

به نام خدا



درس هوش مصنوعی و سیستم‌های خبره

---

## پروژه پایانی درس

---

طراحان:

محمد جواد حقیقی

حامد فیض آبادی

محمد امین چینی فروشان

مدرس درس:

دکتر محمد رضا محمدی

مهلت ارسال: ۳ بهمن ماه ۱۴۰۱

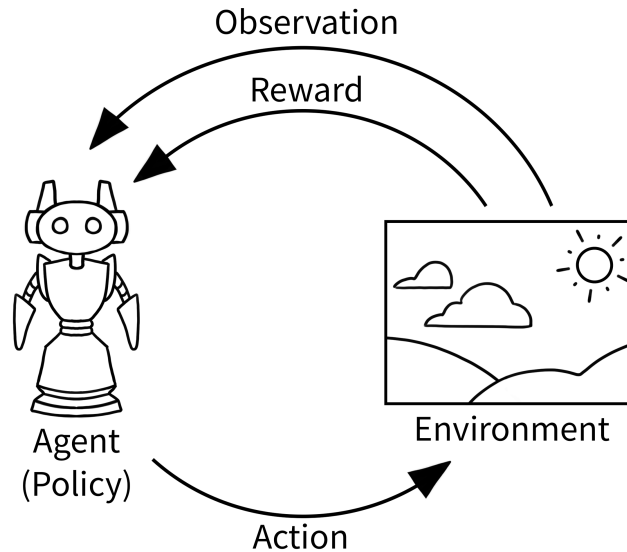
تاریخ ارائه پروژه‌ها: ۴ و ۵ بهمن ماه ۱۴۰۱

## فهرست مطالب

3	1 آماده سازی محیط پروژه
3	1.1 انتخاب سیستم عامل
3	2.1 نصب کتابخانه
4	2 انتخاب پروژه به جهت پیاده سازی
4	1.2 به چه نکاتی برای انتخاب پروژه باید توجه کرد؟
5	2.2 چگونه می توانیم از این پروژه نمره مناسبی بگیریم؟
6	3 بررسی اجرای یک نمونه از بازی ها

## مقدمه

در این سند، روند انجام پروژه پایانی درس هوش مصنوعی بررسی خواهد شد. پروژه این درس با موضوع Reinforcement Learning بر بستر نسخه جدید کتابخانه Gym به نام **Gymnasium** انجام خواهد شد. کتابخانه Gym توسط **OpenAI** توسعه پیدا کرد اما در سال 2021، **OpenAI** توسعه این کتابخانه در ریپازیتوری **Gym** را متوقف کرد و ادامه فرآیند توسعه این کتابخانه با نام **Gymnasium** به **Farama** واگذار شد. در این کتابخانه بستر انجام آزمایش‌هایی برای آموزش یک عامل هوشمند در اختیار شما قرار داده شده است. لینک [صفحه مستندات](#) و [صفحه GitHub](#) برای انجام این پروژه کمک فراوانی به شما می‌کند. در ادامه به بررسی روند نصب این کتابخانه و بررسی یک مسئله نمونه می‌پردازیم و در نهایت روند انتخاب و انجام پروژه شرح داده خواهد شد.



شکل 1: نمای کلی از کتابخانه **Gymnasium**  
ورزشگاه  
زور خانه

## 1 آماده سازی محیط پروژه

### 1.1 انتخاب سیستم عامل

طبق توضیحات صفحه گیت هاب این کتابخانه، پروژه‌های این کتابخانه تنها بر روی سیستم عامل‌های لینوکس و مک تست شده و پشتیبانی می‌شود. توصیه اکید ما این است که سیستم عامل لینوکس را به‌طور مستقیم بر روی رایانه خود نصب کنید<sup>1</sup>، اما اگر در نظر دارید بر روی ویندوز این پروژه را اجرا کنید این دو لینک می‌تواند به شما کمک کند:

- [How to Install OpenAI Gym in a Windows Environment](#)
- [Install OpenAI Gym Atari on Windows](#)

همچنین اگر در نظر دارید سیستم عامل linux را با ماشین مجازی اجرا کنید، مطمئن شوید که Driver های مناسب کارت گرافیک شما بر روی ماشین مجازی مورد نظر نصب شود تا پروژه به درستی اجرا شود.<sup>2</sup>

### 2.1 نصب کتابخانه

Gymnasium به عنوان یک کتابخانه در زبان برنامه نویسی پایتون در اختیار شما قرار گرفته است که مطابق بخش نصب صفحه گیت هاب از ورژن‌های 3.7, 3.8, 3.9, 3.10, 3.11 این زبان برنامه نویسی پشتیبانی می‌کند. این کتابخانه شامل چندین محیط بازی RL در زیربخش‌های Box2D, Classic Control, Toy Text, MuJoCo, Atari است. برای نصب تمامی پروژه‌های این کتابخانه با هم، می‌توانید با استفاده از pip مطابق زیر پروژه‌ها را نصب کنید:

```
pip install "gymnasium[all]" # to install all projects
```

اما برای نصب هر زیر بخش بطور جداگانه می‌توانید به صفحه آن مراجعه کنید.

<sup>1</sup> یعنی واقعا تا الان نصب نکردین؟؟؟ 🤖

<sup>2</sup> این کار احتمالا بیشتر از نصب سیستم عامل از شما وقت خواهد گرفت.

## 2 انتخاب پروژه به جهت پیاده سازی

همانطور که در بخش قبل ذکر شد، محیط‌های شبیه ساز RL در این کتابخانه به زیر بخش‌های نام برده تقسیم بندی شده‌اند که هر کدام از این زیر بخش‌ها ویژگی‌های مشترکی دارند که در جدول 1 مشاهده می‌کنید:

جدول 1: مقایسه دسته‌بندی پروژه‌ها

Hardness Level	Observation Space	نام دسته
Extremely Simple	vector	Toy Text
Intermediate	vector	Classic Control
😊	matrix / image	Box2D
😬	matrix / vector	MuJoCo
😄	matrix / image	Atari

برای انتخاب پروژه این درس بعد از مشاهده و بررسی تمام پروژه‌ها در صفحه مستندات، در شیت پروژه اعضای گروه و 3 اولویت اول خود برای انتخاب موضوع پروژه را وارد کنید. توجه کنید که اولویت سوم شما به هیچ عنوان نباید در مقایسه با سایر پروژه‌های انتخاب شده توسط همه تکراری باشد. در انتهای هر روز شیت پروژه بررسی خواهد شد و پروژه‌ی شما از میان اولویت‌های شما مشخص می‌شود. بنابراین سعی کنید با بررسی هر چه زودتر پروژه‌های دلخواه خود را انتخاب کنید.

### 1.2 به چه نکاتی برای انتخاب پروژه باید توجه کرد؟

1. **Observation Space**، پارامتر مهمی که برای انتخاب پروژه باید به آن توجه کرد. این پارامتر در واقع بیان‌کننده‌ی روش درک Agent از محیط اطراف خود است که می‌تواند به صورت لیست یا ماتریسی از پارامترهای گسسته و یا پیوسته باشد. در برخی موارد Observation Space می‌تواند تصویر محیط بازی باشد که شما برای درک محیط بازی توسط Agent خودتان باید ویژگی‌های مورد نظر خودتان را از تصویر استخراج کنید. این ویژگی‌ها می‌تواند شامل محل اشیاء دیگر و یا شناسایی مسیر حرکت و ... باشد، این کار با توجه به محیط پروژه می‌تواند به شدت سخت یا آسان باشد 😊 و شامل نمره امتیازی می‌شود 😊.

2. قبل از وارد کردن نام پروژه در لیست حتما مطمئن شوید که این پروژه بر روی سیستم عامل شما قابل اجرا است. هر کدام از محیط‌های مختلف شبیه سازی شده (Atari, ...) نیازمندی‌های متفاوتی برای اجرا دارد. که قبل از انتخاب حتما آن‌ها را بررسی کنید. برای صرفه جویی در زمان حداکثر دو دسته را انتخاب کرده و سعی کنید یک پروژه از آن را اجرا کنید. اگر بتوانید پروژه‌ای از یک دسته را اجرا کنید تمام پروژه‌های آن دسته بر روی سیستم عامل شما قابل اجرا خواهد بود.

3. از میان این دسته‌ها، دسته‌ی Toy Text برای گروه‌های تک نفره و دسته‌ی Classic Control برای گروه‌های دونفره می‌تواند نمره کامل پروژه را به همراه داشته باشد.

4. معروف ترین دسته از میان این دسته‌ها دسته‌ی Classic Control است و کدهای زیادی از انواع الگوریتم‌ها برای بازی‌های این دسته در اینترنت موجود است، بنابراین در ارائه این پروژه دقت بیشتری خواهد شد. طبیعتاً به استفاده از کدهای آماده بدون تغییر و فهم آن نمره‌ای تعلق نخواهد گرفت.

5. به آن دسته از دانشجویانی که علاقه‌مند به کسب نمره امتیازی مناسب از این پروژه هستند پیشنهاد می‌شود که پروژه‌های خود را از دسته Atari یا Box2D انتخاب کنند. با انتخاب پروژه از این دو دسته به دلیل پیچیدگی فضا شاید نتوانید به هوشمندی مناسبی برای Agent خود دست‌یابید اما نمره‌دهی برای این پروژه‌ها تنها به این پارامتر وابسته نیست.

6. پس از انتخاب پروژه‌ها، بعضی پروژه‌ها میان تیم‌ها تکراری خواهد شد و درنهایت با مقایسه عملکرد Agent نهایی این پروژه‌ها به بهترین Agent نمره امتیازی اختصاص داده خواهد شد.

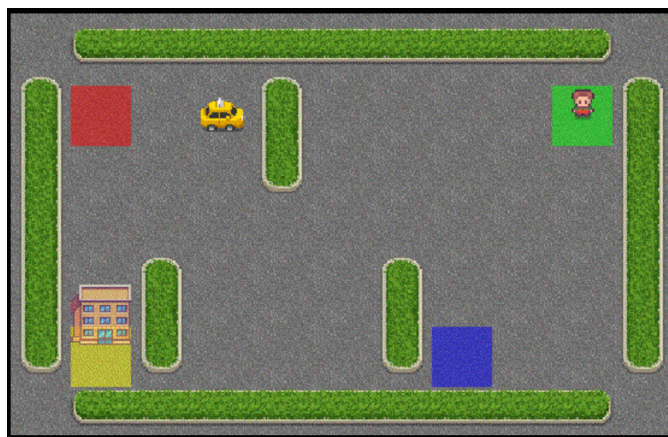
## 2.2 چگونه می‌توانیم از این پروژه نمره مناسبی بگیریم؟

به یاد داشته باشید که تک تک اقدامات شما در راه رسیدن به الگوریتم و Agent نهایی در افزایش نمره شما موثر خواهد بود. الگوریتم نهایی کمتر از ۴۰٪ نمره شما را شامل می‌شود. بنابراین تمام مراحل رسیدن به کد نهایی پروژه و راه حل‌های خلاقانه برای بهبود نتیجه به همراه گزارش نتیجه و مقایسه آن با حالت قبل را در ارائه خود داشته باشید. برای مقایسه رفتار Agent با حالت قبل تهیه یک ویدیو از رفتار آن در محیط بازی می‌تواند مفید باشد<sup>3</sup>. نیازی به نوشتن گزارش برای این پروژه نیست، گزارش شما در واقع سری اسلایدهایی است که برای ارائه خود آماده خواهید کرد. طبیعتاً کیفیت اسلایدهای ساخته شده و جزئیات آزمایش‌های گزارش شده در اسلایدها و کیفیت کدهای پیاده سازی شده در نمره شما موثر خواهد بود بنابراین ساخت اسلایدها را به هیچ عنوان به انتهای کار موکول نکنید و پس از انجام هر آزمایش و پیاده سازی، اسلایدهای خود را تکمیل کنید.

<sup>3</sup> شما بخوانید این بهترین راه است.

### 3 بررسی اجرای یک نمونه از بازی‌ها

در این بخش به اجرای پروژه Taxi از دسته Text Toy می‌پردازیم. این پروژه بر روی توزیع Ubuntu سیستم عامل Linux آزمایش شده است. در شکل 2 نمونه‌ای از تصاویر محیط بازی را مشاهده می‌کنید.



شکل 2: نمونه‌ای از محیط بازی Taxi

در محیط این بازی چهار مکان تعیین شده برای سوار کردن و پیاده کردن (قرمز، سبز، زرد و آبی) مسافر در دنیای شبکه  $[5,5]$  وجود دارد. مسافر در یکی از مکان‌های تعیین شده قرار دارد و تاکسی از یک میدان تصادفی حرکت می‌کند. هدف انتقال تاکسی به محل مسافر، سوار کردن مسافر، حرکت به مقصد مورد نظر مسافر و پیاده کردن مسافر است. هنگامی که مسافر پیاده می‌شود، بازی به پایان می‌رسد و Agent برای پیاده کردن موفقیت آمیز مسافر در مکان صحیح، جوایز مثبت دریافت می‌کند. پاداش‌های منفی نیز برای تلاش‌های نادرست برای سوار کردن یا پیاده کردن مسافر به Agent داده می‌شود. برای مطالعه Action Space و Observation Space این بازی می‌توانید به صفحه بازی مراجعه کنید.

در بخش زیر کد پیاده سازی شده برای کنترل تاکسی آورده شده است:

```
1 import gymnasium as gym
2 env = gym.make("Taxi-v3", render_mode="human")
3 observation, info = env.reset(seed=42)
4 for _ in range(1000):
5     action = env.action_space.sample() # this is where you would
        insert your policy
6     observation, reward, terminated, truncated, info = env.step(
        action)
7     print(f"reward: {reward} for action {action}")
8     if terminated or truncated:
9         observation, info = env.reset()
10 env.close()
```

در این کد تاکسی بصورت تصادفی حرکت می کند و هوشمندی خاصی ندارد. در واقع شما باید با استفاده از منطقی که الگوریتم RL شما یاد می گیرد و با توجه به متغیرهای observation و reward به جای یک action تصادفی، action مناسب را تولید کنید.



### نکات پایانی:

1. پروژه‌های پیشنهادی خود را انتخاب و حداکثر تا تاریخ ۲۴ دی ماه ۱۴۰۱ در **شیت پروژه** وارد کنید.
2. پروژه به صورت انفرادی یا گروه‌های دو نفره انجام خواهد شد اما نمره دهی آن بصورت فردی است، بنابراین تسلط تمامی اعضای گروه بر تمامی بخش‌های پروژه الزامی است.
3. نمره شما بر اساس گزارش راه طی شده برای حل مسئله و پاسخ صحیح خواهد بود لذا از هرگونه اطناب در گزارش پرهیز و به موارد خواسته شده به صورت کامل پاسخ دهید.
4. برای پروژه از شما ارائه شفاهی گرفته خواهد شد بنابراین تسلط لازم را بر کدی که پیاده می‌کنید داشته باشید.
5. برای پیاده سازی استراتژی‌های مختلف در جهت بهبود رفتار Agent خود می‌توانید با سایر گروه‌ها مشورت کنید و در نهایت استراتژی خود را پیاده سازی کنید.
6. در صورتی مشاهده شباهت غیرعادی بین پیاده سازی‌ها نمره طرفین طبق قوانین درس محاسبه خواهد شد.
7. برای تحویل پروژه یک فایل zip شامل کدها و اسلایدها، با نام [Project\_ID\_NAME1\_ID\_NAME2] در سامانه LMS بارگذاری کنید.

1. Q-learning -> AI\_23:page 23, AI\_24:page1,  
سوال 2 تمرین 12

2. Approximate Q-Learning -> AI\_24:page 12

### Approximate Q-Learning

- Generalize the experiences to new, similar situations
- Using a feature representation, we can write a q function (or value function) for any state using a few weights

$$Q(s, a) = w_1 f_1(s, a) + w_2 f_2(s, a) + \dots + w_n f_n(s, a)$$

- Q-learning with linear Q-functions:

$$\text{transition} = (s, a, r, s')$$

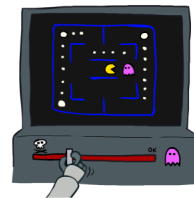
$$\text{difference} = [r + \gamma \max_{a'} Q(s', a')] - Q(s, a)$$

$$Q(s, a) \leftarrow Q(s, a) + \alpha [\text{difference}]$$

$$w_i \leftarrow w_i + \alpha [\text{difference}] f_i(s, a)$$

Exact Q's

Approximate Q's



3. Deep Q-Learning -> AI\_24:page 19