ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201812.00127

Postprint of Dueling Deep Q-Network Learning Based on Prioritized Experience Replay

Authors: Zhou Yaoyao, Li Ye

Date: 2018-12-13T00:00:00+00:00

Abstract

To reduce the training time of Deep Q-Network algorithms, a DQN method combining prioritized experience replay mechanism with dueling network architecture is adopted to investigate two classic control problems on the OpenAI Gym platform: CartPole and Mountain Car, wherein the experience replay employs a rank-based mechanism and the dueling structure utilizes deep neural networks. Simulation results indicate that, compared with conventional DQN, dueling network architecture-based DQN, and prioritized experience replay-based DQN, the proposed method demonstrates superior learning performance with the shortest training time. Simultaneously, the influence of algorithm parameters on learning performance is analyzed in detail, providing valuable reference for practical application of this method.

Full Text

Preamble

Dueling Deep Q Network Learning with Rank-Based Prioritized Experience Replay

Zhou Yaoyao, Li Ye
(School of Optical-Electrical & Computer Engineering, University of Shanghai for Science & Technology, Shanghai 200093, China)

Abstract: To reduce the training time of deep Q-network algorithms, this paper investigates a DQN method that combines prioritized experience replay with the dueling network architecture, applied to two classic control problems on the OpenAI Gym platform: Cart Pole and Mountain Car. The experience replay employs a rank-based mechanism, while the dueling architecture utilizes deep neural networks. Simulation results demonstrate that compared with regular DQN, DQN with dueling network, and DQN with prioritized experience replay, the proposed method achieves better learning performance with the shortest

training time. Additionally, the impact of algorithm parameters on learning performance is analyzed in detail, providing valuable reference for practical application.

Key words: reinforcement learning; deep Q network; dueling network; rank-based prioritized experience replay

0 Introduction

In reinforcement learning, an agent interacts with its environment, continuously adjusting its behavior based on observed feedback to improve performance. Despite many successful applications, reinforcement learning has traditionally been limited to low-dimensional problems due to sampling and computational complexity issues. With the development of deep learning, deep neural networks can reliably represent high-dimensional data in low-dimensional form [1], addressing the computational complexity challenges of reinforcement learning [2].

Mnih et al. [3] first combined reinforcement learning with deep learning, proposing the deep Q-network (DQN) method that attempts to learn optimal control policies directly from raw sensory data such as images and speech. To address the issues of correlated training data and non-stationary data distributions, they employed a uniform random experience replay strategy. To mitigate the overestimation problem in DQN, Hasselt et al. [4] proposed double DQN, which uses separate neural networks for action selection and evaluation. Schaul et al. [5] introduced prioritized experience replay, which prioritizes experiences that provide greater environmental learning benefit, enabling faster agent adaptation. Recognizing that different actions have varying importance in a given state, Wang et al. [6] proposed the dueling network architecture, which uses two separate streams to estimate state value and action-dependent advantage. This approach eliminates the need to evaluate every action option for each state while further improving learning performance when combined with experience replay.

The definition of priority significantly impacts learning performance. When using proportional prioritization, the sampling probability is directly proportional to the temporal difference error of experiences, causing experiences with larger errors to be replayed more frequently. This makes learning vulnerable to adverse effects from outlier temporal difference errors. In contrast, rank-based prioritized experience replay offers greater robustness. This paper investigates a dueling deep Q-network with rank-based prioritized experience replay for two classic control problems, employing a rank-based mechanism for experience replay and deep neural networks in the dueling architecture, while analyzing the influence of algorithm parameters on learning performance.

1 Deep Q Network

Reinforcement learning problems can typically be formulated as Markov decision processes and solved using Q-learning algorithms [7]. When an agent selects an action, the environment provides feedback as a state-action reward. The agent continuously learns and optimizes an iteratively computable Q-function, aiming to find the optimal policy for each state that maximizes expected return. The Q-value update is given by:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

where $Q(S_t, A_t)$ represents the expected return value when the agent selects action A_t in state S_t; R_{t+1} is the immediate reward for selecting action A_t in state S_t; $\max_a Q(S_{t+1}, a)$ denotes the maximum expected return value for all possible actions in state S_{t+1}; γ is the discount factor reflecting the influence of future rewards relative to immediate rewards (lower values indicate less influence); and α is the learning rate.

The Q-learning algorithm uses a Q-table to record and repeatedly update Q-values for each state-action pair. However, in practice, the state space may be too large for tabular storage, necessitating value function approximation. The value function can be linear or nonlinear, such as a neural network—referred to as a Q-network.

Learning from high-dimensional sensory data like video and speech has long been a challenge in reinforcement learning [3]. The performance of previous reinforcement learning systems heavily depended on the quality of manually designed features, whereas deep learning enables extraction of high-level features from raw sensory data. Consequently, incorporating deep learning structures such as convolutional and recurrent neural networks into reinforcement learning has become a trend.

Deep Q-networks use $y_t = R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta^-)$ as the target Q-value and define the loss function L based on the deviation between the Q-network' s output and the target Q-value:

$$L(\theta) = \mathbb{E}\left[\left(y_t - Q(S_t, A_t; \theta) \right)^2 \right]$$

where θ represents the network parameters, a denotes the action taken after state S_t, and $Q(S_t, A_t; \theta)$ is the Q-network' s output value. The deep Q-network weights can be updated using stochastic gradient descent.

2 Dueling Network Architecture

In reinforcement learning, estimating the value of each state is necessary, but for many states, evaluating every action' s value is unnecessary. The dueling

network architecture separates the representation of state value from the action advantages under that state. The state-action value function $Q^{\pi}(s,a)$ represents the expected return when selecting action a in state s under policy π, while the state value function $V^{\pi}(s)$ represents the value of state s—the expected value of all actions generated by policy π in that state. Their difference defines the advantage of selecting action a in state s:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$$

The dueling network therefore employs two data streams: one outputs the state value $V(s;\theta,\beta)$, and the other outputs the action advantage $A(s,a;\theta,\alpha)$, where θ represents the network parameters processing the input layer, and α and β are parameters for the two streams respectively. The output of the dueling deep Q-network is:

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + A(s,a;\theta,\alpha)$$

Since the network directly outputs Q-values without explicit knowledge of state value V and advantage A, we enforce that the advantage estimate for the selected action is zero, modifying the Q-value representation as:

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \max_{a' \in \mathcal{A}} A(s,a';\theta,\alpha)\right)$$

In practical applications of the dueling network architecture, the average of action advantages is typically used instead of the maximum to ensure performance while improving optimization stability [6].

3 Prioritized Experience Replay Mechanism

The uniform random sampling strategy used in DQN is suboptimal. During learning, experiences with significant rewards—such as successful attempts or failure lessons—may persist in memory. Frequently replaying these experiences helps the agent recognize the consequences of correct or incorrect behaviors, enabling continuous self-correction. The key to prioritized experience replay lies in determining experience importance. One approach measures importance directly based on the temporal difference error generated by state-action transitions [8,9]. This paper employs a rank-based priority mechanism, defining experience priority as:

$$p_t = \frac{1}{\mathrm{rank}(t)}$$

where rank(t) is the rank of experience t when sorted by absolute temporal difference error in descending order. The sampling probability for experience t is then defined as:

$$P(t) = \frac{p_t^\alpha}{\sum_k p_k^\alpha}$$

where n is the replay memory size, and parameter $\alpha \in [0, 1]$ controls the degree of prioritization (with $\alpha = 0$ representing uniform sampling).

Since experiences with high temporal difference errors are replayed frequently, certain states may be visited too often, reducing experience diversity and causing network overfitting. This can be corrected through importance sampling weights w [10]:

$$w_i = \left(\frac{1}{N \cdot P(i)}\right)^\beta$$

where p_i is the sampling probability of experience i, $p_{\min}$ is the minimum sampling probability, and parameter β controls the correction degree. The Q-network loss function L is defined as:

$$L = \sum_i w_i \left(y_i - Q(S_i, A_i; \theta, \alpha, \beta)\right)^2$$

where y_i represents the target Q-value at time i, and $Q(S_i, A_i; \theta, \alpha, \beta)$ is the dueling Q-network' s output Q-value.

Action selection follows an ϵ-greedy policy. Initially, when the agent is unfamiliar with the environment, it selects actions randomly. As experience accumulates, the probability of random action selection decreases, biasing the agent toward greedy action selection that maximizes expected return.

4 Algorithm Description

Based on the above, the complete dueling deep Q-network with rank-based prioritized experience replay algorithm (dueling DQN-PR) is presented:

1) For each episode: 2) Initialize environment and obtain initial state S_0. 3) For each step in the episode: 4) Select action using ϵ-greedy policy: either randomly choose an action A_t, or $A_t = \arg\max_a Q(S_t, a; \theta, \alpha, \beta)$. 5) Execute action and observe environment feedback R_t and new state S_{t+1}. Compute temporal difference error:

$$\delta_t = R_t + \gamma \max_a Q(S_{t+1}, a; \theta^-, \alpha^-, \beta^-) - Q(S_t, A_t; \theta, \alpha, \beta)$$

6) Sort temporal difference errors δ_t in descending order to obtain rank(t).
7) Compute priority of state-action transition experience:

$$p_t = \frac{1}{\text{rank}(t)}$$

Compute sampling probability $P(t)$ and importance weight w_t:

$$P(t) = \frac{p_t^\alpha}{\sum_k p_k^\alpha}, \quad w_t = \left(\frac{1}{N \cdot P(t)}\right)^\beta$$

Store transition experience (S_t, A_t, R_t, S_{t+1}) in replay memory with probability $p(t)$. 8) Sample from replay memory according to sampling probabilities. 9) Minimize loss function:

$$L = \sum_t w_t \left(y_t - Q(S_t, A_t; \theta, \alpha, \beta)\right)^2$$

where $y_t = \begin{cases} R_t & \text{for terminal states} \\ R_t + \gamma \max_a Q(S_{t+1}, a; \theta^-, \alpha^-, \beta^-) & \text{otherwise} \end{cases}$

Update network parameters. 10) Every T steps, update target network parameters with dueling Q-network parameters.

The dueling network architecture increases space complexity by adding two streams to compute state value and action advantage separately. With action space dimension M, the additional storage overhead is $\mathcal{O}(M)$, making total storage overhead $\mathcal{O}(M) + \mathcal{O}(1)$. Rank-based prioritized experience replay uses an array-based binary heap to store prioritized experiences, achieving $\mathcal{O}(\log N)$ time complexity for sampling and updating in a replay memory of capacity N.

5 Experiments

5.1 Experimental Setup

To validate the proposed algorithm, we investigate two classic control problems: CartPole-v0 and MountainCar-v0 [11]. As shown in [Figure 1: see original paper], the Cart Pole scenario involves a cart moving left and right to keep a pole balanced upright, while the Mountain Car scenario involves a car at the bottom of a valley between two hills attempting to reach a marked peak on one hill. We implement the simulation environment using the OpenAI Gym reinforcement learning toolkit and TensorFlow 1.0 deep learning platform, with Python 3.5 as the programming language. The replay memory capacity is set to 50, with 32 experiences sampled per batch. The deep neural network is a four-layer fully connected network: for Cart Pole, the first and second hidden layers contain 40 and 30 neurons respectively; for Mountain Car, they contain

90 and 20 neurons. ReLU activation functions are used, with RMSProp for gradient descent optimization. During training, action selection follows an ϵ-greedy policy with initial $\epsilon = 0.5$ and discount factor $\gamma = 0.9$.

5.2 Results and Analysis

We compare DQN, DQN with rank-based prioritized experience replay (DQN-PR), dueling DQN, and dueling DQN-PR algorithms on both Cart Pole and Mountain Car scenarios. The results are shown in [Figure 2: see original paper]. Compared with standard DQN, DQN-PR prioritizes experiences with high temporal difference error for network updates, enabling faster convergence and reduced training time. The dueling DQN architecture updates state values as Q-values are updated, unlike DQN where only one action' s value is updated per iteration, further reducing training time. Our combined approach achieves the shortest training time.

[Figure 3: see original paper] and [Figure 4: see original paper] present training times for dueling DQN-PR under different parameter settings. Figures 3(a) and 4(a) show the effect of ϵ reduction rate. Larger ϵ increments (faster reduction of random action probability) mean the agent more quickly adopts greedy action selection after gaining environmental knowledge, accelerating learning and reducing training time. However, random action selection remains important because exploring unknown actions benefits Q-value updates and yields better policies. Figures 3(b) and 4(b) illustrate learning rate α effects, tested at 0.001, 0.005, and 0.01. For these simple environments, smaller learning rates yield more stable learning and reduced training time. For complex environments, however, learning rate selection requires careful tuning: too small values slow convergence, while too large values cause loss function oscillation. Figures 3(c) and 4(c) show target network update frequency effects (every 200, 500, and 800 steps). In Cart Pole, higher update frequency reduces training time by accelerating network convergence. In Mountain Car, lower update frequency reduces training time because the uphill process provides position-dependent rewards (higher positions yield greater rewards), and learning longer-term climbing sequences benefits velocity selection. Figures 3(d) and 4(d) demonstrate discount factor γ effects. Larger γ values increase the influence of future rewards on current expected returns, with higher proportions of predicted future rewards facilitating environmental learning and reducing training time. Environments with strong temporal correlations benefit from larger γ values.

6 Conclusion

This paper investigates a dueling deep Q-network with rank-based prioritized experience replay for two classic control problems on the OpenAI Gym platform: Cart Pole and Mountain Car. Experimental results demonstrate that the proposed method effectively reduces training time. A detailed analysis of key algorithm parameters' impact on learning performance provides valuable reference for practical application.

References

[3] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning [J]. Computer Science, 2013.
[4] Hasselt H V, Guez A, Silver D. Deep reinforcement learning with double Q-learning [J]. Computer Science, 2015.
[5] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay [J]. Computer Science, 2015.
[6] Wang Ziyu, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning [C]//Proc of the 33rd International Conference on Machine Learning. 2016.
[1] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述 [J]. 计算机学报, 2018, 41(1): 1-27. (Liu Quan, Zhai Jianwei, Zhang Zongchang, et al. A survey on deep reinforcement learning [J]. Chinese Journal of Computers, 2018, 41(1): 1-27.)
[7] Degris T, Pilarski P M, Sutton R S. Model-Free reinforcement learning with continuous action in practice [C]//Proc of American Control Conference. 2012: 2177-2182.
[8] Van Hasselt H, Mahmood A R, Sutton R S. Off-policy TD ( ) with a true online equivalence [C]//Proc of Conference on Uncertainty in Artificial Intelligence. 2014.
[2] Arulkumaran K, Deisenroth M P, Brundage M, et al. Deep reinforcement learning: a brief survey [J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.
[9] Van Seijen, Harm, Sutton R S. True online TD ( ) [C]//Proc of International Conference on International Conference on Machine Learning. 2014: I-692.
[10] Hou Yuenan, Liu Lifeng, Wei Qing, et al. A novel DDPG method with prioritized experience replay [C]//Proc of International Conference on Systems, Man, and Cybernetics. 2017: 316-321.
[11] Malla N, Ni Zhen. A new history experience replay design for model-free adaptive dynamic programming [J]. Neurocomputing, 2017, 266: 141-149.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.