|  |
| --- |
| **Submitted:** 12/11/2017 8:08:01 PM   **Word Count:** 399 |
| ***Reinforcement learning (RL)*** is a type of Machine Learning (ML) technique which allows the learning agent (i.e. the machine/system which is undergoing the process of becoming intelligent) to learn by interacting with an environment. An RL agent learns from the consequences of its actions which were performed based on its current state (position of the agent in the environment at that time), rather than from being explicitly taught and it selects its next action based on its past experiences (exploitation) and by new choices (exploration), which is essentially trial and error learning. The reinforcement (feedback) signal that the RL-agent receives is a numerical reward, which encodes the success of an action's outcome, and the agent seeks to learn a method to select actions that maximize the accumulated reward over time.  There are mainly three approaches to implement RL, namely ***Temporal Difference(TD), Monte Carlo(MC) and Dynamic Programming(DP)***. DP is a model-based approach. A perfect model of the environment; consisting of states, their probability of occurring along with immediate reward; is provided and agent learns a method to use this model and accumulate maximum reward each step. MC is a model-less approach where a perfect model isn't available instead learning is based on average of rewards accumulated whenever the agent goes to that state (experience). Using this experience, the agent finds a best method to accumulate maximum reward till end of one episode (complete round of interaction of agent with the environment, i.e., an episode is one robot battle in a series of *n* battles). TD is a combination of both DP and MC. It is also a model-less approach where RL agent accumulates reward based on experience and updates the rewards each step (bootstrapping) as compared to MC where updates take place at the end of each episode. RL has a limited scope in ML, as RL could be computationally impossible to implement for environments with large/infinite state-space model as it involves visiting and updating rewards for every state repeatedly.  As opposed to other ML approaches, RL is not suitable for problems that doesn't generate distinct end reward for each state. For instance, in ACR, for an image input, the actual output can be a label different from a desired label. Hence, the only teaching feedback is, if the output is correct or incorrect instead of specifying *how* it is incorrect or what is the *negative reward* for that error. |