Github link:
https://github.com/Zhefe/RL_UseBaseLine_Carpole_Pong

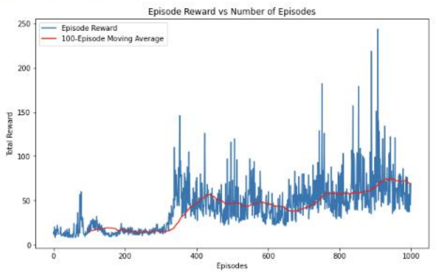
Code usage:
I tried to reuse many parts of the code, you only need to modify the parameter "run = carpole/pong" and "if_use_bl = T/F" on the last chunk of jupyter notebook and click run all.

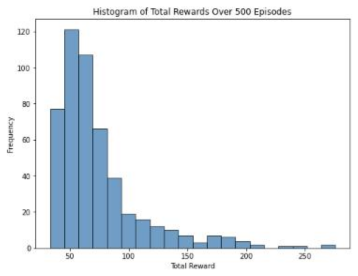
Overall problem:
The performance of the code seems very unstable, it's performance largely depends on early stage training. if training gets surprisingly good performance in early stage, the whole training process goes well and cost a lot of computing resource (especially for use baseline), otherwise it perform bad. This could be explained by trapping on the local maxima in random early stage. Maybe this can be solve by add a "detect and jump out" step but I didn't do that.

Part 1
Though the Overall problem parts show that the result and average seems lack of persuasion, I still provide screenshot for it and part 2 on the following.
Part 2
I use if_use_bl = False, which means the baseline will always be 0. Though it seems less competitive of non constant baseline, it show a more stable and fluently performance on computing resource cost. Which means, it will be less possible to be trapped by a extremely high/poor performance and stuck in computing(when high) or quickly produce a trash optimize(when poor).

Car pole: do not use baseline(right) and use baseline(left)

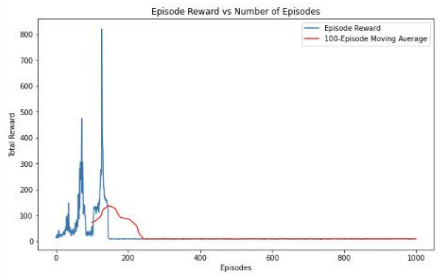
```
Episode 0, Total Reward: 12.0
Episode 100, Total Reward: 19.0
Episode 200, Total Reward: 13.0
Episode 300, Total Reward: 13.0
Episode 400, Total Reward: 40.0
Episode 500, Total Reward: 43.0
Episode 600, Total Reward: 50.0
Episode 700, Total Reward: 80.0
Episode 800, Total Reward: 42.0
Episode 900, Total Reward: 58.0
```

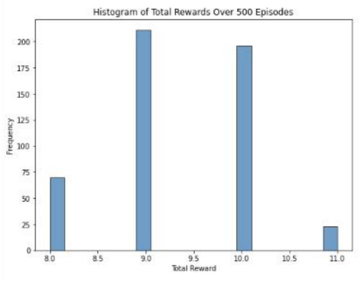




```
Mean Reward: 74.266
Standard Deviation of Reward: 37.36687361821966
```

```
Episode 0, Total Reward: 14.0
Episode 100, Total Reward: 58.0
Episode 200, Total Reward: 10.0
Episode 300, Total Reward: 10.0
Episode 400, Total Reward: 9.0
Episode 500, Total Reward: 9.0
Episode 600, Total Reward: 8.0
Episode 700, Total Reward: 9.0
Episode 800, Total Reward: 10.0
Episode 900, Total Reward: 9.0
```

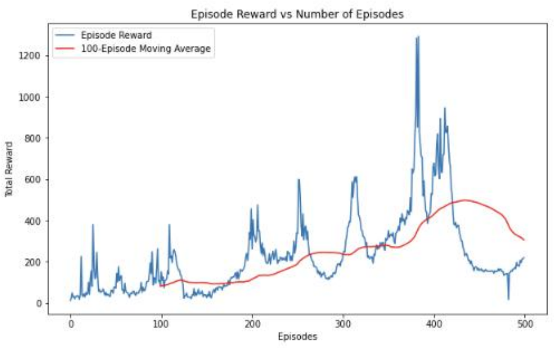


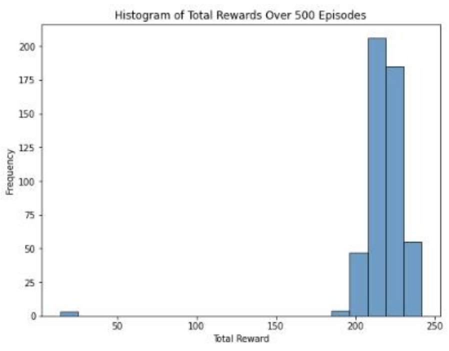


```
Mean Reward: 9.344
Standard Deviation of Reward: 0.7730873171899795
```

pong: do not use baseline(right) and use baseline(left)

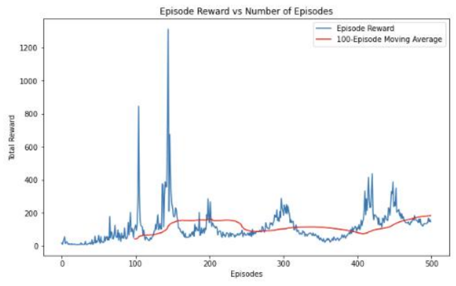
```
Episode 0, Total Reward: 12.0
Episode 100, Total Reward: 151.0
Episode 200, Total Reward: 261.0
Episode 300, Total Reward: 236.0
Episode 400, Total Reward: 678.0
train down
```

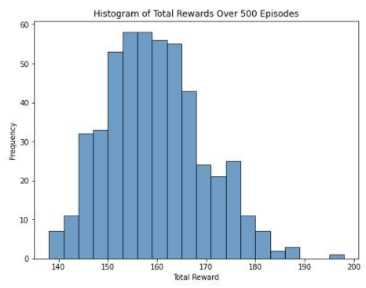




```
Mean Reward: 217.83
Standard Deviation of Reward: 18.105057304521296
```

```
Episode 0, Total Reward: 17.0
Episode 100, Total Reward: 127.0
Episode 200, Total Reward: 140.0
Episode 300, Total Reward: 186.0
Episode 400, Total Reward: 113.0
train down
```





```
Mean Reward: 159.148
Standard Deviation of Reward: 9.852009744209555
```