

Project: Navigation

1) Goal:

To train an agent to navigate through an environment which has lots of bananas (blue and yellow colored). The agent should learn to collect as many yellow bananas as possible. A reward of +1 is provided for a yellow banana and -1 for blue banana. A snapshot of the environment is shown in Figure 1. The goal of the agent is to maximize the reward by collecting yellow bananas.

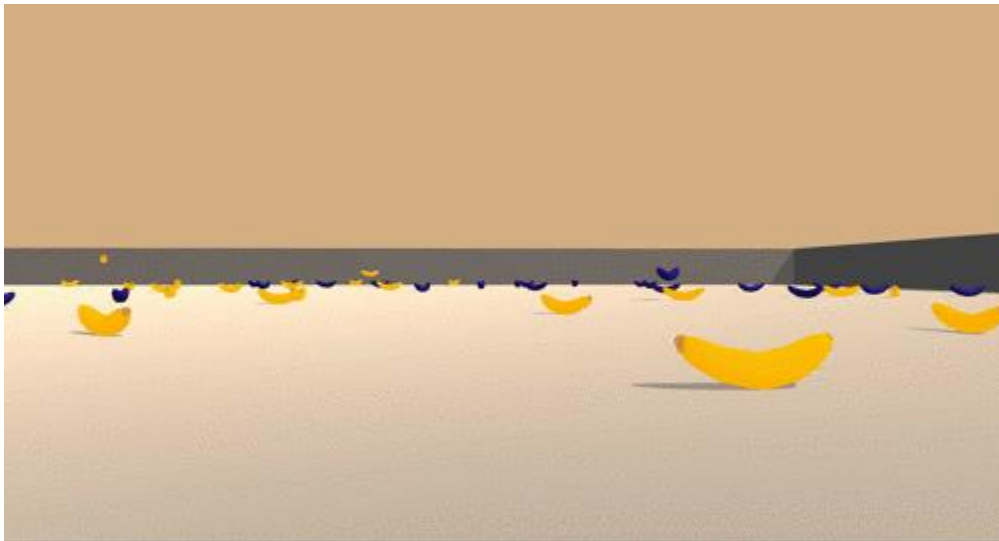


Figure 1 Snapshot of the environment

2) Algorithm:

DQN algorithm was used to solve this environment. State size is 37 (velocity of the agent and ray perception of the objects around it). Action size is 4 (left, right, forward, backward). The architecture used to train this agent is shown in Figure 2.

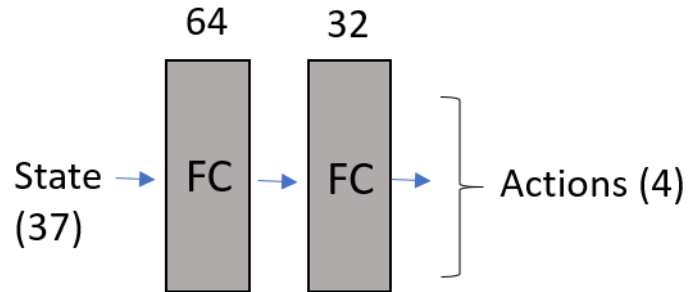


Figure 2 Architecture of the DQN Agent.

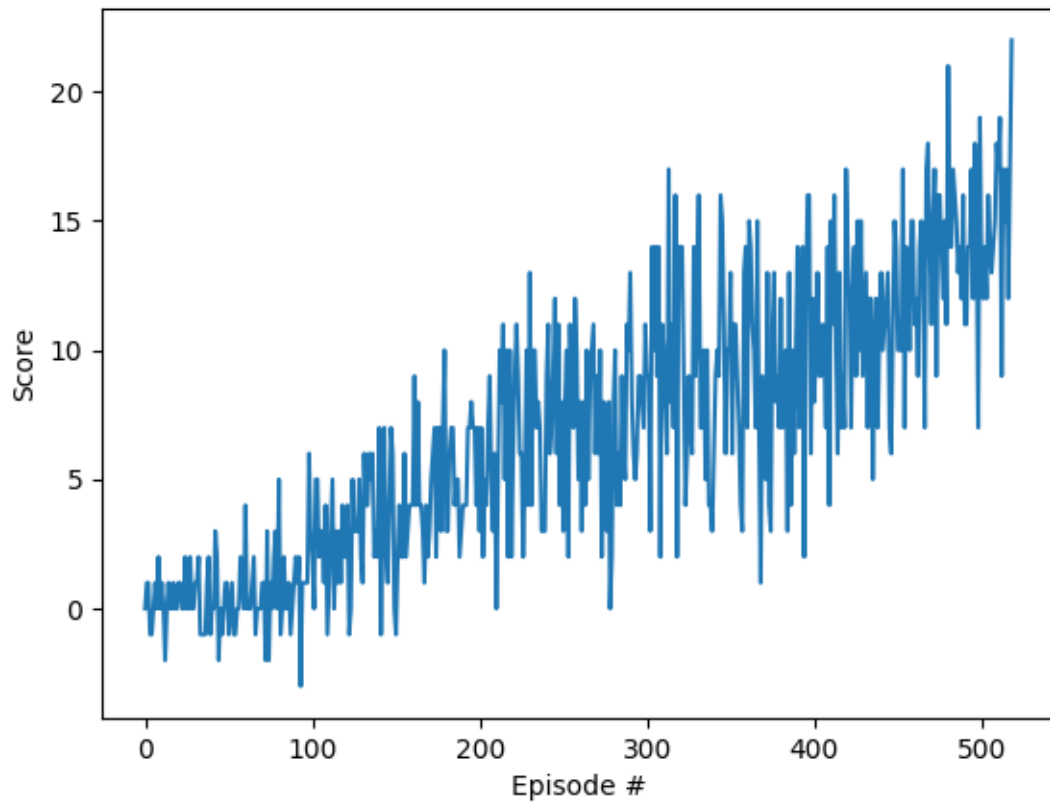
The hyperparameters for the DQN agent are as follows.

- `BUFFER_SIZE = int(1e5)` # replay buffer size
- `BATCH_SIZE = 64` # minibatch size
- `GAMMA = 0.99` # discount factor
- `TAU = 1e-3` # for soft update of target parameters
- `LR = 5e-4` # learning rate
- `UPDATE_EVERY = 4` # how often to update the network.

We collect experience (state, action, reward, next_state) tuples and store in a replay buffer. Uniform sampling is performed and at “UPDATE_EVERY” iterations, the model is updated with the corresponding gradient updates.

3) Results:

The rewards plot during training is as shown in Figure 3.



The game was solved in 419 episodes with the average reward of 13.09.

4) Future Work:

- Implement Double DQN and prioritized experience replay techniques.
- Analyze the effect of hyperparameters and architectures on DQN agent learning.
- To add videos of the trained agent and observe the real scores of the agent.