاست حداقل مربعات

۱۱.۱۲ تکرار سیاست تقریبی

۱۲.۱۲ خلاصه و بحث