[SYNTHESIS]. DEEP REINFORCEMENT LEARNING

Rudresh Mishra, Ricardo Rodriguez
IMT Atlantique

Introduction

Reinforcement learning is the process of making the systems learn from the environment by itself. so the system has to complete the goal by applying the trial and error approach. What makes reinforcement learning different from other machine learning paradigms? 1) There is no supervisor, only a reward signal 2) Feedback is delayed, not instantaneous 3) Time really matters (sequential, non i.i.d data) 4) Agent's actions affect the subsequent data it receives.

what is Rewards:- A reward R_t is a scalar feedback signal which Indicates how well the agent is doing at step t. The agent's job is to maximise cumulative reward Reinforcement learning is based on the reward hypothesis.
Goal: select actions to maximise total future reward, Actions may have long term consequences. The reward may be delayed It may be better to sacrifice immediate reward to gain more long-term reward.

Markov-Decision Process

Agent and Environment.
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+1} Emits scalar reward R_{t+1}
t increments at env. step.

All the sequence of observations, actions, and rewards during the agent's lifetime up to time step t is called the history,
$Ht = S_1, A_1, R_2, \ldots\ldots\ldots, S_{t-1}, A_{t-1}, R_t$

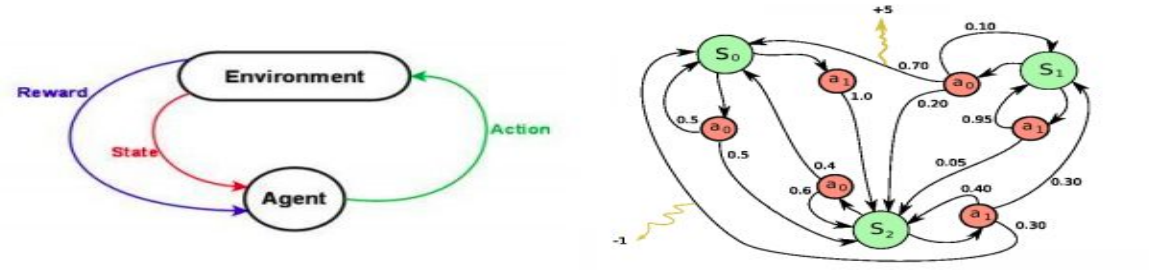

Figure 2: *Left:* reinforcement learning problem. *Right:* Markov decision process. Image credit: Wikipedia.

Information State
An information state a.k.a Markov state contains all useful information from the history. We use that Markov state to represent the agent's state. The future is independent of the past given the present $H_{1:t} \rightarrow S_t \rightarrow H_{t+1:\infty}$ Once the state is known, the history may be thrown away i.e. The state is a sufficient statistic of the future The environment state S^e_t is Markov The history H_t is Markov.

Major Components of an RL Agent

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

Discounted Future Reward

Our environment is stochastic, we can never be sure if we will get the same rewards the next time we perform A good strategy for an agent would be to always choose an action, that maximizes the discounted future reward.m the same actions. The more into the future we go, the more it may diverge. For that reason, it is common to use discounted future reward instead: $Rt=r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \ldots + \gamma^{n-t} r_n$ Here γ is the discount factor between 0 and 1 – the more into the future the reward is, the less we take it into consideration.

Q-learning

Q-learning is an off-policy reinforcement learning algorithm that seeks to find the best action to take given the current state. It's considered off-policy because the q-learning function learns from actions that are outside the current policy, like taking random actions, and therefore the policy isn't needed. More specifically, q-learning seeks to learn a policy that maximizes the total reward.

Deep Q Network

Neural networks are exceptionally good in coming up with good features for highly structured data. We could represent our Q-function with a neural network, that takes the state (four game screens) and action as input and outputs the corresponding Q-value. Alternatively, we could take only game screens as input and output the Q-value for each possible action. This approach has the advantage, that if we want to perform a Q-value update or pick the action with highest Q-value, we only have to do one forward pass through the network and have all Q-values for all actions immediately available.