ADL HW3 REPORT

Huang Liang Ying 05/24/2020

Q1: Models

1. Policy Gradient

   Model structure of actor:

   $state\ vector:\ S$

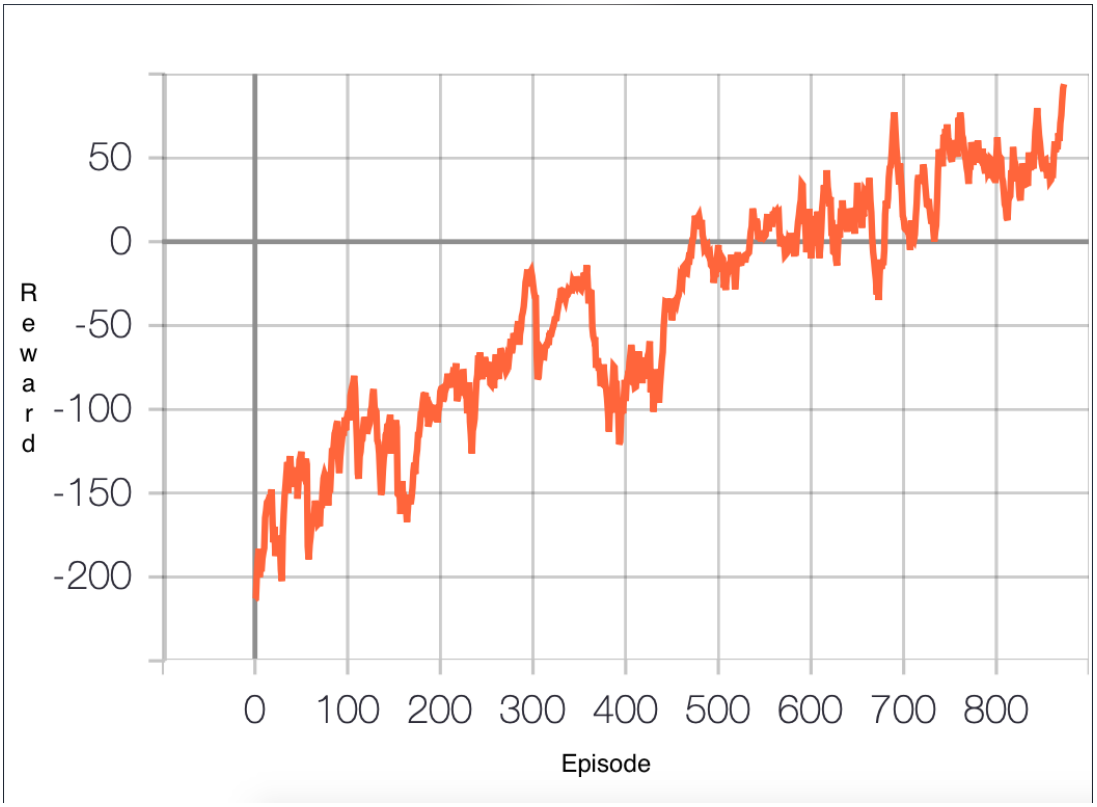
   $X = Relu(S * W_1)$

   $prob = Softmax(W_2 * X)$

   $where W_1 \in R^{8\times64},\ W_2 \in R^{64\times4}$

```
PolicyNet(
  (fc1): Linear(in_features=8, out_features=64, bias=True)
  (fc2): Linear(in_features=64, out_features=4, bias=True)
)
```

Learning curve of policy gradient



Observed from the graph, the actor began to get positive rewards after 480 episodes and it got about 90 scores when finishing training.

2. Deep Q Network

   Model structure of Deep Q Network:

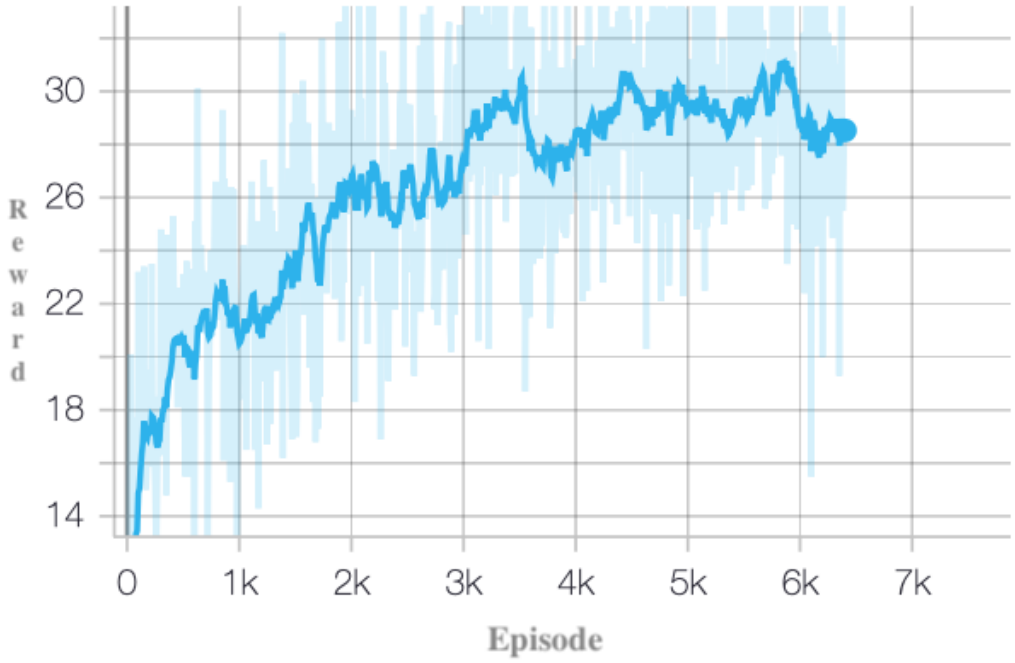
   game image processing part is the same as sample code's model

```
DQN(
  (conv1): Conv2d(4, 32, kernel_size=(8, 8), stride=(4, 4))
  (conv2): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2))
  (conv3): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1))
  (fc): Linear(in_features=3136, out_features=512, bias=True)
  (head): Linear(in_features=512, out_features=9, bias=True)
  (relu): ReLU()
  (lrelu): LeakyReLU(negative_slope=0.01)
)
```

   Loss function: $L(\theta) = E_{(s,a,r,s')}[(r + \gamma max_{a'} \hat{Q}_{\theta'}(s', a') - Q_\theta(s, a))^2]$
   where $\hat{Q}$ is target network and Q is online network while training.

   *episodes shown in the below graph display every ten episodes, i.e. #10, #20....

   Learning curve of Deep Q Network

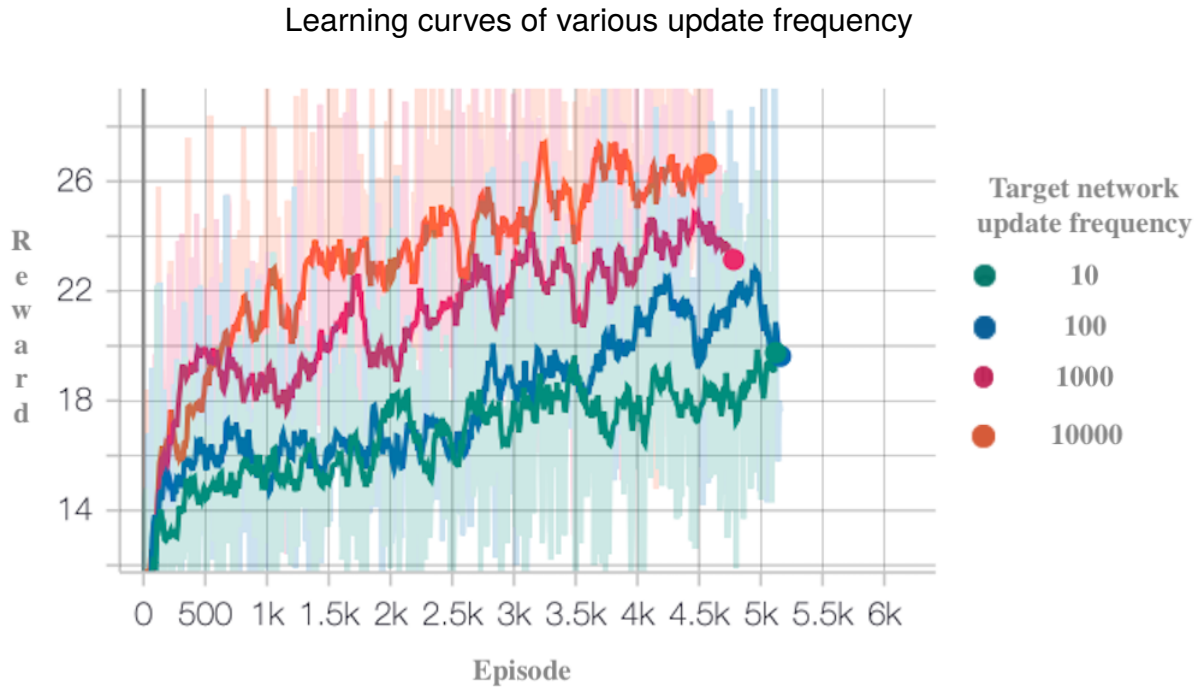


   The above graph showed that:

   - the agent got about 28 scores at the end of training process.
   - after 3500 episodes, the improvement rate appeared to slow down.

Q2: Hyperparameters of DQN

- Target network update frequency



Since I use the tip that fixing target network for a training period and then updating it within an update frequency for more stable training. I supposed that different target network update frequencies will influence training process.

The green, blue, pink, and orange curves are the reward curves over training episodes of updating frequency of 10, 100, 1000 and 10000 respectively.
The graph showed that:

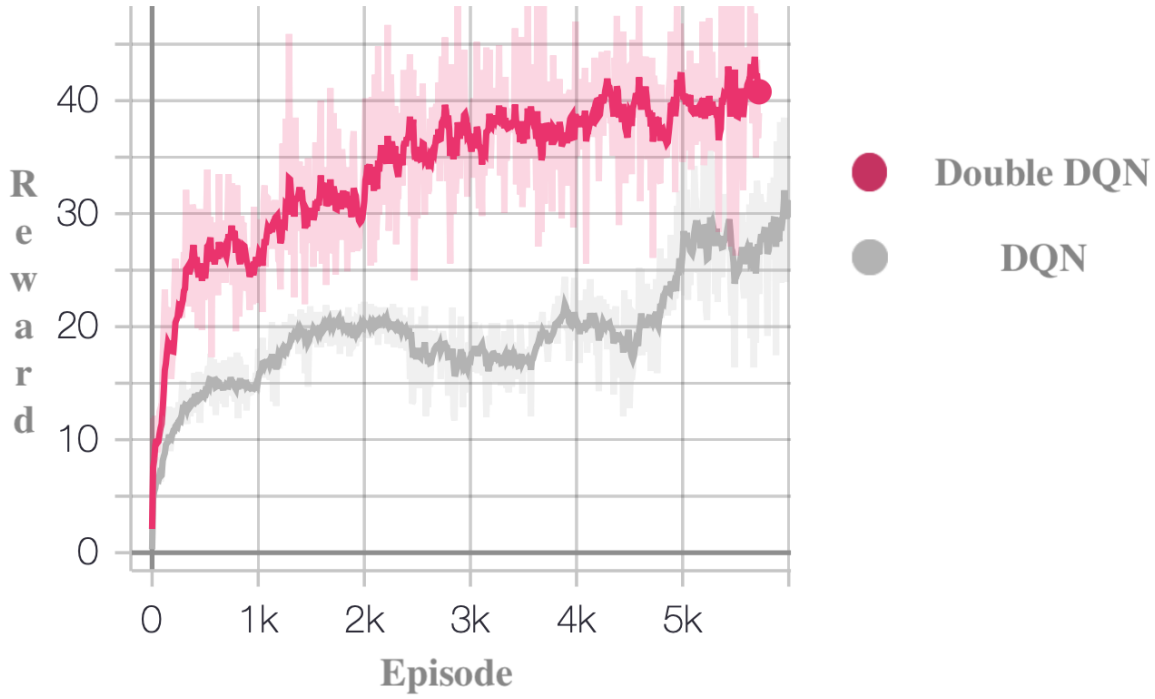
1. The least update frequency got the highest reward in training process.
2. frequency 1000 and 10000 have relative steep slope compared to frequency 10 and 100 in the beginning of training.

I thought the reason is that if we update target network too frequently, the training process might appear to be unstable, thus, resulting in poor performance.

Q3: Improvements of Policy Gradient / DQN

1. Double DQN Experiment setting:
    - Environment: MsPacmanNoFrameskip-v0
    - Hyperparameter:
        - batch size: 256
        - learning rate: 0.0001
        - total training steps: 1500000
        - target network update frequency: 500
        - online network training frequency: 4
        - gamma: 0.99

Learning curves of Double DQN and DQN



- TD target of DQN: $Q(s,a) = r(s,a) + \gamma max_a Q(s',a)$
- TD target of Double DQN: $Q(s,a) = r(s,a) + \gamma Q(s', argmax_a Q_o(s',a))$
- Q_o is the q value of online network

Double DQN uses online network to choose action for next state and then target network calculates the q value of taking the action at that state. This mechanism help reducing q value overestimation and hence help agent choosing proper action based on more accurate reward.

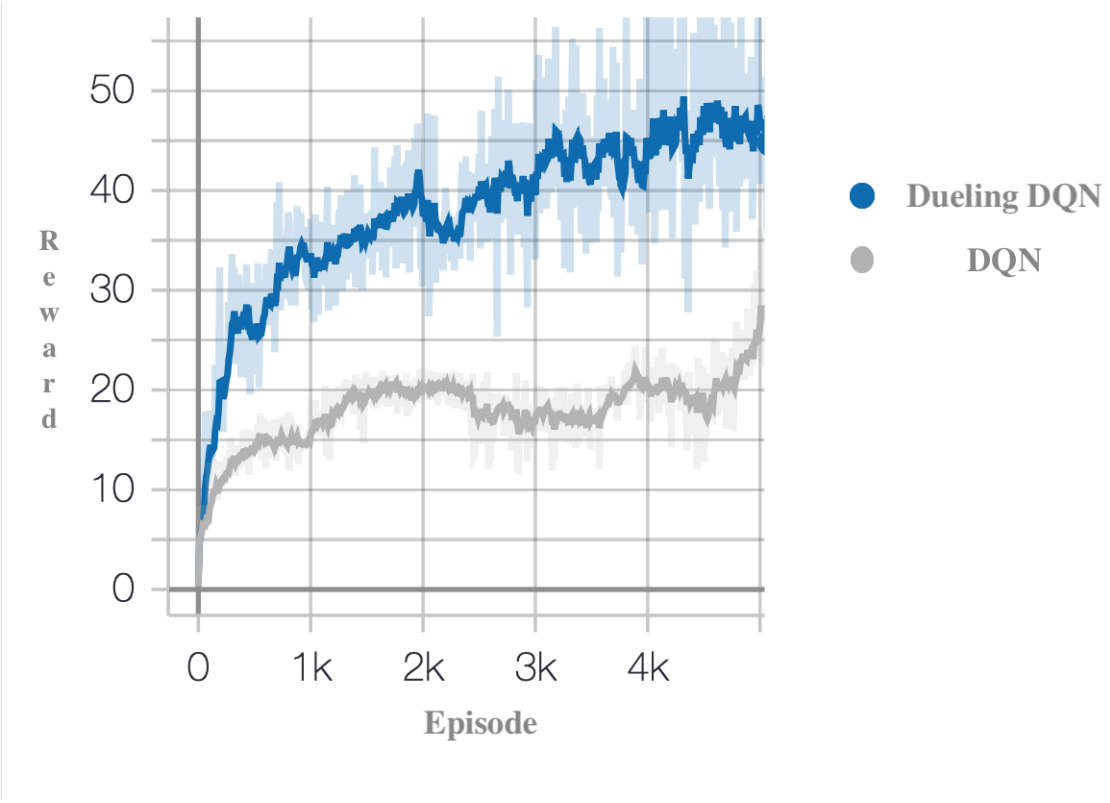
As we can observe from the above graph, the learning curve of Double DQN surpass that of DQN in total training process. That is, Double DQN indeed improve performance

when playing MsPacmanNoFrameskip-v0. The maximum reward gap of them are about 20 scores in Episode #3000 while Double DQN constantly outperform DQN for about 10 scores.

2. Dueling DQN Experiment setting:

   - Environment: MsPacmanNoFrameskip-v0
   - Hyperparameter:
     - batch size: 256
     - learning rate: 0.0001
     - total training steps: 1500000
     - target network update frequency: 500
     - online network training frequency: 4
     - gamma: 0.99

Learning curves of Dueling DQN and DQN



   - Q value of Dueling DQN: $Q(s,a) = A(s,a) + V(s)$
   - $A(s,a)$ stands for the advantage of taking action a in state s
   - $V(s)$ stands for the value in the state

Dueling DQN manage to taking into consideration that there are states where its actions do not affect the environment in any relevant way. The dueling mechanism can learn states

value with no action's effect, i.e. V(s). Thus, Q value of Dueling DQN are more authentic than DQN.

The performances of Dueling DQN and DQN are shown in the above graph. Dueling DQN outperform DQN for 20 scores in average in the entire training process, which means Dueling DQN actually got better performance in MsPacmanNoFrameskip-v0.