Solving Banana collector with deep reinforcement learning

Eduardo Di Santi

June 2019

Abstract

This document shows how to train a deep reinforcement learning agent with DQN to solve the Banana collector environment.

1 Introduction

The aim of the project is to train a Deep Reinforcement Learning agent to collect yellow bananas while avoiding blue bananas in a square world.

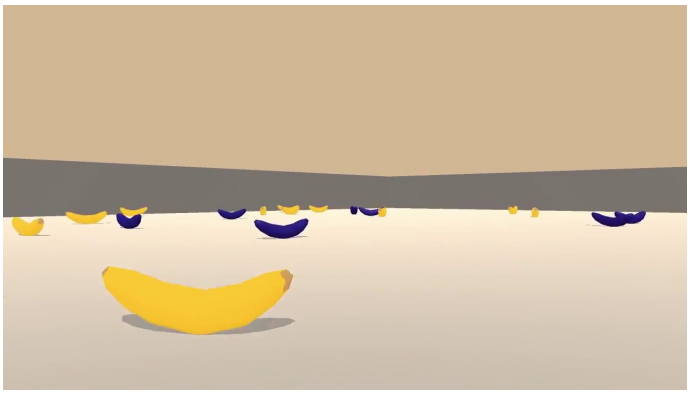

Figure 1: The environment

This environment is a version of Unity ML-Agent Banana collector environment, suitable to test reinforcement algorithms. Solving this continuous spaces environment will need the use of an action value function approximation thru a deep neural network.

2 Environment

The environment is a square world, the action state space has **37 dimensions** representing the agent speed and a ray-based[2] perception of objects around

the agent forward direction.
There reward is representing by an integer, **+1 (positive one)** for collecting a yellow banana and **-1 (negative one)** for collecting a blue banana.
There is no reward, positive or negative for moving or time.
The environment is considered solved when the agent can get a score of 13 over last 100 hundred consecutive episodes.
The available actions are four as follows:

Action	Value
Forward	0
Backward	1
Left	2
Right	3

3 Algorithm

The agent learns using a Deep Q-Learning [1] algorithm with epsilon greedy exploration.
This algorithm feed forwards a deep neural network with the state of the environment, after two hidden layers, the output layer activates is activated and the neuron with strongest signal is selected the fired one, meaning, the most probable best action is selected
.

The neural network used has the following architecture and hyper parameters

Layer	Neurons	Type	Activation	Comment
Input	37			according to the space state dimension
Hidden	32	Linear	ReLU	
Hidden	32	Linear	ReLU	
Output	4	Linear	ReLU	One for each action

The hyper parameters for the e-greedy exploration reinforcement learning are the following:

Parameter	Value	Description
Replay start size	0	Replay memory initial size
Replay size	100000	Replay memory max size
Batch size	32	Batch size * Update every
Gamma	0.99	Discount rate
Hidden layers	[32,32]	Q-Network
Tau	0.05	Soft update rate
Update every	4	Learn every 4
Learning rate	5e-05	

4 Results

As expected, the agent learned how to play fast because the simplicity of the environment; when training, the agent showed a tendency to perform mostly the forward action, which is normal because is the action to get the banana, but, for correcting the direction, during the training the back action was performed more than half of the forward ones and at least twice more than left or right.

4.1 When training

The goal is to achieve an average score of 13 over last 100 episodes, with this hyper-parameters the agent solves the environment in 319 episodes.
The agent achieve this score with following action - frequencies:

Action	Value	Frequency
Forward	0	57234
Backward	1	25189
Left	2	9690
Right	3	4187

The training history is as follows:

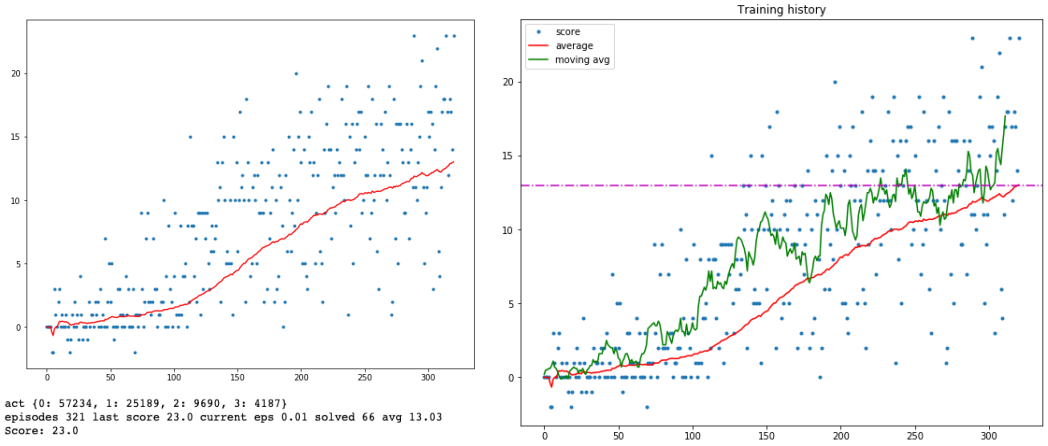

Figure 2: Training history, episodes vs rewards.

Figure 3: Training history, episodes vs rewards with moving average.

4.2 When playing

Playing 100 trials with 2000 steps each, the agent performs as expected with little plays with some scores below 13:

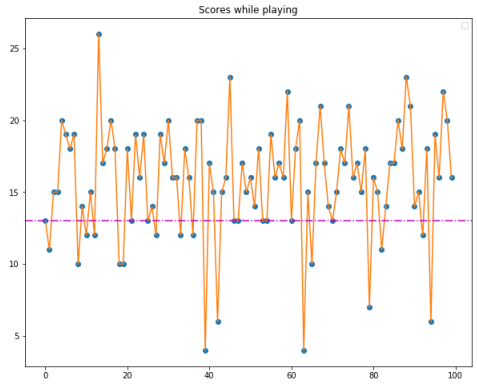

Figure 4: Playing scores 100, episodes vs rewards

5 Conclusions

The agents performs very well solving the environment in the stated conditions, but its unable to solve the situation of being surrounded by blue bananas when yellow bananas are far away; the expected behaviour in this case, is to get a blue banana to go to a yellow bananas area.
In those cases the agent just gives up by turning and moving searching for a hole in order to pass between blue bananas to avoid the negative rewards. Those cases has usually a return between 4 and 11.

6 Future work

The DQN presents a high bias problem, so in future work changing the DQN by a prioritized experience play or a dueling DQN can improve the richness of actions learned.

References

[1] Google deep mind. "Human-level control through deep reinforcement learning". In: *Nature* (2015). DOI: http://www.nature.com/doifinder/10.1038/nature14236.

[2] A.J. van der Ploeg. *Interactive ray tracing, the replacement of rasterization?* URL: `http://www.few.vu.nl/~kielmann/theses/avdploeg.pdf`.