A Survey on Intrinsically Motivated Reinforcement Learning

Shweta Meena
Department of Software Engineering
Delhi Technological University
New Delhi, India

Abhay Singh Hyanki
Department of Software Engineering
Delhi Technological University
New Delhi, India

Tarun Kumar
Department of Software Engineering
Delhi Technological University
New Delhi, India

Abstract—**Machine learning (ML) consists of mainly three further studies that are supervised learning, unsupervised learning, and reinforcement learning. In this paper, we are going to look at the later part that is reinforcement learning. We will review the various methods that have been and are being used in the field of reinforcement learning. Reinforcement learning basically deals with how software agents tend to behave to maximize its reward in a given environment. Rewards are also of two types: extrinsic and intrinsic. Extrinsic reward is some specific outcome that we achieve after following certain procedures and completing a given task. In contrast, the intrinsic reward is the curiosity of the agent to develop new skills that might be useful in the coming future.**

Keywords— Intrinsic Rewards, Extrinsic Rewards, Reinforcement learning

I. INTRODUCTION

Reinforcement learning has developed its place in ML and AI communities within last 7-9 years. It offers to build a way of programming subjects by the medium of rewards for successful trials and punishment for wrong attempts without telling the agent the procedure to complete the given task .But these processes face various problems. In this paper we look at those problems and also look for solutions. Machine learning have three sub-parts that are supervised learning, unsupervised learning, and reinforcement learning. All three parts are depicted in Fig 1.

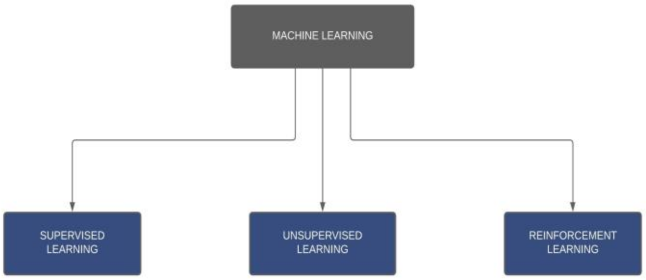

Fig. 1 Sub-parts of machine learning

A. Supervised Learning

Supervised learning is the machine learning technique in which we develop a function which best fits the training dataset. Depending on the training with the dataset, the machine is able to predict future outputs given a random input. It uses a function that is gathered from available labeled training data, which consists of some sets of various training examples. In supervised learning, all the examples consists an input object that is also called vector and an output value also called as supervisory signal. The algorithm that is used in supervised learning examines the given training data and develops an inferred function, that is generally used for making new examples. Figure 2 below depicts the basic structure of supervised learning.

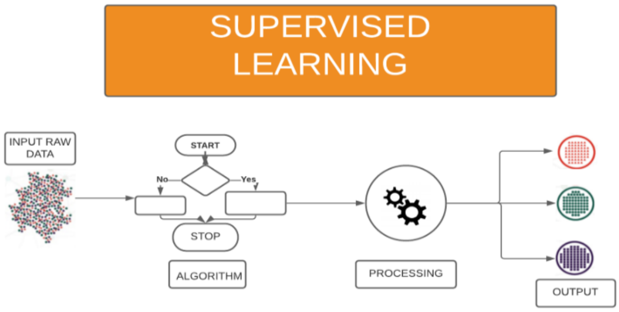

Fig. 2 Structure of Supervised learning

B. Unsupervised Learning

Unsupervised learning is a study in which it looks for previous undetected patterns in a given data set which does not contain any pre-existing labels and with no or very less human interaction or supervision. In comparison with supervised learning, which is known for using human-labeled data, the unsupervised learning, which can also be called as self-organization, permits users for required modeling of the probability densities over the inputs. The structure of unsupervised learning is shown in Fig 3.

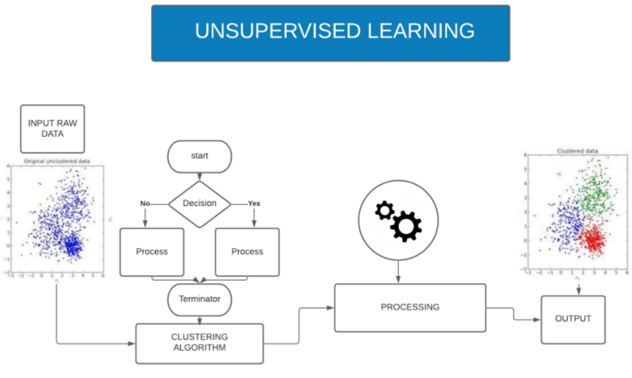

Fig. 3 Structure of Unsupervised learning

C. *Reinforcement Learning*

Reinforcement learning (RL) is considered to be an area that is concerned with how agents or software agents perform actions in a given environment with the main motive of maximizing the cumulative reward [1].

RL is different from supervised learning as it does not requires labeled input data or output pairs to be available, it also differs as it does not requires various sub-optimal actions that are expected to be accurate or corrected. The focus of RL is to maintain an equilibrium between exploration (of the uncharted territory) and exploitation (of the current knowledge)

The algorithms for RL in this context use dynamic programming techniques. Hence, the Markov decision process is used for stating the environment. Reinforcement learning differs from classic dynamic programming algorithms because reinforcement learning usually does not assume an understanding of the exact mathematical model of the Markov Decision Process,[2] and they target large MDPs where the same methods become infeasible. The methodology used in general reinforcement learning is depicted in the below diagram Fig 4.

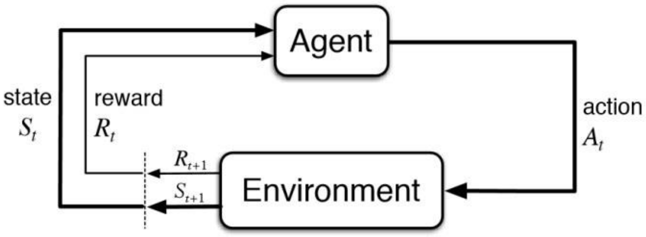

Fig. 4 Reinforcement learning

There have been many breakthroughs in reinforcement learning. Many games such as Pacman, Minigrid, Atari games, etc[3] are developed on the concept of reinforcement learning. In reinforcement learning, the agent tends to maximize its reward. In this paper, we are going to look deep into the field of reinforcement learning and describe the various methods used in reinforcement learning.

II. OVERVIEW OF REINFORCEMENT LEARNING

RL is termed as the training of some machine learning models which are used to make some decisions or sequence of decisions. The agent starts learning to achieve its goal in a potentially complex and uncertain environment [3]. In RL artificial intelligence has to face a situation that is very similar to a game. The computer tries to solve the problem by trial and error method. In order to achieve the task, artificial intelligence gains rewards or gets a penalty for the action it performs. The aim is to maximize the total reward [4].

The reward policy of the game or the rules of the game is set by the designer, but the designer does not give the artificial intelligence or the agent suggestions to solve the game. The agent has to figure it out by himself, and the main motive of the agent is to maximize its rewards. It basically starts with random trial and error methods, but as the game proceeds, it develops skills and starts implementing strategies [5]. Currently, we can say that reinforcement learning is the most efficient method to showcase the creativity of a machine.[6]

A. *Examples of reinforcement learning*

Reinforcement learning was restricted in the past because the infrastructure of the computers was not up to the mark. Actual development happened after the production of backgammon AI.

- Good examples of RL would be that of a car. While driving, the computer doesn't get driving instructions. The programmer will allow the machine to learn from its failures.
- We cannot predict the requirements of the car as to when it should drive slow or when it should go fast. To program this, the programmer would need to use lengthy multiple "if-then" statements. Instead of using these statements, the programmer uses a agent that can learn from its experience in which it gets rewards and penalties for its actions.
- Development of prosthetic legs is a challenging task for a coder as it would require extensive coding to detect the patterns of a walk. Hence, Stanford Neuromuscular Biomechanics Laboratory has developed a musculoskeletal model based on RL that has the ability to detect people's walking patterns.

B. *Challenges with reinforcement learning*

Setting up the environment is the main challenge in RL. It can be easy sometimes, like in the cases of Atari games, chess, etc. but when it comes to real-world problems, it can cause a lot of questions. Moreover, transferring the model into the real world from the simulation environment can get tricky. Scaling the neural network, which is controlling our agent, is another challenge. System of reward and penalties is the only way to communicate with the network, which would result in catastrophic forgetting, which means that when we gain new knowledge causes old knowledge to get erased. Another challenge is when the agent does not perform the task in the required way. For example, If we need our agent to walk straight, but it starts jumping, it can also be a big problem.

RL has two types of rewards that are intrinsic and extrinsic. Earlier, only extrinsic rewards were present, so what was the need of introducing intrinsic rewards.

Sometimes extrinsic rewards were sparse and even not present hence the introduction of intrinsic rewards was necessary. Intrinsic reward is basically the curiosity of the agent to develop new skills as the agent goes further in the game. Intrinsic motivation leads agents to explore the environment driven by curiosity when an explicit reward is not present. Such activities help in the development of competence instead of focusing on external goals. In contrast, generally, machine learning algorithms do not perform well with new problems when they arise after some time. The skills gained during competence are "building blocks," using which an agent possesses the ability to solve various issues as they appear in its lifetime.

We have identified four challenges in Deep Reinforcement Learning to which intrinsic motivation is the answer.

a) Sparse rewards

Major RL algorithms function in environments where there is no shortage of rewards, *i.e.*, the agent will gain a reward after completing an action. The agent will receive a reward signal after executing an immense sequence of steps in cases of an environment with sparse rewards. In a game called Montezuma's revenge, the agent has to pick up different objects while moving from one room to another; the agent gets rewards only when it picks up an object or receives the reward when it has left the room. This game is a typical example of a sparse reward function. Such environments that have sparse rewards are very difficult to solve considering the exploration policies mentioned above as the computer or agent does not get any hint in the way as to how to improve its exploration policy. Due to this, the agent fails to find its rewards concerning the given task [7]. Generally, to encounter this problem, an intermediate function called dense reward is added to the prize associated with the task rather than focusing on improving the exploration policies [8]. And due to this adding of the additional reward function, some more unexpected errors arise, and most of the time, expert knowledge is required.

b) Building a good state representation

It is said that, in RL, the good state representation has to be Markovian, representing the policy's actual value, which should be of low dimesions and be able to generalize [9]. Using these feature space in learning a given task can speed up the learning process and can even help in some other computations that can be learning an effective forward model [10]. Constructing a minimal feature space with disentangle features is the best and most effective way to do this.

Relevant state representation is significant in reinforcement learning. To understand it, we will consider a navigation task in which the agent aims to get to a specified location in a given space. Suppose the computer or agent is able to access pixels input from the empty space above. In that case, the agent will then have to determine its existing position and the required target position by using complex non-linear transformations in order to get the idea in which direction it has to move. In contrast to this [11], if the agent already have the idea or access to its existing position, then the only work of the agent will be comparing its vertical and horizontal positions and check whether they are greater than, smaller than, or equal to the target. In standard RL , this problem is worsened due to the availability of back propagation of reward signal as the only available process of learning, and also due to the availability or presence of sound or noise in its raw state. The interaction among agents is considered to be rich in information, yet the agent doesn't gain any knowledge from the interaction if the reward is less or sparse. Moreover, the state representation, which has a reward associated with it, is entirely dependant on a specific task and cannot take other tasks into account. And in contrast, the state representation, which is learned without associating with any task, can be generalized for the remaining tasks. There have been many works regarding the understanding of good representation of state.

c) Temporal abstraction of actions

High-level actions, usually called options, are used under the Temporal abstraction of actions, and they can be executed at different times [2]. At the time of execution of the option, due to the presence of an intra option policy [12], the action can be defined in each state. The number of actions executed after choosing an option is usually termed as the length of the option and is usually fixed. The options that are to be accomplished are usually selected by the inter option policy. By using options, the number of decisions that are to be taken is reduced, and abstract actions help in attaining so hence abstract actions can be considered as a key element in accelerating the learning process. The credit assignment [2]problem can also be solved by this easily. In this problem, the rewards arise or occurs with a slight delay and don't majorly affect any of the temporary distant states which have occurred before it, however these temporary distant states may be required in the process of obtaining the reward. In fact, the agent will have to carry the reward throughout the list of actions performed in order to apply the state action tuple which was involved at the very beginning. This problem also focuses on determining the importance of the action, and by performing which action the agent can get the reward[13]. If the action sequence is very large, this procedure can be very slow.

d) Building a curriculum

It usually occurs in the framework of the study known as multi-task RL in which a single agent tries to solve more than one task or several tasks [14]. It is all about describing a schedule throughout the process. It can be gathered from the observation that when examples are arranged in a specific or meaningful order [11], the learning becomes much easier. Moreover, a curriculum might arrange or organize tasks or actions in a way that they are more

Algorithm	Merits	Demerits
SARSA	Always tries to find the best policy for exploration.	Does not contain the ability to function efficiently at local minima
DQN	It is comparatively faster in training	It estimates more Values of Q than the desired amount
Q-Learning	Explores more than one policy at a time	It is a bit slow but can handle alternative routes
Monte Carlo	Deals with unknown model risen difficulty i.e. strategy decision	After one episode the strategy can change
Value-function approximation	more stable and less prone to failure	Less sample efficient
Policy Gradient actor critic	Finds the Optimal Deterministic Policy	Typically converge to a local optimum rather than a global optimum

close to each other and can also be increasingly complex. For example, in helping the robot move an ice cube by itself, an effective curriculum would be to teach the robot how to grab the ice cube first. In this way, the robot can learn from the skills that it gained during the process of grasping the cube. Since moving an ice cube requires a vast amount of knowledge like the movement of joints, so without any prior knowledge of movements of joints, which can be learned through the first process, it would be very difficult. Making pre-specified tasks or action sequences as the curriculum is one of the standard methods or expert scores can also be used, which acts as the baseline score [15] Some other methods are also there like one method which was given by Florensa requires strong assumptions, and another method given by Wu relies on task decomposition. It tells us that, generally, in standard methods, there is a requirement of an expert in curriculum learning always.

In table 1, we have discussed the various algorithms used in the field of reinforcement learning. The table consists of name of the algorithm and their merits and demerits.

III. CONCLUSION

There have been many breakthroughs in RL. RL has been used in many games such as Pacman, Atari games, etc. RL is the most efficient way of finding out the creativity of the machine.

Hence, the study of RL is going to become more important in the near future. In this paper, we outlined the importance of reinforcement learning and the challenges that arise in reinforcement learning. However, these challenges can be tackled by intrinsic motivation as a reward. Intrinsic motivation is the next big step in reinforcement learning.

RL is being used in automation of operations, maintenance of machinery, optimization of usage of energy. Many industries like finance are also considering RL to power training systems which work on artificial intelligence. We know that the trial-and-error technique of training robots is time consuming, then also this is practised as it allows the robots in using their skills in order to evaluate the real world more efficiently and also complete the tasks using their skills.

REFERENCES

[1] P. R. Montague, “Reinforcement Learning: An Introduction, by Sutton, R.S. and Barto, A.G.,” *Trends Cogn. Sci.*, 1999, doi: 10.1016/s1364-6613(99)01331-5.

[2] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction,” *IEEE Trans. Neural Networks*, 1998, doi: 10.1109/tnn.1998.712192.

[3] N. Bougie and R. Ichise, “Skill-based curiosity for intrinsically motivated reinforcement learning,” *Mach. Learn.*, 2020, doi: 10.1007/s10994-019-05845-8.

[4] A. G. Barto, “Intrinsic motivation and reinforcement learning,” in *Intrinsically Motivated Learning in Natural and Artificial Systems*, 2013.

[5] T. D. Kulkarni, K. R. Narasimhan, A. Saeedi, and J. B. Tenenbaum, “Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation,” 2016.

[6] S. Singh, A. G. Barto, and N. Chentanez, “Intrinsically motivated reinforcement learning,” 2005.

[7] R. Salakhutdinov and A. Mnih, “Bayesian probabilistic matrix factorization using markov chain Monte Carlo,” 2008, doi: 10.1145/1390156.1390267.

[8] T. Chen and W. Su, “Indirect Customer-to-Customer Energy Trading with Reinforcement Learning,” *IEEE Trans. Smart Grid*, 2019, doi: 10.1109/TSG.2018.2857449.

[9] C. A. Schroeder de Witt, J. N. Foerster, G. Farquhar, P. H. S. Torr, W. Böhmer, and S. Whiteson, “Multi-agent common knowledge reinforcement learning,” 2019.

[10] A. Raffin, S. Höfer, R. Jonschkowski, O. Brock, and F. Stulp, “Unsupervised Learning of State Representations for Multiple Tasks,” *Iclr*, 2017.

[11] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, doi: 10.1109/TPAMI.2013.50.

[12] O. Nachum, H. Lee, S. Gu, and S. Levine, “Data-efficient hierarchical reinforcement learning,” 2018.

[13] P. Henderson, W. Di Chang, P. L. Bacon, D. Meger, J. Pineau, and D. Precup, “Optiongan: Learning joint reward-policy options using generative adversarial inverse reinforcement learning,” 2018.

[14] A. Wilson, A. Fern, and P. Tadepalli, “Using trajectory data to improve Bayesian Optimization for Reinforcement Learning,” *J. Mach. Learn. Res.*, 2014.

[15] A. Karpathy and M. Van De Panne, “Curriculum learning for motor skills,” 2012.