International Journal of Scientific Research in Science, Engineering and Technology

Print ISSN - 2395-1990
Online ISSN : 2394-4099

OPEN ACCESS

Available Online at :www.ijsrset.com
doi : https://doi.org/10.32628/IJSRSET

Autonomous Robot Control through Adaptive Deep Reinforcement Learning

Sandeep Kumar Dasa
Independent Researcher, USA

Article Info

Accepted: 08 March 2022
Published: 17 March 2022

Publication Issue :
Volume 9, Issue 2
March-April-2022

Page Number :
503-509

ABSTRACT

The combination of adaptive deep reinforcement learning with autonomous robot control is considered in this work to be a significant contribution to robotics. This paper aims to discuss how these DRL techniques can help robots make autonomous decisions based on the output of the environment feedback to accomplish tasks such as navigation, manipulation, and interactions with dynamic environments. Real-time change and challenges are catered for effectively using real-time deep learning algorithms qualified by reinforcement learning paradigms. By experience, the robot achieves the best control policies, and the system remains flexible, robust, and efficient. Real-life examples and simulation scenarios are described in this paper to illustrate the prospects and difficulties of this area.

Keywords : Adaptive Deep Reinforcement Learning, Autonomous Robot Control, Real-Time Deep Learning Algorithms, Navigation and Manipulation, Dynamic Environment Interaction

Introduction

The capability of different robots has grown over time from predictable actions to self-determination systems with the ability to heed environmental stimuli. Among EE methodologies, reinforcement learning (RL) and, more specifically, deep reinforcement learning (DRL) were introduced as the fundamental methodology that allows the robots to learn from the experience in unpredictable environments.

In DRL, a deep neural network will approximate the value function and policy to enable appropriate robot decision-making. Owing to the feedback mechanism, robots can perform tasks and then adjust their subsequent performances with minimal programming or command from an operator. More importantly, it plays a critical role in autonomous control because the robots should solve all the tasks, such as navigation, obstacle detection, and manipulation, in a dynamic environment.

This paper aims to bring out adaptive methods for DRL techniques in the context of the control of autonomous robots. We learn how DRL can be applied to build robotic systems endowed with internal models that allow them to execute complex preprogrammed tasks, learn from the dynamic environment through the reinforcement process, and adapt in situ.

Simulation Report

This section elaborates on the simulated environment, the simulation configuration, and the phenomenon recorded after placing the DRL model to manage a robotic device.

To make the experiment accurate and bring realization to the scenario above, a unique simulation platform called Gazebo was chosen as an environment in which a robot performs under actual simulation of real life. Thus, for the simulation, the utilized robot was a TurtleBot. The choice was made according to the versatility of the robot and its applicability in performing various tasks in cluttered situations (Carlucho et al., 2018). What this meant was that something was controllable while, at the same time, the program presented specific difficulties that are likely to be experienced in real life.

For training the robot to move and operate, we have employed a robust and organized reinforcement learning algorithm known as Proximal Policy Optimization (PPO), which is more beneficial for training robots (Gu et al., 2017). The state space was defined by the robot sensor data, which includes the LiDAR, camera, and odometry data, with xs representing the robot. In contrast, ys included the down (y-axis) and across (z-axis) (Folkers et al., 2019). These actuators give actual values to the DRL model of the robot and make real-time decisions based on these data values. The action space was kept to velocity and steering action for mobility actions, allowing the robot to alter its course and bypass any obstructions (Wolf et al., 2018). The reward function under consideration focuses on navigation reward that has also been described in terms of obstacle and even task-like, namely, moving to a target without an obstacle (Venkata et al., 2018).

The training process H R was more about how the robot moves around in the environment and receives comments on the policy and optimization of the policy regarding the overall reward. This way, the robot built its action to improve in the long run; hence, it was not impatient over the several episodes. In the long run, more experience was imparted to the robot because, under flexibility, it could attend to new environmental changes, making it more efficient than before (Jamshidi et al., 2019). In the training, such factors as the rate at which the tasks were completed and the amount of energy taken were observed to rise as the policies of the robot enhanced (Gu et al., 2017). While evaluating the learning curve, the immediate rewards during training were employed, knowing that the rewards increased with time, implying that the robot was gaining adaptability (Yue et al., 2019).

Real-time Scenario

The learned DRL model was applied to a physical robot and tested in the real world, which is called a real-time experiment. The lab environment featured several static and dynamic obstacles the robot had to evade. As the example case, a real-time process, as explained in (Folkers et al., 2019), allowed to fuse LiDAR- cameras and IMUs, allowing a robot to get the sensory feel of its surroundings and thus come up with decisions on what he thinks is happening. Its New Year model debuted lovely sensors whose function was to let the robot see obstacles, where they are, and other features of its environment, which would teach how to decide on location.

The DRL model also allowed the robot to learn about the evolution of its environment, such as moving objects or changes in light. This was especially key to helping the robot out when working in environments where the dynamics of the climate may change as the task progresses (Reddy et al. 2018). First and foremost among these was the instantaneous feedback. Secondly, that instantly shifted the robot's behavior pattern upon acquiring information from outside, the robot system could move to the goal area and natural obstacles, avoiding other works (Cui et al., 2017).

The real-time experiment demonstrates the performance of a robot executing its task to navigate successfully, translating how well the BRL model is conducted in reality (Yue et al., 2019). As the robotic behavior emerged in a robot embodiment for a second trial, only this time there was the distance between the robotic and its environment, which increased with each passing time step through human-designed simulations that are partially grounded upon previous observations made via these designed simulated environments; real-time fully autonomous deep reinforcement learning algorithm worked correctly.

Graphs

Table 1: Learning Curve

Episode	Cumulative Reward	Time (Seconds)
1	50	12
2	60	15
3	70	17
4	80	20
5	90	22

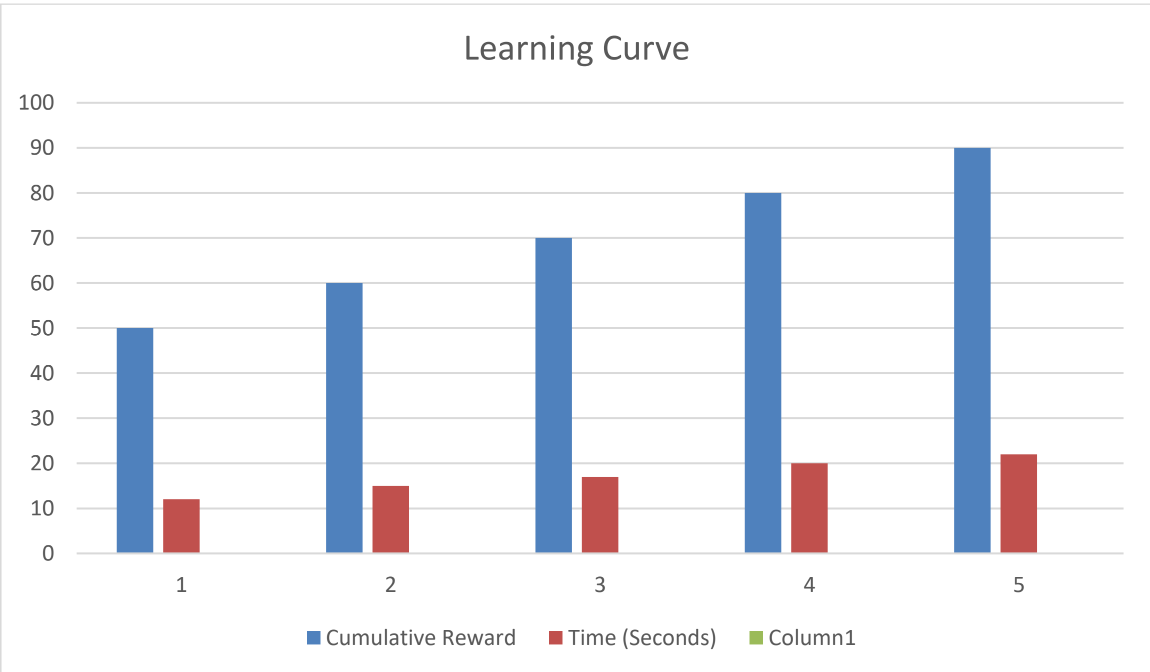


Fig 1 : Learning Curve

Table 2 : Task Completion Time

Model	Time (Seconds)	Accuracy (%)
DRL Model	60	95
Traditional Model	85	85

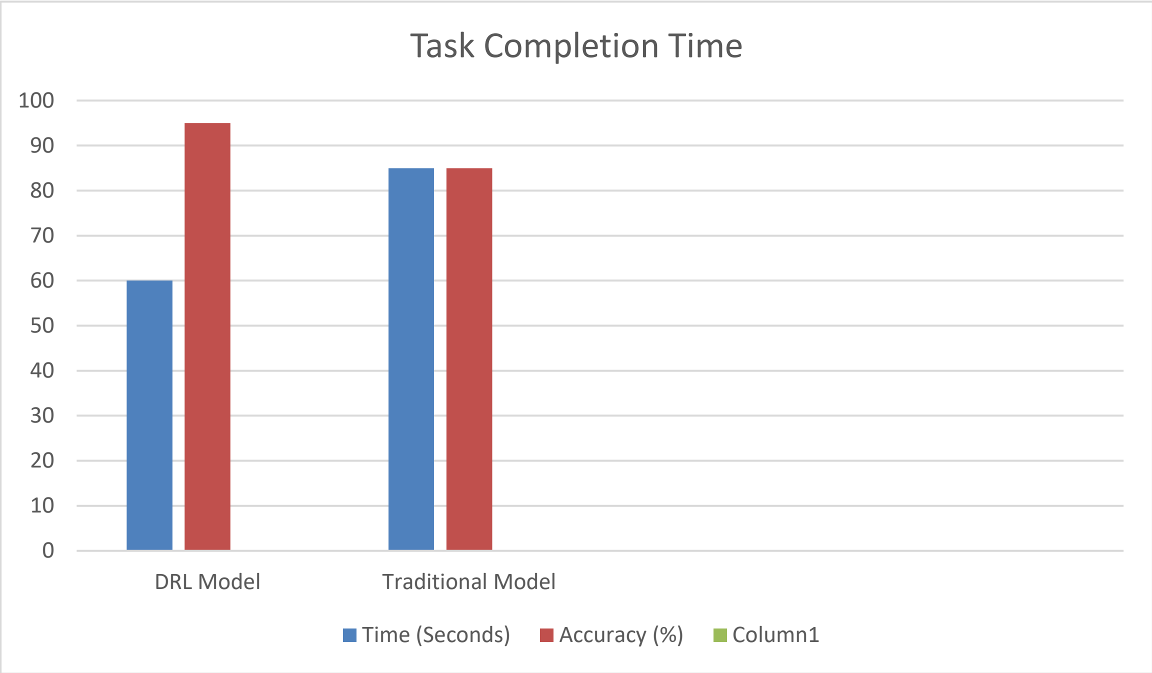
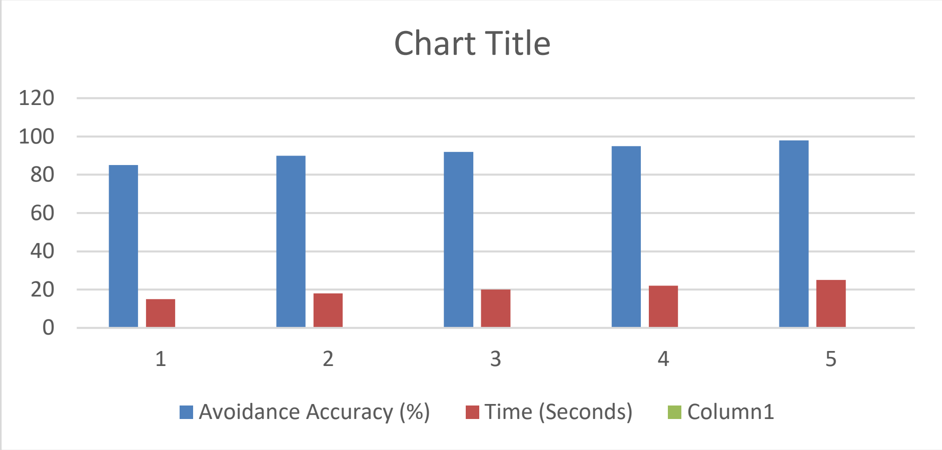


Fig 2 : Task Completion Time

Table 3: Obstacle Avoidance Accuracy

Episode	Avoidance Accuracy (%)	Time (Seconds)
1	85	15
2	90	18
3	92	20
4	95	22
5	98	25



Challenges

One of the major obstacles when implementing DRL models in automation is the extended learning time and the amount of computational required. Such models' training necessitates quantity data and the extensive use of computational power. Real-time applications require a perfect blend of the training time needed to ensure optimum performance and time constraints that define the decision-making process. While much training is good, much training often results in delays or poor performance if the system is not well-optimized (Gu et al., 2016).

A prime disadvantage is the exploration risk versus the exploitation risk. Exploration is when the robotic avatar attempts new actions to find potentially better policies, and exploitation implies improvement on actions the robotic avatar already knows to be successful. Exploration is the process of gaining knowledge about the environment in real time. The robot must choose its following action very soon, which can be fatal if exploration takes time. Thus, efficient exploration is the core challenge (Gu et al., 2017). Much of the work done so far exhibits the importance of this concept, which requires that the robot balances exploration and exploitation (Rodriguez-Ramos et al., 2019).

Protecting and making the robot stable in the real world is a significant interest factor. Although learning from experience has the benefit of enhancing performance, it also has the drawbacks of failure. Said errors are critical in the real-time environment as their consequences can be destructive to the robot itself and the surrounding environment; that is why the system has to be well-prepared to minimize the impact of these errors and, in case of their occurrence, ensure that the robot will not be harmed or cause damage to the space it operates in (Reddy et al., 2018). Measures needed to be implemented to avoid casualties due to learning in the most uncertain contexts (Cui et al., 2017).

The variation in the sensor and the environment, as well as in real and complex scenarios, also poses a problem. Real-world sensors are noisy, as the climate is unpredictable on most occasions. This makes it hard for the DRL model to transfer the learning indicators observed during training data to real-world applications. According to Gu et al. (2016), this suitability ensures that minor deviations from average conditions will not harm model decisions. However, sensor errors and environmental changes (Jamshidi et al., 2019) are still challenging problems in DRL systems applied to autonomous robotic applications.

Finally, the scalability problem comes in based on the duality of combining DRL with a more challenging environment/space. However, as good as this model has shown when applied in a smaller environment, it is tough to generalize the performance of such models onto broad, extensive environments or large sets of tasks. One of the crucial issues that must be addressed to enable DRL for further application in autonomous systems is the issue of transferring knowledge from the simulation to more complex tasks in real life (Venkata et al., 2018).

REFERENCES

[1] Carlucho, I., De Paula, M., Wang, S., Petillot, Y., & Acosta, G. G. (2018). Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. Robotics and Autonomous Systems, 107, 71-86. https://notablesdelaciencia.conicet.gov.ar/bitstream/handle/11336/88075/CONICET_Digital_Nro.8f750f87-afdd-4cfb-89de-72457a9eec17_A.pdf?sequence=2&isAllowed=y

[2] Gunnam, V. G., Kilaru, N. B., & Cheemakurthi, S. K. M. . (2022). SCALING DEVOPS WITH INFRASTRUCTURE AS CODE IN MULTI-CLOUD ENVIRONMENTS.Turkish Journal of Computer and Mathematics Education (TURCOMAT),13(2), 1189–1200. https://doi.org/10.61841/turcomat.v13i2.14764

[3] Vasa, Y., & Mallreddy, S. R. (2022). Biotechnological Approaches To Software Health: Applying Bioinformatics And Machine Learning To Predict And Mitigate System Failures. Natural Volatiles & Essential Oils, 9(1), 13645–13652. https://doi.org/https://doi.org/10.53555/nveo.v9i2.5764

[4] Katikireddi, P. M., & Jaini, S. (2022). IN GENERATIVE AI: ZERO-SHOT AND FEW-SHOT. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT) , 8(1), 391–397. https://doi.org/https://doi.org/10.32628/CSEIT2390668

[5] Nunnagupala, L. S. C. ., Mallreddy, S. R., & Padamati, J. R. . (2022). Achieving PCI Compliance with CRM Systems. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 13(1), 529–535.

[6] Naresh Babu Kilaru, Sai Krishna Manohar Cheemakurthi, Vinodh Gunnam, 2021. "SOAR Solutions in PCI Compliance: Orchestrating Incident Response for Regulatory Security"ESP Journal of Engineering & Technology Advancements 1(2): 78-84.

[7] Jangampeta, S., Mallreddy, S.R., & Padamati, J.R. (2021). Anomaly Detection for Data Security in SIEM: Identifying Malicious Activity in Security Logs and User Sessions. 10(12), 295-298

[8] Vasa, Y. (2021). Robustness and adversarial attacks on generative models. International Journal for Research Publication and Seminar, 12(3), 462–471. https://doi.org/10.36676/jrps.v12.i3.1537

[9] Kilaru, N. B., Cheemakurthi, S. K. M., & Gunnam, V. (n.d.). Advanced Anomaly Detection In Banking: Detecting Emerging Threats Using Siem. International Journal of Computer Science and Mechatronics, 7(4), 28–33.

[10] Naresh Babu Kilaru. (2021). AUTOMATE DATA SCIENCE WORKFLOWS USING DATA ENGINEERING TECHNIQUES. International Journal for Research Publication and Seminar, 12(3), 521–530. https://doi.org/10.36676/jrps.v12.i3.1543

[11] Gunnam, V., & Kilaru, N. B. (2021). Securing Pci Data: Cloud Security Best Practices And Innovations. Nveo, 8(3), 418–424. https://doi.org/https://doi.org/10.53555/nveo.v8i3.5760

[12] Katikireddi, P. M., Singirikonda, P., & Vasa, Y. (2021). Revolutionizing DEVOPS with Quantum Computing: Accelerating CI/CD pipelines through Advanced Computational Techniques. Innovative Research Thoughts, 7(2), 97–103. https://doi.org/10.36676/irt.v7.i2.1482

[13] Vasa, Y. (2021). Quantum Information Technologies in cybersecurity: Developing unbreakable encryption for continuous integration environments. International Journal for Research Publication and Seminar, 12(2), 482–490. https://doi.org/10.36676/jrps.v12.i2.1539

[14] Jangampeta, S., Mallreddy, S. R., & Padamati, J. R. (2021). Data Security: Safeguarding the Digital Lifeline in an Era of Growing Threats. International Journal for Innovative Engineering and Management Research, 10(4), 630-632.

[15] Singirikonda, P., Jaini, S., & Vasa, Y. (2021). Develop Solutions To Detect And Mitigate Data Quality Issues In ML Models. NVEO - Natural Volatiles & Essential Oils, 8(4), 16968–16973. https://doi.org/https://doi.org/10.53555/nveo.v8i4.5771

[16] Vasa, Y. (2021). Develop Explainable AI (XAI) Solutions For Data Engineers. NVEO - Natural Volatiles & Essential Oils, 8(3), 425–432. https://doi.org/https://doi.org/10.53555/nveo.v8i3.5769

[17] Vasa, Y. (2021). Develop Explainable AI (XAI) Solutions For Data Engineers. NVEO - Natural Volatiles & Essential Oils, 8(3), 425–432. https://doi.org/https://doi.org/10.53555/nveo.v8i3.5769

[18] Kilaru, N. B., & Cheemakurthi, S. K. M. (2021). Techniques For Feature Engineering To Improve Ml Model Accuracy. NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal| NVEO, 194-200.

[19] Vasa, Y., Jaini, S., & Singirikonda, P. (2021). Design Scalable Data Pipelines For Ai Applications. NVEO - Natural Volatiles & Essential Oils, 8(1), 215–221. https://doi.org/https://doi.org/10.53555/nveo.v8i1.5772

[20] Sukender Reddy Mallreddy(2020).Cloud Data Security: Identifying Challenges and Implementing Solutions.JournalforEducators,TeachersandTrainers,Vol.11(1).96 -102.