

به نام خدا



پروژه اول

درس: مبانی هوش مصنوعی

اعضا:

علی پورقیصری

طاها داوری

محمد امین مولوی زاده

توضیحات:

این کد پیاده سازی یک الگوریتم فرآیند تصمیم گیری مارکوف (MDP) است. در ابتدا کتابخانه های مورد نیاز را اضافه می کنیم. حال از کتابخانه `gymnasium` یک محیط تصادفی انتخاب می کنیم که با استفاده از یک سری پارامترها تعریف می شود. در حین تعریف محیط تصادفی، لغزندگی را در نظر میگیریم که باعث احتمال انتخاب یک سوم از خانه های رو به رو و یا اطراف عامل می شود. اگر پارامتر لغزندگی را در نظر نگیریم می بینیم که عامل هر بار از یک مسیر ثابت (غیر از اوایل فرایند که هنوز مقادیر همگرا نشده) به طرف هدف حرکت می کند و به هدف می رسد. پس از پیاده سازی محیط، پارامترهای مربوطه را هم تنظیم می کنیم.

سپس ماتریس تک بعدی `env.unwrapped.P` که حرکت های موجود برای حرکت کردن از یک خانه به خانه دیگر را در خود دارد و مقادیری مانند احتمال انتخاب آن خانه، مقصد، جایزه و اتمام فرایند در آن ذکر شده، را ذخیره می کنیم تا بعدا بتوان از آن استفاده کرد. همچنین یک ماتریس به اندازه محیط برای ارزش و جایزه هر خانه مشخص می کنیم که در ابتدا همه خانه ها به غیر از خانه آخر مقدار صفر را در خود دارند.

سپس یک آرایه تک بعدی از نوع `numpy` تعریف می کنیم که در ادامه برای تبدیل شماره خانه به اندیس مورد استفاده قرار می گیرد. این ماتریس برای کارایی بیشتر تعریف شده زیرا تعداد دفعاتی که نیاز داریم شماره خانه را به اندیس تبدیل کنیم بسیار است و محاسبه آن زمان بر می شود ولی با این کار یک بار محاسبات انجام می شود. در حلقه اصلی برنامه از دو تابع `MDP()` و `get_action()` استفاده می کنیم که کار هر کدام به صورت زیر مشخص می شود:

تابع `MDP()`:

این تابع، تابع اجرای فرایند تصمیم مارکوف است که بدین صورت عمل می کند که ابتدا دو حلقه برای پیمایش ماتریس از آخر به اول پیاده سازی می شوند و در درون هر حلقه مشخص می شود که آیا خانه هدف است یا نه، اگر خانه هدف بود مقدار آن که ۱ در آن جایگذاری شده بدون تغییر باقی می ماند. اگر خانه، چاله بود، ارزش آن خانه صفر است و اگر آن خانه هیچ کدام از این دو نبود ارزش آن به این صورت محاسبه می شود که هر ۴ حالت بالا، پایین، چپ و راست اجرا شده و مقدار ارزش آنها حساب می شود و در یک لیست به نام `state_list` ریخته می شود. در آخر بزرگ ترین عدد بین این ۴ حالت انتخاب می شود و در خانه متناظر آن در ماتریس ارزش ها ریخته می شود. توجه شود که ارزش هر خانه با استفاده از فرمول `Bellman` به دست می آید که احتمال هر خانه در ارزش آن ظرب شده و عدد حاصل در یک متغیر (`gama`) که در اینجا ۰/۹ در نظر گرفته شده است ضرب می شود.

تابع `get_action()`:

این تابع، برای انتخاب حرکت می باشد، به این صورت که ارزش خانه های اطراف مکانی که عامل در آن است را گرفته و آن را در یک دیکشنری به عنوان مقدار می ریزد. عبارات `if` بررسی می کنند که آیا خانه های همسایه در جهت شمال، غرب، جنوب یا شرق وجود دارد یا خیر، و مقادیر امتیاز مربوط به آن ها را اضافه می کنند. سپس بزرگ ترین ارزش را به همراه کلید آن که همان عدد متناظر با آن حرکت است را بر می گرداند و آن را به `env.step()` می دهد تا حرکت مورد نظر را به احتمال یک سوم اجرا کند.

سپس در آخر هر اجرا در چاله افتادن یا به هدف رسیدن عامل بررسی می شود و با توجه به این بررسی محیط باز نشانی می شود.

مقادیر واقع در ماتریس ارزش ها در ابتدا مقادیری غیر دقیق هستند ولی با چند دور تکرار این مقادیر به مقادیر اصلی همگرا می شوند.

توجه شود که در صورت عدم تفهیم درست به کد مراجعه شود. کد به صورت کامل کامنت گذاری شده است.