ML HW12 Report

物理三 潘阜承 B08202036

1. Implement Advanced RL algorithm.
   1. For this homework, I chose Deep Q Network (DQN) as my RL Algorithm.
   2. In policy gradient method, it tries to learn a policy function directly, which means it aims to directly optimize in the action space. On the other hand, DQN belongs to another class of RL called Q learning. Q learning is a type of value iteration method aims to estimate the Q function. Here, Q function is simply an action-value function that maps state-action pairs to their expected discounted reward, and finding the optimal policy would be simply choosing the action that maximizes the Q function at every step. Therefore, the main difference between policy gradient and DQN is that the goal of DQN is to train a network to approximate Q function and use it to infer the optimal policy while policy gradient aims to optimize the policy space directly.
   3. There are several steps to implement DQN. First of all, I created a network which takes the state as input and output action values. Note that this network is also used as the target network which would produce the target. Then, I created a class called “ReplayBuffer” which aims to store the previous experiences including state, action, reward, next state and done in order to perform Q learning. Furthermore, I created a class called “Agent”. There are several functions in Agent. First, I defined a function called “step”. This function will first add the current information of present step such as state, action, reward, etc., into the ReplayBuffer. Then update agent every fixed time steps. There is also a function called “act” which will produce the action of current state. Note that this function also takes a parameter called “epsilon” as input. Epsilon is introduced in order to avoid the circumstance that some of the actions might never occurred before. Thus, I used epsilon as a probability threshold to whether generate random action or not. Then, there is a “learn” function. In this function, first take sample experiences from ReplayBuffer as input and a hyperparameter gamma for target function. Then, use the previously defined network to produce target and expected value and calculate the loss between them using “smooth\_L1\_loss”. Next, perform gradient descent step and update models’ parameters using soft update. Eventually, we can start the training procedure.
2. Below are descriptions about MuZero, **which one is not correct**?