CS446 HW6

Junsheng Huang

April 2024

1 Bellman Equation

1.1

$$Q^{\pi}(s,a) = E[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k) | s_0 = s, a_0 = a, a_k \sim \pi(a|s_k), s_{k+1} \sim p(s_{k+1}|s_k, a_k)]$$

1.2

$$\begin{aligned}
Q^{\pi}(s,a) &= E[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k) | s_0 = s, a_0 = a, a_k \sim \pi(a|s_k), s_{k+1} \sim p(s_{k+1}|s_k, a_k)] \\
&= R(s,a) + E[\sum_{k=1}^{\infty} \gamma^k R(s_k, a_k) | s_0 = s, a_0 = a, a_k \sim \pi(a|s_k), s_{k+1} \sim p(s_{k+1}|s_k, a_k)] \\
&= R(s,a) + \gamma E[\sum_{k=1}^{\infty} \gamma^{k-1} R(s_k, a_k) | s_0 = s, a_0 = a, a_k \sim \pi(a|s_k), s_{k+1} \sim p(s_{k+1}|s_k, a_k)] \\
&= R(s,a) + \gamma E_{s' \sim Pr(s'|s_0,a_0)}[E_a[\sum_{k=1}^{\infty} \gamma^{k-1} R(s_k, a_k) | \pi, s_1 = s^{'}]] \\
&= R(s,a) + \gamma E_{s' \sim Pr(s'|s_0,a_0)}[V^{\pi}(s^{'})]
\end{aligned}$$

And we know $V^{\pi}(s) = Q^{\pi}(s, \pi(s))$, combining together we have the Bellman equation.

1.3

$$Q^{'}(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \gamma \ max_{a' \in A} Q(s^{'}, a^{'}))$$

1.4

$$max_{s,a} Q^*(s,a) = R(s,a) + \gamma E_{s' \sim Pr(s'|s,a)}[max_{a' \in A} Q^*(s^{'}, a^{'})]$$

Since it is optimal, consider $max_{a' \in A} Q(s^{'}, a^{'})$ is $max_{s,a} Q^*(s,a)$. Thus, we have $Q^*(s,a) = \frac{R_{max}}{1-\gamma}$

1.5

We know $Q^{'}(s,a) = 0.5Q(s,a) + 0.5(r + 0.5max_{a^{'} \in A}Q(s^{'}, a^{'}))$.
In (a), we have $Q^{'}(s_1, a_1) = 0.5Q(s_1, a_1)+0.5(-10+0.5max_{a^{'} \in A}Q(s_1, a^{'})) = -5$

-5	0
0	0

In (b), we have $Q^{'}(s_1, a_2) = 0.5Q(s_1, a_2)+0.5(-10+0.5max_{a^{'} \in A}Q(s_2, a^{'})) = -5$

-5	0
-5	0

In (c), we have $Q^{'}(s_2, a_1) = 0.5Q(s_2, a_1) + 0.5(18.5 + 0.5max_{a^{'} \in A}Q(s_1, a^{'})) = 8$

-5	8
-5	0

In (d), $Q^{'}(s_1, a_2) = 0.5Q(s_1, a_2) + 0.5(-10 + 0.5max_{a^{'} \in A}Q(s_2, a^{'})) = -5.5$

-5	8
-5.5	0

The final strategy is: $\pi(s_1) = a_1$, $\pi(s_2) = a_1$

2 Combination Lock

2.1

consider E_i as the expected number of steps from s_1 to s_i. We have:

$$E_i = \frac{1}{2}(E_{i-1} + 1) + \frac{1}{2}(E_{i-1} + 1 + E_i)$$

The start point is $E_1 = 0$, $E_2 = 2$. Solving the equation we have:

$$E_n = 2^n - 2$$

2.2

consider $Q_i = Q(s_i, a_1) + Q(s_i, a_2)$. With this notation and definition of Q-value,for Q_n, we have:

$$Q(s_n, a_1) = \frac{1}{2}\gamma Q(s_n, a_1) + \frac{1}{2}\gamma Q(s_n, a_2) + 1$$

$$Q(s_n, a_2) = \frac{1}{2}\gamma Q(s_n, a_1) + \frac{1}{2}\gamma Q(s_n, a_2)$$

Thus, $Q_n = \frac{1}{1-\gamma}$,$Q(s_n, a_1) = \frac{1-\frac{1}{2}\gamma}{1-\gamma}$, $Q(s_n, a_2) = \frac{\frac{1}{2}\gamma}{1-\gamma}$ For $i = 1, 2...n-1$, we have:

$$Q(s_i, a_1) = \frac{1}{2}\gamma Q(s_{i+1}, a_1) + \frac{1}{2}\gamma Q(s_{i+1}, a_2)$$

$$Q(s_i, a_2) = \frac{1}{2}\gamma Q(s_1, a_1) + \frac{1}{2}\gamma Q(s_1, a_2)$$

$$\begin{aligned} Q_1 &= \frac{1}{2}\gamma Q_2 + \frac{1}{2}\gamma Q_1 \\ &= (\frac{1}{2}\gamma)^{i-1} Q_i + (\frac{1}{2}\gamma + ... + (\frac{1}{2}\gamma)^{i-1}) Q_1 \\ &= (\frac{1}{2}\gamma)^{n-1} Q_n + (\frac{1}{2}\gamma + ... + (\frac{1}{2}\gamma)^{n-1}) Q_1 \end{aligned}$$

So, we have:

$$Q_1 = \frac{1-\frac{1}{2}\gamma}{1-\gamma+(\frac{1}{2}\gamma)^n}(\frac{1}{2}\gamma)^{n-1}\frac{1}{1-\gamma}$$

$$Q_i = \frac{1-\gamma+(\frac{1}{2}\gamma)^i}{1-\gamma+(\frac{1}{2}\gamma)^n}(\frac{1}{2}\gamma)^{n-i}\frac{1}{1-\gamma}$$

Thus, for $i = 1, 2...n-1$:

$$Q(s_i, a_1) = \frac{1}{2}\gamma Q_{i+1} = \frac{1-\gamma+(\frac{1}{2}\gamma)^{i+1}}{1-\gamma+(\frac{1}{2}\gamma)^n}(\frac{1}{2}\gamma)^{n-i-1}\frac{1}{1-\gamma}$$

$$Q(s_i, a_2) = \frac{1}{2}\gamma Q_1 = \frac{1-\frac{1}{2}\gamma}{1-\gamma+(\frac{1}{2}\gamma)^n}(\frac{1}{2}\gamma)^{n}\frac{1}{1-\gamma}$$

2.3

From (2), we know $Q(s_i, a_1) > Q(s_i, a_2)$ for every $i = 1, 2...n-1$, thus always choose a_1, so the expected number of steps to get from state s_1 to s_k is k-1.

3 Q-value initialization

3.1

For Q_1, the Q-value is all 0, so the same as Question2, $E = 2^n - 2$.
For Q_2, the Q-value is all 1, so also the same as Question2, $E = 2^n - 2$

3.2

We have:

$$Q_1(s_i, a_1) = (1-\alpha)Q_1(s_i, a_1) + \alpha(R(s_i, a_1) + \gamma max_{a'}(Q_1(s_{i+1}, a')))$$

$$Q_1(s_i, a_2) = (1-\alpha)Q_1(s_i, a_1) + \alpha(R(s_i, a_2) + \gamma max_{a'}(Q_1(s_{i+1}, a')))$$

Since for all $i = 1, 2...n-1$ and $j = 1, 2$,$Q_1(s_i, a_j) = 0$ and $R(s_i, a_j) = 0$, we simply update nothing for the first time to reach s_n from s_1, thus the answer is the same as part 1, $E = 2^n - 2$

3.3

since we reset onetime, the $Q_1(s_n, a_1) = \alpha$. However, it still doesn't affect the policy we choose in $i = 1, 2...n-1$, thus the answer is the same as part 1, $E = 2^n - 2$

3.4

The replay buffer can allow the agent to learn from the past experience which might not happen in the near future, so it can reduce the number if episodes before π_1 is optimal.

3.5

The minimum number of steps for π_2 to reach s_n from s_1 is $n-1$, because the policy can randomly choose a_1 every time.

3.6

In the worst case, we always choose a_2 when meeting a random choice. And we have:

$$Q(s_{i-1}, a_2) = (1-\alpha)Q(s_{i-1}, a_2) + \alpha(r + \gamma max_a(Q(s_1, a)))$$

$$Q(s_{i-1}, a_1) = (1-\alpha)Q(s_{i-1}, a_1) + \alpha(r + \gamma max_a(Q(s_i, a)))$$

For every time you go to a new state, you always choose a_2 first, then go back to s_1. One simple thing is that $max_a(Q(s_i, a)) \leq max_a(Q(s_1, a))$. (Easy to think about since every time you update, you are reducing and s_1 update most of the time.) So the maximum number of steps is $2^n - 2$, since every time you go back to s_1, you need to update all states first.

4 Policy Gradient

4.1

$$\begin{aligned} E_{\tau\sim P_\pi}[R(\tau)\nabla_\theta log P_\pi(\tau)] &= E_{\tau\sim P_\pi}[R(\tau)\frac{1}{P_\pi(\tau)}\nabla_\theta P_\pi(\tau) \\ &= \sum_\tau P_\pi(\tau)R(\tau)\frac{1}{P_\pi(\tau)}\nabla_\theta P_\pi(\tau) \\ &= \nabla_\theta(\sum_\tau R(\tau)P_\pi(\tau)) \\ &= \nabla_\theta E_{\tau\sim\pi_\theta}[R(\tau)] \\ &= \nabla_\theta J_\theta \end{aligned}$$

$$P_\pi(\tau) = d_0(s_0)\prod_{i=1}^{T-1}\pi(a_i|s_i)P(s_{i+1}|s_i,a_i)\pi(a_T|s_T)$$

$$log(P_\pi(\tau)) = log(d_0(s_0)) + \sum_{i=1}^{T-1}(\pi(a_i|s_i) + P(s_{i+1}|s_i,a_i) + \pi(a_T|s_T))$$

$$\nabla_\theta log(P_\pi(\tau)) = \sum_{i=1}^{T}\nabla_\theta log(\pi_\theta(a_i|s_i))$$

So, we have:

$$\nabla_\theta J(\pi_\theta) = E_{\tau\sim P_\pi}[R(\tau)\nabla_\theta log\pi_\theta(a_i|s_i)]$$

4.2

$$\begin{aligned} J(\pi_\theta) &= E_{s\sim d_0}[V^{\pi_\theta}(s)] \\ &= E_{s\sim d_0, a\sim\pi_\theta(s)}[Q^{\pi_\theta}(s,a)] \\ &= E_{s\sim d_0}[\int_a Q^{\pi_\theta}(s,a)\pi_\theta(a|s)da] \end{aligned}$$

So, we have:

$$\begin{aligned} \nabla_\theta J(\pi_\theta) &= E_{s\sim d_0}[\int_a \nabla_\theta(Q^{\pi_\theta}(s,a)\pi_\theta(a|s))da] \\ &= E_{s\sim d_0}[\int_a \nabla_\theta(Q^{\pi_\theta}(s,a))\pi_\theta(a|s)da] + E_{s\sim d_0}[\int_a Q^{\pi_\theta}(s,a)\nabla_\theta(\pi_\theta(a|s))da] \\ &= E_{s\sim d_0, a\sim\pi_\theta(s)}[Q^{\pi_\theta}(s,a)\nabla_\theta log(\pi_\theta(a|s))] + E_{s\sim d_0}[\int_a Q^{\pi_\theta}(s,a)\nabla_\theta(\pi_\theta(a|s))da] \end{aligned} \tag{1}$$

And we know:

$$\nabla_\theta(Q^{\pi_\theta}(s,a)) = \nabla_\theta(R(s,a)+\gamma E_{s'\sim p(s'|s,a)}[Q^{\pi_\theta}(s',\pi(s'))]) = \gamma\nabla_\theta E_{s'\sim p(s'|s,a)}[Q^{\pi_\theta}(s',\pi(s'))]$$

Thus, the latter term in equation (1) become:

$$E_{s\sim d_0}[\int_a Q^{\pi_\theta}(s,a)\nabla_\theta(\pi_\theta(a|s))da] = E_{s\sim d_0, a\sim\pi_\theta(s)}[\nabla_\theta(Q^{\pi_\theta}(s,a))]$$
$$= \gamma E_{s'\sim d_1^\pi, a\sim\pi_\theta(s')}[\nabla_\theta(Q^{\pi_\theta}(s',a))]$$

Finally, we have:

$$\nabla_\theta J(\pi_\theta) = E_{s\sim d_0, a\sim\pi_\theta(s)}[Q^{\pi_\theta}(s,a)\nabla_\theta log(\pi_\theta(a|s))] + \sum_{i=1}^{\infty}\gamma^i E_{s\sim d_i^{pi}, a\sim\pi_\theta(s)}[Q^{\pi_\theta}(s,a)\nabla_\theta log(\pi_\theta(a|S))]$$
$$= \int_s E_{a\sim\pi_\theta(s)}[Q^{\pi_\theta}(s,a)\nabla_\theta log(\pi_\theta(a|s))](d_0(s) + \sum_{i=1}^{\infty}\gamma^i d_i^\pi(s))ds$$
$$= E_{s\sim d^\pi(s), a\sim\pi_\theta(s)}[Q^{\pi_\theta}(s,a)\nabla_\theta log(\pi_\theta(a|s))]$$

4.3

consider the constant part: if we fix s, then $f(s) = b$ is a constant.

$$E_{s\sim d^\pi, a\sim\pi_0(s)}[f(s)\nabla_\theta log(\pi_\theta(a|s))] = \int \pi_\theta(a|s) b\nabla_\theta log(\pi_\theta(a|s))da$$
$$= \int b\nabla_\theta \pi_\theta(a|s)da$$
$$= b\nabla_\theta \int \pi_\theta(a|s)da$$
$$= b\nabla_\theta \int 1$$
$$= 0$$

So, we have:

$$\nabla_\theta J(\pi_\theta) = E_{\tau\sim P_\pi}[(Q^\pi(s,a) - f(s))\nabla_\theta log\pi_\theta(a_i|s_i)]$$

5 Coding: Tabular Q-Learning

5.1

5.1.1

States correspond to taxi location, passenger location and destination. The possible states are 500. The action space is [move south, move north, move east, move west, pick up a passenger, drop off a passenger]

5.1.2

env.step() returns observation, reward, done and info(contains "p" and "action mask"). env.reset() resets the environment to an initial state and returns the

initial observation. States are represented by (taxi row, taxi col, passenger location, destination).

5.1.3

The valid render modes in gym.make is human,rgb array, rgb array list and ansi.
For Taxi-v3, in human mode, it has an animation of the whole process.
In ansi mode, it will return a strings containing a terminal-style text representation for each time step(car, locations...).
In rgb array mode, it will return a single frame representing the current state of the environment. A frame is a numpy.ndarray with shape (x, y, 3) representing RGB values for an x-by-y pixel image.
In rgb array list mode, it will return a list of frames representing the states of the environment since the last reset. Each frame is a numpy.ndarray with shape (x, y, 3), as with rgb array.

5.1.4

If self.render mode == "ansi", it will call self. render text().
If self.render mode in "human", "rgb array", it will calls self. render gui(self.render mode)

5.2

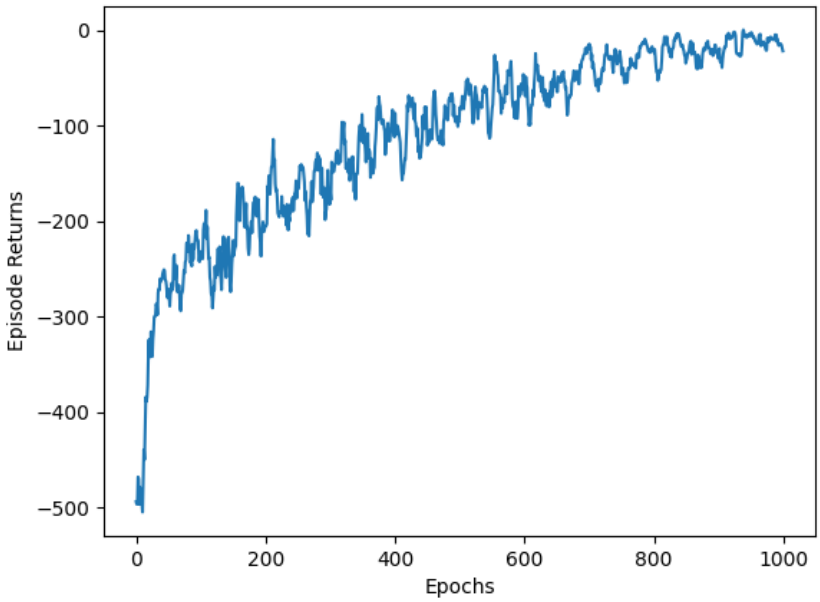

Figure 1: train rewards 5b

5.3

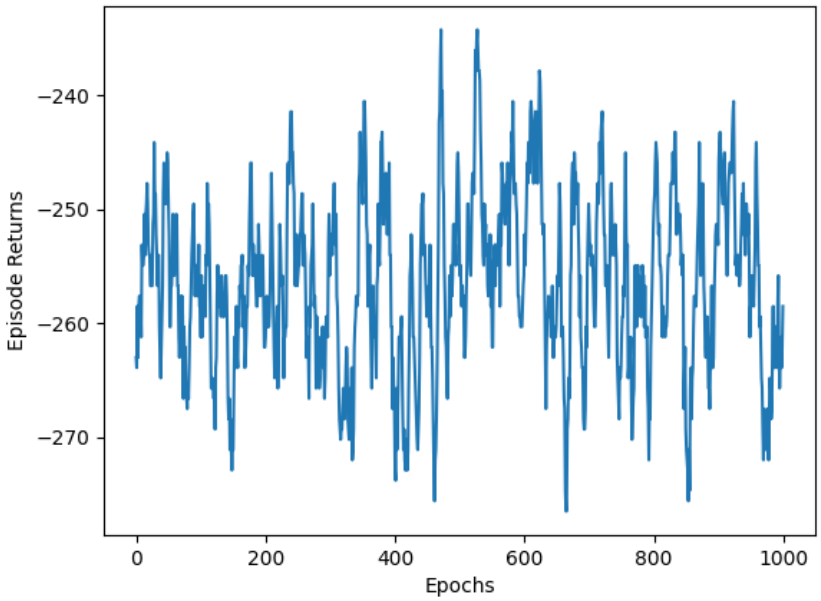

Figure 2: train rewards 5c

It become worse because all Q-value is too negative at first for the agent to explore new actions. Also, the agent need more epochs to learn the optimal strategy since it doesn't have a positive reward.

5.4

The percentage is 0.0454.

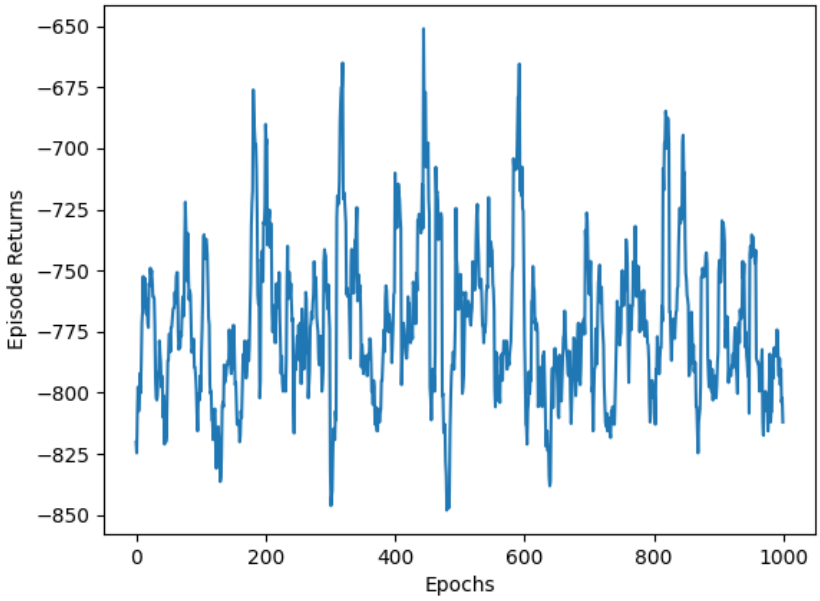

Figure 3: train rewards 5d

5.5

The $\epsilon = 0.7$

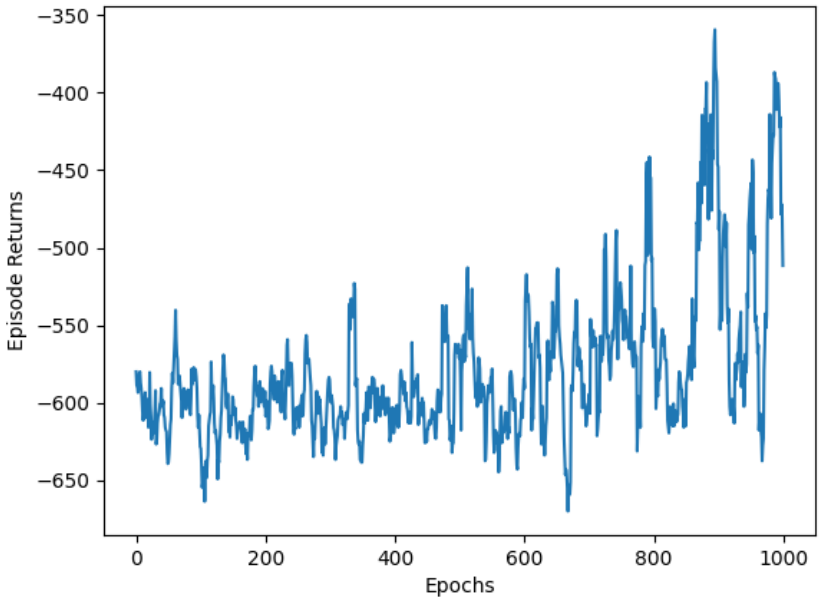

Figure 4: train rewards 5e

5.6

The percentage is 0.134.

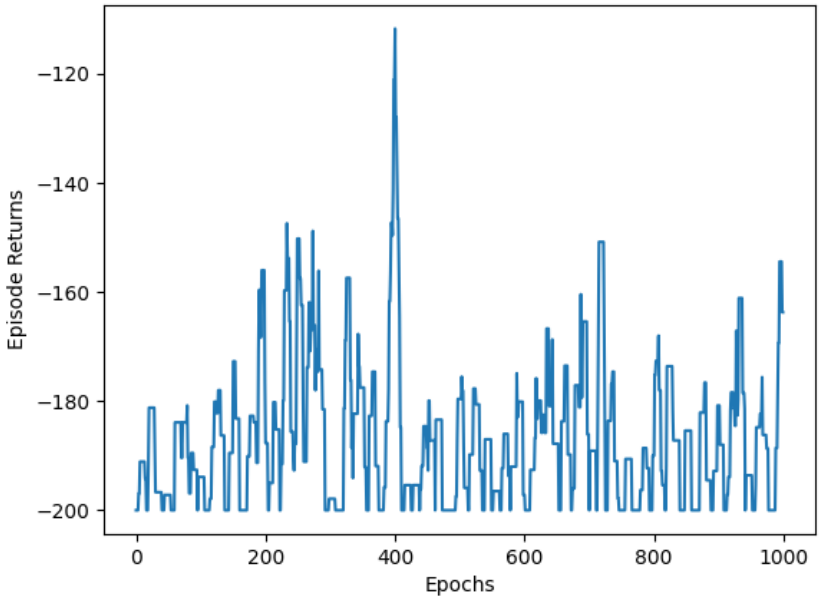

Figure 5: train rewards 5f

5.7

the percentage is 0.959.

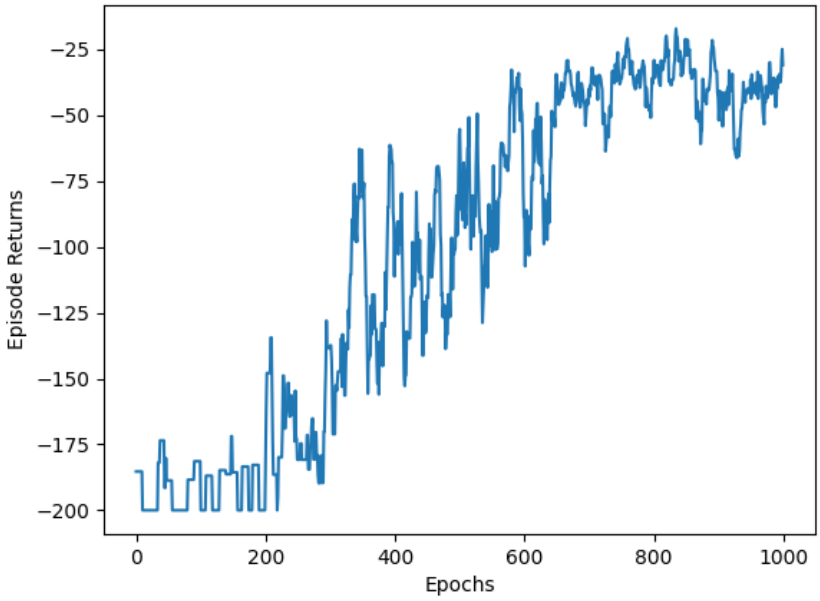

Figure 6: train rewards 5g