Project1: Navigation

Mengheng Xue

June 11, 2019

1 Introduction

In this report, I will discuss the implementation of my solution to the Banana Collector environment as part of the Navigation Project. Besides standard deep Q-network (DQN) [1], I also implement double DQN to compare their performances [2].

2 Deep Q-Network Architecture

Deep Q-network maps state to Q-values for each action with a dense neural network (DNN). In the DNN structure, the input layer contains 37 nodes which represent each states and the output layer contains 4 nodes which represent each actions. For hidden layers, after experiments to achieve fast training to obtain target average score 13, my final implementation uses a DNN with three hidden layers with size [128, 64, 32]. The activation function I choose for each hidden layer is ReLU, which can result in faster training. Given a reasonable choice of ε parameter, the training process can be finished in less than 500 episodes.

3 Standard DQN vs. Double DQN

I implemented both standard DQN and double DQN in this project, and find that double DQN outperforms standard DQN to achieve fast training. The reason is that instead of overestimating Q-values, double DQN will the incidental high rewords that may be obtained by chance, and provide a more robust estimation of Q-values.

4 Exploration vs. Exploitation

For ε in ε-greedy action selection, I adopt the multiplicative decay with minimum threshold, i.e., $\varepsilon = \max(\varepsilon_{end}, \varepsilon_{decay}^{k} \cdot \varepsilon_{start})$, where k denotes the episode. The settings are $\varepsilon_{end} = 0.05$, $\varepsilon_{start} = 1$ and $\varepsilon_{decay} = 0.97$. I observe that relatively lower $\varepsilon_{decay} = 0.97$ will achieve faster training process, which indicates that the environment is relatively stable and needs less exploration.

5 Performance Result

From Fig. 1, we can see that our trained agent can achieve average score of +13 over past 100 consecutive episodes after 352 episodes.

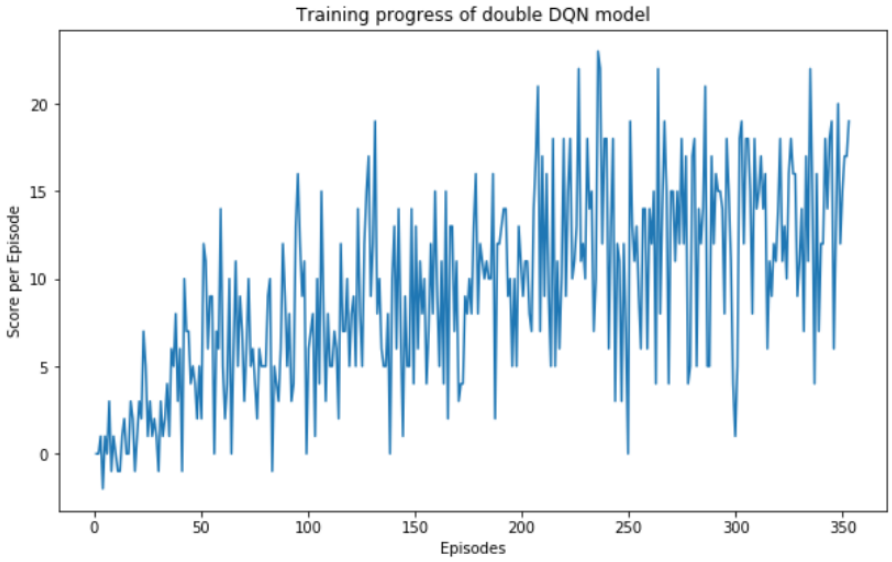

Fig. 1: Training performance of double DQN model

6 Further Work

In terms of achieving faster training performance, I think more systematic approaches for tuning hyper-parameters, e.g., grid search, may be helpful. Also, to achieve more robust estimation performance, some ensemble methods to combine other DQN models, e.g., noisy DQN and dueling DQN, with our double DQN are worth trying.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, p. 529, 2015.

[2] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.