Deep Q-Learning on Unity's Banana Environment

Christopher J. Miles

November 1, 2018

Abstract

This report is done in fulfillment of the *Udacity Deep Reinforcement Nanodegree*. This project implements a Deep Q-Network (DQN) model and its variants such as DDQN, Dueling DQN on the Unity *Banana* environment. There are two observational state space encodings: ray-based perception and raw pixel information.

1 Introduction

Deep reinforcement learning (in particular, the DQN algorithm) is applied to the Unity Banana environment [1]. The environment is a virtual 3D space where an agent is confined to a square region. The space is filled with randomly placed blue and yellow bananas. The objective is for the agent to collect as many yellow bananas as possible in the allotted time of 300 time steps. The agent receives a +1 reward for every yellow banana collected and a −1 for every blue banana collected.

To solve this problem, variants of the Deep Q-Learning algorithm are employed [4, 3, 2]. The details of each algorithms are discussed in the next section.

2 Algorithms

A 'vanilla' DQN [2] is implemented as follows:

3 Results

References

[1] Unity: https://unity3d.com/.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K.

Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[3] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized Experience Replay. *ICLR 2016*, pages 1–21, 2016.

[4] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling Network Architectures for Deep Reinforcement Learning. *arXiv preprint arXiv:1511.06581v3*, 2016.