Assignment 3 - Playing Pong with Deep Q-Network

Haitong Li

Part 1: Written Questions

1. What is a replay buffer and why do we need it.

   A replay buffer stores the data (state, action, reward, next_state, etc) of the last #buffer_size frames played. The algorithm will then randomly sample mini-batches and use them to continue to train the model. We need a replay buffer because it will make the training stabler. With a replay buffer, we can reuse our previous experience multiple times, which is especially useful for an algorithm without human input. Also, it will make the sampled data as independent as it can get, which will lead to a better convergence performance.

2. The goal of line 48 and line 57 of dqn.py?

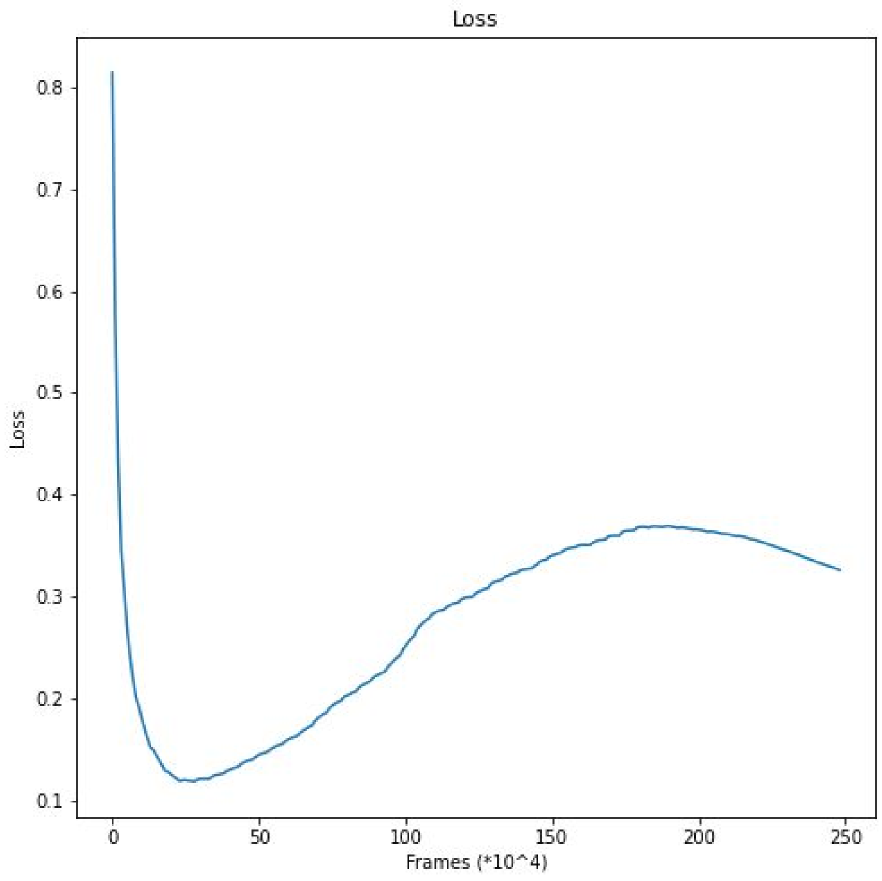
   These lines are performing the epsilon-greedy policy. At the beginning, the epsilon is set to 1 and then gradually decays as the training goes. This approach is to make sure the maximum exploration takes place in the early stage of training, which will gradually switch to exploitation later as the epsilon decays and the model is trained.

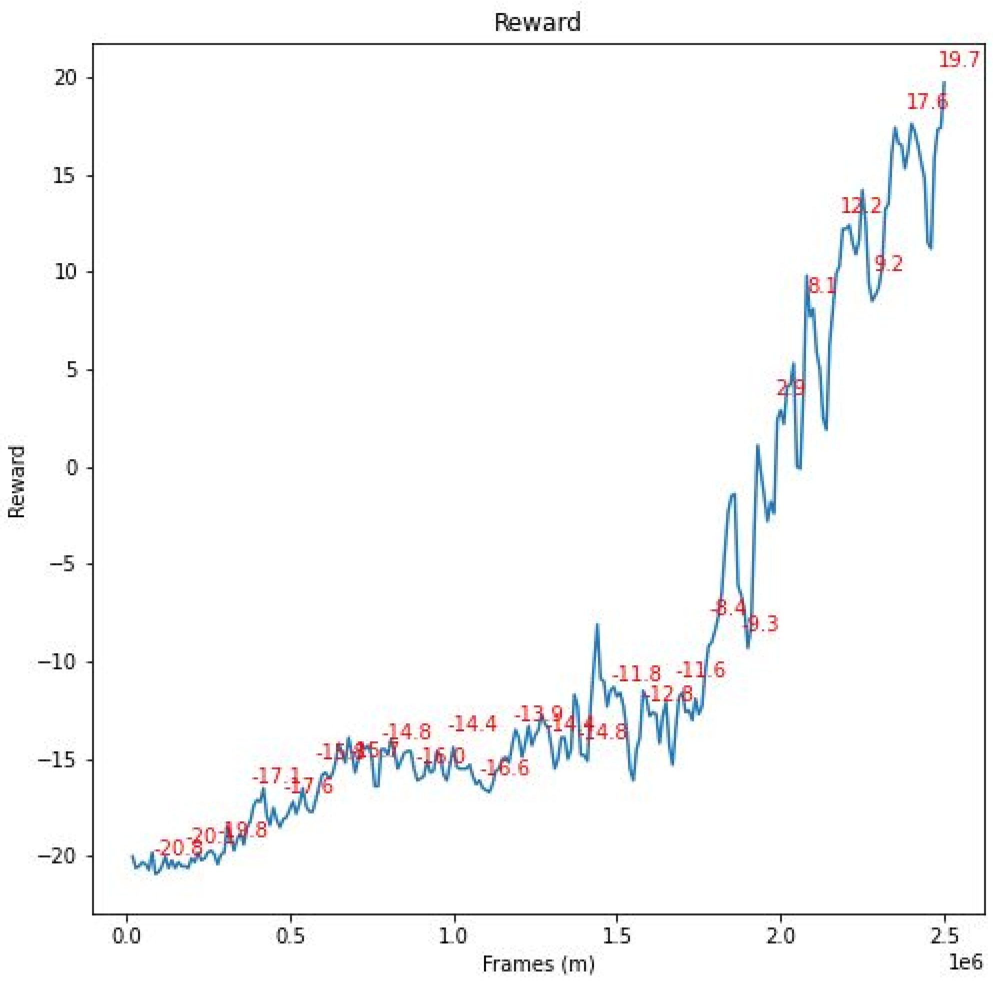
3.
    a. Tuning

       I changed gamma to different values in 0.01 steps, and found little difference between the results, so I just kept the gamma = 0.99 for the training. The same happened for the size of the replay buffer so the original setting was kept.

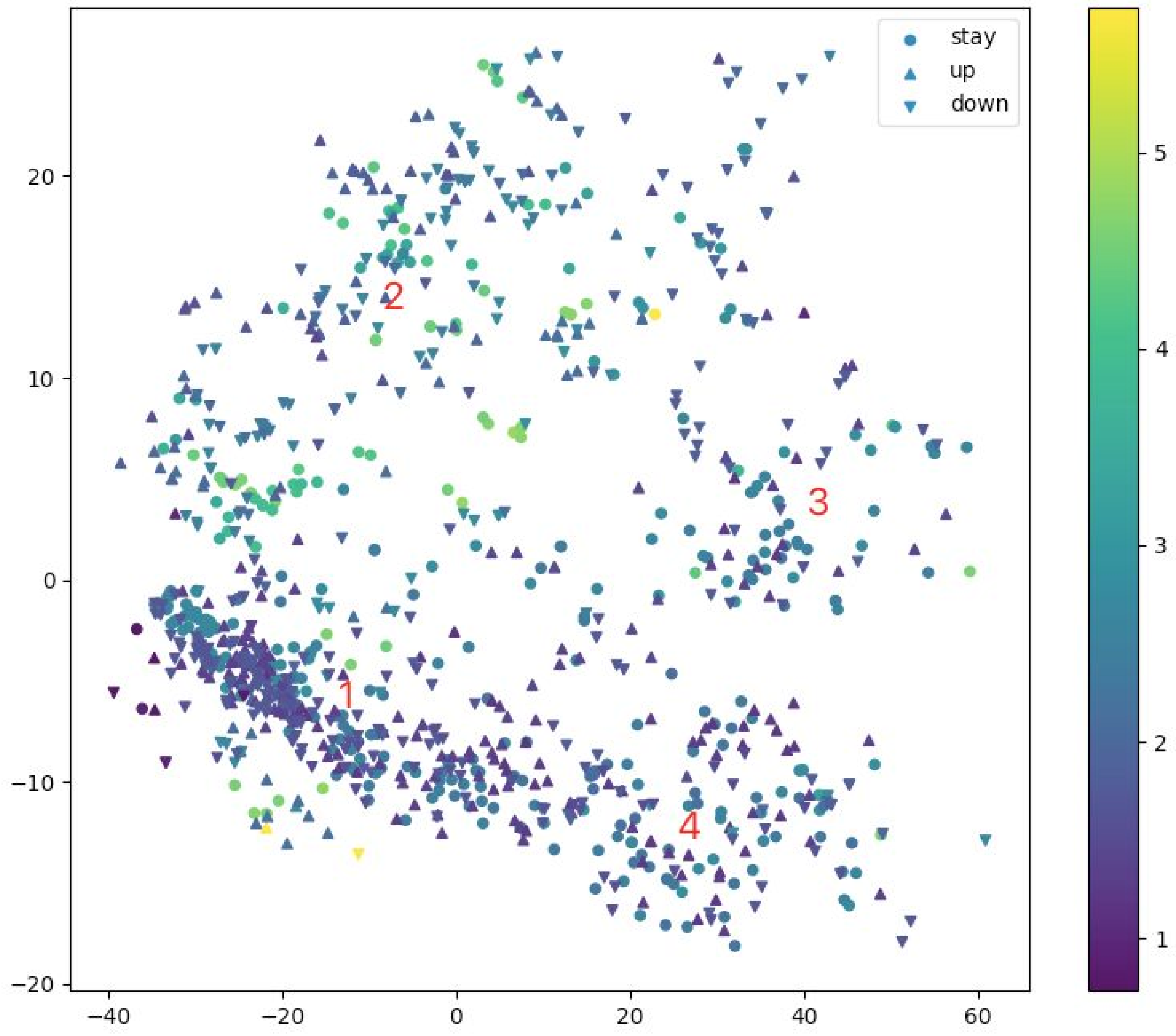
       For the total number of frames, I first trained the model with the preset number, which is 1 million, and the reward was much lower than the expected value (-15 vs. ~20). So I converted the code into an infinite loop which breaks when the average reward gets to the desired point. At the end, it took around 2.5~3.5 million frames to reach the desired level of reward.

b. Loss and reward plots




Reward
Reward
Frames (m)
1e6
20
15
10
5
0
−5
−10
−15
−20
0.0
0.5
1.0
1.5
2.0
2.5
19.7
17.6
12.2
9.2
8.1
2.9
-8.4
-9.3
-11.8
-11.6
-12.8
-13.9
-14.4
-14.8
-14.4
-14.8
-16.0
-16.6
-15.6
-15.7
-17.1
-17.6
-20.8
-20.1
-19.8

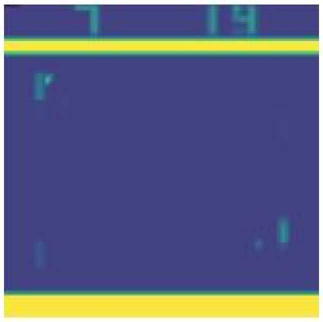
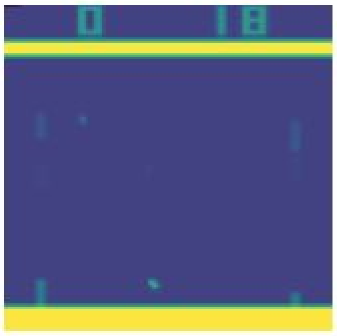
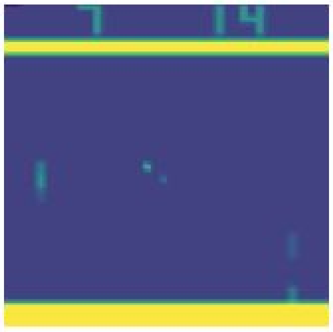
Part 2: Analysis

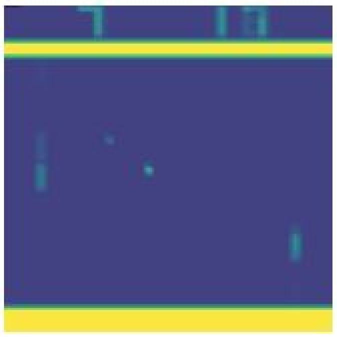
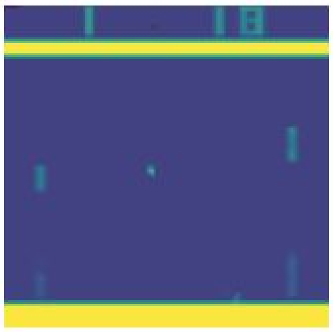
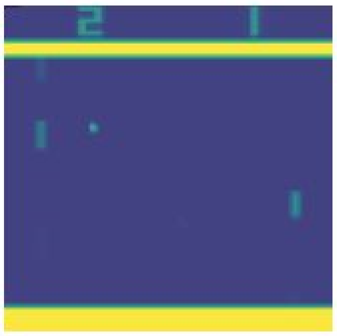
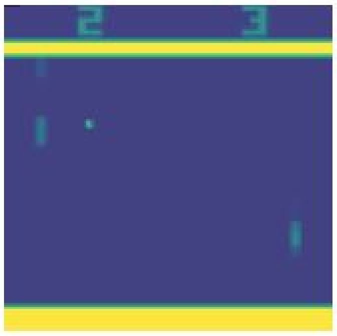


I collected the features of the second last layer of 1000 frames and used PCA to embed the data into 2d place and color the embedded data points according to their q values.

The embedding shows that there are roughly 4 clusters in the features which are marked with red numbers in the plot. And most of the q values are within the range of 1-2.

I will select some of the frames in each region to further analyze:

Figure	Region	Q value	Action	Analysis
	1	1.555923	stay	The ball is bouncing between the player and the opponent, which shows that the actions in this region should be in a defensive position. And that's why the q values in this region are not very high because the player generally is not getting a point.
	2	2.71756	stay	The q value is high and it means that if the player chooses to stay, it will likely win this point in this situation.
	3	1.55	up	Trying to bounce the ball back to the opponent. Relatively low q value means that the player is not winning a point with this action.

	3	1.5759	stay	Also a defensive position. The player chooses to stay to wait for the ball bouncing back from the opponent.
	3	1.66	down	It looks like the player is trying to catch the ball coming from the opponent. Region 3 seems to contain the actions that are in a waiting position.
	4	1.60	stay	The plot and the q value are probably showing that the player might lose the point with this action.
	4	1.60	up	This plot is almost in the same position as the previous one, so points in region 4 may all be this kind of actions by which the player is not in a winning position.