دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

به نام خدا

تمرین چهارم هوش مصنوعی

استاد درس:

دکتر اکبری

موعد تحویل: ۱۴۰۱/۰۲/۰۶

سوالات تشریحی

سوال اول

توضیح دهید که چگونه الگوریتم‌های minimax و هرس alpha–beta برای بازی‌های دو نفره با مجموع غیرصفر تغییر می‌کنند. اگر هیچ محدودیتی در دو مقادیر دو تابع وجود نداشته باشد، آیا ممکن است گره ای توسط آلفا-بتا هرس شود؟ (در این بازی ها، هر بازیکن یک تابع utility مجزا دارد و هر دو تابع برای هر دو بازیکن شناخته شده است.)

سوال دوم

در این سوال یک درخت max تنها دارای رئوس max است و درخت expectimax شامل راس درخت max و لایه های تناوبی از رئوس max و رئوس chance که در آنها هر احتمال غیر صفر است، می باشد. هدف یافتن مقدار راس درخت با جستجو عمق محدود است. هر مورد را با ذکر دلیل رد کنید یا ممکن بودن آن را با ذکر یک مثال نشان دهید.

الف) با فرض اینکه مقادیر متناظر برگ ها همه متناهی باشند، آیا هرس (مانند alpha–beta) در درخت max ممکن است؟ در درخت expectimax چطور؟

ب) فرض کنید مقادیر همه برگ ها در بازه [0,1] باشند. در این صورت آیا هرس در درخت max یا درخت expectimax ممکن است؟

سوال سوم

الگوریتم minimax بهترین حرکت را برای MAX با این فرض برمی‌گرداند که MIN به طور بهینه بازی می‌کند. اگر MIN به صورت غیربهینه بازی کند چه اتفاقی می‌افتد؟ آیا هنوز هم استفاده از الگوریتم minimax ایده خوبی است؟

سوال چهارم

برای درستی هر مورد استدلال کنید یا با مثال نقض رد کنید.

الف) اگر $\alpha \vDash (\beta \wedge \gamma)$ آنگاه $\alpha \vDash \beta$ و $\alpha \vDash \gamma$

ب) اگر $\alpha \vDash (\beta \vee \gamma)$ آنگاه $\alpha \vDash \beta$ و یا $\alpha \vDash \gamma$ یا هر دو.

سوال پنجم

با استفاده از resolution گزاره $\sim A \wedge \sim B$ را از بند های زیر ثابت کنید.

S۱: $A \Leftrightarrow (B \vee E)$

S۲: $E \Rightarrow D$

S۳: $C \wedge F \Rightarrow \neg B$

S۴: $E \Rightarrow B$

S۵: $B \Rightarrow F$

S۶: $B \Rightarrow C$

پیاده سازی

برنامه ای بنویسید که به کمک الگوریتم minimax (شرح در بخش 5.2 کتاب و فایل "جستجوی خصمانه") با شما x-o بازی کند.

توضیحات: بازی x-o یک بازی مجموع صفر دو نفره است که در آن یک بازیکن نماد x و دیگری نماد o را استفاده میکند. یک صحفه ۳ در۳ داریم و در هر مرحله بازیکنی که نوبت اوست یکی از خانه های خالی را انتخاب میکند و نماد خود را در آن مینویسد.

x	x	o
x	x	o
o	o	x

اتمام بازی در سه حالت ممکن است رخ دهد. با فرض اینکه نماد ما o باشد:

x	o	x
x	o	
	o	

۱. بازیکن حریف یک سطر، ستون یا قطر از صفحه را با نماد های خود پر کند یعنی باخت.

۲. ما یک سطر، ستون یا قطر از صفحه را با نماد های خود پر کنیم یعنی برد.

۳. تمام خانه ها پر شوند و دو حالت بالا رخ نداده باشد یعنی تساوی.

o	x	o
o	x	x
x	o	x

یک راه استفاده از الگوریتم minimax، نسبت دادن مقادیر ۱-، ۱+ و ۰ به حالات پایانی ذکر شده و شروع درخت از یک راس max یا min است.

کد شما هنگام اجرا باید یک عدد بین ۱ تا ۹ به عنوان ورودی بگیرد و نماد o را در خانه متناظر آن اندیس بگذارد. سپس به کمک الگوریتم minimax بهترین خانه برای گذاشتن نماد خود یعنی x را انتخاب کرده، جایگذاری کند و صفحه را چاپ کند، سپس دوباره از ورودی عددی بین ۱ تا ۹ بخواهد. اگر خانه متناظر عددی که به عنوان ورودی دریافت کرد پر بود یا عدد به فرم عدد صحیح بین ۱ و ۹ نبود، پیام خطا چاپ کند و دوباره درخواست ورودی کند. همچنین پس از گرفتن هر ورودی و پر کردن خانه متناظرش و همچنین پس از انتخاب و پر کردن خانه در نوبت خودش، چک کند که آیا حالت پایانی رخ داده یا خیر. اگر یکی از سه حالت را دید کار را متوقف کرده و نتیجه بازی یعنی برد، باخت یا تساوی را به همراه حالت فعلی صفحه چاپ کند.

مراحل پیاده سازی این مساله:

- تعریف یک متغیر به عنوان صفحه بازی. (مثلا یک آرایه ۹ تایی یا ماتریس ۳در۳.)
- تعریف یک تابع مثلا به نام check که یک حالت از صفحه را به عنوان ورودی گرفته و یک زوج مرتب برمیگرداند. مولفه اول یک متغیر Boolean است و اگر صفحه متناظر یک حالات پایانی بود مقدار true و در غیر این صورت false است. مولفه دوم برابر مقدار متناظر حالت پایانی یعنی ۱-، ۱+ یا ۰ است. (اگر متغیر اول false بود، مقدار متغیر دوم عملا استفاده نمی شود، می تواند یک مقدار نامربوط مثل ۲ باشد.)
- تعریف یک حلقه که تا رسیدن به یکی از حالات پایانی برای بازی ادامه می دهد.

در هر تکرار از این حلقه، یک ورودی از کاربر درخواست می کند. پس از دریافت ورودی به فرمت مناسب، در آن خانه از صفحه o گذاشته و بلافاصله شرط توقف یعنی رسیدن به حالت پایانی را چک می کند. سپس تابع minimax را برای حالت فعلی صفحه فراخوانی کرده و بر اساس آن تصمیم بگیرد که در کدام خانه x بگذارد و دوباره شرط توقف را بررسی کند. اگر در حالت پایانی نبود، صفحه کنونی را چاپ کرده و حلقه تکرار شود. در هر جا اگر شرط توقف برقرار بود، نتیجه بازی و صفحه را چاپ کند و از حلقه خارج شود.

- بخش اصلی مساله، تعریف تابع بازگشتی برای پیاده کردن الگوریتم minimax است. این تابع باید صفحه کنونی را به عنوان ورودی بگیرد و با انجام مقدار دهی های ممکن، شاخه های درخت بازی را با فراخوانی بازگشتی خود بررسی کند. در این مساله هر راس از درخت جستجو معادل یک مقدار دهی برای صفحه است و برگ های این درخت، صفحه هایی اند که در یکی از سه حالت پایانی قرار دارند. در هنگام بررسی درخت در هر مرحله تابع check را فراخوانی کرده و اگر مولفه اول خروجی true بود یعنی به برگ رسیدیم پس مولفه دوم خروجی را استفاده می‌کنیم و در غیر این صورت، با تمام مقدار دهی های ممکن خانه های خالی، تابع minimax را بصورت بازگشتی فراخوانی می‌کنیم. برای درک جزییات این تابع، به شبه کد موجود در کتاب یا اسلاید ها مراجعه کنید.

دقت کنید که شبه کد ها طوری نوشته شده اند که مقدار راس درخت (در این مساله یعنی ۰ یا ۱ یا ۱-) را برمیگرداند اما برای انجام این بازی باید بدانید کدام شاخه از درخت یعنی کدام مقدار دهی ما را به آن جواب رسانده تا بتوانیم آن خانه از صفحه را پر کنیم. پس نیاز است تابع را به منظور استفاده برای مساله خود کد کنید.

بخش امتیازی:

تابع بازگشتی خود را طوری تغییر دهید که هرس انجام دهد. (یعنی الگوریتم هرس alpha-beta را کد کنید.)

شما باید صرفا تابع را طوری تغییر دهید که مقادیر alpha و beta را نیز به عنوان ورودی گرفته و در هر مرحله آنها را به روز رسانی و استفاده کند.

در پایان کدتان یک صفحه نیمه تمام دلخواه را تعریف کنید. مثلا :

x		
x	x	o
o		

سپس هر دو تابع را روی آن فراخوانی کنید. با اضافه کردن نوعی شمارنده، تعداد رئوسی که هر یک از دو تابع minimax و alpha-beta بررسی میکنند یا تعداد دفعات فراخوانی آنها را نمایش دهدید تا تفاوت عمکرد آنها دیده شود.

فایل جواب:

- یک فایل ipynb شامل کد و یک فایل متنی برای توضیح کد از شما خواسته شده است.
- در فایل توضیح، عمکرد تابع بازگشتی (هر دو در صورت انجام بخش امتیازی) و همچنین تابع check و حلقه را شرح دهید.
- **استفاده از قطعه کدهای آماده برای پیاده سازی الگوریتم مجاز نیست.**

نکات مهم

- جهت انجام بخش پیاده سازی ،میتوانید از Jupyter استفاده کنید و فایل نهایی را با پسوند. ipynb یا pdf آپلود کنید. همچنین شماره دانشجوی خود را به عنوان نام فایل در نظر بگیرید.
- نکته مهم در گزارش نویسی و سوال تشریحی روشن بودن پاسخ میباشد نه حجم زیاد، اگر فرضی برای حل سوال استفاده میکنید حتما آن را ذکر کنید ،و پاسخ نهایی را به صورت واضح بیان کنید.
- هرگونه شباهت در گزارش و پاسخ تشریحی به منزله تقلب میباشد و کل نمره تمرین صفر میباشد. **(میتوانید از اینترنت به عنوان منبع کمکی هم در سوالات تشریحی و هم در سوالات پیاده سازی استفاده کنید، اما کپی برداری ممنوع می باشد و نمرهی صفر تعلق میگیرد)**
- گزارش کد و پاسخ سوال تشریحی باید در یک فایل pdf باشد.
- **توجه شود: در این تمرین، پاسخ به سوالات تشریحی که تنها نیاز به توضیح و تشریح مسئله دارد ، باید به صورت تایپ شده باشد. اما در مسائل حل کردنی که نیاز به رسم یا استفاده از فرمولهای ریاضی وجود دارد، تایپ ضرورتی ندارد.**
- فایل pdf و کدها را بصورت یکجا در قالب یک فایل zip در سامانه کورسز آپلود کنید (نام فایل = شماره دانشجویی).

معیار ارزیابی شما

1. سوال تشریحی **(۵۰ نمره)**
   - سوال یک (۱۰ نمره)
   - سوال دو (۱۰ نمره) هر بخش ۵ نمره
   - سوال سه (۵ نمره)
   - سوال چهار (۱۰ نمره) هر بخش ۵ نمره
   - سوال پنج (۱۵ نمره)
2. قسمت پیاده سازی **(۵۰ نمره)**

دقت شود، نمره ی گزارش و توضیحات هر بخش در همان بخش در نظر گرفته شده است.

سخن آخر

تدریسیاران سعی کردند با استفاده از توضیحات اضافی در هر بخش، از تمامی ابهامات احتمالی جلوگیری کنند. بنابراین قبل از پرسیدن سوال و ابهامی در رابطه با تمرین، سعی کنید صورت سوال را چندین بار مطالعه کنید و پس از جستجو در اینترنت سوال خود را مطرح کنید.

ارتباط با ما

جهت مطرح کردن سوالات و ابهام هایی که دارید میتوانید از طریق ایمیل های زیر با ما در ارتباط باشید.

شیرین محمدی Shirin.m.100@gmail.com

نوا میرمطلبی n.mirsohi@gmail.com