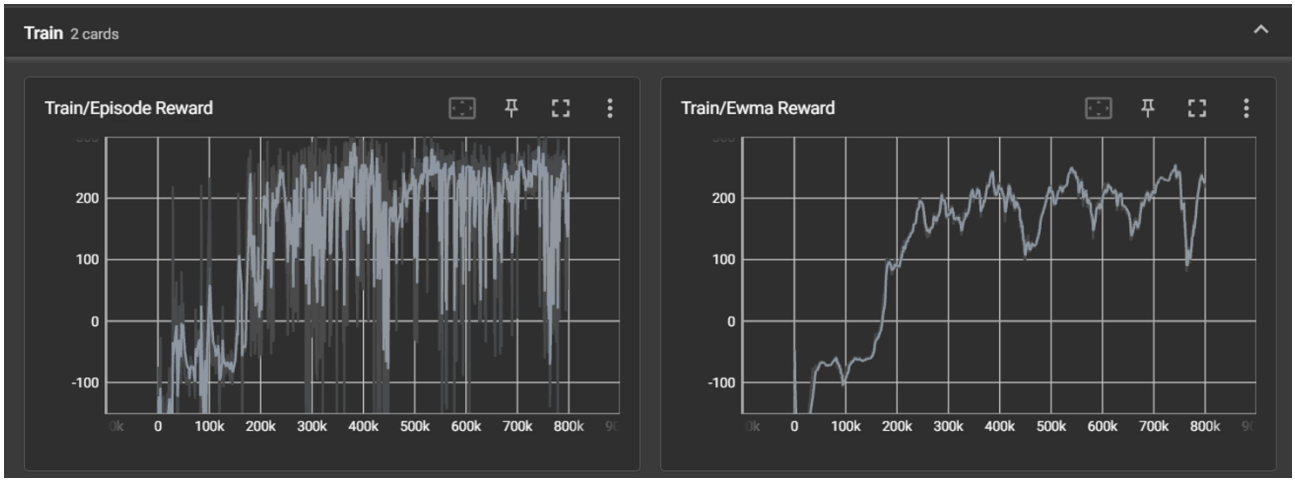
Experiment Report

311552024 詹偉翔

1. Result

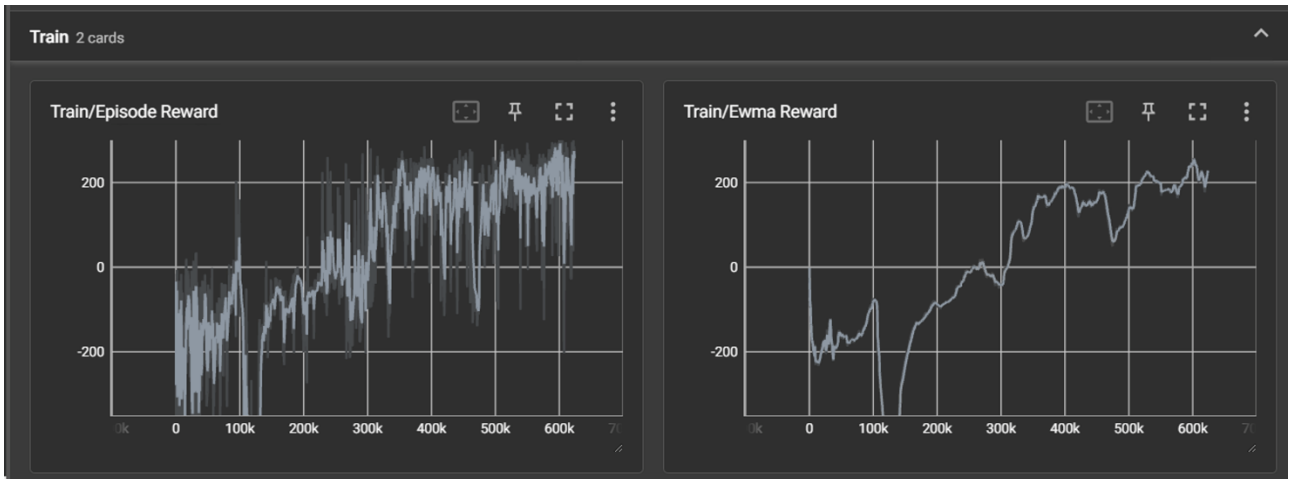
- LunarLanders – DQN

```
(DLP) C:\Users\weishon\Desktop\深度學習\lab06>python dqn.py --test_only
C:\Users\weishon\anaconda3\envs\DLP\lib\site-packages\gym\logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
Start Testing
episode: 0: 243.61
episode: 1: 278.00
episode: 2: 257.82
episode: 3: 268.22
episode: 4: 31.58
episode: 5: 257.70
episode: 6: 297.02
episode: 7: 293.87
episode: 8: 65.21
episode: 9: 279.00
Average Reward 227.20256109254856
```



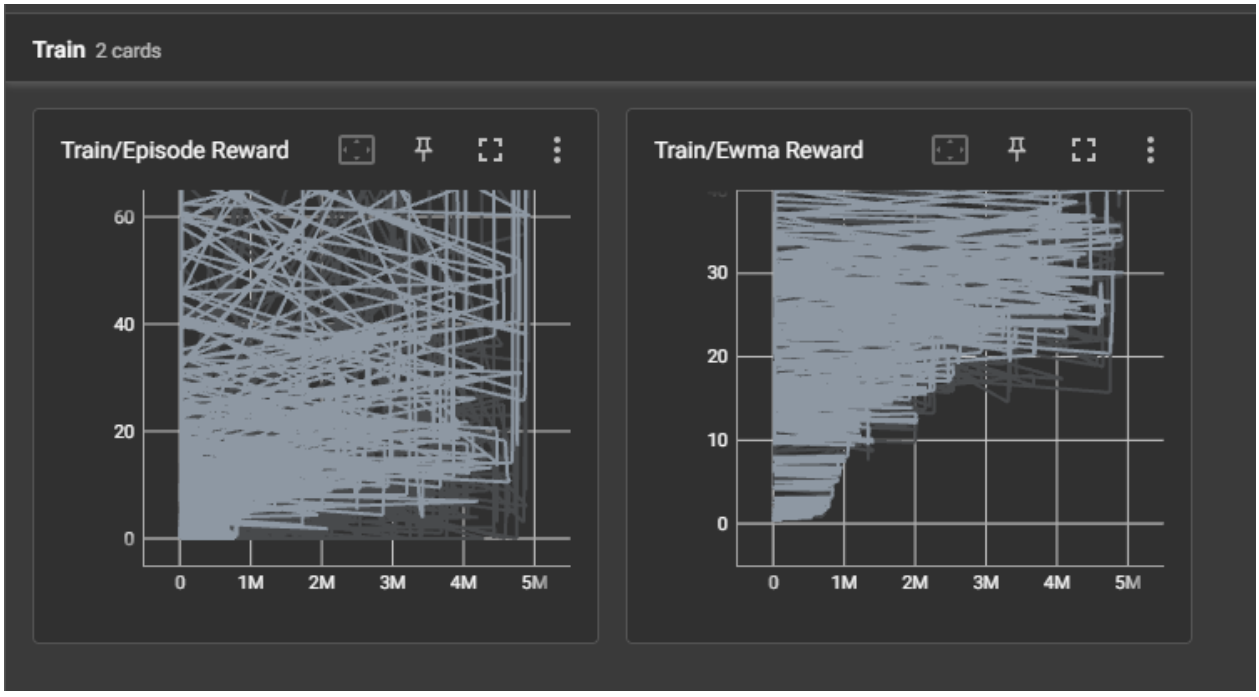
- LunarLanders – DDPG

```
(DLP) C:\Users\weishon\Desktop\DeepLearning\lab06>python ddpg.py --test_only
C:\Users\weishon\anaconda3\envs\DLP\lib\site-packages\gym\logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
Start Testing
episode: 0: 248.44
episode: 1: 262.90
episode: 2: 283.98
episode: 3: 266.65
episode: 4: 13.19
episode: 5: 275.82
episode: 6: 207.32
episode: 7: 238.63
episode: 8: 249.20
episode: 9: 268.84
Average Reward 231.49877403398168
```



- BreakoutNoFramskip – DQN

```
(DLP) C:\Users\weishon\Desktop\DeepLearning\lab06>python dqn_breakout.py --test_only
Start Testing
episode 5: 388.00
episode 5: 396.00
episode 5: 423.00
episode 5: 402.00
episode 5: 406.00
episode 5: 416.00
episode 5: 187.00
episode 5: 402.00
episode 5: 347.00
episode 5: 407.00
Average Reward: 377.40
```



2. Discussions

- Describe implementation of DQN, DDPG

  i. Implementation of Q network updating in DQN

A: I use the q_value and q_target to calculate the loss that network should be updated. Q_value is from behavior network. Q_target is from target network multiplied with gamma and adding reward.

ii. Implementation and the gradient of actor updating in DDPG

A: It use the minus critic_net with the sate and action to update the network.

iii. Implementation and the gradient of critic updating in DDPG

A: It is same as the Implement of Q network updating in DQN

- Explain benefits of epsilon-greedy in comparison to action selection

A: Because we need to let the model explore the action as many as possible. After we move a lot of actions, we can choose the action from the previous memory.

- Explain the necessity of the target network

A: We need have the target so that we can update the behavior network, and then we can use the best to become target. And for loop that action, we can get the best model.

- Describe the tricks you used in Breakout and their effects, and how they differ from those used in LunarLander

A: Since breakout's state is a picture, so I stack the 4 pictures to become the state. Only this way can let the model know where is the ball. I need

to deal with a lot of data transport, permute, and so on because of above trick I used.