

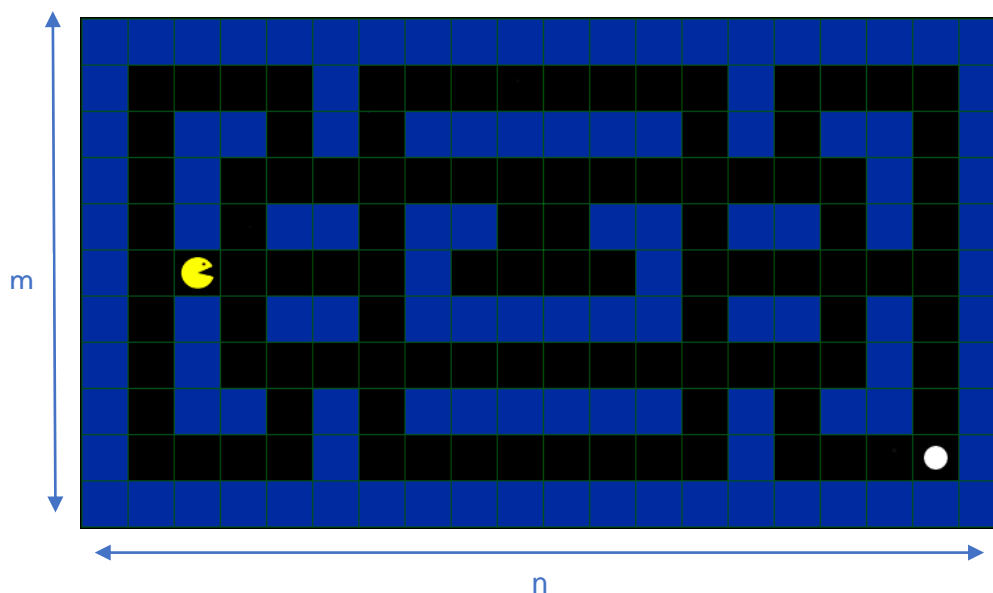


دانشگاه گیلان

پروژه‌ی اول درس هوش مصنوعی و سیستم‌های خبره

پیاده‌سازی هوش مصنوعی برای بازی Pac-Man

تاریخ تحویل : 1402/12/26



بخش اول:

مطابق تصویر فرض کنید بازی Pac-Man دارای صفحه‌ای به تعداد m سطر و n ستون می‌باشد. یعنی تعداد کل خانه‌ها $m*n$ است. هر یک از خانه‌ها می‌تواند یک دیوار یا فضای خالی باشد و در یکی از خانه‌ها عامل (agent) بازی و در یکی دیگر از خانه‌ها غذا وجود دارد. برنامه‌ای بنویسید که عامل بازی بتواند غذا را پیدا کند. در این برنامه باید از توابع «عامل» و «محیط» که در زیر توضیح داده شده است استفاده کنید.

تابع محیط:

$$[\text{Percept}] = \text{Environment (Action)}$$

ورودی این تابع در هر لحظه یک عمل و خروجی آن ادراک عامل که در واقع شامل خانه‌ای که عامل در آن است و وضعیت آن خانه (فضای خالی یا بلاک یا غذا) می‌باشد. این تابع با توجه به موقعیت فعلی عامل و عملی که به عنوان ورودی تابع وارد شده است می‌تواند موقعیت جدید عامل و وضعیت آن خانه را بدست بیاورد. توجه شود از آنجا که این قسمت به عنوان یک تابع نوشته می‌شود و شما لازم دارید که وضعیت کلی خانه‌ها و موقعیت فعلی عامل را در این برنامه داشته باشید، می‌توانید از متغیرهای سراسری (Global) استفاده کنید. **Percept** یک آرایه شامل سه المان است که المان اول و دوم موقعیت عامل و المان سوم وضعیت اینکه آن خانه شامل غذا می‌باشد یا خیر را نشان می‌دهد. (البته می‌توانید بجای در نظر گرفتن آرایه برای **Percept** / آن را به عنوان یک **class** نیز پیاده سازی کنید)

تابع عامل:

$$[\text{Action}] = \text{Agent (Percept)}$$

ورودی تابع عامل، ادراک و خروجی آن عمل می‌باشد. در اینجا نیز اگر می‌خواهید تاریخچه‌ی ادراکات را نگه دارید، می‌توانید از متغیر سراسری مانند آنچه در بند قبلی گفته شد استفاده کنید. توجه شود تابع مذکور تنها ورودی آن ادراکات است و به هیچ اطلاعات دیگری دسترسی ندارد مگر آنکه خودش با ذخیره‌ی تاریخچه‌ی ادراکات و یا پردازش روی آن‌ها، اطلاعاتی را برای خودش تولید کند.

بخش دوم: ایجاد محیط بازی تصادفی (امتیازی)

با گرفتن دو ورودی n , m صفحه‌ای به تعداد m سطر و n ستون پیاده سازی کنید. یعنی تعداد کل خانه‌ها $m*n$ است. دور تا دور محیط را دیوار قرار دهید و محل قرارگیری سایر دیوارها را به صورت تصادفی مشخص کنید. مکان اولیه‌ی عامل و غذا را نیز به صورت تصادفی انتخاب کنید و همانند بخش اول با استفاده از توابع "**عامل**" و "**محیط**" برنامه‌ای بنویسید که عامل بازی بتواند غذا را پیدا کند.

بخش سوم: محیط گرافیکی (امتیازی)

در این بخش می‌توانید شمایی گرافیکی از نحو کارکرد و پیمودن مسیر توسط عامل ایجاد کنید.

بخش چهارم: ورودی ها

1. ابعاد صفحه بازی
2. موقعیت فعلی عامل
3. نقشه بازی

```
6,10
3,2

*****
*-----*
*--***--*
*-a---f-*
*-*-*-*
*****
```

تصویر 1: نمونه ورودی

ورودی ها در قالب یک فایل txt. به شما داده می شود و باید این موارد را از آنجا بخوانید. همانطور که در تصویر آمده/ در خط اول ابعاد صفحه بازی/ در خط دوم موقعیت شروع عامل سپس یک خط خالی وجود دارد و در سطرهای بعد محیط بازی آمده که * به معنی بلاک/ a به معنی agent و f به معنی food می باشد.

بخش پنجم: خروجی ها

1. محل شروع
2. محل غذا
3. تعداد حرکات
4. دنباله تمام حرکات انجام شده تا رسیدن به غذا (به صورت یک سری از محل خانه ها و حرکت انجام شده)

```
Number Of Moves : 38
Moves Steps :
[8,3]--UP-->[7,3]--DOWN-->[8,3]--LEFT-->[8,2]--RIGHT-->[8,3]--UP-->[7,3]--RIGHT-->[7,4]--LEFT-->[7,3]--LEFT-->[7,2]--RIGHT-->
[7,3]--RIGHT-->[7,4]--UP-->[6,4]--DOWN-->[7,4]--RIGHT-->[7,5]--LEFT-->[7,4]--UP-->[6,4]--RIGHT-->[6,5]--LEFT-->[6,4]--RIGHT-->[6,5]--UP-->
[5,5]--DOWN-->[6,5]--RIGHT-->[6,6]--LEFT-->[6,5]--DOWN-->[7,5]--UP-->[6,5]--UP-->[5,5]--RIGHT-->[5,6]--LEFT-->[5,5]--RIGHT-->[5,6]--UP-->
[4,6]--DOWN-->[5,6]--RIGHT-->[5,7]--LEFT-->[5,6]--DOWN-->[6,6]--UP-->[5,6]--UP-->[4,6]--UP-->[3,6]--DOWN-->[4,6]--RIGHT-->[4,6]
```

تصویر 2: نمونه دنباله حرکات انجام شده

توجه داشته باشید/ سیستم شما باید خروجی ها را دقیقاً طبق موارد گفته شده تولید کند. سیستم شما نیاز است خروجی ها را به این ترتیب برگرداند:

۱. یک tuple عنوان مختصات محل شروع. مثال: (2, 4)


۲. یک tuple عنوان مختصات محل غذا. مثال: (3, 6)

۳. یک عدد integer به عنوان تعداد حرکات. مثال: 3

۴. یک لیست متشکل از جفت مختصات و حرکت انتخاب شده. مثال:

[(2, 4), 2, (2, 5), 2, (2, 6), 3, (3, 6)]

توضیح: از خانه (2, 4) شروع میکند با عمل 2 به خانه (2, 5) می رود/ سپس دوباره با انتخاب عمل 2 به خانه (2, 6) می رود و در نهایت با عمل 3 به خانه (3, 6) می رود.

جهت حرکات: 

1 → Up, 2 → Right, 3 → Down, 4 → Left

5. حافظه ساخته شده در طی پیمایش مسیر (امتیازی)

برای مثال فرض کنید زمینی همانند زیر داشته باشیم:

```
*****
*-----*
* _*** _*
* -a---f-*
* _* _* _*
*****
```

و فرض کنید دنباله حرکت ها به این صورت بوده:

(3,2) -right-> (3,3) -up-> (2,3) -down-> (3,3) -right-> (3,4) -right-> (3,5)
-up-> (2,5) -down-> (3,5) -down-> (4,5) -up-> (3,5) -right-> (3,6)

حافظه خروجی به این شکل خواهد بود:

```
*****
*???????*
*??*??*??*
*?-f??*
*????*??*
*****
```

💡 نکته: خانه هایی که پیمایش نشده اند و از آنها اطلاعی نداریم با مقدار " ؟ " خانه هایی که پیمایش شده اند و خلی بوده اند با مقدار " - " ، خانه هایی که پیمایش شده اند و بلاک(دیوار) بوده اند با مقدار " * " و محل غذا با " f " نمایش داده شود.

💡 نکته: همیشه یک مسیر از agent به food وجود دارد.

💡 نکته: الگوریتم باید کوتاه ترین مسیر را پیدا کند.

بخش ششم: گزارش پروژه:

در یک فایل، چند نقشه‌ی متفاوت از بازی یک‌من که به شما داده می شود را کنید. گزارشی از نحوه‌ی اجرای برنامه و تعداد گام‌های طی شده در رسیدن عامل به غذا و همچنین مسیر حرکت عامل برای رسیدن به غذا در طی اجراهای مختلف در قالب یک فایل PDF تهیه کنید و همراه کد برنامه در قالب یک فایل فشرده‌ی ZIP یا RAR در سامیاد تحویل دهید.

نام فایل تحویلی: HW1-LastName-Studentumber.zip

بخش هفتم: نحوه ارزیابی

برای ارزیابی، کد شما در چندین فایل خروجی تست خواهد شد و جواب ها بررسی می شوند، همچنین لازم است تا دانشجویان در خصوص الگوریتم کلی و همچنین ابهامات به مصحح توضیحات لازم را اراعه کنند.

💡 حتما به نحوه خروجی ها دقت داشته باشید زیرا ارزیابی به صورت نرم فزاری انجام خواهد شد.

موفق باشید