Reinforcement Learning

- learner is a decision making agent that
- takes actions in an environment and
- receives reward (or penalty) for its actions in trying to solve a problem
- after a set of trial and error runs, it should learn the best policy,
- which is the sequence of actions that maximize the total reward
- example, a machine that learns to play chess
- supervised learning not possible
- costly to take through many games and indicate the best move for each position
- no such thing as the best move, goodness of a move depends on the moves that follow

Reinforcement Learning

- a single move does not count, a sequence of moves is good if after playing we win the game
- only feedback is at the end of the game when we win or lose the game
- robot that is placed in maze
- robot can move in one of four directions, a sequence of movements to reach the exit
- there is no feedback and robot tries many moves until it reaches the exit
- there is no opponent, preference for shorter trajectories - play against time
- both applications: chess and robot
- there is a decision maker called agent
- that is placed in an environment

Reinforcement Learning

- chess game: game player is the decision maker and the environment is the board
- robot in maze: the maze is environment of robot
- at any time, the environment is in a certain state that is one of a set of possible states
- e.g. state of the board, the position of the robot in the maze
- decision maker has a set of actions possible
- legal movement of pieces on the chess board
- movement of the robot in possible directions without hitting the wall and so forth

Reinforcement Learning

- once an action is chosen and taken, the state changes
- solution to the task requires a sequence of actions and get feedback in the form of a reward
- rarely, generally only when the complete sequence is carried out
- reward defines the problem and is necessary if we want a learning agent
- learning agent learns the best sequence of actions to sove a problem where the best is
- quantified as the sequence of actions that has the maximum cumulative reward

Reinforcement Learning

- reinforcement learning differs from other learing methods in the following respects:
- learning with a critic as opposed to learning with a teacher which is supervised learning
- critic differs from a teacher in that it does not tell us what to do but
- only how well we have been doing in the past; critic never informs in advance
- feedback from the critic is scarce and when it comes, it comes late
- this leads to the credit assignment problem
- after taking several actions and getting the reward
- like to assess the individual actions did in the past and
- find the moves that led us to win the reward so that it can be recorded and recall them later on

Reinforcement Learning

- learns to generate an internal value for the intermediate states or
- actions as to how good they are in leading us to the goal and getting us to the real reward
- once internal reward mechanism is leared the agent can just take the local actions to maximize it
- solution to the task requires a sequence of actions and from this perspective,
- use Markov decision process to model the agent
- the difference is that in case of Markov models, there is an external process that generates a sequence of signal, example speech, observe and model

Reinforcement Learning

- in the reinforcement learning, it is the agent that generates the sequence of actions
- similarly, observable and hidden Markov models, the states are observed or hidden
- here, sometimes, a partially observable Markov decision process where
- the agent does not know its state exactly but should infer it with some uncertainty through observations using sensors
- for robot moving in a room, the robot may not know its exact position in the room, nor th exact location of obstacles nor the goal, and
- should make decisions through a limited image provided by a camera

Reinforcement Learning

- Single State Case: K-armed bandit
- K-armed bandit hypothetical slot machine with K levers
- action is to choose and pull one of levers and
- win a certain amount of money that is the reward associated with lever (action)
- task is to decide which lever to pull to maximize the reward
- this is a classification problem choose one of K
- if supervised learning teacher would tell us the correct class, the lever leading to maximum earning
- in reinforcement learning only try different levers and keep track of the best

Reinforcement Learning

- this example, one state, one slot machine, only decide on action, get a reward after a sinlge action, reward is not delayed, immediately see the value of action
- let say $Q(a)$ is the value of action a
- initially $Q(a) = 0$ for all a, try action a and get reward $r_a \geq 0$
- if rewards are deterministic, always get the same r_a for any pull of a, in such case, just set $Q(a) = r_a$
- want to exploit, once we find an action a such that $Q(a) > 0$, keep choosing it and get r_a at each pull
- quite possible that there is another lever with a higher reward, so need to explore
- choose different actions and store $Q(a)$ for all a

Reinforcement Learning

- whenever we want to exploit, we can choose the action with the maximum value, that is
- choose a^* if $Q(a^*) = \max_a Q(a)$
- if rewards are not deterministic but stochastic, we get a different reward each time we choose the same action
- the amount of reward is defined by the probability distribution $p(r|a)$
- in such case define $Q_t(a)$ as the estimate of the value of action a at time t
- it is an average of all rewards received when action a was chosen before time t
- online update can be defined as - delta rule

$$Q_{t+1}(a) \leftarrow Q_t(a) + \eta[r_{t+1}(a) - Q_t(a)]$$

where $r_{t+1}(a)$ is the reward received after taking action a at time $(t+1)$st time

Reinforcement Learning

- η is learning factor (gradually decreased in time for convergence)
- r_{t+1} is desired output, $Q_t(a)$ is current prediction
- $Q_{t+1}(a)$ is expected value of action a at time $t+1$
- and converges to the mean of $p(r|a)$ as t increases
- generalizing for full reinforcement learning
- several states - several slot machines with different reward probabilities $p(r|s_i, a_j)$
- need to learn $Q(s_i, a_j)$ which is the value of taking action a_j when in state s_i
- the actions affect not only the reward but also the next state and move from one state to another
- rewards are delayed and need to be able to estimate immediate values from delayed rewards

Reinforcement Learning

- Elements of reinforcement learning
- learning decision making is called the agent
- agent ineracts with the environment that includes everything outside the agent
- agent has sensors to decide on its state in the environment and takes an action that modifies its state
- when the agent takes an action, the environment provides a reward

Reinforcement Learning

- when the agent in state s_t takes the action a_t, the clock ticks,
- reward $r_{t+1} \in \mathcal{R}$ is received and the agent moves to the next state s_{t+1}
- the problem is modeled using a Markov decision process
- the reward and next state are sampled from their respective probability distributions $p(r_{t+1}|s_t, a_t)$ and $P(s_{t+1}|s_t, a_t)$
- Markov system where the state and reward in the next time step depend only on the current state and action