ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 7, July 2018

Reinforcement Learning

Mahesh Joshi[1], Lav Sharma[2], Arvind Pawar[3], Omkar Ghadge[4], Prof. Vijaya Chavan[5]

Student, Computer Technology, Bharati Vidyapeeth's Institute of Technology, Kharghar, India[1,2,3,4]

Professor, Computer Technology, Bharati Vidyapeeth's Institute of Technology, Kharghar, India[5]

Abstract: Reinforcement learning is used to automatically determine the ideal behaviour of a machine with the help of machine learning algorithms to maximize the performance of the machine. Explicit goals are not given to the algorithm. They have to learn this optimal goal by trial and error .think of game contra in which the movement of the player which is done by clicking the buttons that decide the result of the optimal gameplay, by pressing the button the error occurs and the reward are given accordingly. A formula is used to determine the reward of the machine and according to it, the rewards are given from which the machine learns. The rewards can be positive or they can be negative and on the basis of rewards, the machine's accuracy is denoted.

Keywords: Reinforcement learning, RL, base of AI, machine language, machine accuracy

I. INTRODUCTION

Reinforcement learning is the base of artificial intelligence from which the machine learns and is inspired by psychology and it tells how the software agents take actions in an environment so as to maximize the rewards and collect it. The problem is studied in many different disciplines such as game theory, information theory, and many others algorithm. In reinforcement learning, the machine gets positive or negative rewards. In this, the machine should take positive and negative rewards. Through which the machine learns and the machine is trained accordingly. The machine should also take negative rewards and it should know what he is doing wrong. The rewards are generated through a formula. The main difference is that the classical dynamic programming methods and reinforcement learning program and algorithm are that the occurring does not assume the exact knowledge of the mathematical model of the MDP and their target is large MDPs where the exact methods are not capable of being carried out. Reinforcement learning is different from standard learning as they learn from positive and negative rewards pairs that do not show, and some sub-optimal actions are not externally corrected. Instead of sub-optimal actions, it focuses on the performance that involves finding the balance between not- territory and of current knowledge. This exploration is a trade-off that has been mostly studied and is then solved from multi-armed problems and in finite MDPS.

II. LITERATURE REVIEW

Inverse reinforcement learning (IRL) is the general problem of recovering a reward function[1] from demonstrations provided by an expert. By incorporating Gaussian process (GP) into IRL, they present an approach to recovering both rewards and uncertainty information in continuous state and action spaces.

They propose Simple Reinforcement Learning[2] for a reinforcement learning agent that has small memory. In the real world, learning is difficult because there are an infinite number of states and actions that need a large number of stored memories and learning times. To solve a problem, estimated values are categorized as "GOOD" or "NO GOOD" in the reinforcement learning process.

It presents a framework for coordinating multiple intelligent agents[3] within a single virtual environment. Coordination is accomplished via a "next available agent" scheme while learning is achieved through the use of the Q-learning and Sarsa temporal difference reinforcement learning algorithms. To assess the effectiveness of each learning algorithm, experiments were conducted that measured an agent's ability to learn tasks in a static and dynamic environment while using both a fixed (FEP) and variable (VEP) ϵ-greedy probability rate.

It provides an approach to build and adapt a tutoring model by using both artificial neural networks[4] and reinforcement learning. The underlying idea is that tutoring rules can be, firstly, learned by observing human tutors' behavior and, then, adapted, at run-time, by observing how each learner reacts within a learning environment at different states of the learning process.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 7, July 2018

III. METHODOLOGY

1. Agent

Agent is machine that learns from the close interaction with the environment that it senses the state in which it is in and takes the action and which causes the environment to change.

There are basically three components of Agent in Reinforcement Learning:

i) Policy

ii) Value Function

iii) Model

i) Policy

Say, you are looking for a short way and you have 2 options at a particular time, left or right. Since, you don't have any idea which way to do, you will assign a random probability to each direction. This is known as Stochastic Policy as for every decision you have the probability given the current state. Over experience your agent will learn and will go for the option that gives higher probability of success.

$\Pi(a|s) = P[A=a \mid S=s]$

So, it is defined as the probability of taking an action (a) on a particular current state (s). 'π' refers to the policy chosen at the time of decision.

On the other hand say if you are taking the decision by doing proper procedure, you would be knowing that which direction to choose at any point of time. So, it is basically a function that tracks the state to actions. This is known as Deterministic Policy.

ii) Value Function

Value Function defines how good it is to be in a particular state.

Considering the previous example. This time you have a rough idea about the two ways i.e. you now know the obstacles that will come in each way and the estimate time required in each path. So, you to calculate which path will for practical for you getting both heavy rewards and obstacles. This is known as Value Function

iii) Model

Model is the agent's representation of the environment.

We again consider the example of two roads (left and right). You are standing of the crossing. You can see a little bit of the path ahead in each road. For example, the road to the left does not have proper lighting and also, the roads are not quite good. While the other road has proper lighting and the roads are also concrete road. From that point of view you will predict the road ahead and accordingly you will decide which path to go for. So, for our agent it will predict the dynamic environment with the data he has and accordingly it will choose the path. It is known as Transition model.

2. Environment

Environment is nothing but the surrounding in which the agent moves or perform its task i.e. it is the physical world in which the agent operates. Here the agent performs its task and gain reward. The environment is the place where we can find out whether our agent is getting positive reward of negative reward. The environment consists of number of obstacles designed for the agent to perform its task. Environment plays an important role in reinforcement learning. Without environment the reinforcement learning is incomplete.

3. Reward

The feedback sends by the environment to determine the last action. In reinforcement learning the rewards are received on the basis of the agent's behaviour in the environment and the state that the agent is in. If the agent is in the state in which he should not be then it will receive a negative reward but if the agent is the state in which it should be then it will receive a positive reward. The amount of positive or negative reward a agent receives according to it the accuracy of the agent is decided. The reward is decided on the basis of the value of gamma. While the agent is in testing process gamma's value is given 0, if the agent is in perfect state the value of gamma given is 1. Even if the value we are not sure about the agent's action the gamma is kept between 0 and 1. The positive reward indicates that the agent is learning properly and the task given to agents is being done properly. If the agent has more negative reward then the positive

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

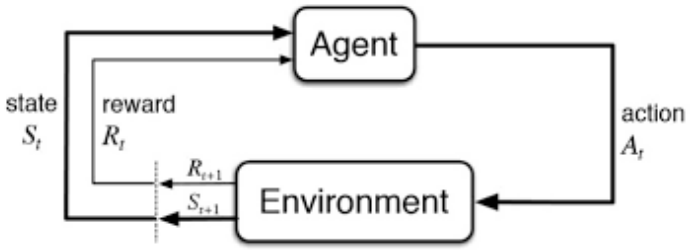
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 7, July 2018

reward then the agent is not working properly. So to make it work properly we have to make some changes in the agent's program.

4. State
State is something that the machine it is in. For example, a car is halted at the signal. Currently the signal is red so the car will stop. Stop is the current state of the car which it is in and when the signal turns green the state is changed and the car should move now. The change of state should be done which will result in some action taken by the car.

5. Action
Action is taken when state changes. Considering the above example, when the signal turns red the state is stop and car stops. The action taken by the car is to stop and when the signal turns green the car should move and the action taken is to move. When the car moves it will receive a positive reward +1 and if doesn't it will receive a negative reward -1. This is how an agent learns from the action.

Working of reinforcement learning with the help of diagram:



$V_\pi(s) = E_\pi[R_t + yR_{t+1} + y^2R_{t+2} + \ldots \mid S_t = s]$

Here, 's' is our current state. 'π' is the policy that we use to define our behaviour. In the situation above 'π' is the behaviour of being unwilling, i.e., we got to the current state by being unwilling to do a work. 'R' is the reward that we will get for travelling to a state. 'y' is the discount factor which helps us weight the impact of future rewards. So, basically the value of being in a state is the sum of the present reward we are getting for being in that state and the future rewards that we will get if we take different actions on a particular state. We are not thinking about the future award as the future rewards as we cannot predict the future rewards. Since, it is stochastic model we are not sure if we will actually get those rewards or not.

IV. CONCLUSION

In this we have studied and gathered the information about reinforcement learning i.e. how reinforcement learning works and what are the element of reinforcement learning which is also known as elements of machine learning. As we saw each element has its own task and it helps us to know whether the specific task is being performed properly or not. The behaviour of AGENT, ENVIRONMENT, REWARD, STATE, ACTION, help us to know whether the machine has learn successfully or not.

REFERENCES

[1] Zhuo-Jun Jin, Hui Qian and Miao-Liang Zhu "Gaussian processes in inverse reinforcement learning" https://ieeexplore.ieee.org/document/5581063/
[2] Akira Notsu, Katsuhiro Honda and Hidetomo Ichihashi "Simple Reinforcement Learning for Small-Memory Agent".https://ieeexplore.ieee.org/document/6147019/
[3] William Sause." Coordinated Reinforcement Learning Agents in a Multi-agent Virtual Environment" https://ieeexplore.ieee.org/document/6784616/
[4] Giuseppe Fenza, Francesco Orciuoli and Demetrios G. Sampson "Building Adaptive Tutoring Model Using Artificial Neural Networks and Reinforcement Learning" https://ieeexplore.ieee.org/document/8001832/
[5] Yue-Sheng He and Yuan-Yan Tang "Path planning of virtual human by using reinforcement learning" https://ieeexplore.ieee.org/document/4620548/
[6] Toshiaki Takano, Haruhiko Takase and Hiroharu Kawanaka"Transfer Method for Reinforcement Learning in Same Transition Model -- Quick Approach and Preferential Exploration" https://ieeexplore.ieee.org/document/6147021/
[7] Barış Gökçe and H. Levent Akın "Implementation of Reinforcement Learning by transfering sub-goal policies in robot navigation" https://ieeexplore.ieee.org/document/6531546/
[8] Chun-Gui Li, Meng Wang and Qing-Neng Yuan "A Multi-agent Reinforcement Learning using Actor-Critic methods"https://ieeexplore.ieee.org/document/4620528/
[9] Bo Wu and Yanpeng Feng"Monte-Carlo Bayesian Reinforcement Learning Using a Compact Factored Representation" https://ieeexplore.ieee.org/document/8110331/
[10] Wang Qiang and Zhan Zhongli "Reinforcement learning model, algorithms and its application" https://ieeexplore.ieee.org/document/6025669/
[11] Bing-Qiang Huang,Guang-Yi Cao and Min Guo "Reinforcement Learning Neural Network to the Problem of Autonomous Mobile Robot Obstacle Avoidance" https://ieeexplore.ieee.org/document/1526924/
[12] Ilya Kachalsky, Ilya Zakirzyanov and Vladimir Ulyantsev"Applying Reinforcement Learning and Supervised Learning Techniques to Play Hearthstone" https://ieeexplore.ieee.org/document/8260800/