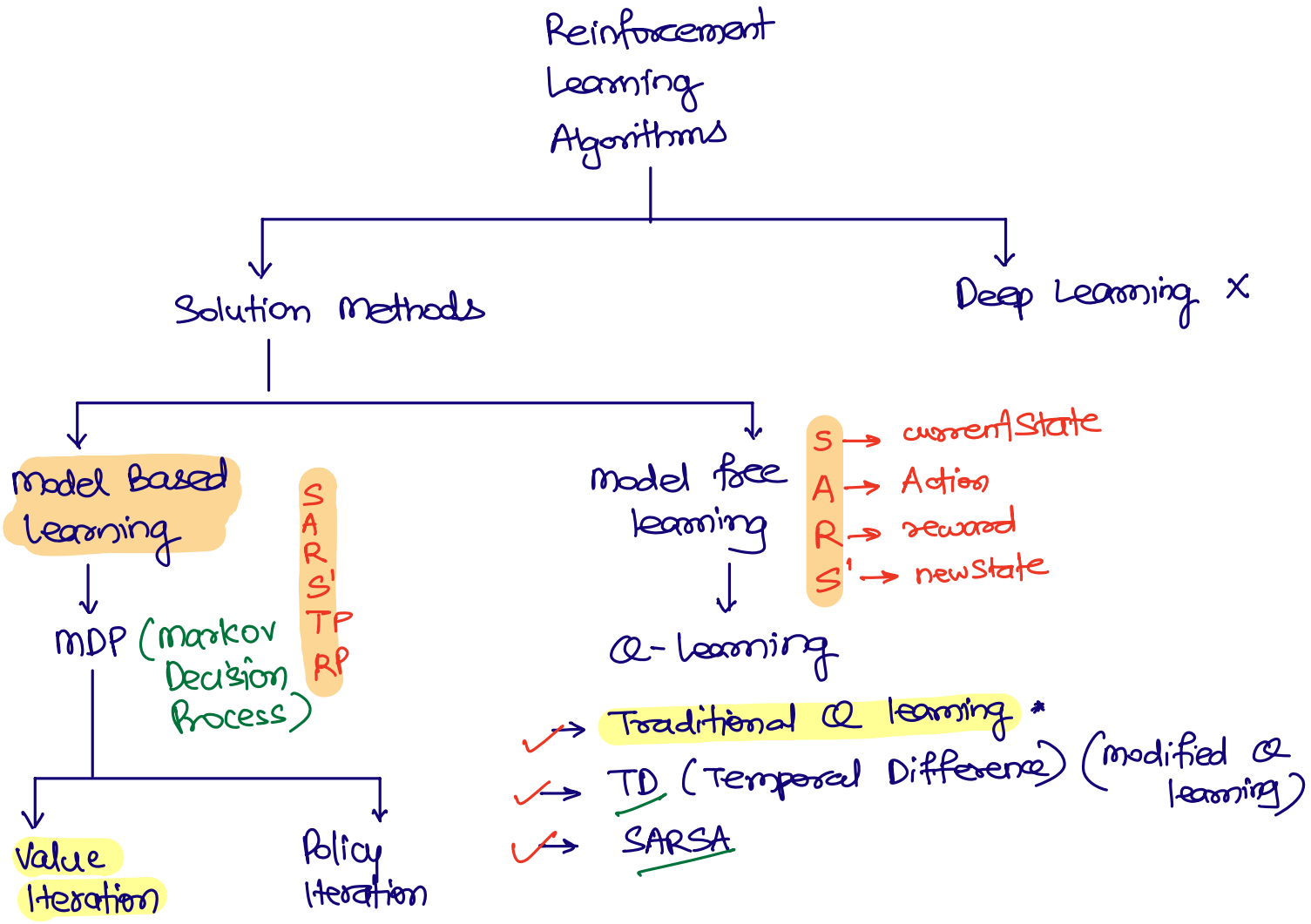
Reinforcement learning
Trainer: Prashant N.

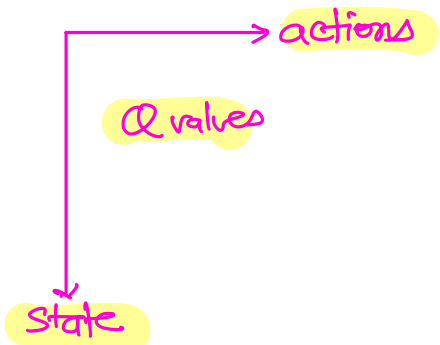


Q-Learning

- model free learning approach.
- S A R S'

current state – S
action – A
reward – R
new state – S'

Q Table



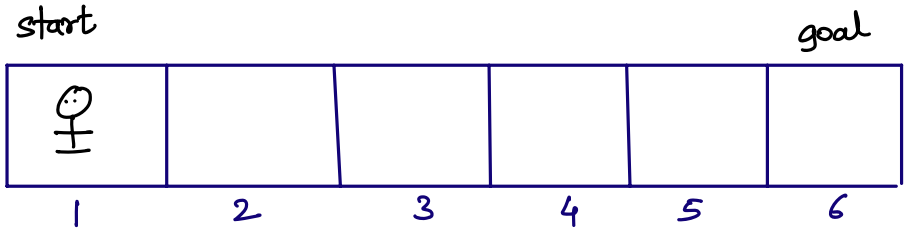
- Goal of Q learning is to train Q Table
- There exists two HYPERPARAMETERS
  - a) gamma (DISCOUNT FACTOR) ..... Applied on all Q learning methods
  - b) alpha (LEARNING RATE) ..... Applied on TD & SARSA

0 - 1

Use-case

Given env

Goal: make agent reach the goal



Valid states → 1, 2, 3, 4, 5, 6

Valid Action → Left, Right

state \ Action	Left	Right
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0

Q values

Traditional Q learning *

$$Q(S,A) = IR + (gamma * max(S', a^*))$$

- $Q(S,A)$: current state, current action
- IR: immediate reward
- gamma: discount factor (hyperparameter)
- $max(S', a^*)$: next state, all valid actions

Q value of next state for all possible action and extract the max q value.

Modified Q learning *

$$Q(S,A) = Q(S,A) + \alpha [R(S,A) + \gamma \max Q'(S', a^*) - Q(S,A)]$$

- $Q(S,A)$: current Q value from Q table
- α: learning rate
- $R(S,A)$: IR
- γ: discount factor
- $\max Q'(S', a^*)$: max q value of next state from all possible actions
- $Q(S,A)$: current Q value from Q table.

Algorithm to train the Q-Table

① initialize all Q-values in the Q-Table as 0.

② Observe the current state 's'.

③ Do forever

exploration or epsilon greedy.

a. Select an action 'a' and execute it.

b. Receive an IR

c. Observe new state s'.

d. Update the q-table entry for the given (S,a) using valid formula*

until S is the terminal state | max episode count reaches