Udacity Deep Reinforcement Learning Nano Degree
Project 1 Navigation

November 13, 2018

1 Introduction

Navigation is the first project for the Udacity Deep Reinforcement Learning specialisation. In this project students are required to develop an agent that is capable of navigating a large, square world in a manner so as to collect as many yellow bananas as possible. Furthermore, the agent should avoid blue bananas. In this regard, the agent is given a reward of +1 for collecting a yellow banana, and a reward of -1 for collecting a blue banana. The state space has 37 dimensions that include 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 given states. Four discrete actions are available, corresponding to:

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

In addition it should be noted that the task is episodic, and in order to solve the environment, the agent must get an average score of +13 over 100 consecutive episodes. This report forms part of the submission criteria for the project and it will describe the details of this submission.

2 Learning Algorithm

For the purposes of the submission, an implementation of a Double Deep Q Network (Double DQN) included. The hyperparameters used are as follow: the maximum number of training episodes is 10000, the maximum number of time steps per episode is 1000, the starting value of epsilon, for epsilon-greedy action selection, is 1.0, the minimum value of epsilon is 0.01, the multiplicative factor

(per episode) for decreasing epsilon is 0.995, the seed to initialise the pseudo random number generator is 0, the maximum size of buffer is 1e5, the size of each training batch is 64, the discount factor is 0.99, the interpolation parameter for soft updating is 1e-3, the learning rate is 5e-4, the rate of learning is set to every 4 time steps.

The deep Q network architecture used in this submission consists of a single neural network with three hidden layers consisting of 128 neurons in the first two hidden layers and 64 neurons in the final hidden layer. Each layers uses ReLu activation.

2.1 Training Results

The results obtained during training of the agent are

Episode 100 Average Score: 0.25
Episode 200 Average Score: 3.02
Episode 300 Average Score: 7.13
Episode 400 Average Score: 9.44
Episode 500 Average Score: 12.30
Episode 590 Average Score: 13.03

Environment solved in 490 episodes! Average Score: 13.03

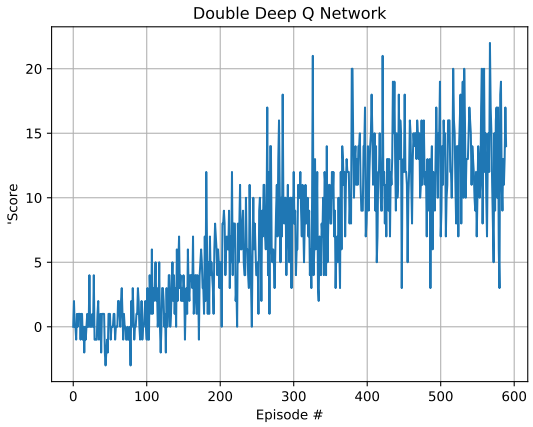

Figure 1: Training performance of Double Deep Q Network

Notes:

The Double (DQN) implementation yielded a good improvement in performance over the vanilla DQN. In other experiments Prioritised Experience replay was

investigated. However, while this did yield superior results in the beginning, towards the end of training the performance degraded. This was particularly the case when the agent had an average score of above +10.

3 Ideas for Future Work

This work can be furthered through experimentation with pixel based learning using a convolution neural network (CNN) and additionally a Duelling Deep Q Network (DDQN) as this was not investigated. Furthermore, a more rigorous approach to selecting the hyperparameters can be investigated. By example one could consider cyclical learning rates. Lastly, Prioritized Experience replay will need to be revisited to understand the behaviour that was observed during training.