Reinforcement learning.

How do we learn?

↓

we learn using trial & error based learning, through interacting with the environment around us.

↓

understanding the cause & effects of the environment using this interaction i.e we understand the response to our actions. and accordingly remember which actions yielded relatively good & bad results.

↓

once we gain this understanding as to what action yields what reward (+ve or -ve), we can try to accomplish other secondary goals.

(maybe complex)

↓

the computational approach to understand how this learning happens is called Reinforcement learning.

↓

Simpler environments → well defined rules between action, rewards, etc

↓

these will have their limitations as well

↓

but these will be of great motivation to learn more complex algorithms

famous applications of DRL:

i) Self driving car

ii) Games: a) TD Gammon → 10^{20} states.

↓

RL agent learns more about the game than we could and helps choose better strategies

b) Alphago → more states possible than the no of atoms in the universe

c) Atari games → learning to just play from pixels.

3. Robotics → Robot learning to walk
4. Finance
5. Biology
6. Inventory Management

* Basic Intuition

↓

agent : learner or decision maker in a certain given situation.

↓

for ex. a puppy which has just entered the world

↓

has a very complex body structure, allowing it to execute different possible actions (like sitting, walking, jumping, etc.)

↓

Now when being trained, it is given commands by its owner, and based on the commands it needs to decide what action to take

↓

If the action it takes matches with the command, it gets a treat (reward) and if it doesn't, then its gets no reward

↓

The pup does have the ability to perform a lot of different actions, but does not have the sense of cause & effect of these actions initially, i.e does not have any idea as to which action yields a reward and which doesn't based on the given command

so how does it match an action to a command (choose) initially?

↓

picks one at random. (having full understanding that it has no idea what it is doing)

↓

for example, given a command to sit down, and choosing a ~~rde~~random action, it chooses to sit down, after which it waits for a feedback for its action, where it receives a treat (which has +ve sentiment)

↓

Behaviour pattern would be to wanting to maximise this reward.

↓

Next, it is given an instruction to maybe walk and it decides to perform some other than just sitting (again randomly) and waits for feedback, but does not receive any reward this time. (relatively discouragers or -ve sentiment)

↓

This could mean two things → a) action could be bad in general.

b) wrong mapping case.

↓

As time goes on, and more interaction is done, the puppy will have a better understanding of what command maps to what action.

↓

and accordingly try to maximise the incoming future.

↓

more interaction → more feedback → better mapping of commands and actions, and hence better chance to maximise reward.

This process of systematically proposing and testing hypothesis is the basic concept of RL.

* hypothesis → proposition made on the basis of reasoning, without any assumption of its truth.

↓

This process is not this simple, and there are some basic problems in RL i.e

a) exploration - exploitation dilemma

↓

may result in +ve or -ve reward, but ↑ inc knowledge

exploration: exploring potential hypothesis for now, to choose actions. (important to improve knowledge)

exploitation: exploiting the already available knowledge to make the best possible decision (important to increase reward)

↓

problem is on how to balance both i.e to be satisfied with what we already know or to experiment & see if there is a better strategy or action given that it may result in a worse result.

b) Delayed rewards → Certain actions might have less pay-offs (rewards) in the short term but higher in the long term & vice-versa. Making the decision to on which action to take also a dilemma.

↓

like dropping a catch in a match of cricket; might just feel like a wicket not taken, but might end up losing the match.

↓

or saving a run in a match, could in the end be the deciding factor

↓

so accounting for such actions in a delayed feedback system i.e. their reward is realised later is an issue to be worked on

The main takeaway → Learning from experience is the main approach to learning here.