Neural Networks and Deep Learning Assignment 6

Instructors: **Dr. Bahrak** TA:محمدرضا محمدهاشمی

Deadline: 1404/03/27

مقدمه:

در این تکلیف کامپیوتری با هدف ارتقاء مهارت‌های تحقیق و پیاده‌سازی در حوزه یادگیری تقویتی، دو روش پیشرفته از مقالات معتبر را اجرا و ارزیابی خواهید کرد. در تسک ۱، روش DQN را در مسئله «perishable inventory management» تقویت خواهید کرد و با استفاده از تکنیک Reward Shaping به بهبود همگرایی و کاهش هزینه‌های ناشی از نگهداری و فاسدشدن کالا می‌پردازید. در تسک ۲، الگوریتم Dueling Munchausen DQN را برای مسیریابی ربات پیاده‌سازی کرده و عملکرد آن را در مقایسه با نسخه‌های پایه DQN و M-DQN در محیط‌های شبیه‌سازی Gym مورد سنجش قرار می‌دهید.

تمامی پیاده‌سازی‌ها باید با استفاده از Stable-Baselines3 برای الگوریتم‌ها و OpenAI Gym برای ساخت محیط‌ها انجام شود. در پایان، نمودارهای همگرایی، جداول مقایسه‌ای و تحلیل‌های کمی و کیفی نتایج باید در گزارش تحلیلی ارائه شوند تا درک عمیق‌تری از رفتار و مزایای هر روش به دست آورید.

پرسش 1: تقویت DQN با Reward Shaping در مدیریت موجودی فاسدشدنی

1-1. معرفی مقاله

در این مقاله مسئله‌ی تصمیم‌گیری در مدیریت موجودی کالای فاسدشدنی مطرح شده است. سازوکار اصلی DQN برای یادگیری سیاست بهینه در این محیط کاربرد دارد، اما به دلیل فضای حالت بزرگ و پاداش‌های پراکنده، همگرایی کند و ناپایدار است. نویسندگان با معرفی Potential-based Reward Shaping هدف دارند با افزودن سیگنال پاداش کمکی (shaping) به پاداش اصلی، سرعت یادگیری و کیفیت سیاست نهایی را بهبود بخشند، بدون آنکه بهینه‌بودن سیاست را تحت تأثیر قرار دهند.

1-2. مطالعه و تحلیل مقاله (دو نمره)

در این قسمت باید نکات کلیدی مقاله را استخراج کرده و در گزارش خود قید کنید: ساختار فضای حالت (بردار pipeline و میزان موجودی)، فضای عمل (مقادیر سفارش)، و معادلات پاداش اصلی. همچنین مروری بر مفاهیم ε-greedy، target network، DQN Replay Buffer و اصول Potential-based Reward Shaping انجام دهید تا بتوانید پیاده‌سازی را دقیق مطابق مقاله پیش ببرید و هر دو روش استفاده شده در مقاله (base-stock heuristic, BSP-low-EW).

1-3. طراحی محیط شبیه‌سازی با Gym (پنج نمره)

یک کلاس جدید از `gym.Env` تعریف کنید که:

- پارامترهای m (عمر کالا) و L (تاخیر تحویل) قابل تنظیم باشد.
- متدهای `reset()` و `step()` شامل منطق FIFO/LIFO در تحویل و فاسدشدن کالا باشند.
- خروجی‌ها (`observation_space` و `action_space`) با مقاله مطابقت داشته باشند.

در پایان، با چند نمونه run کوچک صحت عملکرد محیط را بررسی کنید.

1-4. فراهم کردن شرایط اموزش به کمک تابع پاداش با Shaping (هشت نمره)

ابتدا پاداش پایه را بر اساس هزینه‌های نگهداری، کمبود و فاسدشدن کالا بنا کنید. سپس یک تابع پتانسیل $\Phi(s)$ انتخاب کرده و سیگنال shaping را مطابق

$$\Phi(s) - \gamma\Phi(s') = F(s, a, s')$$

و محیط اموزش بر پایه ی هر دو مدل reward shaping را به کمک custom gym environment ها فراهم کنید.

1-5. پیاده‌سازی مدل DQN در Stable-Baselines3 (پنج نمره)

از کلاس `stable_baselines3.DQN` استفاده کنید و معماری شبکه (تعداد لایه‌ها و نورون‌ها) را مطابق مقاله تنظیم نمایید. پارامترهای کلیدی مانند اندازه‌ی Replay Buffer، نرخ یادگیری، نرخ ε و فاصله به‌روزرسانی target network را مطابق تنظیمات مقاله مقدار دهی کنید. اسکریپتی بنویسید که آموزش را اجرا کرده و مدل نهایی را ذخیره نماید.

1-6. آموزش مدل ها (ده نمره)

سه حالت زیر را با پارامتر های مختلف اجرا و مقایسه کنید(هرکدام را حداقل با 3 seed تصادفی مختلف اجرا کنید):

1. DQN بدون reward shaping
2. DQN با base stock reward shaping
3. DQN با BSP-low-EW reward shaping

خروجی مورد انتظار برای هر حالت و هر seed:

- آرایه‌ای از میانگین هزینهٔ اعتبارسنجی در طول آموزش (هر ۵۰۰۰ گام یک نمونه).
- عدد μ (میانگین هزینه) و σ (انحراف معیار هزینه) در «آخرین N دورهٔ اعتبارسنجی» (مثلاً آخرین ۵ یا ۱۰ نقطه).

7-1. تحلیل نتایج (ده نمره)

۱. مقایسهٔ کمی عملکرد

1. **میانگین و انحراف معیار هزینهٔ نهایی**
   - از نتایج هر run، میانگین هزینهٔ اعتبارسنجی در چند آخرین نقطه (N آخر: مثلاً آخرین ۵ نقطه از در بازهٔ ۲۰۰,۰۰۰ گام) را محاسبه کنید.
     همچنین انحراف معیار آن را به‌دست آورید.
   - جدولی شبیه زیر برای هر حالت (بدون Shaping، BSP-low-EW Shaping، Base-Stock Shaping) بسازید:

حالت مدل	μ(هزینه)	σ(هزینه)	تعداد Run
DQN بدون Shaping			5
DQN + Base-Stock Shaping			5
DQN + BSP-low-EW Shaping			5

2. **اختلاف نسبی هزینه نسبت به BSP-low-EW (برای *m*=2)**

   - فرض کنید هزینهٔ پایدار (به‌عنوان مبنا) همان «BSP-low-EW» است. سپس برای هر حالت و هر run محاسبه کنید:

$$\text{RelDiff}(\%) = \frac{\text{Cost(model)} - \text{Cost(BSP-low-EW)}}{\text{Cost(BSP-low-EW)}} \times 100.$$

   - جدولی تهیه کنید با ستون‌های: «حالت مدل»، «μ(هزینه) از مدل»، «μ(هزینه) از BSP-low-EW»، «RelDiff (%)».

۲. تحلیل بصری همگرایی

1. **نمودار «Optimality Gap» (برای experiment 2، m=2)**

- فرض کنید در طول آموزش (برای experiment 2)، در هر دوره (مثلاً هر ۵۰۰۰ گام) هزینهٔ اعتبارسنجی را اندازه‌گیری کرده‌اید.
- برای هر یک از سه حالت (بدون BSP-low-EW، Base-Stock Shaping، Shaping)، میانگینِ اختلاف هزینه نسبت به یک «حد پایینِ فرضی» (مثلاً هزینهٔ بهینهٔ دینامیک‌برنامه‌ریزی یا هزینهٔ BSP-low-EW) را در هر نقطه محاسبه کنید.
- سپس:
  - نمودار بالایی (Top Panel): برای هر حالت، منحنی میانگینِ اختلاف هزینه (Optimality Gap) را بر حسب اپیزود (هر اپیزود برابر با یک بلوک مشخص از گام‌ها، مثلاً هر ۵۰۰۰ گام) رسم کنید و برای هر نقطه نوار خطای ±۱.۹۵σ را روی ۵ run به‌صورت "نواحی سایه‌دار (shaded 95% CI)" نمایش دهید.
  - نمودار پایینی (Bottom Panel): برای هر حالت، تنها منحنی «بهترینِ run» هر مدل (Lowest-gap run) را رسم کنید (بدون نوار خطا).
- هدف:
  - مقایسهٔ سرعت همگرایی (کدام حالت زودتر gap را تا مثلا زیر ۵٪ می‌رساند)
  - بررسی پایداری (طیف نواحی سایه‌دار در نمودار بالایی)
  - مشاهدهٔ بهترین run (نمودار پایینی) تا ببینید در عمل حداکثر چقدر می‌توان gap را کاهش داد.

2. **نمودار «رفتار هزینه تجمعی» (Cumulative Reward/Cost)**

- برای هریک از سه حالت و هر run، «میانگین هزینهٔ اعتبارسنجی» را به‌صورت یک منحنی پیوسته بر حسب مراحل آموزشی (هر نمونه = ۵۰۰۰ گام) رسم کنید.
- پنج منحنی در یک شکل (هر run یک رنگ) نشان دهد که چطور مدل‌ها نوسان دارند و کجا همگرا می‌شوند.

۳. تحلیل سیاست و توزیع پایدار

۳-۱. نمایش سیاست‌ها و توزیع «Steady State» برای experiment 1 و experiment 2

شکل ۳ (Figure 3):

- هر ستون مربوط به یک experiment است (ستون ۱: experiment 1، ستون ۲: experiment).
- هر نمودار (= هر ردیف):
  1. ردیف اول: «heatmap سیاست» که نشان می‌دهد در هر state (سناریوی age_2 on Y و age_1 on X)، تعداد سفارش (action) چقدر باشد (۰ تا ۳).
  2. ردیف دوم: «heatmap احتمال حالت پایدار» (steady-state probability) برای همان state.

برای هر experiment:

1. تولید سیاست‌ها
   - حالت A: Base-Stock (با S بهینه)
   - حالت B: BSP-low-EW
   - حالت C: Unshaped DQN
   - حالت D: Shaped DQN (BSP-low-EW)

   در هر نمودار، عدد در هر خانه برابر با «سفارش ایده‌آل» (۰–۳) است.
2. تولید توزیع Steady State
   - برای هر سیاست، مدتی طولانی (مثلاً ۲۰۰,۰۰۰ گام) شبیه‌سازی کنید و فرکانس ورود به هر state (age_2, age_1) را محاسبه نمایید.
   - سپس آن را نرمالیزه کنید تا احتمال برود.
   - «heatmap احتمال» را با طیفی از زرد (احتمال کم) تا قرمز (احتمال زیاد) نشان دهید.

تحلیل مورد انتظار برای شکل ۳:

- مقایسهٔ ساختار سیاست‌ها (ردیف اول):
  - Base-Stock چه الگویی دارد؟
  - BSP-low-EW چه تفاوتی با Base-Stock نشان می‌دهد؟
  - Unshaped DQN چگونه نواحی «سفارش (۳) یا ۰» را تشخیص داده؟
  - Shaped DQN چگونه به‌تدریج (۱ و ۲) سفارش می‌دهد و نواحی policy را بهبود می‌بخشد؟
- مقایسهٔ توزیع‌ها (ردیف دوم):
  - سیاست‌های متفاوت کدام states را بیشترین احتمال پایدار می‌بینند؟
  - آیا شاپ شده‌ها (Base-Stock shaping یا BSP-low-EW shaping) توزیع را به سمت states با age_2 کم و age_1 متوسط متمایل کرده‌اند؟

- آیا Unshaped DQN توزیع را به ناحیه‌ای با هزینهٔ نسبی بالاتر هدایت می‌کند؟

۴. «Relative Cost Difference» برای m=2,3,4,5

شکل مربوط: نمودار میله‌ای «Relative Cost Difference (%) نسبت به BSP-low-EW» برای همهٔ پروفایل‌های مختلف (جدول ۱ مقاله).

1. **هشت تنظیم (Table 1)**:
   - پارامترها (m, L, cp, issuing) دقیقاً همان هشت حالت از مقاله:
     1. (m=2, L=1, cp=7, LIFO)
     2. (m=2, L=1, cp=7, FIFO)
     3. (m=2, L=1, cp=10, LIFO)
     4. (m=2, L=1, cp=10, FIFO)
     5. (m=2, L=2, cp=7, LIFO)
     6. (m=2, L=2, cp=7, FIFO)
     7. (m=2, L=2, cp=10, LIFO)
     8. (m=2, L=2, cp=10, FIFO)

   برای هر هشت حالت، مدل‌های زیر را اجرا کنید و هزینهٔ نهایی BSP-low-EW و Shaped DQN (BSP-low-EW shaping) را با شبیه‌سازی طولانی (مثلاً ۵۰,۰۰۰ گام) به‌دست آورید.
2. **محاسبهٔ «Relative Cost Difference»**

   برای هر حالت از هشت پروفایل:

   گروه‌بندی بر حسب عمرهای مختلف (m=2,3,4,5)
   - سه دسته مجزا رسم کنید (یا یک شکل بزرگ با چهار بخش sub-plot):
     1. نتایج برای m = 2 (هر ۸ حالت را با هم)
     2. نتایج برای m = 3
     3. نتایج برای m = 4
     4. نتایج برای m = 5
   - محور عمودی: «RelDiff (%)»
   - محور افقی: ایندکسِ ۸ حالت (می‌توانید زیرنویسِ مختصری بنویسید: "(L=1,cp=7,LIFO)", ...)؛ یا اگر sub-plot استفاده کردی، محور افقی از 1 تا 8 اندیس‌بندی شود.

3. **تحلیل نسبی**
   - نشان دهید که در هر عمر (m)، چقدر Shaped DQN بهبود یا افت نسبت به BSP-low-EW دارد.
   - در عمل، مقاله نشان می‌دهد که با افزایش «RelDiff ،m» به صفر نزدیک‌تر می‌شود (چون BSP-low-EW به‌مرور نزدیک به بهینه می‌شود).
   - همچنین، مجموعهٔ ۸ حالت را با هم مقایسه کنید تا ببینید کدام ترکیب (مثلاً LIFO vs FIFO یا cp=7 vs cp=10) بیشترین سود را از shaping می‌برد.

۵. نتیجه‌گیری کلی

در نهایت، همهٔ مشاهدات بالا را در ۳–۴ جمله خلاصه کنید. مثلاً:

- "در experiment 2 (m=2, L=1), Shaped DQN با BSP-low-EW تقریباً 0.1 واحد هزینهٔ کمتری نسبت به Base-Stock دارد. نمودار همگرایی نشان می‌دهد که Shaped DQN نه‌تنها سریع‌تر به یک هزینهٔ پایدار رسید، بلکه نوسانات آن نیز در 95% بازهٔ اطمینان کمتر است.
- با بررسی RelDiff در هشت پروفایل و m=2..5، مشاهده شد که وقتی m افزایش پیدا می‌کند، بهبود Shaped DQN نسبت به BSP-low-EW به تدریج کاهش می‌یابد—چرا که خودِ BSP-low-EW به بهینه نزدیک می‌شود.
- در نهایت، Shaped DQN به ویژه برای عمرهای کوتاه (m=2 یا 3) و زمانی که cp کم است یا LIFO اجرا می‌شود، بیشترین سود را نسبت به BSP-low-EW داشته است."

پرسش 2: مسیریابی ربات با DM-DQN و مقایسه با DQN و M-DQN

2-1. معرفی مقاله

در این قسمت مقاله‌ای با عنوان
DM-DQN: Dueling Munchausen deep Q network for robot path planning مطالعه و پیاده‌سازی می‌شود. این مقاله ترکیبی از دو بهبود کلیدی بر معماری DQN را برای حل مسئله مسیریابی ربات در محیط‌های پیچیده پیشنهاد می‌دهد:

- **معماری Dueling DQN**:

در این معماری، شبکه Q به دو شاخه مجزا تقسیم می‌شود:

- یک شاخه برای تخمین مقدار وضعیت (State-Value) یعنی $V(s)$
- و شاخه دیگر برای تخمین مزیت هر عمل نسبت به میانگین اعمال (Advantage) یعنی $A(s,a)$

در نهایت این دو خروجی ترکیب می‌شوند تا مقدار Q نهایی محاسبه شود:

$$\left(A(s,a) - \frac{1}{|A|}\sum_{'a} A(s,a')\right) + V(s) = Q(s,a)$$

این ساختار به شبکه اجازه می‌دهد تفاوت ظریف بین اعمال را بهتر یاد بگیرد و به سرعت همگرایی بالاتری برسد.

- $Munchausen\ Reinforcement\ Learning\ (M-RL)$:

در این تکنیک، به پاداش دریافتی عامل، یک ترم اضافی اضافه می‌شود که به صورت log-policy تعریف شده است:

$$\alpha\tau \log \pi(a|s) + r = {}'r$$

که در آن:

- α ضریب اهمیت ترم جدی
- τ پارامتر دما (temperature)

- و $\pi(a|s)$ احتمال انتخاب عمل در حالت s می‌باشد.

این ترم باعث افزایش **اکتشاف ایمن و پایدار** می‌شود و سیاست یادگیری را به سمت حفظ اطلاعات قدیمی‌تر و کاهش نوسانات هدایت می‌کند.

و در این رابطه ترم $log\pi(a|s)$ با استفاده از softmax بر روی Q-values محاسبه می‌شود:

$$\frac{e^{Q(s,a)/\tau}}{\sum_{b} e^{Q(s,b)/\tau}} = \pi(a|s)$$

- ترم log-policy نقش **راهنمایی برای سیاست اکتشافی عامل** را دارد و اجازه می‌دهد عامل به جای رفتار کاملاً حریصانه، رفتار منطقی و پایدار داشته باشد.

مزایای استفاده از این ترم:

- کاهش overestimation در Q-values
- افزایش تنوع در اکتشاف
- سرعت همگرایی بیشتر نسبت به DQN یا Dueling DQN

- تابع پاداش مبتنی بر $Artificial\ Potential\ Field\ (APF)$:

در این روش پاداش‌دهی، نیروهای جاذبه از سوی هدف و نیروهای دافعه از سوی موانع تعریف می‌شوند تا عامل به‌طور طبیعی مسیر بهینه و بدون برخورد را یاد بگیرد. مزیت این روش، طراحی مسیرهای نرم‌تر و امن‌تر برای حرکت ربات است.

- **مروری بر معادلات Artificial Potential Field**

پاداش کل به‌صورت ترکیبی از سه بخش تعریف می‌شود:

1. نیروی جاذبه (جذب به هدف):

$$\zeta d^2(q, q_{goal}) \frac{1}{2} = U_{att}(q)$$

2. نیروی دافعه (دوری از موانع):

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{Q^*} - \frac{1}{D(q)}\right)^2, & D(q) < Q^* \\ 0, & D(q) \geq Q^* \end{cases}$$

3. پاداش جهت (زاویه حرکت):

$$reward_{yaw} = \frac{\phi\%(2\pi)}{\pi} \quad \text{که} \quad \phi = \arccos\left(\frac{{}_aF_q \cdot F}{|{}_aF_q||F|}\right)$$

پاداش نهایی ترکیبی:

$$reward_{yaw} + reward_{rep} + reward_{att} = R$$

2-2. مطالعه و تحلیل مقاله (ده نمره)

- برای درک بهتر ایده‌های مقاله، با توجه به مفاهیم توضیح داده شده در قسمت قبل باید مقاله را مطالعه کرده و کاربرد های هر یک از مفاهیم ارائه شده را در مقاله به دقت بررسی کنید.

2-3. پیاده‌سازی مدل‌ها و تابع پاداش در محیط رانندگی Highway (سی و پنج نمره)

در این بخش، شما باید سه الگوریتم یادگیری تقویتی را در یک محیط رانندگی مجازی (`highway-env`) پیاده‌سازی و آموزش دهید و تابع پاداش آن را نیز بر اساس روش $Artificial\ Potential\ Field\ (APF)$ طراحی کنید. هدف کلی ما پیاده‌سازی و آموزش سه مدل زیر است:

1. DQN ساده (baseline)
2. Dueling DQN (تقسیم شاخه‌های ارزش و مزیت)
3. DM-DQN (ترکیب Dueling + ترم Munchausen)

همچنین می‌بایست به طراحی تابع پاداش مبتنی بر میدان پتانسیل مصنوعی (APF) پرداخته و عملکردها را مقایسه و رفتارهای مدل‌ها را تحلیل کنیم

4-2. انتخاب فریم‌ورک

شما میتوانید یکی از دو مسیر زیر را انتخاب کنند:

مسیر ۱: استفاده از Stable-Baselines3 (ساده‌تر و ساختاریافته‌تر)

1. **محیط سفارشی `highway-env` را بسازید**
   - استفاده از Wrapper (لایه‌ای است که روی محیط اصلی قرار می‌گیرد و امکان تغییر یا گسترش رفتار محیط را بدون تغییر در کد اصلی فراهم می‌کند.) برای تنظیم ویژگی‌ها و فضای عمل:

```
{
  "observation": {
    "type": "Kinematics",
    "vehicles_count": 5,
    "features": ["presence", "x", "y", "vx", "vy"],
    "absolute": True
  },
  "action": {"type": "DiscreteMetaAction"},
  "duration": 40,
  "lanes_count": 3,
  "collision_reward": -1,
  "high_speed_reward": 0.4,
  "right_lane_reward": 0.1,
  "normalize_reward": True
}
```

2. شبکه‌های سفارشی زیر را پیاده سازی کنید:

- `CustomDuelingNetwork`: با دو شاخهٔ V و A
- `CustomMunchausenNetwork`: پیاده‌سازی log-policy و softmax در forward
- استفاده از `DQNPolicy` و تعریف کلاس‌های جدید برای Dueling و DM-DQN

3. Munchausen Term را در تابع loss اضافه کنید:

```
target = r + ατ log π(a|s) + γ V(s')
```

4.حلقه آموزش و ارزیابی را برای حداقل ۱۰,۰۰۰ گام(با حداقل سه seed رندوم مختلف) آموزش داده و میانگین پاداش episode 10 پایانی متوسط طول مسیر ، درصد برخورد را به دست بیاورید و نمودار پاداش تجمعی و عملکرد نهایی مدل ها را رسم کنید.

مسیر ۲: پیاده‌سازی کامل با PyTorch (پیشرفته‌تر و منعطف‌تر)

مراحل پیاده‌سازی:

1. محیط `highway-v0` Gym را بسازید: با استفاده از `env.configure({...})` برای تنظیمات مشابه بالا
2. شبکه‌های زیر را تعریف کنید:
   - `DM_DQNNet`، `DuelingNet`، `DQNNet` با استفاده از PyTorch
   - در DM-DQN از softmax برای محاسبه سیاست و log-policy استفاده شود.
3. ReplayBuffer و کلاس Agent را مطابق موارد زیر تعیریف کنید:
   - انتخاب عمل با Epsilon-Greedy
   - محاسبه Q-target با Munchausen یا بدون آن
   - به‌روزرسانی شبکه با MSE Loss یا Huber
4. حلقه آموزش و ارزیابی را برای حداقل ۱۰,۰۰۰ گام(با حداقل سه seed رندوم مختلف) آموزش داده و میانگین پاداش episode 10 پایانی متوسط طول مسیر ، درصد برخورد را به دست بیاورید و نمودار پاداش تجمعی و عملکرد نهایی مدل ها را رسم کنید.

2-5. طراحی تابع پاداش با APF

برای اینکه عامل به‌درستی به هدف نزدیک شود و از موانع دور بماند، باید تابع پاداش را مطابق موارد توضیح داده شده پیاده سازی کنید.

6-2. تحلیل نتایج و ارزیابی نهایی (پانزده نمره)

معیارهای ارزیابی (Evaluation Metrics)

برای هر یک از مدل‌ها (DQN، Dueling DQN، DM-DQN)، باید پس از آموزش، مدل‌ها را به‌صورت کمی ارزیابی کنید. موارد زیر باید اندازه‌گیری و گزارش شوند:

پاداش میانگین نهایی

- میانگین پاداش در ۱۰ اپیزود پس از اتمام آموزش (evaluation episodes)
- هدف: بررسی میزان یادگیری سیاست بهینه

تعداد گام‌ها تا رسیدن به هدف (میانگین طول مسیر)

- میانگین تعداد مراحل اپیزودها تا رسیدن به هدف نهایی
- هدف: مقایسه کارایی و بهره‌وری مدل‌ها در یافتن مسیر کوتاه‌تر

نرخ برخورد (collision rate)

- درصد اپیزودهایی که در آن‌ها عامل با مانع برخورد داشته است
- هدف: ارزیابی ایمنی و توانایی اجتناب از مانع

نمودارهای مورد نیاز (Visualization Requirements)

برای تحلیل روند یادگیری و مقایسه نهایی، باید نمودارهای زیر را در گزارش قرار دهید:

نمودار پاداش تجمعی در طول اپیزودها

- نمایش پاداش در هر اپیزود
- رسم Moving Average با پنجره ۲۰
- برای هر مدل به صورت جداگانه (شامل نسخه خام و MA)

مقایسه روند یادگیری

- نمودار ترکیبی Moving Average پاداش ۳ مدل در کنار هم
- هدف: مقایسه نرخ همگرایی و ثبات عملکرد

نمودار میله‌ای مقایسه عملکرد نهایی

- Bar chart برای میانگین پاداش نهایی مدل‌ها (پس از ارزیابی)
- شامل برچسب مدل‌ها و مقدار عددی بالای هر ستون

تحلیل مورد انتظار (Required Interpretation & Insights)

در گزارش نهایی، شما باید:

- توضیح دهید که چرا DM-DQN بهتر (یا بدتر) از DQN یا Dueling DQN عمل کرده است.
- به نقش ترم Munchausen و ساختار Dueling در یادگیری بهتر اشاره کنید.
  تابع پاداش APF، تأثیر آن در کاهش برخوردها یا بهبود نرمی مسیر را بررسی کنید.
- تحلیل کنید که در کدام محیط (با موانع متحرک یا ثابت) مدل‌ها بهتر عمل کردند (در صورت اجرای چنین سناریویی).