Supplementary Material for "Adaptive Learning Recommendation Strategy Based on Deep Q-learning"

Chunxi Tan, Ruijian Han, Rougang Ye and Kani Chen*

Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

*Corresponding Author: makchen@ust.hk (Kani Chen)

Supplementary Material for "Adaptive Learning Recommendation Strategy Based on Deep Q-learning"

We provide the full algorithm and training details below.

- $\boldsymbol{\theta}$ is updated for each iteration following the gradient of $L(\boldsymbol{\theta})$, using the method of stochastic gradient descent with Adam (Kingma & Ba, 2014).
- In order to mitigate the overestimate issue in predicting Q-values (Thrun & Schwartz, 1993) for the DQN method, we use the approach proposed by Hasselt, Guez, and Silver (2016) and rewrite the target as

$$y_i = R(t) + Q_{i-1}(\hat{\boldsymbol{s}}(t+1), \arg\max_a Q_i(\hat{\boldsymbol{s}}(t+1), a, t+1; \boldsymbol{\theta}_i), t+1; \boldsymbol{\theta}_i^-).$$

It decomposes the max operation in the target into the action selection, resulting in more stable and reliable results.

- Hyperparameters setting in the training: begin exploration rate $\epsilon_0 = 0.9$, end exploration rate $\epsilon_M = 0.05$, decay of rate $\tau = 3000$, learning rate $\eta = 0.01$, memory capacity $|D| = 30000$, batch size $n = 128$.
- Since the policy is stochastic, we conduct several independent networks to reduce effects from randomness.
- In the toy experiment, we conduct 15,000 episodes in the training and test the policy over 300 trajectories, while we generate 30,000 episodes in the training and test on over 1,000 trajectories in other experiments.

Algorithm 1 Double DQN with assessment model

Input: Begin exploration rate ϵ_0; Decay of rate τ; End learning rate ϵ_M; Learning rate η; Initialized knowledge state $\boldsymbol{s}(0)$; Batch size n;

Output: Action-Value function Q;

Initialize replay memory capacity D to capacity N;

Initialize action-value function Q with random weights $\boldsymbol{\theta}$ and target action-value function $\hat{Q}$ with weights $\boldsymbol{\theta}^{-} = \boldsymbol{\theta}$;

for episode = 1,. . . , M **do**

  Initialize knowledge state $\hat{\boldsymbol{s}}(0) = \boldsymbol{s}(0)$;

  $\epsilon = \epsilon_M + (\epsilon_0 - \epsilon_M) * \exp\{-\text{episode}/\tau\}$;

  for $t = 0, \ldots, T-1$ **do**

    Set $r(t) = \begin{cases} R(T) + R(T-1), & \text{if } t = T-1 \\ R(t), & \text{otherwise} \end{cases}$

    if have not already taking action a_s **then**

      With probability ϵ select a random action $a(t)$;

      otherwise select $a(t) = \arg\max_a Q(\hat{\boldsymbol{s}}(t), a, t; \boldsymbol{\theta})$;

    else

      $a(t) =$ 'null'

    Execute action $a(t)$ in emulator, observe reward $r(t)$ and estimate the next knowledge state $\hat{\boldsymbol{s}}(t+1)$ through assessment model;

    Store transition $(\hat{\boldsymbol{s}}(t), a(t), r(t), \hat{\boldsymbol{s}}(t+1))$ in D;

    Sample random minibatch with size n of transitions $(\hat{\boldsymbol{s}}(t), a(t), r(t), \hat{\boldsymbol{s}}(t+1))$ from D;

    Select $a' = \arg\max_a Q(\hat{\boldsymbol{s}}(t+1), a, t+1; \boldsymbol{\theta})$;

    Set $y(t) = \begin{cases} r(t), & \text{if } t = T-1 \\ r(t) + \hat{Q}(\hat{\boldsymbol{s}}(t+1), a', t+1; \boldsymbol{\theta}^{-}), & \text{otherwise} \end{cases}$

    Perform a gradient descent step on $(y(t) - Q(\hat{\boldsymbol{s}}(t), a(t), t; \boldsymbol{\theta}))^2$ with respect to the network parameters $\boldsymbol{\theta}$ with learning rate η;

    Every C steps reset $\hat{Q} = Q$;

  end for

end for

  return $Q(\hat{\boldsymbol{s}}(t), a, t; \boldsymbol{\theta})$;

References

Hasselt, H. v., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the thirtieth aaai conference on artificial intelligence* (pp. 2094–2100). AAAI Press. Retrieved from http://dl.acm.org/citation.cfm?id=3016100.3016191

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv preprint arXiv:1412.6980*. Retrieved from http://arxiv.org/abs/1412.6980

Thrun, S., & Schwartz, A. (1993). Issues in using function approximation for reinforcement learning. In D. T. J. E. M. Mozer P. Smolensky & A. Weigend (Eds.), *Proceedings of the 1993 connectionist models summer school.* Mahwah, NJ: Erlbaum Associates.