

تمرین سری ۴: یادگیری تقویتی

لطفاً به نکات زیر توجه فرمایید:

- مهلت ارسال این تمرین تا ۲۳ خرداد است.
- در صورتی که به اطلاعات بیشتری نیاز دارید می‌توانید به صفحه‌ی تمرین در وب‌سایت درس مراجعه کنید.
- این تمرین شامل سوال‌های برنامه‌نویسی می‌باشد، بنابراین توجه کنید که حتماً موارد خواسته‌شده در سوال را رعایت کنید. در صورتی که به هر دلیلی سامانه‌ی داوری نتواند آن را اجرا کند مسئولیت آن تنها به عهده‌ی شماست.
- ما همواره هم‌فکری و هم‌کاری را برای حل تمرین‌ها به دانشجویان توصیه می‌کنیم. اما هر فرد باید تمامی سوالات را به تنهایی تمام کند و پاسخ ارسالی حتماً باید توسط خود دانش‌جو نوشته‌شده باشد. لطفاً اگر با کسی هم‌فکری کردید نام او را ذکر کنید. در صورتی که سامانه‌ی تطبیق، تقلبی را تشخیص دهد متأسفانه هیچ مسئولیتی بر عهده‌ی گروه تمرین نخواهد بود.
- لطفاً برای ارسال پاسخ‌های خود از راهنمای موجود در صفحه‌ی تمرین استفاده کنید.
- هر سوالی درباره‌ی این تمرین را می‌توانید در گروه درس مطرح کنید و یا از دستیاران حل تمرین بپرسید.

موفق باشید.

برای این تمرین از محیط کمکی PLE و gym استفاده می‌کنیم .
ابتدا باید به کمک لینک زیر ان‌ها را نصب کنید و سپس سوال زیر را پیاده سازی کنید:

<https://pygame-learning-environment.readthedocs.io/en/latest/user/home.html#installation>

پیشنهاد میشود برای دوستانی که از ویندوز استفاده می‌کنند از anaconda برای نصب راحت استفاده کنند.

روش استفاده از ماژول PLE به شکل زیر می باشد:

```
from ple.games.flappybird import FlappyBird
from ple import PLE

game = FlappyBird()
p = PLE(game, fps=30, display_screen=True)
agent = myAgentHere(allowed_actions=p.getActionSet())

p.init()
reward = 0.0

for i in range(nb_frames):
    if p.game_over():
        p.reset_game()

    observation = p.getScreenRGB()
    action = agent.pickAction(reward, observation)
    reward = p.act(action)
```

Snake

هدف این سوال پیاده سازی عامل یادگیری تقویتی برای بازی مار است.

ابتدا فایل snake.py را از سایت درس دریافت کنید که یک عامل رندوم برای بازی می‌باشد و در داخل آن کد عامل RL خود را پیاده‌سازی کنید. برای این کار ابتدا احتیاج دارید یک تابع پیاده‌سازی کنید که state و action را به عنوان ورودی دریافت کند و پاداش این عمل را به عنوان خروجی برگرداند سپس با استفاده از الگوریتم Q-Learning یک عامل برای این بازی پیاده‌سازی کنید.

این بازی همانند نسخه کلاسیک آن می‌باشد که در آن کنترل یک مار در دست عامل شما می‌باشد و غذاها به شکل رندوم بر روی صفحه قرار می‌گیرند و شما باید با کنترل عامل خود غذاهای روی صفحه را بخورید. همچنین تضمین می‌شود که غذاها بر روی بدن مار قرار نمی‌گیرند و با خوردن هر غذا طول مار افزایش پیدا می‌کند همچنین در صورت برخورد مار به خود مار یا دیوار بازی پایان می‌یابد.

برای ارسال تمرین خود فایل snake.py را در سایت کوئرا قسمت تمرین ۴ بارگزاری کنید.

حرکات

بالا – پایین – چپ – راست (توجه شود مار نمیتواند به روی خود برگردد یعنی اگر به راست می رود حرکت بعدی نمی‌تواند چپ باشد)

پایان بازی

سر مار با دیوار یا بدنش تماس پیدا کند.

امتیاز

اگر غذا را بخورد مثبت یک اگر بمیرد منفی یک امتیاز دریافت می کند.

توابع مورد نیاز

```
Class ple.games.snake.Snake(width=64, height=64, init_length=3)
```

```
width : int
```

عرض صفحه را مشخص می کند.

```
height : int
```

طول صفحه را مشخص می کند.

```
init_length : int
```

طول ابتدایی مار را مشخص می کند.

```
getState()
```

یک دیکشنری شامل X و Y سر مار و X و Y غذای موجود و فاصله ی سر مار تا غذا را برمی گرداند.

```
getActionSet()
```

اکشن های ممکن را برمی گرداند.

```
act(action)
```

یک اکشن بدست آمده از تابع (getActionSet) را اجرا می کند.

رای اطلاعات بیشتر به صفحه زیر مراجعه کنید.

<https://pygame-learning-environment.readthedocs.io/en/latest/user/games/snake.html>