Implementation Report: Project 1 Navigation

Deep Reinforcement Learning Nanodegree

Thomas Gallien

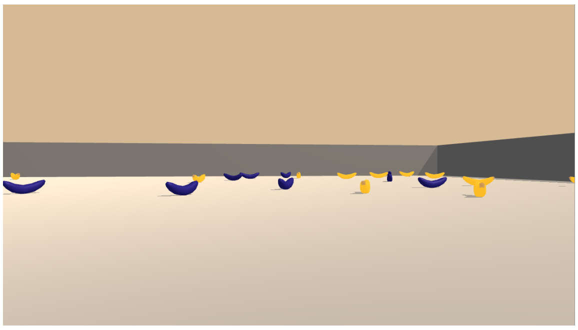
August 29, 2019

Problem Description

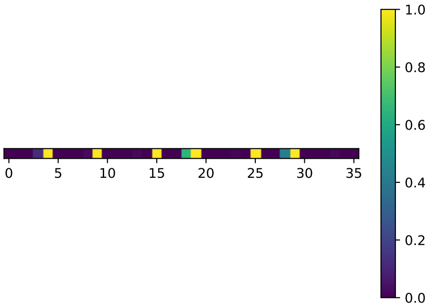
The task is to train a reinforcement agent to collect yellow bananas in a modified Unity environment (see figure 1a). The agent perceives the environment by means of 36 ray-based measurements which basically defines it's field of view (figure 1b). Along with the agent's velocity the vector of ray-based observations define the agent's 37 dimensional state space.

The action space is 4 dimensional and consists of the actions move forward ($a = 0$), move backward ($a = 1$) turn left ($a = 2$) and turn right ($a = 3$).

The environment is episodic with a fixed episodic duration. The agent gets a reward of +1 if it picks a yellow banana and -1 if it picks a blue one. If no banana is collected a reward of 0 is returned.



(a) Modified Unity banana collector environment.



(b) The agent's ray-based perception of the environment.

Algorithm

The problem is solved using a deep Q-learning agent according to [1] which basically seeks to approximate the optimal state-action function $q_*(s, a)$ by an artificial neural network. Actually, this approach takes advantage of two identical neural networks since the target values of Bellman equation depend on the weights. Mnih et. al. solved this problem by means of an identical target network using the weights of the previous iteration.

Architecture

A fully connected multilayer perceptron with two hidden layers is used for the architecture of the Q-network. Both hidden layer consist of 64 neurons and use rectified linear units for activation. No activation is used for the output layer. Essentially, this architecture is identical to the one used to solve the lunar lander environment DQN and proved to be also well suited for the particular task since more sophisticated architectures (deeper networks, sigmoid activation or significant more hidden neurons per layer) showed significantly worse performance.

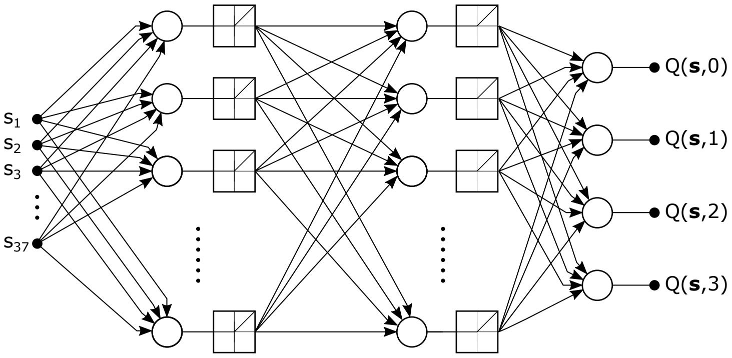


Figure 2: Chosen architecture of the agent's fully connected Q-network(s). Two hidden layers of 64 neurons each and both activated by means of rectified linear units followed by a linear output layer is used.

Hyperparameters

The table below shows the set of hyperparameters used to solve the banana collector environment. The majority of the parameters are taken from the solution of the lunar lander environment (DQN). The discount factor γ is defined to be slightly less in order to encourage the agent to take immediate reward more into account. Moreover, the learn rate has been increased.

Hyperparameter	Abbreviation	Value
Buffer size replay memory	M	10^5
Batch size	N	64
Discount factor	γ	0.98
Initial exploration parameter	ϵ_0	1
Minimum exploration parameter	ϵ_{min}	0.01
Exploration decay parameter	ϵ_{dec}	0.995
Interpolation parameter	τ	10^{-3}
Learn rate	α	10^{-3}
Update rate	l	40

Table 1: Used set of hyperparameters solving the environment utilizing a deep Q-learning agent.

Results

With the given configuration, the agent was able to achieve an average cumulative reward of 15 after 564 episodes. The figure below shows the cumulative episodic reward (score) and the filtered score using a moving average filter of 100 filter taps.

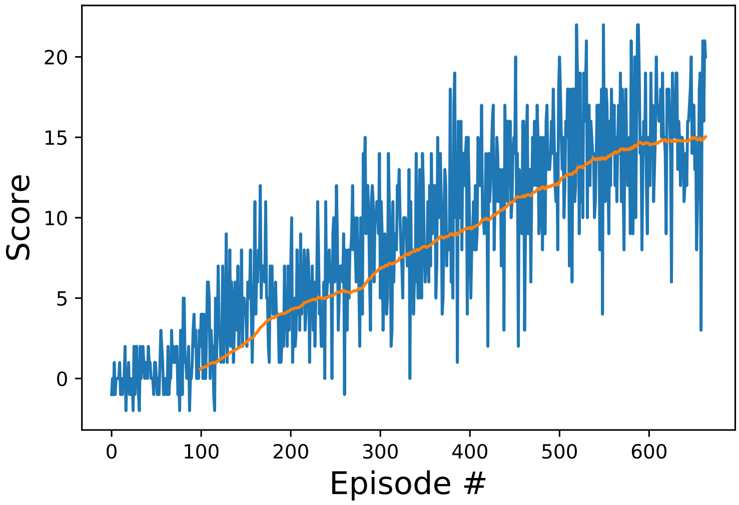


Figure 3: Performance of the DQN-learning agent in terms of episodic cumulative reward for 654 episodes. Blue: Episodic cumulative reward (score). Red: Moving average filtered (100 filter taps) score signal.

Conclusion & Outlook

Mnih et. al. deep Q-learning approach was applied to train a agent to collect yellow bananas in a modified Unity environment. Although the achieved performance meets the expectations the used algorithm tends to overestimate the optimal q-functions. In order to tackle this problem the workarounds like double Q-learning [2] will be investigated. Moreover, improvements like dueling Q-learning [3] and prioritized experience replay [4] will be investigated.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver et. al. *Human-level control through deep reinforcement learning.* Nature vol 518, pages 529-53, 2015.

[2] Hado van Hasselt, Arthur Guez and David Silver *Deep Reinforcement Learning with Double Q-learning.* Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pages 2094-2100.

[3] Ziyu Wang, Tom Schaul, Matteo Hessel et. al. *Dueling Network Architectures for Deep Reinforcement Learning.* Proceedings of the 33$^{\mathrm{rd}}$ Conference on Machine Learning.

[4] Tom Schaul, John Quan and David Silver. *Prioritized Experience Replay.* Proceedings of the 2016 International Conference on Learning Representation (ICLR), 2016.