

Research on Deep Learning Algorithms and Applications in Robotic Arm Control

Wenyu Zhou

*Zhejiang University - University of Illinois Urbana-Champaign Institute, Zhejiang University,
Hangzhou, China
wenyu.22@intl.zju.edu.cn*

Abstract. Machine learning is one of the significant computer science fields in the modern world. As a subset of machine learning, deep learning is also becoming increasingly important. This paper mainly focuses on the application of deep learning in robotic arm control through analyzing the performance of various algorithms in different tasks in the real world. First, the paper introduces characteristics of multiple neural network architectures, including convolutional neural networks (CNNs), feedforward neural nets (FNNs), and recurrent neural networks (RNNs). Then the paper introduces multiple algorithms related to reinforcement learning, such as proximal policy optimization (PPO), deep deterministic policy gradient (DDPG), and deep Q-networks (DQN), providing their basic information and methods. The paper also describes the advantages of imitation learning. After a brief account of the theoretical part of the knowledge, the paper links theory to practice. The paper evaluated the performance of these deep learning methods in robotic arm tasks across three distinct application scenarios, providing important references for algorithm selection and performance optimization. In the end, the paper discusses the current shortcomings and disadvantages of deep learning algorithms and offers some suggestions for its future development.

Keywords: artificial intelligence, deep learning, reinforcement learning, robot control

1. Introduction

In recent years, with the rapid development of robotics technology, robotic arms have been used in a wide range of applications such as industrial manufacturing, warehousing and logistics, agricultural harvesting, and even medical assistance. Conventional robotic arms can only perform fixed actions, such as moving items on an assembly line in a factory. Traditional robotic arms can only perform fixed motions, such as moving items on an assembly line in a factory. The operation of traditional robotic arms relies on pre-written basic control codes. Compared to pre-written basic control codes, deep learning is rooted in the biology of the brain, whose purpose is to allow machines to recognize images and hear sounds like humans. After several years of rapid development, deep learning is widely used in areas such as computer vision and natural language processing. Deep learning allows computational models consisting of multiple processing layers to learn data representations with multiple levels of abstraction [1]. From the perspective of bionics, deep learning is like a neural

network with many layers, and adjusting the sensitivity of each neuron in the network enables the network to activate different neurons and produce different responses when receiving different stimulus signals. If deep learning is combined with a robotic arm, the robotic arm can realize more flexible operational control in dynamic environments while significantly enhancing the robustness of the system [2]. For example, at the beginning, the traditional robotic arm could only grasp fixed-shaped items, while the improved robotic arm was able to adjust its grasping strategy according to the shape of the item.

Despite progress in related research, the application of deep learning in robotic arm control still faces several challenges, including high costs in obtaining training samples, a lack of precision in robotic movements and unpredictability of movements, and high demands on computational resources. So, this paper conducts a systematic analysis and comparison of various deep learning algorithms in robotic arm control, evaluating their performance and applicability in tasks such as grasping, trajectory planning, and human-robot collaboration. To provide more accurate conclusions, this paper presents a side-by-side comparison of multiple real-world examples of manipulators combined with deep learning. This paper aims to provide references for algorithm selection and to offer theoretical and methodological support for the practical deployment of intelligent robot systems.

2. Deep learning techniques in robotic arm control

2.1. Neural Network architectures

Neural networks are not only an important machine learning technology, but also the foundation of deep learning. Various neural network models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Feedforward Neural Networks (FNNs), are the foundational building blocks of modern deep learning systems. These artificial neural networks attempt to process information by simulating the way the brain's neural networks process and store information.

FNNs are simple in structure, easy to understand and implement, and suitable for tasks such as classification, regression, and scoring of structured data. They serve as the basic building blocks for more complex neural networks (such as CNNs and RNNs) [3]. But FNNs are less suitable for tasks that require sequential modeling or spatial relation analysis, such as image processing. CNNs are well-suited for processing spatially structured data, such as images, by applying convolutional filters to capture local features. They are widely used in robotic vision tasks, including object recognition and scene understanding. RNNs have recurrent connections, and the network's output depends not only on the current input but also on all previous inputs. This structure enables RNNs to process sequence data of arbitrary length [4].

2.2. Reinforcement Learning

Reinforcement Learning (RL) is a subfield of machine learning and mainly focuses on training agents to learn to make optimal decisions through interactions within a given environment, which can give feedback. The learning process is driven by evaluative feedback, which is expressed in the form of rewards or penalties. Rewards reflect the effectiveness of the actions taken by the agent. Through iterative refinement of its policy based on such feedback, the agent seeks to maximize the expected cumulative reward over time. A schematic representation of the RL decision-making framework is presented in Figure 1.

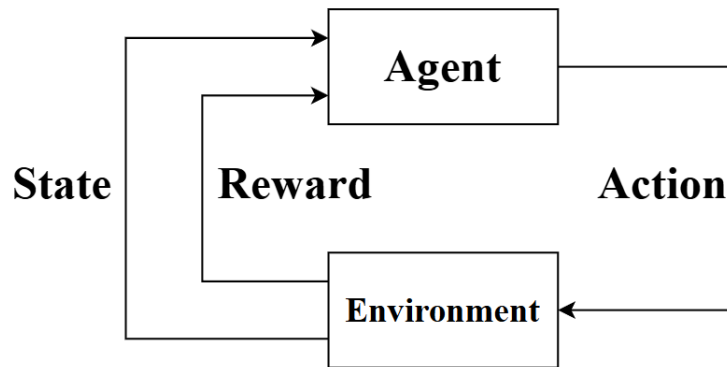


Figure 1. Sequential decision-making framework

The environment constitutes an external dynamical system that encapsulates the agent. Through perceptual interfaces, the agent acquires state observations and executes policy-derived actions that induce state transitions within the environment's operational framework. The state provides a complete description of the environment at a given time. The action refers to the decisions made by the agent to interact with the environment. Different environments permit different sets of actions, collectively referred to as the action space, which may be either discrete or continuous. The reward is a scalar feedback signal provided by the environment, indicating the quality of the agent's action at a specific step. A higher reward means better performance of the agent under the given policy. The program uses different reward functions for different tasks. For example, in the grasping task of an intelligent robotic arm, the basic reward function is related to height, and the reward increases as the height of the target object increases. In the gripping task of an intelligent robotic arm, the basic reward function is related to the number of activated sensors. The more sensors are activated, the higher the gripping stability, and the reward increases.

The DQN algorithm combines a deep neural network with the classical reinforcement learning algorithm Q-learning to achieve the processing of a high-dimensional, continuous state space. The DQN algorithm stores the experience of the agent in a replay memory bank, and then randomly draws samples from it and uses these experiences for model updating. This has the advantage of avoiding correlation between samples and improving the stability and convergence speed of the model. The DQN algorithm uses two neural networks: an online network for selecting actions and updating the model, and a target network for computing the target Q-value. This reduces the fluctuation of the target and improves the stability of the model. However, DQN is limited when dealing with continuous action spaces [5].

To address this challenge, the Deep Deterministic Policy Gradient (DDPG) algorithm was created for the execution of continuous control operations. DDPG is similar to DQN because this algorithm uses two distinct networks: an actor network, which determines the actions, and a critic network, which evaluates them. DDPG is suitable for continuous motion spaces, such as the joint angle of a robotic arm, because it directly outputs a continuous motion instead of selecting from a limited set of motions.

Another popular algorithm is Proximal Policy Optimization (PPO). PPO is a reinforcement learning algorithm that can both ensure performance optimization and make the algorithm more stable and efficient. In reinforcement learning, directly optimizing the policy can lead to unstable training, and the model may crash because of excessive parameter changes. PPO can limit the magnitude of the policy changes so that each step of the training does not deviate too much from the current policy while making full use of the sampled data. PPO has been successfully applied to both

discrete and continuous control problems and is known for its robustness and ease of implementation in high-dimensional tasks [6].

2.3. Imitation learning

Imitation learning is a special kind of machine learning. Agents learn to perform a task by observing an expert's demonstration, rather than through direct programming like reward function design. Imitation learning reduces the need for task-specific knowledge and is particularly suitable for complex tasks where it is difficult to design reward functions. Imitation learning has a wide range of applications, including robotics, autonomous driving, and human-computer interaction [7].

In many cases, designing an appropriate reward function for reinforcement learning is difficult, and manually defined rewards often lead to uncontrollable agent behaviors and unintended performance results. Using robotic grasping tasks as an example, where the reward function is a linear function of the object's height, and imitation learning is not used. After training, the agent is motivated to maximize vertical displacement as soon as possible, which may result in the robot arm throwing the object upward at high speed rather than performing a stable and controlled lift after grasping. This outcome reflects a difference and misunderstanding between the reward function and the actual task objective. As a result of this, learning directly from human demonstrations enables agents to quickly and safely acquire complex skills that are typically attainable only through human experience, rather than having to learn from scratch and explore how to do things correctly and efficiently.

3. Algorithm evaluation and performance

3.1. Performance comparison of algorithms

To evaluate the performance of different deep learning algorithms in robot control tasks, this section compares three representative algorithms: Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), Artificial Neural Network (ANN), and Artificial Deep Q Network (ADQN).

Sekkat's study shows that DDPG is applied to a five-degree-of-freedom robotic arm grasping task [8]. You Only Look Once version 5 (YOLOv5) is used for object detection, and the DDPG algorithm is used to construct a neural inverse kinematics solver that significantly improves performance compared to traditional methods. Experiments show that 95.5% accuracy is achieved with only 400 training cycles, significantly higher than traditional analytical solutions. Additionally, it exhibits good training stability, fast convergence, and enough robustness. In the future, the same technique can be even applied to a seven-degree-of-freedom robotic arm grasping task.

Gao's research compared the grasping performance of DQN, ANN, and the ADQN algorithm under dynamic and static conditions [9]. The study applies the three algorithms in different directions: DQN for a base reinforcement learning model, ANN for object recognition and localization, and ADQN combining the first two. The experiment consists of two environments: a static environment with a fixed pedestal and a fixed target object, and a dynamic environment with a moving pedestal and a randomly moving object. During the initial training phase, ADQN reaches a success rate of 33.67% in the initial training, which is significantly higher than DQN's 28.39% and ANN's 20.13%. After training for 1 million rounds, the ADQN even reaches a success rate of 97.59%, which is still better than the DQN's 82.16% and the ANN's 88.66% [9]. After the introduction of demonstration learning, there is a 7-13% increase in task success rate and a significant increase in training stability.

Liu's review notes that DQN is suitable for discrete action space tasks, while DDPG is suitable for continuous action space tasks. This study systematically analyzes various deep reinforcement learning algorithms and specifically recommends that the combined approach of DDPG with Hindsight Experience Replay (HER) is the most outstanding performer for robotic manipulation tasks [10]. In terms of grasping task success rate, DDPG+HER's 92% success rate is not only better than ANN's 59%, but also better than DQN's 64%. Finally, the article shows that through interdisciplinary collaboration and continuous algorithmic optimization, industrial-grade, reliable applications are expected to be achieved within 5-8 years. Future research should focus on key issues such as proof of safety, optimization of energy consumption, and establishment of ethical frameworks.

3.2. Application comparison in tasks

Deep learning algorithms exhibit varying levels of performance when applied to specific sub-tasks in robotic grasping. This section focuses on three representative application scenarios: object recognition and localization, trajectory planning and control, and grasp execution under dynamic environmental conditions.

In object recognition and localization, the accuracy of visual perception directly affects the reliability of downstream control. CNNs are commonly integrated into grasping pipelines to extract high-level spatial features, providing stable visual input for control strategies such as DDPG. Enhanced architectures like ADQN further improve localization accuracy by combining classification and decision-making modules. In comparative evaluations, ADQN-based perception achieved an average localization error of only 2.8 mm, compared to 5.6 mm for standalone CNN models under partial occlusion [10].

Trajectory generation and control typically require solutions to high-dimensional, nonlinear problems. Compared with traditional analytical methods that often struggle in redundant or unstructured kinematic chains, learning-based approaches such as DDPG and ADQN exhibit greater flexibility. These algorithms are capable of learning effective grasp trajectories across different poses and environmental setups. For instance, DDPG-trained controllers reduced end-effector path deviation by 34% compared to ANN baselines in a cluttered environment [9].

In dynamic and unstructured environments, factors such as system responsiveness and adaptability to disturbance become critical. Algorithms that integrate both perception and control, such as ADQN, are better suited for tasks involving moving targets or variable interaction forces. These models demonstrate faster response times and higher action stability under perturbation. In contrast, models like DQN and ANN are more susceptible to performance degradation due to perception noise or unpredictable system dynamics.

3.3. Benefits and limitations

Multiple factors are influencing the effectiveness of deep learning algorithms in robotic grasping. Current deep reinforcement learning has significant performance limitations in robot control. First of all, it shows a serious bottleneck in sample efficiency, with an average of more than 800,000 attempts of training data requirements for a single task, and complex tasks often require 2-3 months of continuous GPU computing to converge, and long training times can also lead to energy consumption problems. And there is an online update delay of 200-500 milliseconds, which is difficult to meet the demand for real-time control; secondly, there are obvious defects in the generalization ability, which will lead to performance degradation when the work scene is changed,

and it can only deal with object morphology that is not very similar to the training set. For example, adding a square object to multiple columnar objects will interfere with the learning efficiency of the agent. The success rate of policy porting between different robotic arm platforms is very low, not to mention the collaboration of multiple robotic arms.

In addition, transparency and accuracy in decision-making are common issues that all deep learning algorithms need to face. Although deep learning significantly improves the environmental adaptability of the robotic arm, the process of deep learning is an invisible black box, where humans can only observe the results and do not have access to the process of machine learning. Experts are unable to trace the basis for the generation of critical actions, which also makes troubleshooting difficult, with 90% of anomalies impossible to locate through log analysis. Moreover, the optimal model in the experimental tests achieves a 92% object grasping success rate in the simulated environment. But in industrial production, especially those involving hazardous chemicals, the model needs to achieve a near 100% success rate to minimize the safety risk.

4. Conclusion

Comparative analysis shows that advanced ADQN and DDPG perform more balanced and better in generalization ability and robustness. These algorithms enable the robotic arm to work stably in various environments after training. In contrast, other algorithms have some limitations. But they can make full use of their advantages when putting simplicity first.

However, deep learning algorithms also have shortcomings. Now, many deep learning-based algorithms rely on large-scale and high-quality training data, and their performance can degrade significantly in an unfamiliar environment. In addition, the high computational overhead and real-time requirements of advanced algorithms may limit their deployment on resource-constrained platforms, while the lack of interpretability of their decision-making process makes it difficult to apply them to domains that prioritize security.

Looking ahead, research on deep learning control frameworks should pay more attention to improving data efficiency and real-time adaptive capability. Potential directions include introducing multimodal perceptual fusion, developing lightweight but expressive network structures, and exploring residual or hybrid control strategies that combine model-based stability with policy learning flexibility. In addition, improving the migration technique from Sim-to-Real is crucial for bridging the gap between training environments and real-world applications. This will ultimately enable robotic arms to accomplish complex tasks with higher safety and reliability in real-world environments.

References

- [1] Lecun, Y., Bengio, Y. and Hinton, G. (2015) Deep learning. *Nature*, 521(7553), 436.
- [2] Shinde, P.P. and Shah, S. (2018) A review of machine learning and deep learning applications. *Fourth International Conference on Computing Communication Control and Automation*, 000(8681925), 1-6.
- [3] Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J. (2021) A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 6999-7019.
- [4] Li, S., Li, W., Cook, C., Zhu, C. and Gao, Y. (2018). Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5457-5466.
- [5] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y. and Wierstra, D. (2015) Continuous control with deep reinforcement learning. *arXiv preprint arXiv: 1509.02971*.
- [6] Guo, Z., Fu, H., Wu, J., Han, W., Huang, W., Zheng, W. and Li, T. (2025) Dynamic Task Planning for Multi-Arm Apple-Harvesting Robots Using LSTM-PPO Reinforcement Learning Algorithm. *Agriculture*, 15(6), 588.

- [7] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. (2017) Imitation Learning: A Survey of Learning Methods. *ACM Comput. Surv.* 50, 2, Article 21 (March 2018), 35 pages.
- [8] Sekkat, H., Tigani, S., Saadane, R. and Chehri, A. (2021). Vision-Based Robotic Arm Control Algorithm Using Deep Reinforcement Learning for Autonomous Objects Grasping. *Applied Sciences*, 11(17), 7917.
- [9] Tianci Gao. (2024) Optimizing robotic arm control using deep Q-learning and artificial neural networks through demonstration-based methodologies: A case study of dynamic and static conditions, *Robotics and Autonomous Systems*, Volume 181, 104771.
- [10] Liu, R., Nageotte, F., Zanne, P., de Mathelin, M. and Drespe-Langley, B. (2021). Deep Reinforcement Learning for the Control of Robotic Manipulation: A Focussed Mini-Review. *Robotics*, 10(1), 22.