IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 10, Issue 1, January 2020, ||Series -VI|| PP 35-40

History of Reinforcement Learning

#1P.Sushma, #2, Dr.Yogesh Kumar Sharma, #3, Dr.S. Naga Prasad
Ph.D Scholar, Dept of CSE, JJTU University.
Guide, Dept of CSE, JJTU University.
Co- Guide, Dept of CSE, JJTU University

Received 14 January 2020; Accepted 30 January 2020

Abstract: Reinforcement learning (RL) can be subdivided into two fundamental problems: learning and planning. The goal of learning is for an agent to improve its policy from its interactions with the world. Thegoalofplanningisforanagenttoimproveitspolicywithoutfurtherinteractionwiththe world. The agent can deliberate, reason, ponder, think or search, so as to find the best behavior in the available computation time.Despite the apparent differences between these two problems, they are intimately related. During learning, the agent interacts with the real world, by executing actions and observing their consequences. During planning the agent can interact with a model of the world: by simulating actions andobserving theirconsequences. Inbothcasestheagentupdatesitspolicyfromitsexperience. Our thesis is that an agent can both learn and plan effectively using reinforcement learning algorithms.

I. INTRODUCTION

Reinforcement learning is like many topics with names ending in -ing, such as machine learning, planning, and mountaineering, in that it is simultaneously a problem, a class of solution methods that work well on the class of problems, and the field that studies these problems and their solution methods. Reinforcement learning problems involve learning what to do—how to map situations to actions—so as to maximize a numerical reward signal. In an essential way they are closed-loop problems because the learning system's actions influence its later inputs. Moreover, the learner is not told which actions to take, as in many forms of machine learning, but instead must discover which actions yield the most reward by trying them out. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These three characteristics—being closed-loop in an essential way, not having direct instructions as to what actions to take, and where the consequences of actions, including reward signals, play out over extended time periods—are the three most important distinguishing features of reinforcement learning problems.

A full specification of reinforcement learning problems in terms of optimal control of Markov decision processes must wait until Chapter 3, but the basic idea is simply to capture the most important aspects of the real problem facing a learning agent interacting with its environment to achieve a goal. Clearly, such an agent must be able to sense the state of the environment to some extent and must be able to take actions that affect the state. The agent also must have a goal or goals relating to the state of the environment. The formulation is intended to include just these three aspects—sensation, action, and goal—in their simplest possible forms without trivializing any of them.

Any method that is well suited to solving this kind of problem we consider to be a reinforcement learning method. Reinforcement learning is different from supervised learning, the kind of learning studied in most current research in field of machine learning. Supervised learning is learning from a training set of labeled examples provided by a knowledgeable external supervisor. Each example is a description of a situation together with a specification—the label—of the correct action the system should take to that situation, which is often to identify a category to which the situation belongs. The object of this kind of learning is for the system to extrapolate, or generalize, its responses so that it acts correctly in situations not present in the training set. This is an important kind of learning, but alone it is not adequate for learning from interaction. In interactive problems it is often impractical to obtain examples of desired behavior that are both correct and representative of all the situations in which the agent has to act. In uncharted territory—where one wouldexpect learning to be most beneficial—an agent must be able to learn from its own experience.

Reinforcement learning is also different from what machine learning researchers call unsupervised learning, which is typically about finding structure hidden in collections of unlabeled data. The terms supervised learning and unsupervised learning appear to exhaustively classify machine learning paradigms, but they do not. Although one might be tempted to think of reinforcement learning as a kind of unsupervised learning because it does not rely on examples of correct behavior, reinforcement learning is trying to maximize a reward signal instead of trying to find hidden structure. Uncovering structure in an agent's experience can certainly be useful in reinforcement learning, but by itself does not address the reinforcement learning agent's problem of maximizing a reward signal. We therefore consider reinforcement learning to be a third machine learning paradigm, alongside of supervised learning, unsupervised learning, and perhaps other paradigms as well.

Reinforcement learning takes the opposite tack, starting with a complete, interactive, goal-seeking agent. All reinforcement learning agents have explicit goals, can sense aspects of their environments, and can choose actions to influence their environments. Moreover, it is usually assumed from the beginning that the agent has to operate despite significant uncertainty about the environment it faces. When reinforcement learning involves planning, it has to address the interplay between planning and real-time action selection, as well as the question of how environment models are acquired and improved. When reinforcement learning involves supervised learning, it does so for specific reasons that determine which capabilities are critical and which are not. For learning research to make progress, important subproblems have to be isolated and studied, but they should be subproblems that play clear roles in complete, interactive, goal-seeking agents, even if all the details of the complete agent cannot yet be filled in.

Examples of Reinforcement learning

A good way to understand reinforcement learning is to consider some of the examples and possible applications that have guided its development.

- A master chess player makes a move. The choice is informed both by planning—anticipating possible replies and counterreplies—and by immediate, intuitive judgments of the desirability of particular positions and moves.
- An adaptive controller adjusts parameters of a petroleum refinery's operation in real time. The controller optimizes the yield/cost/quality trade-off on the basis of specified marginal costs without sticking strictly to the set points originally suggested by engineers.
- A gazelle calf struggles to its feet minutes after being born. Half an hour later it is running at 20 miles per hour.
- A mobile robot decides whether it should enter a new room in search of more trash to collect or start trying to find its way back to its battery recharging station. It makes its decision based on the current charge level of its battery and how quickly and easily it has been able to find the recharger in the past.
- Phil prepares his breakfast. Closely examined, even this apparently mundane activity reveals a complex web of conditional behavior and interlocking goal–subgoal relationships: walking to the cupboard, opening it, selecting a cereal box, then reaching for, grasping, and retrieving the box. Other complex, tuned, interactive sequences of behavior are required to obtain a bowl, spoon, and milk jug. Each step involves a seriesof eye movements to obtain information and to guide reaching and locomotion. Rapid judgments are continually made about how to carry the objects or whether it is better to ferry some of them to the dining table before obtaining others. Each step is guided by goals, such as grasping a spoon or getting to the refrigerator, and is in service of other goals, such as having the spoon to eat with once the cereal is prepared and ultimately obtaining nourishment. Whether he is aware of it or not, Phil is accessing information about the state of his body that determines his nutritional needs, level of hunger, and food preferences.

Elements of Reinforcement Learning

Beyond the agent and the environment, one can identify four main sub elements of a reinforcement learning system: a policy, a reward signal, a value function, and, optionally, a model of the environment.

A policy defines the learning agent's way of behaving at a given time. Roughly speaking, a policy is a mapping from perceived states of the environment to actions to be taken when in those states. It corresponds to what in psychology would be called a set of stimulus–response rules or associations (provided that stimuli include those that can come from within the animal). In some cases the policy may be a simple function or lookup table, whereas in others it may involve extensive computation such as a search process. The policy is the core of a reinforcement learning agent in the sense that it alone is sufficient to determine behavior. In general, policies may be stochastic.

A reward signal defines the goal in a reinforcement learning problem. On each time step, the environment sends to the reinforcement learning agent a single number, a reward. The agent's sole objective is to maximize the total reward it receives over the long run. The reward signal thus defines what are the good and bad events for the agent. In a biological system, we might think of rewards as analogous to the experiences of pleasure or pain. They are the immediate and defining features of the problem faced by the agent. The reward sent to the agent at any time depends on the agent's current action and the current state of the agent's environment. The agent cannot alter the process that does this. The only way the agent can influence the reward signal is through its actions, which can have a direct eff ect on reward, or an indirect eff ect through changing the environment's state. In our example above of Phil eating breakfast, the reinforcement learning agent directing his behavior might receive diff erent reward signals when he eats his breakfast depending on how hungry he is, what mood he is in, and other features of his of his body, which is part of his internal reinforcement learning agent's environment. The reward signal is the primary basis for altering the policy. If an action selected by the policy is followed by low reward, then the policy may be changed to select some other action in that situation in the future. In general, reward signals may be stochastic functions of the state of the environment and the actions taken.

Limitations and Scope

Most of the reinforcement learning methods we consider in this book are structured around estimating value functions, but it is not strictly necessary to do this to solve reinforcement learning problems. For example, methods such as genetic algorithms, genetic programming, simulated annealing, and other optimization methods have been used to approach reinforcement learning problems without ever appealing to value functions. These methods evaluate the "lifetime" behavior of many non-learning agents, each using a different policy for interacting with its environment, and select those that are able to obtain the most reward. We call these evolutionary methods because their operation is analogous to the way biological evolution produces organisms with skilled behavior even when they do not learn during their individual lifetimes. If the space of policies is sufficiently small, or can be structured so that good policies are common or easy to find—or if a lot of time is available for the search—then evolutionary methods can be effective. In addition, evolutionary methods have advantages on problems in which the learning agent cannot accurately sense the state of its environment.

Our focus is on reinforcement learning methods that involve learning while interacting with the environment, which evolutionary methods do not do (unless they evolve learning algorithms, as in some of the approaches that have been studied). It is our belief that methods able to take advantage of the details of individual behavioral interactions can be much more efficient than evolutionary methods in many cases. Evolutionary methods ignore much of the useful structure of the reinforcement learning problem: they do not use the fact that the policy they are searching for is a function from states to actions; they do not notice which states an individual passes through during its lifetime, or which actions it selects. In some cases this information can be misleading (e.g., when states are misperceived), but more often it should enable more efficient search. Although evolution and learning share many features and naturally work together, we do not consider evolutionary methods by themselves to be especially well suited to reinforcement learning problems. For simplicity, in this book when we use the term "reinforcement learning method" we do not include evolutionary methods.

However, we do include some methods that, like evolutionary methods, do not appeal to value functions. These methods search in spaces of policies defined by a collection of numerical parameters. They estimate the directions the parameters should be adjusted in order to most rapidly improve a policy's performance. Unlike evolutionary methods, however, they produce these estimates while the agent is interacting with its environment and so can take advantage of the details of individual behavioral interactions. Methods like this, called policy gradient methods, have proven useful in many problems, and some of the simplest reinforcement learning methods fall into this category. In fact, some of these methods take advantage of value function estimates to improve their gradient estimates. Overall, the distinction between policy gradient methods and other methods we include as reinforcement learning methods is not sharply defined.

An Extended Example: Tic-Tac-Toe

To illustrate the general idea of reinforcement learning and contrast it with other approaches, we next consider a single example in more detail.

Consider the familiar child's game of tic-tac-toe. Two players take turns playing on a three-by-three board. One player plays Xs and the other Os until one player wins by placing three marks in a row, horizontally, vertically, or diagonally, as the X player has in this game:

X	O	O
O	X	X
		X

If the board fills up neitherplayer getting three in a row, the game is a draw. Because a skilled player can play so as never to lose, let us assume that we are playing against an imperfect player, one whose play is sometimes.

Reinforcement Learning Basics

The Reinforcement Learning basics is the always takes action. The Reinforcement learning is about balance between exploitation and exploration. The exploitation refers to making the best use of knowledge acquired so far, while exploring the action performed. The Reinforcement learning through either or reward of penalties. The valve function is the cumulative effect, while reward is associated with a particular atomic action.

The environment need to that can optimize the value. The Reinforcement Learning function is the effect of the environment.

- The gent is the Intelligent Program.
- The Environment is the Maze
- The state is the place in the maze where the agent is
- The action is the move we take move the next state.
- Then reward is the points associated with reaching the state. It can positive,
 Negative or zero

Faces of Reinforcement Learning

The learning agent can adapt to multiple and changing situations and can handle complex problems. It can succeed in a variety of environments. It has a learning element and a performing element. In short, a rational learning agent should possess the following properties:

- It should be able to gather information—continuously or after a certain time interval, that is, periodically.
- It should able to learn from experience.
- It should have an ability to learn continuously.
- It should augment knowledge.
- It should possess autonomy.

The environment is generally dynamic and changing, and in real life the environment is not deterministic. One of the major limiting factors about available environment information is that it is not fully observable or rather it is partially. The intelligence involves the inference about the unknown facts and taking the right actions in a partially known environment. As we can see, intelligence demands flexibility. Flexibility equips the agent to negotiate with dynamic scenarios. To exhibit the required intelligence, we expect certain properties associated with flexibility from an IA. By flexible, we mean that system should be able to adapt with the changing scenarios and should exhibit rational behavior in those changing conditions. For this purpose it needs to be Actions.

1. Responsive: Respond in a timely fashion to the perceived environment. It should be able to perceive changes appropriately and respond to the changes.

2. Proactive: Should exhibit opportunistic, goal-directed behavior and take the initiative where appropriate.

3. Social: Be able to interact (when they deem appropriate with other artificial agents) with
Humans with the order to complete problem solving.

1. Mobility: It is recommended that it should be mobile. It should not get just a static percept.

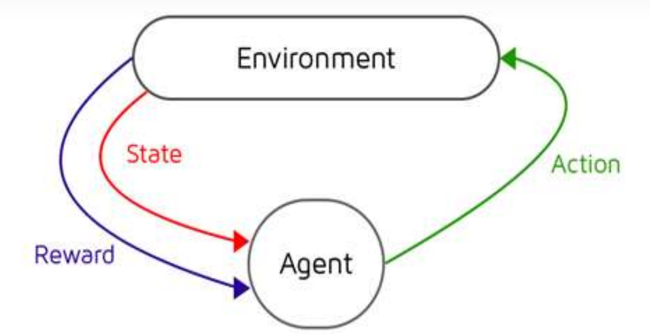
2. Veracity: IA should be truthful. The true correct and rational picture of the environment should be perceived by an agent.

3. Benevolence: Avoid conflict—do what is told.

4. Rationality: It should exhibit rational behavior. It is more like logical behavior.

5. Learning: It should learn from changing scenarios, state transitions, and behavioral changes.

As discussed in the previous chapter, intelligent systems need to have learning capability. Learning with reference to what is already learned, along with learning based on exploration in case of new scenarios, is required. The agent, in order to deal with dynamic scenarios, needs to handle both exploration and exploitation. There is a need for adaptive control and learning abilities. Before discussion about adaptive control, let us discuss about the learning agent.



In machine learning, the environment is formulated as a Markov decision process (MDP), as many reinforcement learning algorithms for this context utilize dynamic programming techniques.

Elements of Reinforcement Learning:

Except for the agent and the environment, we have four sub-elements of reinforcement learning system:

1. Policy: It defines the learning agent's way of behaving at a given time.
2. Reward function: It defines the goal in reinforcement learning problem.
3. Value function: It specifies what is good in the long run.
4. Model of the environment (optional): Models are used for planning, by which we mean any way of deciding on a course of action by considering possible future situations before they are actually experienced.

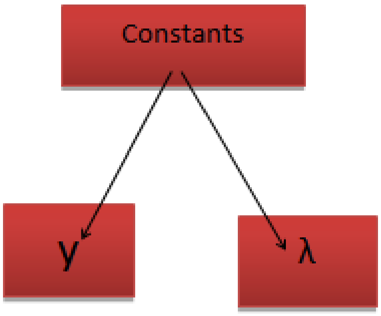
Rewards are in a sense primary, whereas values, as predictions of rewards, are secondary. Without rewards, there could be no values, and the only purpose of estimating values is to achieve more reward.

Positive Reinforcement Learning

Positive Reinforcement learning means getting a positive reward. It is something described that take action. Let me give to an example to understand the concept. Let say the studying the hard you secured the first position in your class. Good action result in a positive reward. Actually try more to continue such good action. This is called Positive Reinforcement Learning

Negative Reinforcement Learning

Negative Reinforcement learning means to getting to the reward or something undesirable given to you take action. For Example you go to a cinema to watch movie and to feel very cold, so it uncomfortable to continue watching the movie. Next time you to the same theater and feel the same gold again. It is surely uncomfortable to watch a movie in the environment. The third time you visit the theater you wear a jacket. With this action, the negative element is again taking ride a cycle example here, that's negative feedback and it's called Negative Reinforcement Learning.**Display values of Constants**



Gamma

Gamma is used in each transition and a constant value each value can be changed . Gamma allows you to information about the type of reward you will be getting every state. The values determine whether we are looking for reward values in each state only.

Lambda

Lambda is generally used when dealing with temporal difference problem. It is more involved with prediction in success states. Increasing values of lambda values it is showing the algorithm of fast. The fast algorithm can be used better results find out the reinforcement learning technique methods.

Difference Between Deep Learning And Reinforcement Learning

Deep learning and reinforcement learning are both systems that learn autonomously. The difference between them is that deep learning is learning from a training set and then applying that learning to a new data set, while reinforcement learning is dynamically learning by adjusting actions based in continuous feedback to maximize a reward.

Deep learning and reinforcement learning aren't mutually exclusive. In fact, you might use deep learning in a reinforcement learning system, which is referred to as deep reinforcement learning and will be a topic I cover in another post.

A good example of using reinforcement learning is a robot learning how to walk. The robot first tries a large step forward and falls. The outcome of a fall with that big step is a data point the reinforcement learning system responds to. Since the feedback was negative, a fall, the system adjusts the action to try a smaller step. The robot is able to move forward. This is an example of reinforcement learning in action.

One of the most fascinating examples of reinforcement learning in action I have seen was when Google's Deep Mind applied the tool to classic Atari computer games such as Break Out. The goal (or reward) was to maximize the score and the actions were to move the bar at the bottom of the screen to bounce the playing ball back up to break the bricks at the top of the screen. You can watch the video here which shows how, in the beginning, the algorithm is making lots of mistakes but quickly improves to a stage where it would beat even the best human players.

II. CONCLUSION:

1. To gain in-depth knowledge of Reinforcement Learning in Artificial Intelligence
2. To explore designs for machines that are effective in solving learning problems of scientific or economic interest.
3. To understand the issues, challenges and discuss several open issues and provide significant improvement of its overall performance.
4. To Propose an adaptive formal framework of Markov decision processes to define the interaction between a learning agent and its environment in terms of states, actions, and rewards.
5. To evaluate the designs through mathematical analysis or computational experiments.

REFERENCES

[1]. GolnazVakili ;SiavashKhorsandi," Self-Organized Cooperation Policy Setting in P2P Systems Based on Reinforcement Learning", IEEE Systems Journal , Volume: 7 , Issue: 1 , March 2013.

[2]. Sanjoy Das ; Sayak Bose ; Siddharth Pal ; Noel N. Schulz ; Caterina M. Scoglio ; BalaNatarajan, "Dynamic reconfiguration of shipboard power systems using reinforcement learning", IEEE Transactions on Power Systems, Volume: 28 , Issue: 2 , May 2013.

[3]. Chunlin Chen ; Daoyi Dong ; Han-Xiong Li ; Jian Chu ; Tzyh-Jong Tarn, "Fidelity-Based Probabilistic Q-Learning for Control of Quantum Systems", IEEE Transactions on Neural Networks and Learning Systems, Volume: 25 , Issue: 5 , May 2014.

[4]. Kei Senda ; Suguru Hattori ; Toru Hishinuma ; TakehisaKohda "Acceleration of Reinforcement Learning by Policy Evaluation Using Nonstationary Iterative Method", IEEE Transactions on Cybernetics, Volume: 44 , Issue: 12 , Dec. 2014.

[5]. GianluigiMongillo ;HananShteingart ; Yonatan Loewenstein, "The Misbehavior of Reinforcement Learning", Proceedings of the IEEE , Volume: 102 , Issue: 4 , April 2014.

[6]. XinChen ; Bo Fu ; Yong He ; Min Wu, "Timesharing-tracking framework for decentralized reinforcement learning in fully cooperative multi-agent system" CAA Journal of AutomaticaSinica, Volume: 1 , Issue: 2 , April 2014.

[7]. Hao Wang ; Shunguo Fan ; Jinhua Song ; Yang Gao ; Xingguo Chen," Reinforcement learning transfer based on subgoal discovery and subtask similarity", CAA Journal of AutomaticaSinica , Volume: 1 , Issue: 3 , July 2014.

[8]. XinXu ;ZhongshengHou ; ChuanqiangLian ; Haibo He "Online Learning Control Using Adaptive Critic Designs With Sparse Kernel Machines" IEEE Transactions on Neural Networks and Learning Systems, Volume: 24 , Issue: 5 , May 2014.

[9]. Bin Xu ;Chenguang Yang ; Zhongke Shi, "Reinforcement Learning Output Feedback NN Control Using Deterministic Learning Technique", IEEE Transactions on Neural Networks and Learning Systems, Volume: 25 , Issue: 3 , March 2014.

[10]. StefanosDoltsinis ; Pedro Ferreira ; NielsLohse, "An MDP Model-Based Reinforcement Learning Approach for Production Station Ramp-Up Optimization: Q-Learning Analysis", IEEE Transactions on Systems, Man, and Cybernetics: Systems , Volume: 44 , Issue: 9 , Sept. 2014.

[11]. Lichao Wang ; KarimLekadir ; Su-Lin Lee ; Robert Merrifield ; Guang-Zhong Yang," A General Framework for Context-Specific Image Segmentation Using Reinforcement Learning", IEEE Transactions on Medical Imaging ,Volume: 32 , Issue: 5 , May 2014.

P.Sushma,etal. "History of Reinforcement Learning". *IOSR Journal of Engineering (IOSRJEN)*, 10(1), 2020, pp. 35-40.