Applied Science and Biotechnology Journal for Advanced Research e-ISSN: 2583-553X
Peer Reviewed Journal
Volume-1, Issue-1 (July 2022), Page: 20-24 https://doi.org/10.31033/abjar.1.1.5

Effects of Self-Driven Reinforcement Learning

Sanjay Rao[1] and Rohit Bharat[2]
[1]P.G. Student, Department of Engineering, Walchand Institute of Technology (W.I.T), Solapur, Maharashtra, India
[2]P.G. Student, Department of Engineering, Walchand Institute of Technology (W.I.T), Solapur, Maharashtra, India

[1]Corresponding Author: sanjay_rao_326@gmail.com

ABSTRACT
Structured Learning, unstructured Learning, and reinforcement Learning is the three main components of machine Learning (ML). In this paper, we'll focus on reinforcement Learning, which is the final stage. There are numerous methods of reinforcement learning, and we'll go over some of the more popular ones. Software agents that use reinforcement learning to maximize their rewards in a given environment are known as reinforcement agents. Extrinsic and intrinsic rewards are the two main classifications of rewards. It's a specific outcome we get after following a set of rules and accomplishing a specific goal. Rather than monetary gain, a better example of an intrinsic reward is the agent's eagerness to learn newly acquired expertise that may prove beneficial in the future.

Keywords: reinforcement learning, structure, limitations, agents

I. INTRODUCTION

The ML and AI communities have seen a rise in the use of reinforcement learning in the past seven to nine years. Program subjects use incentives as a result of fruitful tests and penalties incorrect ones, deprived of teaching the representative how to execute the specified job. Some issues do arise with these procedures. Our focus here is on those issues and the possible remedies. Structured Learning, unstructured Learning, and reinforcement learning all fall under the umbrella term "machine Learning" (ML).

As a result of its broad applicability, reinforcement learning has been explored in a variety of sports ground, including game with control theory, as well as operations research and information theory. In the area of operations research and control, bolstering erudition is referred to as guesstimate vigorous programming or neuro energetic programming.

However, it is in the principle of optimum mechanism, which is less apprehensive with learning or approximation and more focused on the existence and representation of prime elucidations and set of rules for their accurate calculation, that the problems of relevance to reinforcement learning have been examined. Bounded rationality and equilibrium can be explained by using reinforcement learning techniques in economics and game theory.

II. STRUCTURED LEARNING

Using a machine learning technique known as "structured Learning," we can find a function that best fits the training dataset. Predicting future results from a random input depends on how well the machine is trained with the dataset. It gathers a function from the available options.

The labelled training data, which includes a variety of training examples in diverse sets, known as a vector, and a corresponding value sometimes supervisory signals are referred to as building blocks of structured Learning. Suggested examples are created using an algorithm in structured Learning that analyses the training data and creates an inferred function. Figure 1 illustrates the fundamental framework of structured Learning.

e-ISSN: 2583-553X
Peer Reviewed Journal
https://doi.org/10.31033/abjar.1.1.5

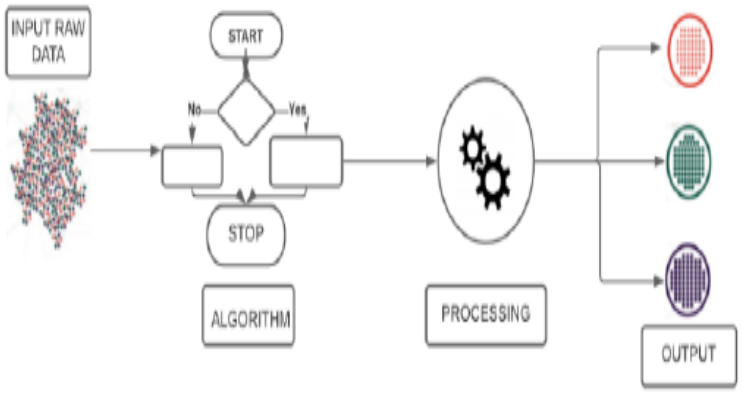

Figure 1: Structure of Directed Learning

III. UNSTRUCTURED LEARNING

As the name suggests, unstructured Learning studies without any pre-existing labels, patterns in a dataset that does not require any or very little human supervision. When compared to structured learning, which uses human-labeled data, unstructured Learning, or self-organization, allows users to model the probability densities over inputs. The unstructured learning process is shown in Figure 2.

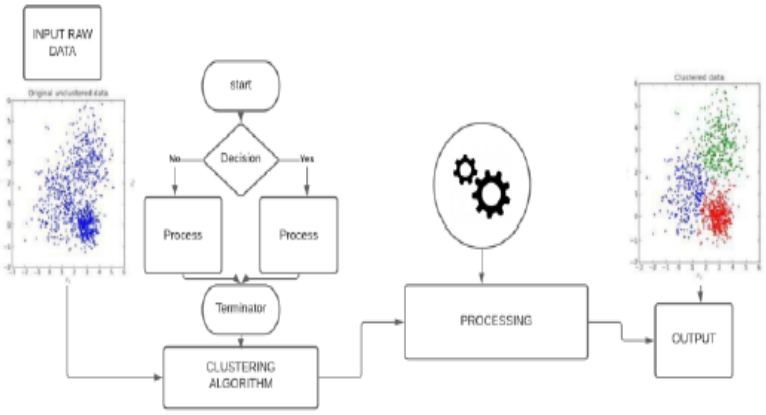

Figure 2: Unsupervised Learning

IV. THE LEARNING OF REINFORCEMENT

RL is a word used to describe the process of building machine learning models that may then be utilised to make decisions. In a potentially complicated and uncertain environment, the agent begins to learn how to achieve its purpose. A circumstance that is eerily reminiscent of a video game must be faced by artificial intelligence in the real world as well. Using the trial and error process, the computer tries to solve the problem. As a result of its actions, artificial intelligence is either rewarded or penalised. Maximizing the entire reward is the goal.

It is up to the creator to determine the game's reward policy or regulations, but he or she not give the AI or agent any advice on how to deal with the game's challenges. It's up to the agent to figure it out, and the agent's primary motivation is to maximise the profits. As the game progresses, it gains abilities and implements strategies through random trial and error, but this is only the beginning. If you want to show off a machine's ingenuity, you should use reinforcement Learning.

To put it another way, ability to learn is important to reinforcement learning or software agents undertake activities in a specific environment in order to maximise the cumulative reward.

RL varies from structured Learning in that no identified input data or output pairs are required. Exploration (of previously unexplored land) and rectified exploitation are central to RL's philosophy (of current knowledge).

It is via the application of dynamic programming techniques that we may develop RL algorithms in this scenario of describing the surrounding circumstances. Unlike traditional dynamic programming methods, Reinforcement Learning does not necessitate a detailed knowledge of the Markov Decision Process mathematical model. For large MDPs, when the same procedures are unfeasible, it is aimed at using this technique. The following image depicts general reinforcement learning (Fig. 3).

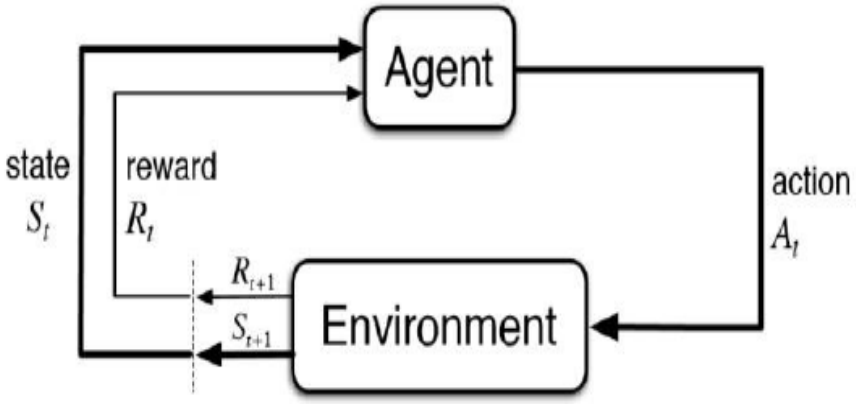

Figure 3: Reinforcement Learning

Reinforcement Learning has seen numerous advancements. Games like Pacman and Minigrid are built around the idea Reward-based Learning. The agent in reinforcement learning seeks to maximise its own gain. There is a lot of information in this paper about reinforcement learning and the numerous approaches employed in this discipline.

4.1 Examples of Learning by Repetition

- Making prosthetic legs that can recognise a user's gait patterns is a difficult problem for a coder. As a result, the Anatomical sensors in the Stanford Neuromuscular Biomechanics Laboratory have been used to create a walking model that may be used to study human gait.
- In terms of how fast or slow it should move, we have no way of knowing what the car will need. This would require the usage of numerous lengthy "if-then" statements in the code. Instead of employing these lines, the programmer employs an agent that is able to learn from its own experiences, in which it receives rewards and punishments for its conduct.
- An automobile would be a good example of RL. The computer does not get driving directions while driving. The machine will be able to learn from its mistakes thanks to the programmer.

4.2 Reinforcement-based Learning Limitations

RL is all about setting up the environment, and this is the most difficult part. There are occasions when it's simple, like in Atari games or chess. But real-world difficulties might raise a lot of questions when it comes to this. Furthermore, transitioning the model from the simulation environment to the real world can be a challenge. Another problem is scaling the neural network that controls our agent. Network communication would be impossible without a reward and penalty system, which would lead. Another issue is when the agent does not accomplish the task in the proper manner. A major issue could arise, for example, if we require our agent to walk straight but he begins to leap.

The development of competence rather than focusing on external goals is the goal of these activities. While this is true for new challenges, machine learning methods generally do not. It's these "building blocks" that an agent acquires throughout its competency that allow it to deal with a wide range of problems that may arise during its lifespan.

Internal rewards were introduced because extrinsic rewards could be scarce or nonexistent at times. When a player progresses in a game, the agent receives. When an explicit incentive isn't available, agents use intrinsic motivation to satisfy their natural curiosity and learn more about their surroundings. Intrinsic and extrinsic rewards are available in RL. There was no need to introduce intrinsic rewards before because there were only extrinsic ones. We believe that intrinsic motivation is the best way to address each of these issues via Deep Reinforcement Learning.

4.3 Sparse Payback

RL algorithms work best when the agent receives a reward for completing an action, which is common in most real-world settings where incentives are abundant. In environments with few rewards, only after completing a lengthy series of actions. During the game Montezuma's Revenge, a player moves from room to room picking up various items and only earns a

prize when it exits the room or picks up an item and obtains the reward. This game is an excellent example of a game with a low reward function. Because the computer or agent isn't given any guidance on how to enhance its exploration policy in these low-reward situations, solving them is nearly impossible when using the exploration policies described above. Consequently, it is unable to locate its reward for the provided assignment. Instead of enhancing exploration policies, an intermediary function known as "dense reward" typically introduced connected job in order to overcome this issue. The additional reward function also introduces certain unforeseen faults, necessitating the use of expert knowledge in several cases.

4.4 Constructing a Strong Statewide Presence

A decent state representation in real life is said to be Markovian, capturing genuine value of the policy, which should be low dimensions and generalizable. In reinforcement learning, the representation of relevant states is critical. A sophisticated non-linear transformation of the agent's current location and desired destination location will then be needed to decide the direction of movement the agent must take.

If the reward is minimal or sparse, the agent doesn't learn anything from the interaction, even when the interaction is considered to be rich in information. This difficulty is exacerbated in typical RL since back propagation of reward signals is the only possible learning mechanism, and because sound or noise is present in its raw state. To better comprehend it, we'll look at a job is trying travel a specific a specific. Let's say that a computer or agent is capable of accessing pixels in the empty region above. Learning a task in this feature space with other calculations that can be learned to create an effective forward model. Disentangled features are the best and most effective approach to accomplish this. On the other hand, independently of particular generalised to include all future tasks. Many books have been written to help us better grasp the importance of effective state representation.

4.5 Abstractions of Actions in Time

Because of execution of the option. A choice's "length" refers to the number of steps that must be taken once a choice has been made, and this number is usually predetermined.

Because the process, the agent must carry this reward with them as they complete their list of tasks that must be made, and abstract actions aid in this process, so they can be regarded as an important part of the Learning process. This can also be used to resolve credit assignment issues. Options, or high-level actions, are employed. The interoption policy usually selects the choices that will be implemented. There may be a delay between the time when the reward is given and when it is received, but these delays may be necessary for the process of acquiring the prize itself.

Additionally, the agent must figure out which action is most important so that they can earn the prize they desire in this scenario. This technique can take a long time if the action sequence is quite large.

4.6 Making a Course of Learning

As part of the research known as multitask RL, a single agent works on more than one or multiple challenges at once. Every step of the process should be described in detail using a timetable. Learning is made considerably simpler when samples are presented in a precise or meaningful order, as may be deduced from this fact. Furthermore, may organise behaviours are closer together and difficult.

For example, teaching the robot how to first grab an ice cube would be an effective method for helping the robot move the ice cube on its own. In this way, the robot will be able to learn from its own experiences. Acquired while holding the cube. In order to move an ice cube, you need to know how joints move, which can be learnt through the first phase. Without this information, it would be extremely impossible. One of the most common techniques is to employ to establish a starting point, experts' scores can be substituted for pre-determined tasks or action sequences. Other ways exist as well, such as Florensa's method, which relies on strong assumptions, and Wu's method, which utilises a task decomposition approach. It shows us that curriculum experts are almost always required in standard procedures.

V. CONCLUSION

The field of RL has made numerous strides forward. RL has been utilised in a variety of games, including Pacman, Atari games, and more. In order to discover the machine's creative potential, RL is the most efficient means.

As a result, the Learning of RL is expected to grow in importance in the near future. The good news is that intrinsic motivation can be used to overcome these difficulties. As a further stage in reinforcement learning, intrinsic motivation will be the focus.

Automation, maintenance, and optimization of energy use are some of the uses of RL in the workplace. It's no secret that many sectors, including banking, are looking at using RL to power artificial intelligence-based Learning programmes. Although the trial-and-error method of teaching robots can take a long time, it is used because it helps the robots to use their abilities to better assess the real environment and to carry out tasks by utilising their abilities.

REFERENCES

1. Ng, A. Y., & Russell, S. J. (2000). Algorithms for inverse reinforcement learning. *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 663–670.
2. Karpathy, & M. Van De Panne. (2012). *Curriculum learning for motor skills*.
3. Barto, A. G. (2013). Intrinsic motivation and reinforcement learning. *Intrinsically Motivated Learning in Natural and Artificial Systems, Berlin, Heidelberg: Springer*, pp. 17–47.
4. Wilson, A. Fern, & P. Tadepalli. (2014). Using trajectory data to improve bayesian optimization for reinforcement learning. *J. Mach. Learn. Res*.
5. N. Bougie, & R. Ichise. (2020). Skill-based curiosity for intrinsically motivated reinforcement learning. *Mach. Learn*.
6. T. D. Kulkarni, K. R. Narasimhan, A. Saeedi, & J. B. Tenenbaum. (2016). *Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation*.
7. Abhay Singh Hyanki, Shweta Meena, & Tarun Kumar. (2021). A survey on intrinsically motivated reinforcement learning. *International Journal of Engineering Research & Technology*, *10*(5), 1150-1153.
8. Kulkarni, Tejas D., Narasimhan, Karthik R., Saeedi, Ardavan, & Tenenbaum, Joshua B. (2016). Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation. *Proceedings of the 30th International Conference on Neural Information Processing Systems*.
9. R. Salakhutdinov, & A. Mnih. (2008). *Bayesian probabilistic matrix factorization using markov chain Monte Carlo*.
10. Raffin, S. Höfer, R. Jonschkowski, O. Brock, & F. Stulp. (2017). Unsupervised learning of state representations for multiple tasks. *ICLR*.