IOSR Journal of Engineering (IOSR JEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 16-18

Study: Deep Reinforcement Learning Overview & Application

Chanda Chouhan[1], Priyanka Sharma[2] , Supriya Mandhare[3] ,
Pragyamani Sharma[4]

[1](Department of Information Technology, University of Mumbai, India)
[2](Department of Information Technology, University of Mumbai, India)
[3](Department of Information Technology, University of Mumbai, India)
[4](Department of Information Technology, University of Mumbai, India)

Abstract: *Reinforcement learning (RL) has made tremendous achievements, e.g., AlphaGo. Here I list (deep) RL applications in the following categories: computer systems, "science, engineeringandarts", finance, businessmanagement, healthcare, education, energy, transportation, autonomous vehicles, games, robotics, computer vision, and natural language processing (NLP). Games, robotics, computer vision, and NLP are intentionally put at the end, since there are too many recent papers in these areas. I list only a few papers for games and robotics, since they are traditional RL application areas, and many people are familiar with them.*

I. Introduction

Reinforcement learning is one of the type of machine learning and also a branch of Artificial intelligence. In this type of machine learning, the machine itself learns how to behave in the environment by performing actions and comparing with the results. It is like machine performing trial and error method to determine the best action possible based on the experience. Reinforcement learning involves goal oriented algorithms, which attain a complex goal with multiple steps which ultimately improves the performance of the machine to predict things. Reinforcement learning can be understood using the concepts of agents, environments, states, actions and rewards, all of which we'll explain below. Capital letters tend to denote sets of things, and lower-case letters denote a specific instance of that thing; e.g. A is all possible actions, while a is a specific action contained in the set. The easiest mental model to aid in understanding Reinforcement Learning (RL) is as a video game, which coincidentally is one of the most popular applications of RL algorithms. [1]

II. Literature Survey

Reinforcement learning is often explained in an example of a video game - it's like playing on high difficulty and dying over and over again until you learn about all obstacles and enemies' weaknesses to successfully finish the level. A reinforcement learning algorithm follows a similar technique - it evaluates current state, takes appropriate action, and then gathers feedback. If it did well - the feedback will be positive (like getting extra points) and if the algorithm failed, it will be punished (like dying or losing lives). Try after try, an agent learns to avoid mistakes to reach the ultimate reward. Here's a visualization of the process:

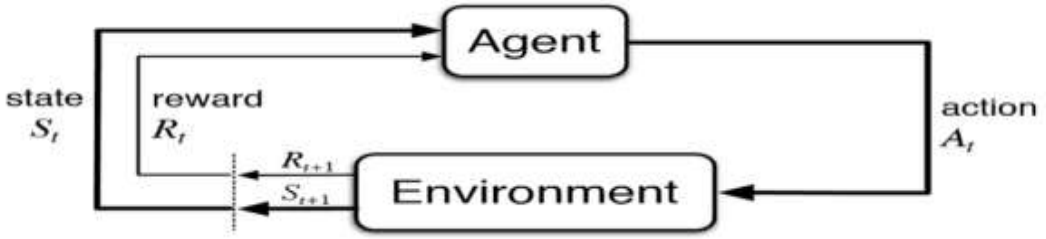

Fig 1: Reinforcement learning with agent and environment

How is RL different from other learning styles? Well, compared to supervised learning, an RL agent doesn't know what actions to take - it only relies on reward or punishment, which means that it often takes a very long time until we receive an outcome. When it comes to unsupervised learning, the main difference is in the goal: while RL focuses on finding the best course of actions, the unsupervised algorithm is looking for connections between data points. Reinforcement learning has a few complexities that slow down its business implementations.

- First, as we already said, sometimes it's unwise to wait until the model finishes training - it may take up to a month and hundreds or thousands of tries.
- Second, it's hard to predict how the model will act in a real environment - not a closed one. When it comes to financial or healthcare use cases, the risk might be too high.
- Third challenge is in posing what should be a reward or a punishment for an agent. It may be as complicated as defining good or bad to a machine.
- And finally, RL simply needs way more data than supervised learning, for instance. If we take self-driving cars, as an example, image how many variables there are on a street - there are simply too many objects acting unpredictably - the difficulty this method raises is often too high for engineers.
  Despite all of this, reinforcement learning is being used in many instances today, successfully so. Here are descriptions of most prominent use cases, if you're interested
- An agent (the player) who moves around doing stuff
- An action that the agent takes (moves upward one space, sells cloak)
- A reward that the agent acquires (coins, killing other players, etc.)
- An environment that the agent exists in (a map, a room)
- A state that the agent currently exists in (on a particular square of a map, part of a room)
- A goal of your agent getting as many rewards as possible

Reinforcement learning is often explained in an example of a video game - it's like playing on high difficulty and dying over and over again until you learn about all obstacles and enemies' weaknesses to successfully finish the level. A reinforcement learning algorithm follows a similar technique - it evaluates current state, takes appropriate action, and then gathers feedback. If it did well - the feedback will be positive (like getting extra points) and if the algorithm failed, it will be punished (like dying or losing lives). [2]

III. Applications

- **Personalization**

News recommendation. Machine learning has made it possible for businesses to personalize customer interactions at scale through the analysis of data on their preferences, background, and online behavior patterns.
However, recommending such content type as online news is still a complex task. News features are dynamic by nature and become rapidly irrelevant. User preferences in topics change as well.
Authors of the research paper DRN: A Deep Reinforcement Learning Framework for News Recommendation discuss three main challenges related to news recommendation methods. First, these methods only try to model current (short-term) reward (e.g., click-through rate that shows the ratios page/ad/email viewers that click on a link). The second issue is that current recommendation methods usually take into account the click/no click labels or ratings as users' feedback. And third, these methods typically continue suggesting similar news to readers, so users can get bored.
The researchers used the Deep Q-Learning based recommendation framework that considers current reward and future reward simultaneously in addition to user return as feedback rather than clicks data.

- **Games personalization.**

Gaming companies also have joined the personalization party. Really, why not tailor a video game experience (rules and content) taking into account an individual player's skill level, playing style, or preferred gameplay? Personalization of game experience is done through player modeling with the goal of increasing their enjoyment. A player model is an abstract description of a player based on their behavior in a game.
Game components that can be adapted include space, mission, character, narrative, music and sound, game mechanics, difficulty scaling, and player matching (in multiplayer games).
RL can be used for optimizing game experience in real-time. In Reinforcement learning for game personalization on edge devices, researchers demonstrate capabilities of this machine learning technique using Pong, last century's arcade game, as the example.

Unity provides an ML toolset for researchers and developers that allows for training intelligent agents with reinforcement learning and "evolutionary methods via a simple Python API."
It's worth mentioning that we haven't found any application of RL agents in production.

- **eCommerce and internet advertising**

Specialists are experimenting with reinforcement learning algorithms to solve a problem of impressions allocation on eCommerce sites like eBay, Taobao, and Amazon. Impressions refer to the number of times a visitor sees some element of a web page, an ad or a product link with a description. Impressions are often used

to calculate how much an advertiser has to pay to show his message on a website. Each time a user loads a page and the ad pops up, it counts as one impression.

These platforms aim to generate maximum total revenue from transactions, that's why they must use algorithms that will allocate buyer impressions (show buyer requests on items) to the most appropriate potential merchants.

Most platforms use such recommendation methods as collaborative filtering or content-based filtering. These algorithms rank merchants using their "historical scores" that rely on the transaction history of the sellers with customers of similar characteristics. Sellers experiment with prices to get higher ranking positions, and these algorithms don't take into account changes in pricing schemes.

As a solution to this problem, researchers applied a general framework of reinforcement mechanism design. The framework uses deep reinforcement learning to develop efficient algorithms that evaluate sellers' behavior.

Online merchants can also conduct fraudulent transactions to improve their rating on eCommerce platforms to draw more buyers. And that, according to researchers, decreases the efficiency of use of buyer impressions and threatens the business environment. However, it's possible to improve the platform's impression allocation mechanism while increasing its profit and minimizing fraudulent activities with reinforcement learning.

In the article on AI and DS advances and trends, we discussed another RL use case – real-time bidding strategy optimization. It allows businesses to dynamically allocate the advertisement campaign budget "across all the available impressions on the basis of both the immediate and future rewards." During real-time bidding, an advertiser bids on an impression, and their ad is displayed on a publisher's platform if they win an auction. [3]

- **Trading in financial industry**

Financial institutions use AI-driven systems to automate trading tasks. Generally, these systems use supervised learning to forecast stock prices. What they can't do is to decide what action to take in a specific situation: to buy, sell, or hold. Traders still must make business rules that are trend-following, pattern-based, or counter-trend to govern system choices. Because analysts may define patterns and confirmation conditions in different ways, there is a need for consistency.

IV. Conclusion

Another way of guiding our agent is not by explicitly telling her what to do at each point, but by giving her **a framework** to make her own decisions. Unlike policy learning, Q-Learning takes *two inputs* – state *and* action – and returns a value for each pair. If you're at an intersection, Q-learning will tell you the expected value of each action your agent could take (left, right, etc.). One of the quirks of Q-Learning is that it doesn't just estimate the *immediate* value of taking an action in a given state: it also adds in all of the potential future value that could be had if you take the specified action. For readers familiar with corporate finance, Q-Learning is sort of like a discounted cash flow analysis – it takes all potential future value into account when determining the current value of an action (or asset). In fact, Q-Learning even uses a *discount-factor* to model the fact that rewards in the future are worth less than rewards now.

References

[1]. https://skymind.ai/wiki/deep-reinforcement-learning#define
[2]. https://blog.algorithmia.com/introduction-to-reinforcement-learning/
[3]. https://mc.ai/reinforcement-learning-applications/