Programming Assignment 3

Prakhar Thakuria(EE16B061)

15 February 2019

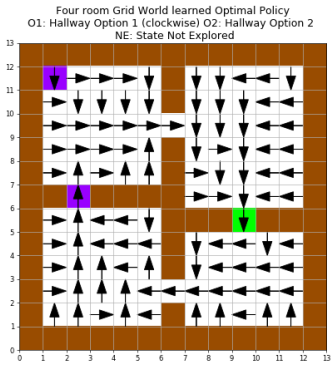

Figure 1: Clockwise Policy

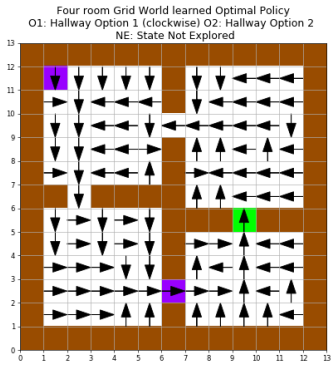

Figure 2: Anti Clockwise Policy

Figure 3: Option Policies

0.1 Introduction

The environment was generated as asked in the question. Then 2 policies were generated for exiting the room to the hallways for every state. The policies were going to the hall ways clockwise or anti-clockwise. The policies are as depicted above and as considered to be from paper two option possibilities. Figure 1 and 2.

0.2 Goal G1

Hyperparametrs:Goal = G1
Action : 1/8
Options = 1/16
Options and actions = 1/8
Figure 4 to 12

0.2.1 Inferences and observation

1. Even though In starting with options do very great job by getting to the solution as compared to the one with action which goes around for quite some time in the start but as episodes keep increasing then with just action we are able to reach the solution much faster than and do as good as SMDP

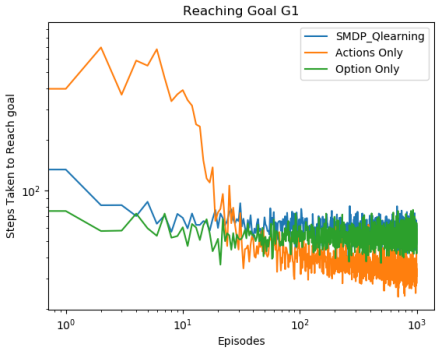

Figure 4: Goal G1

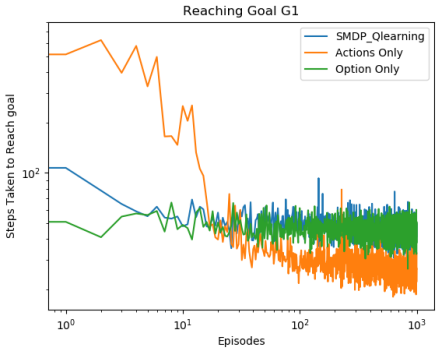

Figure 5: Center of Room 4 as Start State

Figure 6: Reaching Goal G1

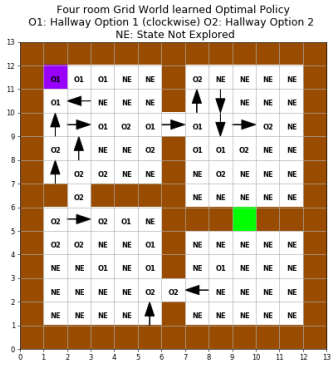

Figure 7: Goal G1

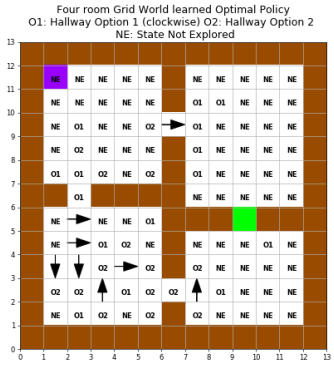

Figure 8: Center of Room 4 as Start State

Figure 9: Optimal Policy Chosen for SMDP

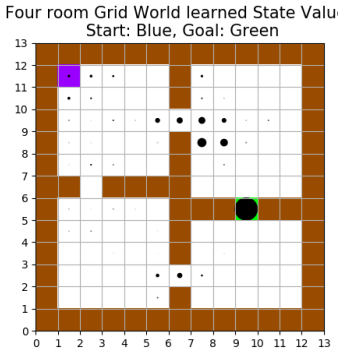

Figure 10: Goal G1

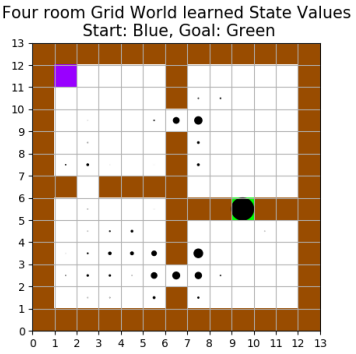

Figure 11: Center of Room 4 as Start State

Figure 12: Q values visualized for SMDP

or just Options.

2. But the Same comparison when done with change when the start state is in the middle of room 4 many changes, as we have decreased the number of steps required to reach to the goal by moving the start state closer to the goal state. And hence we see a significant decrease in the number of steps taken by all the three algorithms.

3. We can see the value function of is quite large around the hallways,as options take from there lead you directly to the goal in the most optimal number of steps. 4. For start state in the first room we have the value function much greater in the upper part of the grid while in the second case it is much greater than in lower regions

0.2.2 Changes by changing start states

1.The Number of steps Taken to reach the goal optimally is decreased as the start is state is quite near to the goal.

2.The Value function is quite is easily seen for quite some number of states when the start state is in Room 4 as the it is closer to the goal state

3.The circle shift down as that path being more optimal in the second case

Figure 13: Goal G2

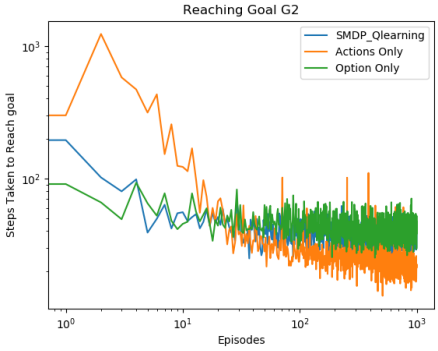

Figure 14: Center of Room 4 as Start State

Figure 15: Reaching Goal G2

0.3 Goal = G2

Action : 1/8
Options = 1/8
Options and actions = 1/4
Figure 13 to 21

0.3.1 Observation and Infrences

1.This time action is quite at par from only option as reaching the goal sate which is in the middle of the room and going there would be very difficult with just options(As the options are from one hallway to another) .

2. We can see the proof of the above as in both cases the goal state by surrounded by the actions leading to it only

3. In this case as compared to the other we have a lot more explorations

4. Again we we have bigger value function in just the hallways nearest to the goal as there is a number of action taken before reaching the goal which result in the decrease because of the discount factor

0.3.2 Changes by changing start states

1.The Number of steps Taken to reach the goal optimally is decreased as the start is state is quite near to the goal and this time is change is quite apparent as the change is pretty large.

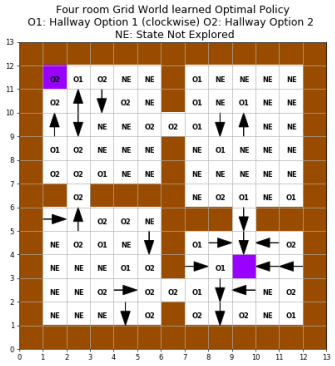

Figure 16: Start State at the top

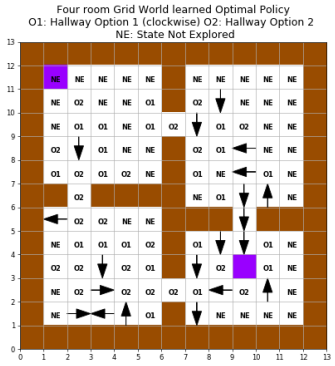

Figure 17: Center of Room 4 as Start State

Figure 18: Optimal Policy Chosen for SMDP

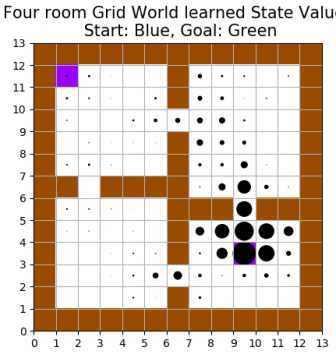

Figure 19: Start State At the top

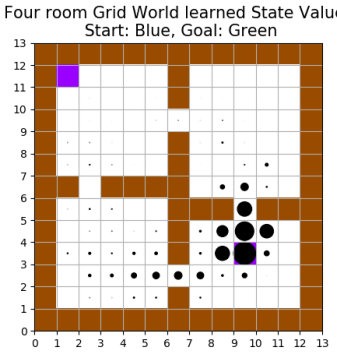

Figure 20: Center of Room 4 as Start State

Figure 21: Q values visualized for SMDP

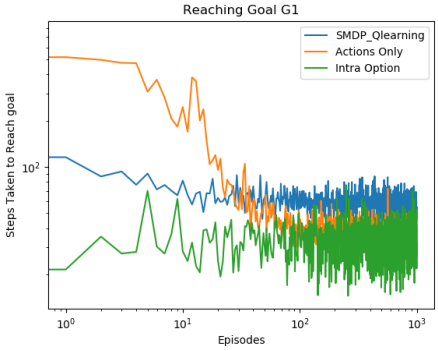

Figure 22: Start State at the top

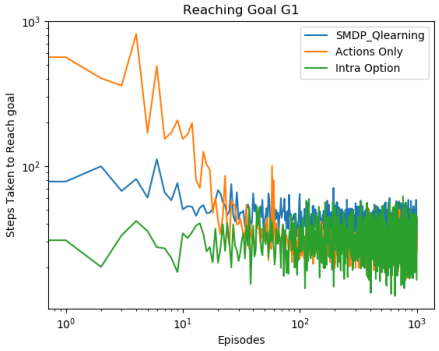

Figure 23: Center of Room 4 as Start State

Figure 24: Intra Option Reaching Goal G1

2.The Value function is quite is easily seen for quite some number of states when the start state is in Room 4 as the it is closer to the goal state

3.The circle shift down as that path being more optimal in the second case

0.4 Intra-Option Q learning

Goal = G1
Action : 1/8
Options = 1/8
Intra-Option Q learning: 1, Decreased at rate = .75
Figure = 24
Goal = G2
Action : 1/8
Options = 1/8
Intra-Option Q learning: 1, Decreased at rate = .75
Figure = 27

0.4.1 Observation and Infrences

1. Intra-Option Q-learning learns about every applicable options from every experience. We can observe that optimal policy learned by Intra-Option Q-learning for goal G1 from just 100 episodes picks only options from every

Figure 25: Start State at the top

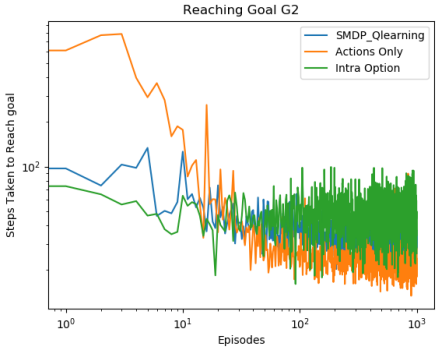

Figure 26: Center of Room 4 as Start State

Figure 27: Intra Option Reaching Goal G2

state which is correct since goal state is termial state for two options so ideally agent should be picking option which will lead to reaching goal more quickly.

2. Policy Seems to be converged pretty quickly in this case and hence we can see it being almost constant earlier than anything else.

3. It converges to policies as good as choosing just the action without options and far too quickly.

0.5 Deep RL

0.5.1 Optimal Hyper Parameters

There were two set of hyper parameters used by me one with 3 layer Deep Hidden layer network and the other one 2 layer network.

2-layer network

$\epsilon = 1$

$\epsilon_{decay} = 0.995$
$Size\ of\ layer1 = 64$
$Size\ of\ layer2 = 64$
$number\ of\ episodes = 500$
$Mini\ Batch\ Size = 20$

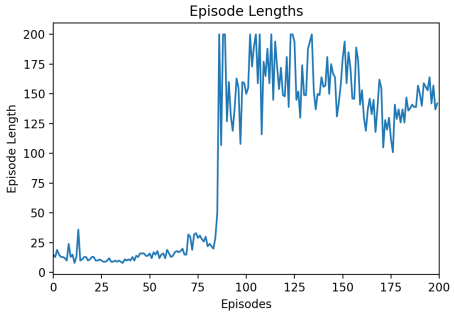

Figure 28: Episode length

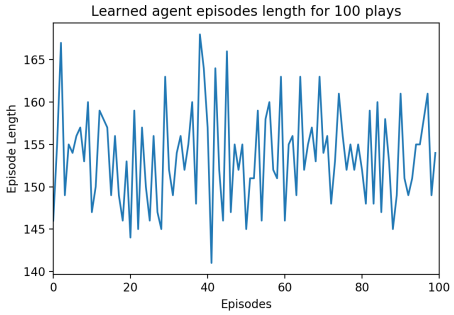

Figure 29: Results

Figure 30: No. episode =200 , minibatchsize = 35,reward = -10

3 -layer network

$\epsilon = 1$

$\epsilon_{decay} = 0.995$
$Size\ of\ layer1 = 20$
$Size\ of\ layer2 = 20$
$Size\ of\ layer3 = 20$
$number\ of\ episodes = 300$
$Mini\ Batch\ Size = 32$

0.5.2 Tuning For 2 layer Network

Figure 30 to 45

0.5.3 Observation and Inferences

1.Speed of learning(Number of Episodes) depends very largely on the Batch Size, The Smaller the batch size the larger time to learn. If we take too big a batch size the updates are lot oscillatory in nature.

2.The updates are smaller for a small network and hence faster and the speed of convergence is also hence faster.

3.After Reducing the DQN to very small hidden layer size the convergence is not good enough. Like for Hidden Layer Reduces too much then you won't be able to Solve the problem at all.

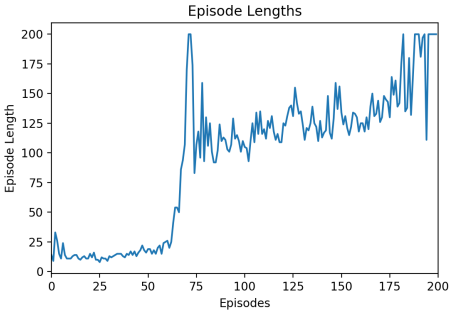

Figure 31: Episode length

Figure 32: Results

Figure 33: No. episode =200 , minibatchsize = 50,reward = -10

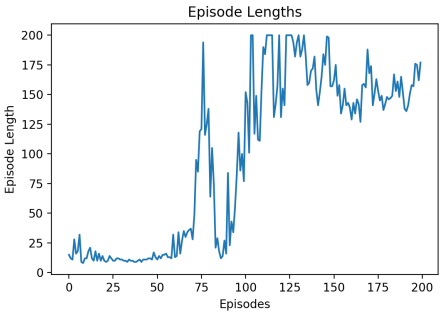

Figure 34: Episode length

Figure 35: Results

Figure 36: No. episode =200 , minibatchsize = 100,reward = -10

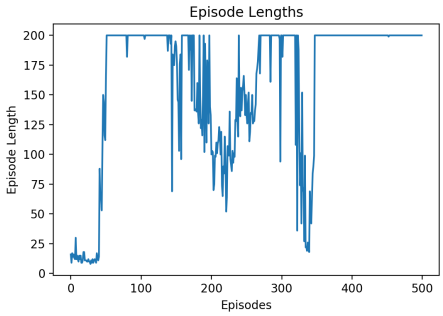

Figure 37: Episode length

Figure 38: Results

Figure 39: No. episode =500 , minibatchsize = 20,reward = -10,hidden layer size= 128,reward = -100

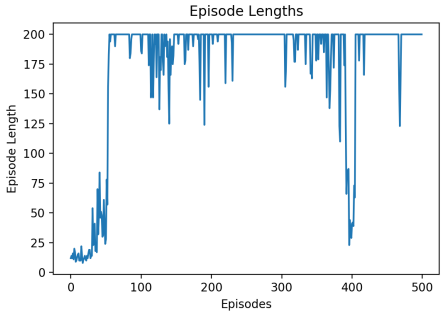

Figure 40: Episode length

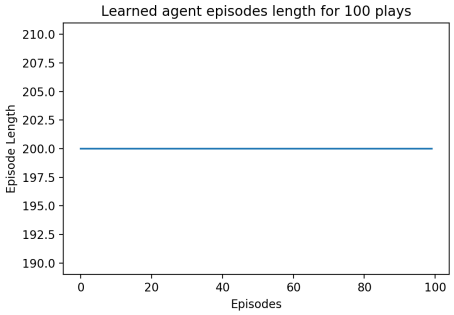

Figure 41: Results

Figure 42: No. episode =500 , minibatchsize = 20,Hidden Layers Size= 64,reward = -100

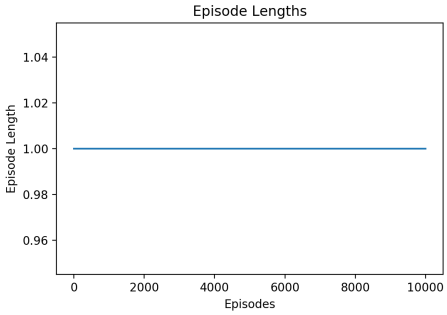

Figure 43: Episode length

Figure 44: Results

Figure 45: No Replay

0.6 NO Replay

Figure 45

0.7 References

1. Sutton, Richard S., Doina Precup, and Satinder Singh. "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning." Articial intelligence 112, no. 1-2 (1999): 181-211.
2.Nirav Bhaskar: Github Link For Envirinment https://github.com/niravnb/Reinforcement-learning/tree/master/DQN
and
3. Viswanathgs: CDN Implementation https://gist.github.com/viswanathgs/abe4a8732a81c66