The study of decision making.

RL paradigm – there is no supervisor, but only a reward signal.
– feedback is delayed, not instantaneous.
– time really matters (sequential, non i.i.d data)
– agent can modify the data it sees.

Eg. – Fly stunt manoeuvres in a helicopter
– Defeat the world at Backgammon
– Manage an investment portfolio.
– Control a power station
– Make a humanoid robot walk.
– Play many different Atari games better than humans.

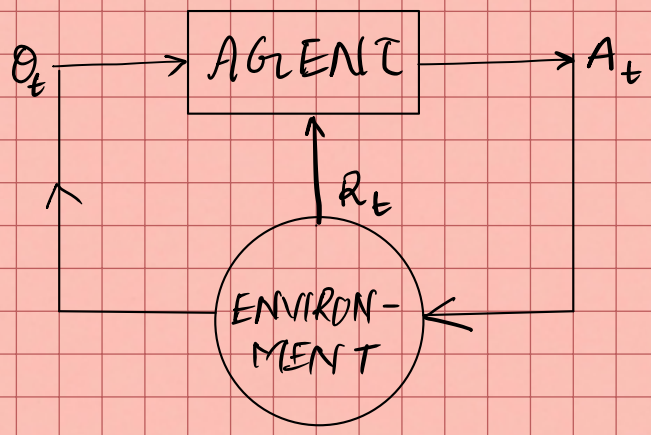
Rewards:

- A reward R_t is a scalar feedback signal
- Indicates how well agent is doing at step t
- The agent's job is to maximize cumulative reward.

Reward Hypothesis – All goals can be described by maximization of expected cumulative reward.

Sequential Decision Making

⇒ Goal: Select actions to maximize future rewards.
– Actions may have long term consequences
– Reward may be delayed.
– It may be better to sacrifice immediate reward to gain long-term reward.



At each step t the agent:
– Executes action A_t
– Receives observation O_t
– Receives scalar reward R_t

The environment:
– Receives action A_t
– Emits observation O_t
– Emits scalar reward R_t

History and State: The history is the sequence of observations, actions and rewards.

$$H_t = A_1, O_1, R_1, \ldots\ldots, A_t, O_t, R_t.$$

i.e., all variables upto time t.

What happens next depends on history. Our goal is to make a mapping from $H_t \rightarrow A_{t+1}$. The environment selects observations/rewards.

State is the information used to determine what happens next.
Formally state is a function of history:

$$S_t = f(H_t)$$

Environment State: S_t^e is an internal representation of the environment. It is used by the environment to pick the next observation / reward.

Agent State: It is the agent's internal representation (S_t^a): set of numbers which live inside our algorithm and help take the next action.

It can be a function of history: $S_t^a = f(H_t)$

Information State: An information state contains all useful information from history.

Def: A state S_t is Markov iff:

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

The future is independent of the past given the present.

The environment state S_t^e is Markov. (by def.)
The history H_t is Markov. (not a very useful one though)

Fully Observable Environments: Agent directly observes environment state.

$$O_t = S_t^a = S_t^e$$

- Agent state = environment state = information state

Formally this is a Markov Decision State (MDP)

Partial Observability: Agent indirectly observes the environment.

Now agent state ≠ environment state.

- Formally this is a partially observable Markov Decision Process (POMDP)
- Agent must construct its own environment S_t^a -
- Complete history $S_t^a = H_t$
- Beliefs of environment state $S_t^a = (P[S_t^e = s^1], \ldots, P[S_t^e = s^n])$
- RNN - $S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o)$

Major Components of an RL Agent:

An RL agent may include one or more of these components:

- Policy: agent's behaviour function
- Value function: how good is each state / action
- Model: agent's representation of environment

Policy - It is a map from state to action.

eg. deterministic policy $\pi(s) = a$

stochastic policy $\pi(a|s) = P[A = a | S = s]$

Value function: Is a prediction of a future reward.

Used to evaluate goodness / badness of state.

And therefore to select between actions -

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

Model: A model predicts what the environment will do next.

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

Rewards – R predicts the next immediate reward e.g.

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

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

Categorizing RL agents:

- Value based
  - No policy (implicit)
  - Value function
- Policy based
  - Policy
  - No value function
- Actor Critic
  - Policy
  - Value function
- Model free
  - Policy and/or value function
  - No model
- Model based
  - Policy and/or value function
  - Model

Problems with RL:

Two fundamental problems in sequential decision making

- Reinforcement learning
  - The environment is initially unknown
  - The agent interacts with the environment
  - The agent improves its policy.
- Planning
  - The model of the environment is fully known
  - The agent performs computations with its model (without any external interaction)
  - The agent improves its policy

Prediction and Control:

- Prediction: evaluate the future
  - given a policy
- Control: optimise the future
  - find the best policy.