Assignment Deep Q-Learning for Cartpole Balancing

The Deep Q-learning algorithm is marked as one of the most famous algorithms for deep reinforcement learning. In this assignment, you are going to complete the core functions in the attached python files for DQN algorithm. Specifically, you represent the control policy by a fully-connected neural network which outputs Q-value for each state-action pair. The cartpole environment is a standard gym environment(https://gym.openai.com/envs/CartPole-v0/) where a "pole is attached by an un-actuated joint to a cart by moving along a frictionless track. The system is controlled by applying a force of +1(moving right) or -1(moving left) to the cart. The pendulum starts upright, and the goal is to prevent it from falling over. A reward is +1 is provided for every timestep that pole remains upright and -100 when falling". The cartpole system has its state as 4-dimensional vector including cart position, cart velocity, pole angle, and pole angular velocity. More information can be found at https://github.com/openai/gym/blob/master/gym/envs/classic_control/cartpole.py.

The DQN algorithm essentially implements the Q-learning algorithm with neural network representation. You need to complete **replay()** function for Q-learning and **act()** function for epsilon-greedy exploration. The explored data including states, actions, rewards is stored in memory and is used by the **replay()** function to update the policy. For more details of DQN, please see the references

1. https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html.
2. https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf
3. https://web.stanford.edu/class/psych209/Readings/MnihEtAlHassibis15NatureControlDeepRL.pdf

Some tips:

To set up the python environment,

You can create a new python environment by

"conda create -n dqn python=3.7"

"conda activate dqn"

"pip install tensorflow==2.8.0"

"pip install gym"

"pip install pygame"

Or you can use the IDEs you are comfortable with.

Submission:

please submit your code with a plot of total reward for each episode for multiple runs. You may need to use matplotlib package for plotting.