SOC-retroRL-Week-1-report

Chinthaparthi Babu Sujan Reddy

June 2024

1 Introduction

Since, I have little knowledge of reinforcement learning,I have started from knowing what is reinforcement learning and it's applications.Then I got familiar with basic terminology required and some popular algorithms used to solve the k-armed bandit problem(and other problems).Then I have read about Markov Decision Process,the cartpole problem and a bit about dynamic programming.

2 Reinforcement Learning

Reinforcement learning is basically learning from experience to achieve the goal of the agent while interacting with the environment and learning from different outcomes without any initial knowledge bias.This absence of knowledge bias differentiates it from supervised learning.

3 k-armed bandit problem

The k-armed bandit problem can be considered a slot machine with k levers and each lever may have a different reward and you need to maximize your reward without knowing how the reward system works.

3.1 Algorithms I have studied

- The greedy algorithm
- $\epsilon-$greedy algorithm
- greedy and $\epsilon-$greedy algorithm with constant step size(for environements in which rewards vary with time)
- Algorithms with optimistic initial values
- UCB(Upper-Confidence-Bound)
- Gradient Bandit algorithm

4 Markov Decision Process

I got an insight about how a lot of real life problems can be broken down into this system.A problem can be said to be a Markov Process, if it's reward and next state depends on the current state and action.The universe can be broken down into agent and the environment, the user interacts(action) with the current environment(state) and goes to a new state and can receive a reward.The aim is to maximize the final reward. In a mathematical sense, the rewards can be adjusted to achieve our goal.For example, for chess the reward is +1 for a win, -1 for a loss and 0 for other scenarios, for a maze-solving car, each step taken can be -1,etc.But this process can end in finite steps(episodes) or infinite.So to unify these we can discount the reward(r_t) at time t to $\gamma^{r-1} r_t$, if $\gamma < 1$ to make the final reward finite, and episodic process can be extended to inifinity by rewarding zero forever after the episode has finished.

5 Cart-pole problem

I have gone through the problem and the code and got a sense of reinforcement learning.