ECS 170 Winter 2020 (03/15) Programming #3
Bingwei Wang 914683636

Part 1.

1). The latter one would be easier for the agent to learn from. First, Q learner needs to do some calculations and choose an optimal action. Arrays could make this process easier, whereas images need Q learner more time to make decisions. Second, arrays are more easier to store and transfer than images.

2). Neural network learns to map inputs to outputs from training data, making the model become stronger. We substitute neural network for the original lookup table and using each $\hat{Q}_(s, a)$, update as a training example. The inputs are state s and action a. The output is the target value of $\hat{Q}$ according to the training rules. State is the current situation (each positions of players and ball) of the game, and action is the movement that the agent takes. Target value is a certain improved Q value after some training.

3). The purpose is to choose the action for the current state, according to the epsilon greedy policy. It will not always perform action chosen by Q learner. Sometimes it will choose a random action, by checking if it jumps into **epsilon**. The probability of choosing a random action starts at $\text{epsilon}_{\text{start}}$, and decay towards $\text{epsilon}_{\text{final}}$. The decay rate is controlled by $\text{epsilon}_{\text{decay}}$.

Part 2. By the Q learning training rule (Mitchell Pg.383),

$$Q^{(1)}(s_t, a_t) = r_t + \gamma max_a \hat{Q}(s_{t+1}, a).$$

This training rule has difference error between two sides of the equation. Thus, to minimize this error, we need to compute temporal difference error using **MSE**, which is $Loss_i(\Theta_i)$.

y is the Q value for the (state, action) pair using the model. $Q(s, a; \Theta)$ is the Q value that update rule, which should be using $\text{target}_{\text{model}}$. s would be state and a would be action. And Θ is the parameters for this loss function.

Loss function helps us to minimize the error, resulting in optimizing the parameters of the neural networks, and making Q learner easier to achieve maximum score. Loss function distills all aspects of variables down to a single number, which could significantly improve the model by only seeing one number to indicate it is better or worse. The error renders the algorithm to change the weights to let next evaluation reduce the error.

Part 4.

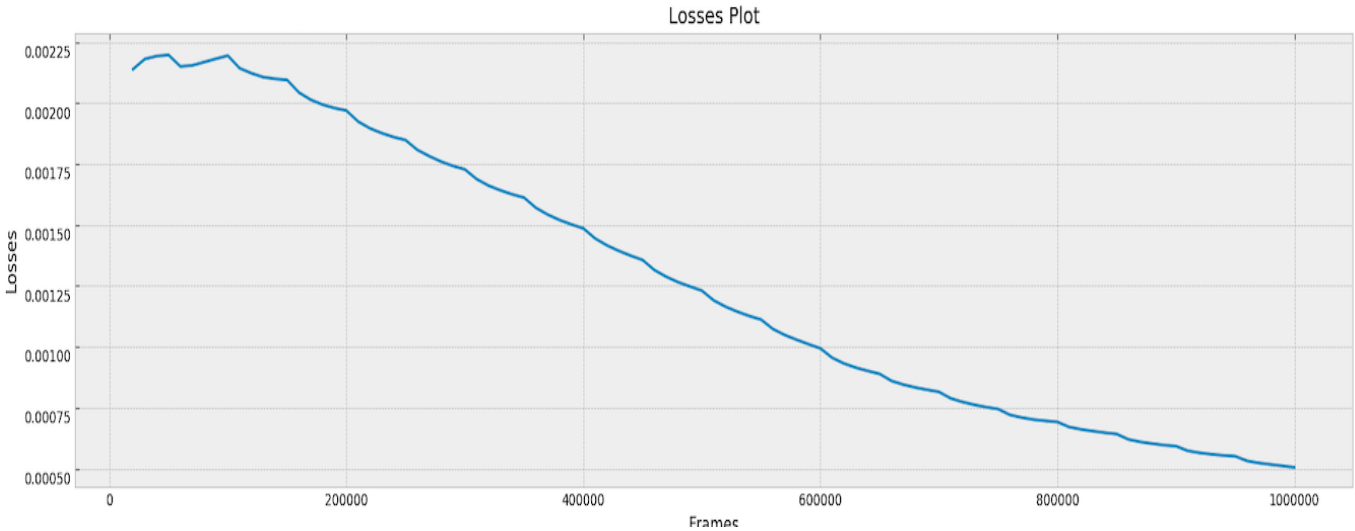

Figure 1: Losses

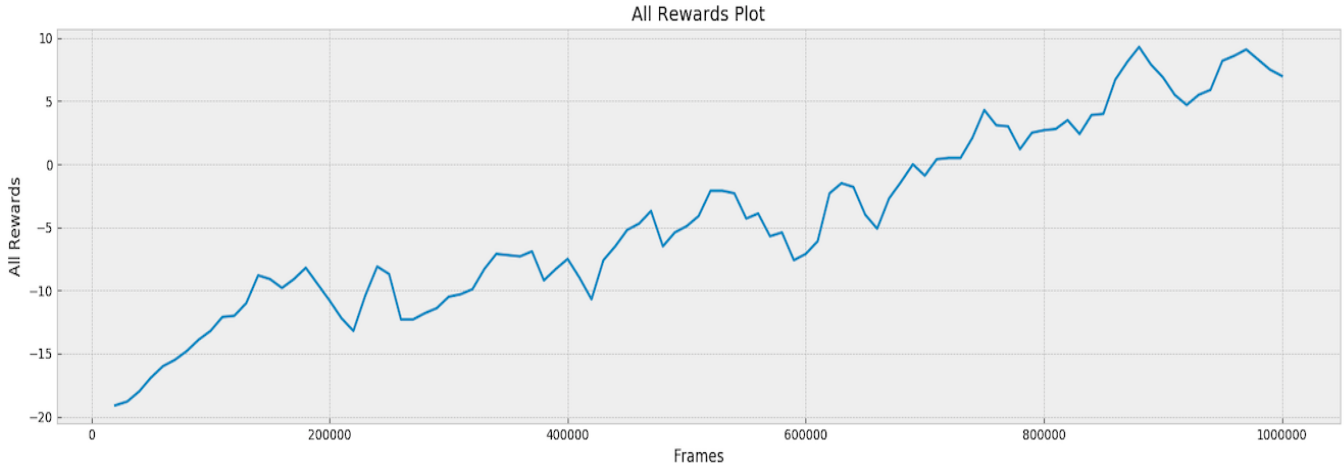

Figure 2: All Rewards