مینی‌پروژهٔ شمارهٔ چهار

در انجام این مینی‌پروژه حتماً به نکات زیر توجه کنید:

- موعد تحویل این مینی‌پروژه، ساعت ۱۸:۰۰ روز جمعه ۱۵ تیرماه ۱۴۰۳ است.
- برای این مینی‌پروژه ملزم به ارائهٔ گزارش متنی شامل توضیحات کامل هر قسمت هستید. هم گزارش و هم کدهای خود را در گیت‌هاب و سامانهٔ دانشگاه بارگذاری کنید.
- برای گزارش لازم است که پاسخ هر سوال و زیربخش‌هایش به‌ترتیب و به‌صورت مشخص نوشته شده باشند. بخش زیادی از نمره به توضیحات دقیق و تحلیل‌های کافی شما روی نتایج بستگی خواهد داشت.
- لازم است که در صفحهٔ اول گزارش خود لینک پوشهٔ گیت‌هاب و گوگل‌کولب مربوط به مینی‌پروژهٔ خود را در حالتی که دسترسی Public دارد به اشتراک گذاشته باشید. دفترچه‌کد گوگل‌کولب باید به‌صورت منظم و با بخش‌بندی مشخص تنظیم شده باشد، و خروجی سلول‌های اجراشده قابل مشاهده باشد. در گیت‌هاب هم برای هر مینی‌پروژه یک پوشهٔ مجزا ایجاد کنید.
- هرجا از دفترچه‌کد گوگل‌کولب شما نیاز به فراخوانی فایلی خارج از محیط داشت، مطابق آموزش‌های ارائه‌شده ملزم هستید از دستور gdown استفاده کنید و مسیرهای فایل‌ها را طوری تنظیم کنید که صرفاً با اجرای سلول‌های کد، امکان فراخوانی و خواندن فایل‌ها توسط هر کاربری وجود داشته باشد.
- در تمامی مراحل تعریف داده و مدل و هرجای دیگری که مطابق آموزش‌های ویدیویی و به لحاظ منطقی نیاز است، Random State را برابر با دو رقم آخر شمارهٔ دانشجویی خود در نظر بگیرید.
- استفاده از ابزارهای هوشمند (مانند ChatGPT) در کمک‌گرفتن برای بهبود کدها مجاز است؛ اما لازم است تمام جزئیات مواردی که در خروجی‌های مختلف گزارش خود عنوان می‌کنید را به خوبی خوانده، درک و تحلیل کرده باشید. استفاده از این ابزارهای هوشمند در نوشتن گزارش و تحلیل‌ها ممنوع است.
- در جاهایی که با توجه به دو رقم آخر شمارهٔ دانشجویی خود محدود به انتخاب عدد، متغیر و یا داده‌ای خاص شده‌اید، برای تست‌های اضافه‌تر و نمایش بهبود در نتایج خود، مجاز هستید از مقادیر دیگر هم استفاده کنید. ۱۵ تا ۲۰ درصد از نمرهٔ هر سوال به بهترین نتایج کسب‌شده اختصاص خواهد یافت.
- رعایت نکات بالا به حرفه‌ای‌ترشدن شما کمک خواهد کرد و اهمیتی معادل مطالب درسی فراگرفته‌شده دارد؛ بنابراین، درصورت عدم رعایت هریک از این نکات، گزارش شما تصحیح نخواهد شد.

به انتخاب خود به یکی از پرسش‌های یک یا دو پاسخ دهید.

۱ پرسش یک: حل دنیای Wumpus

Wumpus World یک مسئله کلاسیک در هوش مصنوعی و یادگیری تقویتی است که شامل یک محیط مبتنی بر شبکه است که در آن یک عامل باید برای یافتن طلا حرکت کند و در عین حال از خطراتی مانند چاله ها و Wumpus اجتناب کند.

- اهداف پیمایش در شبکه Grid: عامل باید یاد بگیرد که به طور موثر در شبکه حرکت کند.
- اجتناب از خطرات: عامل باید یاد بگیرد که از چاله ها و Wumpus اجتناب کند.

- جمع آوری طلا: عامل باید طلا را پیدا کرده و جمع آوری کند.
- کشتن Wumpus: عامل می تواند برای کشتن Wumpus تیری شلیک کند و آن را به عنوان تهدید از بین ببرد.
- راه اندازی محیط شبکه: یک شبکه ۴*۴ که در آن هر سلول می تواند خالی باشد، حاوی یک گودال، Wumpus یا طلا باشد.
- فضای اکشن ها: حرکت به بالا، پایین، چپ، راست.
  یک فلش را در هر یک از چهار جهت (بالا، پایین، چپ، راست) شلیک کنید (امتیازی).
- تصورات: Wumpus در شبکه با هر تغییر اکشن به اندازه یک خانه در راستای چپ، راست، بالا یا پایین حرکت می‌کند (امتیازی).
- فضای Reward:
  - +۱۰۰ برای گرفتن طلا
  - -۱۰۰۰ برای افتادن در گودال یا خورده شدن توسط Wumpus
  - +۵۰ برای کشتن Wumpus (امتیازی)
  - -۱ برای هر حرکت
- تعریف محیط: یک شبکه ۴×۴ با موقعیت های دلخواه برای چاله ها، Wumpus و طلا ایجاد کنید. حالت اولیه و حالت های ممکن را بعد از هر عمل تعریف کنید.
- تنظیم پارامترها:
  - نرخ یادگیری : ۰.۱
  - ضریب تخفیف: ۰.۹
  - نرخ اکتشاف: از ۱.۰ شروع می شود و در طول زمان کوچک میشود.

با توجه به موارد کلی گفته شده راجع به مسئله، موارد زیر را پاسخ دهید.

آ. برای این مسئله یک بار با روش Q-learning و یک بار با روش Deep Q-learning عاملی را طراحی کرده و آموزش دهید.

ب. عملکرد Policy:

- پاداش تجمعی را در اپیزودها برای هر دو عامل Q-learning و DQN ترسیم کنید. چگونه عملکرد عامل در طول زمان بهبود می یابد؟
- میانگین پاداش در هر اپیزود را برای هر دو عامل پس از ۱۰۰۰ اپیزود مقایسه کنید. کدام الگوریتم عملکرد بهتری داشت؟

ج. بحث کنید که چگونه نرخ اکتشاف اپسیلون بر فرآیند یادگیری تأثیر می گذارد. وقتی اپسیلون بالا بود در مقابل وقتی کم بود چه چیزی را مشاهده کردید؟

د. کارایی یادگیری:

- چند اپیزود طول کشید تا عامل Q-learning به طور مداوم طلا را بدون افتادن در گودال یا خورده شدن توسط Wumpus پیدا کند؟
- کارایی یادگیری Q-learning و DQN را مقایسه کنید. کدام یک Policy بهینه را سریعتر یاد گرفت؟

ه. معماری شبکه عصبی مورد استفاده برای عامل DQN را شرح دهید. چرا این معماری را انتخاب کردید؟

۲ پرسش دو: Deep Q-Learning برای محیط Lunar Lander

در این سوال از شما می‌خواهیم با استفاده از روش Deep Q-Learning برای محیط Lunar Lander یک عامل طراحی کرده و آموزش دهید که بتواند مسأله را حل کند. برای حل این تمرین می‌توانید از این دفترچه‌کد استفاده کرده و در صورت نیاز آن را تکمیل کرده و تغییر دهید. اگر از دفترچه‌کد ارائه‌شده و کدهای آماده استفاده می‌کنید، لازم است که تمام اجزای کد را با دقت توضیح دهید. دقت داشته باشید که هدف، توضیح روند عملکردی کد است، نه تحلیل خشک و لغتی برآمده از ChatGPT!

آ. در مورد محیط Lunar Lander مطالعه کرده و به‌صورت خلاصه ویژگی‌های آن را شرح دهید. ویژگی‌های مدنظر عبارتند از مشخصات فضای حالت، مشخصات فضای عمل و سیستم پاداش.

ب. عملکرد عامل را با رسم پاداش تجمعی در هر episode و برای batch sizeهای ۳۲، ۶۴ و ۱۲۸ بررسی کنید. تنها برای بهترین حالت به ازای episodeهای ۵۰، ۱۰۰، ۱۵۰، ۲۰۰ و ۲۵۰ فیلمی از عملکرد عامل تهیه کنید.

در صورتی که عملکرد عامل به‌ازای هر سه مقدار batch size مشابه یکدیگر شد، یکی از آن‌ها را به دلخواه به‌عنوان بهترین حالت انتخاب کنید. در رابطه با انتخاب بهترین حالت علاوه بر معیار سرعت همگرایی به پاداش بهینهٔ معیار regret را نیز به‌صورت شهودی بررسی کنید.

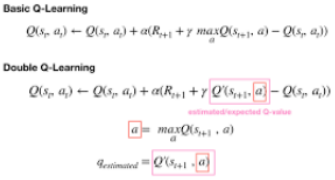
ج. عملکرد مدل DQN و DDQN را با رسم پاداش تجمعی در هر episode و به‌ازای batch size برابر مقایسه کنید. برای هر دو مدل به‌ازای episodeهای ۱۰۰ و ۲۵۰، قیلمی از عملکرد مدل تهیه کنید.

هر بار آموزش عامل با استفاده از GPUهای رایگان گوگل‌کولب حدوداً بین ۱۰ تا ۱۵ دقیقه زمان لازم خواهد داشت. برای تهیهٔ خروجی می‌توانید checkpointهای مدل را دانلود کرده و روی سیستم خود فیلم‌ها را تهیه کنید.

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$
  end for
end for

شکل ۱: شبه‌کد الگوریتم DQN



شکل ۲: تفاوت مدل DDQN با DQN

منابع

[1] https://github.com/MJAHMADEE/MachineLearning2024W