

Lab 3: DASH Based on Deep Q-Learning

Kyeong Soo (Joseph) Kim
 Department of Electrical and Electronic Engineering
 Xi'an Jiaotong-Liverpool University
 25 May 2020

I. INTRODUCTION

In this Lab, you are to carry out simulation experiments for dynamic adaptive streaming over HTTP (DASH) where you optimize the quality of experience (QoE) of video streaming based on deep Q-learning framework.

You need to submit a Lab report and program source code with any supporting files through the ICE by the end of Sunday, 14 June 2020.

II. DASH BASED ON DEEP Q-LEARNING

Fig. 1 shows the architecture of D-DASH, an adaptive streaming framework for DASH based on deep Q-learning [1].

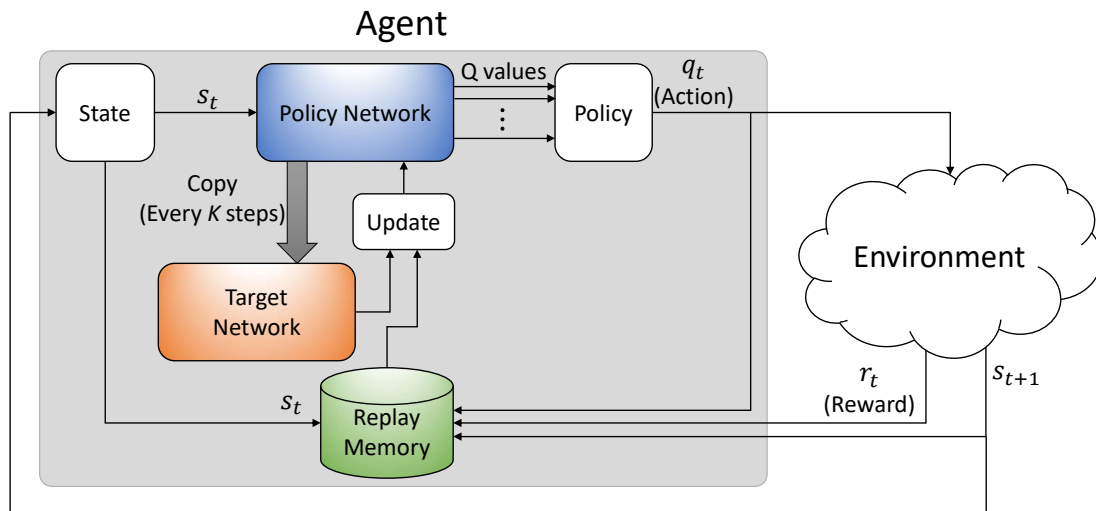


Fig. 1. D-DASH architecture [1].

You are to implement step by step DASH simulation based on the D-DASH architecture shown in Fig. 1 to investigate the improvement of the quality of experience (QoE) in video streaming made by each step, which is a good example of *cross-layer design* and *joint source/channel coding (JSCC)*.

- 1) Run DASH simulation without *target network* and *replay memory*, where the policy network is based on the traditional feedforward neural network (FNN). The implementation of this case will be provided as a baseline for comparison with the following steps.
- 2) Implement the policy network based on the recurrent neural network (RNN) using long short-term memory (LSTM) cells, replacing the FNN in step 1, and run DASH simulation.

- 3) Implement the target network, which has the same architecture as the policy network, and run DASH simulation.
- 4) Implement the experience replay [2] technique based on the replay memory and run DASH simulation.

III. TASK: DASH SIMULATION

The following files are provided on ICE for this task:

- Python program files:
 - *d-dash.py*: Baseline implementation of D-DASH without target network and replay memory.
- Data files in Python NumPy format:
 - *bandwidths.npy*: Channel bandwidths in kbps.
 - *bear.npy*: Segment bitrates in kbps for DASH video “Bear”.
 - *bigbuckbunny.npy*: Segment bitrates in kbps for DASH video “Big Buck Bunny”.
 - *test.npy*: Segment bitrates in kbps for DASH video “Test”.

You need to submit a Lab report and program source code summarizing the following activities for this task based on the provided files:¹

- #1 **[40 points]** Implement the policy network based on the RNN using LSTM cells, replacing the FNN in the baseline implementation, run DASH simulation, and obtain the QoE results.
- #2 **[20 points]** Implement the target network on top of the the results from #1, run DASH simulation, and obtain the QoE results.
- #3 **[15 points]** Implement the experience replay technique on top of the results from #2, run DASH simulation, and obtain the QoE results.
- #4 **[5 points]** Generate a plot comparing all the results from the baseline and #1–#3.
- #5 Discuss the following points based on the results from your analysis above:
 - **[10 points]** Are there any advantages of using RNN with LSTM cells over FNN in terms of performance (i.e., QoE) and implementation? If so, explain them.
 - **[5 points]** Is there any improvement in terms of performance by using the target network? If so, explain why.
 - **[5 points]** Is there any improvement in terms of performance by using the experience replay technique? If so, explain why.

REFERENCES

- [1] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, “D-dash: A deep Q-learning framework for DASH video streaming,” *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 703–718, Dec. 2017.
- [2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, and A. K. Fidjeland, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529—533, Feb. 2015.

¹Further details of implementing and running the simulation not discussed in this document will be provided during the Lab sessions.