محمدحسین صباغی ۹۹۲۳۰۴۴

گزارش پروژه ۳ هوش مصنوعی

۱)

سوال : حداقل ۳ مورد از مشکلات روش Value Iteration را نام برده و توضیح دهید.

همگرا شدنش بسیار کند است و policy ها زودتر همگرا میشوند.

محاسبات زیادی دارد زیرا در هر مرحله تمام استیت ها را چک میکنیم.

ممکن است به اطلاعاتی از جمله استیت ها و احتمالات و ریوارد ها دسترسی نداشته باشیم.

۲)

سوال: دلیل انتخاب این مقادیر را به زبان ساده و به صورت شهودی توضیح دهید.

زیرا با کم شدن نویز تا میل کردن به صفر(یا حتی خود صفر) باعث میشود که با انتخاب هر جهتی ایجنت دقیقا به یک استیت خاص برود و به همین دلیل مقدار q انتخاب هایی که به سمت امتیاز ۱۰ میروند زیاد میشود و باعث میشود ایجنت سیاستش را به سمت آن رفتن بگذارد.

سوال: چگونه به این نتیجه رسیدیم که در حل مسائل با روش Value Iteration از Discount Factor استفاده میکنیم و این فاکتور چه کمکی به ما میکند؟ (میتوانید از نتایجی که در این بخش گرفتید نیز کمک بگیرید و به کمک آن‌ها توضیح دهید).

زیرا به دلیل وجود نویز بهتر است در حرکات کمتری امتیاز را دریافت کنیم زیرا با حرکات بیشتر احتمال خطا رفتن ایجنت به دلیل نویز بیشتر میشود. بنابراین با گذاشتن تخفیف ایجنت را تشویق میکنیم که زودتر امتیاز را دریافت کند.

سوال: راه حل Value Iteration راهی زمانبر است که باید برای هر State همه حالت‌ها را بسنجیم و گاهی ناگزیر به انجام آن هستیم. اما در این مسئله به خصوص آیا راه حل ساده‌تری نسبت به Value Iteration وجود دارد که تعداد حالت‌های بررسی شده را کاهش دهد؟ این روش را نام ببرید و توضیح دهید و سپس آن ها را از نظر پیچیدگی زمانی مقایسه کنید.

بله در این مسیله استفاده از policy iteration بهتر است زیرا مشخص است که حرکت افقی و به سمت شرق بهتر است و با قرار دادن آن برای اکشن های اولیه میتوان سریعتر مسیله را حل کرد.

همانطور که مشخص است همگرایی این روش سریعتر است بنابراین از نظر زمانی پیچیدگی کمتری دارد.

۳)

سوال: دلیل انتخاب خود برای هریک از مقادیر پارامترهای مذکور را در هر سیاست بیان کنید.

برای حالت اول مشخص است که با کم یا حتی صفر کردن نویز از سمت صخره ها میرود زیرا بعید است اشتباه برود و همچنین با زیاد کردن امتیاز منفی زنده ماندن و کاهش تخفیف باعث میشود که به سراغ ۱ که نزدیکتر است برود.

برای حالت دوم با زیاد کردن نویز احتمال خطا رفتن از سمت صخره ها زیاد میشود و ترجیح میدهد که از بالا برود و از صخره ها دوری کند.

در حالت سوم با کاهش نویز و افزایش تخفیف باعث میشود که از پایین برود و به سمت امتیاز بیشتر برود زیرا تاثیر تخفیف کم است و در نتیجه امتیاز ۱۰ خیلی کم نمیشود.

برای حالت چهارم نویز را بیشتر میکنیم تا به خاطر احتمال خوردن به صخره از پایین نرود.

برای حالت پنجم با مثبت کردن امتیاز زنده موندن و صفر کردن نویز باعث میشود که تا بینهایت زنده بماند و امتیاز دریافت کند.

سوال: در سیاست پنجم، همانطور که مشاهده کردید در یک لوپ بینهایت می‌افتادیم و عامل علاقه‌ای به پایان بازی نداشت. برای حل این مشکل چه راه حل هایی به نظرتان میرسد. آن‌ها را توضیح دهید.

باید امتیاز زنده ماندن را منفی کنیم. یا متوانیم تا یک عمق محدودی را تحلیل کنیم که با اینکار ایجنت فقط تا یک تعداد حرکت خاصی را پیش بینی میکند و ترجیح میدهد امتیاز بزرگ پایان بازی را دریافت کند تا اینکه امتیاز های کم زنده ماندن تا حرکات محدود را دریافت کند.

سوال: آیا استفاده از الگوریتم تکرار ارزش تحت هر شرایطی به همگرایی می‌انجامد؟

اگر مقدار تخفیف برابر یک نباشد قطعا همگرا میشود زیرا از یک تعداد حرکتی به بعد اینقدر مقدار تخفیف کم میشود که ریوارد به صفر میل میکند و همگرا میشود.

۴)

سوال: روش‌های بروزرسانی‌ای که در بخش اول (بروزرسانی با استفاده از batch) و در این بخش (بروزرسانی به صورت تکی) پیاده کرده‌اید را با یکدیگر مقایسه کنید. (یک نکته مثبت و یک نکته منفی برای هرکدام)

مزیت batch و عیب راه دوم : راه اول پیچیدگی کمتری دارد و تعداد تحلیل تمام استیت ها با هم برابر است.

عیب اولی و مزیت دومی: در حالت دوم امتیازها سریعتر همگرا میشوند زیرا با آپدیت شدن هر استیت سریعتر تاثیرش را روی بقیه استیت ها میگذارد و نیاز نیست تا یک دور کامل صبر کند.

۶)

سوال: توضیح دهید که اگر مقدار Q برای اقداماتی که عامل قبلا ندیده، بسیار کم یا بسیار زیاد باشد چه اتفاقی می‌افتد.

میتواند باعث شود در اپیزود های بعدی از رفتن به آن استیت ها خودداری کند یا اینکهبیشتر به سمت آن استیت ها برود و در هر دو مورد باعث میشود استیت های دیگری کمتر ویزیت شوند یا اصلا ویزیت نشوند.

سوال: بیان کنید Q-learning یک الگوریتم Off-policy است یا On-policy؟ Value-based است یا Policy-based ؟ توضیح دهید.

Off-policy است زیرا policy که آپدیت میشود باpolicy رفتار متفاوت است.یعنی به دنبال بهینه تر کردن یک policy نیستیم بلکه به دنبال به دست آوردن یک policyهستیم.

Value-based است زیرا به دنبال به دست آوردن q value ها و استفاده از آنها برای تصمیم گیری میباشد.

سوال: الگوریتم Q-leaning از TD-Leaning استفاده میکند آن را با Monte Carlo مقایسه کنید و بیان کنید استفاده هر کدام چه مزایا و چه معایبی دارند.

در td یا temporal difference بعد از هر گام ولیو ها را آپدیت میکنیم ولی درmonte carlo پس از اتمام هر اپیزود آپدیت میکنیم.

مزیت td اینست که سریعتر ولیوهای ما آپدیت میشوند و در نتیجه زودتر همگرا میشوند و همچنین از شرایط مارکوف طبعیت میکند یعنی امتیاز هر استیت صرفا به حال و اکشن آینده بستگی دارد ولی از طرفی هزینه بیشتری دارد و توان پردازشی بیشتری میخواهد ولی monte carlo ساده تر است ولی از مارکوف طبعیت نمیکند.

۷)

سوال: هدف از استفاده از اپسیلون و به کارگیری روش اپسیلون حریصانه چیست؟

اینکار باعث میشود استیت های امتحان نشده را هم امتحان کند یا به عبارتی باعث exploration میشود و اگر از اپسیلون استفاده نکنیم ممکن است جواب بهینه به ما ندهد زیرا تمام استیت ها را امتحان نکرده است تا به بهینه برسد.

(۸

سوال: حال، همین کار را با اپسیلون ۰ دوباره تکرار کنید. آیا مقدار اپسیلون و ضریب یادگیری‌ای وجود دارد که با استفاده از آن‌ها، سیاست بهینه با احتمال خیلی بالا (بیشتر ۹۹ درصد) بعد از ۵۰ بار تکرار یاد گرفته شود؟

خیر زیرا باید چهار خانه متوالی به سمت راست حرکت کند تا به امتیاز بهینه برسد و حتی اگر اپسیلون را برابر یک درنظر بگیریم بازهم احتمال خیلی کمی دارد که به خانه سمت راست برسد و حتی اگر برسد هم از آنجایی که اپسیلون را صفر در نظر گرفتیم حتی پس از آن هم سعی نمیکند به سراغ آن برود.

سوال: به صورت ساده و شهودی توضیح دهید که با کم یا زیاد کردن مقدار epsilon روند یادگیری عامل چگونه تغییر می‌کند.

با زیاد کردن اپسیلون استیت های جدید بیشتری را اکسپلور میکند ولی کمتر به سراغ policy بهینه ای که پیدا کرده میرود. با کم کردن اپسیلون کمتر اکسپلور میکند ولی اگر policy بهینه ای پیدا کند بیشتر به سراغ آن میرود. به طور کلی با افزایش اپیسلون مقدار اکسپلور بیشتر میشود.

سوال: تغییرات و فعالیت‌هایی که در این بخش انجام داده‌اید را توضیح دهید.

هیچ تغییری نیاز نداشت و درست بود.

سوال: درباره Deep Q-leaning تحقیق کنید و بیان کنید در چه مواردی این الگوریتم به جای الگوریتم Q-learning عادی استفاده می‌شود. هر کدام از این الگوریتم‌ها، Approximate Q-learning و Deep Q-learning چه مشکلی را از Q-learning حل می‌کنند.

مانند q learning است با این تفاوت که با گرفتن یک استیت و با استفاده از شبکه های عصبی برای هر اکشن از آن استیت یک q value محاسبه میکند.

در مواردی که استیت ها خیلی زیاد است و پیچیدگی زیادی دارد نمیشود از q learning استفاده کرد و بهتر است از deep q learning استفاده کنیم.

Approximate q learning باعث کاهش سایز table در q learning نیشود.

Deep q learning پیچیدگی q learning برای مواردی که پیچیدگی خیلی زیادی دارند کاهش میدهد.