A

Seminar Report

on

“Reinforcement Learning and Deep Reinforcement Learning with applications”

Submitted to the

Savitribai Phule Pune University

In partial fulfillment for the award of the Degree of

Bachelor of Engineering

in

Information Technology

by

Aditya Kangune

(33323)

Under the guidance of

Mrs. Usha. A. Jogalekar

Department Of Information Technology

Pune Institute of Computer Technology College of Engineering
Sr. No 27, Pune-Satara Road, Dhankawadi, Pune - 411 043.

A. Y. 2021-2022

CERTIFICATE

This is to certify that the seminar report entitled **“Reinforcement Learning and Deep Reinforcement Learning with applications”** being submitted by **Aditya Kangune (33323)** is a record of bonafide work carried out by him under the supervision and guidance of **Prof. Mrs. Usha. A. Jogalekar** in partial fulfillment of the requirement for **TE (Information Technology Engineering) – 2019 course** of Savitribai Phule Pune University, Pune in the academic year 2021-22.

Date: 14/11/2021

Place: Pune

Mrs. Usha. A. Jogalekar
Guide

Dr.A.M.Bagade
Head of the Department

Dr. R. Sreemathy
Principal

ACKNOWLEDGEMENT

“As our circle of knowledge expands, so does the circumference of darkness surrounding it.”

The seminar has helped me a lot to discover various new things. I extend my gratitude to Pune Institute of Computer Technology for giving me an opportunity to enhance my knowledge through this seminar.

I am extremely grateful to Dr. R. Shreemathy, Principal, PICT, and Dr. Anant Bagade, Head of the Department (Information Technology), for providing all the required resources for the successful completion of my seminar.

My heartfelt gratitude to my seminar guide Mrs. Usha. A. Jogalekar, Department of Information Technology, for her valuable suggestions and guidance in the preparation of the seminar report.

I specially thank my friend Aagaaz for working on this report together with me and constantly being there for helping me with difficulties.

I express my thanks to all the staff members and friends for all the help and coordination extended in bringing out this seminar successfully in time.

I acknowledge with grateful thanks to the authors of the references and other pieces of literature referred to in this seminar.

Kang

Aditya Kangune
(33323)

ABSTRACT

Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Deep reinforcement learning is the combination of reinforcement learning (RL) and deep learning. DRL has applications in many fields like medicine, robotics, games, etc. RL works on the Markov Decision Process which leads to Q-learning. MDP provides a mechanism to maximize the reward in a given environment. Combining DL and RL leads to the formation of Deep Q-Networks. RL holds great importance in creating AI for games. Few AIs have managed to defeat world champions in a few games like chess. When we dive deeper into RL, we encounter more complex multi-agent environments. The main challenge in multi-agent environments is to understand the interaction between agents. There are certain techniques to deal with such environments including Graph Convolutional Reinforcement Learning. Particular focus is on exploring these concepts and generalizing DRL and exploring its use in more practical applications like personalized recommendations systems, etc.

Keywords: Reinforcement Learning, Deep Reinforcement Learning, Machine Learning, Artificial Intelligence, MDP, Neural Networks.

CONTENTS

LIST OF FIGURES

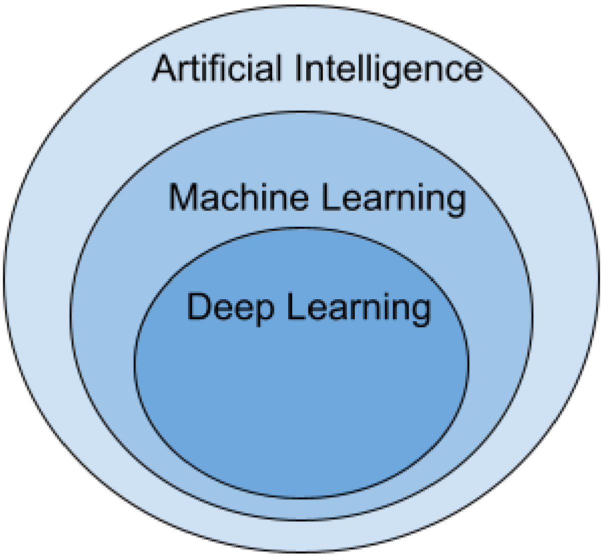
LIST OF TABLES

CHAPTER 1

INTRODUCTION TO SEMINAR TOPIC

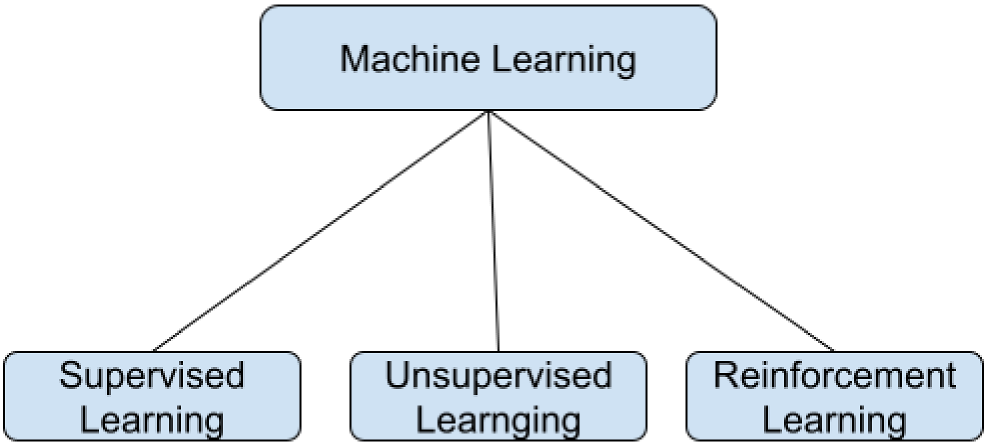
1.1 Introduction to Seminar:

Artificial Intelligence is a concept that has been discovered and tweaked since around the 1940s. Under AI we have Machine Learning where we work with statistics and algorithms. Finally, we have Deep Learning under ML where we deal with deep concepts like image and speech recognition.



Branches of Automation

Briefly put, there are 3 branches in ML- Supervised Learning, Unsupervised Learning, and Reinforcement Learning.
In RL, the model keeps on increasing its performance using Reward Feedback to learn the behavior or pattern. These algorithms are specific to a particular problem.



ML Types

1.2 **Motivation behind seminar topic**:

For new age problems in this decade, we need a solution that can create the perfect model to solve that particular problem. That's what the RL aims for.

RL has various interesting applications in the modern world viz medicine, healthcare, self-driving cars, robotics, gaming, marketing, advertising, etc.

1.3 Aim and Objective(s) of the work

Seminar aim:

The aim of this seminar is to understand the complex subject of RL and DRL and simplify them with studying about applications of the same.

Seminar objectives:

- Studying basic concepts in Reinforcement Learning.
- Learning how to integrate Reinforcement Learning with Deep Learning or neural networks.
- Exploring the applications of single-agent environments.
- Exploring the techniques involved in multi-agent environments.

1.4 Introduction to RL and DRL:

Reinforcement Learning is a subfield of machine learning that teaches an agent how to choose an action from its action space, within a particular environment, in order to maximize rewards over time.

When Deep Learning is integrated with RL it is known as Deep Reinforcement Learning. It is seen in many real-world applications around us like games, recommendation engines for famous e-commerce websites, video-streaming platforms, etc.

This seminar will mostly cover the application of DRL which is used in recommendation systems. Recommendation systems are something that we all come across daily from recommendations on Google maps to OTT services like Netflix, Amazon. Increasing watch time for video streaming platforms or increasing sales for e-commerce websites can be done by self-supervised recommendation systems. Multiple factors like the interactions between users and items are taken into consideration. Recommendation systems can be enhanced by RL by concentrating on the interaction between the agent i.e. the user and the environment i.e. the website. Resulting in maximizing the cumulative reward for the agent based on the interaction.

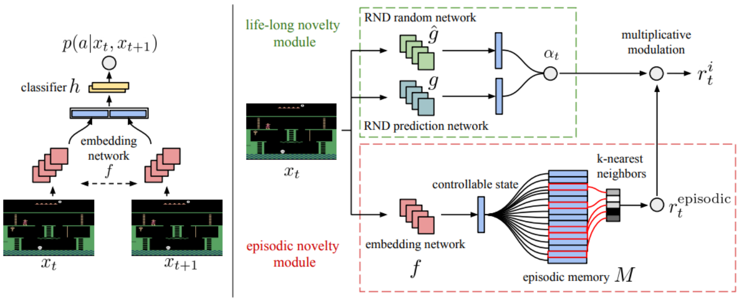
This seminar focuses on how reinforcement learning and deep reinforcement learning work and their use in 2 states of art recommendations frameworks namely Self-Supervised Q-learning (SQN) and Self-Supervised Actor-Critic (SAC).

Objectives of this seminar include gaining a good understanding of MDP and applying this knowledge to understand the 2 recommendations frameworks mentioned above.

CHAPTER 2

LITERATURE SURVEY OF Seminar Title/Topic

One of the best papers according to the ICLR 2020 Conference was "Never Give Up: Learning Directed Exploration Strategies". The authors proposed a reinforcement learning agent to solve hard exploration games by learning a range of directed exploratory policies.



Atari game RL

"End to End Learning for Self-Driving Cars" is a brilliant work where the authors dug deep into the world of self-driving cars and trained a convolutional neural network (CNN) which is used to map raw pixels coming from a single front-facing camera directly to steering commands.

"Genie: A Generator of Natural Language Semantic Parsers for Virtual Assistant Commands" - This paper has the Genie toolkit that can handle new compound commands with variably less manual effort.

"Recommendation systems: Principles, methods and evaluation" - This paper explores potentials of different prediction techniques and the different characteristics and in recommendation systems.

"An Overview of Chatbot Technology" - The authors explain about chatbot classification based on various factors, such as the need they serve, the area of knowledge they refer to, etc. They present the architecture of modern chatbots and also mention the main platforms for their creation.

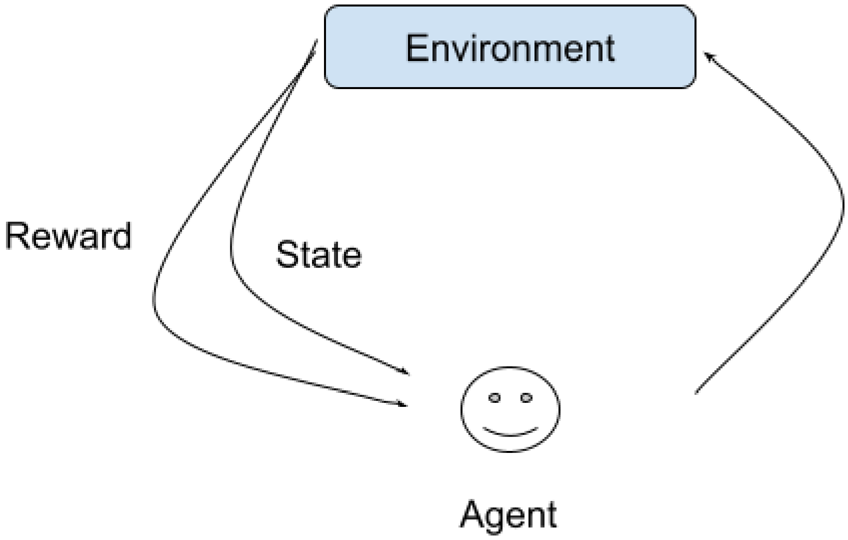
"DynaMIT: a simulation-based system for traffic prediction" - DynaMIT supports both prescriptive and descriptive information. It generates prediction-based guidance with respect to departure time, pre-trip path and mode choice decisions and en-route path choice decisions.

CHAPTER 3

Reinforcement Learning & Deep Reinforcement Learning

Reinforcement Learning is a Machine Learning method that deals with how software agents should take actions in an environment.

Reinforcement Learning is a part of the deep learning method, This helps to maximize some portion of the cumulative reward. This neural network learning method helps to learn how to attain a complex objective or maximize a specific dimension over many steps.



RL Terms

Take the example of a self-driving car, where the car is the agent, and the track that it has to cover is the environment. Positive rewards may include breaking or turning at the right point, stopping at a red light, etc and negative rewards may include crashing with another car/person, etc. So the clear goal for the agent, in this case, would be to reach the destination with maximum reward. The agent is programmed to make the right decisions at the right time. Markov decision processes give us a way to formalize sequential decision-making.

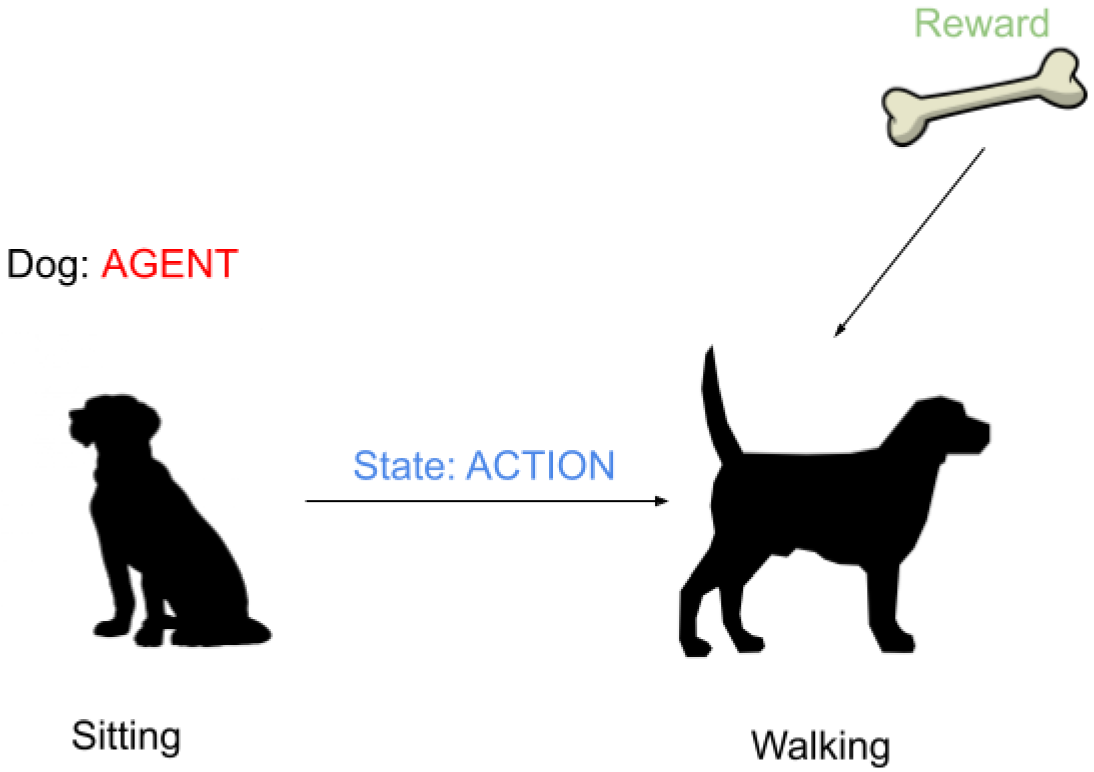
Terminologies:

- **Agent:** It is an assumed entity which performs actions in an environment to gain some reward.
- **State (s):** State refers to the current situation returned by the environment.
- **Policy (π):** It is a strategy which is applied by the agent to decide the next action based on the current state.
- **Environment (e):** A scenario that an agent has to face.
- **Reward (R):** An immediate return given to an agent when he or she performs specific action or task.
- **Value (V):** It is expected long-term return with discount, as compared to the short-term reward.
- **Model of the environment:** This mimics the behavior of the environment. It helps you to make inferences to be made and also determine how the environment will behave.
- **Model based methods:** It is a method for solving reinforcement learning problems which use model-based methods.
- **Value Function:** It specifies the value of a state that is the total amount of reward. It is an agent which should be expected beginning from that state.
- **Q value or action value (Q):** Q value is quite similar to value. The only difference between the two is that it takes an additional parameter as a current action.

Working of Reinforcement Learning:

Consider the scenario of teaching new tricks to a dog.

- As a dog doesn't understand English or any other human language, we can't tell him directly what to do. Instead, we follow a different strategy.
- We emulate a situation, and the dog tries to respond in many different ways. If the dog's response is the desired way, we will give him food.
- Now whenever the dog is exposed to the same situation, the dog executes a similar action with even more enthusiasm in expectation of getting more reward(food).
- That's like learning that a dog gets "what to do" from positive experiences.
- At the same time, the dog also learns what not to do when faced with negative experiences.



RL dog example

Reinforcement Learning Algorithms:

There are three approaches to implement a Reinforcement Learning algorithm.

Value-Based:

In a value-based Reinforcement Learning method, one should try to maximize a value function **V(s)**. In this method, the agent is expecting a long-term return of the current states under policy **π**.

Policy-based:

In a policy-based RL method, one should try to come up with such a policy that the action performed in every state helps you to gain maximum reward in the future.
Two types of policy-based methods are:

- Deterministic: For any state, the same action is produced by the policy π.
- Stochastic: Every action has a certain probability, which is determined by the following equation.Stochastic Policy :

n{a\s) = P\A, = a\S, =S]

Model-Based:

In this Reinforcement Learning method, you need to create a virtual model for each environment.

The agent learns to perform in that specific environment.

Important traits of Reinforcement Learning:

- There is no supervisor, only a real number or reward signal
- Sequential decision making
- Time plays a crucial role in Reinforcement problems
- Feedback is always delayed, not instantaneous
- Agent’s actions determine the subsequent data it receives

Types of Reinforcement Learning

Two kinds of reinforcement learning methods are:

Positive:

It is defined as an event that occurs because of specific behavior. It increases the strength and the frequency of the behavior and impacts positively on the action taken by the agent.

This type of Reinforcement helps you to maximize performance and sustain change for a more extended period. However, too much Reinforcement may lead to over-optimization of state, which can affect the results.

Negative:

Negative Reinforcement is defined as strengthening of behavior that occurs because of a negative condition which should have stopped or avoided. It helps you to define the minimum standard of performance. However, the drawback of this method is that it provides enough to meet up the minimum behavior.

Learning Models of Reinforcement

There are two important learning models in reinforcement learning:

- Markov Decision Process
- Q learning

Markov decision processes

In MDP, there is an agent that interacts with the environment that it is placed in. The agent gets a representation of the current state of the environment sequentially over time. Based on the state, the agent chooses a particular action which leads to a change in the environmental state, and the agent is rewarded as a cause of the action that it took.

There are 5 major components involved in an MaDP:

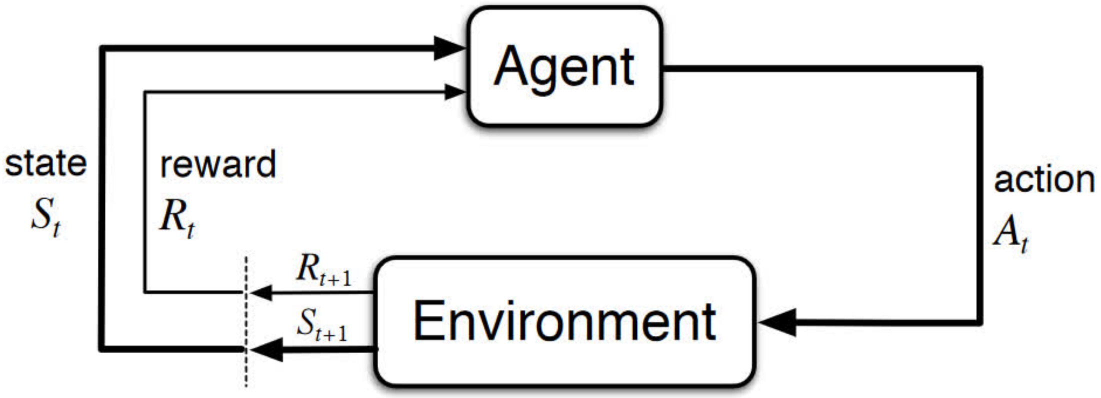
1. Agent
2. Environment
3. State Space (S)
4. Action Space (A)
5. Reward Space (R)

The process of transitioning from one state to another happens sequentially over and over again. Throughout this process, it is the agent's goal to maximize not just the immediate reward, but the cumulative rewards it receives over time

At each time step *t*, the agent receives a representation of the environment's current state S_t ∈ S. Based on this state, the agent selects an action A_t ∈ A. This gives a state-action pair for time *t* as (S_t, A_t).

Time is incremented to the next time step *t+1,* and the environment is transitioned to a new state S_{t+1} ∈ S. At this time, the agent receives a numerical reward R_{t+1} ∈ R for the state-action pair (S_t, A_t).

The following diagram (fig 1) illustrates the above-mentioned algorithm.



Markov decision processes

Expected Return

The goal of the agent is to maximize the total rewards which can be represented as the sum of future rewards till the last time step **T.** This is called the expected return denoted by G. G is given by

$$G = R_{t+1} + a\, R_{t+1} + a^{2} R_{t+1} + \dots$$

$$G = \sum_{k=0}^{\infty} a^{k} R_{t+k+1}$$

Where R_t is the return at time *t* and *a* is a number ranging from 0 to 1.

Policies and Value Functions

A policy is a function that maps a given state to probabilities of selecting each possible action from that state. It is denoted by π. If an agent follows policy π at time t, then π(a|s) is the probability that A_t= a if S_t, = s. An optimal policy is a policy that gives the highest expected reward among all the policies.

Value functions give a measure of how good a particular action is. This is achieved by using an action-value function which is denoted by q_π. The value of action *A* in state *S* under policy π is the expected return from starting from state *S* at time *t*, taking action *A*, and following policy π thereafter is given by

$$q_{\pi}(S,\ A)\ =\ E_{\pi}[\ \sum_{k=0}^{\infty} a^{k}R_{t+k+1}\,|\,S_{t},\ =\ S,\ A_{t}=\ A]$$

An optimal state value function is denoted by q_* and is defined as

$$q_*\ =\ max\,q_{\pi}(S,\ A)$$

q_*must always satisfy an equation called as Bellman Optimality Equation which is given as

$$q_*(S,\ A)\ =\ E[R_{t+1}\ +\ a\ *\ max\,q_*(S',\ A')]$$

S' and A' denote the next state and action respectively. This equation ensures that the agent is following an optimal policy and the next state S' will be the state from which best possible action A' can be taken at time $t + 1$.

Q-Learning

Q Learning keeps a lookup table of values Q(s, a) with one entry for every state-action pair [2]. The Q-learning algorithm makes use of the Bellman equation for the Q-value function.

Deep Q-Learning

The use of a deep neural network to estimate the Q-values for each state-action pair in a given environment is called deep Q-Learning and the resultant network is called a deep Q-Network. For each given state input, the network outputs the Q-values for each action that can be taken from that state. The Bellman equation is used for this purpose.

$$q_*(S,\ A)\ =\ E[R_{t+1}\ +\ a\ *\ max\,q_*(S',\ A')]$$

The loss associated with the neural network is calculated by comparing the output Q-values with the target Q-values. This loss is then used for backpropagation in order to update the weights in the

Reinforcement Learning vs. Supervised Learning:

Parameters	**Reinforcement Learning**	**Supervised Learning**
Decision style	Reinforcement learning helps you to take your decisions sequentially.	In this method, a decision is made on the input given at the beginning.
Works on	Works on interacting with the environment.	Works on examples or given sample data.
Dependency on decision	In RL method learning, decision is dependent. Therefore, you should give labels to all the dependent decisions.	Supervised learning the decisions which are independent of each other, so labels are given for every decision.
Best suited	Supports and works better in AI, where human interaction is prevalent.	It is mostly operated with an interactive software system or applications.
Example	Chess game	Object recognition

CHAPTER IV

APPLICATIONS OF RL and DRL

- OTT platforms like Netflix, Amazon prime, etc.
- Social Media (Eg: Facebook suggestions).
- Gaming.
- Google Maps.
- CNN.
- Virtual Assistants.
- Self-driving cars.
- Robotics for industrial automation.
- Business strategy planning.
- Machine learning and data processing.
- It helps you to create training systems that provide custom instruction and materials according to the requirements of students.
- Aircraft control and robot motion control.

Advantages of RL:

- It helps to find which situation needs action.
- Helps you to discover which action yields the highest reward over the longer period.
- Reinforcement Learning also provides the learning agent with a reward function.
- It also allows it to figure out the best method for obtaining large rewards.

Disadvantages of DRL:

- When there is enough data to solve the problem with a supervised learning method reinforcement learning model can’t be applied.
- Reinforcement Learning is computing-heavy and time-consuming, in particular when the action space is large.
- Feature/reward design which should be very involved.
- Parameters may affect the speed of learning.
- Realistic environments can have partial observability.
- Too much Reinforcement may lead to an overload of states which can diminish the results.
- Realistic environments can be non-stationary.

CHAPTER V

CONCLUSION

- Reinforcement Learning is a Machine Learning method.
- Helps you to discover which action yields the highest reward over the longer period.
- Three methods for reinforcement learning are 1) Value-based 2) Policy-based and Model based learning.
- Agent, State, Reward, Environment, Value function Model of the environment, Model based methods, are some important terms using in RL learning method.
- The example of reinforcement learning is your cat is an agent that is exposed to the environment.
- The biggest characteristic of this method is that there is no supervisor, only a real number or reward signal.
- Two types of reinforcement learning are 1) Positive 2) Negative.
- Two widely used learning models are 1) Markov Decision Process 2) Q learning.
- Reinforcement Learning method works on interacting with the environment, whereas the supervised learning method works on given sample data or example.
- Application or reinforcement learning methods are: Robotics for industrial automation and business strategy planning.
- This method should not be used when there is enough data to solve the problem.
- The biggest challenge of this method is that parameters may affect the speed of learning.

REFERENCES

List all the material used from various sources for making this seminar.

[1] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, Karol Zieba – "**End to End Learning for Self-Driving Cars**".

[2] Giovanni Campagna, Silei Xu, Mehrad Moradshahi, Richard Socher, Monica S. Lam - **"Genie: A Generator of Natural Language Semantic Parsers for Virtual Assistant Commands"**

[3] Eleni AdamopoulouEmail authorLefteris Moussiades - "**Recommendation systems: Principles, methods and evaluation",** Egyptian Informatics Journal, Volume 16, Issue 3, November 2015.

[4] Adamopoulou E., Moussiades L. (2020) "**An Overview of Chatbot Technology"**. In: Maglogiannis I., Iliadis L., Pimenidis E. (eds) Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, vol 584. Springer, Cham. https://doi.org/10.1007/978-3-030-49186-4_31

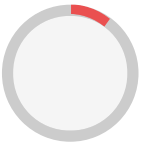
[5] Ben-Akiva, Moshe & Bierlaire, Michel & Koutsopoulos, Haris & Mishalani, Rabi. (2000). **"DynaMIT: a simulation-based system for traffic prediction"**.

[6] Chris and Mandy. "Reinforcement Learning - Goal Oriented Intelligence". ***deeplearning.ai***, Sep 2018

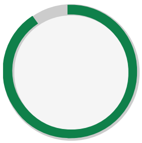
PLAGIARISM REPORT

Dupli Checker

PLAGIARISM SCAN REPORT



10%
Plagiarised



90%
Unique

Date	2021-11-13
Words	993
Characters	6412

Content Checked For Plagiarism

CHAPTER 1

INTRODUCTION TO SEMINAR TOPIC

Introduction to Seminar:

Artificial Intelligence is a concept that has been discovered and tweaked since around the 1940s. Under AI we have Machine Learning where we work with statistics and algorithms. Finally, we have Deep Learning under ML where we deal with deep concepts like image and speech recognition.
Branches of Automation

Introduction to RL and DRL:

Reinforcement Learning is a subfield of machine learning that teaches an agent how to choose an action from its action space, within a particular environment, in order to maximize rewards
over time.
When Deep Learning is integrated with RL it is known as Deep Reinforcement Learning. It is seen in many real-world applications around us like games, recommendation engines for famous e-commerce websites, video-streaming platforms, etc.
This seminar will mostly cover the application of DRL which is used in recommendation systems. Recommendation systems are something that we all come across daily from recommendations on Google maps to OTT services like Netflix, Amazon. Increasing watch time for video streaming platforms or increasing sales for e-commerce websites can be done by self-supervised recommendation systems. Multiple factors like the interactions between users and items are taken into consideration. Recommendation systems can be enhanced by RL by concentrating on the interaction between the agent i.e. the user and the environment i.e. the website. Resulting in maximizing the cumulative reward for the agent based on the interaction.
This seminar focuses on how reinforcement learning and deep reinforcement learning work and their use in 2 states of art recommendations frameworks namely Self-Supervised Q-learning (SQN) and Self-Supervised Actor-Critic (SAC).
Objectives of this seminar include gaining a good understanding of MDP and applying this knowledge to understand the 2 recommendations frameworks mentioned above.

CHAPTER 3

Reinforcement Learning & Deep Reinforcement Learning

Reinforcement Learning is a Machine Learning method that deals with how software agents should take actions in an environment.

Reinforcement Learning is a part of the deep learning method, This helps to maximize some portion of the cumulative reward. This neural network learning method helps to learn how to attain a complex objective or maximize a specific dimension over many steps.

Markov decision processes

In MDP, there is an agent that interacts with the environment that it is placed in. The agent gets a representation of the current state of the environment sequentially over time. Based on the state, the agent chooses a particular action which leads to a change in the environmental state, and the agent is rewarded as a cause of the action that it took.

There are 5 major components involved in an MaDP:
Agent
Environment
State Space (S)
Action Space (A)
Reward Space (R)

The process of transitioning from one state to another happens sequentially over and over again. Throughout this process, it is the agent's goal to maximize not just the immediate reward, but the cumulative rewards it receives over time

At each time step t, the agent receives a representation of the environment’s current state St ∊ S. Based on this state, the agent selects an action At ∊ A. This gives a state-action pair for time t as (St, At).

Time is incremented to the next time step t+1, and the environment is transitioned to a new state St+1 ∊ S. At this time, the agent receives a numerical reward Rt+1 ∊ R for the state-action pair (St, At).
The following diagram (fig 1) illustrates the above-mentioned algorithm.

Markov decision processes

Expected Return

The goal of the agent is to maximize the total rewards which can be represented as the sum of future rewards till the last time step T. This is called the expected return denoted by G. G is given by

G = Rt+1 + aRt+1 + a2 Rt+1 + ...
G = k = 0∞akRt+k+1

Where Rt is the return at time t and a is a number ranging from 0 to 1.

Policies and Value Functions

A policy is a function that maps a given state to probabilities of selecting each possible action from that state. It is denoted by ***π***. If an agent follows policy ***π***
at time t, then ***π***(a|s) is the probability that At= a if St, = s. An optimal policy is a policy that gives the highest expected reward among all the policies.

Value functions give a measure of how good a particular action is. This is achieved by using an action-value function which is denoted by qπ. The value of action A in state S under policy ***π*** is the expected return from starting from state S at time t, taking action A, and following policy ***π*** thereafter is given by

qπ(S, A) = Eπ[k = 0∞akRt+k+1 | St, = S, At= A]

An optimal state value function is denoted by q* and is defined as

q* = max qπ(S, A)

q*must always satisfy an equation called as Bellman Optimality Equation which is given as

q*(S, A) = E[Rt+1 + a*max q*(S', A')]

S' and A' denote the next state and action respectively. This equation ensures that the agent is following an optimal policy and the next state S' will be the state from which best possible action A' can be taken at time t + 1.

Working of Reinforcement Learning:
Consider the scenario of teaching new tricks to a dog.
As a dog doesn’t understand English or any other human language, we can’t tell him directly what to do. Instead, we follow a different strategy.

We emulate a situation, and the dog tries to respond in many different ways. If the dog’s response is the desired way, we will give him food.

Now whenever the dog is exposed to the same situation, the dog executes a similar action with even more enthusiasm in expectation of getting more reward(food).

That’s like learning that a dog gets “what to do” from positive experiences.

At the same time, the dog also learns what not to do when faced with negative experiences.

Matched Source

Similarity 25%

Title:Reinforcement Learning, Part 1: A Brief Introduction - Medium

Oct 16, 2019 — Welcome back to my blog for engineers who want to learn AI! In my last post, I talked about how to level up your machine learning career ...

https://medium.com/ai%C2%B3-theory-practice-business/reinforcement-learning-part-1-a-brief-introduction-a53a849771cf

Similarity 4%

Title:Reinforcement Learning: What is, Algorithms, Applications ...

Oct 6, 2021 — Reinforcement Learning is defined as a Machine Learning method that is concerned with how software agents should take actions in an environment.

https://www.guru99.com/reinforcement-learning-tutorial.html

Similarity 3%

Title:Reinforcement Learning - Policies and Value Functions

Sep 27, 2018 — A policy is a function that maps a given state to probabilities of selecting each possible action from that state. We will use the symbol π ...

https://deeplizard.com/learn/video/eMxOGwbdqKY

Similarity 3%

Title:Introduction to Machine Learning | Types of Machine ...

· We emulate a situation, and the dog tries to respond in many different ways. If the dog’s response is in the desired way, we will give it a treat. If the dog’s response is in ...

https://harshal-pawar.medium.com/introduction-to-machine-learning-516322d64628