111550177_HW2

1.

Problem 1 (Baseline for Variance Reduction) (8+8+8=24 points)

Consider an example similar to that in the slides of Lecture 8 for explaining the baseline. Suppose there are only 1 non-terminal starting state (denoted by s) and 3 actions (denoted by a, b, c) in the MDP of interest. After any one of the action is applied at the starting state s, the MDP would evolve from s to the terminal state, with probability 1. Moreover, consider the following setting:

- The rewards are deterministic, and the reward function is defined as $r(s,a) = 100$, $r(s,b) = 98$, and $r(s,c) = 95$. Moreover, there is no terminal reward.
- We consider a softmax policy with parameters $\theta_a, \theta_b, \theta_c$ such that $\pi_\theta(\cdot|s) = \exp(\theta_\cdot)/(\exp(\theta_a) + \exp(\theta_b) + \exp(\theta_c))$. Moreover, currently the parameters are $\theta_a = 0$, $\theta_b = \ln 5$, $\theta_c = \ln 4$.
- We would like to combine PG with SGD. At each policy update, we would construct an unbiased estimate $\widehat{\nabla} V$ of the true policy gradient $\nabla_\theta V^{\pi_\theta}$ by sampling one trajectory (Note: $\widehat{\nabla} V$ is a random vector. In this example, each trajectory has only one time step, and $s_0 = s$, a_0 is either a, b, or c, and s_1 is the terminal state).

(a) What are the mean vector of $\widehat{\nabla} V$ (denoted by $\mathbb{E}[\widehat{\nabla} V]$) and the covariance matrix of $\widehat{\nabla} V$ (i.e., $\mathbb{E}[(\widehat{\nabla} V - \mathbb{E}[\widehat{\nabla} V])(\widehat{\nabla} V - \mathbb{E}[\widehat{\nabla} V])^\intercal]$)?

(a)

$$\pi_\theta(a|s) = \frac{e^{\theta_a}}{e^{\theta_a} + e^{\theta_b} + e^{\theta_c}} = \frac{e^0}{e^0 + e^{(\ln 5)} + e^{(\ln 4)}} = \frac{1}{10}$$

$$\pi_\theta(a|s) = \frac{e^{\theta_b}}{e^{\theta_a} + e^{\theta_b} + e^{\theta_c}} = \frac{e^{(\ln 5)}}{e^0 + e^{(\ln 5)} + e^{(\ln 4)}} = \frac{5}{10}$$

$$\pi_\theta(a|s) = \frac{e^{\theta_c}}{e^{\theta_a} + e^{\theta_b} + e^{\theta_c}} = \frac{e^{(\ln 4)}}{e^0 + e^{(\ln 5)} + e^{(\ln 4)}} = \frac{4}{10}$$

$$\nabla \log \pi_\theta(a|s) = \begin{bmatrix} \frac{\partial \log \pi_\theta(a|s)}{\partial \theta_a} \\ \frac{\partial \log \pi_\theta(a|s)}{\partial \theta_b} \\ \frac{\partial \log \pi_\theta(a|s)}{\partial \theta_c} \end{bmatrix} = \begin{bmatrix} 1 - \pi_\theta(a|s) \\ -\pi_\theta(b|s) \\ -\pi_\theta(c|s) \end{bmatrix} = \begin{bmatrix} 1 - 0.1 \\ -0.5 \\ -0.4 \end{bmatrix} = \begin{bmatrix} 0.9 \\ -0.5 \\ -0.4 \end{bmatrix}$$

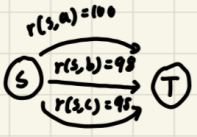
$$\nabla \log \pi_\theta(a|s) = \begin{bmatrix} \frac{\partial \log \pi_\theta(b|s)}{\partial \theta_a} \\ \frac{\partial \log \pi_\theta(b|s)}{\partial \theta_b} \\ \frac{\partial \log \pi_\theta(b|s)}{\partial \theta_c} \end{bmatrix} = \begin{bmatrix} -\pi_\theta(a|s) \\ 1 - \pi_\theta(b|s) \\ -\pi_\theta(c|s) \end{bmatrix} = \begin{bmatrix} -0.1 \\ 1 - 0.5 \\ -0.4 \end{bmatrix} = \begin{bmatrix} -0.1 \\ 0.5 \\ -0.4 \end{bmatrix}$$

$$\nabla \log \pi_\theta(a|s) = \begin{bmatrix} \frac{\partial \log \pi_\theta(c|s)}{\partial \theta_a} \\ \frac{\partial \log \pi_\theta(c|s)}{\partial \theta_b} \\ \frac{\partial \log \pi_\theta(c|s)}{\partial \theta_c} \end{bmatrix} = \begin{bmatrix} -\pi_\theta(a|s) \\ -\pi_\theta(b|s) \\ 1-\pi_\theta(c|s) \end{bmatrix} = \begin{bmatrix} -0.1 \\ -0.5 \\ 1-0.4 \end{bmatrix} = \begin{bmatrix} -0.1 \\ -0.5 \\ 0.6 \end{bmatrix}$$

$$E[\hat{\nabla} V] = \nabla_\theta V^{\pi_\theta}$$

$$= E_{\tau \sim p_u^{\pi_\theta}}\left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_\theta}(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t|s_t)\right]$$

$$= \sum_{a_0 \in \{a,b,c\}} r(s, a_0) \cdot \nabla_\theta \log \pi_\theta(a_0|s) \cdot \pi_\theta(a_0|s)$$



$$= 0.1 \times 100 \times \begin{bmatrix} 0.9 \\ -0.5 \\ -0.4 \end{bmatrix} + 0.5 \times 98 \times \begin{bmatrix} -0.1 \\ 0.5 \\ -0.4 \end{bmatrix} + 0.4 \times 95 \times \begin{bmatrix} -0.1 \\ -0.5 \\ 0.6 \end{bmatrix}$$

$$= \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}$$

$$E[(\hat{\nabla} - E[\hat{\nabla}]) \cdot (\hat{\nabla} - E[\hat{\nabla}])^T]$$

$$= E_{\tau \sim p_u^{\pi_\theta}}[(\hat{\nabla} - E[\hat{\nabla}]) \cdot (\hat{\nabla} - E[\hat{\nabla}])^T]$$

$$= \sum_{a_0 \in \{a,b,c\}} (\hat{\nabla}_{a_0} - E[\hat{\nabla}])(\hat{\nabla}_{a_0} - E[\hat{\nabla}])^T \cdot \pi(a_0|s)$$

$$= \left(100 \times \begin{bmatrix} 0.9 \\ -0.5 \\ -0.4 \end{bmatrix} - \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}\right) \cdot \left(100 \times \begin{bmatrix} 0.9 \\ -0.5 \\ -0.4 \end{bmatrix} - \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}\right)^T \times 0.1$$

$$+\left(98\times\begin{bmatrix}-0.1\\0.5\\-0.4\end{bmatrix}-\begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)\cdot\left(98\times\begin{bmatrix}-0.1\\0.5\\-0.4\end{bmatrix}-\begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)^T\times 0.5$$

$$+\left(95\times\begin{bmatrix}-0.1\\-0.5\\0.6\end{bmatrix}-\begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)\cdot\left(95\times\begin{bmatrix}-0.1\\-0.5\\0.6\end{bmatrix}-\begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)^T\times 0.4$$

$$=\begin{bmatrix}894.03 & -509.75 & -384.28\\ -509.75 & 2352.75 & -1834\\ -384.28 & -1834 & 2227.28\end{bmatrix}$$

(b) Suppose we leverage the value function $V^{\pi_\theta}(s)$ as the baseline and denote by $\tilde{\nabla}V$ the corresponding estimated policy gradient. Then, what are the mean vector and the covariance matrix of $\tilde{\nabla}V$? (Note: $\tilde{\nabla}V$ is also a random vector)

$$V^{\pi_\theta}(S)=0.1\times 100+0.5\times 98+0.4\times 95=97$$

$$E[\tilde{\nabla}V]=\nabla_\theta V^{\pi_\theta}$$

$$=E_{\tau\sim p_a^{\pi_\theta}}\left[\sum_{t=0}^{\infty}\gamma^t\left(Q^{\pi_\theta}(s_t,a_t)-B(s_t)\right)\nabla_\theta\log\pi_\theta(a_t|s_t)\right]$$

$$=\sum_{a_0\in\{a,b,c\}}\left(r(s,a_0)-V^{\pi_\theta}\right)\cdot\nabla_\theta\log\pi_\theta(a_0|s)\cdot\pi_\theta(a_0|s)$$

$$=0.1\times(100-97)\times\begin{bmatrix}0.9\\-0.5\\-0.4\end{bmatrix}+0.5\times(98-97)\times\begin{bmatrix}-0.1\\0.5\\-0.4\end{bmatrix}+0.4\times(95-97)\times\begin{bmatrix}-0.1\\-0.5\\0.6\end{bmatrix}$$

$$= \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}$$

$$E[(\tilde{v} - E[\tilde{v}]) \cdot (\tilde{v} - E[\tilde{v}])^T]$$

$$= E_{\tau \sim P_u^{\pi_\theta}}[(\tilde{v} - E[\tilde{v}]) \cdot (\tilde{v} - E[\tilde{v}])^T]$$

$$= \sum_{a_0 \in \{a,b,c\}} (\tilde{v}_{a_0} - E[\tilde{v}])(\tilde{v}_{a_0} - E[\tilde{v}])^T \cdot \pi(a_0 | s)$$

$$= \left((100-97) \times \begin{bmatrix} 0.9 \\ -0.5 \\ -0.4 \end{bmatrix} - \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}\right) \cdot \left((100-97) \times \begin{bmatrix} 0.9 \\ -0.5 \\ -0.4 \end{bmatrix} - \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}\right)^T \times 0.1$$

$$+ \left((98-97) \times \begin{bmatrix} -0.1 \\ 0.5 \\ -0.4 \end{bmatrix} - \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}\right) \cdot \left((98-97) \times \begin{bmatrix} -0.1 \\ 0.5 \\ -0.4 \end{bmatrix} - \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}\right)^T \times 0.5$$

$$+ \left((95-97) \times \begin{bmatrix} -0.1 \\ -0.5 \\ 0.6 \end{bmatrix} - \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}\right) \cdot \left((95-97) \times \begin{bmatrix} -0.1 \\ -0.5 \\ 0.6 \end{bmatrix} - \begin{bmatrix} 0.3 \\ 0.5 \\ -0.8 \end{bmatrix}\right)^T \times 0.4$$

$$= \begin{bmatrix} 0.66 & -0.5 & -0.16 \\ -0.5 & 0.5 & 0 \\ -0.16 & 0 & 0.16 \end{bmatrix}$$

(c) Let $B(s)$ denote a baseline function and ∇V_B be the corresponding estimated policy gradient (∇V_B is again a random vector). Suppose we say that a baseline function $B(s)$ is *optimal* if it attains the minimum trace of the corresponding covariance matrix of ∇V_B among all possible state-dependent baselines. Please try to find one such optimal $B(s)$.

The covarience matrix with baseline is

$$E[\tilde{\nabla}] = 0.1\times(100-B)\times\begin{bmatrix}0.9\\-0.5\\-0.4\end{bmatrix} + 0.5\times(98-B)\times\begin{bmatrix}-0.1\\0.5\\-0.4\end{bmatrix} + 0.4\times(95-B)\times\begin{bmatrix}-0.1\\-0.5\\0.6\end{bmatrix} = \begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}$$

$$E[(\tilde{\nabla}-E[\tilde{\nabla}])\cdot(\tilde{\nabla}-E[\tilde{\nabla}])^T] \qquad \begin{bmatrix}a\\b\\c\end{bmatrix}\begin{bmatrix}a&b&c\end{bmatrix} = \begin{bmatrix}a^2&ab&ac\\ba&b^2&bc\\ca&cb&c^2\end{bmatrix}$$

$$= \left((100-B)\times\begin{bmatrix}0.9\\-0.5\\-0.4\end{bmatrix} - \begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)\cdot\left((100-B)\times\begin{bmatrix}0.9\\-0.5\\-0.4\end{bmatrix} - \begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)^T\times 0.1$$

$$+ \left((98-B)\times\begin{bmatrix}-0.1\\0.5\\-0.4\end{bmatrix} - \begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)\cdot\left((98-B)\times\begin{bmatrix}-0.1\\0.5\\-0.4\end{bmatrix} - \begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)^T\times 0.5$$

$$+ \left((95-B)\times\begin{bmatrix}-0.1\\-0.5\\0.6\end{bmatrix} - \begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)\cdot\left((95-B)\times\begin{bmatrix}-0.1\\-0.5\\0.6\end{bmatrix} - \begin{bmatrix}0.3\\0.5\\-0.8\end{bmatrix}\right)^T\times 0.4$$

The trace of covariance matrix :

$$t(B(s)) = 0.1\times\left\{[(100-B(s))\times 0.9-0.3]^2+[(100-B(s))\times(-0.5)-0.5]^2+[(100-B(s)\times(-0.4)+0.8]^2\right\}$$

$$+ 0.5 \times \left\{ [(98 - B(s)) \times (-0.1) - 0.3]^2 + [(98 - B(s)) \times 0.5 - 0.5]^2 + [(98 - B(s)) \times (-0.4) + 0.8]^2 \right\}$$

$$+ 0.4 \left\{ [(95 - B(s)) \times (-0.1) - 0.3]^2 + [(95 - B(s) \times (-0.5) - 0.5]^2 + [(95 - B(s)) \times 0.6 + 0.8]^2 \right\}$$

$$\frac{df(B(s))}{dB(s)} = 0 \Rightarrow 0.2 \times (121.66 - 1.22 B(s)) + 1 \times (40.62 - 0.42 \cdot B(s)) + 0.8 \times (59.66 - 0.62 B(s)) = 0$$

$$\Rightarrow B(s) = 97.137931 \text{ is the optimal Baseline}$$

Problem 2 (Non-Uniform Polyak-Lojacsiewicz Condition in RL) (8+8=16 points)

As described in Lecture 12, let us prove the fundamental Polyak-Lojacsiewicz condition in RL: Let π^* be an optimal policy and let $a^* := \arg\max_a \pi^*(a|s)$ (essentially, a^* is an optimal action). Under softmax policies, we have

$$\left\| \frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta} \right\|_2 \geq \frac{1}{\sqrt{S}} \left\| \frac{d_\mu^{\pi^*}}{d_\mu^{\pi_\theta}} \right\|_\infty^{-1} \cdot \min_{s \in S} \pi_\theta(a^*(s)|s) \cdot [V^*(\mu) - V^{\pi_\theta}(\mu)]. \tag{1}$$

To show this, you would also need the celebrated "Performance difference lemma" as follows: For any two policies π_1 and π_2, we always have

$$V^{\pi_2}(\mu) - V^{\pi_1}(\mu) = \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d_\mu^{\pi_2}} \mathbb{E}_{a' \sim \pi_2(\cdot|s')} \left[A^{\pi_1}(s', a') \right]. \tag{2}$$

(a) To begin with, show the following result:

$$\left\| \frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta} \right\|_2 \geq \frac{1}{1-\gamma} \cdot \frac{1}{\sqrt{S}} \sum_s d_\mu^{\pi_\theta}(s) \cdot \pi_\theta(a^*(s)|s) \cdot \left| A^{\pi_\theta}(s, a^*(s)) \right|. \tag{3}$$

(Hint: You would need to first apply Cauchy-Schwarz inequality and leverage the Policy Gradient expressio under softmax policies. This subproblem shall require about 5-8 lines of proof.)

$$\left\| \frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta} \right\|_2 \geq \left[\sum_{s \in S, a \in A} \left(\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta(s,a)} \right)^2 \right]^{\frac{1}{2}}$$

$$\geq \left[\sum_{s \in S} \left(\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta(s, a^*)} \right)^2 \right]^{\frac{1}{2}}$$

$$\geq \frac{1}{\sqrt{S}} \sum_{s \in S} \left| \frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta(s, a^*)} \right|$$

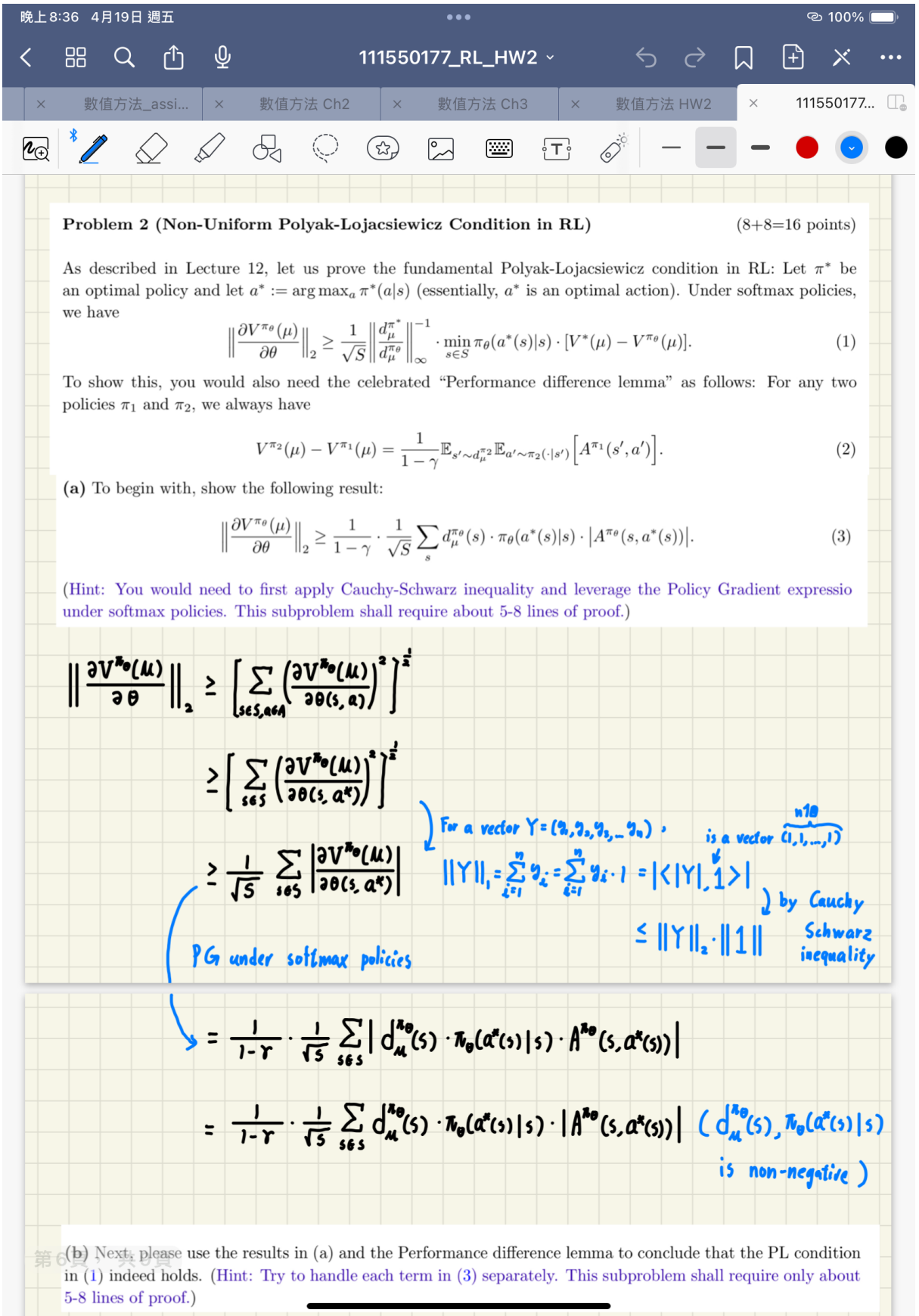
For a vector $Y = (y_1, y_2, y_3, \ldots, y_n)$, is a vector $\overbrace{(1,1,\ldots,1)}^{n\text{個}}$

$$\|Y\|_1 = \sum_{i=1}^n y_i = \sum_{i=1}^n y_i \cdot 1 = |\langle |Y|, \mathbf{1} \rangle|$$

$$\leq \|Y\|_2 \cdot \|\mathbf{1}\|$$ by Cauchy Schwarz inequality

PG under softmax policies

2.



Problem 2 (Non-Uniform Polyak-Lojacsiewicz Condition in RL) (8+8=16 points)

As described in Lecture 12, let us prove the fundamental Polyak-Lojacsiewicz condition in RL: Let π^* be an optimal policy and let $a^* := \arg\max_a \pi^*(a|s)$ (essentially, a^* is an optimal action). Under softmax policies, we have

$$\left\|\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta}\right\|_2 \geq \frac{1}{\sqrt{S}}\left\|\frac{d_\mu^{\pi^*}}{d_\mu^{\pi_\theta}}\right\|_\infty^{-1} \cdot \min_{s\in S} \pi_\theta(a^*(s)|s) \cdot [V^*(\mu) - V^{\pi_\theta}(\mu)]. \quad (1)$$

To show this, you would also need the celebrated "Performance difference lemma" as follows: For any two policies π_1 and π_2, we always have

$$V^{\pi_2}(\mu) - V^{\pi_1}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_\mu^{\pi_2}}\mathbb{E}_{a'\sim\pi_2(\cdot|s')}\left[A^{\pi_1}(s',a')\right]. \quad (2)$$

(a) To begin with, show the following result:

$$\left\|\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta}\right\|_2 \geq \frac{1}{1-\gamma}\cdot\frac{1}{\sqrt{S}}\sum_s d_\mu^{\pi_\theta}(s)\cdot\pi_\theta(a^*(s)|s)\cdot\left|A^{\pi_\theta}(s,a^*(s))\right|. \quad (3)$$

(Hint: You would need to first apply Cauchy-Schwarz inequality and leverage the Policy Gradient expressio under softmax policies. This subproblem shall require about 5-8 lines of proof.)

$$\left\|\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta}\right\|_2 \geq \left[\sum_{s\in S, a\in A}\left(\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta(s,a)}\right)^2\right]^{\frac{1}{2}}$$

$$\geq \left[\sum_{s\in S}\left(\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta(s,a^*)}\right)^2\right]^{\frac{1}{2}}$$

$$\geq \frac{1}{\sqrt{S}}\sum_{s\in S}\left|\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta(s,a^*)}\right|$$

For a vector $Y=(y_1,y_2,y_3,\dots y_n)$, $\|Y\|_1 = \sum_{i=1}^n y_i = \sum_{i=1}^n y_i\cdot 1 = |\langle |Y|, \underbrace{1}_{\text{n個}} \rangle|$ (1 is a vector $(1,1,\dots,1)$)

$\leq \|Y\|_2 \cdot \|1\|$ (by Cauchy Schwarz inequality)

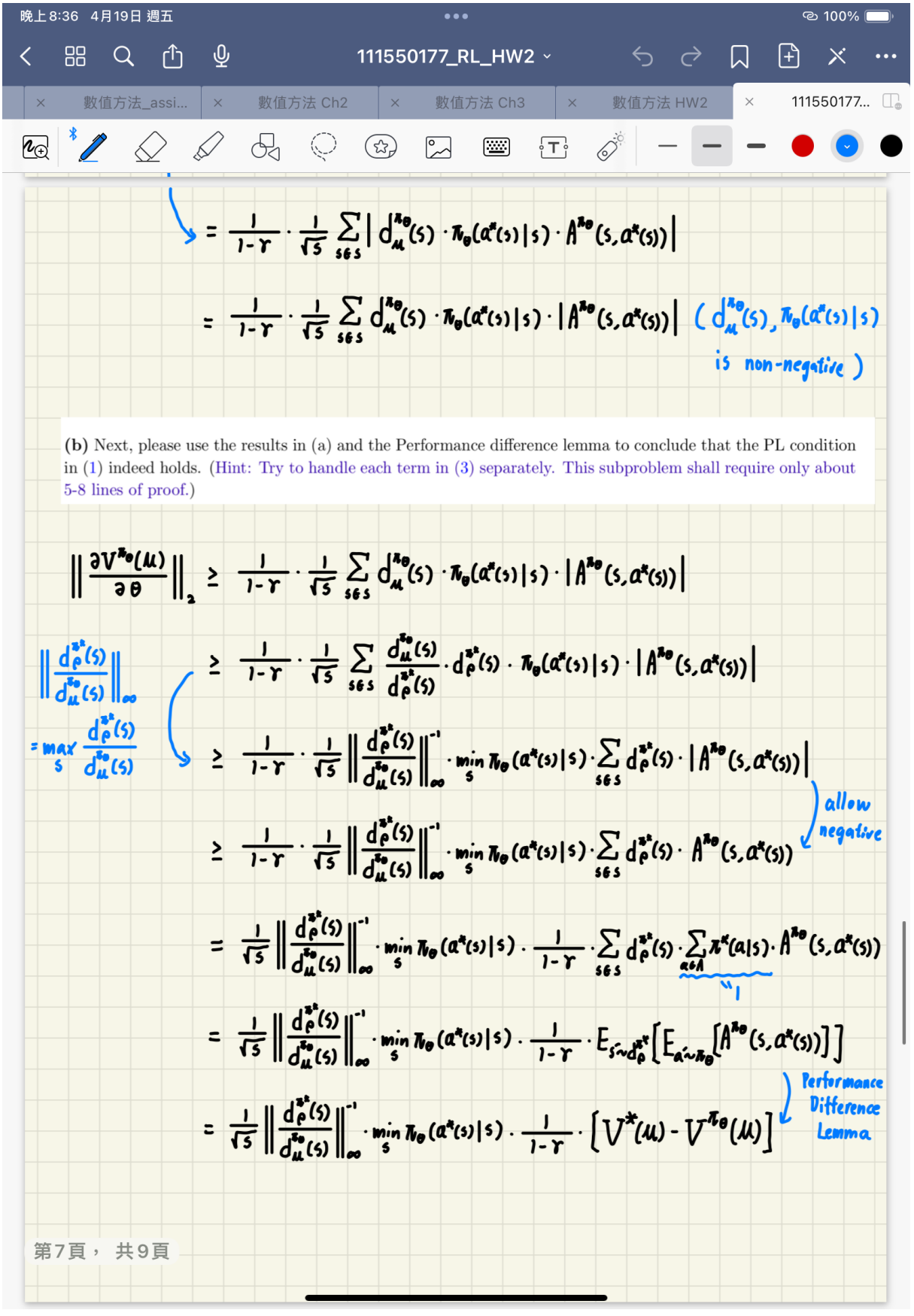
PG under softmax policies

$$= \frac{1}{1-\gamma}\cdot\frac{1}{\sqrt{S}}\sum_{s\in S}\left|d_\mu^{\pi_\theta}(s)\cdot\pi_\theta(a^*(s)|s)\cdot A^{\pi_\theta}(s,a^*(s))\right|$$

$$= \frac{1}{1-\gamma}\cdot\frac{1}{\sqrt{S}}\sum_{s\in S} d_\mu^{\pi_\theta}(s)\cdot\pi_\theta(a^*(s)|s)\cdot\left|A^{\pi_\theta}(s,a^*(s))\right|$$

($d_\mu^{\pi_\theta}(s)$, $\pi_\theta(a^*(s)|s)$ is non-negative)

(b) Next, please use the results in (a) and the Performance difference lemma to conclude that the PL condition in (1) indeed holds. (Hint: Try to handle each term in (3) separately. This subproblem shall require only about 5-8 lines of proof.)


晚上 8:36 4月19日 週五
100%
111550177_RL_HW2
數值方法_assi...
數值方法 Ch2
數值方法 Ch3
數值方法 HW2
111550177...
$= \frac{1}{1-\gamma} \cdot \frac{1}{\sqrt{S}} \sum_{s \in \mathcal{S}} \left| d_\mu^{\pi_\theta}(s) \cdot \pi_\theta(a^*(s)|s) \cdot A^{\pi_\theta}(s, a^*(s)) \right|$
$= \frac{1}{1-\gamma} \cdot \frac{1}{\sqrt{S}} \sum_{s \in \mathcal{S}} d_\mu^{\pi_\theta}(s) \cdot \pi_\theta(a^*(s)|s) \cdot \left| A^{\pi_\theta}(s, a^*(s)) \right|$ ($d_\mu^{\pi_\theta}(s), \pi_\theta(a^*(s)|s)$ is non-negative)
(b) Next, please use the results in (a) and the Performance difference lemma to conclude that the PL condition in (1) indeed holds. (Hint: Try to handle each term in (3) separately. This subproblem shall require only about 5-8 lines of proof.)
$\left\| \frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta} \right\|_2 \geq \frac{1}{1-\gamma} \cdot \frac{1}{\sqrt{S}} \sum_{s \in \mathcal{S}} d_\mu^{\pi_\theta}(s) \cdot \pi_\theta(a^*(s)|s) \cdot \left| A^{\pi_\theta}(s, a^*(s)) \right|$
$\left\| \frac{d_\rho^{\pi^*}(s)}{d_\mu^{\pi_\theta}(s)} \right\|_\infty = \max_s \frac{d_\rho^{\pi^*}(s)}{d_\mu^{\pi_\theta}(s)}$
$\geq \frac{1}{1-\gamma} \cdot \frac{1}{\sqrt{S}} \sum_{s \in \mathcal{S}} \frac{d_\mu^{\pi_\theta}(s)}{d_\rho^{\pi^*}(s)} \cdot d_\rho^{\pi^*}(s) \cdot \pi_\theta(a^*(s)|s) \cdot \left| A^{\pi_\theta}(s, a^*(s)) \right|$
$\geq \frac{1}{1-\gamma} \cdot \frac{1}{\sqrt{S}} \left\| \frac{d_\rho^{\pi^*}(s)}{d_\mu^{\pi_\theta}(s)} \right\|_\infty^{-1} \cdot \min_s \pi_\theta(a^*(s)|s) \cdot \sum_{s \in \mathcal{S}} d_\rho^{\pi^*}(s) \cdot \left| A^{\pi_\theta}(s, a^*(s)) \right|$
$\geq \frac{1}{1-\gamma} \cdot \frac{1}{\sqrt{S}} \left\| \frac{d_\rho^{\pi^*}(s)}{d_\mu^{\pi_\theta}(s)} \right\|_\infty^{-1} \cdot \min_s \pi_\theta(a^*(s)|s) \cdot \sum_{s \in \mathcal{S}} d_\rho^{\pi^*}(s) \cdot A^{\pi_\theta}(s, a^*(s))$ allow negative
$= \frac{1}{\sqrt{S}} \left\| \frac{d_\rho^{\pi^*}(s)}{d_\mu^{\pi_\theta}(s)} \right\|_\infty^{-1} \cdot \min_s \pi_\theta(a^*(s)|s) \cdot \frac{1}{1-\gamma} \cdot \sum_{s \in \mathcal{S}} d_\rho^{\pi^*}(s) \cdot \underbrace{\sum_{a \in \mathcal{A}} \pi^*(a|s)}_{=1} \cdot A^{\pi_\theta}(s, a^*(s))$
$= \frac{1}{\sqrt{S}} \left\| \frac{d_\rho^{\pi^*}(s)}{d_\mu^{\pi_\theta}(s)} \right\|_\infty^{-1} \cdot \min_s \pi_\theta(a^*(s)|s) \cdot \frac{1}{1-\gamma} \cdot E_{s \sim d_\rho^{\pi^*}} \left[E_{a \sim \pi_\theta} \left[A^{\pi_\theta}(s, a^*(s)) \right] \right]$
$= \frac{1}{\sqrt{S}} \left\| \frac{d_\rho^{\pi^*}(s)}{d_\mu^{\pi_\theta}(s)} \right\|_\infty^{-1} \cdot \min_s \pi_\theta(a^*(s)|s) \cdot \frac{1}{1-\gamma} \cdot \left[V^*(\mu) - V^{\pi_\theta}(\mu) \right]$ Performance Difference Lemma
第7頁， 共9頁

3.

Problem 3 (Monte Carlo Policy Evaluation) (8+8=16 points)

As discussed in Lecture 10, we know that the every-visit Monte Carlo estimate is biased. Let us quickly verify this fact under the simple 2-state MRP (which serves as a reduction from any MRP), as shown below. Specifically, we need to check the following two properties:

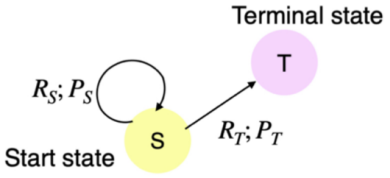


Figure 1: A simple 2-state MRP.

- Property 1: Show that the true value function at state S (denoted by $V(S)$) satisfies that

$$V(S) = \frac{P_S}{P_T} R_S + R_T. \tag{4}$$

Possible scenario :

$S \to T : P_T R_T$

$S \to S \to T : P_S P_T (R_S + R_T)$

$S \to S \to S \to T : P_S^2 P_T (2R_S + R_T)$

$\vdots$

$S \to S \to \dots \to S \to T : P_S^n P_T (n \cdot R_S + R_T)$

$$\Rightarrow V(S) = P_T R_T + P_S P_T (R_S + R_T) + P_S^2 P_T (2R_S + R_T) + \dots + P_S^n P_T (n \cdot R_S + R_T)$$

$$= \sum_{n=0}^{\infty} P_S^n \cdot P_T (n \cdot R_S + R_T)$$

$$= P_T \cdot R_S \cdot \sum_{n=0}^{\infty} n \cdot P_S^n + P_T \cdot R_T \cdot \sum_{n=0}^{\infty} P_S^n$$

$$= P_T \cdot R_S \cdot \frac{P_S}{(1-P_S)^2} + P_T \cdot R_T \cdot \frac{1}{1-P_S}$$

$\sum_{i=0}^{\infty} a^i = \frac{1}{1-a}$, $\sum_{i=0}^{\infty} i \cdot a^i = \frac{a}{(1-a)^2}$, if $|a|<1$

$(P_S + P_T = 1)$

$$= \frac{P_S}{P_T} \cdot R_S + R_T$$

- Property 2: Suppose we construct an every-visit MC estimate based on only 1 trajectory τ (denoted by $\hat{V}_{\mathrm{MC}}(S;\tau)$). Then, please show that

$$\mathbb{E}_\tau\left[\hat{V}_{\mathrm{MC}}(S;\tau)\right] = \frac{P_S}{2P_T} R_S + R_T. \tag{5}$$

(Hint: To begin with, you shall consider all possible trajectories and the corresponding probabilities. Accordingly, you would obtain that $\mathbb{E}_\tau\left[\hat{V}_{\mathrm{MC}}(S;\tau)\right] = \sum_{k=0}^{\infty} P_T P_S^k \left(\frac{R_S+2R_S+\cdots+kR_S+(k+1)R_T}{k+1}\right)$.)

Possible scenario :

$S \to T : P_T R_T$

$S \to S \to T : P_T P_S \cdot \left(\frac{R_S + 2R_T}{2}\right)$

$S \to S \to S \to T : P_T P_S^2 \cdot \left(\frac{R_S + 2R_S + 3R_T}{3}\right)$

$\vdots$

$S \to S \to \dots \to S \to T : P_T P_S^n \cdot \left(\frac{R_S + 2R_S + \dots + nR_S + (n+1)R_T}{n+1}\right)$

$$\mathbb{E}_\tau\left[\hat{V}_{\mathrm{MC}}(S;\tau)\right] = \sum_{n=0}^{\infty} P_T P_S^n \left(\frac{R_S + 2R_S + \dots + nR_S + (n+1)R_T}{n+1}\right)$$

$$= \sum_{n=0}^{\infty} P_T P_S^n \cdot \left[\frac{\frac{n(n+1)}{2}}{n+1} + R_T\right]$$

$$= \frac{P_T}{2} \sum_{n=0}^{\infty} P_S^n \cdot n + P_T R_T \sum_{n=0}^{\infty} P_S^n$$

$$= \frac{P_T}{2} \cdot \frac{P_S}{(1-P_S)^2} + P_T \cdot R_T \cdot \frac{1}{1-P_S}$$

$$= \frac{P_S}{2P_T} R_S + R_T$$

$\sum_{i=0}^{\infty} a^i = \frac{1}{1-a}$, $\sum_{i=0}^{\infty} i \cdot a^i = \frac{a}{(1-a)^2}$, if $|a|<1$

$(P_S + P_T = 1)$

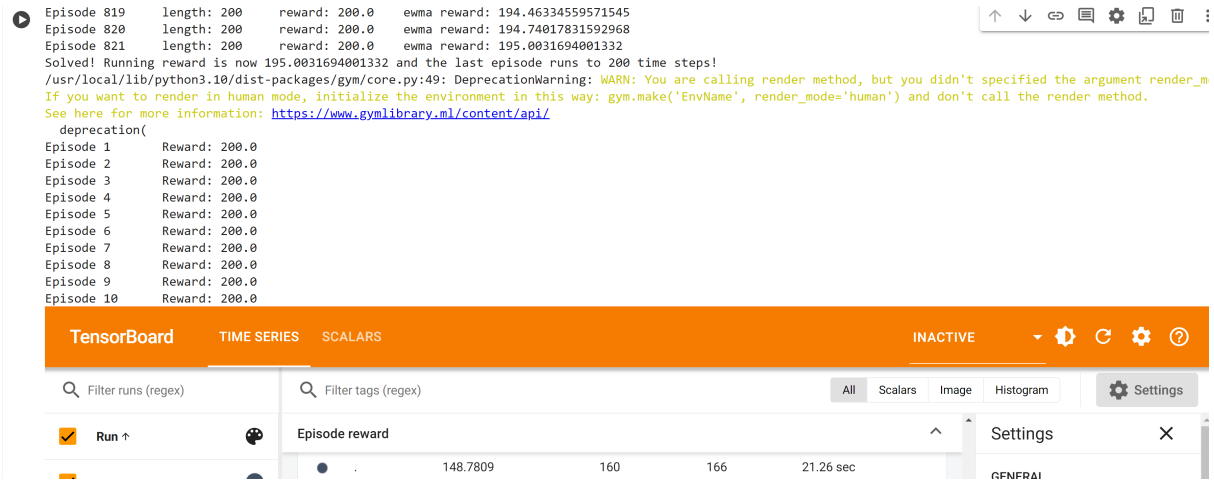
4.

(a)

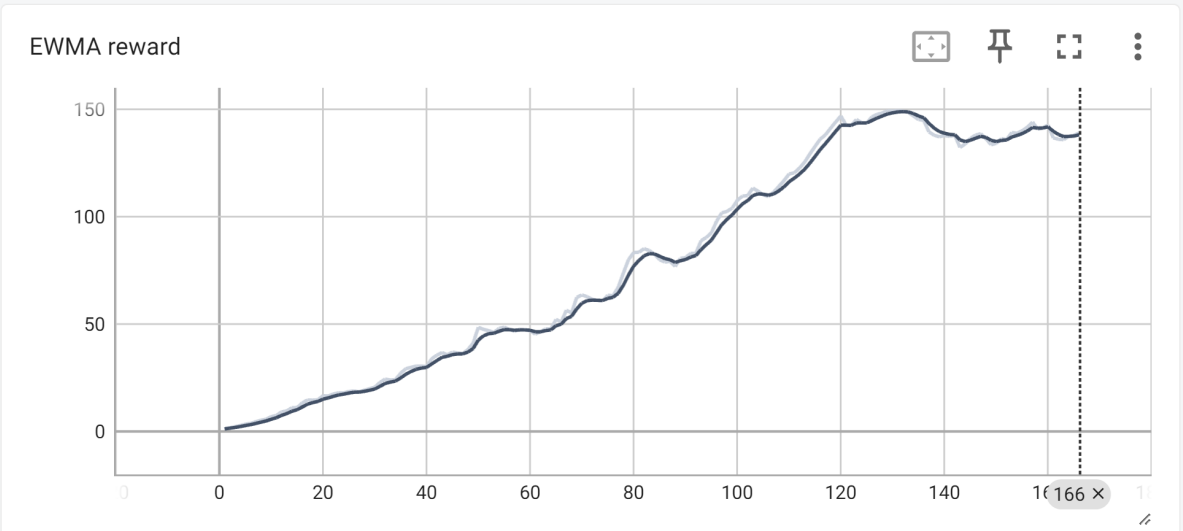
I add two shared layer in my neuro network, and the activation function are both ReLu. I also add one dropout betweeen layer one and two with dropout probability 0.2.

I found that when the gamma goes down, the episode needed to converge become more. The best number of hidden layer is 256. If the learning rate is too large, the model can’t converge.

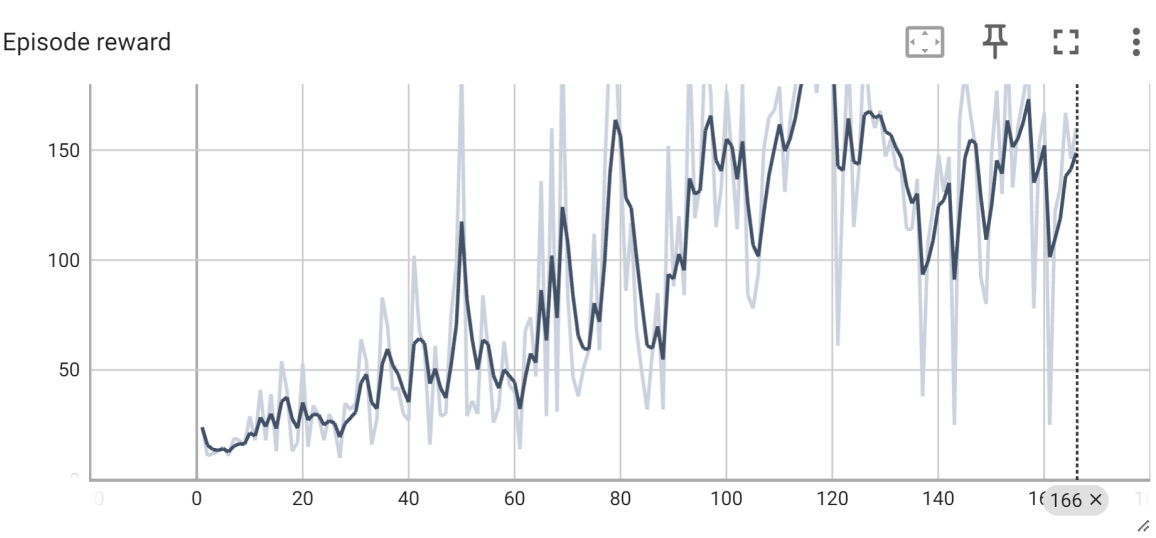
hiddden layer	gamma	learning rate	episode needed
256	0.999	0.01	821



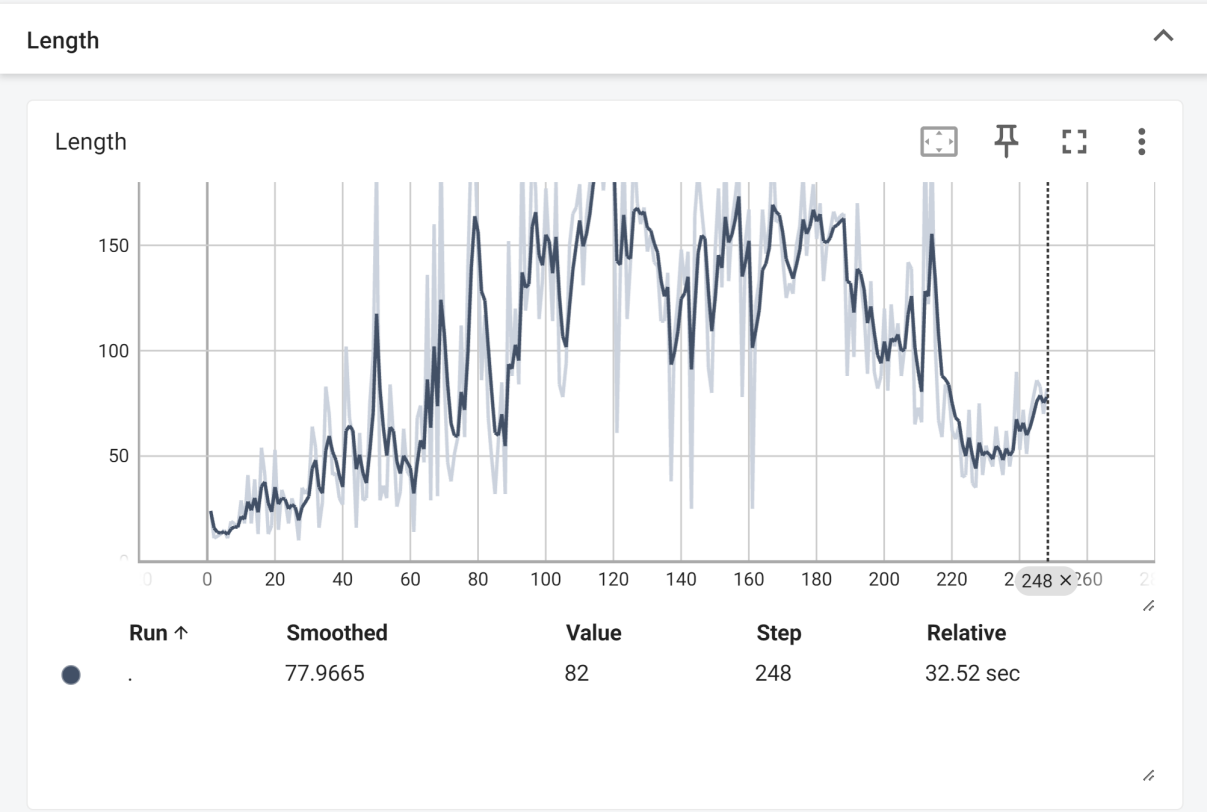
EWMA reward



Run ↑	Smoothed	Value	Step	Relative
.	138.1189	138.9607	166	21.26 sec



Run ↑	Smoothed	Value	Step	Relative
.	148.7809	160	166	21.26 sec



(b)

I add two shared layer in my neuro network, and the activation function are both ReLu. I don't use dropout here.

I found that when the gamma goes down, the episode needed to converge become more. The best number of hidden layer is around 128 - 256. If the learning rate is too large, the model can't converge.

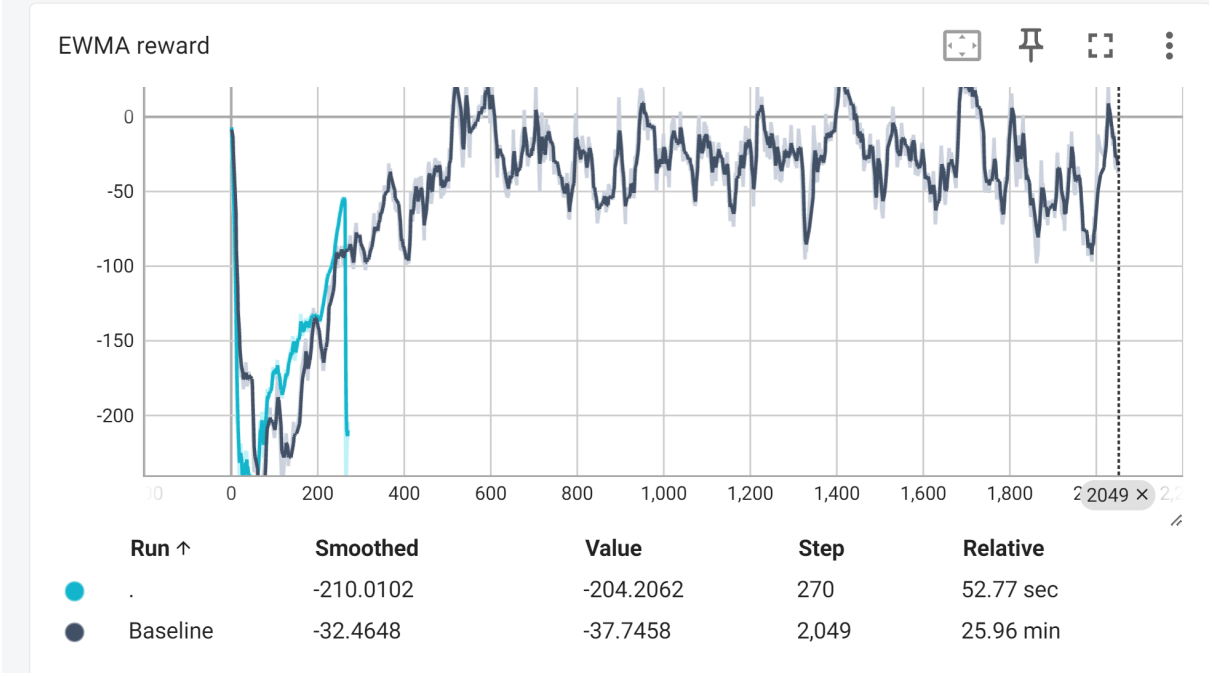
I found that if you don't normalization the return value, you the model can't not converge.

I directly use the Value function as my baseline because I saw that most of the people on the Internet use it, and the performance of Value function as baseline is much better than others baseline created by my brain.

hiddden layer	gamma	learning rate	episode needed
200	0.999	0.02	700

```
Episode 689     length: 291     reward: 41.609508309897734      ewma reward: 64.38382495270776
Episode 690     length: 317     reward: 250.28554953617203      ewma reward: 73.67891118188096
Episode 691     length: 248     reward: 42.695163497003875      ewma reward: 72.1297237976371
Episode 692     length: 590     reward: 129.00051895718104      ewma reward: 74.97326355561428
Episode 693     length: 366     reward: 222.22824313959796      ewma reward: 82.33601253481346
Episode 694     length: 656     reward: 210.3714486727946       ewma reward: 88.73778434171251
Episode 695     length: 405     reward: 253.6177610339569       ewma reward: 96.98178317632473
Episode 696     length: 424     reward: 220.9088162994571       ewma reward: 103.17813483248133
Episode 697     length: 479     reward: 229.13946877145312      ewma reward: 109.47620152942993
Episode 698     length: 417     reward: 153.90794014093436      ewma reward: 111.69778846000516
Episode 699     length: 648     reward: 193.97304454563888      ewma reward: 115.81155126428683
Episode 700     length: 638     reward: 240.39862393505047      ewma reward: 122.04090489782502
Solved! Running reward is now 122.04090489782502 and the last episode runs to 638 time steps!
/usr/local/lib/python3.10/dist-packages/gym/core.py:49: DeprecationWarning: WARN: You are calling render method, but you didn't specified the argument render_m
If you want to render in human mode, initialize the environment in this way: gym.make('EnvName', render_mode='human') and don't call the render method.
See here for more information: https://www.gymlibrary.ml/content/api/
  deprecation(
Episode 1       Reward: 196.5905449644424
Episode 2       Reward: -25.48382761693746
Episode 3       Reward: 210.03617741709996
Episode 4       Reward: 240.3397107606958
Episode 5       Reward: 213.14751780437075
Episode 6       Reward: 159.83732163576428
Episode 7       Reward: 232.70809434744083
Episode 8       Reward: 194.09863794563637
Episode 9       Reward: 142.92447458829315
Episode 10      Reward: 231.16624599209732
```

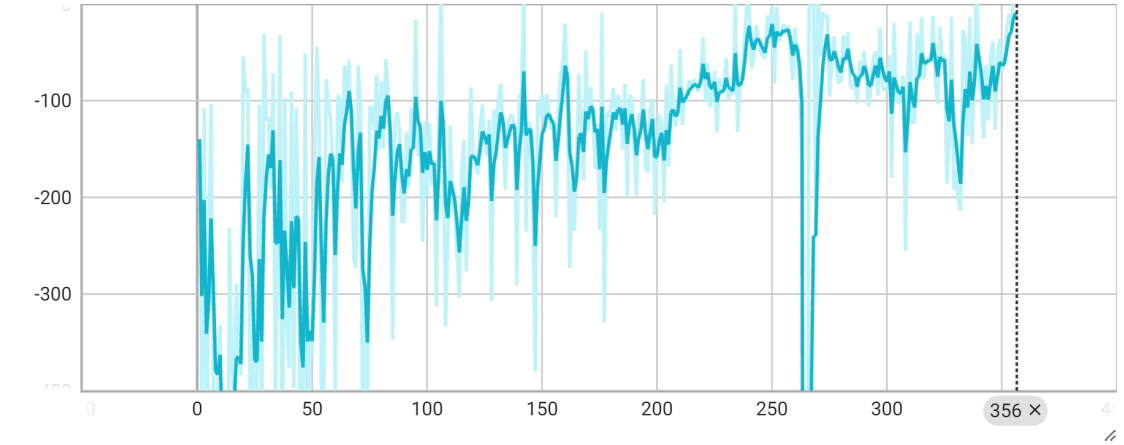
EWMA reward



Run ↑	Smoothed	Value	Step	Relative
.	-210.0102	-204.2062	270	52.77 sec
Baseline	-32.4648	-37.7458	2,049	25.96 min

Episode reward

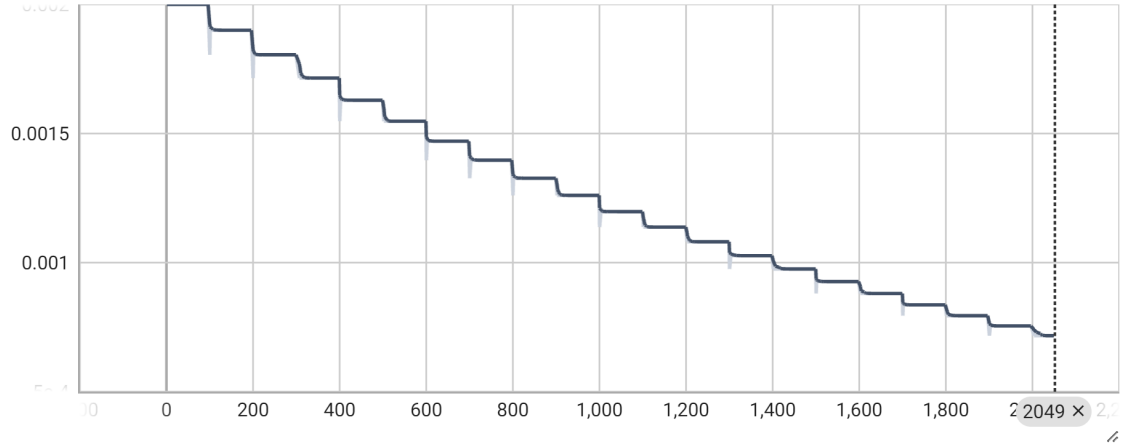
Episode reward



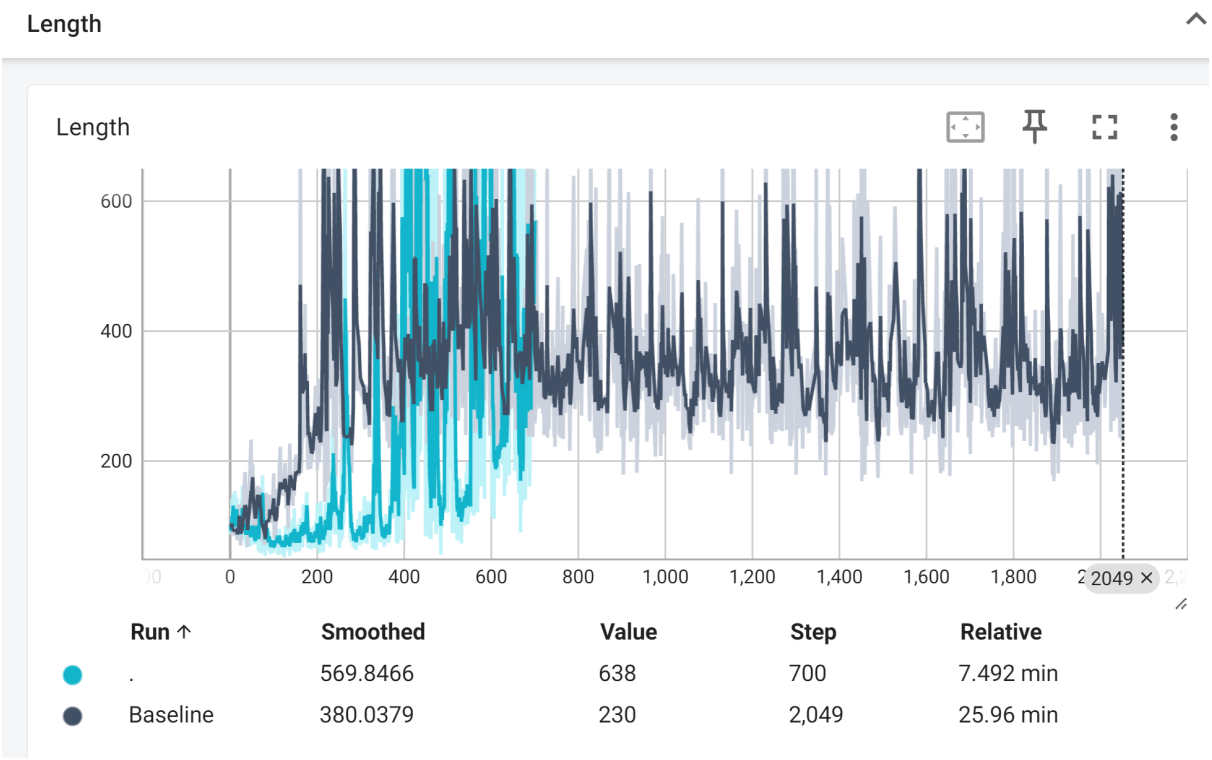
Run	Smoothed	Value	Step	Relative
.	-9.5563	-2.5712	356	1.301 min

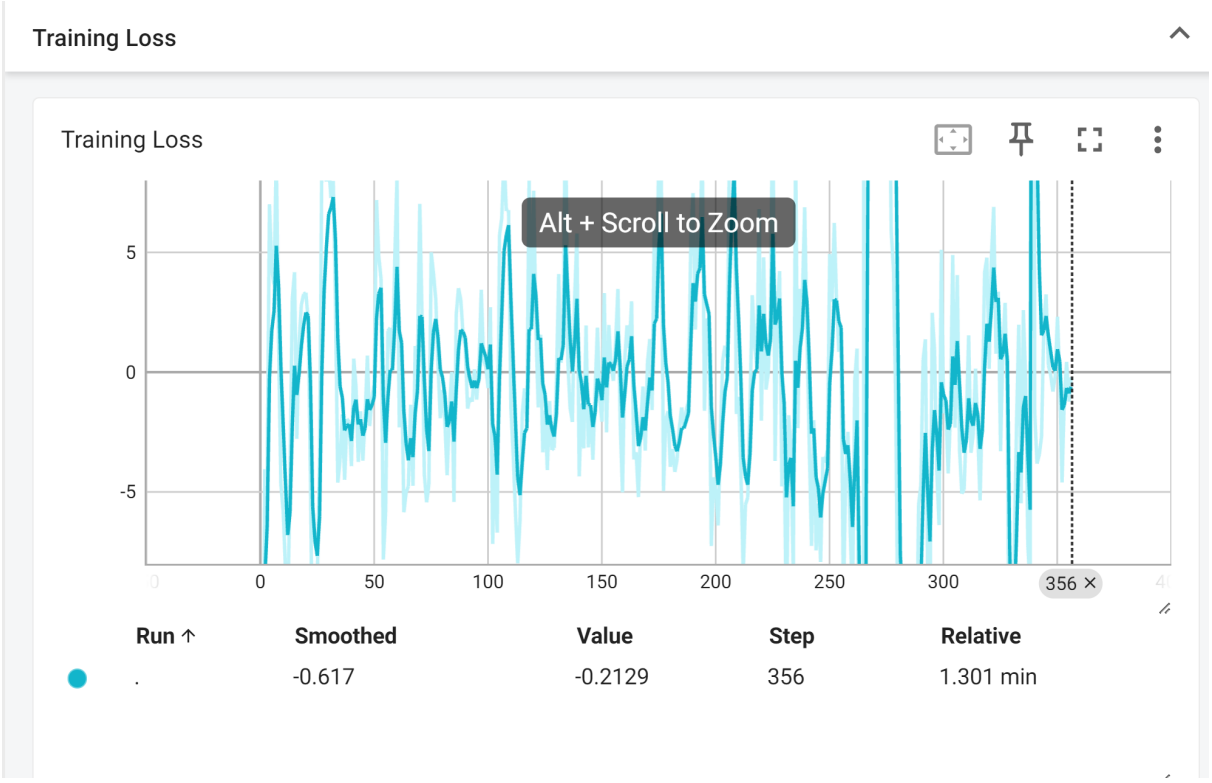
Learning Rate

Learning Rate



Run	Smoothed	Value	Step	Relative
Baseline	0.0007	0.0007	2,049	25.96 min





(c)

I add two shared layer in my neuro network, and the activation function are both ReLu. I don't use dropout here.

I found that when the gamma goes down, the episode needed to converge become more. The best number of hidden layer is around 128 - 256. If the learning rate is too large, the model can’t converge.

I found that if you don’t normalization the return value, you the model can’t not converge.

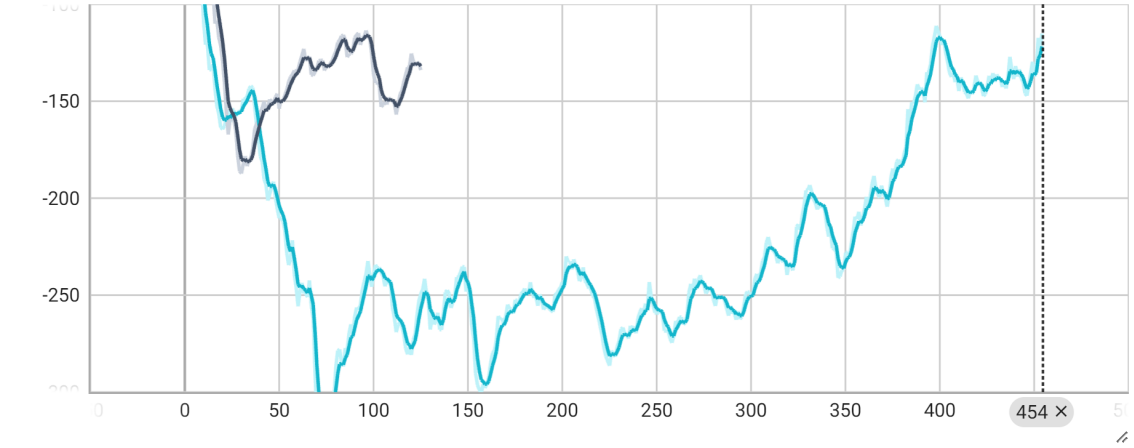
I found that if the lambda value is too small, it becomes much more harder to converge. I have tried lambda = 0.5, and although the episode grows to 10000, it hadn’t converge. As you can see in the following picure below, the big lambda value require more episode to converge.

lambda = 0.999

hiddden layer	gamma	learning rate	lambda	episode needed
128	0.999	0.02	0.999	991

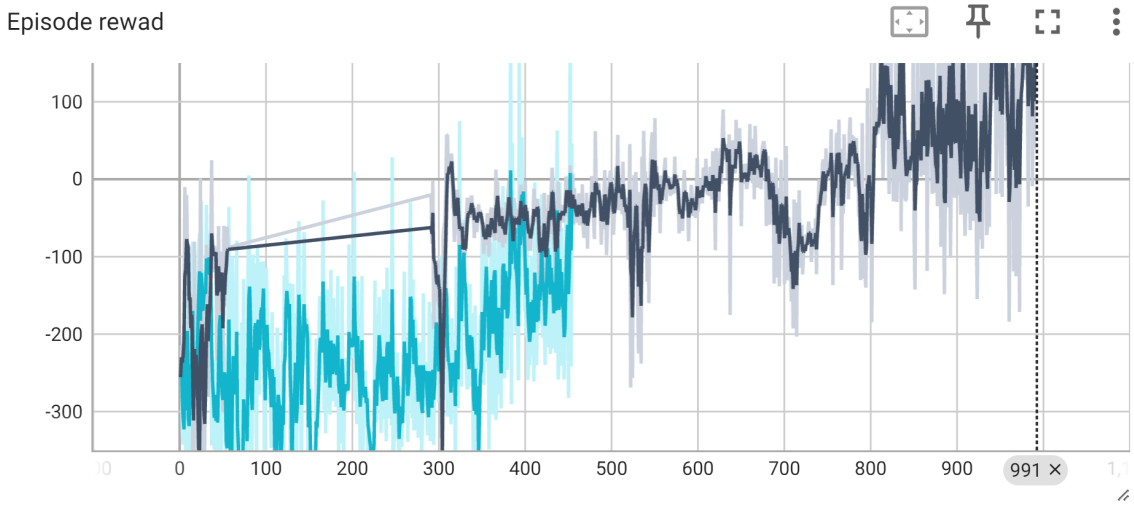
```
Episode 974     length: 510     reward: 189.02468647594657      ewma reward: 80.69886187841036
Episode 975     length: 735     reward: 144.2105305077076       ewma reward: 83.87444530987521
Episode 976     length: 565     reward: 183.41215531368823      ewma reward: 88.85133081006586
Episode 977     length: 457     reward: 177.92791015005682      ewma reward: 93.30515977706541
Episode 978     length: 636     reward: 128.23865190689764      ewma reward: 95.05183438355702
Episode 979     length: 468     reward: 185.56345186737983      ewma reward: 99.57741525774816
Episode 980     length: 533     reward: 251.25945417964465      ewma reward: 107.16151720384298
Episode 981     length: 750     reward: 154.819534502473        ewma reward: 109.54441806877446
Episode 982     length: 414     reward: -35.35073228711242      ewma reward: 102.29966055098011
Episode 983     length: 605     reward: 155.54759061840082      ewma reward: 104.96205705435113
Episode 984     length: 568     reward: 198.2579583478523       ewma reward: 109.62685211902618
Episode 985     length: 663     reward: 134.37448917789726      ewma reward: 110.86423397196974
Episode 986     length: 769     reward: 137.4608181848917       ewma reward: 112.19406318261582
Episode 987     length: 331     reward: -9.1279184458146        ewma reward: 106.1279641011943
Episode 988     length: 583     reward: 234.4044884500602       ewma reward: 112.54179031863758
Episode 989     length: 266     reward: 38.75072584283291       ewma reward: 108.85223709484734
Episode 990     length: 498     reward: 221.9270852886358       ewma reward: 114.50597950453677
Episode 991     length: 660     reward: 229.05294490445988      ewma reward: 120.23332777453291
Solved! Running reward is now 120.23332777453291 and the last episode runs to 660 time steps!
Episode 1       Reward: 209.33887503677477
Episode 2       Reward: -15.40489345366359
Episode 3       Reward: 162.82639343497598
Episode 4       Reward: 123.37926464268543
Episode 5       Reward: 238.9275000757813
Episode 6       Reward: 165.20897533675662
Episode 7       Reward: -202.39344194732544
Episode 8       Reward: 197.6434686248573
Episode 9       Reward: 153.55577212815058
Episode 10      Reward: 193.14567038029531
```

EWMA reward



Run	Smoothed	Value	Step	Relative
.	-131.9227	-133.9458	125	20.93 sec
GAE	-121.8082	-115.0897	454	1.974 min

Episode rewad



Run	Smoothed	Value	Step	Relative
.	181.2416	229.0529	991	10.02 min
GAE	-36.7066	46.0534	454	1.974 min

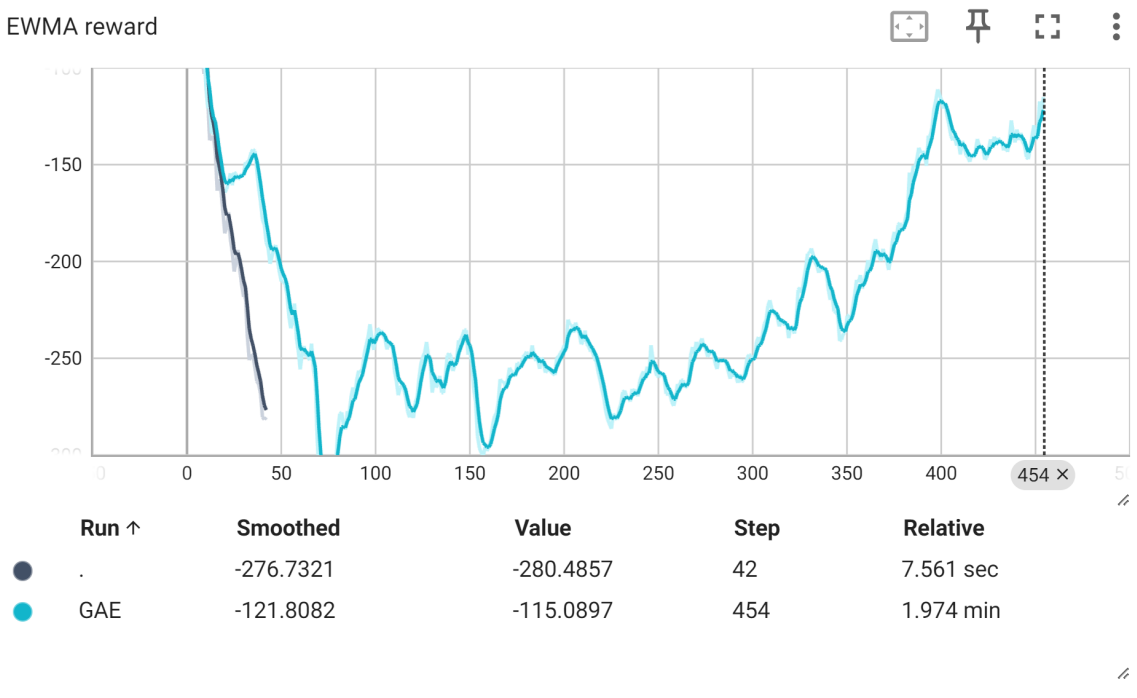
lambda = 0.8

hiddden layer	gamma	learning rate	lambda	episode needed

128	0.999	0.02	0.8	1121

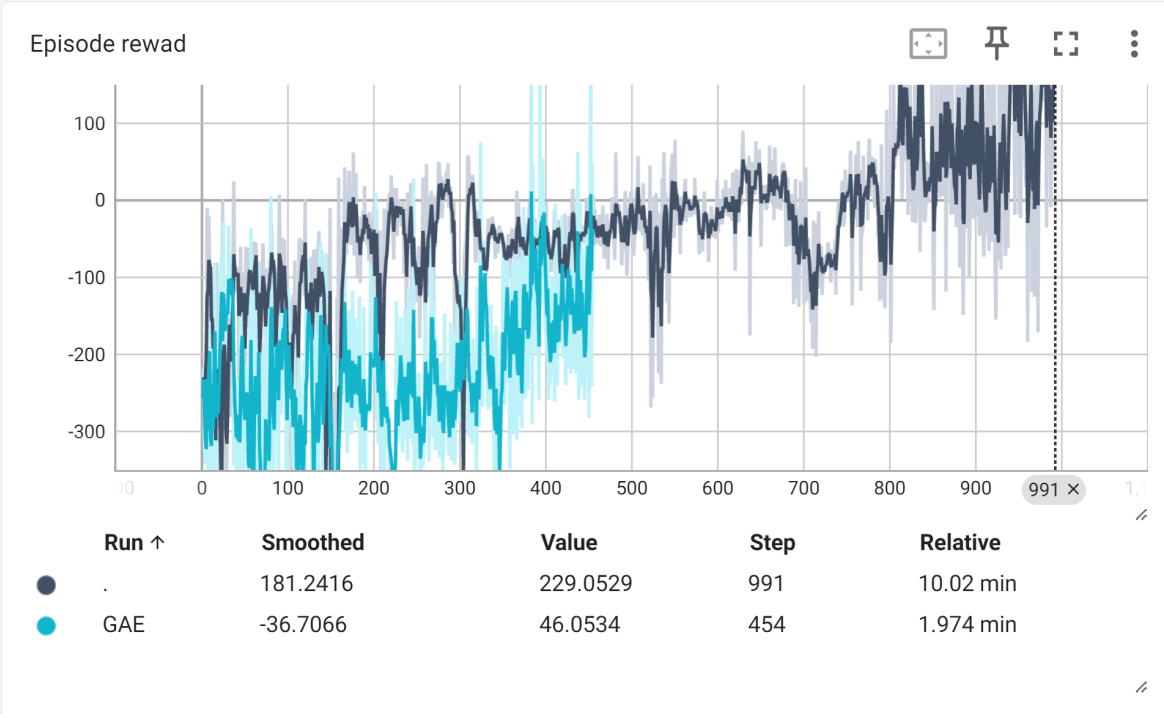
```
Episode 1109    length: 268     reward: -2.1990687801162494     ewma reward: 78.13826421349869
Episode 1110    length: 319     reward: 257.04478432324817      ewma reward: 87.08359021898616
Episode 1111    length: 683     reward: 276.35733249192134      ewma reward: 96.54727733263292
Episode 1112    length: 572     reward: -57.97371374585377      ewma reward: 88.82122777870858
Episode 1113    length: 170     reward: 30.151994966943107      ewma reward: 85.8877661381203
Episode 1114    length: 210     reward: 5.320178056627967       ewma reward: 81.85938673404569
Episode 1115    length: 546     reward: 226.13028461859477      ewma reward: 89.07293162827314
Episode 1116    length: 349     reward: 245.6056284469832       ewma reward: 96.89956646920864
Episode 1117    length: 272     reward: 43.93170538494422       ewma reward: 94.2511734149954
Episode 1118    length: 333     reward: 291.5794074600534       ewma reward: 104.11758511724831
Episode 1119    length: 313     reward: 281.84165732882417      ewma reward: 113.0037887278271
Episode 1120    length: 397     reward: 214.8232475401364       ewma reward: 118.09476166844256
Episode 1121    length: 451     reward: 233.01837359097956      ewma reward: 123.8409422645694
Solved! Running reward is now 123.8409422645694 and the last episode runs to 451 time steps!
/usr/local/lib/python3.10/dist-packages/gym/core.py:49: DeprecationWarning: WARN: You are calling render method, but you didn't specified the argument render_m
If you want to render in human mode, initialize the environment in this way: gym.make('EnvName', render_mode='human') and don't call the render method.
See here for more information: https://www.gymlibrary.ml/content/api/
  deprecation(
Episode 1       Reward: 247.0316636598362
Episode 2       Reward: 229.62821142472328
Episode 3       Reward: 223.99276251029988
Episode 4       Reward: 25.098684707868827
Episode 5       Reward: -87.95116761930181
Episode 6       Reward: 218.0064286611953
Episode 7       Reward: 281.23469829704686
Episode 8       Reward: 214.73610650776502
Episode 9       Reward: 121.92930109999357
Episode 10      Reward: -5.497648375451604
```

EWMA reward



Run ↑	Smoothed	Value	Step	Relative
.	-276.7321	-280.4857	42	7.561 sec
GAE	-121.8082	-115.0897	454	1.974 min

Episode rewad



Run ↑	Smoothed	Value	Step	Relative
.	181.2416	229.0529	991	10.02 min
GAE	-36.7066	46.0534	454	1.974 min

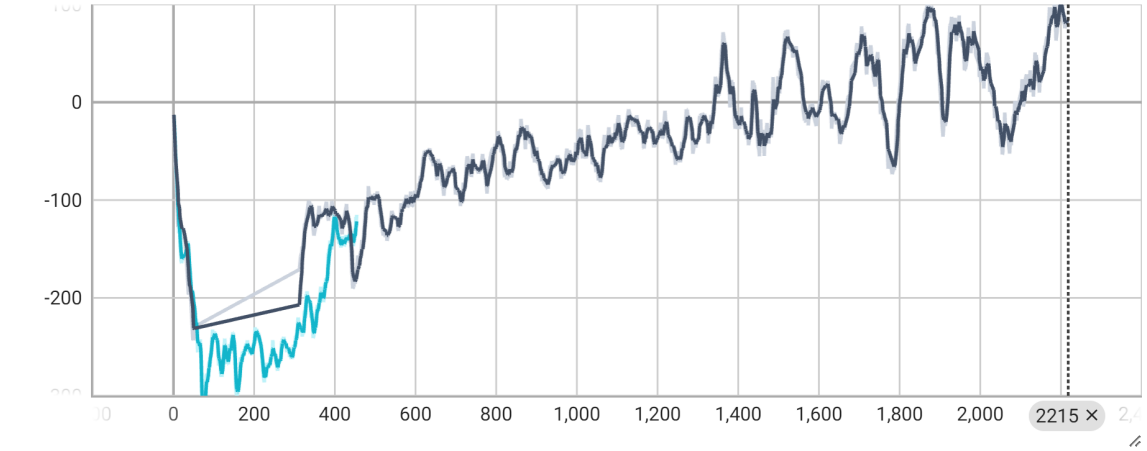
lambda = 0.7

hiddden layer	gamma	learning rate	lambda	episode needed
128	0.999	0.02	0.7	2476

```
Episode 2462    length: 310     reward: 290.19495643991803       ewma reward: 49.081531996575386
Episode 2463    length: 266     reward: 274.93390514705203       ewma reward: 60.37415065409921
Episode 2464    length: 221     reward: -153.69216235960477      ewma reward: 49.670835003414005
Episode 2465    length: 363     reward: 22.17562643233373        ewma reward: 48.29607457485999
Episode 2466    length: 403     reward: 234.34283204580527       ewma reward: 57.59841244840725
Episode 2467    length: 397     reward: 246.37512616003065       ewma reward: 67.03724813398841
Episode 2468    length: 369     reward: 247.52191982691144       ewma reward: 76.06148171863455
Episode 2469    length: 344     reward: 245.5563548976075        ewma reward: 84.53622537758321
Episode 2470    length: 418     reward: 228.44987440524326       ewma reward: 91.73190782896621
Episode 2471    length: 388     reward: 238.45261939867464       ewma reward: 99.06794340745164
Episode 2472    length: 338     reward: 270.84159383275266       ewma reward: 107.6566259287167
Episode 2473    length: 387     reward: 231.959651228703         ewma reward: 113.871777193716
Episode 2474    length: 176     reward: 27.291134333496856       ewma reward: 109.54274505070504
Episode 2475    length: 331     reward: 271.27647731765416       ewma reward: 117.62943166405249
Episode 2476    length: 570     reward: 173.3700016465018        ewma reward: 120.41646016317496
Solved! Running reward is now 120.41646016317496 and the last episode runs to 570 time steps!
/usr/local/lib/python3.10/dist-packages/gym/core.py:49: DeprecationWarning: WARN: You are calling render method, but you didn't specified the argument render_
If you want to render in human mode, initialize the environment in this way: gym.make('EnvName', render_mode='human') and don't call the render method.
See here for more information: https://www.gymlibrary.ml/content/api/
  deprecation(
Episode 1       Reward: 273.93468688881615
Episode 2       Reward: 4.750339050426021
Episode 3       Reward: 241.03648524614556
Episode 4       Reward: 230.78047857005654
Episode 5       Reward: 305.57814213651943
Episode 6       Reward: 199.7426668547442
Episode 7       Reward: 234.50587611265348
Episode 8       Reward: -69.17246394382774
Episode 9       Reward: -2.688902531771703
Episode 10      Reward: 228.81462087743918
```

EWMA reward

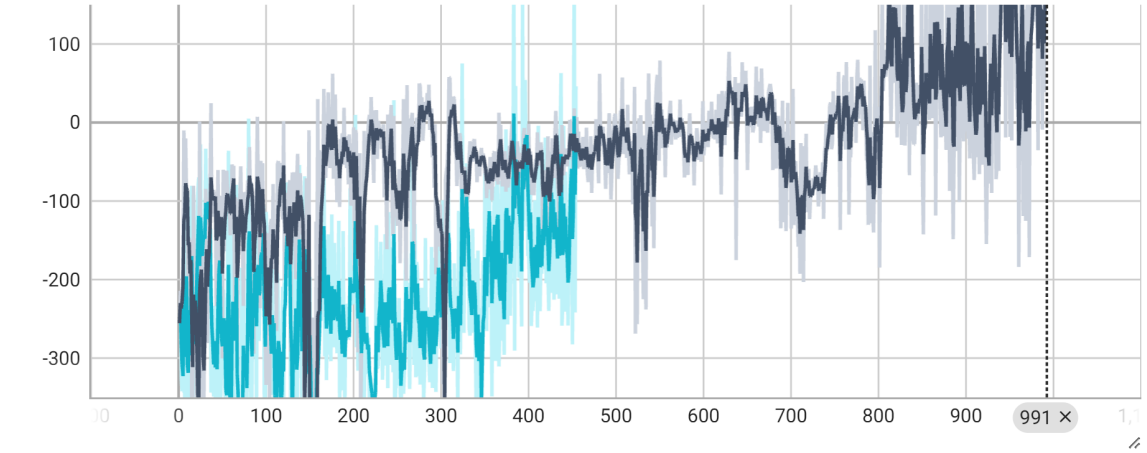
EWMA reward



Run ↑	Smoothed	Value	Step	Relative
.	83.8591	87.3847	2,215	25.58 min
GAE	-121.8082	-115.0897	454	1.974 min

Episode rewad

Episode rewad



Run ↑	Smoothed	Value	Step	Relative
.	181.2416	229.0529	991	10.02 min
GAE	-36.7066	46.0534	454	1.974 min