Deep Reinforcement Learning Project 1: Navigation

*Note: Sub-titles are not captured in Xplore and should not be used

Jaeyoung Lee
Seongnam-si, Korea
jylee0121@gmail.com

***Abstract*—This project shows DQN and improvements by its several variants on 37 input state from Unity based Video Game .**

***Index Terms*—reinforcement learning, DQN, Double DQN, Dueling DQN, Noisy Network**

I. Introduction

This project shows DQN algorithms on 37 input state and improvement by Double DQN and Duelilng DQN. In Sec II over all algorithms are explained shortly. In Sec III all parameters used for individual experiment are described. Sec IV shows learning curve graphs and analyzes improvements.

II. Applied Algorithms

A. DQN

DQN [1] uses deep neural network to approximate Q value function. If system estimates Q value over input actions, it also estimate which action is the most optimal one. In DQN methods learners trains weights of Deep Q network using time difference error (1) [2] as losses to train Deep Q network.

$$\text{td error} := \max_a Q(S_t, a; \theta) - (R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta^-)) \tag{1}$$

B. Double DQN

In Double DQN [2], 'Double' means double networks. Even though vanilla DQN also uses 2 networks, one for training another for targeting, it estimates naively a action for target network in calculating time difference error. In Double DQN (1) changes as follows:

$$\begin{aligned} \text{td error} := & \max_a Q(S_t, a; \theta) \\ & -(R_{t+1} + \gamma Q(S_{t+1}, \arg\max_a Q(S_{t+1}, a; \theta); \theta^-)) \end{aligned} \tag{2}$$

C. Dueling DQN

Dueling DQN [3] reduce high variance by using specially designed neural network.

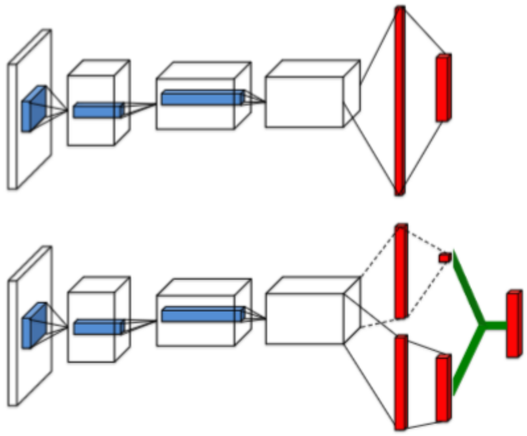

Figure 1. A popular single stream Q-network (**top**) and the dueling Q-network (**bottom**). The dueling network has two streams to separately estimate (scalar) state-value and the advantages for each action; the green output module implements equation (9) to combine them. Both networks output Q-values for each action.

Fig. 1. Dueling Network.

D. Noisy Network for Exploration

For efficient exploration, Noisy Network [4] add small noise to weights and bias of last few layers. Additionally this network learns the standard deviation size of this noise, not just adding fixed values. This strategy replace naive Epsilon-Greedy. The paper said that this noise learned over training process remove useless and redundant explorations.

E. Prioritized Replay Buffer

For fast convergence of neural network optimization, Prioritized Replay Buffer [5] maintains the magnitude of td error and uses it when sampling on experience replay buffer. The state with larger td error in the previous stochastic descent step means it gets high probability to be sampled in the next training. The paper says that this helps faster training theoretically.

III. Parameters

A. Common Parameters

Table I shows the shared parameters over all experiments.

Identify applicable funding agency here. If none, delete this.

TABLE I
SHARED PARAMATERS

Name	Value
Max episode	2,000
Ending condition	16+ points
ϵ start	1.0
ϵ end	0.01
ϵ decay per episode	0.995
Experience replay buffer size	$1e^5$
Batch size	64
Discount factor	0.99
tau for soft update of target network	$1e^{-}3$
Learning rate	$5e^{-}4$
Target network update frequency	every 4 frame

B. DQN and Double DQN

Difference between DQN and Double DQN is logical only on calculating time difference error. All constants are same in these 2 methods. Table II shows the neural network settings.

TABLE II
PARAMETERS FOR DOUBLE DQN

Name	Value
NN setup	2 x 64 fully connected layers.

C. Dueling + Double + DQN

Table III shows the specific parameters on Dueling Double DQN.

TABLE III
PARAMETERS FOR DUELING + DOUBLE DQN

Name	Value
NN setup for frontend	1x 64 fully connected layers.
NN setup for value network	1x 64 fully connected layers.
NN setup for advantage network	1x 64 fully connected layers.

D. Prioritized Replay Buffer + Dueling + Double + DQN

IV. RESULTS

As shown in Table I, ending condition of these experiments is greater than 16+ points, not 13+ points specified in the project rubric. The reason is because condition of +16 points show contributions and performance improvements by each algorithms.

A. DQN

Vanilla DQN got an average 16+ points over 100 episodes after 1288 episodes. Fig 2 shows the result.

B. Double DQN

Fig 3 shows learning curve graph of Double DQN with green color over vanilla DQN. Double DQN method reached 16+ points in 888 episodes. Double DQN saves $1288 - 888 = 400$ episodes.

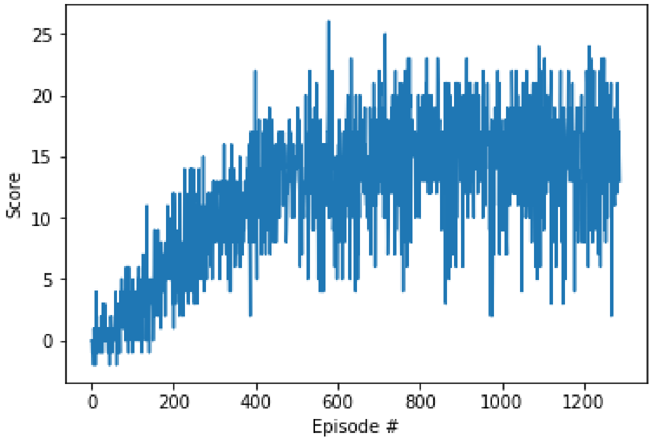

Fig. 2. Learning Curve of DQN.

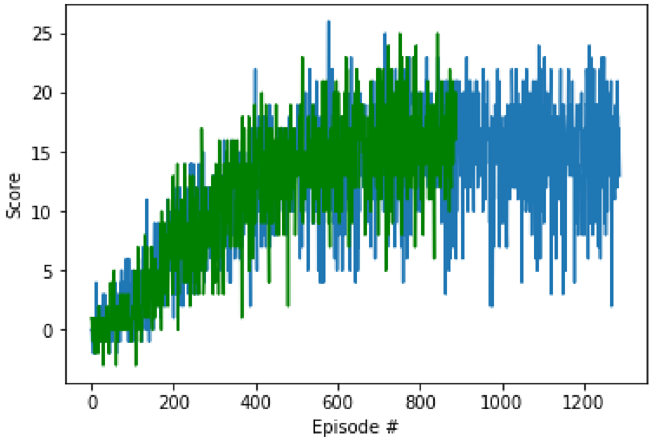

Fig. 3. Learning Curve of Double DQN.

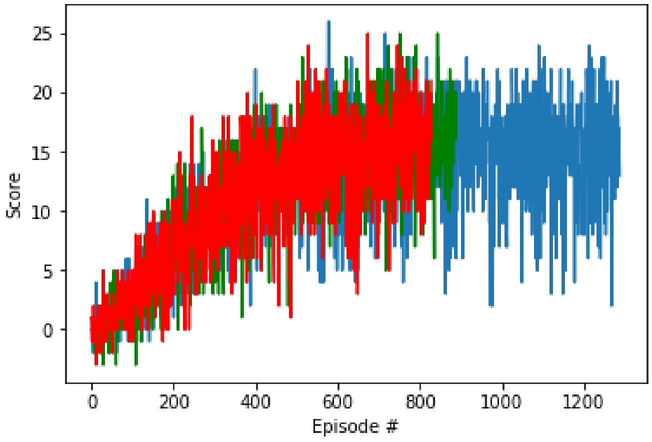

Fig. 4. Learning Curve Graph of Dueling Double DQN.

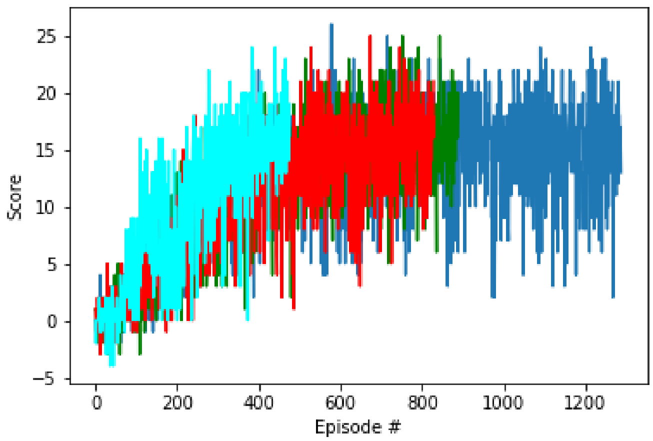

Fig. 5. Learning Curve Graph of Noisy Dueling Double DQN.

C. Dueling DQN

Fig 4 shows learning curve graph of Dueling + Double DQN method with red color. It reached 16+ points slightly 829 Episodes faster than the previous algorithm.

D. Noisy Network + DQN

Fig 5 shows learning curve graph of Noisy Network + Dueling + Double DQN method with cyan color. It reached 16+ points dramatically faster than Dueling Double DQN in 476 episode. The improvement is $829 - 476 = 353$ episodes.

E. Comparison and Analysis on the Results

Noisy Dueling Double DQN improves RL system $1288 - 476 = 812$ episodes faster than Vanilla DQN. Even though multiple tests and statistical analysis should be followed, Saving more than 800 episodes means certain improvement clearly.

V. Future work

Implementation of Prioritized Replay Buffer failed to show any improvements. As future work Prioritized Replay Buffer should be implemented again.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013.

[2] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2094–2100, AAAI Press, 2016.

[3] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 1995–2003, PMLR, 20–22 Jun 2016.

[4] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, "Noisy networks for exploration," *CoRR*, vol. abs/1706.10295, 2017.

[5] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *CoRR*, vol. abs/1511.05952, 2015.