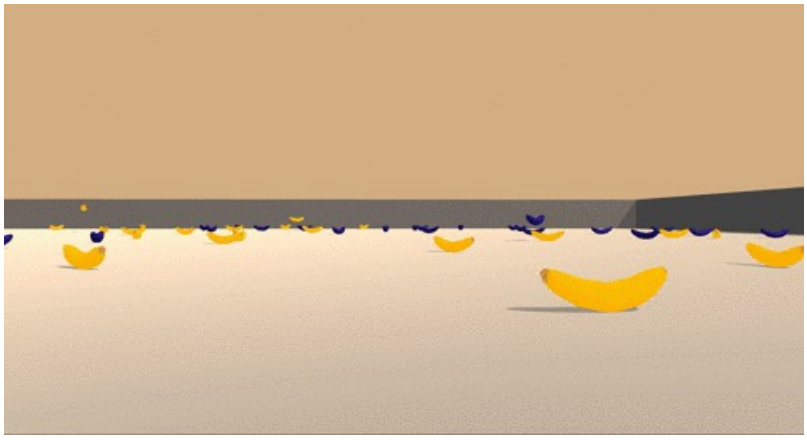
Report of
Project 1 : Navigation

Author : **Saoussen Chaabnia**

Project Description :

For this project we will use Deep Reinforcement Learning Value-Based-Methods such as Deep Q-Network (*DQN*) to train an agent to navigate the world and collect yellow bananas while avoiding blue bananas.



A reward of +1 is provided for collecting a yellow banana, and a reward of -1 is provided for collecting a blue banana. Thus, the goal of your agent is to collect as many yellow bananas as possible while avoiding blue bananas.

The state space has 37 dimensions and contains the agent's velocity, along with ray-based perception of objects around the agent's forward direction. Given this information, the agent has to learn how to best select actions. Four discrete actions are available, corresponding to:

- 0 - move forward.

- 1- move backward.
- 2- turn left.
- 3- turn right.

The task is episodic, and in order to solve the environment, your agent must get an average score of +13 over 100 consecutive episodes.

Implementation

Deep Q-Learning Algorithm is used to solve this problem. Deep Q-Learning is Value-Based Algorithm capable of learning a suitable policy in a model-free Reinforcement Learning setting.

Deep Q-learning was proved to surpass human-level performance in Atari games. It is an off-policy learning algorithm where the policy being evaluated is different from the policy being learned. Deep Q-Learning uses a deep learning network as a Function Approximator.

Two techniques contributed significantly towards stabilizing the training :

Experience Replay: In Experience Replay, we maintain a Replay Buffer of fixed size in which we store some experience tuples as we interact with the environment. After a fixed number of iterations, we sample a few experiences from this replay buffer and use that to calculate the loss and eventually update the parameters. Sampling randomly this way breaks the sequential nature of experiences and stabilizes learning. It also helps us learn from an experience multiple times and recall rare occurrences. Using Experience replay, the value based learning of the reinforcement learning problem is reduced to a supervised learning problem.

Fixed Q-targets: The idea behind fixed q-targets is to decouples the target from the parameters by fixing the parameters w used to generate the target that we call w-. w^- are the weights of a separate target network that are not changed during the learning step.

Hyperparameters:

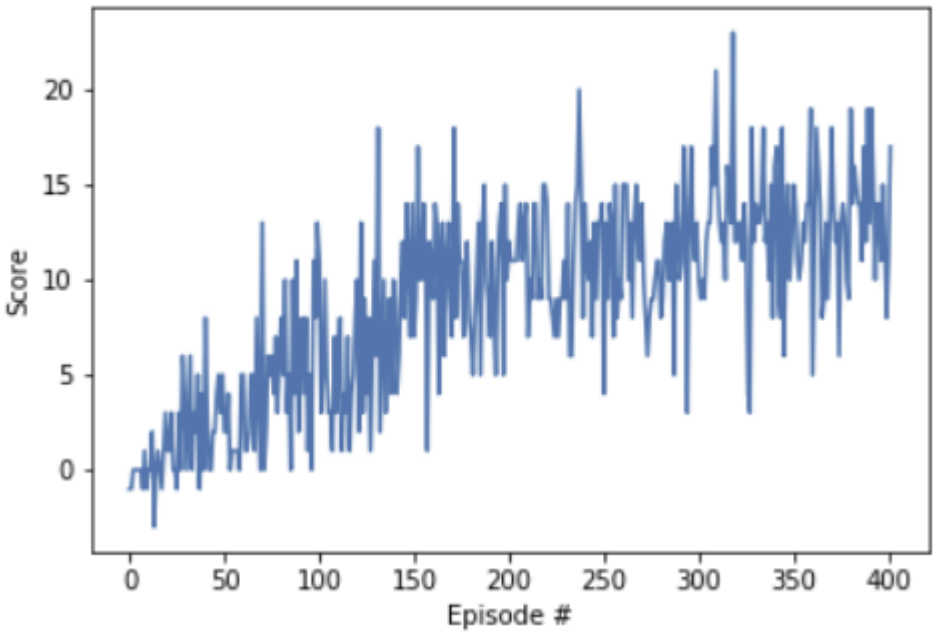
1. n_episodes : 5000
2. m_tax : 1000
3. eps_start : 1
4. eps_end : 0.1
5. eps_decay : 0.955

The Deep Learning Model:

Our model constitutes of a two-hidden layers neural network. Both layers have a 64 hidden units with ReLu activation function applied after each fully-connected layer. Adam was used as the optimizer for finding the optimal weights.

Result:

The environment was solved in 402 episodes using the Deep Q-Learning solutions. The plot of the reward DQN is shown below:



Improvement :

The results can be improved by using the following methods:

- More hyper-parameters Tuning
- Using the Double DQN Learning Algorithm

- Using the Prioritized Experience Replay Learning Algorithm
- Using the Dueling DQN Learning Algorithm