Reinforcement Learning Course Projects

Partha Sarathi Mohapatra
EE18D703

21 July 2020

(Not so) Deep RL

1 DQN for Cartpole Task

The control task of the 'CartPole' environment is solved by using Deep Q-Network (DQN) with experience replay and a separate target network to make the learning stable by decreasing the correlation among the training samples. We used early stopping to avoid over-fitting of training data by upper bounding the maximum number of weight updates (using maximum global steps). The learning curve and the optimal values of the hyperparameters are presented below.

1.1 Hyperparameter values and Plots

To solve this Cartpole problem we try different hyperparametes values and observe the reward evolution with the help of "TensorBoard" and settle with the following optimal values:

replay memory size = 10000
size of hidden layer 1 = 256
size of hidden layer 2 = 256
learning rate = 0.0001
minibatch size = 25
target update frequency = 50
discount factor(γ) = 0.99
starting epsilon value = 1.0
ending epsilon value = 0.02
exponential decay multiplier = 0.0001
maximum global steps = 75000 (for early stopping to avoid overfitting)

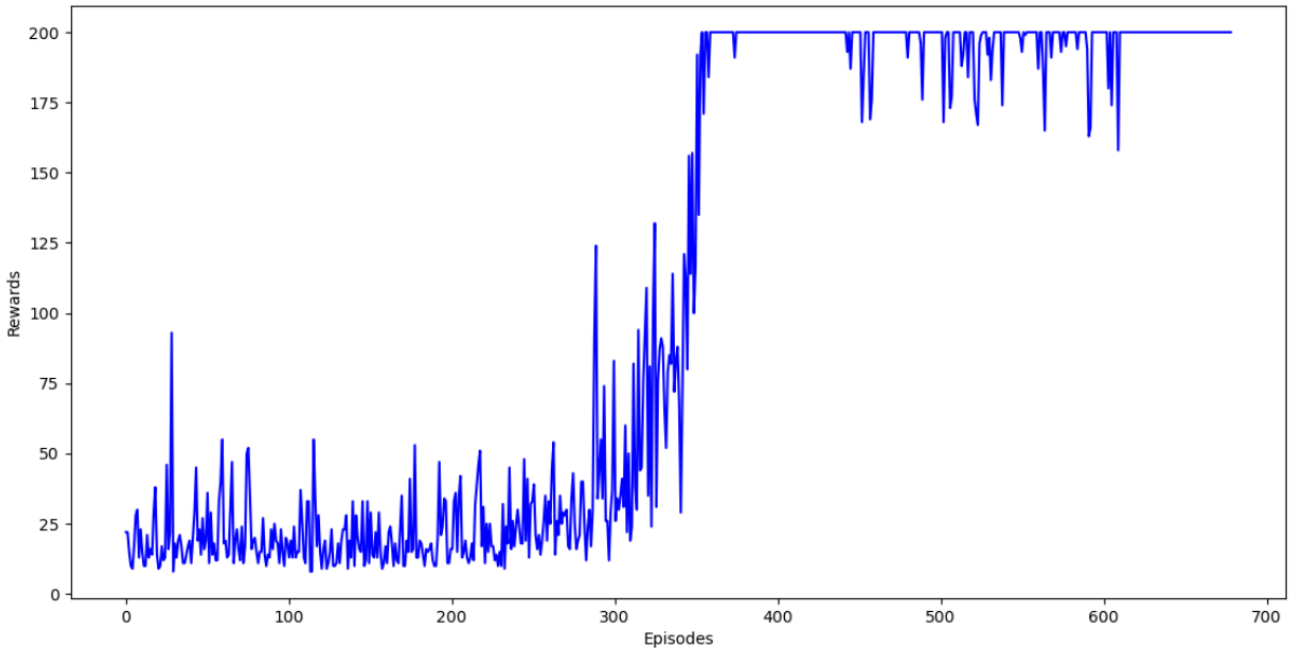

Figure 1: Rewards vs Episodes for the "Cartploe" task during training

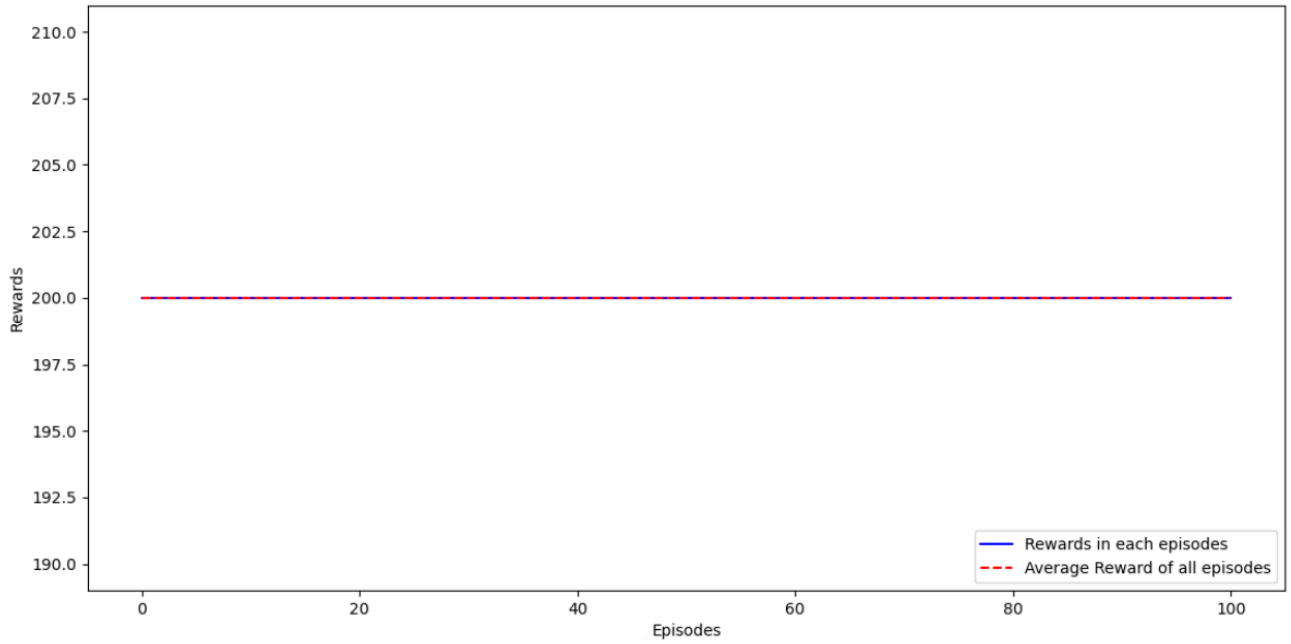

Figure 2: Rewards vs Episodes for the "Cartploe" task during play policy

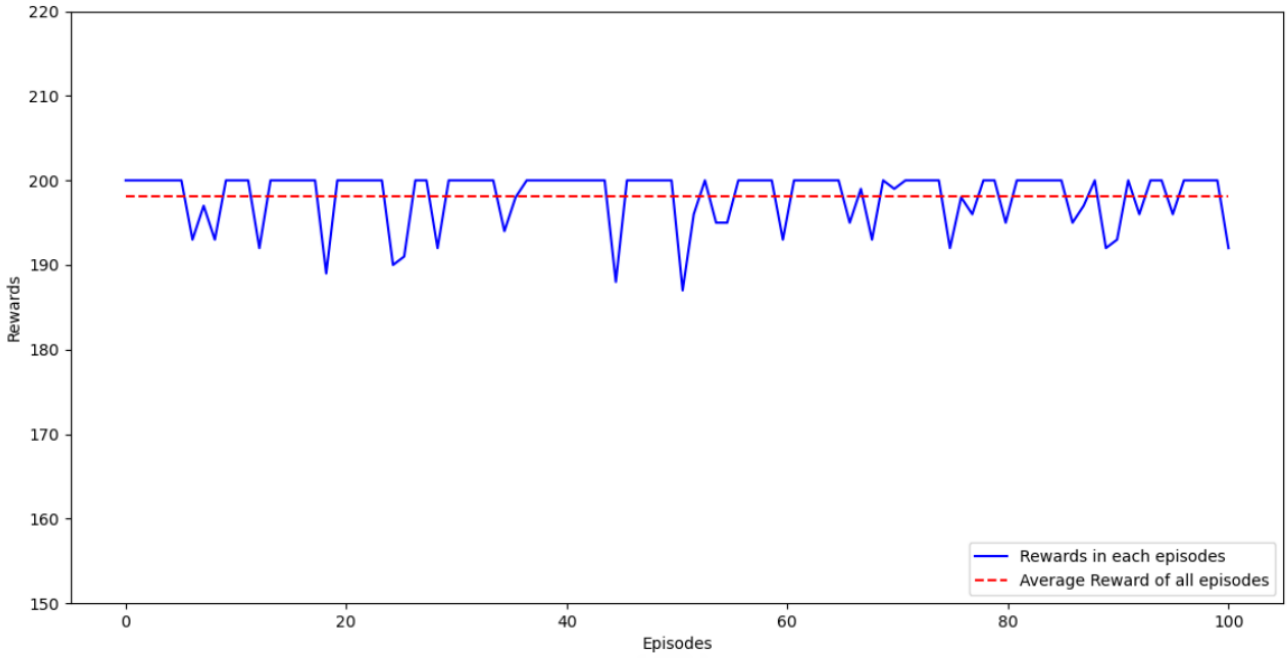

Figure 3: Rewards vs Episodes for the "Cartploe" task during play policy

In figure 19, we plot reward progress with episode for a typical instance of training, where the training is stopped much earlier (before the maximum episode number 2000) using maximum global steps to avoid over-fitting.

As shown in figure 20, almost always we get a reward of 200 in each episodes in the play policy. In very small occasion the rewards are below 200, but even then the average reward is above 195. One such rare instance is shown in the figure 21, where the average reward over 100 episodes is 198.84.

1.2 Observations and Inferences

During the tuning process, effect of variations of certain hyperparameters on the performance of DQN are observed and inferences from these observations are listed below:

(i) **Hidden layer size(s):**
Both hidden layer sizes are fixed at 256. Though smaller sizes also gives good results, these values are chosen to provide more flexibility (in terms of more weights to learn) and at the same time higher values are not selected to avoid over-fitting. We know neural networks are universal function approximator and for this reason they may over-fit the training data (which in this case can be biased data when exploration reaches its minimum value), so to avoid this over-fitting we use early stopping by restricting maximum global steps to some suitable value (75000).

(ii) **Epsilon:**
We observed that high exploration is needed at the beginning so as to fill the experience replay memory with uncorrelated transitions for stable learning, so we begin with epsilon value of 1.0. But as we progress through training and approach more and more closer to (local) optimal weights, we need to exploit by taking greedy actions, so we gradually decreased (by choosing exponential decay multiplier = 0.0001) epsilon to minimum value of 0.02. Epsilon is not decreased to 0 to avoid filling the experience replay with transitions only along taken trajectories (correlated data).

(iii) **Minibatch size:**
Descent number of transitions need to be sampled from the experience replay over which each Stochastic Gradient Descent (SGD) update can be computed. But we infer that making minibatch size very large will defeat the purpose of experience replay as the randomness of the sampled transitions, over which SGD is computed, will decrease making them more correlated.

(iv) **Learning rate:**
It was observed that the training process was very much sensitive to learning rate and higher values can lead to instability where as lower value was observed to have sluggish effect on learning. After many trial a suitable value of 0.0001 was chosen for the learning rate.

(v) **Target update frequency:**
Weights of the original network were copied to the corresponding weights of the target network at every 50th step (i.e. target update frequency was chosen 50). We infer that at lower value of target update frequency the non-stationary effect (due to frequent change in the target) can be high. On the other hand high frequency value can result over-fitting to (fixed) incorrect target.

2 DQN without Experience Replay and/or Target Network (Bonus)

After fixing the hyperparameters at the optimal values, we remove the experience replay and/or target network, to see its effect on the learning. We have listed all the cases below along with the corresponding learning curves.All the observations for the cases of removing experience replay and/or target network are listed in section 2.2.1. at the end.

(a) Without Target Network but with Experience Replay:

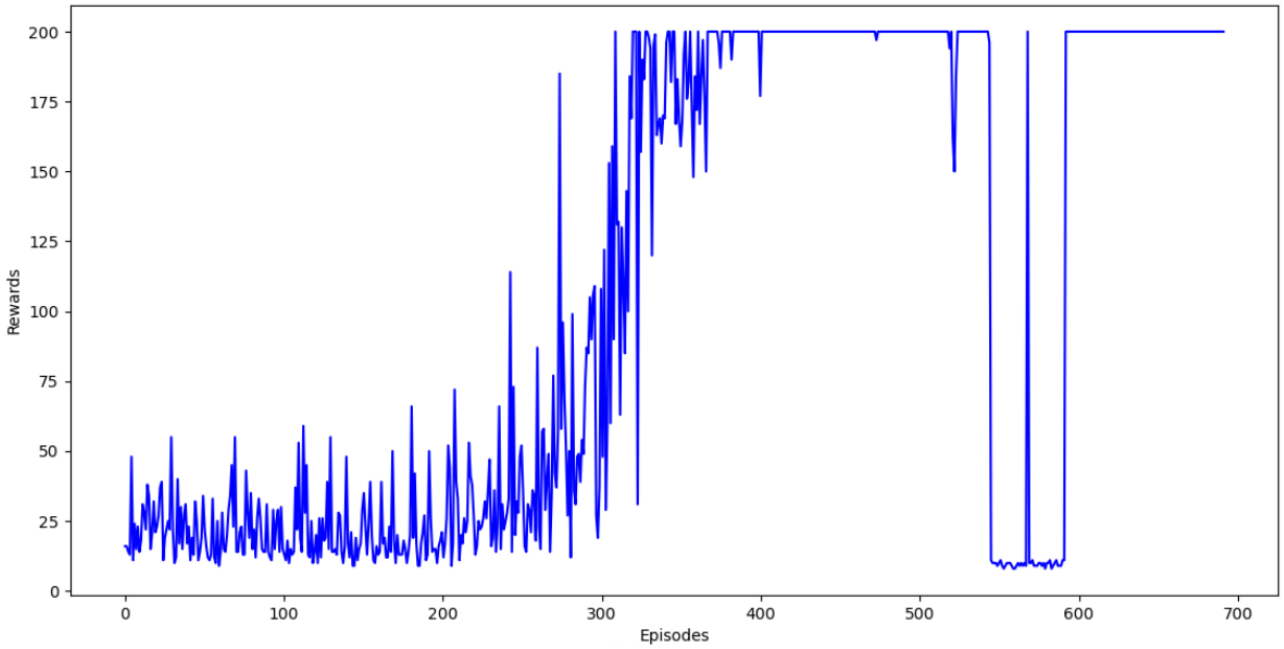

Figure 4: Rewards vs Episodes for the "Cartploe" task during training without target network

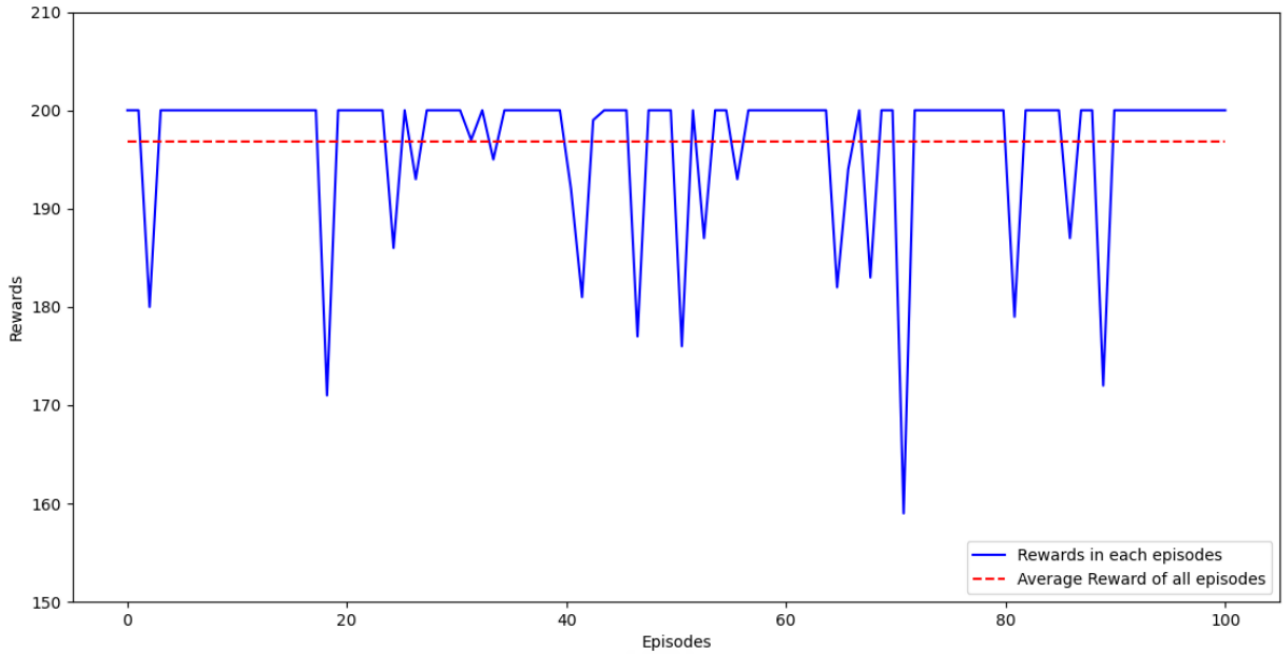

Figure 5: Rewards vs Episodes for the "Cartploe" task during play policy without target network

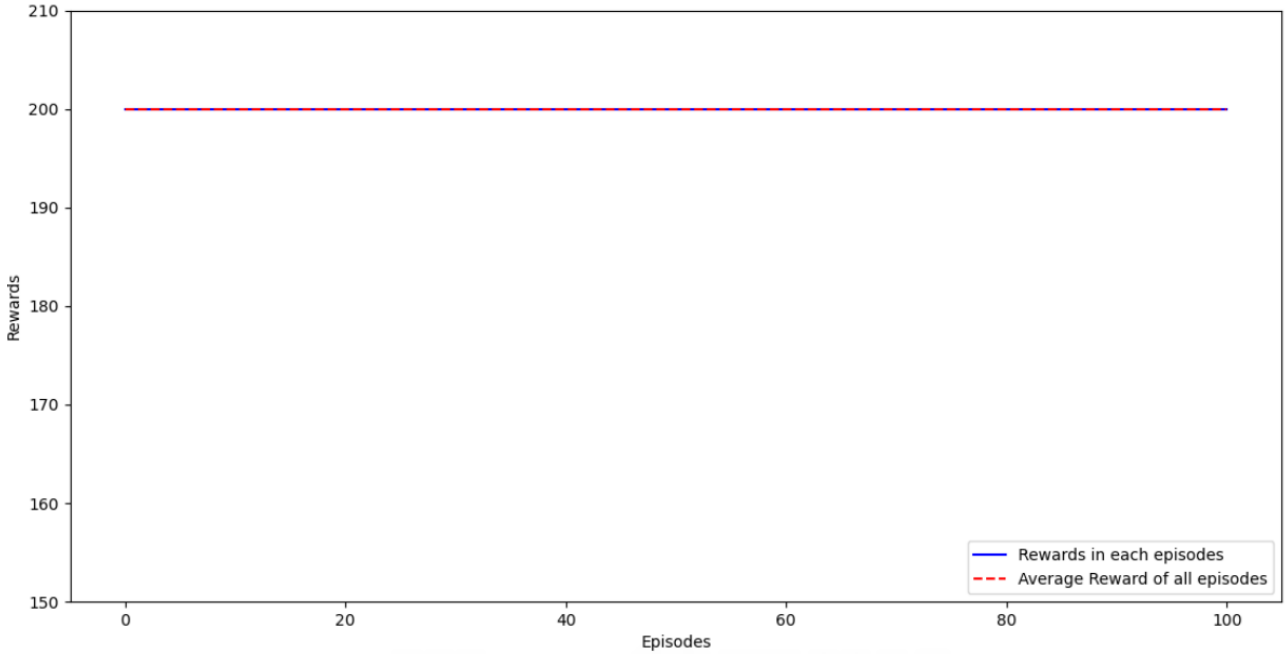

Figure 6: Rewards vs Episodes for the "Cartploe" task during play policy without target network

(b)Without Experience Replay but with Target Network:

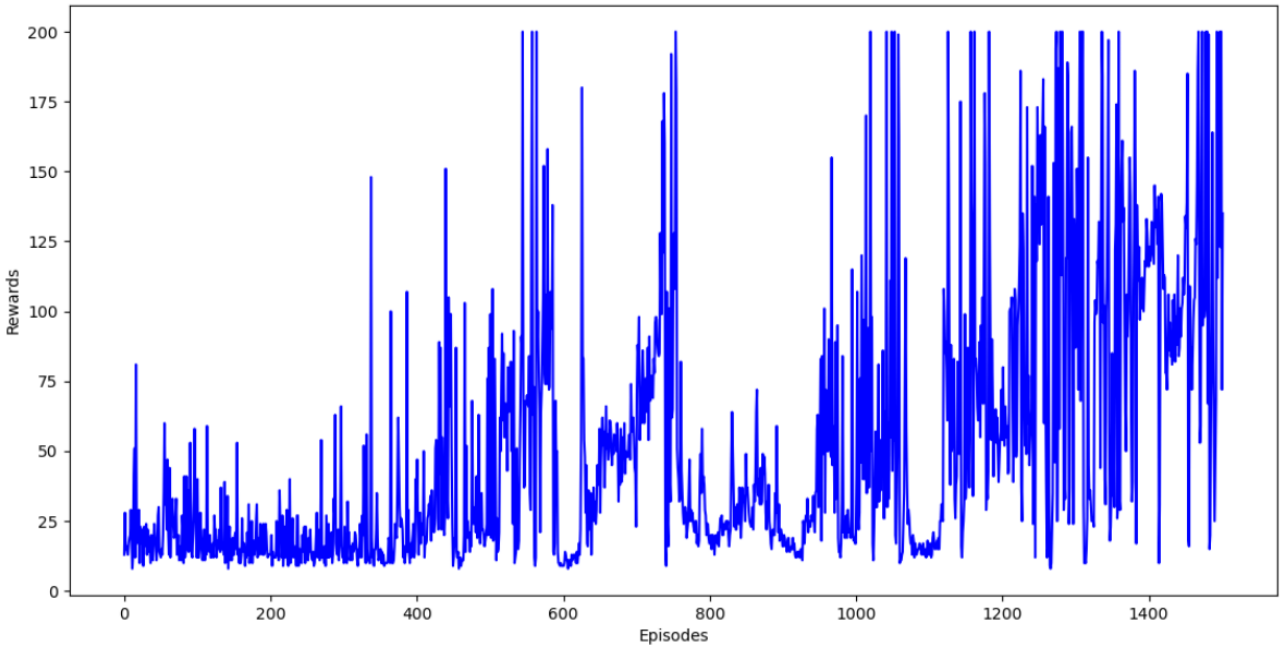

Figure 7: Rewards vs Episodes for the "Cartploe" task during training without experience replay

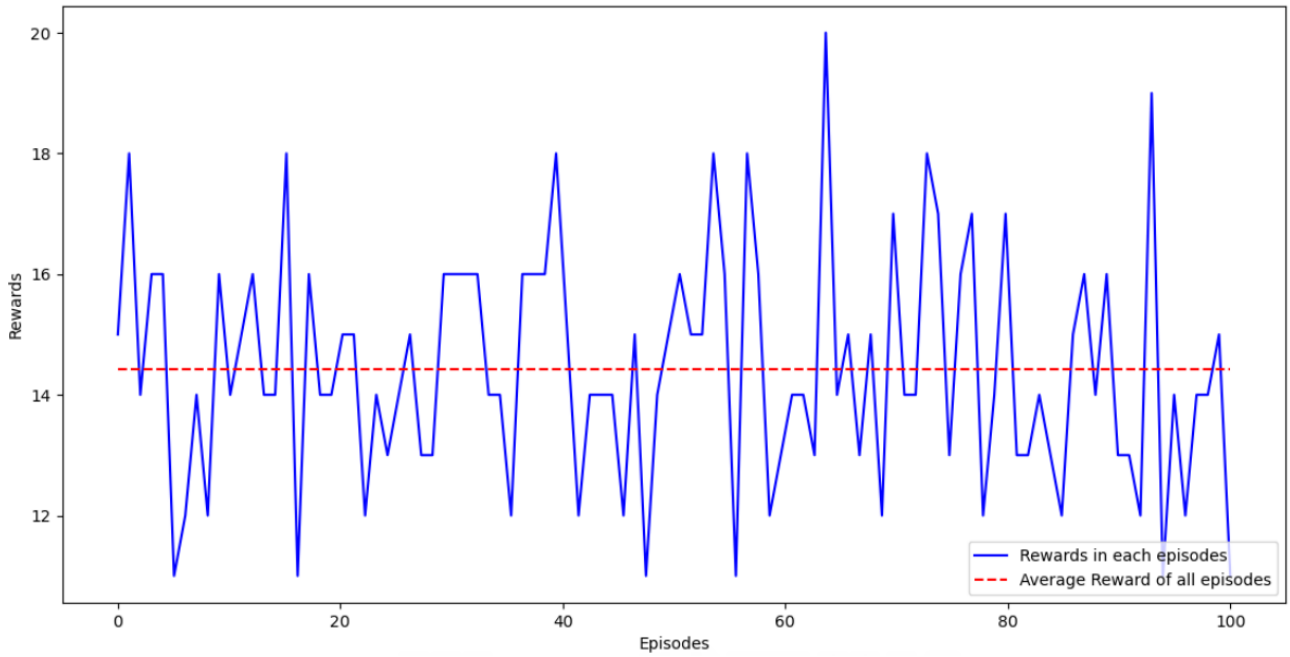

Figure 8: Rewards vs Episodes for the "Cartploe" task during play policy without experience replay

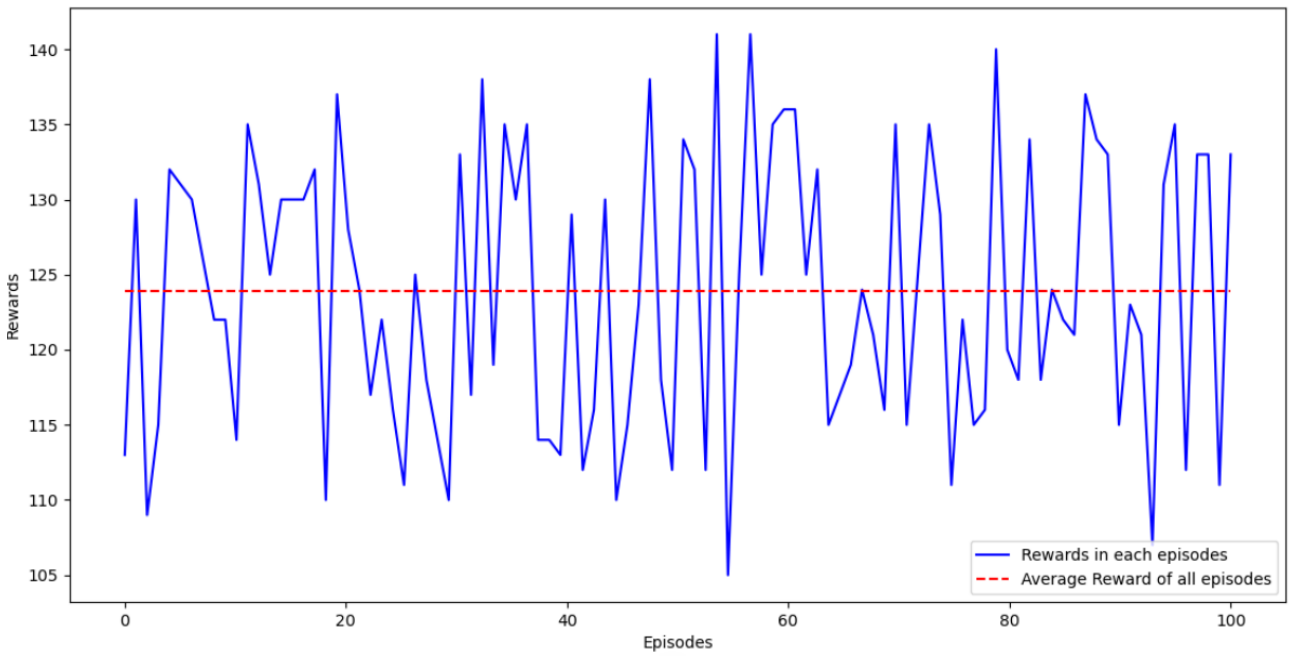

Figure 9: Rewards vs Episodes for the "Cartploe" task during play policy without experience replay

(c)Without Target Network and Experience Replay:

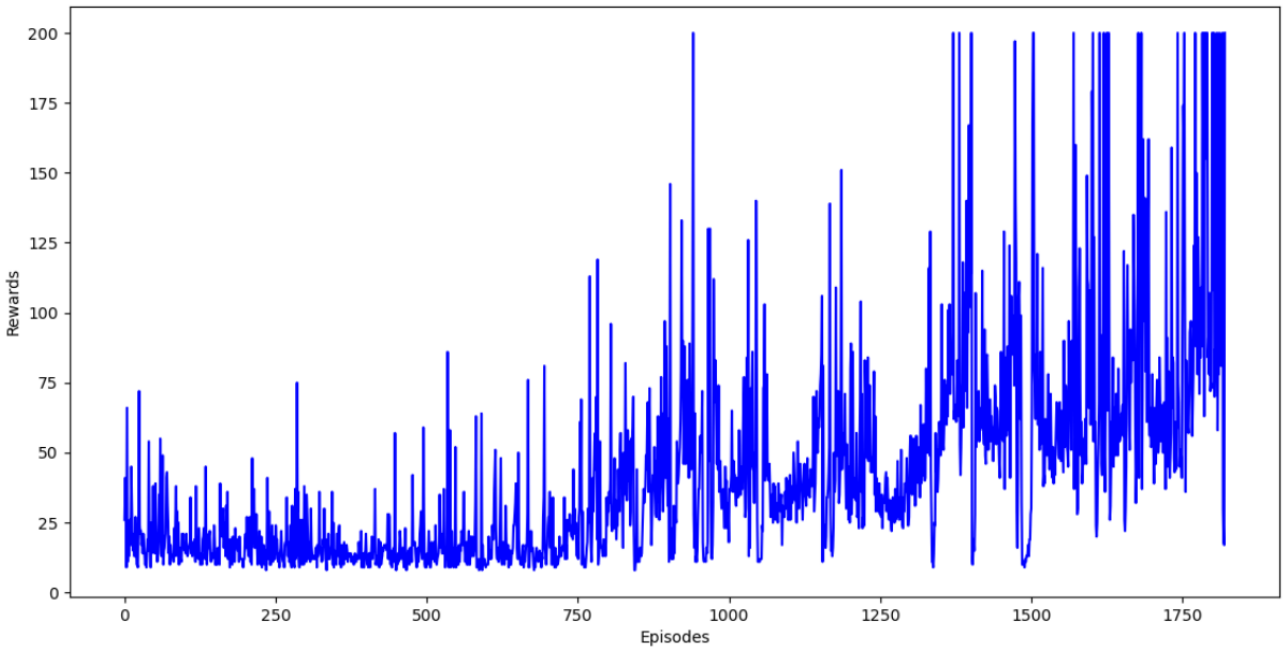

Figure 10: Rewards vs Episodes for the "Cartploe" task during training without experience replay and target network

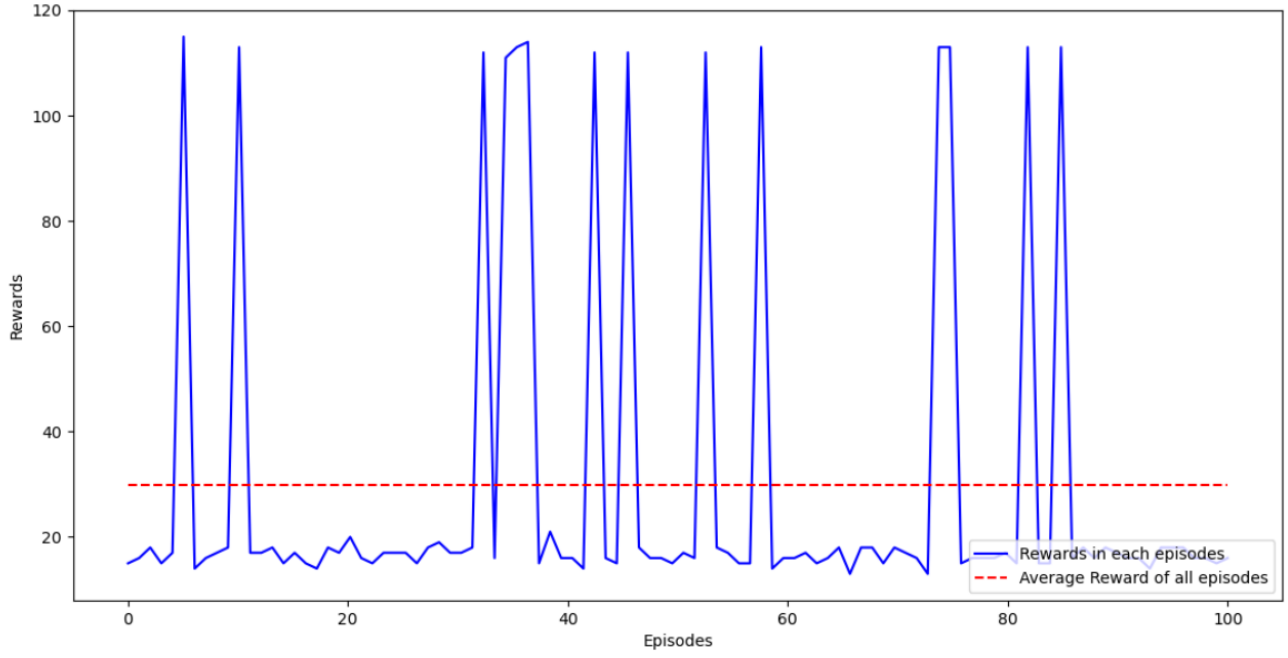

Figure 11: Rewards vs Episodes for the "Cartploe" task during play policy without experience replay and target network

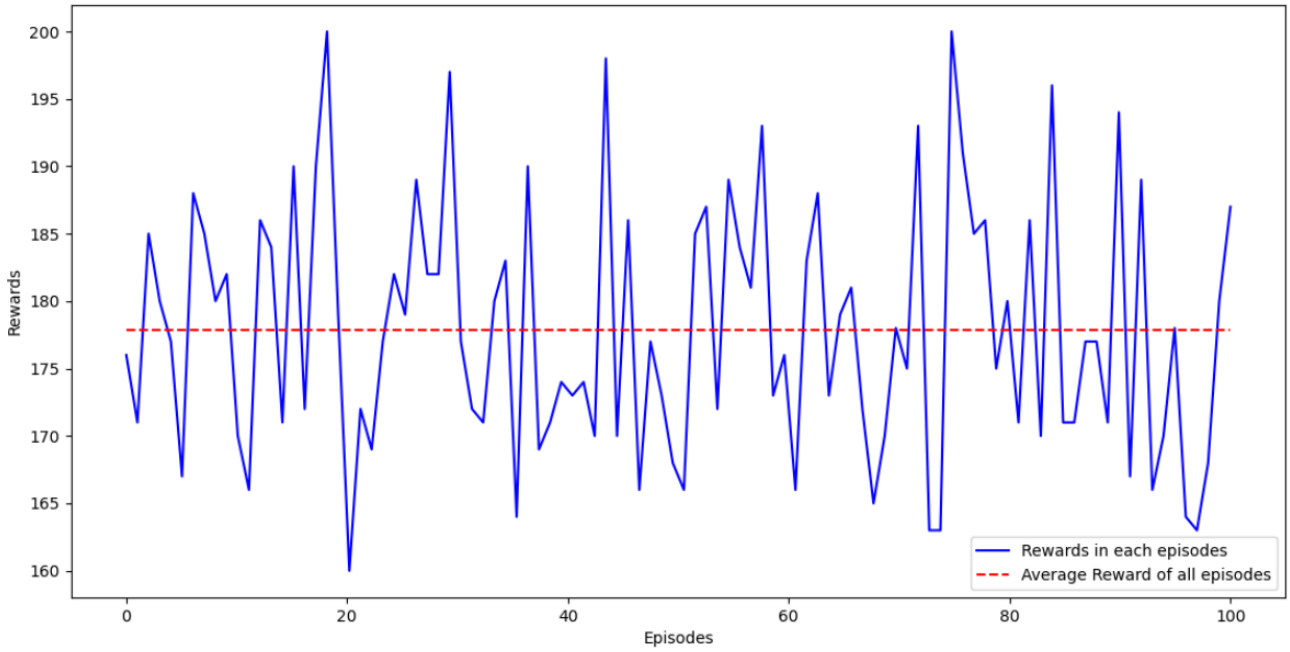

Figure 12: Rewards vs Episodes for the "Cartploe" task during play policy without experience replay and target network

2.1 Observations (for the bonus question)

(a) **Without Target Network but with Experience Replay:**
In the case where only target network is removed, we observe very less deterioration of performance. As can be observed from figure 22-24, we still get average reward of 200, but less often as compared to DQN case. Though the average reward is still greater than 195, we see lot of oscillation in reward values in figure 23, and in some episodes the reward gets very low value.

(b) **Without Experience Replay but with Target Network:**
If we remove the experience replay but keep the target network, we see significant degradation of performance during training as in figure 25 and also during play policy as shown in figures 26-27. Now we can see very low average reward and significant amount of oscillation in rewards from episodes to episodes.

(c) **Without Target Network and Experience Replay:**
Out of all the cases, in this case the performance is the worst. From figures 28-30, we notice very low average reward and high amount of oscillation in rewards from episodes to episodes. During training also there is high amount of oscillation in the rewards.