INFORMATION
Volume 29, Number 1, pp.109-116
ISSN 1343-4500 eISSN 1344-8994

Enhancement of Exploration Efficiency in DQN-Based Reinforcement Learning

Shuai Yuan* and Lei Li

*Graduate School of Science and Engineering, Hosei University, Tokyo, Japan
E-mail: shuai.yuan.9m@stu.hosei.ac.jp

Abstract

Previous research in Deep Q-Network (DQN) mainly focused on improving the accuracy of Q-value estimation and enhancing the adaptability of agents to their environments. However, this study aims to improve the exploration efficiency of agents by refining the exploration algorithm used in DQN. The conventional ε-greedy method, while simple and effective in balancing exploration and exploitation, becomes inefficient in high-dimensional environments where random exploration can slow down learning. To address this problem, this paper proposes a dynamic ε-greedy approach that adjusts ε based on the Q-value to enhance exploration efficiency and accelerate learning.

Keywords: Deep Reinforcement Learning, DQN, Exploration Efficiency, ε-Greedy, Dynamic Exploration

1. Introduction

Reinforcement learning (RL) is a machine learning paradigm in which an agent learns to make sequential decisions through interaction with an environment, guided by the goal of maximizing cumulative rewards. In recent years, RL has been successfully applied in a wide range of domains such as industrial robotics, autonomous driving, finance, and strategic game playing. Among various RL algorithms, Q-Learning serves as a fundamental approach that learns an action-value function estimating the expected future return for each state–action pair. However, the scalability of Q-Learning is limited in complex and continuous environments due to its reliance on tabular representation.

To overcome this limitation, Deep Q-Network (DQN) was introduced by Mnih et al. [1], combining Q-Learning with deep neural networks to approximate the Q-function. This breakthrough enabled agents to learn directly from high-dimensional input spaces such as images, leading to significant performance improvements in tasks like Atari game playing. Since then, a series of extensions have been developed to further stabilize training and enhance learning performance, including Prioritized Experience Replay (PER) [2], Double DQN [3], and Dueling Network Architectures [4]. Each of these methods addresses specific weaknesses

of the original DQN, such as instability, over-estimation of Q-values, or inefficiency in value decomposition.

While these improvements have advanced the reliability and accuracy of DQN, exploration efficiency—the ability of an agent to effectively discover optimal actions—remains an unresolved challenge. Most DQN implementations rely on the simple ε-greedy exploration strategy, in which the agent selects a random action with probability ε and chooses the best-known action with probability (1−ε). Although ε-greedy effectively balances exploration and exploitation in simple tasks, it becomes inefficient in high-dimensional or continuous environments where random exploration is unlikely to yield meaningful experience. As a result, agents may spend excessive time exploring sub-optimal regions of the state space, slowing down convergence and overall learning efficiency.

Therefore, improving the exploration strategy within DQN is essential for advancing the effectiveness of deep reinforcement learning. This research focuses on enhancing exploration efficiency by introducing a dynamic ε-greedy method that adapts ε based on the agent's confidence, represented by its Q-values. By linking ε to the Q-value distribution, the proposed method aims to dynamically adjust the exploration intensity according to the learning progress. This approach is expected to accelerate convergence, reduce unnecessary random actions, and improve learning efficiency in both discrete and continuous environments.

2. Related Work

The Deep Q-Network (DQN) algorithm [1] revolutionized reinforcement learning by introducing deep neural networks to approximate the Q-function, enabling agents to learn effective policies directly from raw, high-dimensional sensory inputs. The standard DQN employs an experience replay buffer to store tuples of state, action, reward, and next state, allowing decorrelation of training samples and improved data efficiency. The Q-value is updated using the Bellman equation, as expressed in Equations (1) and (2):

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a') \tag{1}$$

$$a' = argmaxQ(s',a') \tag{2}$$

While this approach achieved great success, it also introduced several limitations, such as Q-value overestimation, instability during training, and inefficient exploration. Consequently, several extensions have been proposed to address these issues.

2.1 Prioritized Experience Replay (PER):

Schaul et al. [2] proposed PER to improve sample efficiency by assigning higher sampling priorities to transitions with larger temporal-difference (TD) errors. This allows the agent to focus more on learning from unexpected or informative experiences, accelerating convergence compared to uniform sampling. However, PER does not directly address exploration inefficiency, as it primarily improves the learning phase rather than the data collection strategy.

2.2 Double Deep Q-Network (DDQN):

Van Hasselt et al. [3] introduced Double DQN to mitigate the overestimation bias in Q-value updates. In the standard DQN, the same network is used for both action selection and value estimation, which tends to overestimate expected rewards. DDQN separates these two functions by using the online network for action selection and the target network for value evaluation. The update rule is modified as follows:

$$Q(s,a) = r + \gamma Q_{target}(s', arg\max_{a'} Q(s',a')) \quad (3)$$

This adjustment effectively stabilizes learning and provides more accurate value estimation.

2.3 Dueling Network Architecture:

Wang et al. [4] proposed the Dueling Network Architecture, which decomposes the Q-value into two components: the state-value function $V(s)$ and the advantage function $A(s,a)$. The final Q-value is computed as:

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + (A(s,a;\theta,\alpha) - \frac{1}{|A|}\sum_{a'} A(s,a';\theta,\alpha)) \quad (4)$$

This structure enables the network to better estimate the relative importance of actions in each state, improving stability and efficiency when learning in complex environments.

Despite these advancements, most of the improvements focus on value function approximation and sample efficiency, while exploration efficiency remains an under-explored topic. The commonly used ε-greedy strategy, which randomly selects actions with probability ε, is simple and effective for low-dimensional tasks but becomes inefficient in complex or

continuous environments. Excessive random exploration often leads to wasted computation and slower convergence.

Therefore, it is essential to develop an exploration mechanism that adapts dynamically to the agent's learning progress. In this study, a dynamic ε-greedy algorithm is proposed, where ε is adjusted based on the Q-value distribution. This approach aims to enhance exploration efficiency by linking the exploration rate with the agent's confidence in its current policy, allowing more intelligent and goal-oriented exploration compared to the traditional static ε-greedy approach.

3. Proposed Method

The traditional ε-greedy method controls exploration using a manually scheduled ε, which decays linearly or exponentially with training progress. However, such static scheduling does not adapt to the agent's actual learning state. To address this issue, this study proposes a Q-value–based dynamic ε-greedy strategy that incorporates a scaling coefficient α to flexibly control the exploration intensity.

3.1 Dynamic ε Formulation

In the proposed method, the exploration rate ε at time step t is defined as:

$$\varepsilon_t = \alpha\left(1 - \frac{|\,Q_t\,|}{|\,Q_{max}\,|}\right) \tag{5}$$

where:

$Q_t = Q(s_t, a_t)$ is the current estimated Q-value for the selected action at time step t, $Q_{\max}$ is the maximum Q-value observed so far during training, and α $(0 < \alpha \leq 1)$ is a tunable coefficient that controls the global magnitude of exploration.

This equation ensures that ε decreases dynamically as the agent becomes more confident in its learned policy (i.e., when $|\,Q_t\,|$ approaches $|\,Q_{\max}\,|$), while maintaining a higher exploration rate when uncertainty is high or Q-values are small.

In practice, α can be adjusted depending on the environment's complexity: a higher α encourages more exploration, whereas a smaller α leads to faster convergence.

3.2. Algorithm Overview

The integration of this adaptive ε into the DQN training loop requires minimal modification. The overall procedure can be summarized as follows:

1.At each time step t, compute $Q_t = Q(s_t, a_t)$using the online Q-network.

2.Calculate ε□ according to Equation (5).

3.With probability ε□, select a random action; otherwise, select the greedy action $a_t = \arg\max_a Q(s_t, a)$.

4.Store the transition (s_t, a_t, r_t, s_{t+1})in the replay buffer and update the network parameters via mini-batch learning.

5.Continuously update $Q_{\max}$ as the highest Q-value observed so far during training.

This dynamic ε-greedy algorithm preserves the simplicity of the standard DQN while enabling adaptive exploration driven by real-time Q-value feedback.

4. Experiment and Results

To evaluate the effectiveness of the proposed dynamic ε-greedy exploration strategy, experiments were conducted on two benchmark environments from OpenAI Gym: MountainCar-v0 and LunarLander-v2.

Both environments were tested under two exploration strategies:

1.a conventional fixed ε-greedy schedule, and

2.the proposed Q-value-based dynamic ε-greedy (Equation (5)).

All hyperparameters were kept identical across both settings to ensure fair comparison.

4.1. Experimental Setup

Each agent used a DQN architecture with two hidden layers (128 and 64 neurons), ReLU activation, Adam optimizer (learning rate = 0.001), and a replay buffer of 2000 samples. The discount factor γ was 0.99, batch size = 64, and target network updates occurred every 200 steps. For the baseline, ε was linearly decayed from α, while the proposed method computed ε dynamically as Equation 6.

4.2. Results on MountainCar-v0

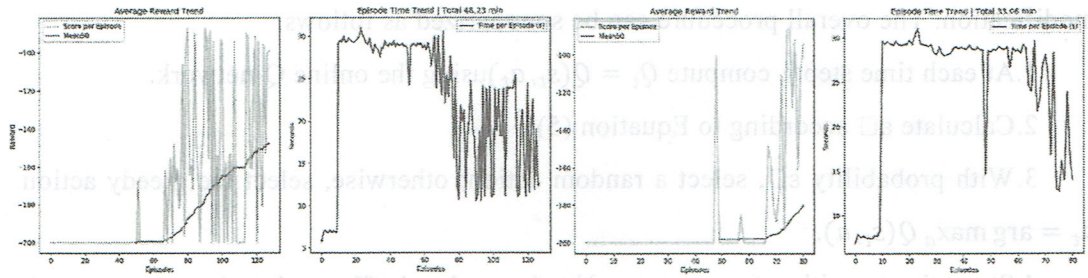

Figure 1(a) Figure 1(b)

Figure 1(a) and Figure 1(b) illustrate the reward progression and the time trend of each episode.

Under the fixed ε-greedy condition, the agent required approximately 127 episodes (48.23 min) to achieve stable convergence, with high reward variance.

In contrast, the dynamic ε-greedy agent achieved comparable or higher average reward in about 80 episodes (33.06 min)—a 45 % reduction in total training time.

Figure 1(a) and Figure 1(b) shows that the moving-average reward (red curve) under the proposed method rose more smoothly and reached higher asymptotic performance, while the fixed ε baseline exhibited oscillations and occasional regressions.

4.3. Results on LunarLander-v2

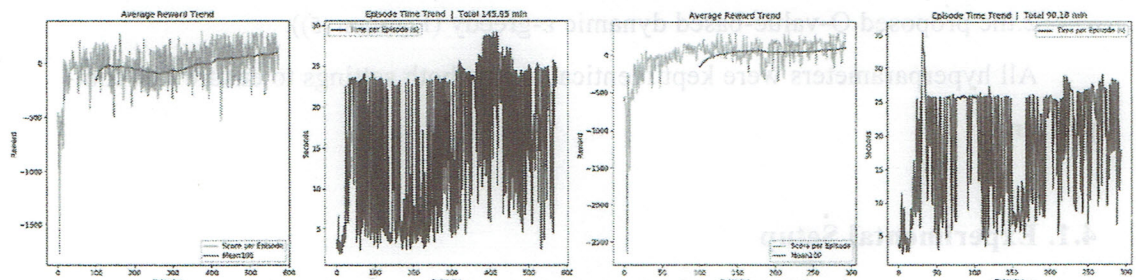

Figure 2(a) Figure 2(b)

For the LunarLander environment (Figure 2(a) and Figure 2(b)), a similar improvement trend was observed.

The baseline fixed ε agent required roughly 500–600 episodes to reach a mean score > 100, which represents a successful landing policy.

The proposed dynamic ε agent achieved this performance within 300 episodes, corresponding to a 38% reduction in convergence time.

The reward trajectory demonstrated faster growth and smaller fluctuations, indicating more directed exploration guided by Q-value confidence.

In addition, the dynamic ε schedule avoided excessive random actions in late training, leading to more stable descent and reduced variance in landing outcomes.

5. Conclusion

This study proposed a dynamic ε-greedy exploration strategy for Deep Q-Networks (DQN), in which the exploration rate ε is adaptively adjusted based on the magnitude of the Q-values.

By introducing a scaling coefficient α and defining ε as Equation (5) the agent dy-namically modifies its exploration behavior according to its learning confidence.

This formulation allows the agent to explore extensively in uncertain states and gradually shift toward exploitation as the Q-values stabilize, eliminating the need for manually tuned decay schedules.

Experimental results on MountainCar-v0 and LunarLander-v2 demonstrated that the proposed method improves learning efficiency and convergence speed compared to the conventional fixed ε-greedy approach. In MountainCar, training time was reduced by approximately 45%, while in LunarLander, a 38% reduction was achieved on average. Moreover, the reward trends exhibited smoother and more stable learning, confirming that dynamically linking ε to Q-values effectively balances exploration and exploitation throughout the training process.

In summary, the proposed method provides a simple yet effective enhancement to standard DQN, improving exploration efficiency without additional network complexity or parameter tuning.

Future work will extend this approach to continuous control tasks (e.g., Pendulum-v1, BipedalWalker-v3) and investigate its integration with advanced exploration techniques such as parameter noise, curiosity-driven exploration, and entropy-based regularization to further enhance sample efficiency and adaptability in complex reinforcement learning environments.

References

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[2] Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

[3] Van Hasselt, H., Guez, A., & Silver, D. (2016, March). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30, No. 1).

[4] Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016, June). Dueling network architectures for deep reinforcement learning. In *International conference on machine learning* (pp. 1995-2003). PMLR.

[5] Tokic, M. (2010, September). Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *Annual conference on artificial intelligence* (pp. 203-210). Berlin, Heidelberg: Springer Berlin Heidelberg.