



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Deep Reinforcement Learning

2022-23 Second Semester, M.Tech (AIML)

Session #1: Introduction to the Course

Instructors :

1. Prof. S. P. Vimal (vimalsp@wilp.bits-pilani.ac.in),
2. Dr. V Chandra Sekhar (chandrasekhar.v@wilp.bits-pilani.ac.in)



What is Reinforcement Learning ?

- reward based learning / feedback based learning
- not a type of NN nor it is an alternative to NN. Rather it is an approach for learning
- Autonomous driving, gaming

Why Reinforcement Learning ?

- a goal-oriented learning based on interaction with environment



Course Objectives

Course Objectives:

1. Understand
 - a. the conceptual, mathematical foundations of deep reinforcement learning
 - b. various classic & state of the art Deep Reinforcement Learning algorithms
2. Implement and Evaluate the deep reinforcement learning solutions to various problems like planning, control and decision making in various domains
3. Provide conceptual, mathematical and practical exposure on DRL
 - a. to understand the recent developments in deep reinforcement learning and
 - b. to enable modelling new problems as DRL problems.



Learning Outcomes

1. understand the fundamental concepts of reinforcement learning (RL), algorithms and apply them for solving problems including control, decision-making, and planning.
2. Implement DRL algorithms, handle challenges in training due to stability and convergence
3. evaluate the performance of DRL algorithms, including metrics such as sample efficiency, robustness and generalization.
4. understand the challenges and opportunities of applying DRL to real-world problems & model real life problems



Course Operation

- **Instructors**

Prof. S.P.Vimal

Dr. V Chandra Sekhar

- **Textbooks**

1. Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, Second Ed. , MIT Press
2. Foundations of Deep Reinforcement Learning: Theory and Practice in Python (Addison-Wesley Data & Analytics Series) 1st Edition by Laura Graesser and Wah Loon Keng



Course Operation

- **Evaluation**

Two Quizzes for 5% each; Best of two will be taken for 5% (in final grading);

Whatever be the points set for quizzes, the score will be scaled to 5%

NO MAKEUP, for whatever be the reason. Ensure to attend at least one of the quizzes.

Two Assignments - Tensorflow/ Pytorch / OpenAI Gym Toolkit → 25 %

Assignment 1: Partially Numerical + Implementation of Classic Algorithms - 10%

Assignment 2: Deep Learning based RL - 15%

Mid-Term Exam - 30% [Only to be written in A4 pages, scanned and uploaded]

Comprehensive Exam - 40% [Only to be written in A4 pages, scanned and uploaded]

- **Webinars/Tutorials**

4 tutorials : 2 before mid-sem & 2 after mid-sem

- Teaching Assistants will be introduced to you in the next class



Course Operation

- Schedule of Schedule of Quizzes

Sunday, January 14, 2024	7:00:00 PM	Monday, January 15, 2024	7:00:00 PM
Sunday, March 10, 2024	7:00:00 PM	Monday, March 11, 2024	7:00:00 PM

- Schedule of Assignments

Assignment - #1: 02 Jan, 2024 7:00 PM 18 Jan, 2024 11:59 PM

Assignment - #2: 01, Mar 2024 7:00 PM 15, Mar 2024 11:59 PM

- Schedule of Webinars

21-Dec-23; 10-Jan-24; 22-Feb-24; 13-Mar-24



Course Operation

- How to reach us ? (for any question on lab aspects, availability of slides on portal, quiz availability , assignment operations)

1.Prof. S. P. Vimal (vimalsp@wilp.bits-pilani.ac.in),
2.Dr. V Chandra Sekhar (chandrasekhar.v@wilp.bits-pilani.ac.in)

- **Plagiarism [Important]**

All submissions for graded components must be the result of your original effort. It is strictly prohibited to copy and paste verbatim from any sources, whether online or from your peers. The use of unauthorized sources or materials, as well as collusion or unauthorized collaboration to gain an unfair advantage, is also strictly prohibited. Please note that we will not distinguish between the person sharing their resources and the one receiving them for plagiarism, and the consequences will apply to both parties equally.

In cases where suspicious circumstances arise, such as identical verbatim answers or a significant overlap of unreasonable similarities in a set of submissions, will be investigated, and severe punishments will be imposed on all those found guilty of plagiarism.



Reinforcement Learning

Reinforcement learning (RL) is based on rewarding desired behaviors or punishing undesired ones. Instead of one input producing one output, the algorithm produces a variety of outputs and is trained to select the right one based on certain variables – Gartner




When to use RL?

RL can be used in large environments in the following situations:

1. A model of the environment is known, but an analytic solution is not available;
2. Only a simulation model of the environment is given (the subject of simulation-based optimization)
3. The only way to collect information about the environment is to interact with it.



(Deep) Reinforcement Learning

<u>Paradigm</u>	 Supervised Learning	 Unsupervised Learning	 Reinforcement Learning
<u>Objective</u>	$p_{\theta}(y x)$	$p_{\theta}(x)$	$\pi_{\theta}(a s)$
<u>Applications</u>	→ Classification → Regression	→ Inference → Generation	→ Prediction → Control



Types of Learning

Criteria	Supervised ML	Unsupervised ML	Reinforcement ML
<i>Definition</i>	Learns by using labelled data	Trained using unlabelled data without any guidance.	Works on interacting with the environment
<i>Type of data</i>	Labelled data	Unlabelled data	No – predefined data
<i>Type of problems</i>	Regression and classification	Association and Clustering	Exploitation or Exploration
<i>Supervision</i>	Extra supervision	No supervision	No supervision
<i>Algorithms</i>	Linear Regression, Logistic Regression, SVM, KNN etc.	K – Means, C – Means, Apriori	Q – Learning, SARSA
<i>Aim</i>	Calculate outcomes	Discover underlying patterns	Learn a series of action
<i>Application</i>	Risk Evaluation, Forecast Sales	Recommendation System, Anomaly Detection	Self Driving Cars, Gaming, Healthcare

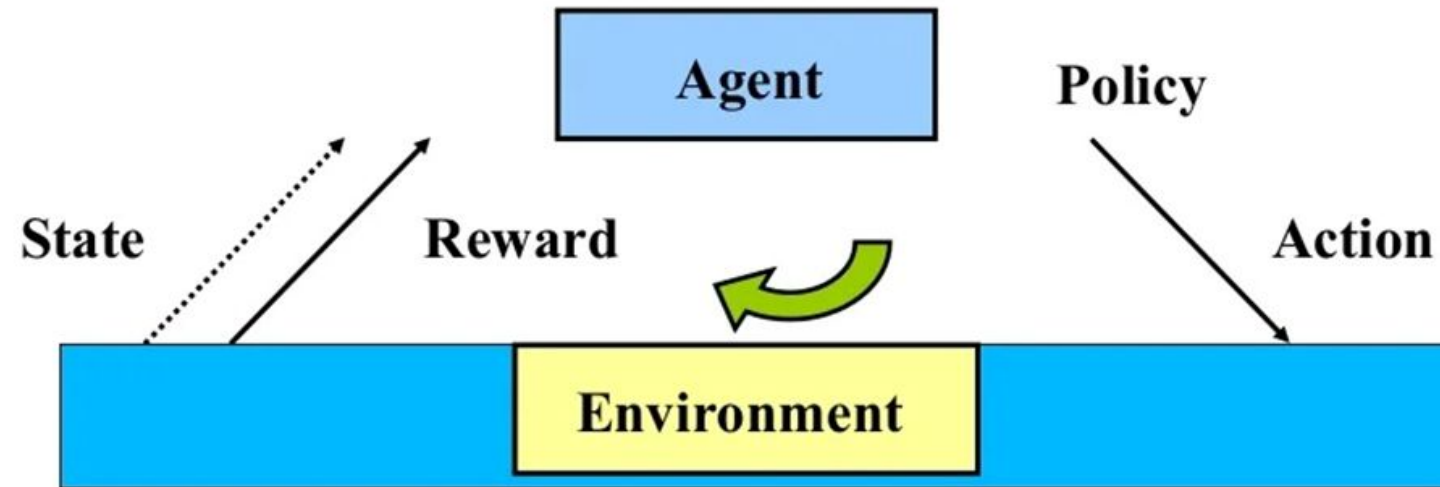


Characteristics of RL

- No supervision, only a real value or reward signal
- Decision making is sequential
- Time plays a major role in reinforcement problems
- Feedback isn't prompt but delayed

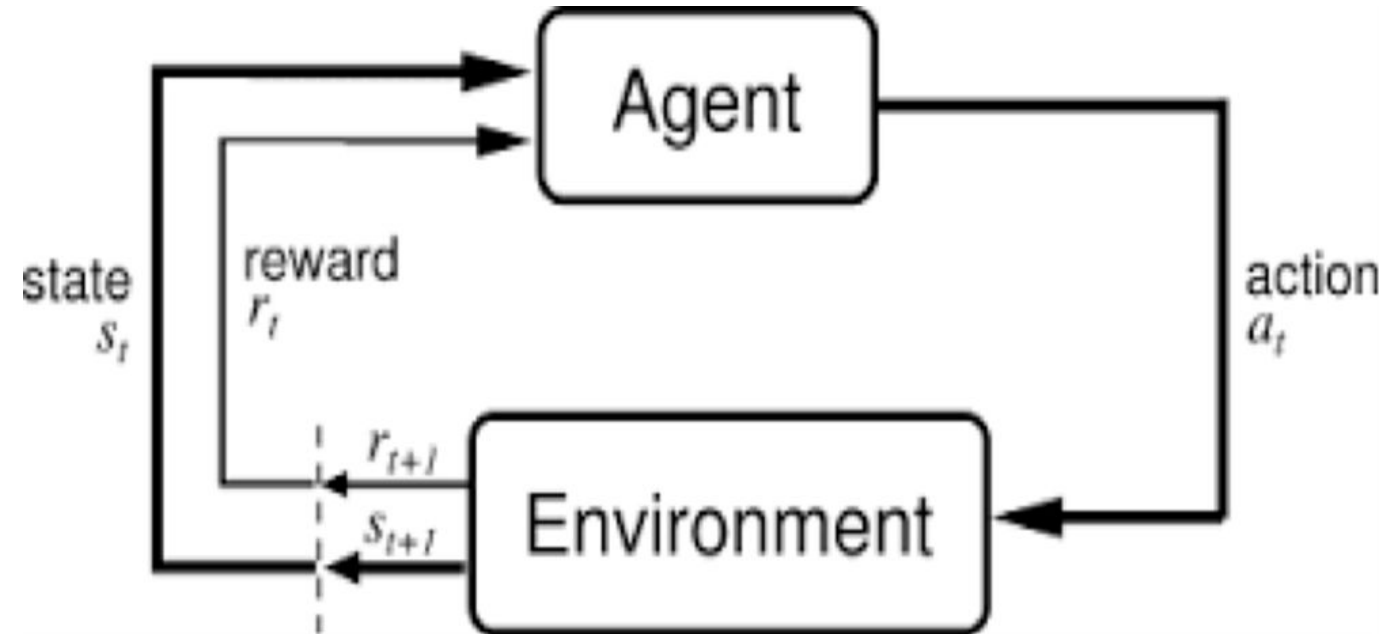


Elements of Reinforcement Learning





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*, *a value function*, and, optionally, *a model* of the environment.



Elements of Reinforcement Learning

•Agent

- an **entity** that tries to learn the best way to perform a specific task.
- In our example, the child is the agent who learns to ride a bicycle.

•Action (A) -

- **what the agent does** at each time step.
- In the example of a child learning to walk, the action would be “walking”.
- A is the set of all possible moves.
- In video games, the list might include running right or left, jumping high or low, crouching or standing still.



Elements of Reinforcement Learning

•State (S)

- **current situation** of the agent.
- After doing performing an action, the agent can move to different states.
- In the example of a child learning to walk, the child can take the action of taking a step and move to the next state (position).

•Rewards (R)

- feedback that is given to the agent based on the action of the agent.
- If the action of the agent is good and can lead to winning or a positive side then a positive reward is given and vice versa.



Elements of Reinforcement Learning

- **Environment**

- outside world of an agent or physical world in which the agent operates.

- **Discount factor**

- The **discount factor** is multiplied by future rewards as discovered by the agent in order to dampen these rewards' effect on the agent's choice of action. Why? It is designed to make future rewards worth less than immediate rewards.

Often expressed with the lower-case Greek letter gamma: γ . If γ is .8, and there's a reward of 10 points after 3 time steps, the present value of that reward is $0.8^3 \times 10$.



Elements of Reinforcement Learning

Formal Definition - ***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**.*

End of Session #1



Elements of Reinforcement Learning

- **Goal of RL** - maximize the total amount of rewards or cumulative rewards that are received by taking actions in given states.

- Notations –

- a set of states as \mathcal{S} ,

- a set of actions as \mathcal{A} ,

- a set of rewards as \mathcal{R} .

At each time step $t = 0, 1, 2, \dots$, some representation of the environment's state $\mathbf{S}_t \in \mathcal{S}$ is received by the agent. According to this state, the agent selects an action $\mathbf{A}_t \in \mathcal{A}$ which gives us the state-action pair $(\mathbf{S}_t, \mathbf{A}_t)$. In the next time step $t+1$, the transition of the environment happens and the new state $\mathbf{S}_{t+1} \in \mathcal{S}$ is achieved. At this time step $t+1$, a reward $\mathbf{R}_{t+1} \in \mathcal{R}$ is received by the agent for the action \mathbf{A}_t taken from state \mathbf{S}_t .



Elements of Reinforcement Learning

- Maximize cumulative rewards, Expected Return G_t

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\ &= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \end{aligned}$$

- Discount factor γ is introduced here which forces the agent to focus on immediate rewards instead of future rewards. The value of γ remains between 0 and 1.



Elements of Reinforcement Learning

- Policy (π)

- Policy in RL decides which action will the agent take in the current state.
- It tells the *probability that an agent will select a specific action from a specific state*.
- Policy is a function that maps a given state to probabilities of selecting each possible action from the given state.

• If at time t , an agent follows policy π , then $\pi(a/s)$ becomes the probability that the action at time step t is $a_{t=a}$ if the state at time step t is $s_{t=s}$. The meaning of this is, the probability that an agent will take an action a in state s is $\pi(a/s)$ at time t with policy π .



Elements of Reinforcement Learning

- **Value Functions**

- a simple measure of how good it is for an agent to be in a given state, or how good it is for the agent to perform a given action in a given state.

- **Two types**

- state- value function
- action-value function



Elements of Reinforcement Learning

•State-value function

- The *state-value function* for policy π denoted as v_{π} determines the *goodness of any given state for an agent who is following policy π* .
- This function gives us the value which is the expected return starting from state s at time step t and following policy π afterward.

$$\begin{aligned} v_{\pi}(s) &= E_{\pi}[G_t \mid S_t = s] \\ &= E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right]. \end{aligned}$$



Elements of Reinforcement Learning

•Action value function

- determines the goodness of the action taken by the agent from a given state for policy π .
- This function gives the value which is the expected return starting from state s at time step t , with action a , and following policy π afterward.
- The output of this function is also called as *Q-value* where q stands for Quality. Note that in the *state-value function*, we did not consider the action taken by the agent.

$$\begin{aligned} q_{\pi}(s, a) &= E_{\pi}[G_t \mid S_t = s, A_t = a] \\ &= E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]. \end{aligned}$$



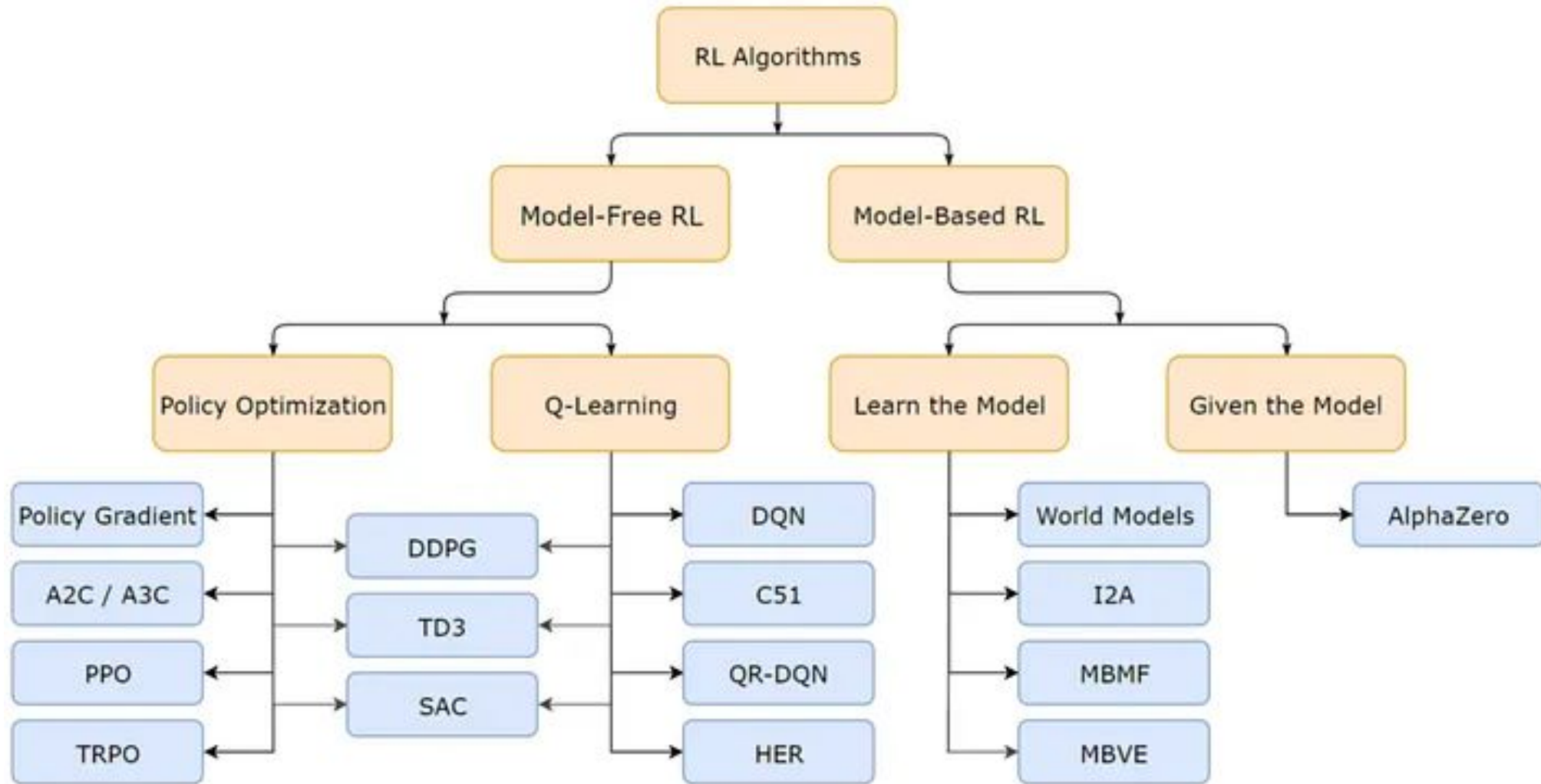
Elements of Reinforcement Learning

- **Model of the environment**

- mimics the behavior of the environment, or more generally, that allows inferences to be made about how the environment will behave.
- For example, given a state and action, the model might predict the resultant next state and next reward.
- Models are used for planning



(Deep) Reinforcement Learning





Advantages of Reinforcement Learning

- solve very complex problems that cannot be solved by conventional techniques
- achieve long-term results
- model can correct the errors that occurred during the training process.
- In the absence of a training dataset, it is bound to learn from its experience
- can be useful when the only way to collect information about the environment is to interact with it
- Reinforcement learning algorithms maintain a ***balance between exploration and exploitation***. Exploration is the process of trying different things to see if they are better than what has been tried before. Exploitation is the process of trying the things that have worked best in the past. Other learning algorithms do not perform this balance



An example scenario - 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

Assumptions

- playing against an imperfect player, one whose play is sometimes incorrect and allows you to win

Aim

How might we construct a player that will find the imperfections in its opponent's play and learn to maximize its chances of winning?



Reinforcement Learning for Tic-Tac-Toe

- We set up a table of numbers, one for each possible state of the game. Each number will be the latest estimate of the probability of our winning from that state.
- We treat this estimate as the state's value, and the whole table is the learned value function.
- State A has higher value than state B, or is considered better than state B, if the current estimate of the probability of our winning from A is higher than it is from B.



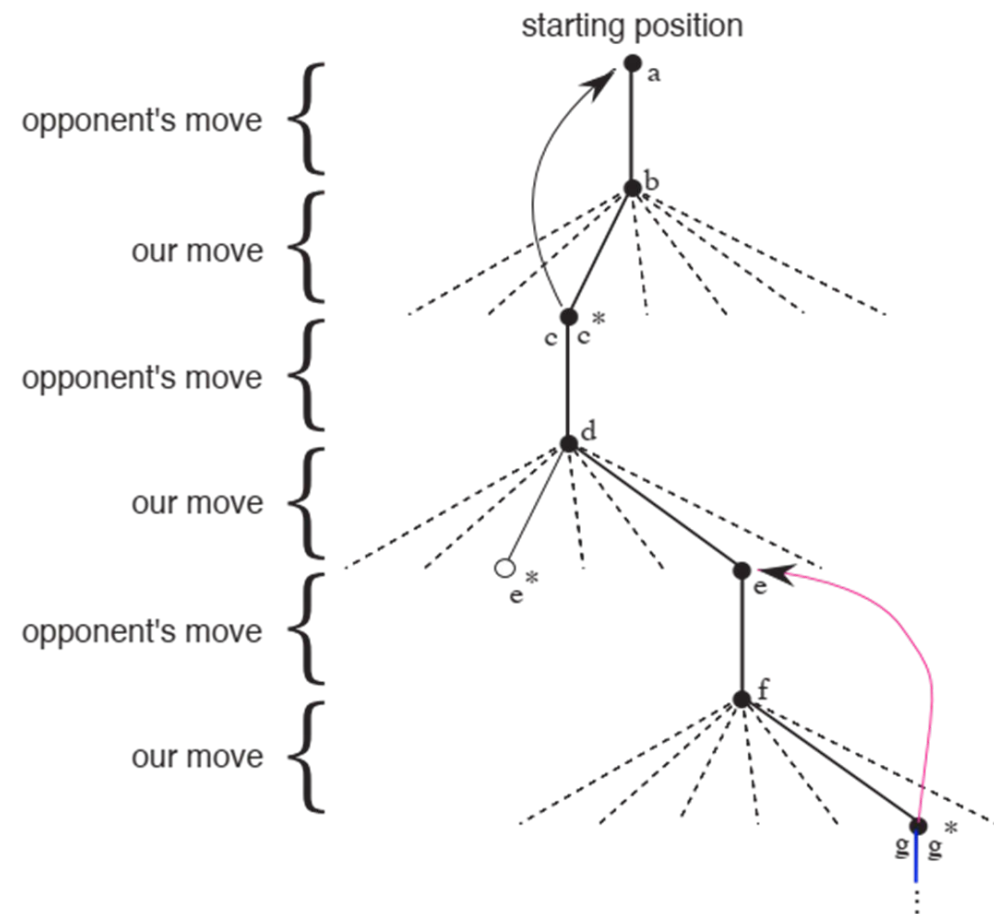
Reinforcement Learning for Tic-Tac-Toe

- Assuming we always play X's, three X = probability is 1, three O = probability = 0 .
Initial = 0.5
- Play many games against the opponent. To select our moves we examine the states that would result from each of our possible moves and look up their current values in the table.
- Most of the time we move greedily, selecting the move that leads to the state with greatest value, i.e, with the highest estimated probability of winning.
- Occasionally, however, we select randomly from among the other moves instead. These are called **exploratory moves** because they cause us to experience states that we might otherwise never see.

Reinforcement Learning for Tic-Tac-Toe

- The solid lines represent the moves taken during a game; the dashed lines represent moves that we (our reinforcement learning player) considered but did not make.

- Our second move was an exploratory move, meaning that it was taken even though another sibling move, the one leading to e^* , was ranked higher.





Reinforcement Learning for Tic-Tac-Toe

- Once a game is started, our agent computes all possible actions it can take in the current state and the new states which would result from each action.
- The values of these states are collected from a ***state_value vector***, which contains values for all possible states in the game.
- The agent can then choose the action which leads to the state with the highest value(exploitation), or chooses a random action(exploration), depending on the value of epsilon.



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Thank you