10-703 Deep RL and Controls
Homework 3

April 23, 2017

Problem 1

LQR

Answer:
In this problem, the infinite discrete LQR is implemented.

1. LQR on TWoLinkArm-v0 has the total reward -417.57, steps to reach the goal 411. The q, $\dot{q}$ and u plots are in the Figure 1 and 2

Figure 1: LQR on TwoLinkArm-v0: state trajectory

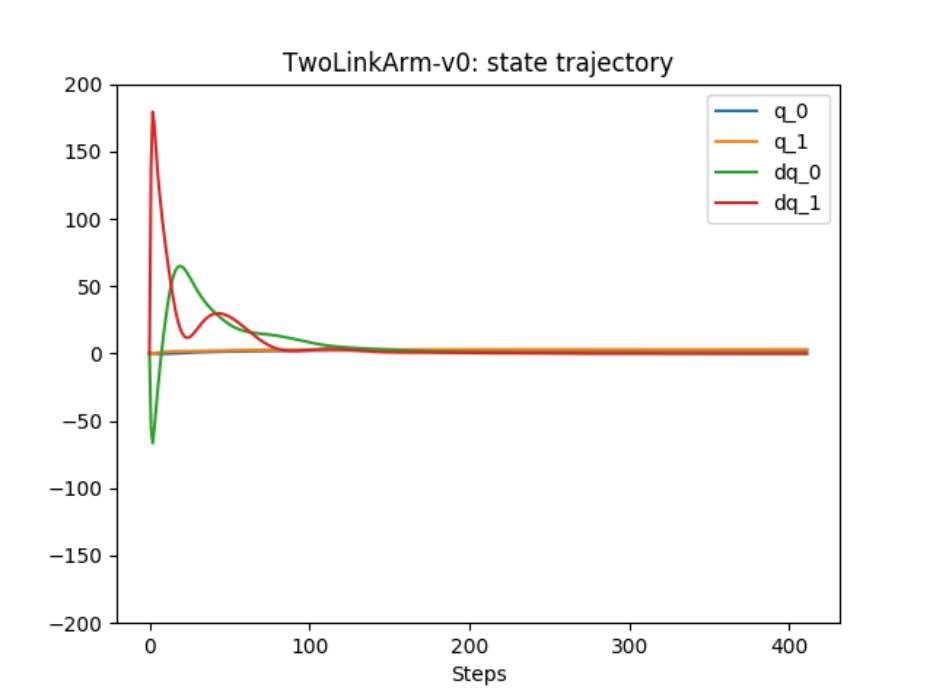
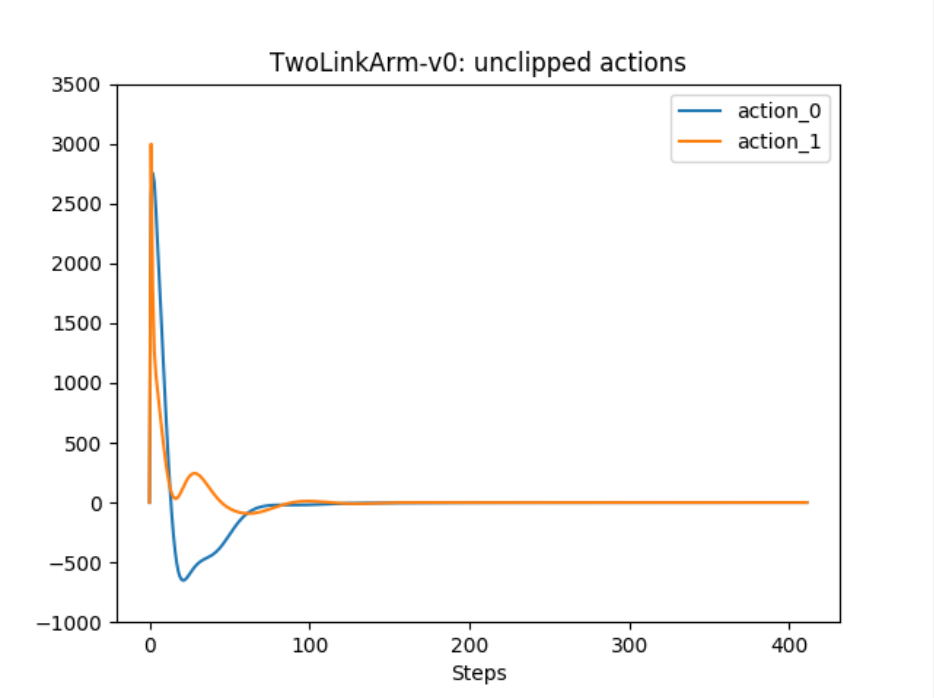


Figure 2: LQR on TwoLinkArm-v0: raw action trajectory



2. LQR on TWoLinkArm-limited-torque-v0 has the total reward -4288.58, steps to reach the goal 1836. The q, $\dot{q}$, u and the clipped u plots are in the Figure 3, 4, 5.

Figure 3: LQR on TwoLinkArm-limited-torque-v0: state trajectory

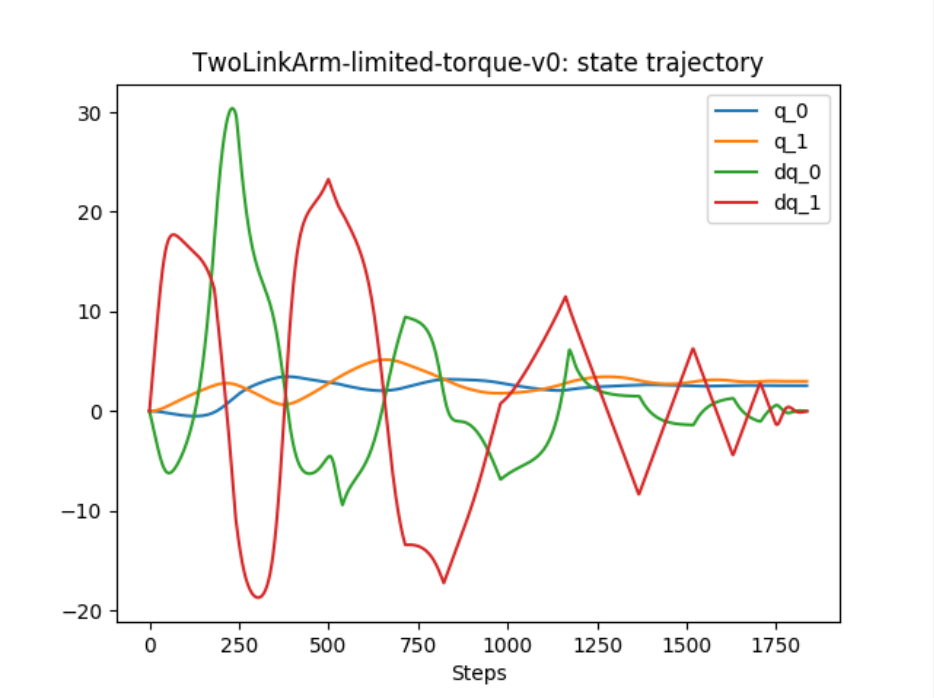


Figure 4: LQR on TwoLinkArm-limited-torque-v0: raw action trajectory

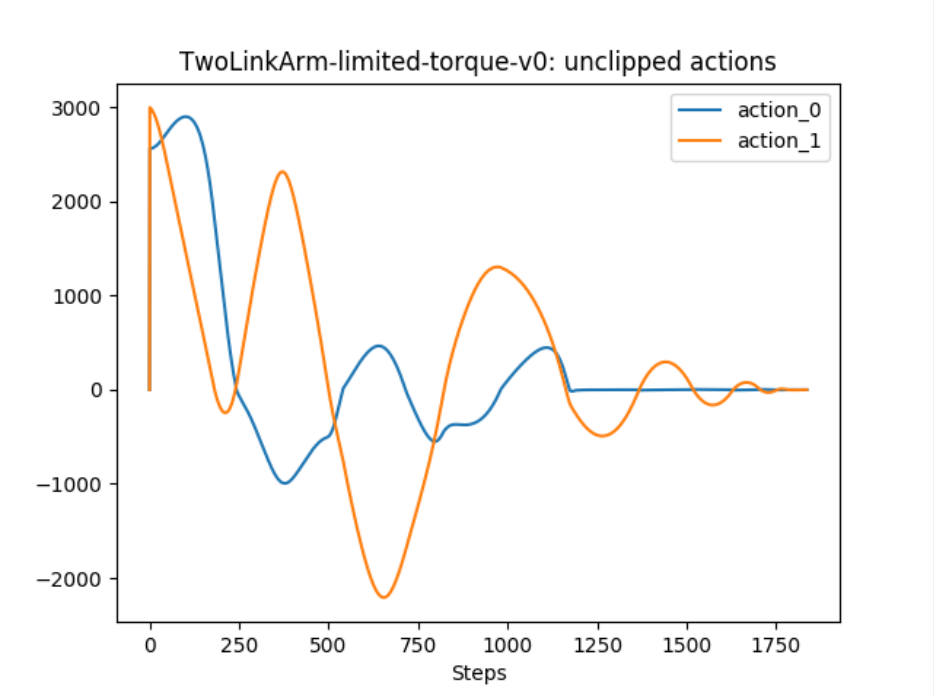
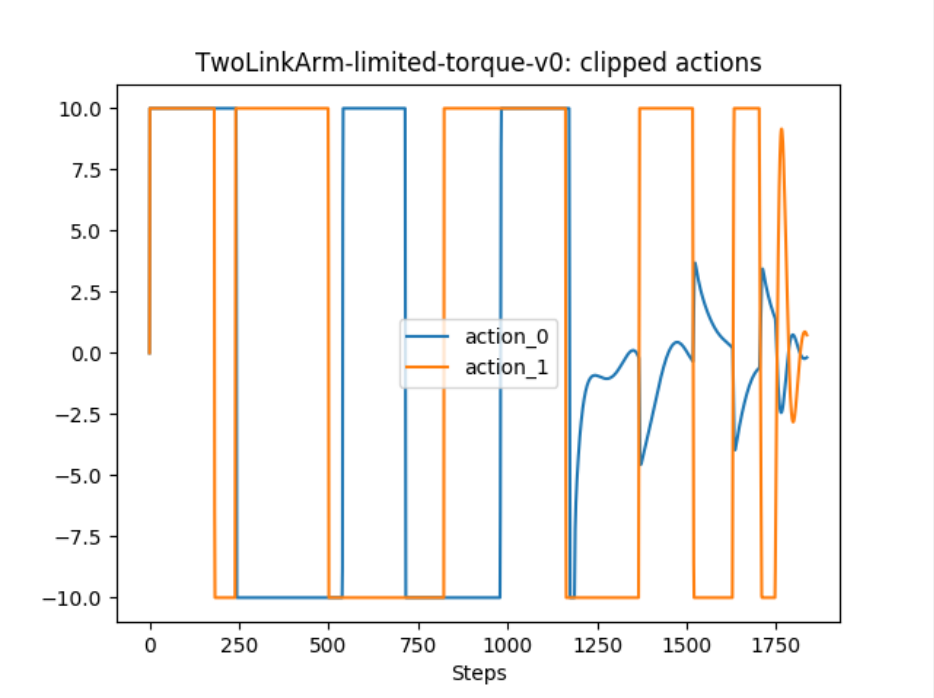


Figure 5: LQR on TwoLinkArm-limited-torque-v0: clipped action trajectory



3. Under the TwoLinkArm-v0 environment where the torque is unlimited, the lqr controller can soon make the arm reach the goal state because unlimited torque can be forced to the arm. As shown in the Figure 2, the torque can reach to 3000 for the TwoLinkArm-v0 environment. As the results, the $\dot{q}$ can be more than 150, as shown in the Figure 1. It may be hard to see in the Figure 1, the q reaches the goal smoothly with little oscillations.

   Under the TwoLinkArm-limited-torque-v0 environment where the torque is limited to 10 and -10, the LQR controller needs much longer time to make the arms reach the goal state. However, because the torque is limited, the $\dot{q}$ is always small, as shown in Figure 3. Compared to the TwoLineArm-v0, there are obvious oscillations for the q before it converges to the goal, as shown in the Figure 3.

4. LQR on TWoLinkArm-v1 has the total reward -2405.65, steps to reach the goal 1844. The q, $\dot{q}$ and u plots are in the Figure 6 and 7

Figure 6: LQR on TwoLinkArm-v1: state trajectory

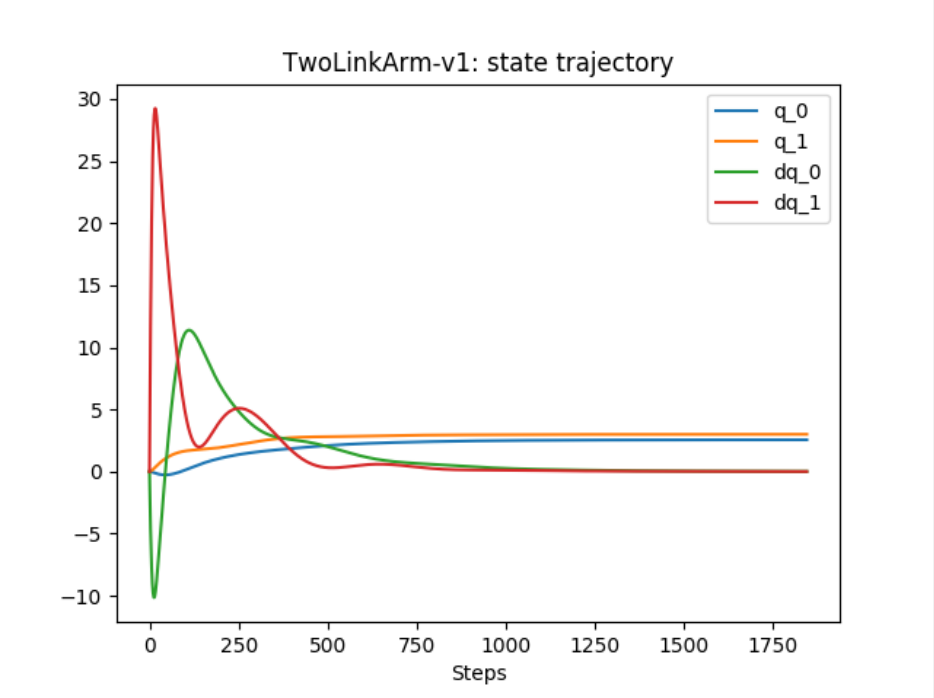
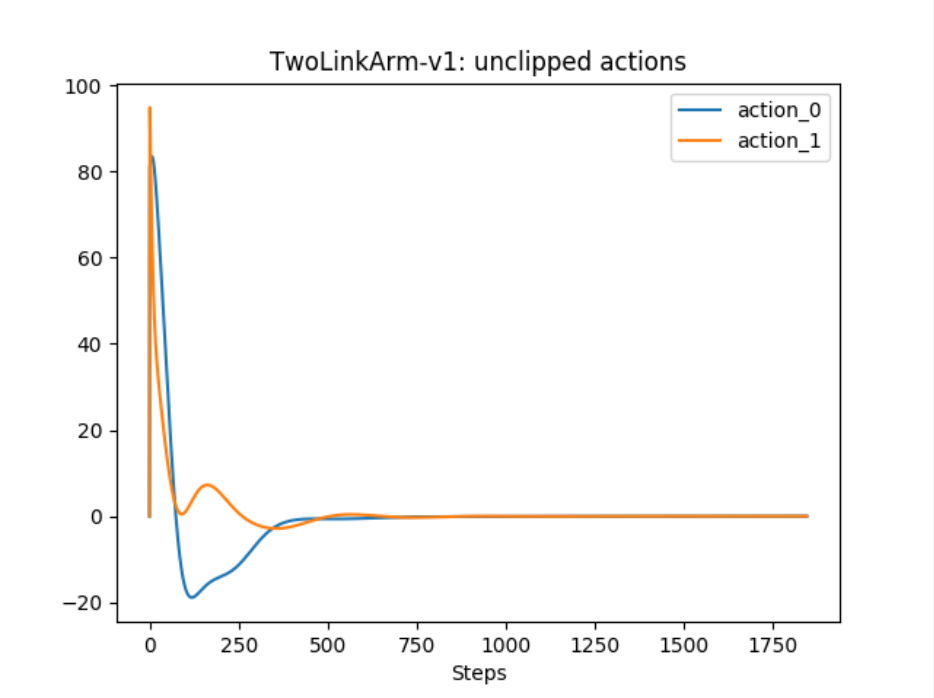


Figure 7: LQR on TwoLinkArm-v1: raw action trajectory



5. LQR on TWoLinkArm-limited-torque-v1 has the total reward -3249.94, steps to

reach the goal 1950. The q, $\dot{q}$, u and the clipped u plots are in the Figure 8, 9, 10.

Figure 8: LQR on TwoLinkArm-limited-torque-v1: state trajectory

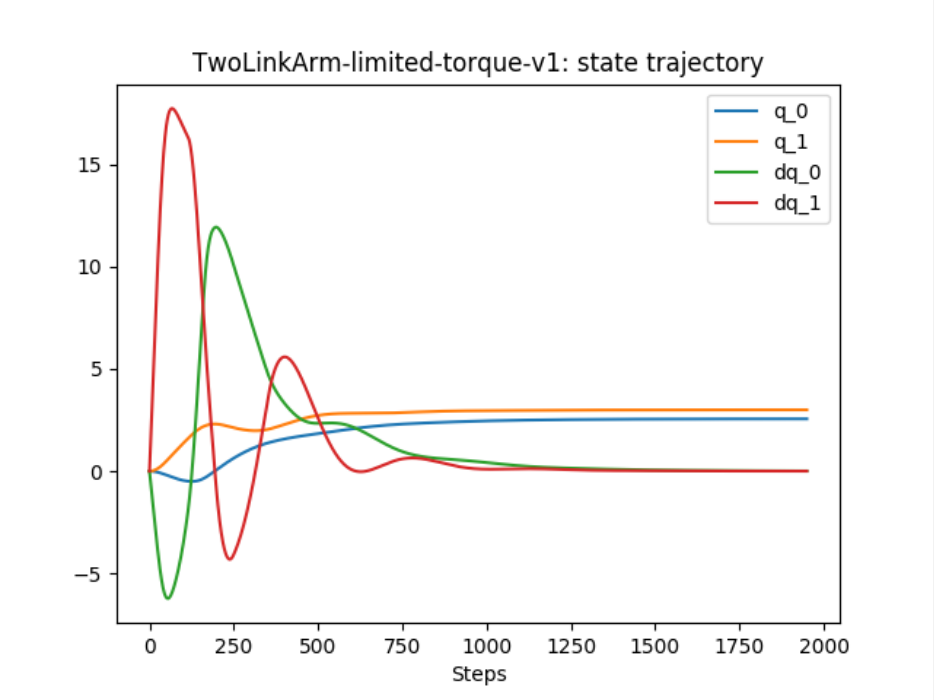


Figure 9: LQR on TwoLinkArm-limited-torque-v1: raw action trajectory

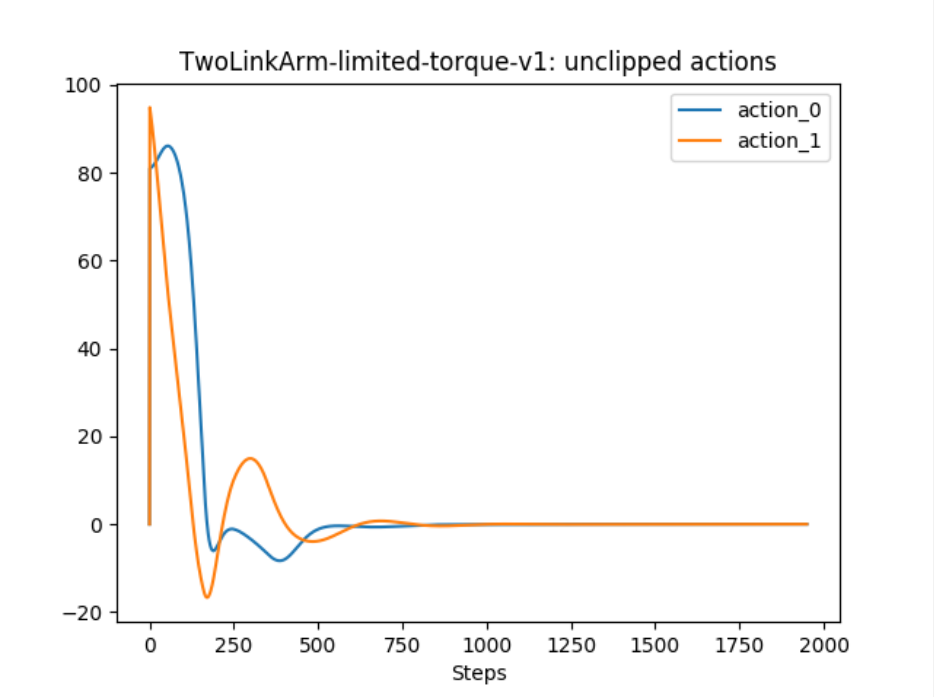
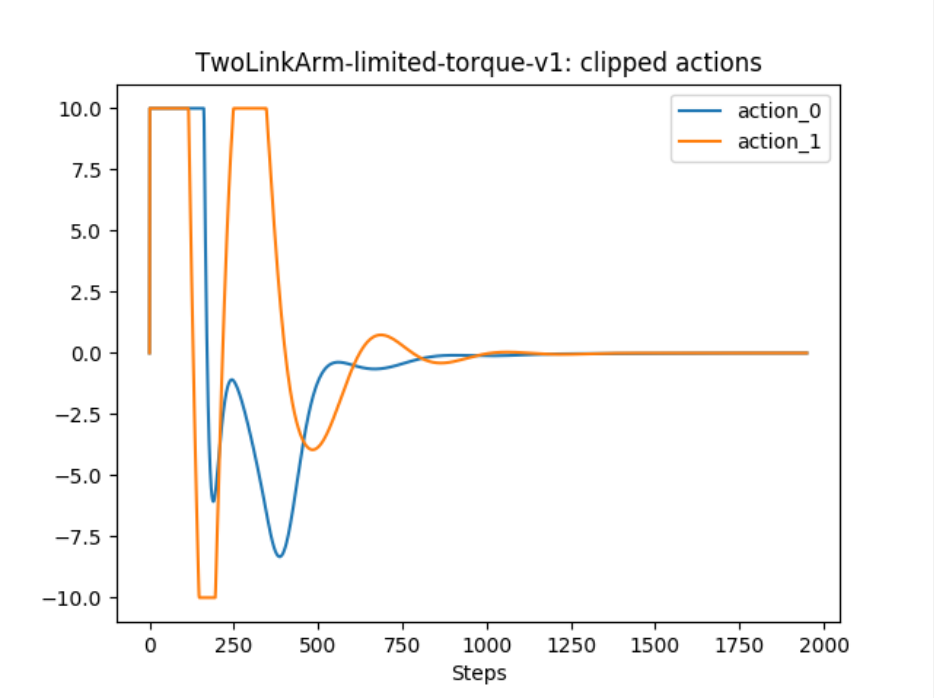


Figure 10: LQR on TwoLinkArm-limited-torque-v1: clipped action trajectory



6. For TwoLinkArm-v1, the action trajectory has the similar shape as its v0 version, but the magnitude is much less than the v0 version because u has larger penalty in v1. Basically, the action plot of the v1 (Figure 7) is like a horizontally streched and vertically compressed version of Figure 2. As the results, the state trajectory of the v1 version is also like a horizontally streched and vertically compressed version of Figure 1. Therefore, the total reward of v1 is much smaller and the total step is much larger than the v0 version.

   For TwoLinkArm-limite-torque-v1, its state action trajectory looks very different from its v0 version, probably because the raw action from LQR is much smaller than v0 and there are less actions to be clipped. Therefore, the state trajectory of the v1 version reaches the goal with much less oscillations than the v0 version. Also, the actions of the v1 version are generally smaller than the v0 version. As the results, the total reward of the v1 version is much higher than the v0 version, even though the number of steps to reach the goal is similar.

Problem 2

Linear Q-network (no experience replay or target fixing)

Answer:

1. **Experiement setup:**

Table 1: LQN Experiement setup

Hyperparameter	**Value**	**Note**
environment	Enduro-v0	
minibatch size	32	Follow [2]
has replay memory?	False	
agent history length	4	Follow [2]
discount factor	0.99	Follow [2]
action repeat	Uniform random 2, 3 or 4	Default by the Openai Gym Enduro-v0 environment
update frequency	Every 4 steps	Follow [2]
learning rate	0.0001	
optimizer	Adam	
adam-beta1	0.9	Tensorflow default
adam-beta2	0.999	Tensorflow default
adam-epsilon	1e-08	Tensorflow default
initial exploration	1.0	Follow [2]
final exploration	0.05	Follow [2]
exploration linear decay steps	1M	Follow [2]
game frame dim	110*84	

2. **Learning curve**

   The evalution reward of the LQN is zero for the first 1.50M steps, and then fluctuates above zero. The evaluation reward near the end of learning is still close to zero. This may indicate that the simple LQN cannot learn the game well.

3. **Final evaluation reward:**

   100-episode greedy policy: 19.69 ± 1.05

Problem 3

Linear Q-network with experience replay and target fixing

Answer:

Problem 4

Linear double Q-network

Answer:

Problem 5

Deep Q-network

Answer:

1. **Experiement setup:**

Table 2: DQN Experiement setup

Hyperparameter	**Value**	**Note**
environment	Enduro-v0	
minibatch size	32	Follow [2]
has replay memory?	True	
replay memory size	500000	For memory consideration, to ensure not reach memory limit of the Bridges cluster.
agent history length	4	Follow [2]
discount factor	0.99	Follow [2]
action repeat	Uniform random 2, 3 or 4	Default by the Openai Gym Enduro-v0 environment
update frequency	Every 4 steps	Follow [2]
learning rate	0.0001	
optimizer	Adam	
adam-beta1	0.9	Tensorflow default
adam-beta2	0.999	Tensorflow default
adam-epsilon	1e-08	Tensorflow default
initial exploration	varies	See "Non-continuous learn" part for more
final exploration	varies	See "Non-continuous learn" part for more
exploration linear decay steps	varies	See "Non-continuous learn" part for more
replay start size	varies	See "Non-continuous learn" part for more
game frame dim	110*84	

The convoluted neural network has the same architecture as the [1], in which the first hidden layer has 16 8*8 filters with stride 4 and rectifier activation, the second hidden layer has 32 4*4 filters with stride 2 and rectifier activation, the final hidden layer is a fully connected layer with 256 rectifier activation units, the ouput layer is a fully connected linear layer with the output dimension the same as the environment action number.

2. **Non-continuous learn:**

   Because of the limited computing resources on the Bridges clusters, the complete learning process (interact with the environment for 5M steps) may take tens of hours. To speed up the queuing on the Bridges clusters, the learning is segmented into 5 parts. The learned model parameter values of each parts are written into file and imported into the next part as the model initial values. Because the

replay memory of each learning part is not remembered, some hyperparameters listed in the table 2 need to be tweaked for each learning part, as show in table 3. It is awared by the authors that the learning behavior of such non-continuous learning may be different from the continuous learning because at each time a new learning part starts, samples in the replay memory have to be recollected. But learning results are reasonble as will be discussed in the following. To mimic the continuous simulation settings, during the non-continuous learning, the initial exploration probablity and the replay start size are decreasing from the first learning part to the last learning part.

Table 3: DQN non-continuous learning setup

	Part 1	**Part 2**	**Part 3**	**Part 4**	**Part 5**
initial exploration	0.5	0.35	0.5 (intend for more exploration since evaluation reward is still not good)	0.1	0.05
final exploration	0.05	0.05	0.05	0.05	0.05
exploration linear decay steps	1M	700K	300K	10K	N/A
replay start size	50K	50K	50K	10K	1000
final step reached of this part	480K	450K	630K	1.33M	1.88M
total step reached of the learning	480K	930K	1.56M	2.89M	4.77M

3. **Learning curve:**

   The evalution reward of the DQN is zero for the first 500K steps, and then increases with lots of fluctuations. It is reasonble for the agent the learn something after so many interactions with the environment, since gaining the reward requires the agent to continuously speed up until passing some cars. The evaluation reward near the end of learning is around 200.

4. **Final evaluation reward:**

   100-episode greedy policy: 165.28 ± 4.40

Problem 6

Double deep Q-network

Answer:

1. **Experiement setup:**

Table 4: DDQN Experiement setup

Hyperparameter	**Value**	**Note**
environment	Enduro-v0	
minibatch size	32	Follow [2]
has replay memory?	True	
replay memory size	500000	For memory consideration, to ensure not reach memory limit of the Bridges cluster.
agent history length	4	Follow [2]
discount factor	0.99	Follow [2]
action repeat	Uniform random 2, 3 or 4	Default by the Openai Gym Enduro-v0 environment
update frequency	Every 4 steps	Follow [2]
target net update frequency	Every 10000 updates (every 40000 steps)	Follow [2]
learning rate	0.0001	
optimizer	Adam	
adam-beta1	0.9	Tensorflow default
adam-beta2	0.999	Tensorflow default
adam-epsilon	1e-08	Tensorflow default
initial exploration	varies	See "Non-continuous learn" part for more
final exploration	varies	See "Non-continuous learn" part for more
exploration linear decay steps	varies	See "Non-continuous learn" part for more
replay start size	varies	See "Non-continuous learn" part for more
game frame dim	110*84	

The convoluted neural network has the same architecture as the [1], in which the first hidden layer has 16 8*8 filters with stride 4 and rectifier activation, the second hidden layer has 32 4*4 filters with stride 2 and rectifier activation, the final hidden layer is a fully connected layer with 256 rectifier activation units,

the ouput layer is a fully connected linear layer with the output dimension the same as the environment action number.

2. **Non-continuous learn:**

   Similar to problem 5, the learning is segmented into 6 parts. Some tweaked hyparameters based on the table 4 is shown in the table 5.

Table 5: DDQN non-continuous learning setup

	Part 1	**Part 2**	**Part 3**	**Part 4**	**Part 5**	**Part 6**
initial exploration	0.5	0.35	0.5 (intend for more exploration since evaluation reward is still not good)	0.1	0.05	0.05
final exploration	0.05	0.05	0.05	0.05	0.05	0.05
exploration linear decay steps	1M	700K	300K	10K	N/A	N/A
replay start size	50K	50K	50K	10K	1000	1000
final step reached of this part	300K	500K	500K	370K	360K	1.68M
total step reached of the learning	300K	800K	1.30M	1.67M	2.03M	3.71M

3. **Learning curve:**

   The evalution reward of the DDQN is basically zero for the first 1M steps, and then increases faster and with less fluctuations than the DQN case. This may prove that DDQN performs better than DQN in terms of stability and learning speed. The learning does not reach the 5M steps due to the time problem. The evaluation reward near the end of learning is around 400, which is higher than the DQN case even the iteration number is less. There is still an increasing trend in the evaluation rewards when the learning stopped. It can be expected that evluation rewards can increase by more iteractions with the environment.

4. **Final evaluation reward:**

   100-episode greedy policy: 420.17 ± 8.57

Problem 7

Dueling deep Q-network (extended from DDQN in the problem 6)

Answer:

1. **Experiement setup:**

   The hyperparameters for the Duel-DDQN are the same as the table 4.

   The convoluted neural network has the same settings as the problem 5 and 6, except there are two seperate fully-connected neural network layers after the second the convoluted neural network. One is for calculating the advantage values, which has 256 rectifier units and action-number of outputs, the other is for calculating the state value, which has 256 recrifier units and 1 output. The state value and advantage values are then combined to give the Q value for each action.

2. **Non-continuous learn:**

   Similar to problem 6, the learning is segmented into 6 parts. Some tweaked hyparameters is shown in the table 6.

Table 6: Duel-DDQN non-continuous learning setup

	Part 1	**Part 2**	**Part 3**	**Part 4**	**Part 5**	**Part 6**
initial exploration	0.5	0.35	0.5 (intend for more exploration since evaluation reward is still not good)	0.1	0.05	0.05
final exploration	0.05	0.05	0.05	0.05	0.05	0.05
exploration linear decay steps	1M	700K	300K	10K	N/A	N/A
replay start size	50K	50K	50K	10K	1000	1000
final step reached of this part	390K	370K	510K	330K	320K	1.36M
total step reached of the learning	390K	760K	1.25M	1.58M	1.90M	3.26M

3. **Learning curve:**

   The evalution reward of the Duel-DDQN is basically zero for the first 800K steps, and then increases faster with even less fluctuations than the DDQN case. This

proves the advantages of the dueling architecture. The learning does not reach the 5M steps due to the time problem. The evaluation reward near the end of learning is around 450, which is higher than the DDQN case even the iteration number is less. At the time the learning stopped, the evaluation rewards have already become steady. It is expected that further learning may not increase the evaluation rewards more.

4. **Final evaluation reward:**

   100-episode greedy policy: 478.81 ± 7.63

8. Performance Comparison of the Different Methods

Answer:

Table 7: Performance Comparison of the Different Methods

LQN	**LQN-Fixing**	**DLQN**	**DQN**	**DDQN**	**Duel-DDQN**
19.69 ± 1.05	??	??	165.28 ± 4.40	420 ± 8.57	478.81 ± 7.63

Comments ?????? Basically, for the learning steps, LQN > DQN > DDQN > Duel-DDQN, but the evaluation rewards have the reverse relationship. This shows that Duel-DDQN is more effective than DDQN than DQN and than LQN.