



نام و نام خانوادگی: فاطمه حدادی

نام استاد: دکتر هدیه ساجدی

شماره دانشجویی: ۶۱۰۳۹۶۰۸۷

آبان ۹۹

گزارش تمرین حل پازل با یادگیری تقویتی به روش Q-learning

• چکیده

در این گزارش، خلاصه ای از مراحل الگوریتم یادگیری تقویتی Q-learning برای حل پازل ارائه میشود. برنامه ای به زبان پایتون برای پیاده سازی این الگوریتم و حل پازل نوشته شده است. با پیمودن تعداد مراحل کافی یادگیری، این برنامه توانسته است با کمترین حرکت ممکن پازل را حل کند.

فهرست مطالب

۱. اهداف تمرین

۲. شرح بخش های برنامه

۳. توضیح خروجی برنامه و حل پازل

۱. اهداف تمرین

اهداف این فاز مختصرا پیاده سازی الگوریتم از تئوری به عمل است. همچنین نمایش عملکرد این الگوریتم یادگیری برای حل یک پازل. در حوزه هوش مصنوعی علاوه بر پایه تئوری قوی و توانایی بالا در مفاهیم ریاضی، کدزنی و پیاده سازی الگوریتم ها جایگاه ویژه ای دارد. سرعت واقعی یک الگوریتم با اجرای عملی آن قابل مشاهده است و پارامترهای تاثیرگذار را میتوان با اجراهای پی در پی به نحوه خوبی تنظیم کرد.

۲. شرح بخش های برنامه

مطابق مطالب ارائه شده در کلاس، یادگیری تقویتی به روش Q-learning در واقع ساختن یک جدولی از مقادیر پاداش با هر عمل ممکن در هر وضعیت ممکن است. این پاداش ها راهنمایی کارآمد برای حل مسئله مورد نظر است. این ساختار Q نامیده میشود که دلیلی برای نام گذاری این الگوریتم است. پر کردن جدول Q با استفاده از مقادیر قبلی جدول و مسیر حرکت به سمت حالت نهایی انجام میشود. در واقع ابتدا به تعداد بالایی پازل تغییر میکند و متناظر با آن مقدار عناصر Q بروزرسانی میشود. پارامترهای موثر در این بروزرسانی مقدار قبلی خانه و حرکت های آینده است.

این برنامه دو تابع اصلی به نام های play و run_Qlearning دارد. تابع run_Qlearning یادگیری را انجام میدهد و تابع پازل را طبق نتایج به دست آمده از یادگیری حل میکند. در تابع run_Qlearning ابتدا ساختمان داده های Q و R و dic ساخته و مقداردهی اولیه میشود. سپس مسئله به اندازه ۶۰۰ بار مراحل حل را طی میکند و ابتدا تابع select_move اجرا میشود که در آن حرکت در این مرحله را به ما میدهد. برای شناسایی خانه خالی

در پارل از تابع `find_empty_index` استفاده شده است. برای به دست آوردن حرکت بعدی، به صورت رندوم یکی از حرکت های ممکن انتخاب میکند. تابع `make_move` حرکت را انجام میدهد و صفحه بازی جدید را برمیگرداند. لازم به ذکر است صفحه حرکت به صورت یک رشته شبیه سازی است به طوری که نشان های با حروف `m,s` و `o` که حرف اول هر نشان است. تابع `update_Q` مقدار خانه های `Q` را با توجه به حرکت انتخابی بروزرسانی میکند.

در تابع `play` برای رفتن به هر حالت و رسیدن به فرم مطلوب از تابع `next_move` استفاده میشود که در آن طبق جدول `Q` حرکتی که از آن حالت از صفحه بیشترین پاداش را به دست می آورد در هر مرحله انتخاب میشود. تابع `display` وضعیت صفحه بازی در هر مرحله را در خروجی نمایش میدهد.

۳. توضیح خروجی برنامه و حل پازل

خروجی الگوریتم یادگیری تقویتی `Q-learning` همان جدول `Q` است که با آن میتوان پازل را به خوبی حل کرد. این جدول همه ی حالت ها و حرکت های ممکن در هر حالت را پوشش میدهد پس یک ابزار مفید در حل مسئله با سرعتی بسیار بالا خواهد بود. پس تنها جدول `Q` برای حل مسئله متناظر با آن کافی است.

در این برنامه جدول `Q` بعد از مرحله یادگیری در خروجی چاپ میشود. سپس با استفاده از این جدول در عمل برنامه بازی شده و با ۱۳ حرکت به جواب میرسد که در اخر برنامه نمایش میشود. همچنین وضعیت صفحه بازی به عنوان مسیر حل مسئله در خروجی به صورت ماتریس نمایش داده میشود که نشان هر خانه حرف اول اسم آن نشان به زبان انگلیسی است.

خروجی در قالب یک فایل ضمیمه شده است.