CINEFORUM
ISSN: 0009-7039
Vol. 66. No. 2, 2026

Reinforcement Learning in Dynamic Environments: Applications and Limitations

Prof. Ankit Verma
School of Commerce and Business, Lovely Professional University

Abstract

When it comes to handling sequential decision-making challenges, especially in uncertain and dynamic situations, Reinforcement Learning (RL) has become a potent AI paradigm. Reinforcement learning (RL) allows agents to learn optimal behaviors through interaction with the environment by getting feedback in the form of rewards or penalties, unlike standard supervised learning systems. For situations that change over time and necessitate flexible decision-making approaches, RL is an excellent choice. the theories and methods of reinforcement learning in dynamic settings, with an emphasis on how agents figure out how to get the most out of their rewards in the long run regardless of how the circumstances change. It highlights the strengths of important algorithms in dealing with complicated, high-dimensional state spaces, including Q-learning, Deep Q-Networks (DQN), and policy gradient approaches. In addition to theoretical considerations, the paper delves into practical uses of RL in fields where flexibility and ongoing education are crucial, such as robots, autonomous cars, gaming, resource management, and financial trading.

Keywords: Reinforcement Learning (RL) , Dynamic Environments , Sequential Decision-Making , Q-Learning

Introduction

When faced with complicated decision-making difficulties in unpredictable, ever-changing contexts, Reinforcement Learning (RL) has emerged as a leading AI paradigm. Reinforcement learning differs from both supervised and unsupervised methods in that it emphasizes learning by doing, in which an agent acts in response to its environment and gets feedback in the kind of incentives or punishments. The agent is able to adapt to new situations with ease because it learns a strategy that maximizes cumulative rewards over time. The non-stationary character of dynamic environments—in which system states, transitions, and reward structures may alter over time—presents machine learning systems with distinct problems. Conventional, static models frequently do not work well in these kinds of environments. On the other hand, these settings are tailor-made for reinforcement learning, since this method is always learning and adapting. Because of its flexibility, RL is highly useful in domains where making decisions in real-time is crucial, such as robots, autonomous cars, gaming, resource allocation, and financial trading. The Markov Decision Process (MDP) is a mathematical model for making decisions in the face of uncertainty; it forms the basis of reinforcement learning. States, actions, incentives, and transition probabilities are essential parts of this model. These components are

used by RL algorithms to discover the best strategies, for example, Q-learning and policy gradient approaches. By incorporating deep learning approaches, RL's capabilities have been further strengthened, leading to the development of Deep Reinforcement Learning (DRL). DRL is capable of handling complicated situations and high-dimensional state spaces. In dynamic situations, reinforcement learning encounters multiple obstacles, notwithstanding its benefits. Performance can be hindered by issues like instability during training, sample inefficiency, and the exploration-exploitation trade-off. Furthermore, it is of utmost importance to guarantee safety and resilience in practical applications, particularly in risky fields such as healthcare and autonomous driving. what reinforcement learning can and cannot do in ever-changing settings. It delves into important algorithms, evaluates how well they work in practice, and talks about the problems that prevent them from being widely used. This paper aims to contribute to the development of more flexible, efficient, and trustworthy AI systems by offering a complete review of reinforcement learning. It will highlight both the potential and the restrictions of this method.

Fundamentals of Reinforcement Learning

One machine learning paradigm called Reinforcement Learning (RL) teaches agents to follow a predetermined path by observing and responding to their surroundings. Rewards and penalties are the basis of RL, as opposed to supervised learning, which uses labeled data to train models. Learning a policy that maximizes cumulative rewards over time is the purpose of RL agents.
At the core of reinforcement learning lies the interaction between four key components:

- **Agent:** The decision-maker that takes actions.
- **Environment:** The system with which the agent interacts.
- **State (S):** A representation of the current situation of the environment.
- **Action (A):** The set of possible moves the agent can take.
- **Reward (R):** Feedback received after taking an action, indicating its desirability.

A Markov Decision Process (MDP) is a common paradigm for modeling this interaction; it gives a mathematical basis for making decisions when faced with uncertainty. Rules for states, actions, transition probabilities, and reward functions characterize a Markov decision process (MDP). According to the Markov model, the past is irrelevant to predicting the future; all that matters are the present and any actions taken.
The agent's behavior is defined by the policy (π), a key idea in RL, which maps states to actions. Finding the best course of action to maximize the anticipated total benefit, sometimes called the return, is the ultimate objective. The idea of discounted rewards, in which present rewards are valued more highly than future ones, is a popular way to communicate this.
Value functions, which estimate the goodness of an agent's state (state-value function) or action (action-value function or Q-value) in a given state, are another significant notion. As time goes on, the agent is able to make better decisions with the aid of these functions.
Reinforcement learning methods can be broadly categorized into two types:

- **Model-Based RL:** The agent builds a model of the environment and uses it to plan actions.
- **Model-Free RL:** The agent learns directly from interactions without explicitly modeling the environment.

Within model-free approaches, further distinctions include:

- **Value-Based Methods:** Such as Q-learning, where the agent learns value functions to guide decisions.
- **Policy-Based Methods:** Where the agent directly learns the policy function.
- **Actor-Critic Methods:** A hybrid approach that combines value and policy learning.

Making a decision between exploring and exploiting is one of the primary obstacles in RL. The goal of the agent's exploration is to find higher rewards, but it is also important to take advantage of known acts that give large rewards. To handle this equilibrium, effective tactics like ε-greedy policies are employed.

The application of neural networks to approximate value functions or policies is known as Deep Reinforcement Learning (DRL), which emerged from the combination of deep learning and RL. Video games and real-world robots are examples of complicated situations that RL has been able to manage because to this.

At its core, reinforcement learning is all about maximizing long-term rewards, learning from interactions, and striking a balance between exploring and exploiting. These principles lay the groundwork for more complex RL methods and how to use them in dynamic, real-world settings.

Markov Decision Process and RL Framework

Markov decision process (MDP) is a formal framework for modeling sequential decision-making problems under uncertainty, and it is the mathematical basis for reinforcement learning (RL). It lays forth the rules by which an agent navigates its environment, takes decisions, and uses its results to improve its performance in the long run.

1. Components of a Markov Decision Process

An MDP is typically defined by a tuple **(S, A, P, R, γ)**, where:

- **S (States):** All potential environmental states are contained in this set. The current circumstance of an agent is represented by a state.
- **A (Actions):** The set of actions available to the agent in each state.
- **P (Transition Probability):** The probability of moving from one state to another after taking a specific action, denoted as $P(s' \mid s, a)$.
- **R (Reward Function):** The instantaneous benefit gained upon changing states as a result of initiating an action.
- **γ (Discount Factor):** A number between 0 and 1 that indicates how much weight to give to immediate gratification vs. future gratification.

2. The Markov Property

The Markov property is a basic assumption in Markov decision processes (MDPs). It asserts that the future state is solely dependent on the present state and the action taken, rather than on the order in which the states have occurred. Making decisions and learning efficiently are both made easier by this.

- **Implication:** Only the current state needs to be remembered by the agent; the complete history is unnecessary.

3. Policy and Value Functions

A policy (π) that maps states to actions defines the agent's behavior within the MDP framework. Discovering the best course of action that will maximize long-term benefits is the objective. Two important functions guide this process:

- **State-Value Function (V(s))**: Estimates the expected return from a given state.
- **Action-Value Function (Q(s, a))**: Estimates the expected return from taking a specific action in a given state.

These functions help the agent evaluate the quality of different actions and states.

4. The RL Learning Process

In reinforcement learning, the agent interacts with the environment in a loop:

1. Keep an eye on what is happening immediately
2. Choose an action (a) according to policy π.
3. Carry out the operation and go to the next state (s′)
4. Earn a prize (r) through
5. Revise the value or policy function

This iterative process allows the agent to improve its decision-making over time.

5. Types of RL within the MDP Framework

- **Model-Based RL:** The agent uses knowledge of transition probabilities (P) and reward function (R) to plan optimal actions.
- **Model-Free RL:** The agent learns optimal policies directly from experience without explicit knowledge of P and R (e.g., Q-learning).

6. Extensions of MDP

Standard MDP assumptions do not always apply in real-world circumstances due to the complexity of the environments:

- **Partially Observable MDP (POMDP):** Because it lacks a fully functional state observer, the agent must depend on data that is either noisy or incomplete.
- **Multi-Agent MDP:** In a multi-agent system, different entities engage with one another and their shared or conflicting goals.
- **Non-Stationary Environments:** Learning could become more difficult if transition probabilities and incentives are subject to change.

To enable systematic modeling of decision-making in unpredictable contexts, reinforcement learning is based on the Markov Decision Process. Machine learning problems (MDPs) offer a

structured way for agents to learn best policies by outlining states, actions, rewards, and transitions. Adaptability and long-term optimization are of the utmost importance in dynamic and real-world applications, making it imperative to comprehend this framework in order to analyze and design innovative RL algorithms.

Reinforcement Learning in Dynamic Environments

When circumstances change over time and necessitate ongoing adaptation, Reinforcement Learning (RL) shines. State changes, new incentive structures, and unknown transitions are the hallmarks of dynamic environments as opposed to static ones. Because the agent needs to learn the best course of action while simultaneously adjusting to the ever-changing environment, these qualities make decision-making more difficult.

1. Characteristics of Dynamic Environments

Dynamic environments differ from static ones in several key ways:

- **Non-stationarity:** The environment's behavior may change over time, altering state transitions and rewards.
- **Uncertainty:** Outcomes of actions may be probabilistic and unpredictable.
- **Temporal dependency:** Current decisions influence future states and rewards.
- **Real-time interaction:** The agent must continuously learn and respond to new information.

Because of these traits, RL agents need to be adaptable and able to change their policies in response to new information.

2. Adaptation and Learning in Changing Conditions

Reinforcement learning (RL) agents in dynamic settings need to be able to both learn from their mistakes and adjust to new circumstances. In dynamic environments, the assumption of a stationary environment is often not met by traditional RL approaches.

- Value functions and policies are updated continuously by agents.
- To better adapt to changes, learning rates can be modified.
- In order to find new best actions when circumstances change, exploration tactics are kept up to date.

Applications like autonomous systems, robotics, and financial markets rely on this flexibility.

3. Techniques for Handling Dynamic Environments

Several techniques have been developed to improve RL performance in dynamic settings:

- **Online Learning:** The availability of new data allows agents to incrementally update their expertise.
- **Adaptive Learning Rates:** Make it easier to adjust to new circumstances.
- **Experience Replay with Prioritization:** Focuses learning on more relevant or recent experiences.
- **Meta-Learning:** Enables agents to learn how to adapt quickly to new environments.

- **Transfer Learning:** Applies knowledge from previous environments to new but related scenarios.

These methods help RL systems remain effective even when conditions are unstable.

4. Role of Deep Reinforcement Learning

Adapting to changing conditions is now much easier thanks to Deep Reinforcement Learning (DRL), an approach that combines deep learning with RL. Agents can generalize across different contexts because neural networks make it possible to describe complicated, high-dimensional state spaces.

- Large-scale sensory inputs, including pictures and audio, can be processed by DRL.
- As a result, it facilitates learning even in highly interdependent and pattern-laden contexts.
- Robots and self-driving cars are two examples of real-time systems that can benefit from it.

5. Real-World Applications

Reinforcement learning in dynamic environments is widely applied across various domains:

- **Autonomous Vehicles:** Adapting to changing traffic conditions and road environments.
- **Robotics:** Learning to perform tasks in unpredictable physical settings.
- **Finance:** Adjusting trading strategies based on market fluctuations.
- **Game Playing:** Responding to evolving strategies of opponents.
- **Resource Management:** Optimizing allocation in changing demand scenarios.

6. Challenges in Dynamic Environments

Despite its strengths, RL faces several challenges in dynamic settings:

- **Instability in learning:** Frequent changes can disrupt convergence.
- **Delayed rewards:** It may be difficult to associate actions with long-term outcomes.
- **Exploration difficulty:** Identifying new optimal strategies in changing environments is complex.
- **Scalability issues:** Large and complex environments require significant computational resources.

Decisions in uncertain situations can be better supported by reinforcement learning's robust architecture, which allows for both adaptation and continual learning. Robot learning systems are able to adapt to new environments because they use cutting-edge methods like deep learning, meta-learning, and transfer learning. To successfully implement RL in dynamic real-world systems, however, issues of stability, efficiency, and scalability must be resolved.

Conclusion

When faced with complicated decision-making difficulties in ever-changing situations, reinforcement learning (RL) has proven to be a strong and adaptable framework. Applications in the actual world with inherent uncertainty and variability are ideal for its interaction-based

learning, adaptability, and optimization of long-term rewards. The application of RL has shown great promise in enabling intelligent and adaptive behavior across various domains, including robotics, autonomous systems, finance, and resource management. This research has delved into the fundamentals of reinforcement learning, including topics such as the Markov Decision Process, important algorithms, and the difficulties of working in dynamic settings. It exemplifies how RL agents may adapt to non-stationary situations by continuously updating their value functions and rules. Deep reinforcement learning (DRL) methods were developed as a result of RL systems' increased capacity to process complicated and high-dimensional input through the incorporation of deep learning techniques. Nevertheless, there are still a number of restrictions, even with these improvements. Many practical issues prevent RL from being widely used, including inefficient samples, instability during training, difficulties with exploration, and high processing requirements. Another important issue is making sure technology can be used safely, robustly, and ethically in high-stakes situations. The future seems bright for finding solutions to these problems, thanks to new methods like model-based reinforcement learning, meta-learning, and multi-agent systems. To fully utilize reinforcement learning in dynamic and real-world settings, it is crucial to create RL frameworks that are more efficient, scalable, and flexible. Although reinforcement learning has achieved impressive strides, it will only work in ever-changing settings if algorithms are always being refined, computational efficiency is enhanced, and real-world restrictions are thoroughly considered. By tackling these obstacles, RL has the potential to be a game-changer in developing AI and autonomous systems of the future.

References (APA Style)

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning, 8*(3–4), 279–292. https://doi.org/10.1007/BF00992698

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533. https://doi.org/10.1038/nature14236

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484–489. https://doi.org/10.1038/nature16961

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. *International Conference on Learning Representations (ICLR)*.

Schulman, J., Levine, S., Moritz, P., Jordan, M., & Abbeel, P. (2015). Trust region policy optimization. *International Conference on Machine Learning (ICML)*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, *32*(11), 1238–1274.

Busoniu, L., Babuska, R., & De Schutter, B. (2008). A comprehensive survey of multi-agent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C, 38*(2), 156–172.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine, 34*(6), 26–38.

Dulac-Arnold, G., Mankowitz, D., & Hester, T. (2019). Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends in Machine Learning, 11*(3–4), 219–354.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*.