RL Homework 1

Name: MU FENG CHUA
ID: v111136

Problem 1 Q-Value Iteration (a-1)

1. **Show $\boldsymbol{V^*(s) \leq \max_{a\in\mathcal{A}} Q^*(s,a)}$**

   By definition, $V^*(s) = \max_{a\in\mathcal{A}} Q^*(s,a)$ and $Q^*(s,a) \geq Q^\pi(s,a)$ for all π, hence:

   $$V^\pi(s) \leq \sum_{a\in\mathcal{A}} \pi(a|s) \cdot Q^*(s,a) \leq \max_{a\in\mathcal{A}} Q^*(s,a)$$

   Take $\pi^* = \arg\max_{\pi\in\Pi} V^\pi(s)$:

   $$V^*(s) = V^{\pi^*}(s) \leq \max_{a\in\mathcal{A}} Q^*(s,a)$$

   Hence:

   $$V^*(s) \leq \max_{a\in\mathcal{A}} Q^*(s,a)$$

2. **Show $\boldsymbol{V^*(s) \not< \max_{a\in\mathcal{A}} Q^*(s,a)}$**

   Assume $V^*(s) < \max_{a\in\mathcal{A}} Q^*(s,a)$ and prove by contradiction. Let $a^* = \arg\max_{a\in\mathcal{A}} Q^*(s,a)$, then:

   $$Q^*(s,a^*) \geq \max_{a\in\mathcal{A}} Q^*(s,a).$$

   Since $V^*(s)$ is the optimal reward for all states s, we have $V^*(s) \geq Q^*(s,a^*)$ and this assumption leads to a contradiction. Therefore:

   $$V^*(s) \not< \max_{a\in\mathcal{A}} Q^*(s,a).$$

   By 1. and 2.:

   $$V^*(s) = \max_{a\in\mathcal{A}} Q^*(s,a). \qquad \square$$

Problem 1 Q-Value Iteration (a-2)

By definition, $Q^\pi(s,a) = R_{s,a} + \gamma \sum_{s'} P^a_{ss'} V^*(s)$ for all π, since $Q*(s,a) = \max_{\pi\in\Pi} Q^\pi(s,a)$:

$$Q^*(s,a) = \max_\pi [R_{s,a} + \gamma \sum_{s'} P^a_{ss'} V^\pi(s')]$$

and because only $V^\pi(s)$ is affect by $\max_\pi$, we can simplify the equation into:

$$\begin{aligned} Q^*(s,a) &= R_{s,a} + \gamma \sum_{s'} P^a_{ss'} \max_\pi \big(V^\pi(s')\big) \\ &= R_{s,a} + \gamma \sum_{s'} P^a_{ss'} \big(V^*(s')\big) \qquad \square \end{aligned}$$

Problem 1 Q-Value Iteration (b-1)

To show T^* is a $\gamma -$ contraction operator, we simply prove,

$$\| \, T^*(Q) - T^*(Q') \, \|_\infty \leq \gamma \, \| \, Q - Q' \, \|_\infty, \forall \text{ Q-function } Q, Q'$$

where,

$$\| \, Q - Q' \, \|_\infty = \max_{s,a} | \, Q(s,a) - Q'(s,a) \, |$$

For all Q-function Q, Q':

$$\begin{aligned}
[T^*(Q) - T^*(Q')](s,a) &= R_{s,a} + \gamma \sum_{s'} P^a_{ss'} \max_{a'}(Q(s',a')) - [\, R_{s,a} + \gamma \sum_{s'} P^a_{ss'} \max_{a'}(Q'(s',a')) \,] \\
&= \gamma \sum_{s'} P^a_{ss'} [\, \max_{a'}(Q(s',a')) - \max_{a'}(Q'(s',a')) \,] \\
&\leq \gamma \sum_{s'} P^a_{ss'} [\, \max_{a'}(Q(s',a') - Q'(s',a')) \,]
\end{aligned}$$

Since for all (s', a') pair,

$$Q(s',a') - Q'(s',a')) \leq \| \, Q - Q' \, \|_\infty$$

Hence,

$$\begin{aligned}
\gamma \sum_{s'} P^a_{ss'} [\, \max_{a'}(Q(s',a') - Q'(s',a')) \,] &\leq \gamma \| \, Q - Q' \, \|_\infty \sum_{s'} P^a_{ss'} \\
&= \gamma \| \, Q - Q' \, \|_\infty
\end{aligned}$$

And because the above hold for all (s, a) pair

$$[T^*(Q) - T^*(Q')](s,a) \leq \gamma \| \, Q - Q' \, \|_\infty$$

Therefore,

$$\max_{s,a} | \, T^*(Q) - T^*(Q') \, | = \| \, T^*(Q) - T^*(Q') \, \|_\infty \leq \gamma \, \| \, Q - Q' \, \|_\infty \qquad \square$$

Problem 1 Q-Value Iteration (b-2)

Probably still works on some special cases, but normally it will not work. We can prove it by giving a counter-example.

Assume there is only one state s, $\{a, b\}$ two action, reward $= 0$, discount factor $\gamma > 0$ and we define Q-functions as:

$$Q(s,a) = 1, \; Q(s,b) = 0$$

$$Q'(s,a) = 0, \; Q'(s,b) = 0$$

Apply T^* to both Q and Q', we will get,

$$[T^*(Q)](s,a) = \gamma \max\{Q(s,a), Q(s,b)\} = \gamma \max\{1, 0\} = \gamma,$$

$$[T^*(Q)](s,b) = \gamma \max\{Q(s,a), Q(s,b)\} = \gamma \max\{1, 0\} = \gamma.$$

$$[T^*(Q')](s,a) = \gamma \max\{Q'(s,a), Q'(s,b)\} = \gamma \max\{0, 0\} = 0,$$

$$[T^*(Q')](s,b) = \gamma \max\{Q'(s,a), Q'(s,b)\} = \gamma \max\{0, 0\} = 0.$$

And we have:

$$\|Q - Q'\|_1 = 1 \text{ and } \|T^*(Q) - T^*(Q')\|_1 = 2\gamma$$

For T^* to be a γ-contraction in the ℓ_1-norm, we would require:

$$\|T^*(Q) - T^*(Q')\|_1 \leq \gamma \|Q - Q'\|_1.$$

Applying the value we get above, we will get:

$$2\gamma \leq \gamma.$$

Since $\gamma > 0$,

$$2\gamma < \gamma \implies 2 < 1$$

Which is impossible.

Problem 2

By definition, the discounted state visitation distribution is:

$$d_\mu^\pi(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t P\left(s_t = s \mid s \sim \mu, \pi\right)$$

We start the proof from the right-hand side:

$$\begin{aligned}
\text{RHS} &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)}[f(s,a)] \\
&= \frac{1}{1-\gamma} \sum_s d_\mu^{\pi_\theta} \sum_a \pi_\theta(a \mid s)[f(s,a)] \\
&= \frac{1}{1-\gamma} \sum_s (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s \mid s_0 \sim \mu, \pi_\theta) \sum_a \pi_\theta(a \mid s)[f(s,a)] \\
&= \sum_s \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s \mid s_0 \sim \mu, \pi_\theta) \sum_a \pi_\theta(a \mid s)[f(s,a)] \\
&= \sum_{t=0}^{\infty} \gamma^t \sum_s \Pr(s_t = s \mid s_0 \sim \mu, \pi_\theta) \sum_a \pi_\theta(a \mid s)[f(s,a)] \\
&= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{s_t \sim P_\mu^{\pi_\theta}, a_t \sim \pi_\theta(\cdot|s_t)}[f(s_t,a_t)] && (*) \\
&= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\tau \sim P_\mu^{\pi_\theta}}[f(s_t,a_t)]
\end{aligned}$$

By linearity of expectation:

$$\sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\tau \sim P_\mu^{\pi_\theta}}[f(s_t,a_t)] = \mathbb{E}_{\tau \sim P_\mu^{\pi_\theta}} \sum_{t=0}^{\infty} \gamma^t [f(s_t,a_t)] = \text{LHS}$$

Explanation of $(*)$: The expression $\Pr(s_t = s \mid s_0 \sim \mu, \pi_\theta)$ represents the probability of being in state s at time t. Similarly, $\pi_\theta(a \mid s)$ represents the probability of choosing action a given the state s. Multiplying these probabilities together, we obtain:

$$\Pr(s_t = s \mid s_0 \sim \mu, \pi_\theta) \cdot \pi_\theta(a \mid s)$$

which intuitively denotes the probability of being in state s at time t and then choosing action a in that state. Thus, we simplify this combined probability expression into the form given in $(*)$.

Problem 3

The given parameter values are:

$$\theta_a = 0, \quad \theta_b = \ln 5, \quad \theta_c = \ln 4.$$

Thus,

$$\exp(\theta_a) = 1, \quad \exp(\theta_b) = 5, \quad \exp(\theta_c) = 4,$$

and the normalization is

$$Z = 1 + 5 + 4 = 10.$$

The corresponding action probabilities are

$$\pi(a) = 0.1, \quad \pi(b) = 0.5, \quad \pi(c) = 0.4.$$

For a one-step trajectory (from s to termination) the REINFORCE gradient estimate is given by

$$\hat{\nabla} V = \nabla \log \pi_\theta(a \mid s)\, r(s,a),$$

where a is the action sampled according to $\pi_\theta(\cdot \mid s)$. Standard results tell us that

$$\nabla \log \pi_\theta(x \mid s) = \mathbf{e}_x - \pi_\theta,$$

where $\mathbf{e}_x$ is the one–hot vector for action x and $\pi_\theta = (\pi(a), \pi(b), \pi(c))$.

Thus, for each action we have:

- For a: $\nabla \log \pi_\theta(a \mid s) = \big[1 - \pi(a),\ -\pi(b),\ -\pi(c)\big] = [0.9,\ -0.5,\ -0.4]$.
- For b: $\nabla \log \pi_\theta(b \mid s) = \big[-\pi(a),\ 1 - \pi(b),\ -\pi(c)\big] = [-0.1,\ 0.5,\ -0.4]$.
- For c: $\nabla \log \pi_\theta(c \mid s) = \big[-\pi(a),\ -\pi(b),\ 1 - \pi(c)\big] = [-0.1,\ -0.5,\ 0.6]$.

For each action $x \in \{a, b, c\}$ the gradient estimate (before taking expectation) is:

$$g(x) = r(s,x)\, \nabla \log \pi_\theta(x \mid s).$$

Thus, we obtain:

- $g(a) = 100\,[0.9,\ -0.5,\ -0.4] = [90,\ -50,\ -40]$,
- $g(b) = 98\,[-0.1,\ 0.5,\ -0.4] = [-9.8,\ 49,\ -39.2]$,
- $g(c) = 95\,[-0.1,\ -0.5,\ 0.6] = [-9.5,\ -47.5,\ 57]$.

Problem 3 (a)

The mean (expected) gradient is given by:

$$E[\hat{\nabla} V] = \pi(a) g(a) + \pi(b) g(b) + \pi(c) g(c).$$

Calculating coordinate–wise:

- First coordinate:
$$0.1 \cdot 90 + 0.5 \cdot (-9.8) + 0.4 \cdot (-9.5) = 9 - 4.9 - 3.8 = 0.3.$$
- Second coordinate:
$$0.1 \cdot (-50) + 0.5 \cdot 49 + 0.4 \cdot (-47.5) = -5 + 24.5 - 19 = 0.5.$$

- Third coordinate:

$$0.1 \cdot (-40) + 0.5 \cdot (-39.2) + 0.4 \cdot 57 = -4 - 19.6 + 22.8 = -0.8.$$

Thus,

$$E[\hat{\nabla} V] = [0.3,\ 0.5,\ -0.8].$$

The covariance is given by

$$\mathrm{Cov}(\hat{\nabla} V) = E\Big[(\hat{\nabla} V - E[\hat{\nabla} V])(\hat{\nabla} V - E[\hat{\nabla} V])^\top\Big] = E[\hat{\nabla} V\, \hat{\nabla} V^\top] - E[\hat{\nabla} V]\, E[\hat{\nabla} V]^\top.$$

We first compute the second moment matrix:

$$E[\hat{\nabla} V\, \hat{\nabla} V^\top] = \pi(a)\, g(a)g(a)^\top + \pi(b)\, g(b)g(b)^\top + \pi(c)\, g(c)g(c)^\top.$$

The outer products are:

- For a:

$$g(a)g(a)^\top = \begin{bmatrix} 90^2 & 90 \cdot (-50) & 90 \cdot (-40) \\ (-50) \cdot 90 & (-50)^2 & (-50) \cdot (-40) \\ (-40) \cdot 90 & (-40) \cdot (-50) & (-40)^2 \end{bmatrix} = \begin{bmatrix} 8100 & -4500 & -3600 \\ -4500 & 2500 & 2000 \\ -3600 & 2000 & 1600 \end{bmatrix}.$$

  Multiplying by $\pi(a) = 0.1$ yields:

$$0.1\, g(a)g(a)^\top = \begin{bmatrix} 810 & -450 & -360 \\ -450 & 250 & 200 \\ -360 & 200 & 160 \end{bmatrix}.$$

- For b:

$$g(b)g(b)^\top = \begin{bmatrix} (-9.8)^2 & (-9.8)(49) & (-9.8)(-39.2) \\ (-9.8)(49) & 49^2 & 49(-39.2) \\ (-9.8)(-39.2) & (-39.2)(49) & (-39.2)^2 \end{bmatrix} \approx \begin{bmatrix} 96.04 & -480.2 & 384.16 \\ -480.2 & 2401 & -1920.8 \\ 384.16 & -1920.8 & 1536.64 \end{bmatrix}.$$

  Multiplying by $\pi(b) = 0.5$ gives:

$$0.5\, g(b)g(b)^\top \approx \begin{bmatrix} 48.02 & -240.1 & 192.08 \\ -240.1 & 1200.5 & -960.4 \\ 192.08 & -960.4 & 768.32 \end{bmatrix}.$$

- For c:

$$g(c)g(c)^\top = \begin{bmatrix} (-9.5)^2 & (-9.5)(-47.5) & (-9.5)(57) \\ (-9.5)(-47.5) & (-47.5)^2 & (-47.5)(57) \\ (-9.5)(57) & (-47.5)(57) & 57^2 \end{bmatrix} = \begin{bmatrix} 90.25 & 451.25 & -541.5 \\ 451.25 & 2256.25 & -2707.5 \\ -541.5 & -2707.5 & 3249 \end{bmatrix}.$$

  Multiplying by $\pi(c) = 0.4$ gives:

$$0.4\, g(c)g(c)^\top \approx \begin{bmatrix} 36.1 & 180.5 & -216.6 \\ 180.5 & 902.5 & -1083 \\ -216.6 & -1083 & 1299.6 \end{bmatrix}.$$

Now, summing these three contributions we get:

$$E[\hat{\nabla} V\,\hat{\nabla} V^\top] \approx \begin{bmatrix} 810+48.02+36.1 & -450-240.1+180.5 & -360+192.08-216.6 \\ -450-240.1+180.5 & 250+1200.5+902.5 & 200-960.4-1083 \\ -360+192.08-216.6 & 200-960.4-1083 & 160+768.32+1299.6 \end{bmatrix}.$$

This simplifies approximately to:

$$E[\hat{\nabla} V\,\hat{\nabla} V^\top] \approx \begin{bmatrix} 894.12 & -509.6 & -384.52 \\ -509.6 & 2353 & -1843.4 \\ -384.52 & -1843.4 & 2227.92 \end{bmatrix}.$$

Finally, subtract the outer product of the mean vector

$$E[\hat{\nabla} V]\,E[\hat{\nabla} V]^\top = \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix} \begin{bmatrix} 0.3 & 0.5 & -0.8 \end{bmatrix} = \begin{bmatrix} 0.09 & 0.15 & -0.24 \\ 0.15 & 0.25 & -0.4 \\ -0.24 & -0.4 & 0.64 \end{bmatrix}.$$

Thus, the covariance matrix is approximately:

$$\mathrm{Cov}(\hat{\nabla} V) \approx \begin{bmatrix} 894.12-0.09 & -509.6-0.15 & -384.52+0.24 \\ -509.6-0.15 & 2353-0.25 & -1843.4+0.4 \\ -384.52+0.24 & -1843.4+0.4 & 2227.92-0.64 \end{bmatrix} = \begin{bmatrix} 894.03 & -509.75 & -384.28 \\ -509.75 & 2352.75 & -1843.0 \\ -384.28 & -1843.0 & 2227.28 \end{bmatrix}.$$

Problem 3 (b)

When we subtract the baseline $V^{\pi_\theta}(s)$, the gradient estimator becomes

$$\tilde{\nabla} V = \nabla \log \pi_\theta(a \mid s)\,\Big(r(s,a) - V^{\pi_\theta}(s)\Big).$$

Since there is only one state,

$$V^{\pi_\theta}(s) = \sum_{x \in \{a,b,c\}} \pi(x) r(s,x) = 0.1 \cdot 100 + 0.5 \cdot 98 + 0.4 \cdot 95 = 10 + 49 + 38 = 97.$$

Now, for each action we have:

- For a: $r(s,a) - 97 = 100 - 97 = 3, \quad \tilde{g}(a) = 3\,[0.9,\ -0.5,\ -0.4] = [2.7,\ -1.5,\ -1.2].$
- For b: $98 - 97 = 1, \quad \tilde{g}(b) = 1\,[-0.1,\ 0.5,\ -0.4] = [-0.1,\ 0.5,\ -0.4].$
- For c: $95 - 97 = -2, \quad \tilde{g}(c) = -2\,[-0.1,\ -0.5,\ 0.6] = [0.2,\ 1.0,\ -1.2].$

The expected gradient now is

$$E[\tilde{\nabla} V] = 0.1\,\tilde{g}(a) + 0.5\,\tilde{g}(b) + 0.4\,\tilde{g}(c).$$

Compute coordinate–wise:

- First coordinate:
$$0.1 \cdot 2.7 + 0.5 \cdot (-0.1) + 0.4 \cdot 0.2 = 0.27 - 0.05 + 0.08 = 0.3.$$
- Second coordinate:
$$0.1 \cdot (-1.5) + 0.5 \cdot 0.5 + 0.4 \cdot 1.0 = -0.15 + 0.25 + 0.4 = 0.5.$$

- Third coordinate:

$$0.1 \cdot (-1.2) + 0.5 \cdot (-0.4) + 0.4 \cdot (-1.2) = -0.12 - 0.2 - 0.48 = -0.8.$$

Thus, we still have

$$E[\tilde{\nabla} V] = [0.3,\ 0.5,\ -0.8],$$

Now, we compute the second moment for the baseline estimator:

$$E[\tilde{\nabla} V\, \tilde{\nabla} V^\top] = 0.1\, \tilde{g}(a)\tilde{g}(a)^\top + 0.5\, \tilde{g}(b)\tilde{g}(b)^\top + 0.4\, \tilde{g}(c)\tilde{g}(c)^\top.$$

The outer products are:

- For a:

$$\tilde{g}(a)\tilde{g}(a)^\top = \begin{bmatrix} 2.7^2 & 2.7(-1.5) & 2.7(-1.2) \\ 2.7(-1.5) & (-1.5)^2 & (-1.5)(-1.2) \\ 2.7(-1.2) & (-1.5)(-1.2) & (-1.2)^2 \end{bmatrix} = \begin{bmatrix} 7.29 & -4.05 & -3.24 \\ -4.05 & 2.25 & 1.8 \\ -3.24 & 1.8 & 1.44 \end{bmatrix}.$$

After weighting by 0.1:

$$0.1\, \tilde{g}(a)\tilde{g}(a)^\top = \begin{bmatrix} 0.729 & -0.405 & -0.324 \\ -0.405 & 0.225 & 0.18 \\ -0.324 & 0.18 & 0.144 \end{bmatrix}.$$

- For b:

$$\tilde{g}(b)\tilde{g}(b)^\top = \begin{bmatrix} (-0.1)^2 & (-0.1)(0.5) & (-0.1)(-0.4) \\ (-0.1)(0.5) & 0.5^2 & 0.5(-0.4) \\ (-0.1)(-0.4) & 0.5(-0.4) & (-0.4)^2 \end{bmatrix} = \begin{bmatrix} 0.01 & -0.05 & 0.04 \\ -0.05 & 0.25 & -0.2 \\ 0.04 & -0.2 & 0.16 \end{bmatrix}.$$

After weighting by 0.5:

$$0.5\, \tilde{g}(b)\tilde{g}(b)^\top = \begin{bmatrix} 0.005 & -0.025 & 0.02 \\ -0.025 & 0.125 & -0.1 \\ 0.02 & -0.1 & 0.08 \end{bmatrix}.$$

- For c:

$$\tilde{g}(c)\tilde{g}(c)^\top = \begin{bmatrix} 0.2^2 & 0.2 \cdot 1.0 & 0.2 \cdot (-1.2) \\ 1.0 \cdot 0.2 & 1.0^2 & 1.0 \cdot (-1.2) \\ (-1.2)(0.2) & (-1.2)(1.0) & (-1.2)^2 \end{bmatrix} = \begin{bmatrix} 0.04 & 0.2 & -0.24 \\ 0.2 & 1.0 & -1.2 \\ -0.24 & -1.2 & 1.44 \end{bmatrix}.$$

After weighting by 0.4:

$$0.4\, \tilde{g}(c)\tilde{g}(c)^\top = \begin{bmatrix} 0.016 & 0.08 & -0.096 \\ 0.08 & 0.4 & -0.48 \\ -0.096 & -0.48 & 0.576 \end{bmatrix}.$$

Summing these contributions gives:

$$E[\tilde{\nabla} V\, \tilde{\nabla} V^\top] \approx \begin{bmatrix} 0.729 + 0.005 + 0.016 & -0.405 - 0.025 + 0.08 & -0.324 + 0.02 - 0.096 \\ -0.405 - 0.025 + 0.08 & 0.225 + 0.125 + 0.4 & 0.18 - 0.1 - 0.48 \\ -0.324 + 0.02 - 0.096 & 0.18 - 0.1 - 0.48 & 0.144 + 0.08 + 0.576 \end{bmatrix}.$$

That is,

$$E[\tilde{\nabla} V\, \tilde{\nabla} V^\top] \approx \begin{bmatrix} 0.75 & -0.35 & -0.4 \\ -0.35 & 0.75 & -0.4 \\ -0.4 & -0.4 & 0.8 \end{bmatrix}.$$

Subtracting the same outer product of the mean vector $E[\tilde{\nabla} V] = [0.3,\, 0.5,\, -0.8]$ (which is unchanged) yields:

$$\mathrm{Cov}(\tilde{\nabla} V) = \begin{bmatrix} 0.75 - 0.09 & -0.35 - 0.15 & -0.4 + 0.24 \\ -0.35 - 0.15 & 0.75 - 0.25 & -0.4 + 0.4 \\ -0.4 + 0.24 & -0.4 + 0.4 & 0.8 - 0.64 \end{bmatrix} = \begin{bmatrix} 0.66 & -0.5 & -0.16 \\ -0.5 & 0.5 & 0 \\ -0.16 & 0 & 0.16 \end{bmatrix}.$$

Thus, with the baseline $V^{\pi_\theta}(s) = 97$, the gradient estimator remains unbiased (same mean), but its covariance matrix is now

$$\mathrm{Cov}(\tilde{\nabla} V) \approx \begin{bmatrix} 0.66 & -0.5 & -0.16 \\ -0.5 & 0.5 & 0 \\ -0.16 & 0 & 0.16 \end{bmatrix},$$

which is much “smaller” than that without the baseline.

Problem 3 (c)

Suppose we use a baseline $B(s)$ so that the gradient estimator becomes

$$\nabla V_B = \nabla \log \pi_\theta(a \mid s) \Big(r(s, a) - B(s) \Big).$$

The estimator remains unbiased for any $B(s)$ that does not depend on the action. It is known that the optimal baseline in terms of minimizing the trace of the covariance is the one that minimizes the second moment of the “advantage” $r(s, a) - B(s)$. That is, one should choose

$$B^*(s) = E_{a \sim \pi_\theta}[r(s, a)].$$

In our example, we have already computed:

$$V^{\pi_\theta}(s) = 0.1 \cdot 100 + 0.5 \cdot 98 + 0.4 \cdot 95 = 97.$$

Thus, the optimal baseline is

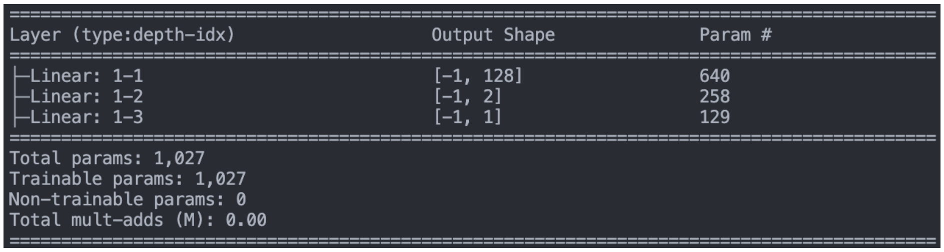
$$B^*(s) = 97.$$

This choice minimizes the variance of the gradient estimator among all state–dependent baselines.

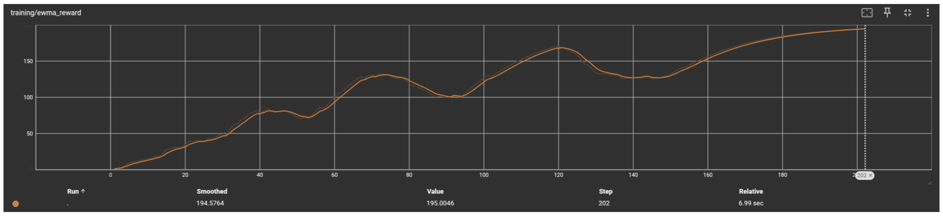
Problem 4

(a) **CartPole-v0 (vanilla)**

- Hyperparameters:
  - learning rate = 0.01
  - hidden layer size = 128
- NN Architecture

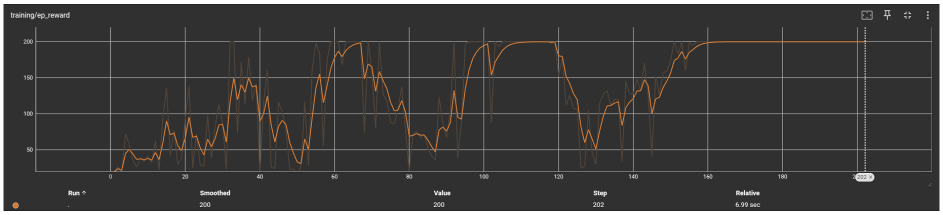


```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
├─Linear: 1-1                            [-1, 128]                 640
├─Linear: 1-2                            [-1, 2]                   258
├─Linear: 1-3                            [-1, 1]                   129
==========================================================================================
Total params: 1,027
Trainable params: 1,027
Non-trainable params: 0
Total mult-adds (M): 0.00
==========================================================================================
```



Run ↑	Smoothed	Value	Step	Relative
.	194.5764	195.0046	202	6.99 sec

Figure 1: ewma reward

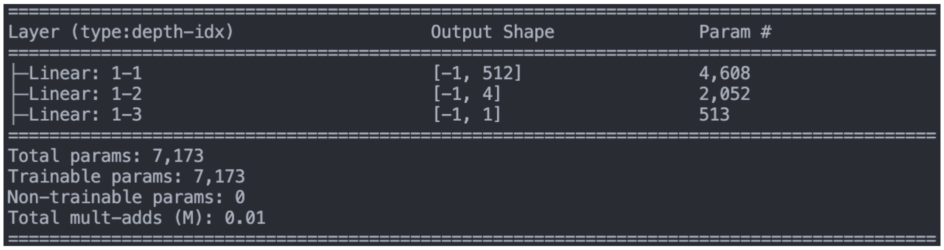


Run ↑	Smoothed	Value	Step	Relative
.	200	200	202	6.99 sec

Figure 2: ep reward

(b) **LunarLander-v2 (Baseline)**

- **Hyperparameters**:
  - learning rate = 0.01
  - hidden layer size = 512
  - reward threshold = 200 (same as the environment-provided reward threshold)
- **NN Architecture**:



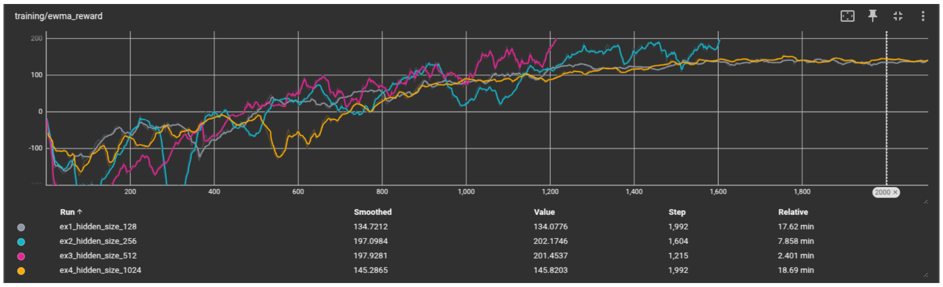
```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
├─Linear: 1-1                            [-1, 512]                 4,608
├─Linear: 1-2                            [-1, 4]                   2,052
├─Linear: 1-3                            [-1, 1]                   513
==========================================================================================
Total params: 7,173
Trainable params: 7,173
Non-trainable params: 0
Total mult-adds (M): 0.01
==========================================================================================
```

- **Baseline**: The baseline used here is the value function $V(s)$.

```
for (log_prob, value), R in zip(saved_actions, returns):
    advantage = R - value.item() # value function as the baseline
    policy_losses.append(-log_prob * advantage)
    value_losses.append(F.smooth_l1_loss(value, torch.tensor([R])))
```

- **Summary**: As shown in Figure 3, both extreme conditions—using an excessively large hidden layer size (1024) or a significantly limited one (128)—prevented the model from reaching the reward threshold, even after 3,000 episodes. In an additional experiment, the model with 128 hidden remained stagnant at around 150 average rewards, even after 60,000 episodes.

  Compared to hidden layer sizes of 256 and 512, 256 configuration took approximately three times longer to achieve the threshold performance. Therefore, the optimal hidden layer size for training LunarLander-v2 using the REINFORCE algorithm is approximately 512.

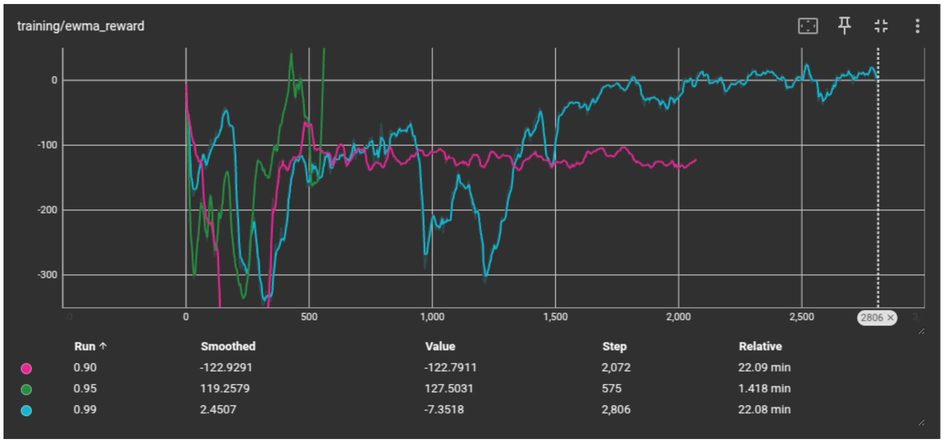


	Run ↑	Smoothed	Value	Step	Relative
●	ex1_hidden_size_128	134.7212	134.0776	1,992	17.62 min
●	ex2_hidden_size_256	197.0984	202.1746	1,604	7.858 min
●	ex3_hidden_size_512	197.9281	201.4537	1,215	2.401 min
●	ex4_hidden_size_1024	145.2865	145.8203	1,992	18.69 min

Figure 3: Experiments of LunarLander-v2 in different hidden layer sizes.

(c) **LunarLander-v2 (GAE)**

- Different values of λ

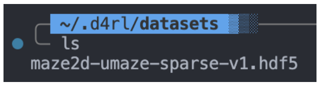


	Run ↑	Smoothed	Value	Step	Relative
●	0.90	-122.9291	-122.7911	2,072	22.09 min
●	0.95	119.2579	127.5031	575	1.418 min
●	0.99	2.4507	-7.3518	2,806	22.08 min

Since $\lambda = 0.95$ performed effectively, I explored whether adjusting this value slightly higher or lower would further improve performance. Specifically, I tested $\lambda = 0.90$ and $\lambda = 0.99$. However, both choices resulted in significantly worse outcomes, failing to reach even the threshold of 120 within 2000 episodes. This suggests that $\lambda = 0.95$ is near optimal for this scenario, and deviations in either direction negatively impact performance.

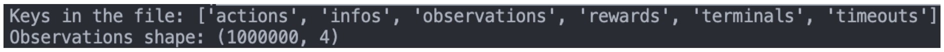
Problem 5

1. **Generated offline dataset**:



```
  ╭ ~/.d4rl/datasets
● ╰ ls
  maze2d-umaze-sparse-v1.hdf5
```

2. **Format**:

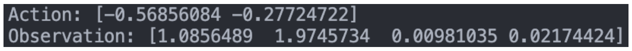


```
Keys in the file: ['actions', 'infos', 'observations', 'rewards', 'terminals', 'timeouts']
Observations shape: (1000000, 4)
```

The offline dataset contains a large collection of state-action pairs and corresponding rewards, originally collected by interacting with an environment. It serves as a **sampled** representation of the online environment, enabling training without further interaction.

The ”terminal” flag indicates whether a state corresponds to an episode ending naturally (similar to the ‘done‘ flag in online interaction), whereas the ”timeout” flag indicates whether an episode ended due to reaching the environment’s maximum episode length.

Shape:



```
Action: [-0.56856084 -0.27724722]
Observation: [1.0856489  1.9745734  0.00981035 0.02174424]
```

Figure 4: example of one piece of data

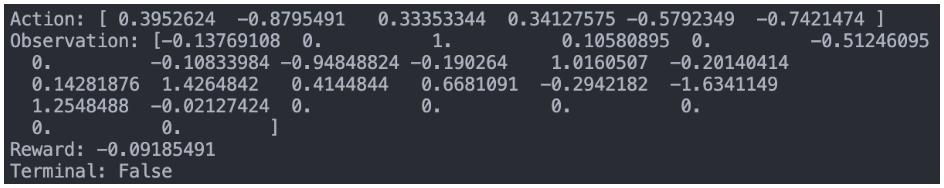
- Observations shape: (1000000, 4)
- Actions shape: (1000000, 2)

3. **Additional offline dataset**: (bullet-halfcheetah-random-v0)

   The format is quit the same, the only difference is the shape of action an observation shape.

   Shape:

   - Observations shape: (1000000, 26)
   - Actions shape: (1000000, 6)



```
Action: [ 0.3952624  -0.8795491   0.33353344  0.34127575 -0.5792349  -0.7421474 ]
Observation: [-0.13769108  0.          1.          0.10580895  0.          -0.51246095
  0.         -0.10833984 -0.94848824 -0.190264    1.0160507  -0.20140414
  0.14281876  1.4264842   0.4144844   0.6681091  -0.2942182  -1.6341149
  1.2548488  -0.02127424  0.          0.          0.          0.
  0.          0.        ]
Reward: -0.09185491
Terminal: False
```