Implementation of Atari Environments using Deep Q-Learning

Miki Padhiary
UBID: mikipadh Person Number: 50286289

December 6, 2018

1 Introduction

In this experiment, I implemented DQN using two environments from OpenAIs Gym library. Following environment was implemented:

- PongNoFrameskip
- ChopperCommandNoFrames

I used Stable Baselines implementation of DQN.

2 Stable Baselines

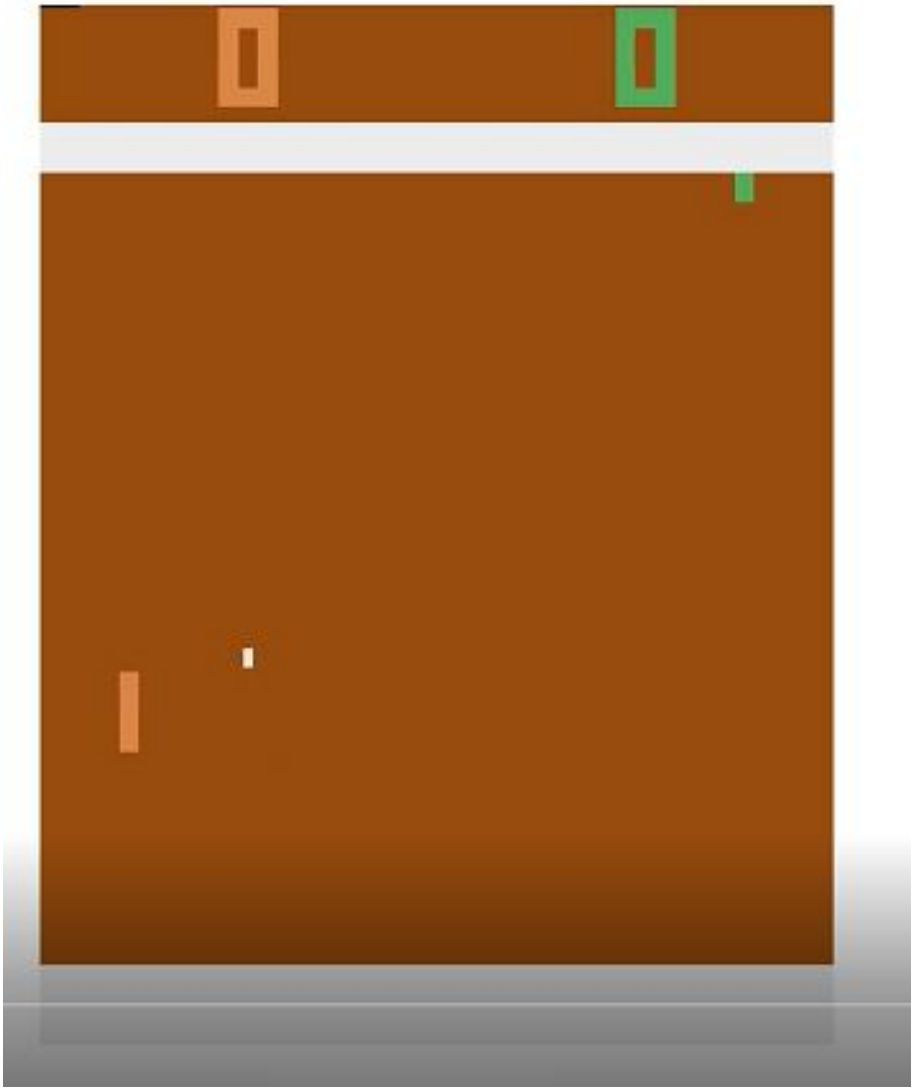
Stable Baselines is a set of improved implementations of Reinforcement Learning (RL) algorithms based on OpenAI Baselines. RL Baselines zoo also offers a simple interface to train and evaluate agents. The main differences with OpenAI Baselines is:

- Unified structure for all algorithms
- PEP8 compliant (unified code style)
- Documented functions and classes
- More tests more code coverage

3 PongNoFrameskip

Pong is a two-dimensional sports game that simulates table tennis. The player controls an in-game paddle by moving it vertically across the left or right side of the screen. They can compete against another player controlling a second paddle on the opposing side. Players use the paddles to hit a ball back and forth. The goal is for each player to reach eleven

points before the opponent; points are earned when one fails to return the ball to the other.

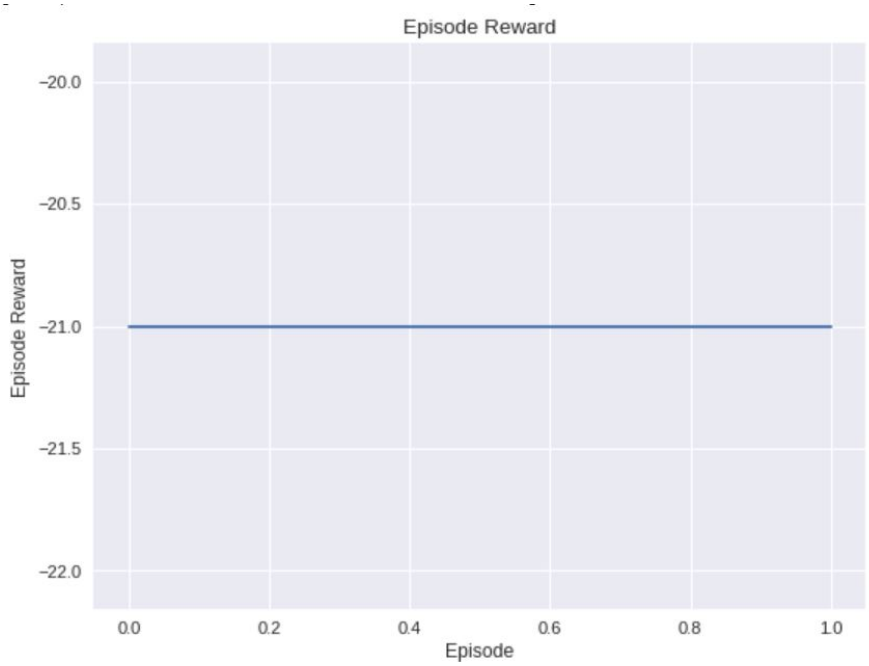


Results The best reward obtained when total_timesteps was set to 10000 is depicted in the figure below:

```
print(plot_rewards)

Last Episode Reward: -21.0
Last Episode Reward: -21.0
[-21.0, -21.0]
```

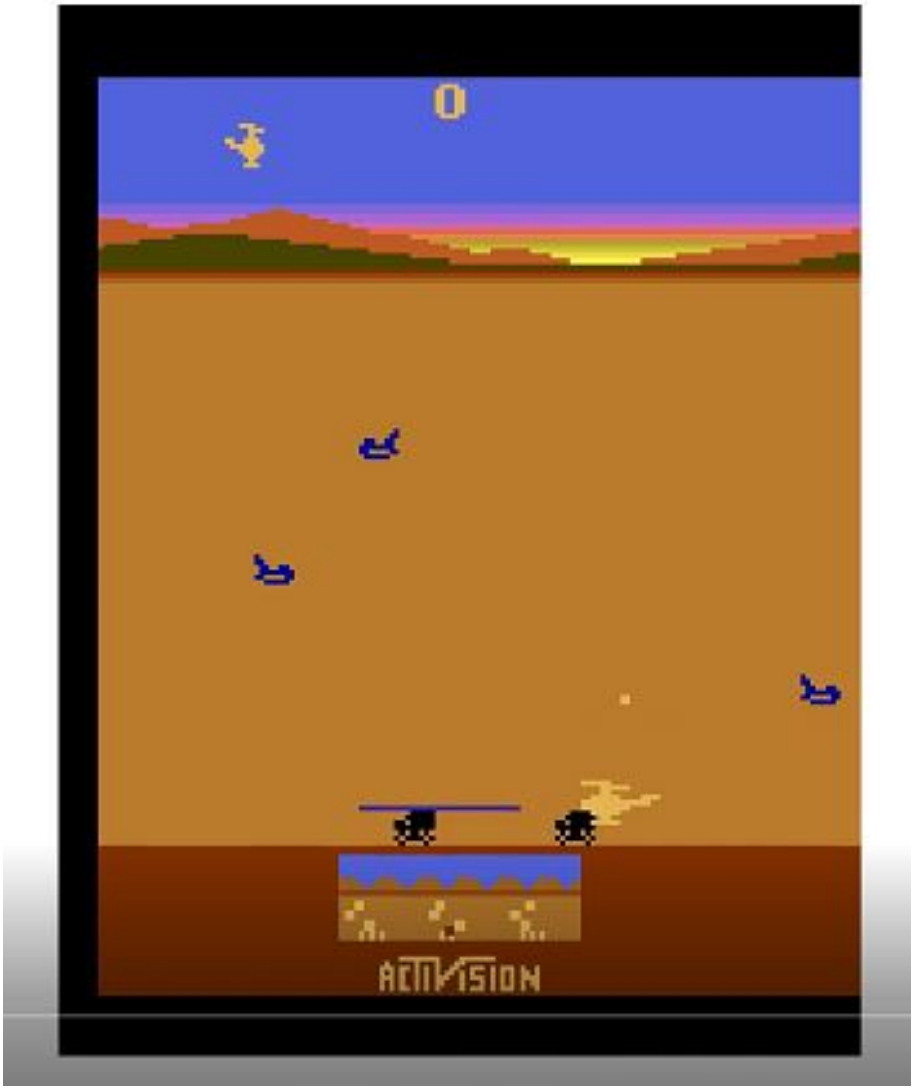
The episode reward graph is as below:



As the number of states is more, the agent takes time to learn and since at the start it is exploring, we are getting a negative reward. If the number of timesteps is increased significantly then the reward will increase and the agent will be able to perform correctly. The reward was not having a significant increase in values.
Changing the **stable_baselines.deepq.policies import MlpPolicy** also resulted in the same rewards.

4 ChopperCommandNoFrames

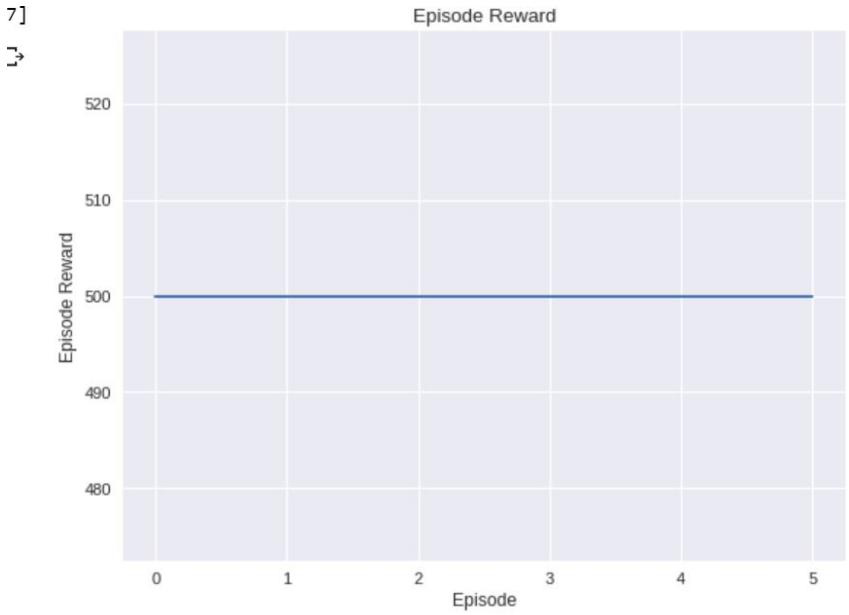
Chopper Command is a horizontally-scrolling shooter. In Chopper Command the player controls a military helicopter in a desert scenario protecting a convoy of trucks. The goal is to destroy all enemy fighter jets and helicopters that attack the player's helicopter and the friendly trucks traveling below, ending the current wave.[2] The game ends when the player loses all of his or her lives or reaches 999,999 points.



Results The best reward obtained when total_timesteps was set to 10000 is depicted in the figure below:

```
Last Episode Reward: 500.0
Last Episode Reward: 500.0
Last Episode Reward: 500.0
```

The episode reward graph is as below:



As the number of states is more, the agent takes time to learn and since at the start it is exploring, we are getting a negative reward. If the number of timesteps is increased significantly then the reward will increase and the agent will be able to perform correctly. Changing the **stable_baselines.deepq.policies import MlpPolicy** also resulted in the same rewards.

References

[1] Stable Baselines
https://stable-baselines.readthedocs.io/en/master/modules/dqn.html

[2] Open AI Gym
https://gym.openai.com/envs/#classic_control

[3] Pong
https://en.wikipedia.org/wiki/Pong

[4] Chopper Command
https://en.wikipedia.org/wiki/Chopper_Command