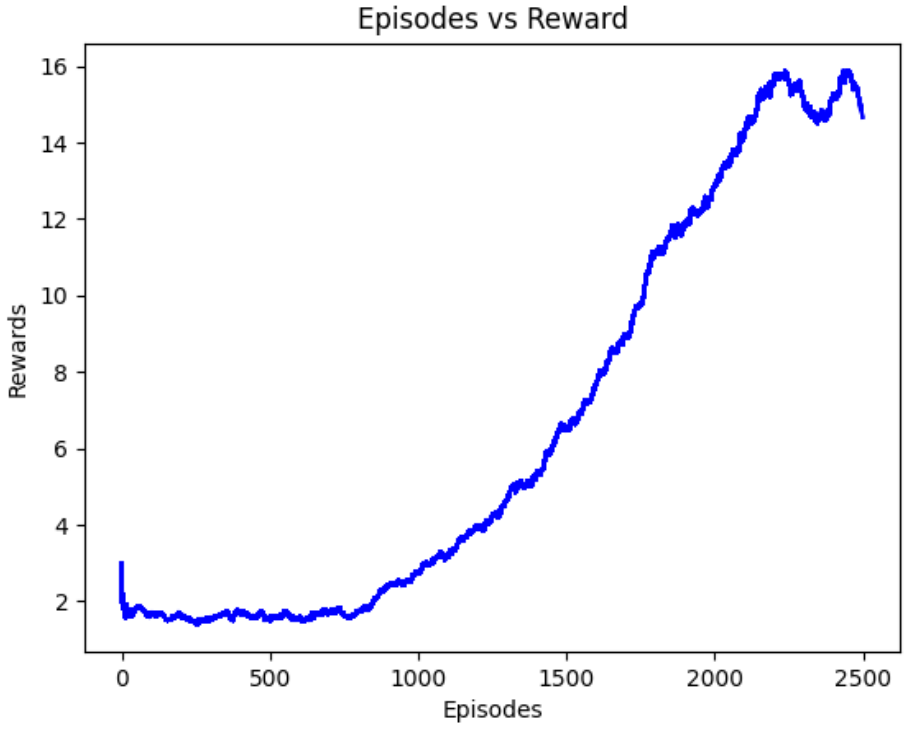
Name(s): Neha Jain, Pallaw Kumar
Netid(s): nehaj4, pllawk2

Mean Reward Reached using DQN and DDQN:
DQN: 6.96
DDQN: 14.65

Uploaded Saved DQN/DDQN Model on Canvas (whichever performs better):
Yes, we uploaded the DDQN model on canvas

Uploaded your Agent.py and Agent_double.py files on Canvas :
Yes, we uploaded Agent.py and Agent_double.py files on Canvas

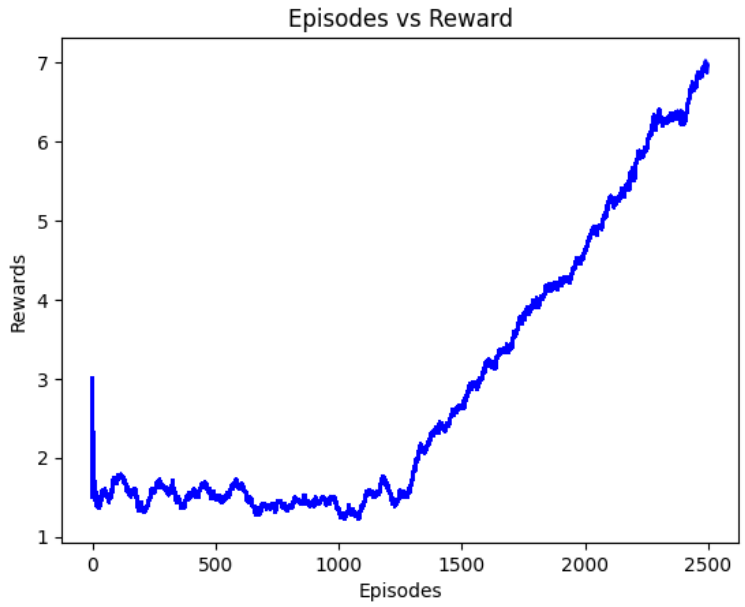
Plot of Mean Evaluation Reward for the model that reaches the target score (Either DQN or DDQN):
Plot of DDQN



The highest evaluation reward that was reached was 14.65 at episode 2499. The learning rate was 1.6384 e-06 and the memory length was 892262.

Provide a few sentences to analyze the training process and talk about some implementation details:

Plot fof DQN:



Implementation details: To maximize the cumulative reward of taking actions in a given state in DQN, we create a Q-function and utilize it to minimize the Huber loss for updating the target network. Double DQN (DDQN) is an extension of DQN that reduce overestimations in the Q function, which can lead to faster and more stable learning compared to DQN. DDQN decouples the selection and evaluation of actions, so one network is used to select actions(policy net) and another network is used to evaluate the selected actions(target net).

Training Process:(We have trained the model on the default configuration as provided) One change was made to the memory.py file in the class definition for a replay memory buffer. The action tensor is first converted to CPU before being appended to the buffer to avoid any GPU-CPU memory conflicts. If the action tensor is already on the CPU, the except block will be executed and the original action tensor will be appended to the buffer.

DDQN model was successful in achieving a score of 10 after approx. 1800 episodes whereas even though DQN was able to achieve only 6.96 as the highest score at 2500 episodes(learning was very slow and saturated for the first 1200 episodes). This result tells that Double DQN has better than DQN.