Navigation Using Deep Q learning

August 26, 2018

1 Goal

The goal of the agent is to navigate (collect yellow bananas while avoid blue bananas!) in a large, square world. 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 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 - move left.

3 - move 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. In order to compare different techniques the training was stopped once the score reached an average of 13.

2 Method

Two different approaches were used in this project. Q-learning (DQN) and Double Q-learning (DDQN).

A deep Q network (DQN) is a multi-layered neural network that for a given state s outputs a vector of action values. I used the fixed target network and experience replay similar to [3] The max operator in standard Q-learning and DQN, uses the same values both to select and to evaluate an action. This makes it more likely to select overestimated values, resulting in overoptimistic value estimates. To prevent this, we can decouple the selection from the evaluation.

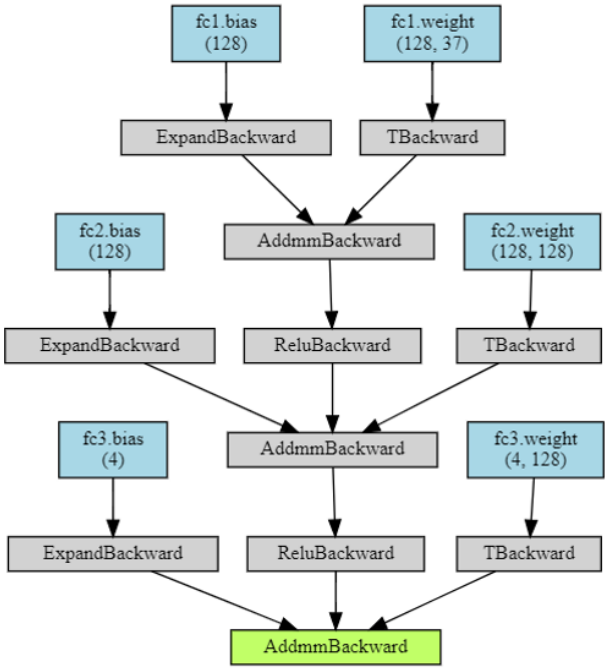


Figure 1: Neural network Architecture

In Double Q-learning, two value functions are learned by assigning experiences randomly to update one of the two value functions, resulting in two sets of weights, and '. For each update, one set of weights is used to determine the greedy policy and the other to determine its [1].

Neural network with following architecture was used to approximate Q value.

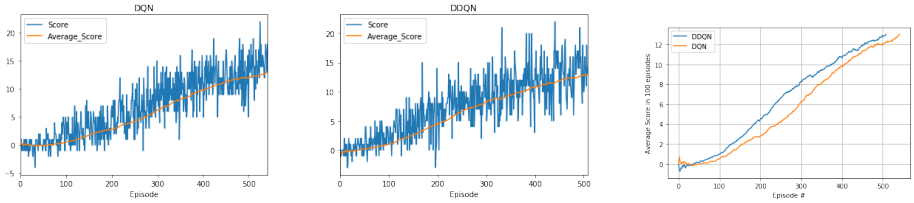
3 Results

The agent based DQN implementation could achieve a score of 13 after 542 episodes and the agent based on DDQN could achieve a score of 13 after 509 episodes.

DDQN shows slight improvement in compare to DQN. Hyper parameter tuning can also improve the performance of the agent.

4 Improvement

Other techniques could be explored to improve the current model such as Prioritized DDQN,Dueling DDQN, A3C, Distributional DQN and rainbow [2]



(a) DQN (b) DDQN (c) DQN vs DDQN

Figure 2: DDQN vs DQN average scores

References

[1] Hado V Hasselt. Double q-learning. pages 2613–2621, 2010.

[2] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*, 2017.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.