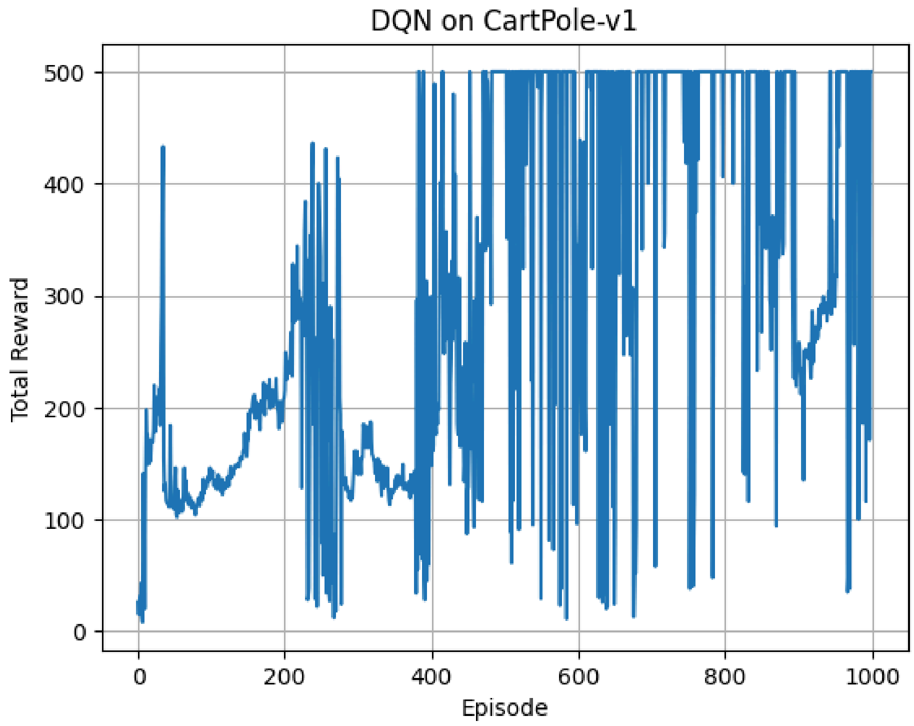

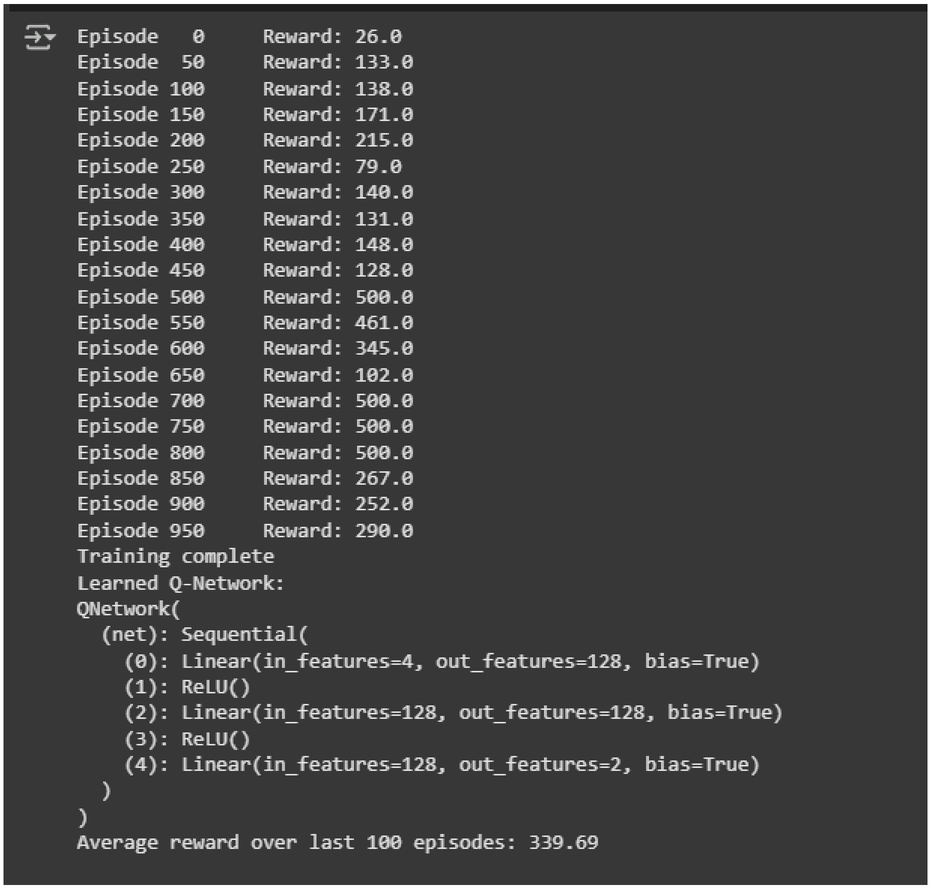

```
Episode   0     Reward: 26.0
Episode  50     Reward: 133.0
Episode 100     Reward: 138.0
Episode 150     Reward: 171.0
Episode 200     Reward: 215.0
Episode 250     Reward: 79.0
Episode 300     Reward: 140.0
Episode 350     Reward: 131.0
Episode 400     Reward: 148.0
Episode 450     Reward: 128.0
Episode 500     Reward: 500.0
Episode 550     Reward: 461.0
Episode 600     Reward: 345.0
Episode 650     Reward: 102.0
Episode 700     Reward: 500.0
Episode 750     Reward: 500.0
Episode 800     Reward: 500.0
Episode 850     Reward: 267.0
Episode 900     Reward: 252.0
Episode 950     Reward: 290.0
Training complete
Learned Q-Network:
QNetwork(
  (net): Sequential(
    (0): Linear(in_features=4, out_features=128, bias=True)
    (1): ReLU()
    (2): Linear(in_features=128, out_features=128, bias=True)
    (3): ReLU()
    (4): Linear(in_features=128, out_features=2, bias=True)
  )
)
Average reward over last 100 episodes: 339.69
```

Outputs

Reflection:

For Task 3 I implemented and trained a Deep Q-Network (DQN) agent on the CartPole-v1 environment in Gymnasium. My goal was to gain hands-on experience with neural-network–based value approximation, ε-greedy exploration, experience replay, and target-network updates, and to visualize how these pieces come together to solve a classic control problem.

I did not entirely understand what CartPole did, but I am happy I chose this one so I could learn about it.

This is what CartPole does:

A cart can move left or right along a one-dimensional, frictionless track. A pole (a rigid rod) is hinged to the cart and initially starts nearly upright. Physics (gravity + simple dynamics) tries to make the pole fall over. When training CartPole-v1, the goal is to train it to make the pole stand upright for 500 steps. When it does that, the reward is 500 (max reward). Essentially, the less steps it does, the less of a reward it gets.

For the outputs:

The first picture shows the graph of all of the episodes, and the reward for each episode. As you can see, at the beginning, the reward was not very high because the model was learning and failing a lot. At around the 400 episode mark, the model learned how to keep the pole upright very consistently. The fluctuations are due to it trying to learn more and see if it can find an even better way to keep the pole upright (the model is constantly trying to better itself, even though it already knows how to keep it up for 500 steps, it wants to find more efficient ways to do it).

The second picture shows the episodes and their rewards (only every 50 episodes, as that would be a lot to show all 1000 episodes). It also shows what it learned after doing all of this, and the average reward level of the last 100 episodes (to see how much better it got from the beginning to the end).