Assignment I

Comparison of On-policy and Off-Policy Monte Carlo Methods

REINFORCEMENT LEARNING
3-1,2024

Apoorva Gayatri K
2022A7PS0232H

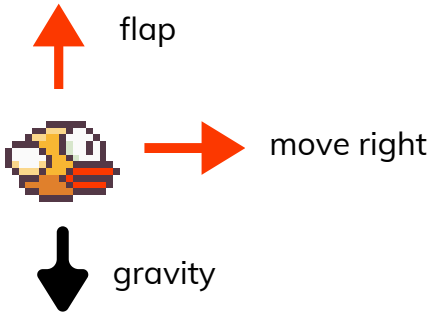
Problem Statement:
The problem inspired by the simple, yet popular game- Flappy Bird.

The original game requires you to pass as many pipes as possible to achieve the highest possible score.

The original game involves you to also take in consideration the projectile trajectory of the bird when it flaps up or falls down due to gravity.

To make our problem simple we have changed the possible actions the bird may move:

1. Go right
2. Go up (flap)
3. No action (Falls due to gravity)

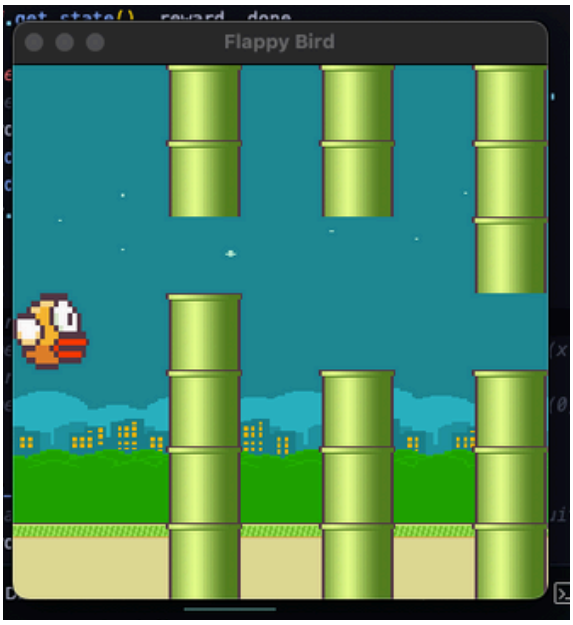


For simplicity we have also considered only
Three pipes the bird has to pass to win the game

How do we formulate this game as a grid problem?

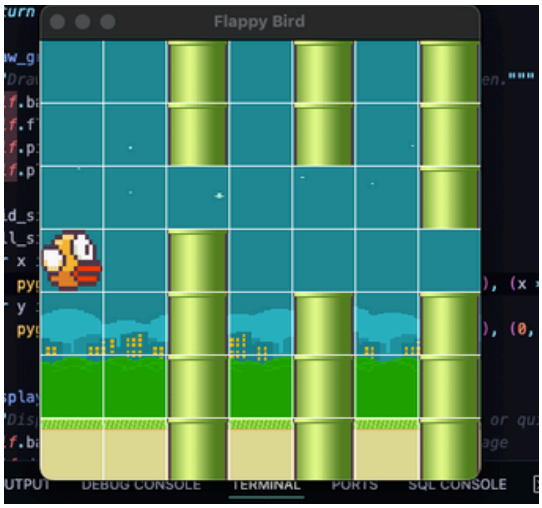
<u>Changes made to the game to suit a grid problem:</u>

- <u>Environment</u>
  - Drew the environment as a 7x7 grid
  - Gravity acts equal across all columns (1 block down
  - Chose (3,0) as start state
  - Pipes are the blocks
    [(0, 2), (1, 2), (3, 2), (4, 2), (5, 2), (6, 2)],
    [(0, 4), (1, 4), (4, 4), (5, 4), (6, 4)],
    [(0, 6), (1, 6), (2, 6), (4, 6), (5, 6), (6, 6)]]
  - Floor covers the blocks of the 7th row
    [(6,0), (6,1), (6,2), (6,3), (6,4), (6,5), (6,6)]



- <u>Game Dynamics</u>
  - action space ['UP','RIGHT','NO ACTION']
  - if you choose to go UP you go to the preceding row and you stay in the same column
  - if you choose to do RIGHT you go to the succeeding column but stay in the same row
  - if you choose "NO ACTION" you go to the succeeding row, stay in the same column(with the effect of gravity)

Prototype



Grid Based Flappy Bird

Formulating this problem as a Reinforcement Learning Problem:

State:

The coordinates of the bird i.e (x-coordinate,y-coordinate) is taken as the state for this problem.The terminal states are all boxes that are pipes and floors and the box of (3,6).

If you reach (3,6) you have won the game.

If you hit any of the pipes or the floor (i.e you reach the boxes they are drawn on),you lose the game.

If you go out of bounds i.e you try to go up when you are on the first row you lose the game.

Reward:

Considering all the cases where the agent tries to find a loophole to maximise returns rather than reaching the terminal states.

The following reward system is proposed:

1. When you hit a pipe or the floor you gain -500
    - To discourage the bird to reach this state - a penalty.
2. When you reach the end of the game you gain +500
    - To motivate the bird to complete the game
3. When you reach any other state you get a base reward of 100, but additive reward increases linearly across columns
    - To encourage forward movement of the bird,(particularly to prevent the bird from staying in the same column and maximising the rewards by just flapping up and down avoiding the pipes the increased reward of the subsequent column forces the greedy policy to go right.)

Action:

As Explained before, there are three actions the bird may choose from:

- Flap (goes up one block)
- Go right (goes right by one block)
- No action(Falls down due to gravity)

7 units

7 units

+100	+110	-500	+130	-500	+150	-500
+100	+110	-500	+130	-500	+150	-500
+100	+110	+120	+130	+140	+150	-500
start +100	+110	-500	+130	+140	+150	Goal +500
+100	+110	-500	+130	-500	+150	-500
+100	+110	-500	+130	-500	+150	-500
+100	+110	-500	+130	-500	+150	-500

Table showing reward on reaching every block

On Policy Monte Carlo Method for Policy Evaluation.

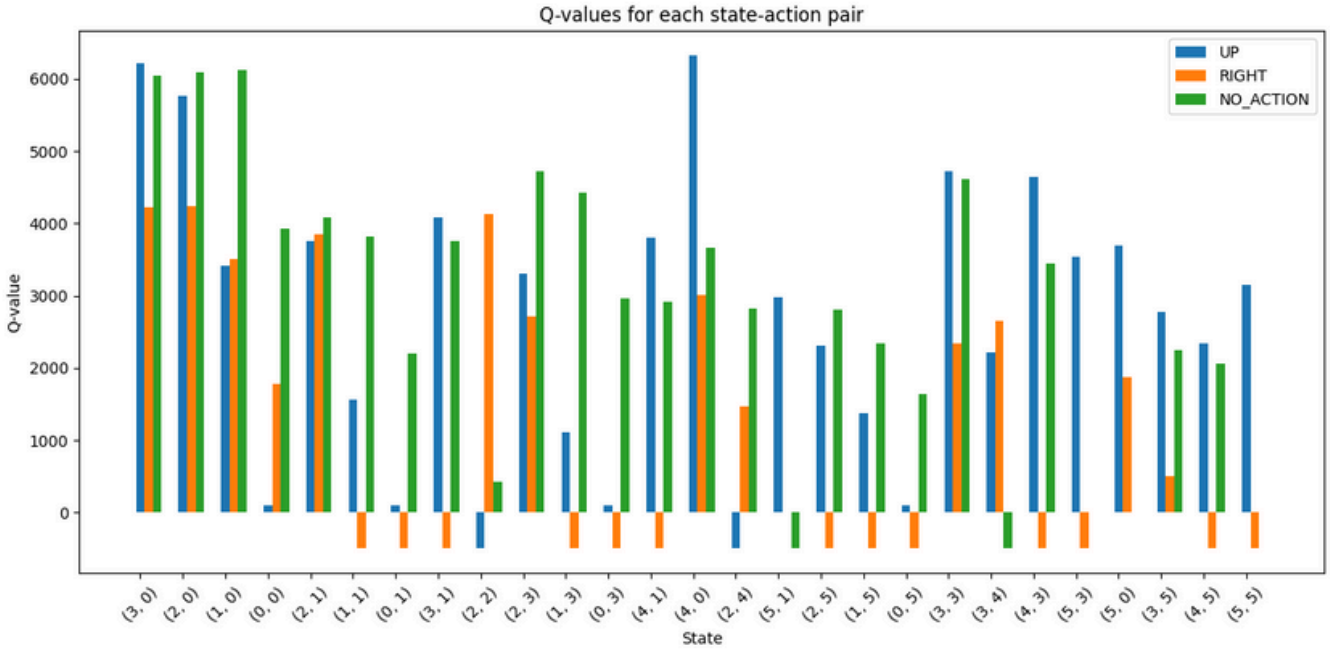
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
  $Q(s,a) \leftarrow$ arbitrary
  $\pi(s) \leftarrow$ arbitrary
  $Returns(s,a) \leftarrow$ empty list

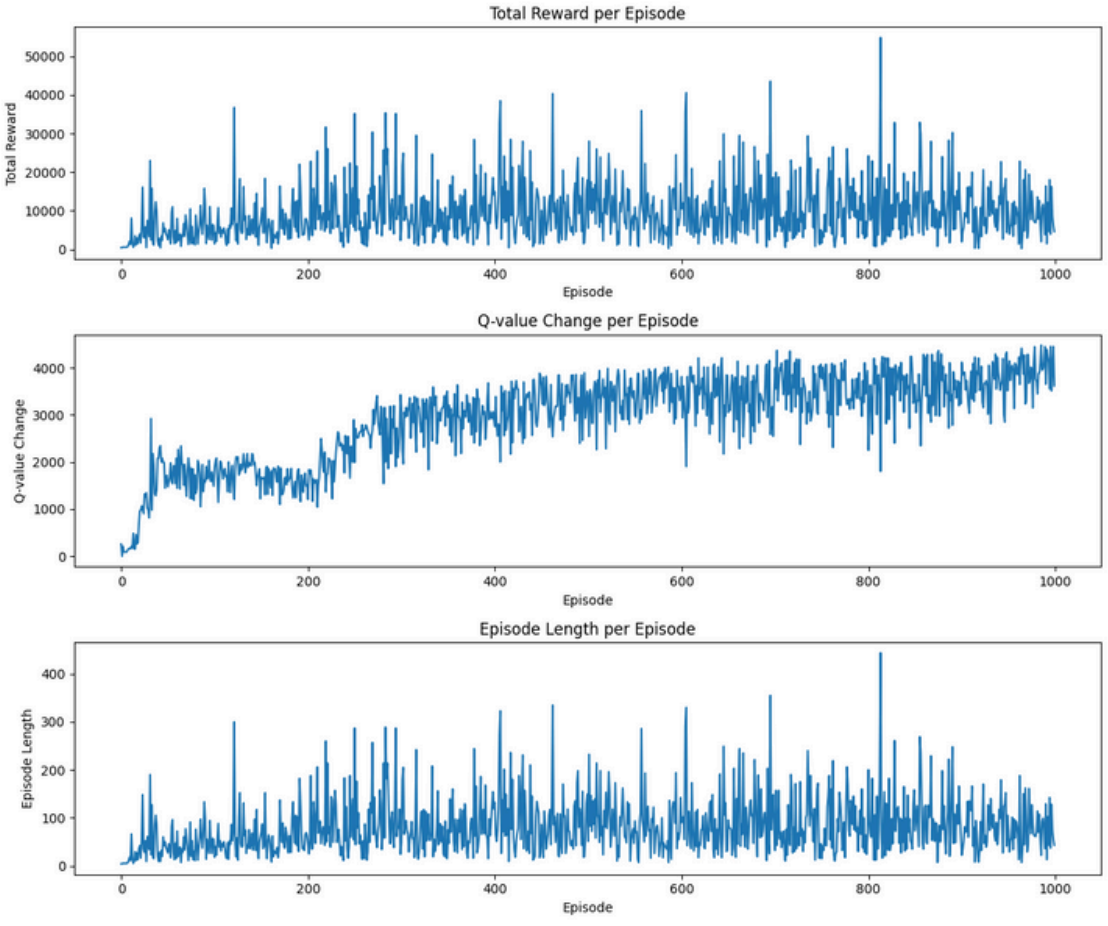
Repeat forever:
  Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ s.t. all pairs have probability > 0
  Generate an episode starting from S_0, A_0, following π
  For each pair s, a appearing in the episode:
    $G \leftarrow$ return following the first occurrence of s, a
    Append G to $Returns(s,a)$
    $Q(s,a) \leftarrow \text{average}(Returns(s,a))$
  For each s in the episode:
    $\pi(s) \leftarrow \arg\max_a Q(s,a)$

Sutton,Bartol,2018

We've trained the agent(bird) for over 1000 episodes using an epsilon greedy policy following the On Policy Monte Carlo Method for evaluation to find the optimal policy
First Visit MC
(Epsilon taken as 0.1, Gamma taken as 0.99)

The results of the training are as follows:


Q-values for each state-action pair
UP
RIGHT
NO_ACTION
Q-value
State
(3, 0)
(2, 0)
(1, 0)
(0, 0)
(2, 1)
(1, 1)
(0, 1)
(3, 1)
(2, 2)
(2, 3)
(1, 3)
(0, 3)
(4, 1)
(4, 0)
(2, 4)
(5, 1)
(2, 5)
(1, 5)
(0, 5)
(3, 3)
(3, 4)
(4, 3)
(5, 3)
(5, 0)
(3, 5)
(4, 5)
(5, 5)


Total Reward per Episode
Total Reward
Episode
Q-value Change per Episode
Q-value Change
Episode
Episode Length per Episode
Episode Length
Episode

Off Policy Monte Carlo method for policy evaluation

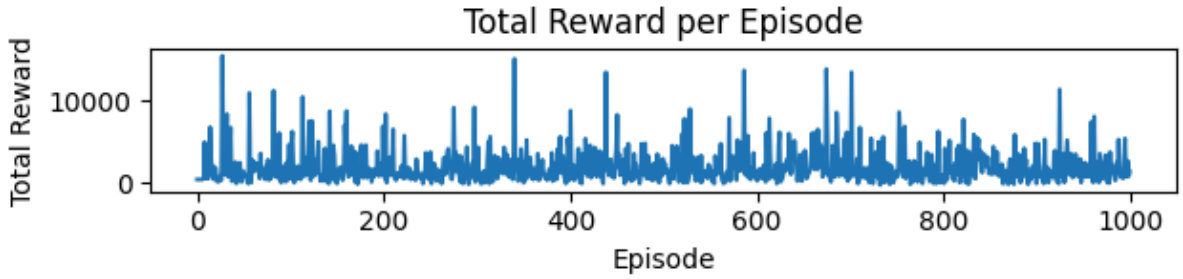
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
$Q(s,a) \leftarrow$ arbitrary
$C(s,a) \leftarrow 0$
$\mu(a|s) \leftarrow$ an arbitrary soft behavior policy
$\pi(a|s) \leftarrow$ an arbitrary target policy

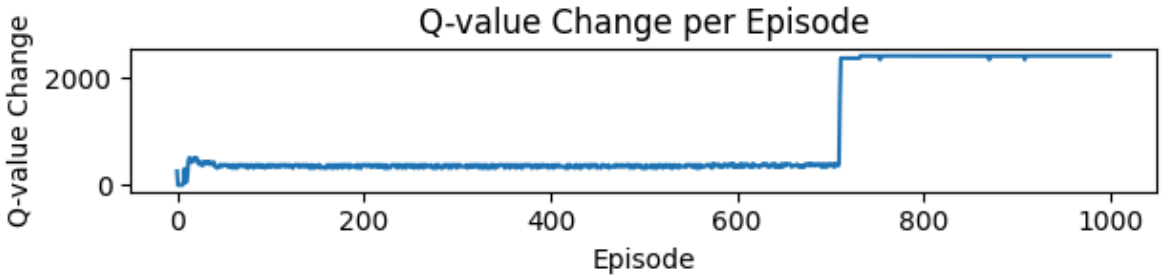
Repeat forever:
Generate an episode using μ:
$S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$
$G \leftarrow 0$
$W \leftarrow 1$
For $t = T-1, T-2, \ldots$ downto 0:
$G \leftarrow \gamma G + R_{t+1}$
$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
$W \leftarrow W \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
If $W = 0$ then ExitForLoop

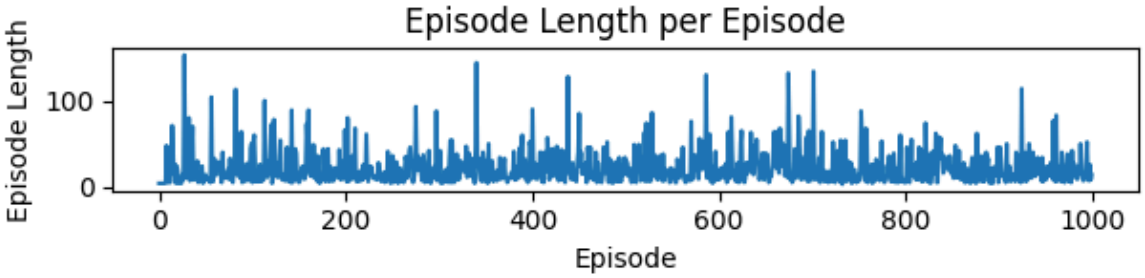
Sutton,Bartol,2018

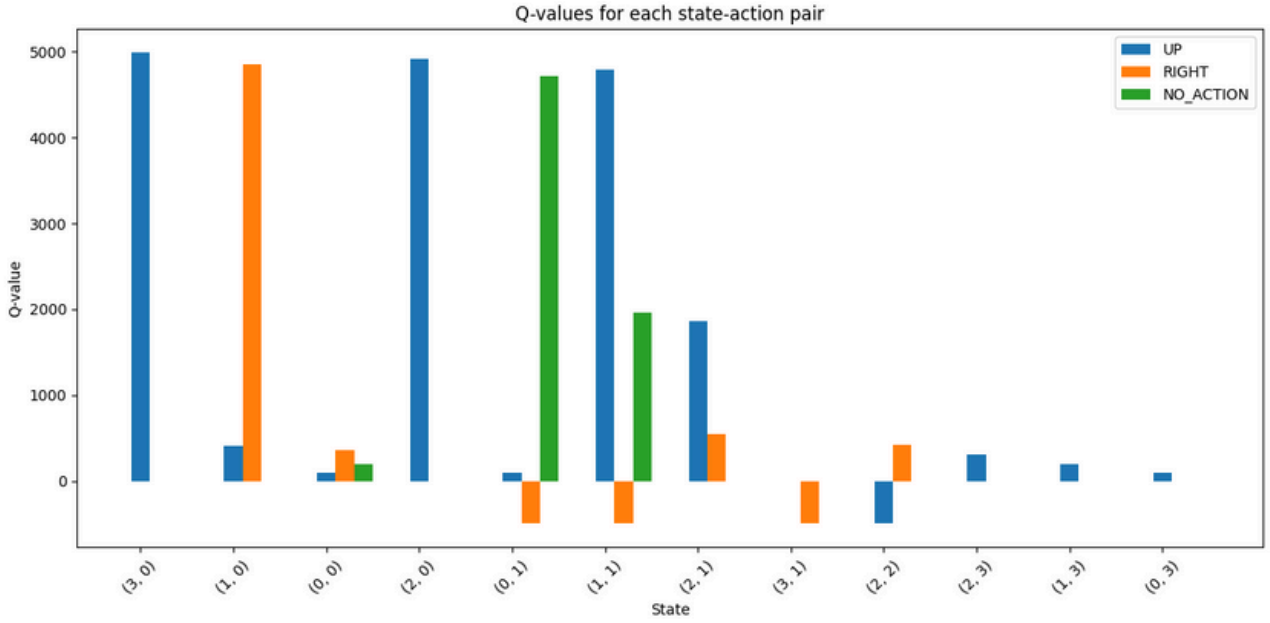
Training the agent with
target policy as: greedy policy
behaviour policy: as epsilon greedy policy with epsilon as 0.1 and gamma 0.99 with ordinary importance sampling. (First visit MC)

The results of training are as follows:









Comparison and Analysis of both variants of Monte Carlo Methods

- Observation: for an equal of 1000 episodes for both the off-policy and on-policy training we see that the average reward per episode for on-policy method training is significantly higher than the average reward per episode for off-policy method training.
- For an equal 1000 episodes each, the on-policy method has explored more states than the off-policy method
- Q-value change of on-policy is much more stable and had been increasing till the 1000th episode and that of the off-policy method is much more unstable and abrupt sudden change after ~700th episode.
- The On-policy Monte Carlo Method training has longer episodes than the off-policy methods(~ 4 times as long)
- The off-policy monte carlo method performs poor in this regard as behaviour policy generates episodes, but the Q-values are updated taking the importance sampling, here in this case it provides slow learning of the target policy of the optimal actions that may have a smaller probability of being covered in the target policy and are less frequently sampled.The behaviour policy thus may not get the truly updated Q-values to be able to explore the optimal action frequently.In other words the importance sampling ratio, weighs down on the reward brought by behavioural exploration and decreases its contribution leading to inefficient selection of greedy actions.
- On-policy monte carlo method performs better in this regard as target policy is constantly improved, episodes are longer due to increased survival with an ever improving policy encouraging exploration and ultimately reaching the goal.

- **Here On-policy Monte Carlo Method has proven to be better** than the Off-policy monte carlo method as it quickly learns to reach the end state of (3,6) much faster than off policy.