Assignment-2 (Checkpoint)

Team-39

(Kunal Beniwal and Vaibhav Rao)

So far we have represented value function by a look up table for every state we have a value V(s) or for every state-action pair has a value Q(s, a). There are few problems which we face with large MDP's like there will be too many states or actions to store in memory and if anyhow we overcome that the model would be too slow to learn the value of each state individually.

Action for large MDPs

Estimate value function with function approximation

$$\hat{v}(s, \mathbf{w}) \approx v_\pi(s)$$
$$\hat{q}(s, a, \mathbf{w}) \approx q_\pi(s, a)$$

Environment Defined:

My Environment has 16 states in total. It is a 4*4.

Start Position is at (0,0)

Goal is at (3,3) - Reward of +30 on reaching Goal.

There is also a bonus reward FOOD at (1,2) which with grant +2 reward.

There is a TRAP at (2,0) that will end up causing the agent to get a negative -5 reward.

The Environment will also give a -1 reward if the agent breaches the size of the grid, i.e. tries to go out of bound.

Hence my Reward set has {0,+30,+2,-5,-1} Rewards for this environment.

ACTIONS: There are 4 possible actions – {UP, DOWN, LEFT, RIGHT}

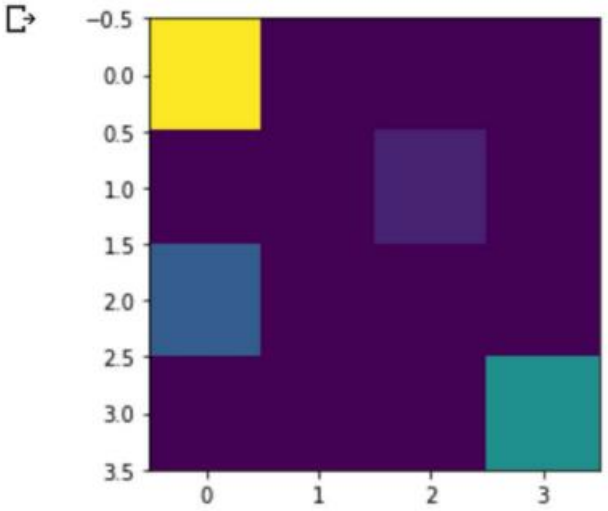
The main objective of the agent is to reach the Goal Position from the starting point Maximizing the reward and also in minimal timesteps.

There are also 2 environments I have defined:

a. Deterministic – Defined in class GridEnvironment in Function step()

b. Stochastic – Defined in class GridEnvironment in Function step_stochastic()

```
env = GridEnvironment()
obs = env.reset()
env.render()
```



We can see the location of Start pos (0,0) Goal (3,3) Food(1,2) and Trap (2,0)

Naive Q-learning oscillates or diverge with neural nets.

- Data is sequential
- Policy changes rapidly with slight changes to Q-values
- Scale of rewards and Q-values is unknown

Deep Q Networks (DQN) provides the solution to above listed problems as it uses experience replay and fixed Q-targets.

- Take action a_t according to ϵ-greedy policy
- Store transition $(s_t, a_t, r_{t+1}, s_{t+1})$ in replay memory $\mathcal{D}$
- Sample random mini-batch of transitions (s, a, r, s') from $\mathcal{D}$
- Compute Q-learning targets w.r.t. old, fixed parameters w^-
- Optimise MSE between Q-network and Q-learning targets

$$\mathcal{L}_i(w_i) = \mathbb{E}_{s,a,r,s' \sim \mathcal{D}_i} \left[\left(r + \gamma \max_{a'} Q(s', a'; w_i^-) - Q(s, a; w_i) \right)^2 \right]$$

- Using variant of stochastic gradient descent