Project 3
CS525: Reinforcement Learning

INTRODUCTION
Project 3 is to build a Deep Q network (DQN) and train the model on Atari breakout game using OpenAi's Gym framework.

1. ARCHITECTURE

The network has four layers, two convolutional layers and two linear fully connected layers. The network takes in four screenshots and outputs the resulting action. The output is four discrete actions ([0,1,2,3]). The ReLu activation function was used for all layers except the last one(out[put layer).
The input is batches of four images of size (84, 84).

Fig.1 Sample screenshot of the breakout game

2. HYPER PARAMETERS

Different parameters have been tried with this environment and the network.

Parameter	**Value**
Number of episodes	15 millions
Learning Rate	0.0001
Batch size	32
Minimum Replay Buffer size	50,000
Maximum Replay Buffer size	1 million
Exploration strategy	Epsilon decay
Maximum epsilon	1
Minimum epsilon	0.2

3. EXPERIMENTS PERFORMED

I have tried different models with different numbers of layers at the beginning. The number of layers did not significantly improve the learning process. More the number of layers, the longer it took to train.But the rewards were almost the same when compared after 10,000 iterations.

Hyper Parameters Tuning

Different hyper parameters were tuned by setting the other once constant. It is found that learning rate in the order of -4 works better as it does not converge fast or miss big steps. The results were terrible with a learning rate of 0.001 and with a much lower learning rate, the model took longer to get rewards.

One of the exhaustive tuning processes of the exploration strategy, I've tried numerous methods. In the end epsilon decay with a decay from 0.999 to 0.2 found to be working better.

I've trained the model with different batch quizzes including 1, 32, 64 and 128. With 128 there was a sudden spike in the reward and took longer to train, but it reduced and became stable like all other batches after a certain point.

Optimizer

The optimizer used was Adam optimizer with a learning rate of 0.0001

Loss function

I implemented smootL1loss

0.5 * x^2   if |x| <= 1

|x| - 0.5   if |x| > 1

Please note that this is not exactly the same as Huber Loss. Huber loss is given below[1].

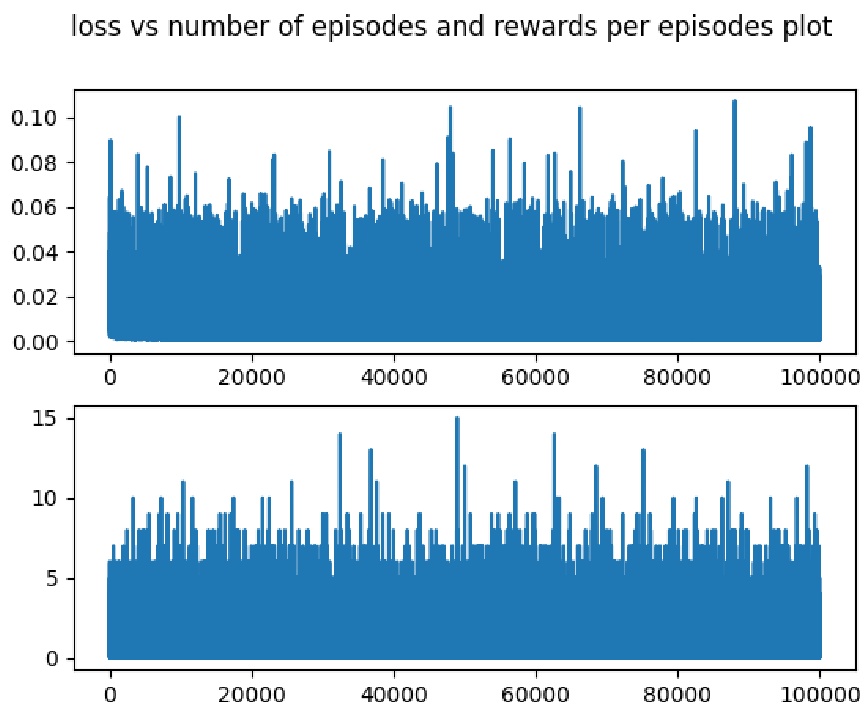
$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta, \\ \delta \cdot \left(|a| - \frac{1}{2}\delta\right), & \text{otherwise.} \end{cases}$$

Yes, I’ve tried Huber loss as well and found that smoothLiLoss works better than Huber loss.

4. RESULTS

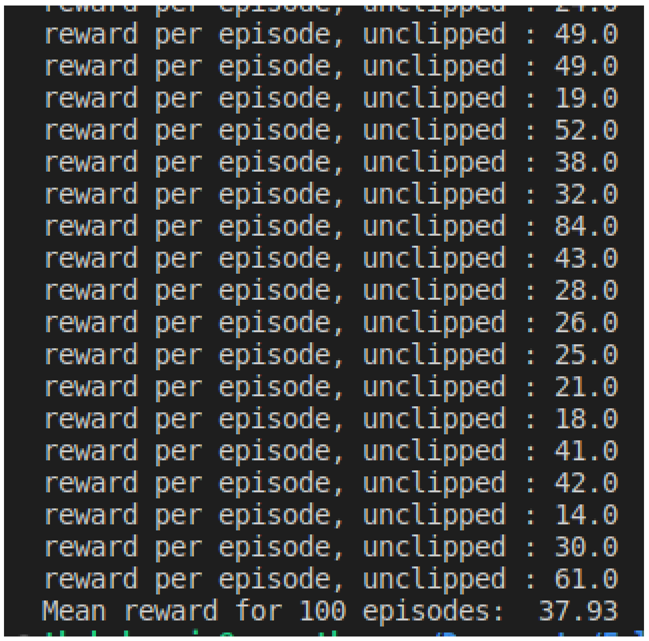
After training more than 15 million episodes, I was able to get rewards that are in comparable standards. While training I got a collective **unclipped** reward of 78 from the last 30 episodes. Note: There has been few uncertainty while training the model, I didn't capture the rewards per episode for 15 million episodes. Recapturing it again means running the training again for 10+ hours. I’ve given the plots with data I captured.

1. Cumulative rewards for 10,000 iterations on training
2. Loss vs number of episodes for 10,000 iterations on training



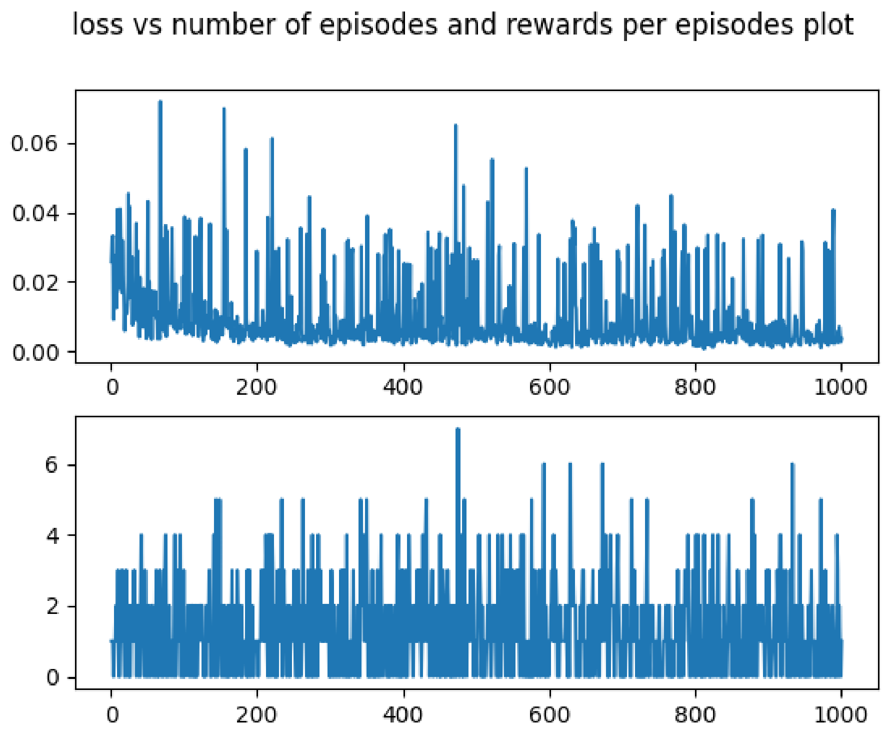
3.

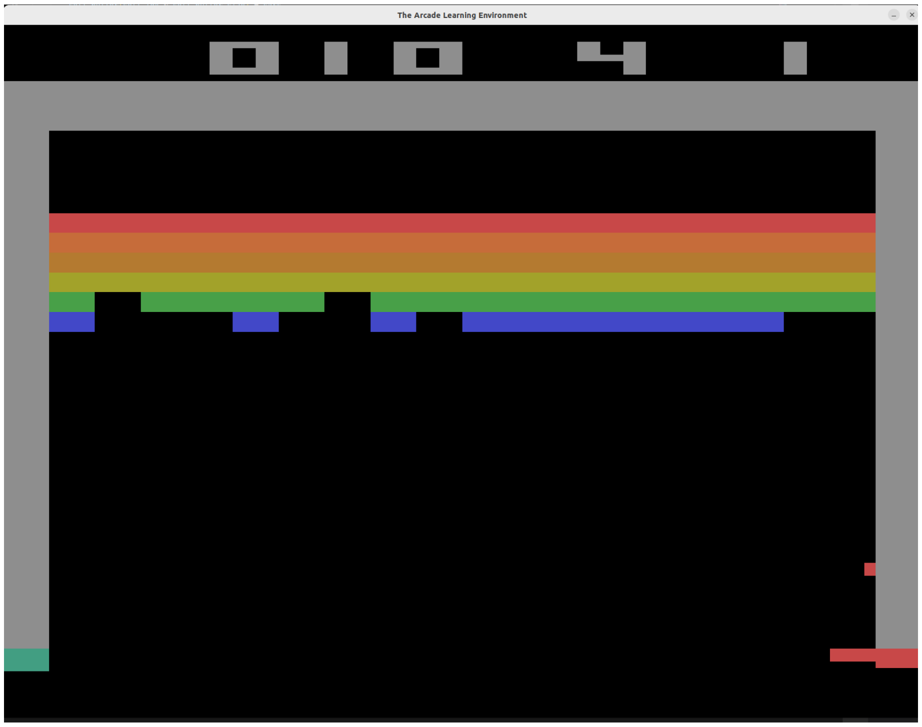
<u>3. Mean reward collected over 100 episodes on testing</u>

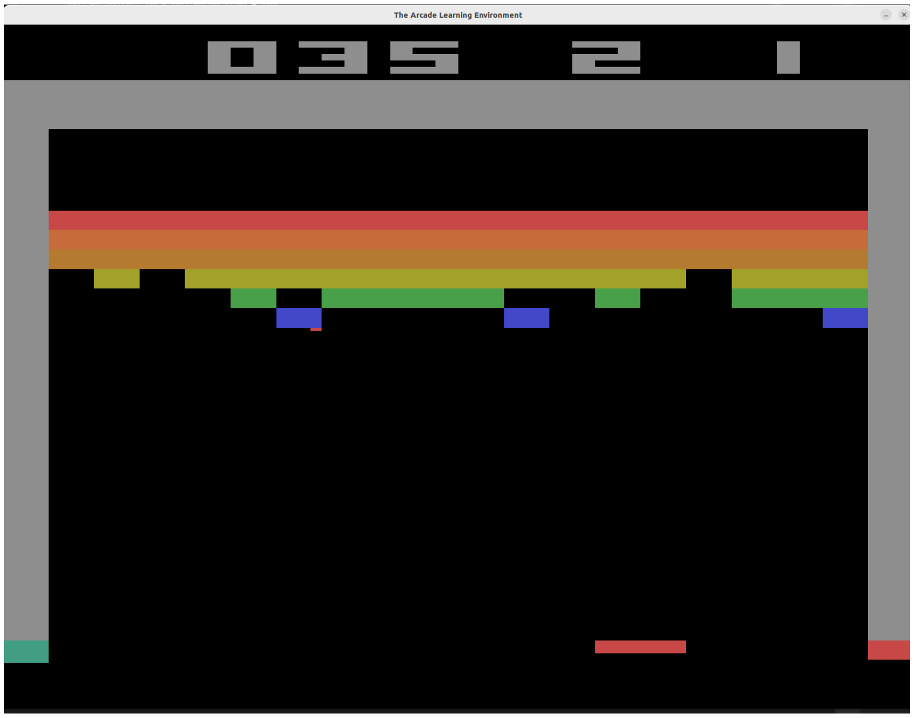


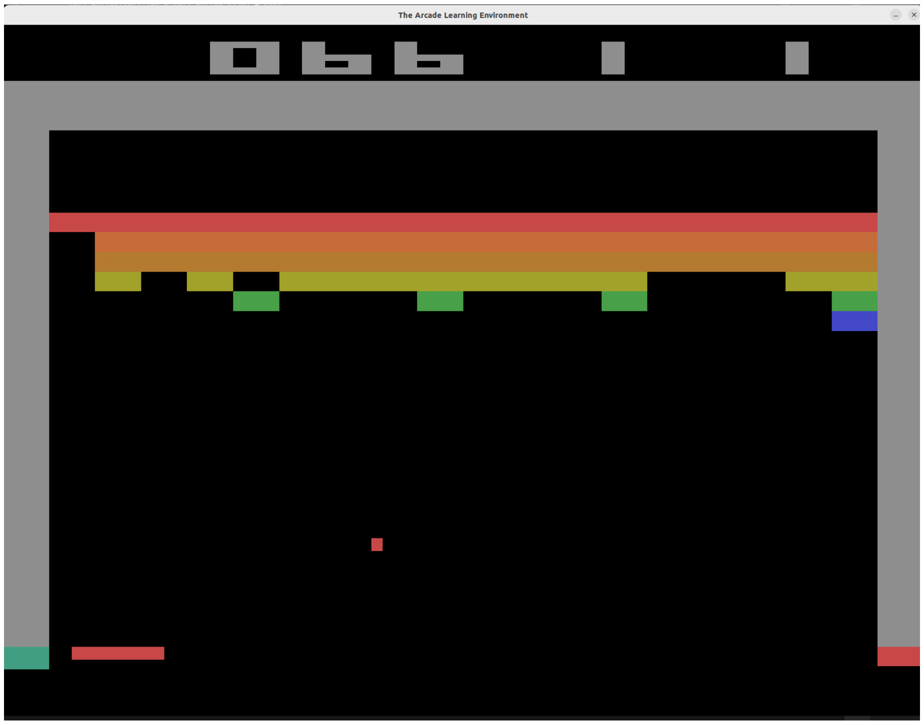
```
reward per episode, unclipped : 49.0
reward per episode, unclipped : 49.0
reward per episode, unclipped : 19.0
reward per episode, unclipped : 52.0
reward per episode, unclipped : 38.0
reward per episode, unclipped : 32.0
reward per episode, unclipped : 84.0
reward per episode, unclipped : 43.0
reward per episode, unclipped : 28.0
reward per episode, unclipped : 26.0
reward per episode, unclipped : 25.0
reward per episode, unclipped : 21.0
reward per episode, unclipped : 18.0
reward per episode, unclipped : 41.0
reward per episode, unclipped : 42.0
reward per episode, unclipped : 14.0
reward per episode, unclipped : 30.0
reward per episode, unclipped : 61.0
Mean reward for 100 episodes:  37.93
```

4. Rewards(individual episodes) vs 1,000 iterations on training
5. Loss vs number of episodes for 1,000 iterations on training




The Arcade Learning Environment
010 4 1


The Arcade Learning Environment
035 2 1



REFERENCES

[1] “Huber loss,” *Wikipedia*. Oct. 06, 2022. Accessed: Nov. 15, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Huber_loss&oldid=1114469353