Reinforcement Learning HW 7

Mert Bilgin (7034879)
mert.bilgin@student.uni-tuebingen.de

Lalitha Sivakumar (6300674)
lalitha.sivakumar@student.uni-tuebingen.de

Kevin Van Le (7314700)
kevin-van.le@student.uni-tuebingen.de

December 4, 2025

1 Score Function for Gaussian Policy

(a)

To compute the policy gradient, we use the log-likelihood trick. This relates the gradient of the policy to the gradient of its logarithm:

$$\nabla_\theta \pi_\theta(s,a) = \pi_\theta(s,a) \cdot \frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)} = \pi_\theta(s,a) \cdot \nabla_\theta \log \pi_\theta(s,a) \tag{1}$$

We now compute the gradient of the policy with respect to θ. Taking the gradient of the Gaussian policy:

$$\begin{aligned} \nabla_\theta \pi_\theta(s,a) &= \nabla_\theta p(a|s,\theta) \\ &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(a-\mu(s,\theta))^2}{2\sigma^2}\right) \cdot \left(-\frac{(a-\mu(s,\theta))}{\sigma^2}\right) \cdot \nabla_\theta \mu(s,\theta) \end{aligned} \tag{2}$$

Since $\nabla_\theta \mu(s,\theta) = \nabla_\theta(\phi(s)^\top \theta) = \phi(s)$, we can simplify:

$$\nabla_\theta \pi_\theta(s,a) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(a-\mu(s,\theta))^2}{2\sigma^2}\right) \frac{(\mu(s,\theta)-a)}{\sigma^2} \cdot \phi(s) \tag{3}$$

Finally, we obtain the gradient of the log policy by dividing the gradient by the policy itself:

$$\nabla_\theta \log \pi_\theta(s,a) = \frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)} = \frac{\phi(s)^\top \theta - a}{\sigma^2} \phi(s) \tag{4}$$

2 Deep Q-Learning (Q-Networks)

(a)

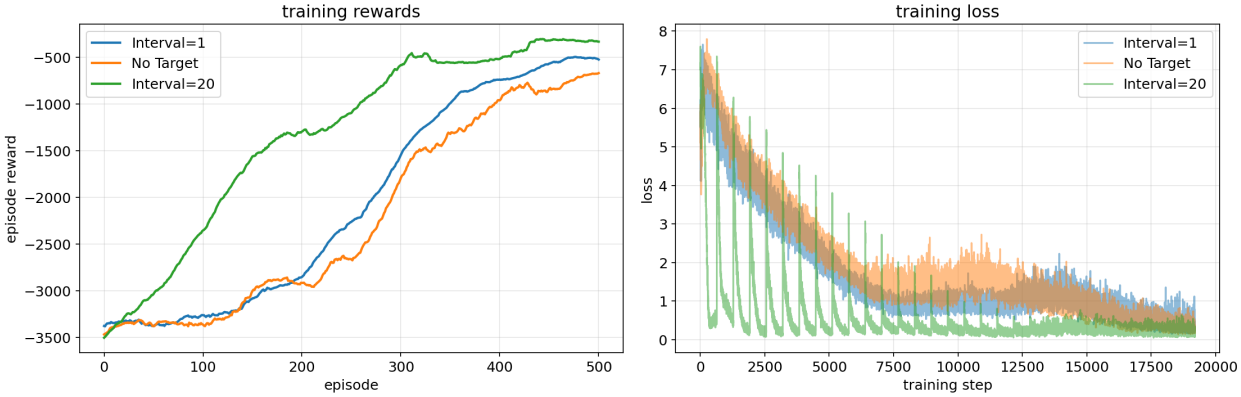
see Gym-DQN.ipynb

(b)

see Gym-DQN.ipynb

(c)

For the Pendulum task we compare three settings: target update every step (Interval=1), no target network, and target update every 20 steps (Interval=20).



The printed results are:

```
Pendulum Reward Summary:
Interval=1:  Train (last ep)=-1032.6, Test=-147.7+/-83.6
No Target:   Train (last ep)=-130.9, Test=-187.5+/-95.9
Interval=20: Train (last ep)=-263.1, Test=-643.2+/-1150.5
```

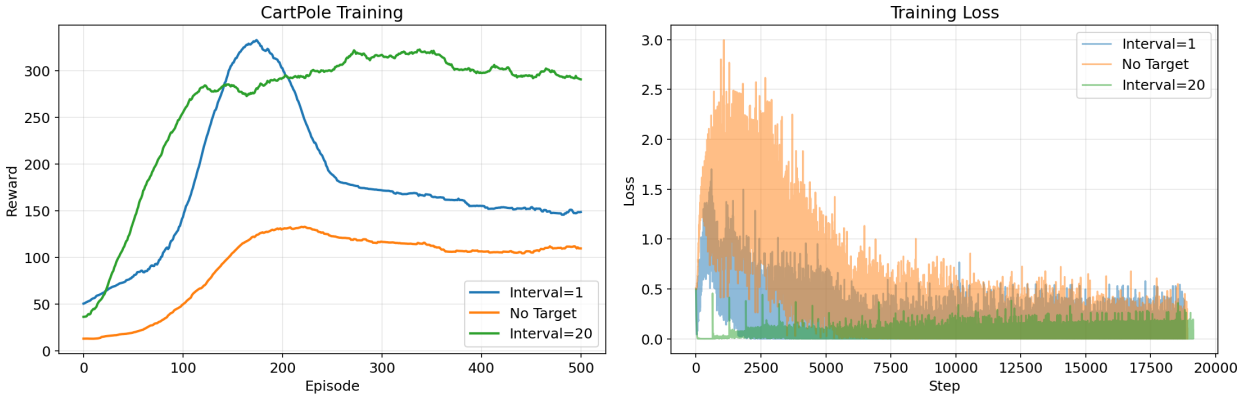
The rewards are very noisy and differ a lot between runs. The train rewards indicate the agent with an interval of 20 yields the highest average rewards, but during the evaluation the performance drops a lot. This indicates overfitting.

(d)

see Gym-DQN.ipynb

(e)

For the CartPole task we use the same three settings.



The printed results are:

```
Cartpole Reward Summary:
Interval=1:  Train (last ep)=176.0, Test=469.8+/-43.6
No Target:   Train (last ep)=115.0, Test=268.0+/-56.4
Interval=20: Train (last ep)=270.0, Test=243.9+/-22.5
```

Here all agents learn to solve the task, but the agent with interval=1 gives the highest test reward on average even though again the agent with interval=20 has the highest average train reward, most likely due to overfitting.