Optimizing Robotic Arm Learning: Curiosity-Driven Deep Deterministic Policy Gradient

Jiarun Liu*

Applied Statistics, University of Toronto, 27 King's College Cir, Toronto, Canada

Abstract. This study explores the application of the Reinforcement Learning (RL) in training robotic arms, particularly using the Deep Deterministic Policy Gradient (DDPG) algorithm enhanced by a curiosity-driven mechanism. Robotic arms have various real-life applications, such as in the surgeries and assistive technologies. However, collecting the large-scale real-world data is costly and impractical, making simulation environments essential for optimization. The DDPG, well-suited for continuous action spaces, was employed to improve the robotic arm's precision and adaptability. Integrating a curiosity mechanism allowed the system to explore and learn more efficiently, significantly improving the training time and success rate. The results demonstrate a 12% reduction in training time and an 18% increase in the success rate when using curiosity-driven exploration. These findings suggest that the enhanced DDPG algorithm not only accelerates learning but also enables better task execution, offering a promising approach for the real-world robotic applications.

1 Introduction

Robotic arms have various real-life applications, such as in robot-assisted surgeries for precise operations and in assistive technologies for individuals with disabilities, like wheelchair-mounted arms that enable daily tasks. These arms benefit from neuromorphic adaptive control, which enhances their adaptability and efficiency in dynamic environments, offering potential for advanced Reinforcement Learning (RL) applications. Collecting large amounts of real-world data for robotic arms is often impractical and expensive. Therefore, it is essential to explore robotic arm mechanisms within simulated environments to facilitate their improvement and optimization in real-life applications [1].

Machine Learning (ML) and RL have seen rapid innovations, extending their application across industries [2-5]. ML has been instrumental in advancing healthcare, where RL is being used to optimize treatment schedules and personalize care plans for diseases like cancer. For instance, RL algorithms help in cell growth optimization for tissue engineering and regenerative medicine, offering adaptive solutions to complex biological problems [6].

In this study, it discusses the application of RL in training robotic arms. RL is a key Artificial Intelligence (AI) technology that allows machines to improve performance through continuous feedback and interaction with their environment. AI has a broad range of

* Corresponding author: ljr.liu@mail.utoronto.ca

applications, including ML and deep learning, which are increasingly used to optimize control systems, making robotic arm training more efficient and adaptive. In robotics, Deep Reinforcement Learning (DRL) has revolutionized control systems, particularly for robotic grasping and manipulation tasks. Recent innovations use DRL for automating object manipulation in complex environments, such as pick-and-place operations for varied shapes. These techniques are also improving the autonomous control of mobile robots in dynamic settings like disaster management and autonomous navigation [7].

In addition to medical and assistive applications, RL has proven valuable in various industrial and real-world scenarios. For example, RL is being used to enhance the performance of robotic arms in manufacturing environments. These robots are tasked with assembling intricate components on production lines, where precision and adaptability are crucial. RL enables the robots to continuously improve their performance by learning from past mistakes, resulting in more efficient production processes [8].

Moreover, RL is contributing to the development of autonomous drones for search-and-rescue missions. In these high-stakes environments, drones must navigate through complex terrains, often without pre-programmed instructions. RL allows them to adapt to new obstacles and optimize their paths in real-time, increasing the likelihood of success in time-sensitive operations [9].

The study explores the application of a curiosity-driven mechanism combined with the Deep Deterministic Policy Gradient (DDPG) algorithm to improve the learning efficiency of robotic arms. The DDPG algorithm, an actor-critic method well-suited for tasks in continuous action spaces, is employed to enhance the robotic arm's precision in task execution. The primary goal is to address the issues of slow convergence and adaptability often encountered in RL models used for robotic control.

2 Method

2.1 The definition of scenarios

The scenario designed for this study involves training a robotic arm to perform precise tasks in a simulated environment. The primary tool used is the DDPG algorithm, known for its efficiency in continuous action spaces. Tool related to the developing this environment. Key elements of the environment include the robotic arm's physical parameters, such as joint angles and positions, along with a goal area the arm needs to reach. The target is to maximize the arm's ability to navigate to this goal under varying conditions. Curiosity-driven exploration, intrinsic rewards, and RL mechanisms are incorporated to ensure the agent learns not just from reaching the goal but also by understanding the environment and reducing uncertainty. Algorithm The environment includes dynamic challenges that help assess the robot's adaptability and learning efficiency, simulating real-world applications like object manipulation or precision tasks in manufacturing or healthcare scenarios.

2.2 DDPG

The DDPG algorithm is a model-free, off-policy actor-critic method designed for environments with continuous action spaces. It stands out from other RL methods due to its ability to handle deterministic policies, making it highly effective for robotic control applications. Unlike traditional Q-learning approaches, which work well in discrete action spaces, DDPG uses a neural network to predict continuous actions, making it more suitable for tasks involving fine-grained control, such as robotic arm manipulation.

DDPG combines the strengths of two critical RL techniques: deterministic policy gradient and deep learning. The core architecture consists of two main networks: an actor network that decides the actions, and a critic network that evaluates them. Both networks are built using fully connected layers specific values with Rectified Linear Unit (ReLU) activation functions and optimized with Adam optimizers. The DDPG algorithm is designed with a sophisticated neural network architecture specifically tailored for environments requiring continuous action control, such as robotic arm manipulation. This architecture is split into two primary components: the actor network and the critic network, each configured to optimize learning and control efficiency in complex tasks.

Actor Network: The actor network functions to map the state space directly to actions, facilitating precise control. It features an input layer whose size is determined by the dimensions of the state space, ensuring that all relevant environmental data is captured. This is followed by two hidden layers with 400 and 300 neurons respectively, employing ReLU activation functions to introduce non-linearity essential for handling diverse and complex state representations. The output layer consists of as many neurons as there are action dimensions, using a tanh activation function to scale the outputs to the range [-1, 1]. This scaling matches the requirements of the control signals needed for precise robotic arm movements.

Critic Network: Complementary to the actor, the critic network estimates the Q-value function, evaluating the effectiveness of actions taken by the actor. It begins with a state input layer that mirrors the state dimensionality of the actor's input layer. An action input layer is integrated at the second layer to assess specific actions taken, which then feeds into two additional hidden layers configured similarly to those in the actor network, supporting the synthesis of state and action information into actionable insights. The output layer of the critic consists of a single neuron with linear activation, outputting a continuous range Q-value estimate for the assessed state-action pairs.

This dual-network setup allows the DDPG algorithm to not only generate but also evaluate actions in a continuous space, making it highly effective for tasks that require fine-grained control adjustments, like robotic arm manipulation. The detailed configuration of each layer is crucial for replication and understanding of the DDPG's application, providing clear insights into how both networks interact and are structured to achieve optimal control and learning efficiency. This structured approach ensures that the DDPG algorithm can effectively learn deterministic policies, handling the complexities and nuances required in advanced robotic tasks. A key feature of DDPG is its use of experience replay, where past experiences are stored in a memory buffer and reused to improve learning efficiency. Additionally, soft target updates stabilize the training by slowly updating the target networks, reducing the risk of divergence in training.

2.3 Implementation details

The implementation of the DDPG algorithm for this project involves several critical components. One of the most important aspects is setting the learning rate. In this study, the actor and critic networks are trained with different learning rates to optimize the performance. The actor network typically uses a smaller learning rate (e.g., 0.0001) to ensure smooth policy updates, while the critic network employs a slightly larger rate (e.g., 0.001) to quickly converge on the value function.

The optimizer used is the Adam optimizer, which is well-suited for deep learning tasks due to its adaptive learning rate capabilities. The loss function plays a crucial role in this system. For the critic network, the loss is calculated using the temporal difference (TD) error, which measures the difference between the predicted Q-values and the target Q-values. For

the actor network, the loss is indirectly minimized by maximizing the expected Q-value, guiding the actor towards better policy decisions.

Several evaluation metrics are used to assess the performance of the DDPG agent. The primary metric is the success rate, which measures how often the robotic arm successfully completes the task. Additionally, the average time to reach the goal and the overall reward collected during training are monitored to evaluate learning efficiency. These metrics provide insights into the agent's adaptability, accuracy, and speed of learning.

The training process is divided into epochs, with each epoch consisting of multiple episodes where the agent interacts with the environment. Over time, the agent learns from these interactions, using the feedback to improve its policy. The inclusion of a curiosity-driven exploration mechanism further enhances the learning process by encouraging the agent to explore unfamiliar states, thus improving the overall training efficiency.

In this experiment, the DDPG algorithm, enhanced by a curiosity-driven mechanism, significantly improved the learning efficiency of the robotic arm in the simulated environment. The structure of the neural networks used in the algorithm is a critical factor. Both the actor and critic networks were fully connected layers, where the actor network guides the action decisions, and the critic network evaluates the chosen actions. The specific architecture consists of two main layers for each network, with the actor network utilizing smaller learning rates (e.g., 0.0001) to ensure stable policy updates and the critic network using slightly larger learning rates (e.g., 0.001) to accelerate value function convergence.

The actor and critic networks each contain two fully connected layers. For the actor network, the first layer comprises 128 neurons with ReLU activations [10], followed by the output layer which directly maps to the action space. The critic network follows a similar structure, with the same number of neurons in the first layer. The architecture ensures that the networks can handle continuous action spaces effectively, which is critical for robotic control.

3 Results and discussion

3.1 Training time

A comparison of training time with and without the curiosity mechanism demonstrates a reduction from 224.31 seconds to 196.25 seconds shown in Fig. 1. The 12% decrease in training time is due to the curiosity-driven mechanism encouraging the agent to explore new state spaces more effectively. By focusing on parts of the environment where its understanding is weaker, the curiosity mechanism enables the agent to gain valuable insights faster, reducing overall training duration.

3.2 Success rate

The success rate saw a marked improvement from 47% to 65% after introducing the curiosity mechanism (shown in Fig. 1). This highlights how intrinsic rewards drive the agent to explore unfamiliar state spaces and develop more robust strategies for completing tasks. The ability to generalize to new situations and achieve goals efficiently is a key benefit of this approach.

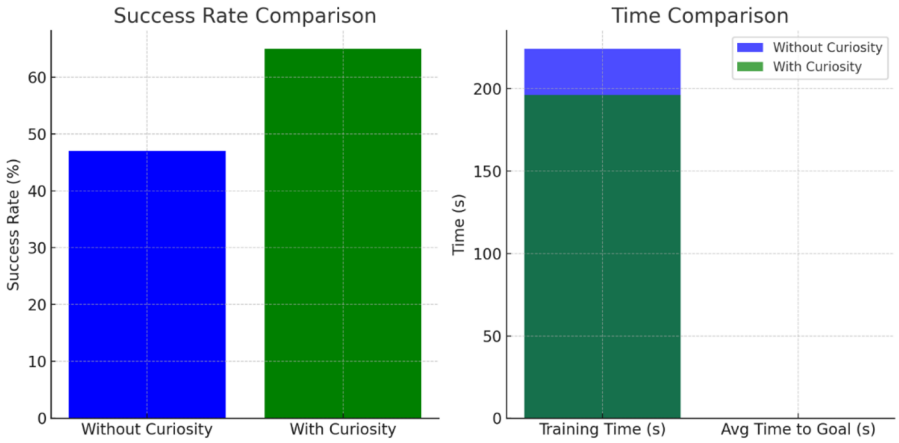

Fig. 1. Impact of Curiosity Mechanism on Robotic Arm Success Rate and Training Efficiency (Photo/Picture credit: Original).

3.3 Average time to goal

The average time to reach the goal decreased slightly from 0.04 seconds to 0.03 seconds. Though the difference seems minor, it reflects a more efficient strategy learned by the agent, leading to quicker task completion. This improvement further validates the hypothesis that the curiosity mechanism enhances the model's decision-making capabilities by reducing unnecessary exploration and focusing on more productive actions.

3.4 Performance analysis

The primary metrics—success rate, training time, and task completion time—show a clear benefit from incorporating the curiosity-driven mechanism. The intrinsic rewards generated by the curiosity mechanism led the agent to explore and learn from challenging or less understood areas of the environment, accelerating the overall learning process. This was particularly evident in the improvement in success rates and faster convergence times, indicating that the algorithm successfully balances exploration and exploitation in continuous control tasks.

3.5 Limitations and future directions

While the results are promising, several limitations remain. For instance, the algorithm's performance may vary in more complex environments with higher-dimensional state spaces. Additionally, while the curiosity mechanism improves exploration, there remains the challenge of balancing intrinsic and extrinsic rewards optimally. Future work could explore hybrid algorithms or modifications to the curiosity mechanism to further enhance performance, especially in real-world applications with more complex and noisy environments.

4 Conclusion

This study successfully applied the DDPG algorithm enhanced with a curiosity-driven mechanism to improve robotic arm control in a simulated environment. The results

demonstrated a notable increase in success rates, reduced training times, and faster task completion, showcasing the efficiency of the curiosity-driven learning. This approach highlights the importance of structured exploration in RL, particularly in tasks where external rewards are sparse or delayed. Future work should focus on optimizing the curiosity mechanism, exploring alternative algorithms like PPO or SAC, and applying this model to real-world robotic control tasks.

References

1. M. Ehrlich, et al., Adaptive control of a wheelchair mounted robotic arm with neuromorphically integrated velocity readings and online-learning. Frontiers in Neuroscience, 16, 1007736 (2022).
2. A. Charpentier, R. Elie, C. Remlinger, Reinforcement learning in economics and finance. Computational Economics, 1-38 (2021).
3. X. Y. Liu, H. Yang, Q. Chen, R. Zhang, L. Yang, B. Xiao, C. D. Wang, FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. arXiv preprint arXiv:2011.09607 (2020).
4. Y. J. Hu, S. J. Lin, Deep reinforcement learning for optimizing finance portfolio management. In 2019 Amity International Conference on Artificial Intelligence (AICAI), pp. 14-20 (2019).
5. S. K. Zhou, H. N. Le, K. Luu, H. V. Nguyen, N. Ayache, Deep reinforcement learning in medical imaging: A literature review. Medical Image Analysis, 73, 102193 (2021).
6. M. N. Al-Hamadani, M. A. Fadhel, L. Alzubaidi, B. Harangi, Reinforcement Learning Algorithms and Applications in Healthcare and Robotics: A Comprehensive and Systematic Review. Sensors, 24(8), 2461 (2024).
7. B. Singh, R. Kumar, V. P. Singh, Reinforcement learning in robotic applications: a comprehensive survey. Artificial Intelligence Review, 55(2), 945-990 (2022).
8. M. Panzer, B. Bender, Deep reinforcement learning in production systems: a systematic literature review. International Journal of Production Research, 60(13), 4316-4341 (2022).
9. A. T. Azar, A. Koubaa, N. Ali Mohamed, H. A. Ibrahim, Z. F. Ibrahim, M. Kazim... G. Casalino, Drone deep reinforcement learning: A review. Electronics, 10(9), 999 (2021).
10. J. He, L. Li, J. Xu, C. Zheng, ReLU deep neural networks and linear finite elements. arXiv preprint arXiv:1807.03973 (2018).