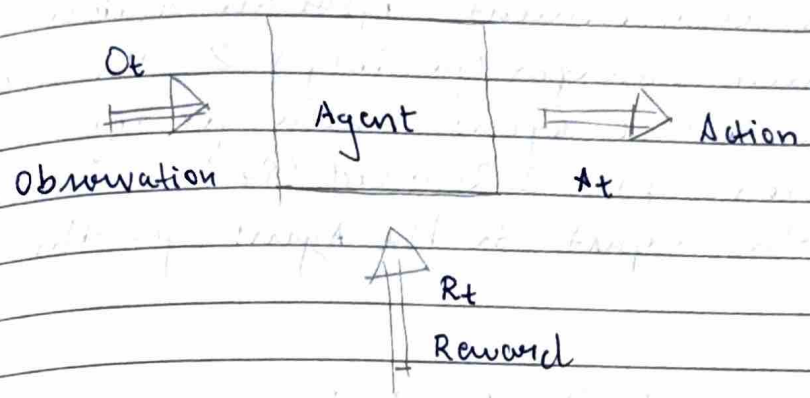
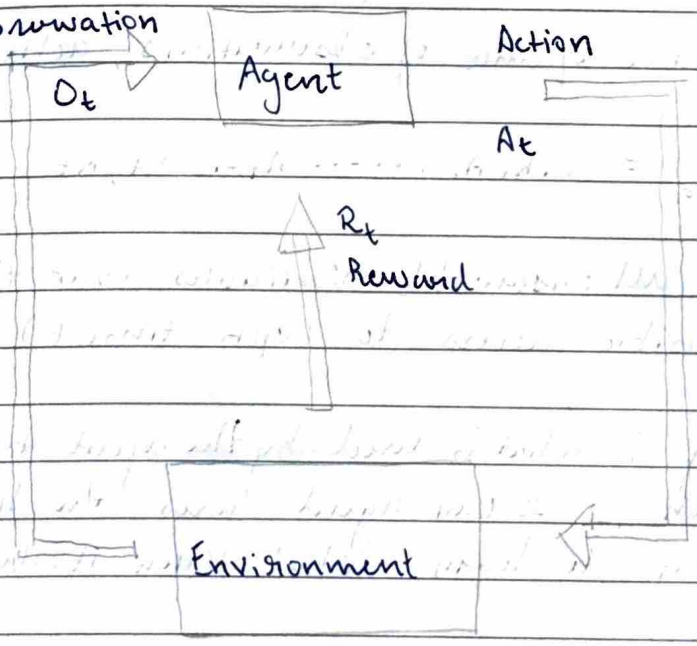
Reinforcement Learning



An agent performs an action based on the inputs it receives.
Namely O_t which is the environment or a piece of it, or a scalar feed back R_t.
Our goal is to find the algorithm for our Agent based on which it can take actions (based on information it gets) at each time step



Our Agent can only influence the environment via actions, beyond that the agent has no control over the environment. After an action, the environment changes (possibly) & the feedback is given to our agent in the form of observation (O_t) & Reward (R_t). This again forms the input to the Agent for the next Action

So at each step t the Agent :
→ recieves O_t & R_t, & based on this it takes an action A_t

And the Environment :
Recieves the action A_t & it emits Observation O_t & Reward R_t

c) History is the sequence of observations, actions, rewards

$$H_t = O_1, R_1, A_1 \ldots\ldots A_{t-1}, O_t, R_t$$

These are all observable variables that the agent has possible access to (upto time t)

So history is what is used by the agent to predict the next action, so our agent uses the history in some form to learn what action it should take in the future

Since histories tend to blowup & are large; we use States, a state is simply a (concise) summary of the history & this is what our agent uses instead of the entire history

Formally a state at time step t is a function of history

$$S_t = f(H_t)$$

2) Types of States

Environment State S_t^e is the environment's private representation & the environment chooses O_t & R_t from S_t^e to give to the agent

This isn't visible to the agent, even if it is it contains irrelevant information

Agent State S_t^a is the agent's internal representation, it summarizes everything its seen so far & uses that to pick the next action, so we simply can choose what to process, what to retain from previous time steps in our agent to help it make sure to take an action

Also it can be any function of the history

$$S_t^a = f(H_t)$$

& we can build out this function, depending on what we want our agent to be

E) Another Mathematical definition of state:

An information or Markov state is one which contains all useful information from the history.

Also a Markov Property for a state S_t is

$$P[S_{t+1} | S_t] = P[S_{t+1} | S_1, \ldots, S_t]$$

The property says that the probability of the next state (S_{t+1}) conditioned on the previous state S_t is the same for the next state even if it is conditioned on all the previous states ($S_1 \ldots S_t$)

So S_t contains enough information that discarding all the past we can still make decisions about the future simply by giving the present state S_t

"Future is independent of the past given the present"

$$H_{1:t} \longrightarrow S_t \longrightarrow H_{t+1:\infty}$$

Given this state S_t, we can throw away the history & hopefully this state is a sufficient statistic of the future. S_t fully characterizes the distribution over future actions & rewards.

To understand a markov state, S_t to be able to discard previous states & still be able to take decisions about the future, so such a state S_t must comprise enough information to be able to do this.

Thus an environment state S_t^e is markov, since the full access to an environment's information is enough to characterize future states regardless of the history

Trivially the history is also markov, since the History itself is the culmination of all previous states (observations), Rewards & Actions.

F) Fully Observable Environments
The Agent directly observes the (entire) environment state S_t^e
So the quantities

$$O_t = S_t^a = S_t^e$$ all happen to be the same

Formally this is called the Markov Decision Process (MDP)

G) Partially Observable Environments
The agent does not get to see the environment directly, entirely but rather indirectly.
Here agent state ≠ environment state, so we are supposed to build our own agent state

This is called a Partially Observable Markov decision Process
An Agent must construct its own state representation S_t^a eg.
→ Complete History : $S_t^a : H_t$

→ Beliefs of the environment state :

$$s_t^a = (P[s_t^e = s^1], \ldots, P[s_t^e = s^n])$$

we build beliefs over where we think we are in the environment

→ Recurrent Neural Network : $s_t^a = \sigma(s_{t-1}^a W_s + O_t W_o)$

4) Components of an RL Agent

Policy :
How an agent goes from a state to picking an action

Value :
How good is each state or action, how much reward do we get if we take a particular action in a particular state

Model :
How Agent thinks the environment works, or this is simply the agents view of the environment

These 3 are not always required, but they may or may not be used

PAGE No:
DATE:

I) Policy is the Agents behaviour
It is a mapping from the agents state to action
The policies could be:

→ Deterministic : $a = \pi(s)$

The goal is to make sure the policy picks such an action that maximizes future reward

→ Stochastic Policy : $\pi(a|s) = P[A = a | S = s]$

So this is just the probability of taking a particular action, after being conditioned of a particular state

II) Value function is a prediction of expected future reward.
It evaluates the goodness/badness of states
Choosing between action/states is done on comparing the total reward one might get between the two.

$$V_\pi(s) = E_\pi [R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots | S_t = s]$$

Value function depends on which way the agent behaves or rather which action it takes given a state, given a policy.

Here γ is the discounting factor based on how valuable the future/distant rewards are.

k) Model Predicts what the environment will do next, It tries to imagine what the environment does next & thus tries to learn the behaviour of the environment, & use that to make a plan & use that to figure out what to do next

We have two parts to our model

→ Transitions : P predicts the next state (i.e dynamics)

$$P^{a}_{ss'} = P[S' = s' \mid S = s, A = a]$$

Given an action a & current state = s what is the probability we transition to state s'

→ Reward : R predicts the next immediate reward

$$R^{a}_{s} = E[R \mid S = s, A = a]$$

The immediate reward we get for taking action a in state s

This is optional & a lot of agents maybe model free