**Deep Reinforcement Learning – Assignment 1**

**Section 1 – Tabular Q learning**

1. Although value and policy iteration are constituted as efficient approaches in the field of dynamic programming, their process requires the evaluations of all available actions or states, respectively. Hence, under complex environments with numerous states, even when the true dynamics are known, those methods are infeasible for implementation.

Moreover, as model-based methods, they are dependent upon the knowledge of the transition probability distribution for the update of the value function (based on the Bellman equation), which is lacking or too complicated to calculate in an unknown dynamics environment.

1. For scenarios where there is no prior knowledge about environment dynamics i.e. the transition probability matrix is missing, model-free algorithms are aimed to directly estimate the expected reward function by observing state-action pairs, instead of modeling the true “physics “of the environment. Since those methods are based on sampling strategies, they can infer from a sufficient number of observations, not requiring to sample of all available states and actions. model-free methods have advantages over model-based methods which tries to construct a sufficiently accurate environment model of a complex problem.
2. The difference between Q-learning and Sarsa algorithms(both based on the temporal difference algorithm) lies in the difference between **on-policy** methods which try to evaluate and improve the policy that is used to make the decision, having interaction with the environment, and **off-policies** methods which attempt to improve a policy other than the one used to generate the data it infers on. **Sarsa** algorithm is an on-policy method that updates the q function based on the current policy while **q-learning** is an off-policy algorithm that updates based on a greedy action which gives the maximum Q-value for the state follows an optimal policy. Hence q-learning will always search for the optimal solution, even at a price of high risk (as in the cliff class example).
3. is an algorithm that aims to balance between exploration and exploitation by randomly acting under each of the approaches. Exploitation chooses the action which maximizes the reward. By being greedy concerning action-value estimates, we may not get the most reward and lead to sub-optimal actions. exploration allows an agent to improve its current knowledge about each action, hopefully leading to long-term benefits. By maintaining the ability to explore forever, under the law of large numbers, (epsilon refers to the probability of choosing to explore) ensure that we will find the optimal policy (by exploring all available actions and exploit them).