# 20231499_CT5134_Answersheet_Part2

### Q 3 A

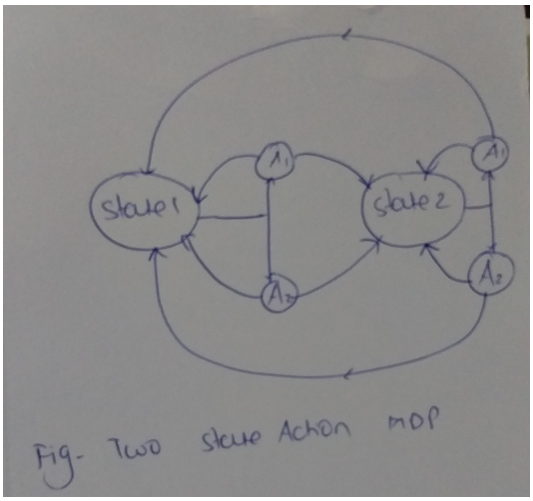
- Performing RL task on very complex tasks like google deep mind , when scaling this process it is going to be very time consuming and require a lot of iterations.
- Also in real time for the agents it might not be possible for observing the complete environment in this case the partial observations might not give optimal solutions.
- Rl learns from continuous exploration and exploitation but in real world more exploration might not be possible which might hamper progress or results.
- It will also be difficult to implement when there is a lot of labelled data too.

### Q 3 B

- Value state Vs is expected value of a policy when agent starts following it from state s while Qsa is the expected value of the first taking action a from state s and then following the policy.
- SO in Q you can first predict a state and then following what the policy might give.
- Suppose you are one step away from terminating state then Vs = -1 , but in Qsa you will take one step then policy will xome in picture then you'll come back to postion and follow correct direction which will give you Qsa = -3.

### Q 3 C

- The MDP formalism has the following components.
- A finite set of states S, fully or partially observalble.
- A discrete set of actions for each state, i.e. A
- The real valued reward R
- Transitioning probability between two states T(S,a)
- Finally the goal that is to find the optimal policy $pi$.




- The markov process states that the current state and only the current state can decide the future state and that the past states have nothing to do with the future states. Like in checkers game

only the current move will matter for the next move and not the past moves.

- In MDP the transition of the next state is guided by the markov process, the state transitions must satisfy the markov property to move forward in the process.

### Q 3 D

- In policy based method we explicitly build representations based upon the policy and keep them in memory during learning
- In Value based explicit policy is not there, only a avalue function i.e it is implicit and directly derived from value function.
- In value based we find optimal value fucntion and policy based on that while in policy based policy evaluation and improvement is done.
- value based is optimality bellman operator i.e non linear while policy is bellman operator.
- Q learning and value iteration are value based while cross entropy or reinforce are policy based.

### Q 4 A

- It is  a model free method, it does not require a complete model or environment to learn a policy.
- It is policy based and on-policy method, the policy being followed requires new data obtained from environment.
- It is a technique to find optima without overhead of calculating derivatives i.e. no back propagation.
- It basically just sees the outputs chose the inputs that gave best outputs and then tune them till outputs are satisfied.
- on and off policy is based on the transitional predictions , whether we update the Q values on actions undertaken to current policy or not.
- So upadation based on best possible action or based on action of best policy is off policy and on policy.
- Q learning is off policy while SARSA is on policy.

### Q 4 B

- Q learning directly learns from optimal policy while SARSA learns a near-optimal policy while exploring.
- Q learning has higher per sample variance while SARSA may suffer from converging.
- SARSA approach convergence while Q learning ignore them , this makes SARSA more conservative.
- If goal is to train optimal agent in simulation or low cost fast iterating then Q learning is the one , while if agents learns online and we care about rewards gained whilst learning then SARSA may be better.

### Q 4 C

- Continuous state define the relation between input and output in state space , input being vector of length and output vector of length and number of states.

- Curse of dimensionality says that if we add more state variable then the computational requirements will also grow exponentially with the no of variables, which will make the decision making more complicated and time consuming.

## Q 4 D

- In DQN we use neural network to approximate the Q value function, state given as input and q value of all possible actions is generated as output.
- Steps:
- All the past experience is stored in user memory
- next action is determined by the maximun output of the Q network.
- Loss function is mean squared error of the predicted Q value. it is a type of regression problem, we do not know actual or target value.