# RL from Scratch - WiDS Report (Project UID 74)

Suketu Patni, 23B1299

January 22, 2024

**Abstract**

Reinforcement Learning (RL) essentially studies which actions an agent must take in an environment (i.e. among a variety of different actions) to maximize the cumulative expected reward that it gets. It forms an important sub-branch of machine learning.

## 1 Week 1

We created the MDPs (Markov Decision Processes, they are environments for our RL agents) for some of the example problems from the "Grokking Deep RL" textbook by Miguel Morales. An assignment on the math of Markov Chains (a set of states with a transition matrix of probabilities) was given so that the MDPs could be worked out by hand. We created MDPs (programmatically) for the following environments:-

1. Bandit Walk (BW)
2. Bandit Slippery Walk (BSW)
3. Slippery Walk Five (SWF)
4. Frozen lake (FL)

Once this was done, we verified if our MDPs were correct by cross checking with the OpenAI Gym library in Python which already contains all of these MDPs pre-built.

## 2 Week 2

We worked on devising optimal policies for our MDPs that were made in Week 1 using 2 of the simplest RL techniques: Policy Iteration (PI) ["evaluates" the entire policy then improves it] and Value Iteration (VI) [improving a policy greedily even when policy evaluation is not complete]. These rely on finding the value function $v_\pi(s)$ and the action-value function $q_\pi(s,a)$ under a given policy $\pi$. This is implemented by the Bellman Equation:-

$$v_\pi(s) = \sum_a \pi(a|s) \left( \sum_{s',r} p(s',r|s,a) \ [r + \gamma v_\pi(s')] \right)$$

We solve such complicated equations by iteratively calculating the RHS and replacing the LHS by it until the $v_\pi(s)$ values start to converge.
Further, we also wrote a function to test our optimal policy on the FL environment, once it had been found, to check how many times (in a set of 10,000 runs) the policy yielded in the agent getting a nonzero reward at the end of an episode.

# 3 Week 3

In this week, we looked at Multi Armed Bandits (MABs), RL agents in which the horizon (i.e. episode length in discretized timesteps) is 1 i.e. they have only one decision to make per episode, but they make use of info gained over all the previous episodes (they don't have access to the MDP). Their aim is to figure out the action which leads to the state with the maximum expected reward. There are multiple strategies to do so, each involving a struggle between exploiting the known information and exploring the unknown information. We tested 6 distinct strategies on both Bernoulli and Gaussian Bandits:

1. Random (pure exploration)
2. Greedy (pure exploitation) (most likely to fail)
3. $\epsilon$-greedy (mostly greedy, sometimes random)
4. Exponentially decaying $\epsilon$-greedy
5. Softmax
6. Upper Confidence Bound (UCB)
7. Thompson Sampling

Further, we learned how to compare 2 strategies by calculating the total "regret" i.e. the sum of the per-episode difference of the true expected reward of the optimal action and the true expected reward of the selected action.