Understanding the Reinforcement Learning

IOP Conf. Series: Journal of Physics: Conf. Series **1207** (2019) 012014 doi:10.1088/1742-6596/1207/1/012014

Understanding the Reinforcement Learning

Nuo Xu

Software Institute, Dalian Jiaotong University, Dalian 116052, Liaoning, China

617705366@qq.com

Abstract. Artificial Intelligence has been a hot topic for a long time. It is a cross-discipline combined with many fields. Among them, machine learning plays the most important role. There are four core subjects in machine learning, supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. In this paper, we are going to talk about the reinforcement learning in the perspective of Markov Decision Process and Partially Observable Markov Decision Process, which are the core algorithms in reinforcement learning. Also, an example of Hearthstone is illustrated to show how to apply reinforcement learning in games for better understanding.

1. Introduction

In recent years, Artificial Intelligence(AI) is becoming one of the most popular topics all over the world. There are more and more researches focusing on AI. With the effort of them, many practical applications have been improved. Along with the development of information technology and computing capability, new approaches and algorithms has been applied, such as machine learning and artificial neural networks. Generally, machine learning algorithms can be divided into four categories: 1) supervised learning, 2) unsupervised learning, 3) semi-supervised learning, and 4) reinforcement learning. This paper is going to talk about one of them, reinforcement learning, especially deep reinforcement learning (DRL), where "deep" refers to more complicated computing layers. The goal of DRL is to get the different optimal strategies according to different environment. When the agent is in a new environment, it can get feedback as input from environment and analysts it and improve the strategy by the result to max the reward. Practically, DRL have been widely applied in the game control [1], robotic control [2], etc. DRL is always used in decision making for better performance of the system.

This paper is going to focus on the utilization of DRL in gaming industry. Similar to Go game, *HearthStone* is also a round-based competing game issued by Blizzard Entertainment Inc., with less logical challenges. Within the game rules and playing strategy, the final goal of *HearthStone* is to beat your opponent. Thus, the total process could be seen as Markov Decision Processes (MDP) which allows players offering potential optimal solution of playing strategy with DRL.

This paper aims to provide an overall discussion of MDP in DRL, including main differences between RL and other ML algorithms, and core concepts of MDP. Combining with the round-based strategy game, *HearthStone*, this paper will also give an example to the algorithm for better understanding.

2. Background

Game playing has dominated the Artificial Intelligence (AI) world as a hot topic ever since the field was born, associating with many related works from scholars. In 1980s, one application with reinforcement-learning, spectacularly far ahead of its time, was Samuel's checkers playing system [3]. Later, Tesauro applied the temporal difference algorithm to *Backgammon* [4, 5]. Tesauro used a back-propagation-based three-layer neutral work as a function algorithm for the value function instead of making a table-based reinforcement learning since *Backgammon* has approximately 10^20 states, which is too many for table-based reinforcement learning. More recently, Littman proposed a reinforcement-learning algorithm which can be adapted to work for a very general of games [6].

There are two main versions of the game learning algorithm were used in the past. the first one we called Basic TD Gammon, which used very little predefined knowledge of the game, and presentation of a broad position was virtually a raw encoding, sufficiently powerful only to permit the neural network to distinguish between conceptually different positions. The second, TD-Gammon, was provided with the same raw state information but supplemented by a number of hand-crafted features of board positions [5].

Also, research team applied reinforcement learning to seven Atari 2600 games from the Arcade Learning Environment, with no adjustment of the architecture or learning algorithm, and found that it outperformed all previous approaches on six of the games and surpassed a human expert on three of them, which was a huge success.

3. Reinforcement learning

As one of the categories of machine learning, reinforcement learning aims to provide a potential optimal strategy in one or more than one schemas. The schema or environment will make the impact on the system and the system will give a feedback to change its status to move forward, like human-like learning process.

3.1 Differences between reinforcement learning and the others

Generally, supervised learning (SL) requires training examples. Training examples are usually in the form of (X_i, Y_i), where each input X_i is N- dimensional vector and each output Y_i is a scalar or a value [7]. In SL, all the inputs need to be labelled. Labels are obtained by training samples and then they are used to classify test dataset, based on the probability distribution $D(x)$, in which way the output values Y_i are assigned to them. The SL's goal is to correctly predict the output values of new data points x drawn from the same distribution $D(x)$.

Semi-supervised learning and unsupervised learning are more likely using clustering method to create labels. Theses labels are not "learnt" but "grouped" by clustering. However, the core thinking of them is similar to supervised learning.

Reinforcement learning is based on supervised learning and semi-supervised, however, in different formats. There are two main differences between reinforcement learning and the others. Firstly, the RL's goal is to get the maximum reward instead of predicting the output data Y for the input data X. Secondly, the learner's task is to choose the values of X, and there is no fixed distribution $D(x)$ from each point X [7]. The main process of reinforcement learning is shown in the Figure 1.

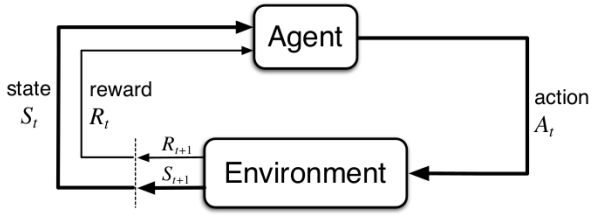

Figure 1. Reinforcement Learning [8]

3.2 Markov Decision Process

Markov Decision Process (MDP) is the basis of reinforcement learning, which provides a logical process to predict or plan future movement when facing uncertainty. According to the definition, MDP can be concluded into a function $M\ (S, D, A, \{P_{sa}(\cdot), \gamma, R\})$, where

- S refers to the set of states;
- D refers to the initial state distribution (probability distribution of S);
- A refers to the set of actions at different time point;
- $P_{sa}(\cdot)$ refers to the state transition distribution, where a belongs to A and s belongs to S;
- γ refers to the reduction parameter, which belongs to $[0, 1]$;
- $R : S \to \mathbb{R}$ refers to the reward function.

Thus, we have

$$R(s, a) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots \quad (1)$$

If the rewards are made based on strategy $\pi : S \to A$, we will have the value function $V^\pi : S \to \mathbb{R}$

$$V^\pi(s) = E_\pi[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots | s_0 = \mathrm{s}] \quad (2)$$

When adding action for each time point, we have the Q-function $Q^\pi : S \times A \to \mathbb{R}$

$$Q^\pi(s, a) = E_\pi[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots | s_0 = \mathrm{s}, a_0 = \mathrm{a}, \forall \mathrm{t} > 0, a_t = \pi(s_t)] \quad (3)$$

Thus, we will have the optimal value function $V^* : S \to \mathbb{R}$ and optimal Q-function $Q^* : S \times A \to \mathbb{R}$

$$V^*(s) = max_\pi V^\pi(s) \quad (4)$$

$$Q^*(s, a) = max_\pi Q^\pi(s, a) \quad (5)$$

According to the Bellman Equation, we have

$$V^*(s) = max_a R(s, a) + \gamma \sum_{s' \in S} P_{sa}(s') V^*(s') \quad (6)$$

$$V^\pi(s) = R\big(s, \pi(s)\big) + \gamma \sum_{s' \in S} P_{s\pi}(s') V^\pi(s') \quad (7)$$

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s' \in S} P_{sa}(s') max_{a' \in A} Q^*(s', a') \quad (8)$$

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} P_{sa}(s') Q^\pi(s', \pi(s')) \quad (9)$$

3.3 Partially Observable Markov Decision Process

No matte single MDP or multi-MDP, it requires the system know s_t for each time t. However, in the practical problems, s_t is hard to be clear somehow. Thus, Partially Observable Markov Decision Process (POMDP) is introduced.

In POMDP, there is an observation set O. At each time point, there is only one observation $o_t = o(s_t)$, where $o : S \to O$ is the observation function. Then the system will give the $O(o|s)$, which refers to the probability based on the observation o and current state s. According to the $O(o|s)$, the system will make the decision with the highest probability to the optimal one.

The probabilities at each state can be seen as a set of belief state. And the strategy of decision making is called belief state tracking. Thus, POMDP problems will be converted into MDP problems, if observation function is clear and $O(o|s)$ is made.

4. Applying reinforcement learning in *Hearthstone*

Hearthstone is a world-famous round-based strategy car-game. Each player is allowed to put card into fields according to the rules. And the final goal of game is to beat the opponent to win. The rule is stated but the cards are random. Each card has its own feature and function to make the player stronger and unbeatable. Thus, the playing strategy of *Hearthstone* can be seen as a POMDP problem.

4.1 Elements in the game

Figure 2 is the game interface of *Hearthstone.* It shows the fundamental elements in the games, which could be parameters in the reinforcement learning algorithms. More details of these elements are listed in the Table 1.

Figure 2. Game interface of *Hearthstone*

Table 1. Core elements in *Hearthstone*

Number	Name	Description
1	Player's mana crystal	Player's mana crystal indicates the cost you may use when using cards. Mana crystal will increase by one in each turn, up to ten.
2	Opponent's mana crystal	Opponent's mana crystal indicates the cost your opponent may use when using cards. Mana crystal will increase by one in each turn, up to ten.
3	Turn's indicator	Play can only take actions in their round.
4	Opponent's health power	You will win when the opponent's health power is lower than 0.
5	Your health power	You will lose when your health power is lower than 0.
6	Weapon	Players with weapon can make damages. The number in box refers to attacking times.
7	Opponent's hand-card	It is the remnant card of your opponent after last round action.
8	Opponent's hero power	Different hero power has different effect.
9	Your hero power	Different hero power has different effect.
10	Card cost	The number of mana crystal needs to be cost to summon a card.
11	Minion's attack	The number shows the damage minion may cause.
12	Minion's health power	Minion will die if health power is lower than 0.

4.2 Data acquisition

Generally, data or parameters listed above can be acquired through two main ways, dataflow and graphic information.

4.2.1 Dataflow. In the Internet communication, dataflow is used to exchange data and information among hosts or servers. Each terminal receives data from the networks and processes them locally. In the *Hearthstone*, heroes' and cards' information is transmitted through the Internet in the same way. Thus, the reinforcement learning system could intercept and capture the data from dataflow. The advantages of this method are 1) all the data in the critical structure which is easy for training; and 2) there is no redundant information which guarantees ideal learning speed. While, the limitation of this method is the risk of security. Data intercept and capture may cause potential security flaw and also the firewall will make influences.

4.2.2 *Graphic information*. Graphic information means the data displayed through visual interfaces. For most of video games, graphic interfaces are essential. These interfaces carry the function of displaying and interacting. Thus, learning form graphic information means a wider application scenario. Furthermore, the primary idea of machine learning and artificial intelligence is to let computer think as human beings. Human uses graphic information think and learn. The computer should do in the same way. In 2013, DeepMind [9] showed how to teach computers learn to play video games via graphic data. They used screenshots of game playing to train the system.

The core technic barrier of using graphic information is image identification. Up to now, researchers have made great effort in this field. From 1960s to 2010s, artificial neural networks, especially convolutional neural networks (CNNs) have been developed and improved to fit practical issues [10], including image identification, pattern recognition, semantic recognition, etc. CNNs use hierarchical structure (layers) to provide main functions of feature extracting, dimension-reduced processing, and category classifying. Another potential algorithm for this issue is YOLO [11], which has faster computing speed and lower accuracy rate than CNNs. The whole detection pipeline of YOLO is a single network, and it can be optimized end-to-end directly on detection performance [11]. In this case, the format and location of each feature are settled. Thus, YOLO could perform better, especially in real-time recognition.

On the other hand, *Hearthstone* is a round-based game, which is suitable for capturing screenshot at each round. It will lower the difficulty of computing.

4.3 Learning process

The logic of learning process is quite simple. The gaming strategy π is based on the game rules, which is settled. Then, other two parameters need to be under consideration as well. First one is the reward factor. In detail, it refers to the number that how many rewards should be given if the player makes damages or is damaged. The second is the number of learning round. The amount of learning time should be controlled, via controlling the number of learning round. In specific number of learning round, the total amount of rewards reflects the effectiveness and efficiency of learning process. On the other hand, the relationship between learning time and learning accuracy is trade-off.

4.4 Potential challenge

The whole learning process is based on Partially Observable Markov Decision Process, which means the observation and belief state are vital. Optimal action is made according to the current state with probability. Thus, strictly there is no optimal solution in true sense but better solution, which will entangle the learning process and add noises. On the other hand, the randomly draw card will affect the convergence. For example, when the agent use a card A, and the A is the optimal choice according to the previous experience. However, in the next turn, your opponent maybe draw a very powerful card and finished the game immediately. Then, the agent will think this choice is not the optimal choice, and this case effect the optimal choice.

5. Conclusion

Reinforcement learning has always been one of the most popular topics in machine learning. This paper has discussed the main idea of reinforcement learning and related algorithms. Also, the example of *Hearthstone* shows how to apply the reinforcement leaning in gaming. Associated with previous researches, reinforcement learning has a strong application prospect in decision making under

uncertain environment. At the same time, the limitation of this methods is also exposed. Markov Decision Process is a powerful mathematic tool in most of decision making issues. While, in reality, the application scenario of MDP is rigorous. Thus, Partially Observable Markov Decision Process is introduced to provide reliable solutions via observation and belief state tracking. However, probability-based POMDP is unstable and cause extra training time. So, the computing speed and more effective algorithm will be next research hotspots. We believe, unsupervised learning method like reinforcement learning will be more powerful in the next generation of computing.

References

[1] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529.
[2] Smart W D, Kaelbling L P. Effective reinforcement learning for mobile robots[C]//Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. IEEE, 2002, 4: 3404-3410.
[3] Samuel A L. Some studies in machine learning using the game of checkers[M]// Computer Games I. Springer New York, 1988:206-226.
[4] Tesauro G. Practical issues in temporal difference learning[J]. Machine Learning, 1992, 8(3-4):257-277.
[5] Tesauro G. Temporal difference learning and TD-Gammon[J]. Communications of the Acm, 1995, 38(3):58-68.
[6] Littman M L. Markov games as a framework for multi-agent reinforcement learning[J]. Machine Learning Proceedings, 1994:157-163.
[7] Si J, Barto A, Powell W, et al. Reinforcement Learning and Its Relationship to Supervised Learning[J]. 2004.
[8] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT press, 1998.
[9] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013.
[10] Koushik J. Understanding Convolutional Neural Networks[J]. 2016.
[11] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// Computer Vision and Pattern Recognition. IEEE, 2016:779-788.