Lab5: Deep Q-Network and Deep Deterministic Policy Gradient

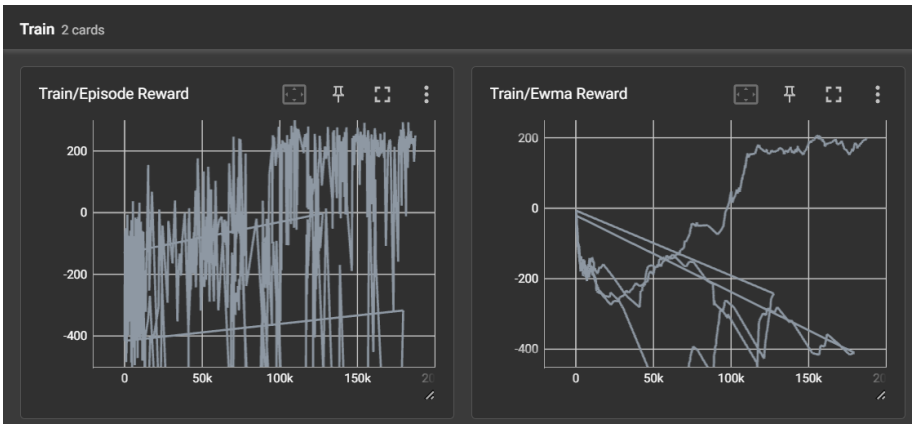
系級:智能系統　學號:312581006　姓名:張宸瑋

1. Experimental Results

A. LunarLander-v2 using deep Q-network DQN

```
PS C:\Users\Steven\Desktop\課程資料\交大\2023DL\Lab\Lab5>  c:; cd 'c:\Users\Steven\Desktop\課程資料\交大\2023DL\
Lab\Lab5'; & 'C:\Users\Steven\AppData\Local\Programs\Python\Python38\python.exe' 'c:\Users\Steven\.vscode\extens
ions\ms-python.python-2023.14.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '56328' '--' 'c:\
Users\Steven\Desktop\課程資料\交大\2023DL\Lab\Lab5\dqn-example.py'
Start Testing
total reward : 216.42117804076997
total reward : 137.86733653250818
total reward : 190.91502459672472
total reward : 236.72293080165664
total reward : 226.33417034095805
total reward : 216.62040914411034
total reward : 225.09228058677695
total reward : 239.00837931773842
total reward : 207.88074499163372
total reward : 213.98828180363165
Average Reward 211.08507361565086
```

Testing Results

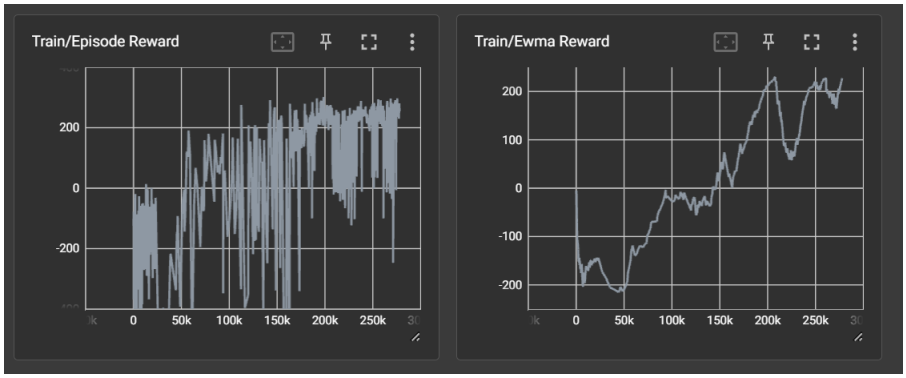


Tensorboard

B. LunarLanderContinuous-v2 DDPG

```
PS C:\Users\Steven\Desktop\課程資料\交大\2023DL\Lab\Lab5>  c:; cd 'c:\Users\Steven\Desktop\課程資料\交大\2023DL\
Lab\Lab5'; & 'C:\Users\Steven\AppData\Local\Programs\Python\Python38\python.exe' 'c:\Users\Steven\.vscode\extens
ions\ms-python.python-2023.14.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '56184' '--' 'c:\
Users\Steven\Desktop\課程資料\交大\2023DL\Lab\Lab5\ddpg-example.py'
Start Testing
total reward : 219.0657732381357
total reward : 196.46912643782832
total reward : 203.7484807977554
total reward : 236.72293080165664
total reward : 225.14383885196978
total reward : 216.3733320664905
total reward : 225.09228058677695
total reward : 238.9707832715679
total reward : 205.66412473090293
total reward : 213.98828180363165
Average Reward 218.12389525867155
```

Testing Results

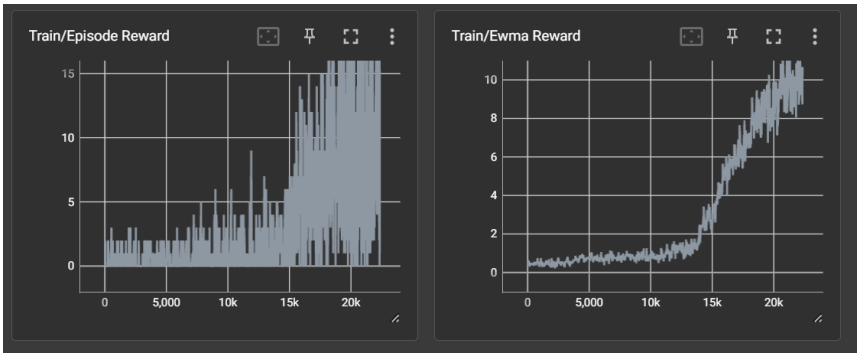


Tensorboard

C. BreakoutNoFrameskip-v4 using deep Q-network (DQN)

```
PS C:\Users\Steven\Desktop\課程資料\交大\2023DL\Lab\Lab5>  c:; cd 'c:\Users\Steven\Desktop\課程資料\交大\2023DL\
Lab\Lab5'; & 'C:\Users\Steven\anaconda3\envs\DLP\python.exe' 'c:\Users\Steven\.vscode\extensions\ms-python.pytho
n-2023.14.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '58537' '--' 'c:\Users\Steven\Desktop
\課程資料\交大\2023DL\Lab\Lab5\dqn_breakout_example.py'
Start Testing
episode 1: 299.00
episode 2: 226.00
episode 3: 331.00
episode 4: 226.00
episode 5: 334.00
episode 6: 401.00
episode 7: 173.00
episode 8: 226.00
episode 9: 279.00
episode 10: 226.00
Average Reward: 272.10
```

Testing Results



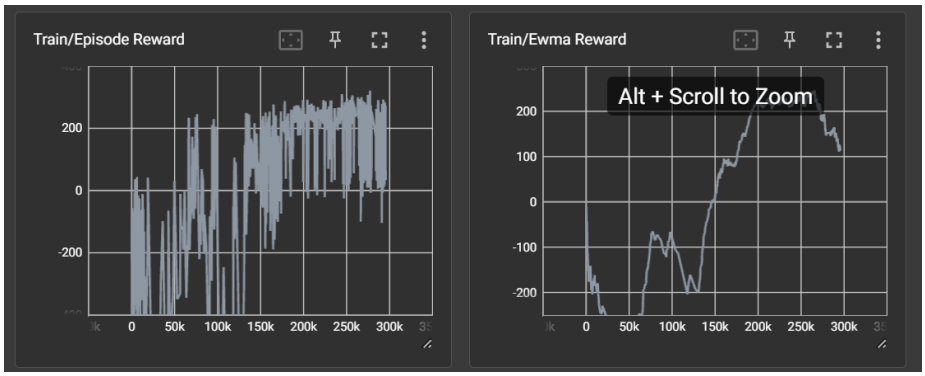
Tensorboard

2. Experimental Results of bonus parts (DDQN, TD3) (15%)

LunarLander-v2 using deep Q-network DDQN

```
PS C:\Users\Steven\Desktop\課程資料\交大\2023DL\Lab\Lab5>  c:; cd 'c:\Users\Steven\Desktop\課程資料\交大\2023DL
Lab\Lab5'; & 'C:\Users\Steven\AppData\Local\Programs\Python\Python38\python.exe' 'c:\Users\Steven\.vscode\exte
ions\ms-python.python-2023.14.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '58276' '--' 'c
Users\Steven\Desktop\課程資料\交大\2023DL\Lab\Lab5\ddqn.py'
Start Testing
total reward : 243.76016493070992
total reward : 308.7624165347863
total reward : -6.2795019129696215
total reward : 253.3679769004798
total reward : 86.5327243055447
total reward : 218.8864389983754
total reward : 217.1325468240284
total reward : 250.9847696909281
total reward : 238.1399776467084
total reward : 246.31374549595046
Average Reward 205.7601259414542
```

Testing Results



Tensorboard

3. Questions

A. Describe your major implementation of both DQN and DDPG in detail. Your description should at least contain three parts

(1) Your implementation of Q network updating in DQN.

在DQN當中，我使用了Experience Replay跟Fixed Target Network，而Experience Replay的實現，我使用了一個緩衝區，用來儲存環境代理，在與環境交互中觀察的狀態，透過這個緩衝區，我會隨機抽樣一批經驗，透過這樣的操作來減少相關性，而透過Fixed Target Network來提高穩定性，當前網路用於計算Q值，而固定目標網路，來用於計算目標Q值，並且定期更新，將當前網路的參數複製到固定目標網路。

```python
def _update_target_network(self):
    '''update target network by copying from behavior network'''
    ## TODO ##
    self._target_net.load_state_dict(self._behavior_net.state_dict())
```

更新參數實現

(2) Your implementation and the gradient of actor updating in DDPG.

我們會根據當前狀態，使用演員網路生成一個行動，透過這個行動，計算演員網路參數的梯度，這個梯度表示如何調整演員網路的參數，並且透過行動梯度來更新演員網路的參數，以最大化期望獎勵。

```
## update actor ##
## TODO ##
action = self._actor_net(state)
actor_loss = -self._critic_net(state, action).mean()
# optimize actor
actor_net.zero_grad()
critic_net.zero_grad()
actor_loss.backward()
actor_opt.step()
```

更新網路參數實現

(3) Your implementation and the gradient of critic updating in DDPG.

評論家網路用於估計狀態-行動對的Q值，以指導演員網路的訓練，主要的更新步驟如下，第一，計算Q值誤差，第二，計算梯度，第三，更新評論家網路。

```
## update critic ##
## TODO ##
state = state.to(torch.float32)
action = action.to(torch.float32)
next_state = next_state.to(torch.float32)
q_value = self._critic_net(state, action)
with torch.no_grad():
    a_next = self._target_actor_net(next_state)
    q_next = self._target_critic_net(next_state, a_next)
    q_target =  reward + gamma * q_next * (1 - done)
criterion = nn.MSELoss()
q_value = q_value.to(torch.float32)
q_target = q_target.to(torch.float32)
critic_loss = criterion(q_value, q_target)
# optimize critic
actor_net.zero_grad()
critic_net.zero_grad()
critic_loss.backward()
critic_opt.step()
```

更新網路參數實現

B. Explain effects of the discount factor

γ 是一個強化學習很重要的參數，它影響著代理學習的和策略的行為，而較低的 γ 則更關注即時獎勵。這對於探索與利用的權衡具有關鍵性影響，較高的 γ 鼓勵探索，而較低的 γ 可能導致更保守的策略。γ 還影響學習的收斂速度，較高的 γ 可能需要更長的時間來收斂，因為代理需要考慮複雜的長期影響。折扣因子的選擇應根據具體的任務和環境來調整，不同的應用場景可能需要不同的 γ 值。

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

epsilon-greedy 策略在強化學習中的優勢在於它的探索能力和適應性，面對不確定性或噪聲的情況下，表現更穩定它可以幫助代理更好地學習和適應不同的環境，同時避免陷入次優解中。

D. Explain the necessity of the target network

目標網路的作用主要是在穩定性、波動的減少，以及處理估算偏差上。目標網路的存在使得訓練更為穩定，有助於更快地達到良好的策略，使智能體能夠更好地處理複雜的環境和學習任務。

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

在這次的實作上，我們使用了在講義中提到的 frame sracking，透過將環境的幀堆疊再一起，我們能夠捕捉動態訊息，減少過去訊息的丟失，以及減少噪音，而在 Breakout 方面，frame sracking 可以幫助代理更好的捕捉球的運動，以及彈跳版的移動，連續的幀能夠提供關於球的速度，和方向的訊息，也能夠幫助代理更容易了解磚塊的位置跟分數情況，而相對於 Breakout 來說 LunarLander，比較不需要太多的 frame sracking 因為遊戲的動作相對緩慢，而且幀之間的關聯性相對較高。因此，在這種情況下，通常可以使用較小的幀堆疊窗口。