# CS5691: Reinforcement Learning
## Assignment #1

**Topics:** Q Learning, SARSA Learning **Deadline:** 8 march 2022

**Teammate 1:** (AKASH SAINI) **Roll number:** CS21M003
**Teammate 2:** (RETURAJ BURNWAL) **Roll number:** CS21D406

- **Question** :
  • Implement SARSA and Q-Learning.
  • For each algorithm, run experiments with wind=False and wind=True; two different start states: (0, 4), (3, 6); two values of p (1.0, 0.7);
  and two types of exploration 2 strategies ($\epsilon - greedy$ and softmax), making it 16 different configurations in total.
  • For each of the 16 configurations, determine the best set of hyperparameters ($\epsilon$ in $\epsilon$-greedy exploration, temperature $\beta$ in softmax exploration, learning rate $\alpha$, and discount factor $\gamma$)
  plot the following:
  1. Reward curves and the number of steps to reach the goal in each episode (during the training phase with the best hyperparameters).
  2. Heatmap of the grid with state visit counts, i.e., the number of times each state was visited throughout the training phase.
  3. Heatmap of the grid with Q values after training is complete, and optimal actions for the best policy.
  For each of the algorithm, provide a written description of the policy learnt, explaining the behavior of the agent, and your choice of hyperparameters. This description should also provide information about how the behavior changes with and without the wind, for different levels of stochasticity and for different start states.

- **Solution:**

- Code link: Colab Notebook

- Wandb link: here

- We know that:

  - $\gamma$ is called a discount factor. A lower value of $\gamma$ makes the algorithm behave more greedily and may not lead to a state with higher future rewards. Conversely, a higher value of $\gamma$ makes the algorithm behave such that it leads to higher future rewards.
  - learning rate parameter $\alpha$: A lower value of $\alpha$ will make the algorithm learn less from the new actions performed in that state. On the other hand, a very high value of $\alpha$ will make the algorithm completely ignore the prior knowledge of Q values.
  - *epsilon* in $\epsilon$-greedy method: A very low value of $\epsilon$ leads to a greedy behaviour with respect to Q values and high value of $\epsilon$ lead to more exploration.

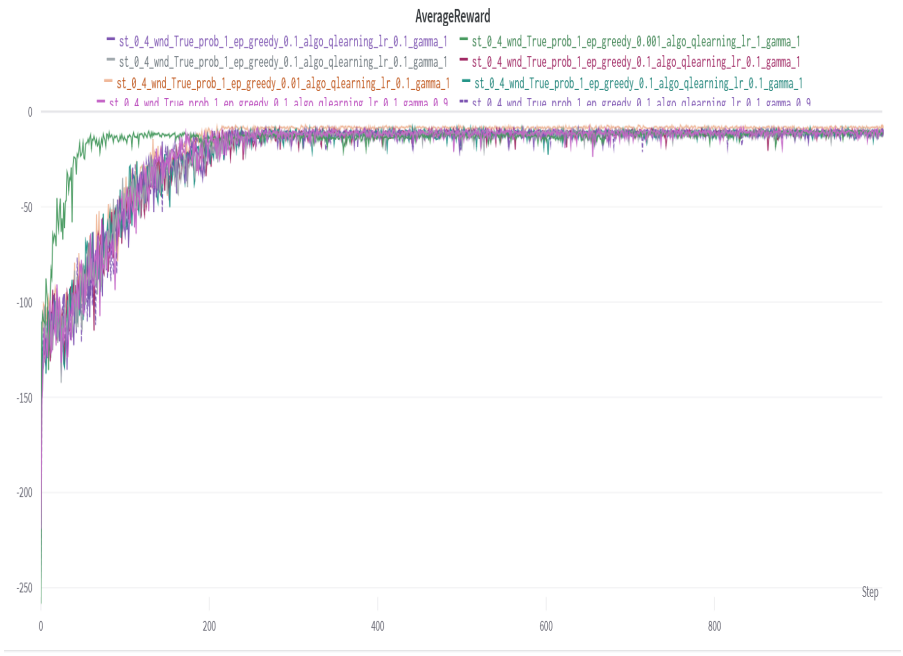
- $\beta$ in softmax method: A very low value of $\beta$ will lead greedy behaviour with respect to Q values and a high value of beta will case the algorithm to explore more. low value of $\beta(< 1)$ indicates model is more accurate in prediction . high value of $\beta(> 1)$ indicates model is less accurate in prediction

- For each configuration of SARSA and Q-Learning we performed the following steps:
  - To search for best hyper-parameters, we searched the combination of following sets of hyper-parameters.
    - For $\epsilon$-greedy method, set of $\epsilon = \{0.001, 0.01, 0.1\}$.
    - For softmax method, set of $\beta = \{0.01, 0.1, 1.0, 2.0\}$
    - $\gamma = \{0.7, 0.8, 0.9, 1.0\}$
    - learning rate $\alpha = \{0.001, 0.01, 0.1, 1.0\}$
  - For each set of hyper-parameters we ran about 1000 episodes and averaged it over 20 independent runs.
  - We choose those hyper-parameters which gave maximum total reward in the end.

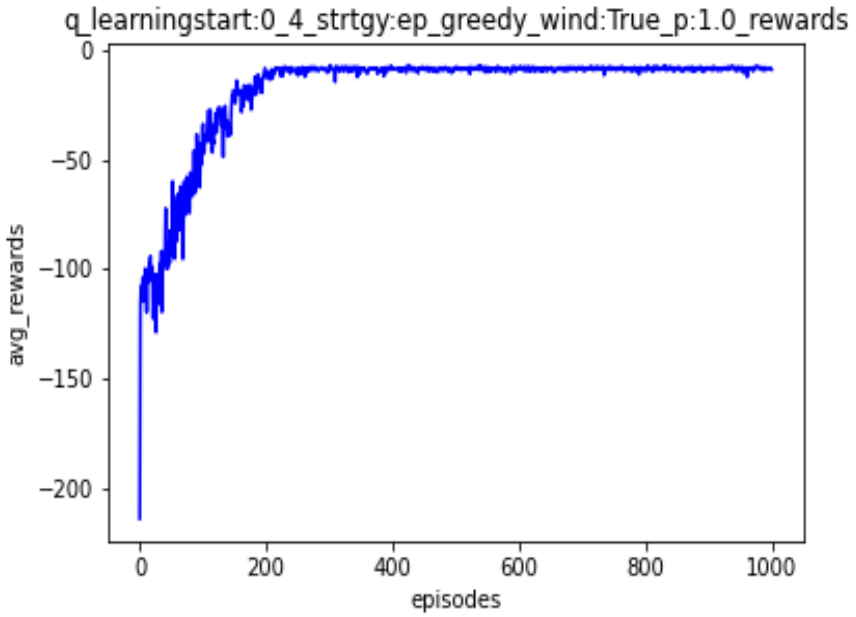
- **Q Learning-**

**Solution:**

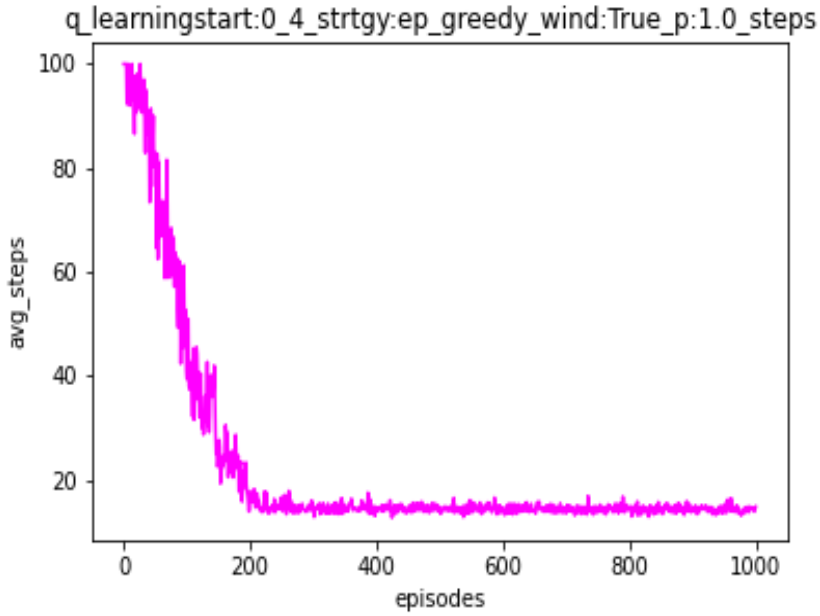
1. **strategy=$\epsilon$-greedy, start_state=(0,4), wind=True, p=1.0**
   - From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\epsilon = 0.01$ gave better performance. Following plot shows some of the best performing hyper-parameters.



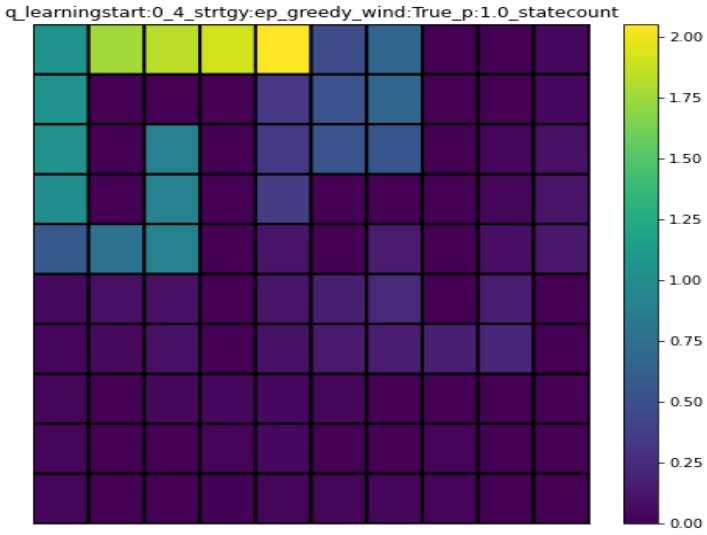
– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach the upper left corner goal.
– Total reward after 1000 episodes averaged over 20 runs = -8.15 , Reward curve:



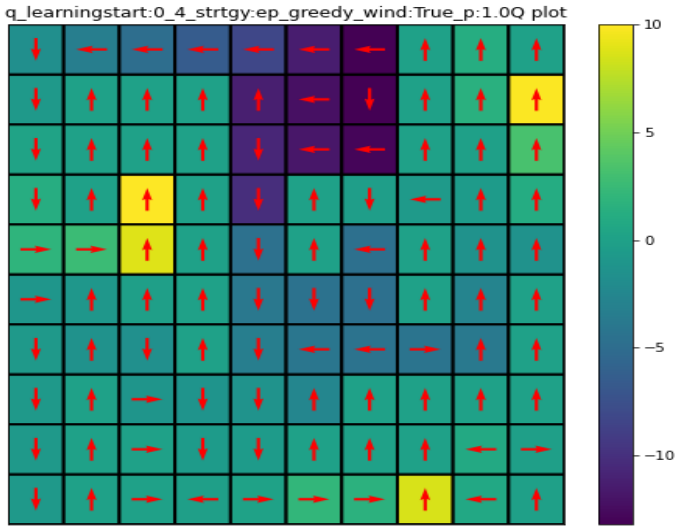

– Number steps to reach goal after 1000 episodes average over 20 runs = 14.15 , Step curve:



– Heatmap of the grid with state visit count:



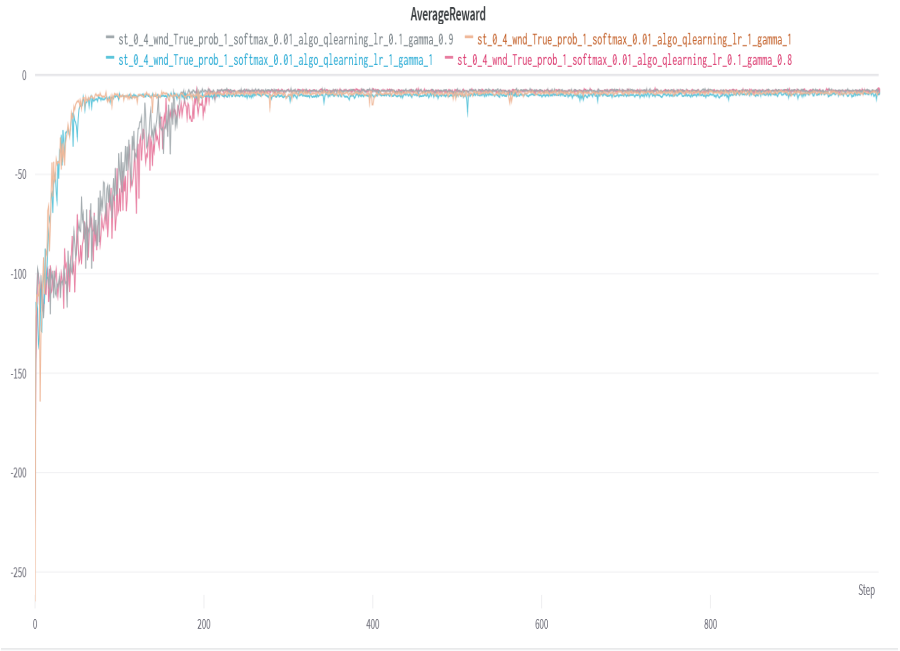

– Heatmap of the grid with Q values after training is complete:




2. **strategy=softmax , start_state=(0,4), wind=True, p=1.0**

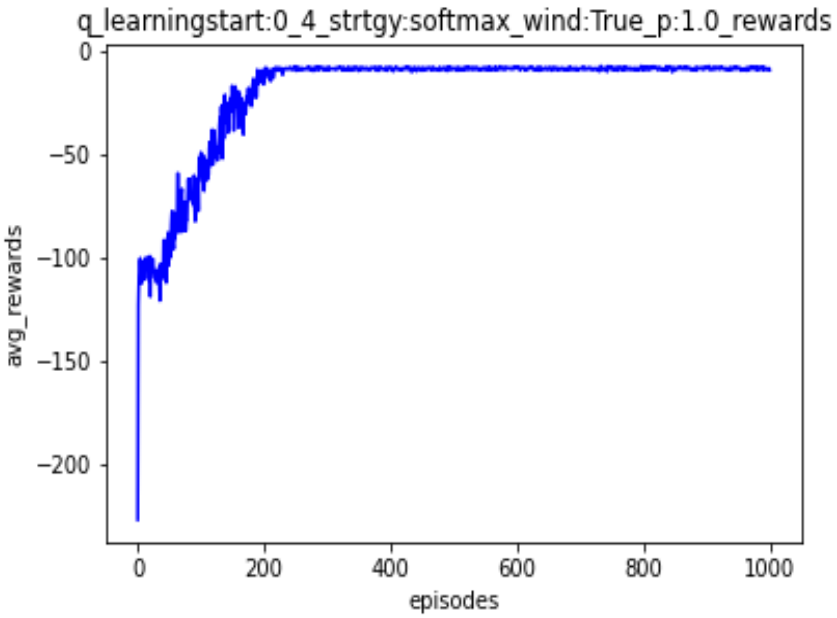
– From the experiments we performed we found that $\gamma = 0.8$, $\alpha = 0.1$, $\beta = 0.01$ gave better performance.Following plot shows some of the best performing

hyper-parameters



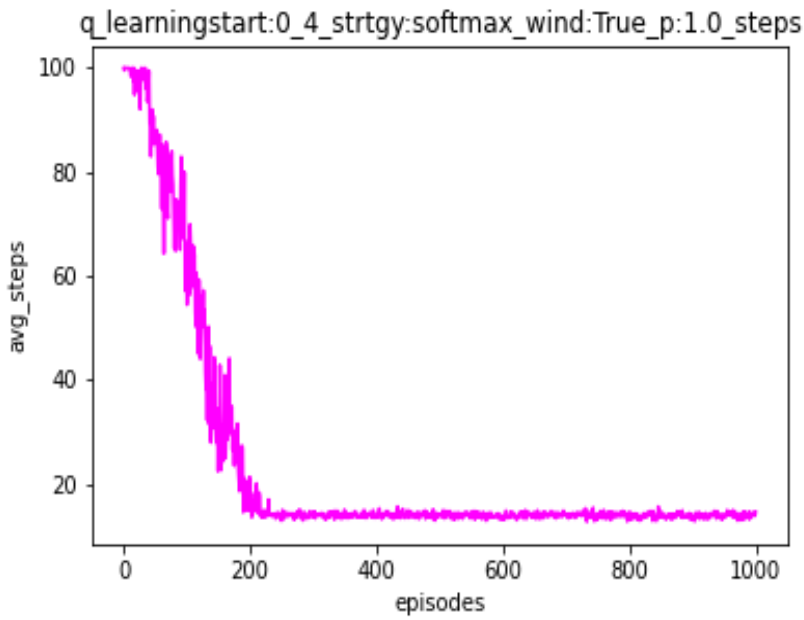

– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach the upper left corner goal.

– Total reward after 1000 episodes averaged over 20 runs = -7.65 , Reward curve:

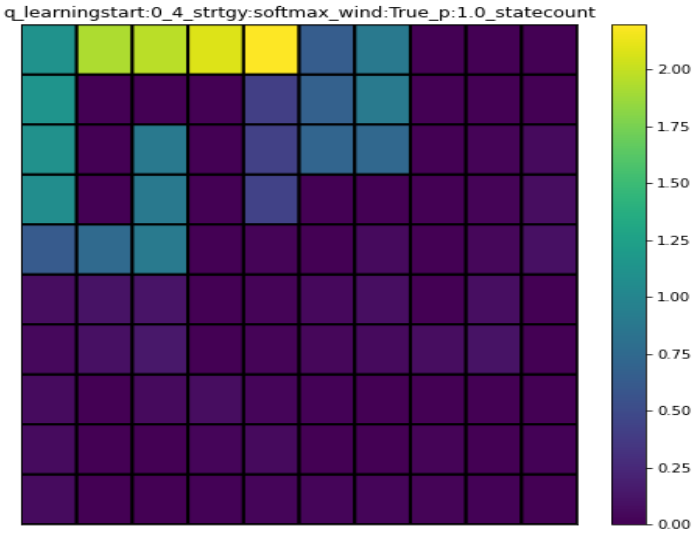



– Number steps to reach goal after 1000 episodes average over 20 runs = 13.65

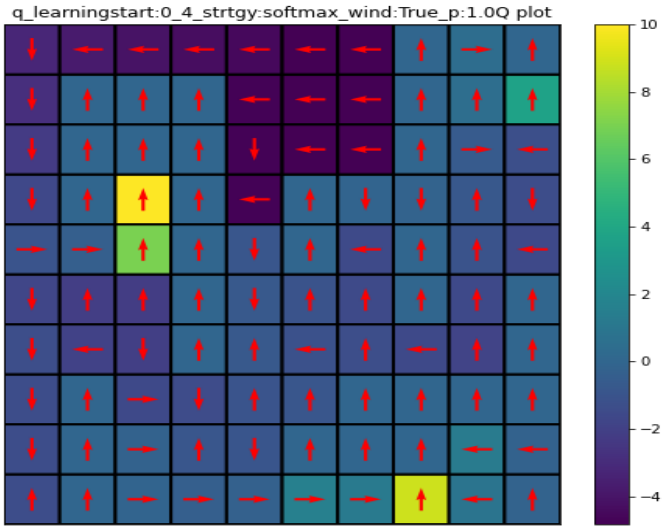
, Step curve:




– Heatmap of the grid with state visit count:

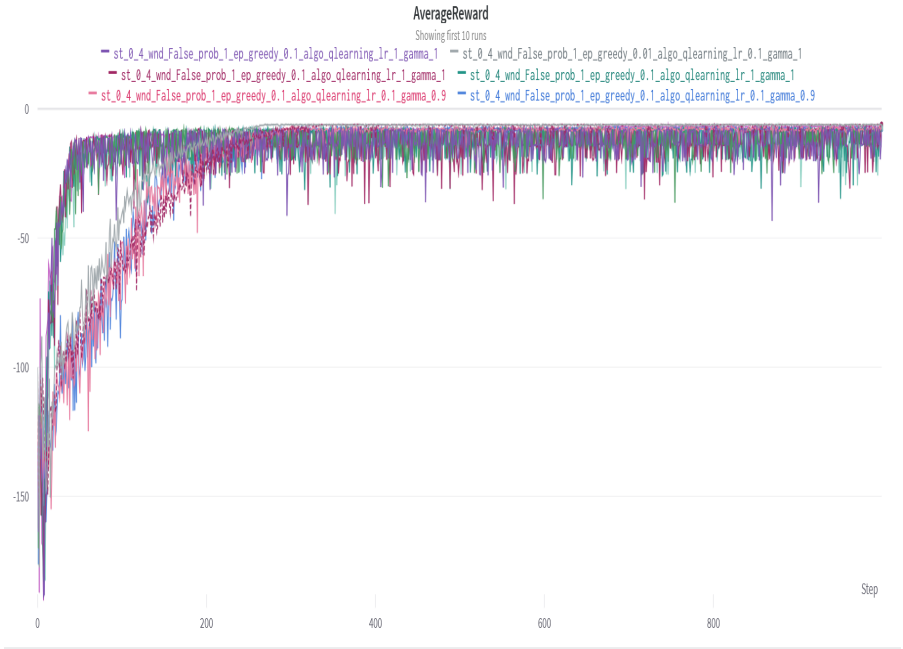



– Heatmap of the grid with Q values after training is complete:




3. **strategy=$\epsilon$-greedy, start_state=(0,4), wind=False, p=1.0**

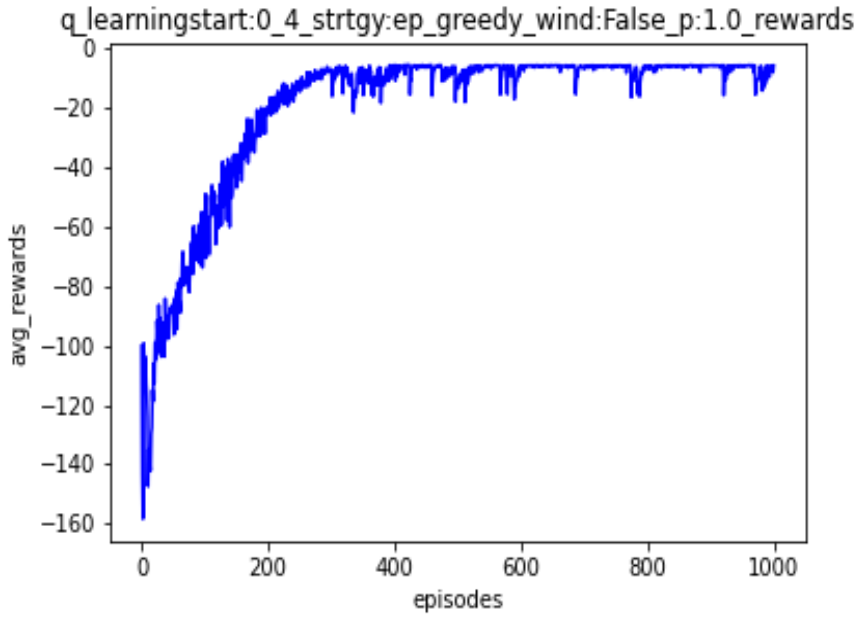
   – From the experiments we performed we found that $\gamma = 0.8$, $\alpha = 0.1$, $\epsilon = 0.01$ gave better performance.Following plot shows some of the best performing hyper-parameters



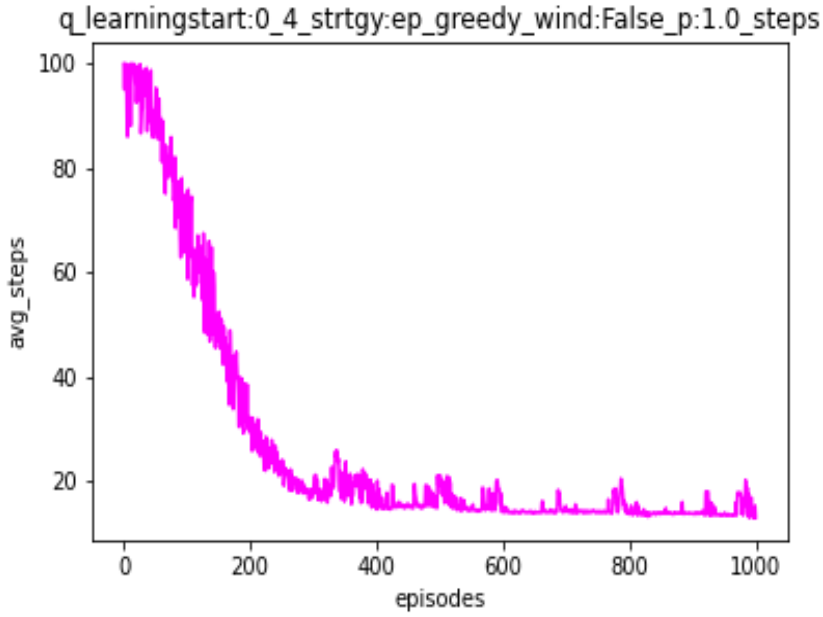

   – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach the both upper left corner goal and even lower right corner goal.

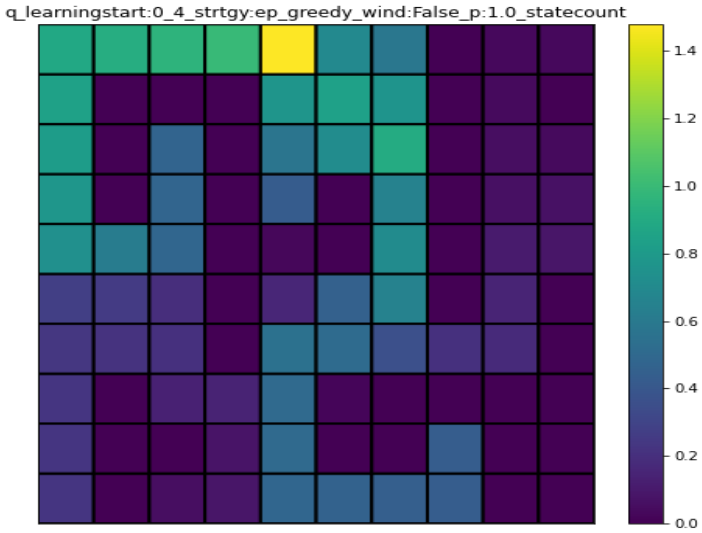
– Total reward after 1000 episodes averaged over 20 runs = -6 , Reward curve:



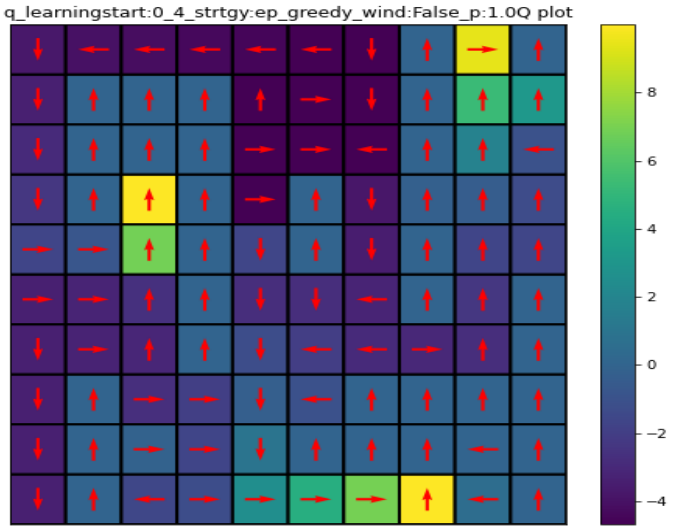

– Number steps to reach goal after 1000 episodes average over 20 runs = 17 , Step curve:



– Heatmap of the grid with state visit count:



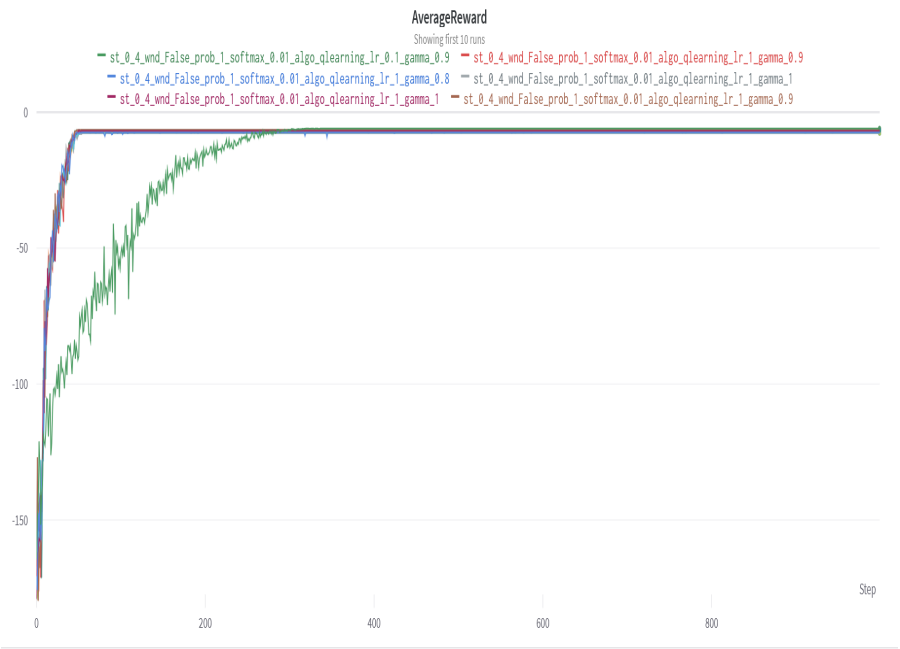

– Heatmap of the grid with Q values after training is complete:



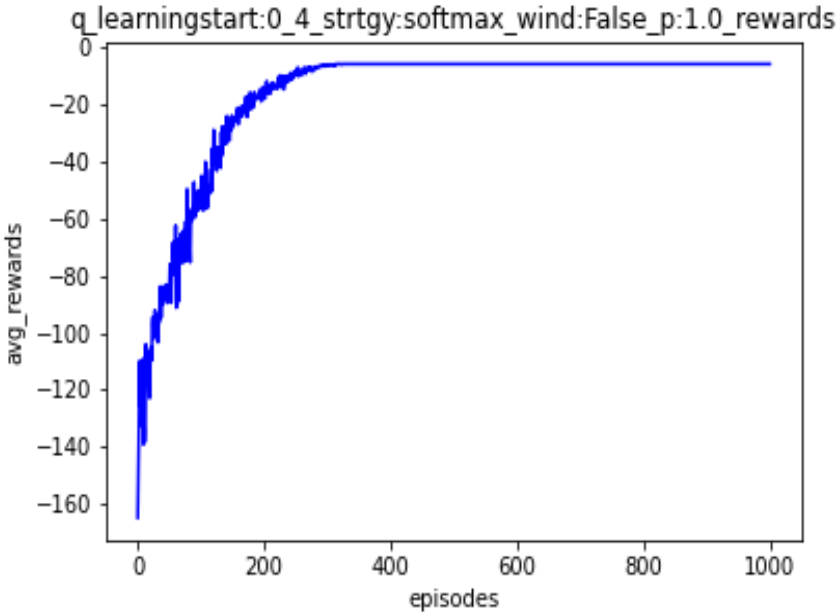

4. **strategy=softmax , start_state=(0,4), wind=False, p=1.0**

– From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\beta = 0.01$ gave better performance.Following plot shows some of the best performing

hyper-parameters.

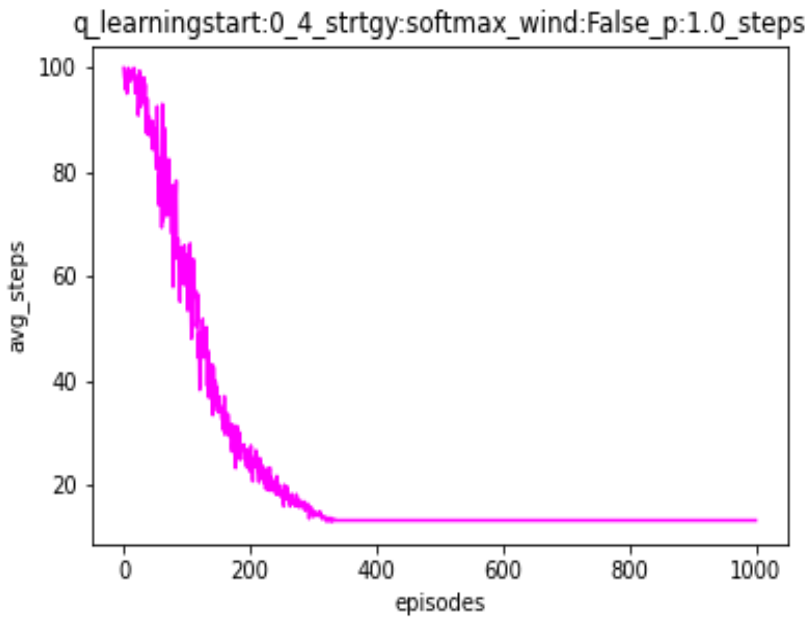



- Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach both upper left corner goal and lower right corner goal.
- Total reward after 1000 episodes averaged over 20 runs = -6 , Reward curve:

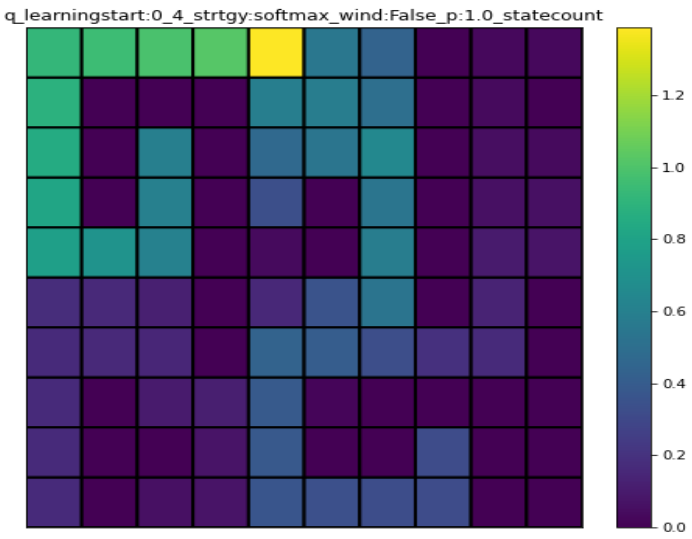



- Number steps to reach goal after 1000 episodes average over 20 runs = 13.25

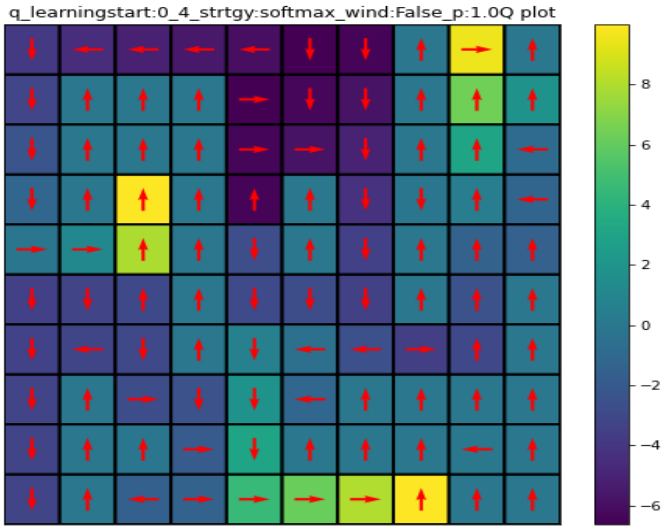
, Step curve:




– Heatmap of the grid with state visit count:

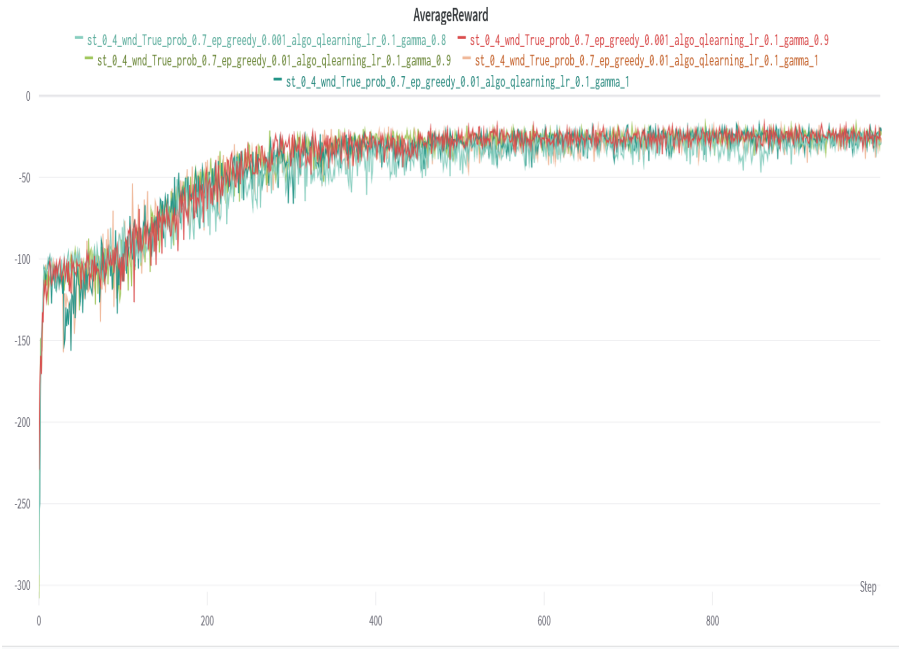



– Heatmap of the grid with Q values after training is complete:




5. **strategy=$\epsilon$-greedy, start_state=(0,4), wind=True, p=0.7**

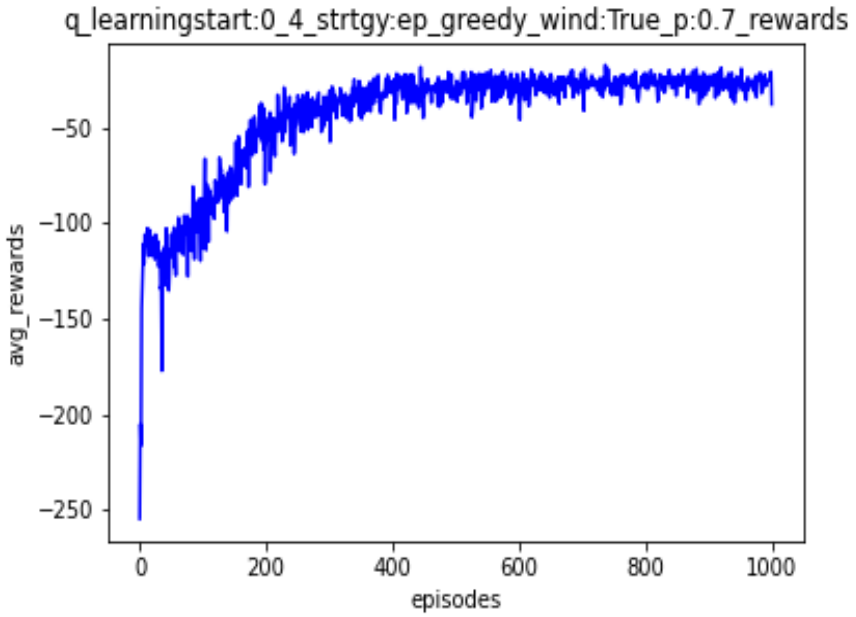
   – From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\epsilon = 0.01$ gave better performance.Following plot shows some of the best performing hyper-parameters.



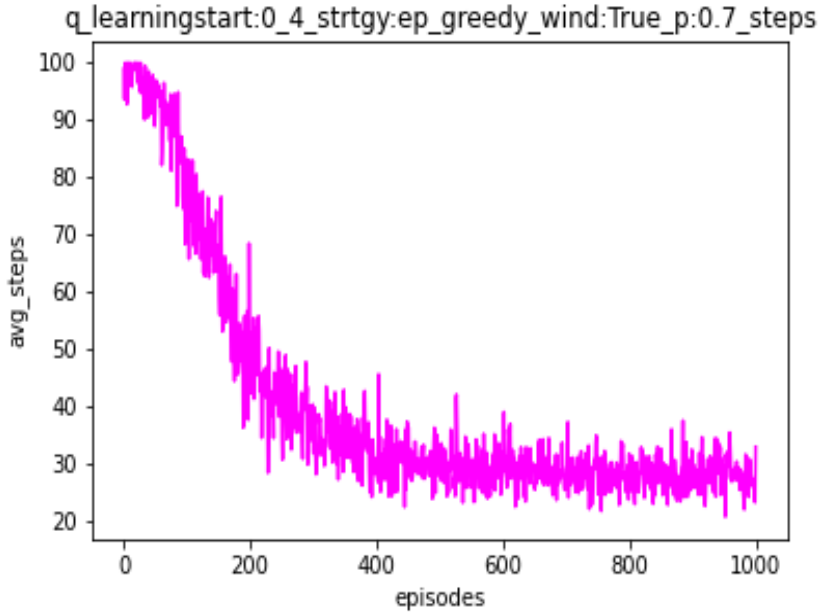

   – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach the upper left corner goal. And some times because of stochastic environment agent has also learnt to reach upper right corner goal.

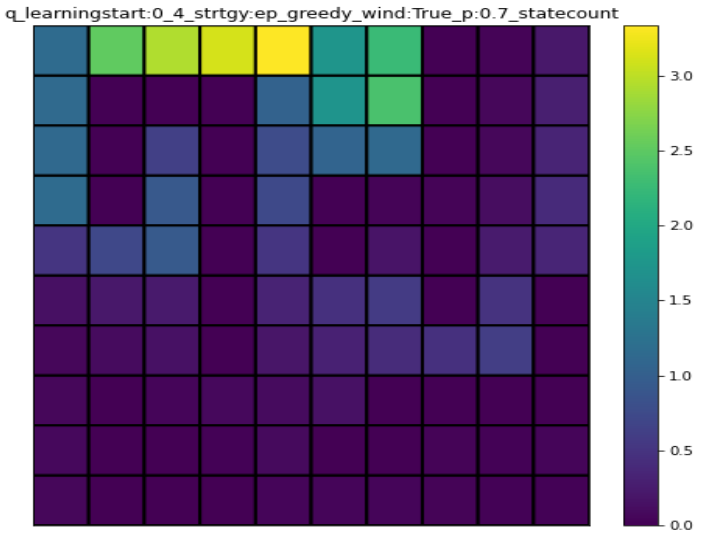
- Total reward after 1000 episodes averaged over 20 runs = -21.05 , Reward curve:



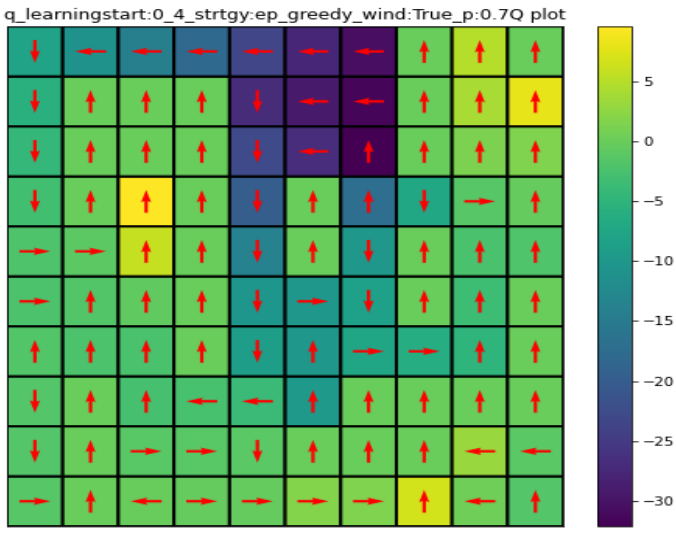

- Number steps to reach goal after 1000 episodes average over 20 runs = 26.05 , Step curve:



– Heatmap of the grid with state visit count:



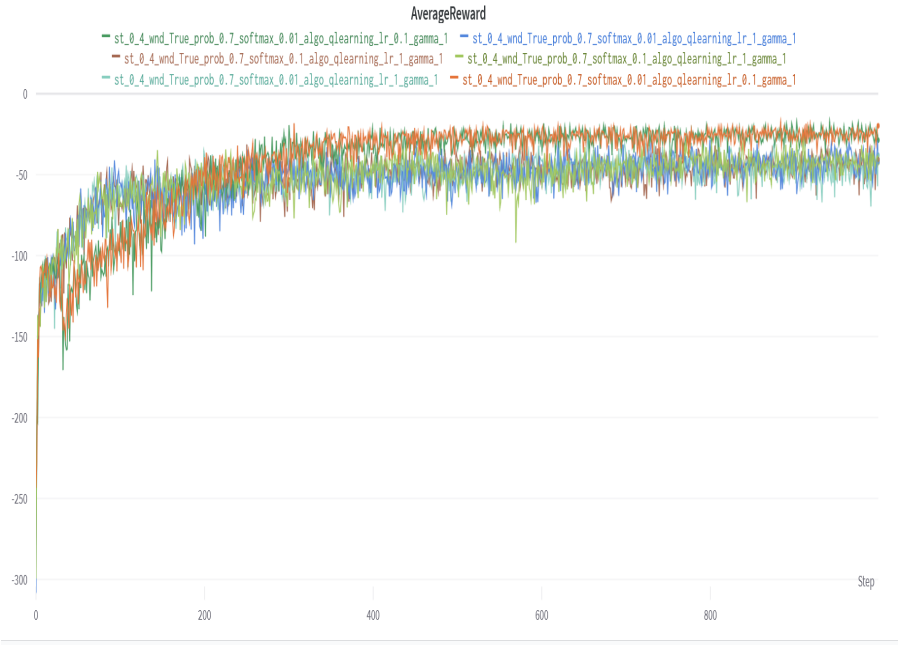

– Heatmap of the grid with Q values after training is complete:




6. **strategy=softmax , start_state=(0,4), wind=True, p=0.7**

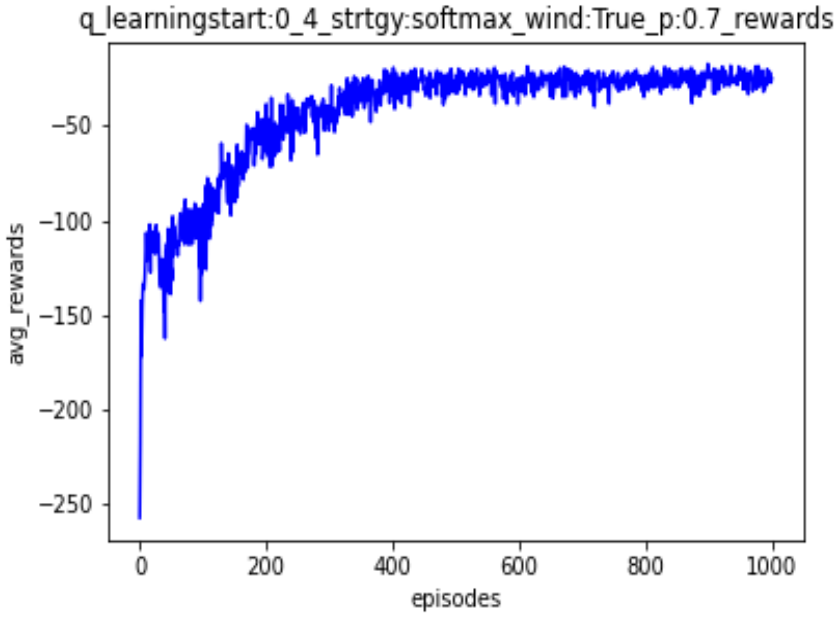
   – From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\beta = 0.01$ gave better performance.Following plot shows some of the best performing

hyper-parameters.

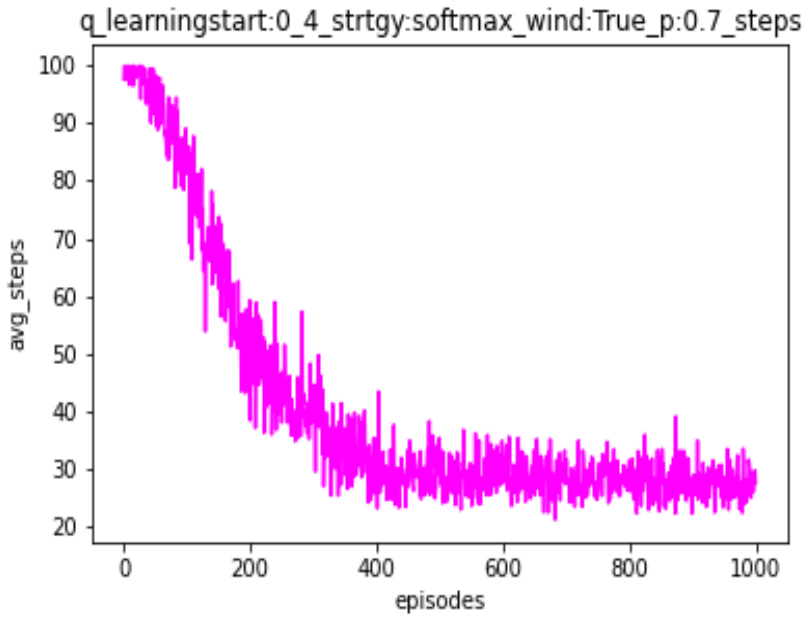



– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach the upper left corner goal. And some times because of stochastic environment agent has also learnt to reach upper right corner goal.

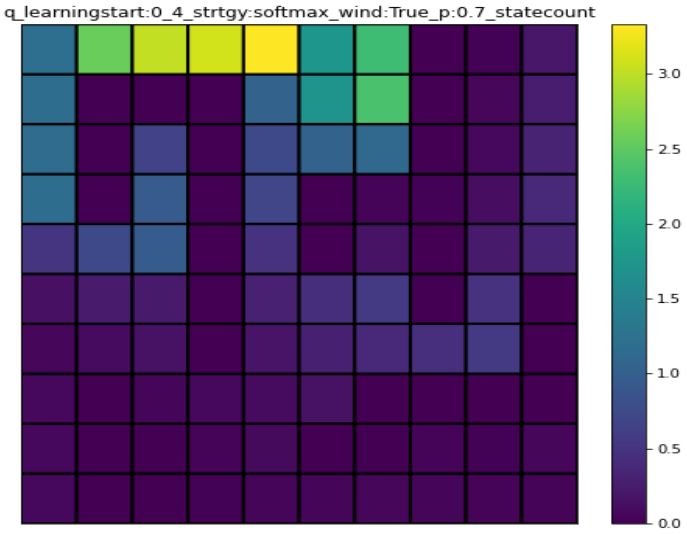
– Total reward after 1000 episodes averaged over 20 runs = -19.8 , Reward curve:



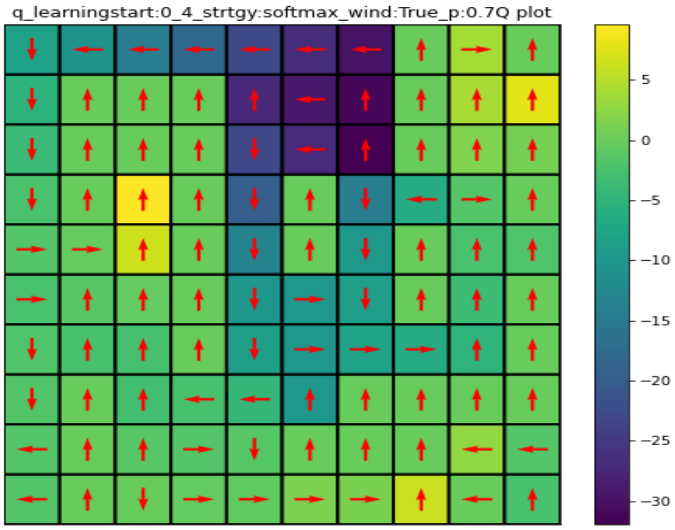
– Number steps to reach goal after 1000 episodes average over 20 runs = 22.55, Step curve:




– Heatmap of the grid with state visit count:




– Heatmap of the grid with Q values after training is complete:




7. **strategy=$\epsilon$-greedy, start_state=(0,4), wind=False, p=0.7**

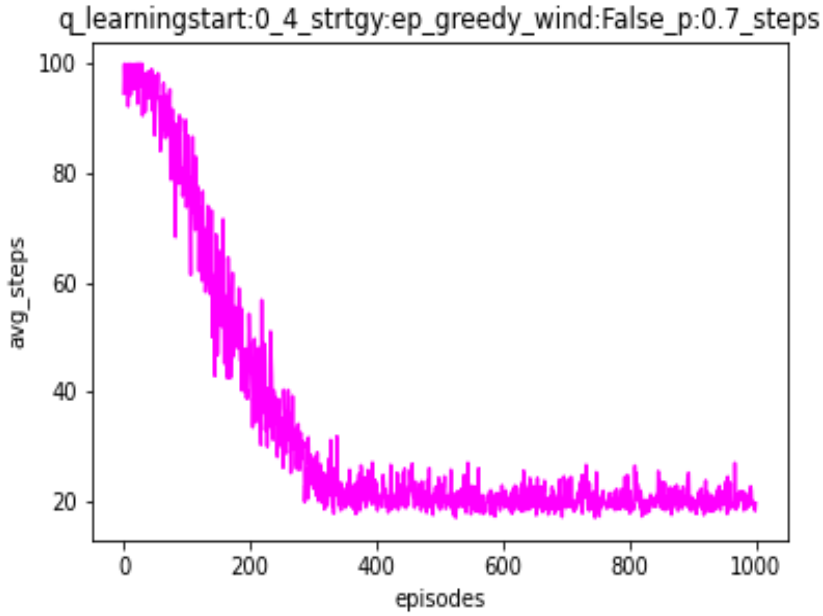
   – From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\epsilon = 0.01$ gave better performance.Following plot shows some of the best performing hyper-parameters.


   – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

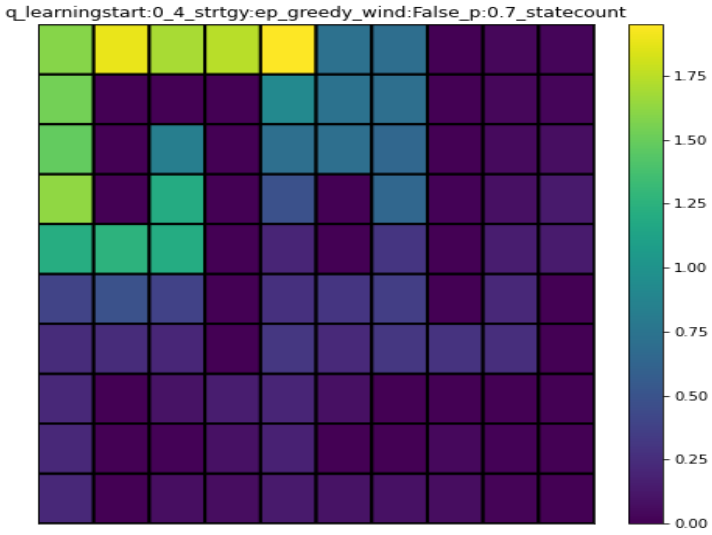
we notice that agent has learnt to reach both upper left corner goal. And due to stochastic action agent has also learnt to reach upper right goal too.

– Total reward after 1000 episodes averaged over 20 runs = -14.75 , Reward curve:

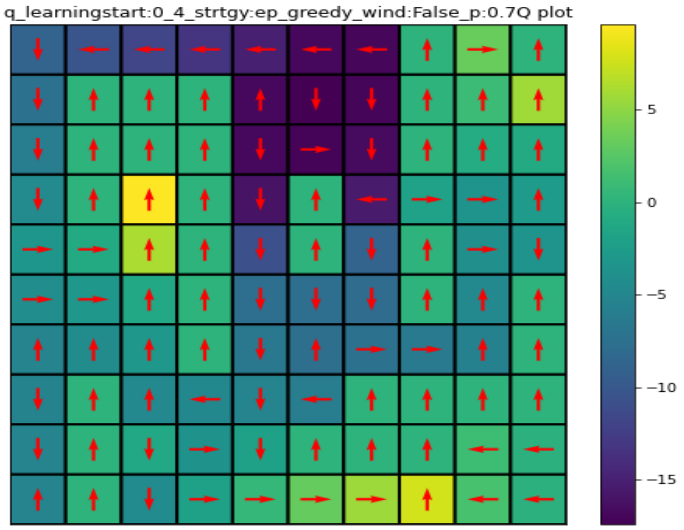

– Number steps to reach goal after 1000 episodes average over 20 runs = 18.75 , Step curve:



– Heatmap of the grid with state visit count:



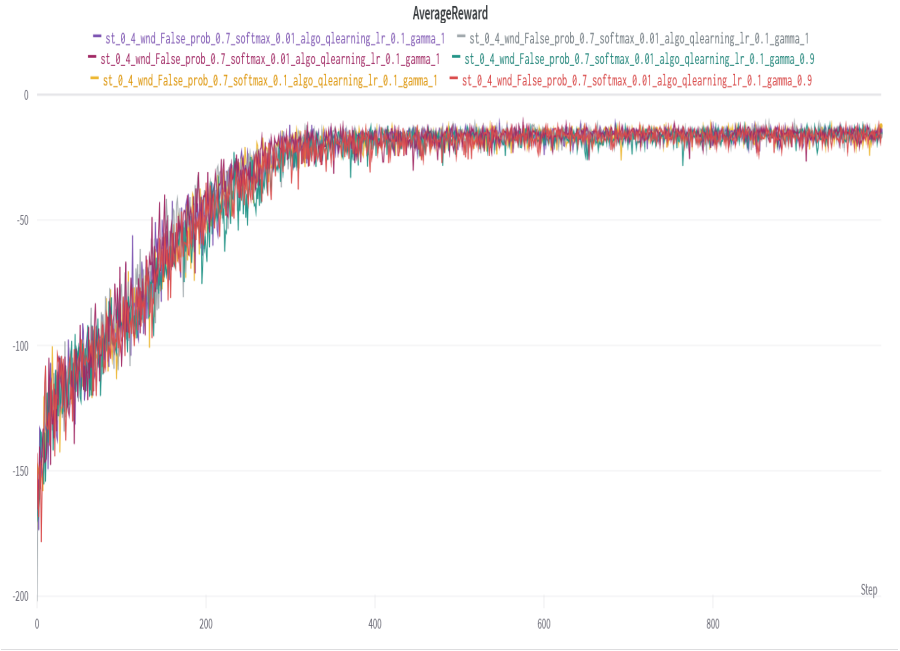

– Heatmap of the grid with Q values after training is complete:




8. **strategy=softmax , start_state=(0,4), wind=False, p=0.7**

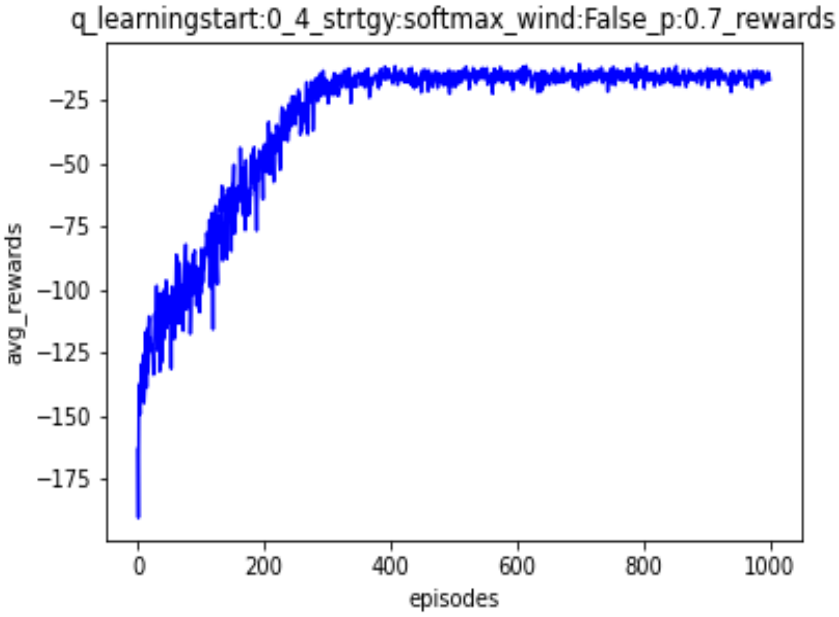
– From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\beta = 0.1$ gave better performance.Following plot shows some of the best performing

hyper-parameters.

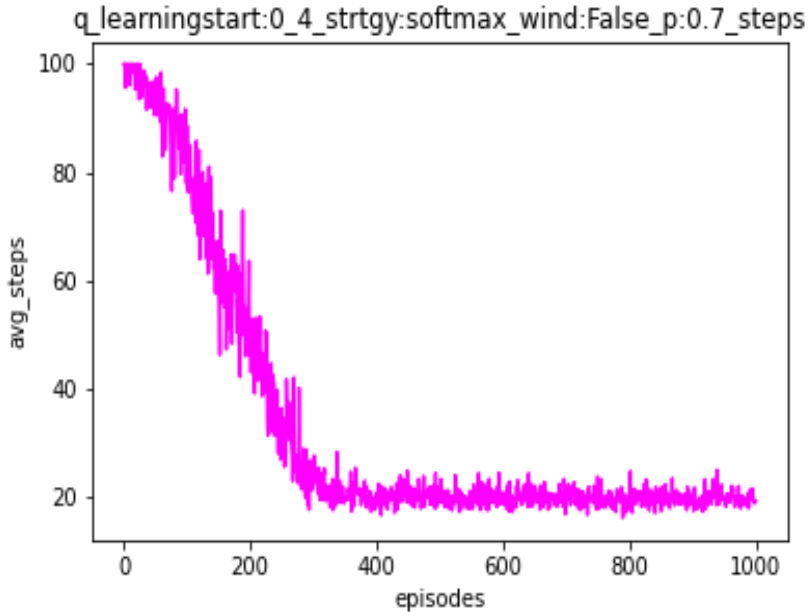



– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach both upper left corner goal. And due to stochastic action agent has also learnt to reach upper right goal too.

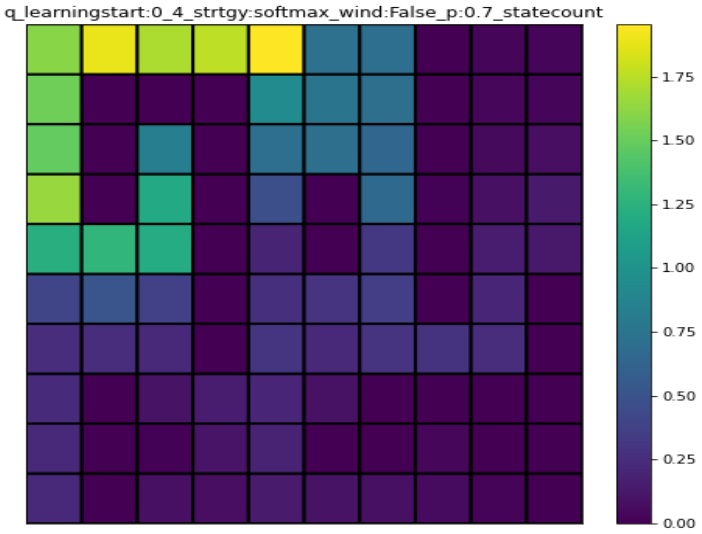
– Total reward after 1000 episodes averaged over 20 runs = -12.5 , Reward curve:



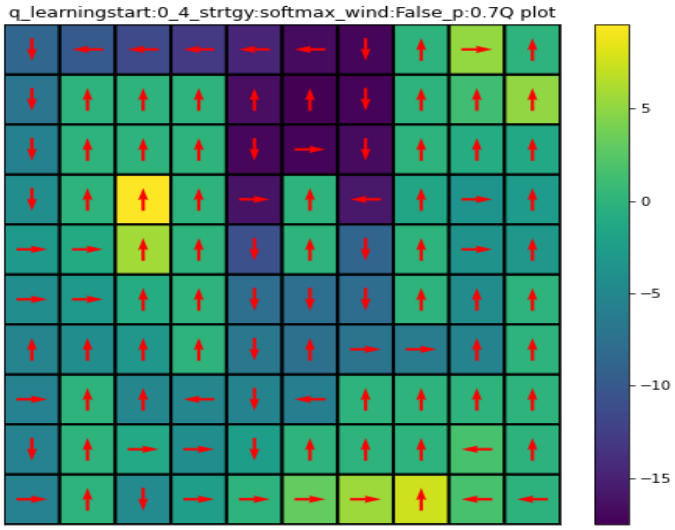
– Number steps to reach goal after 1000 episodes average over 20 runs = 18, Step curve:




– Heatmap of the grid with state visit count:

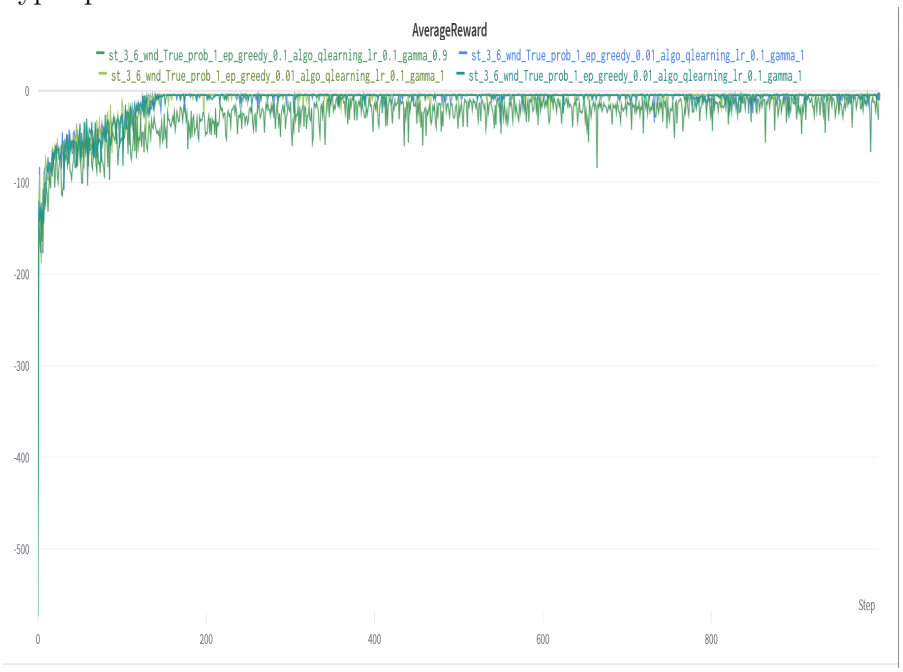



– Heatmap of the grid with Q values after training is complete:




9. **strategy=$\epsilon$-greedy, start_state=(3,6), wind=True, p=1.0**

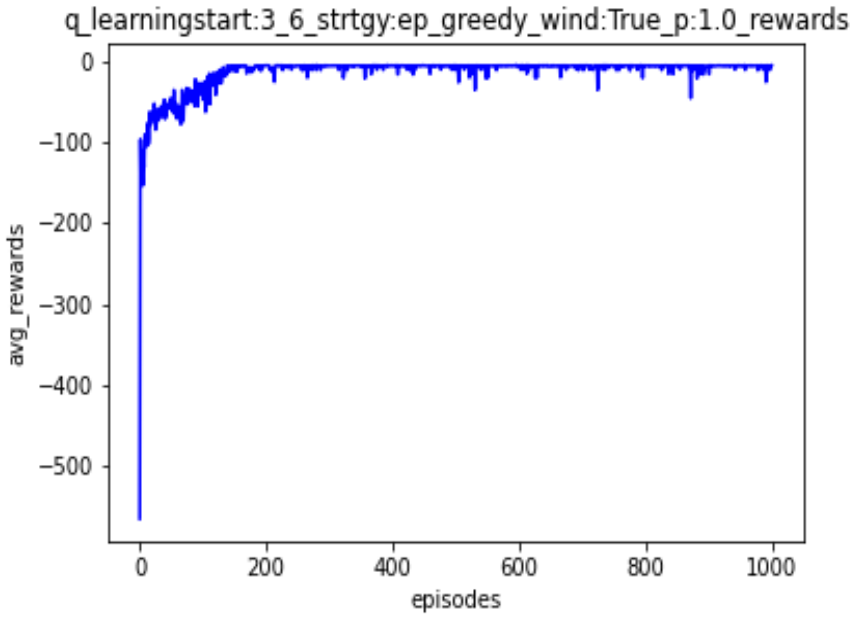
   – From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\epsilon = 0.01$ gave better performance.Following plot shows some of the best performing hyper-parameters.



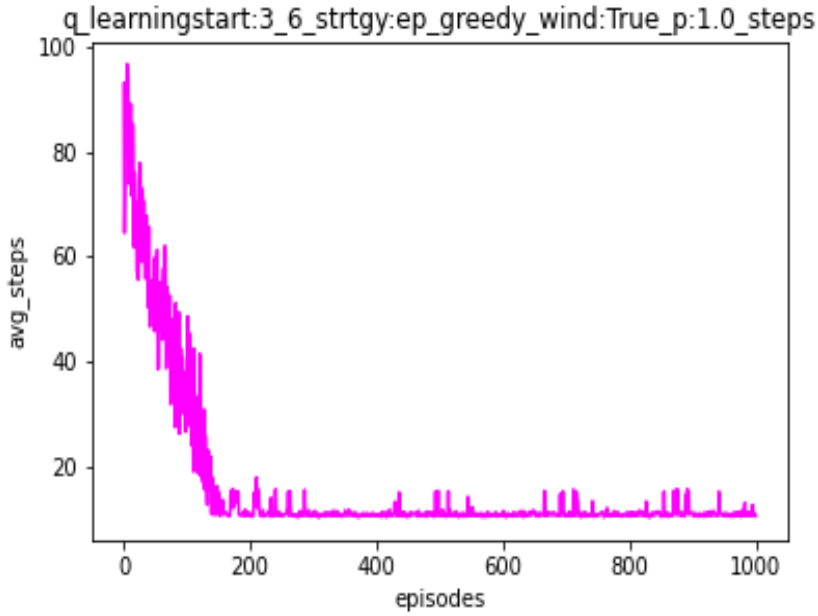

   – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach both upper right corner goal may be because wind has moved agent in a rightward direction.

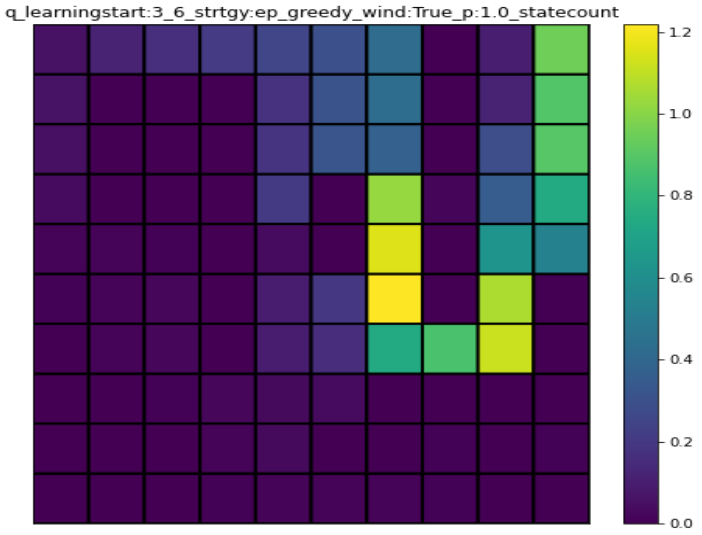
– Total reward after 1000 episodes averaged over 20 runs = -4.35 , Reward curve:



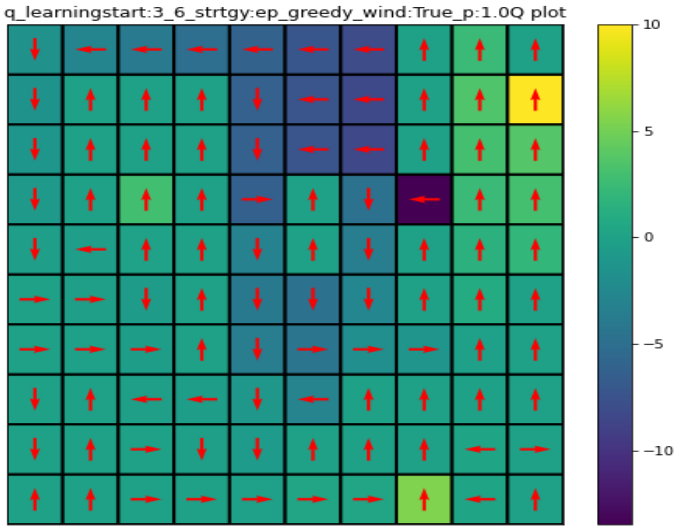

– Number steps to reach goal after 1000 episodes average over 20 runs = 10.85 , Step curve:



– Heatmap of the grid with state visit count:



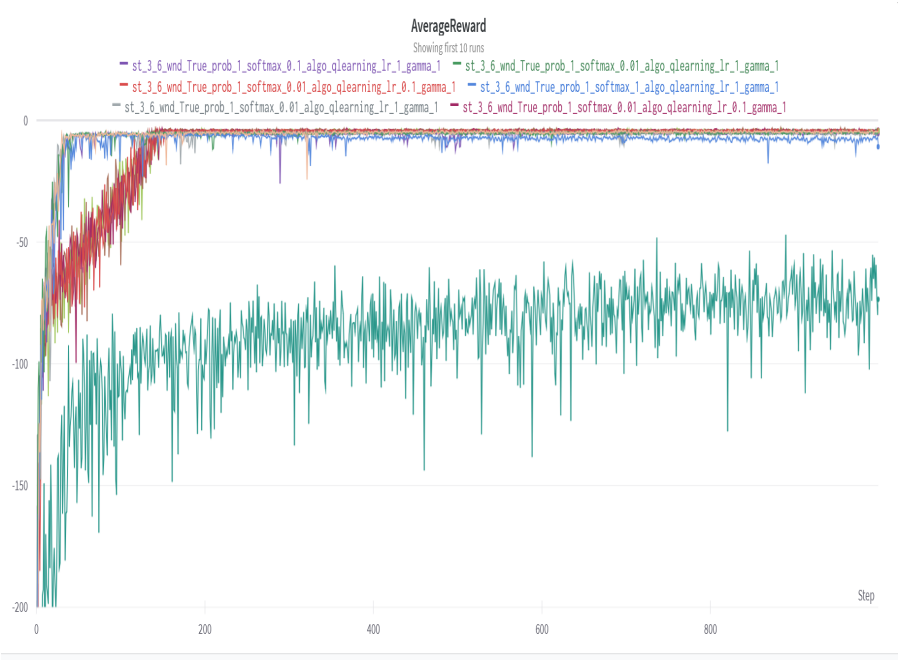

– Heatmap of the grid with Q values after training is complete:




10. **strategy=softmax , start_state=(3,6), wind=True, p=1.0**

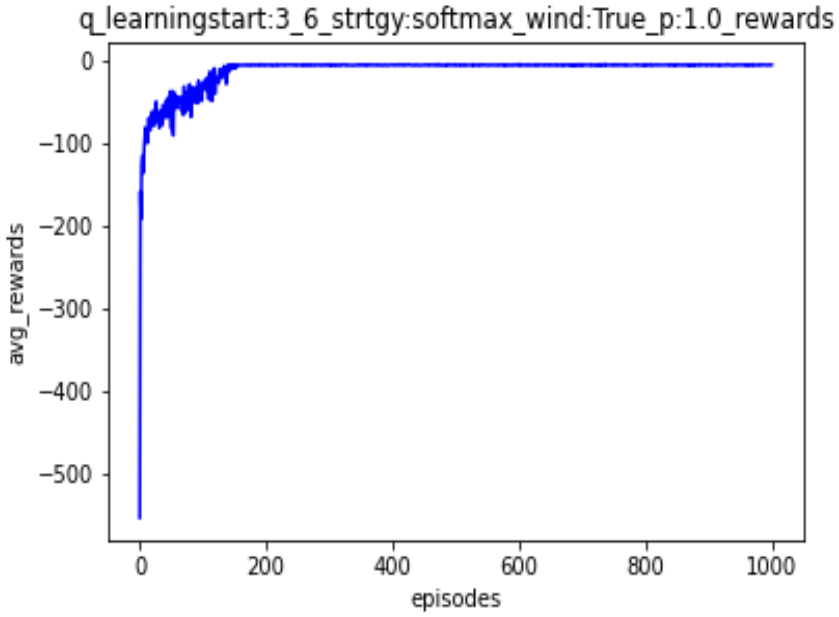
– From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\beta = 0.01$ gave better performance.Following plot shows some of the best performing

hyper-parameters.

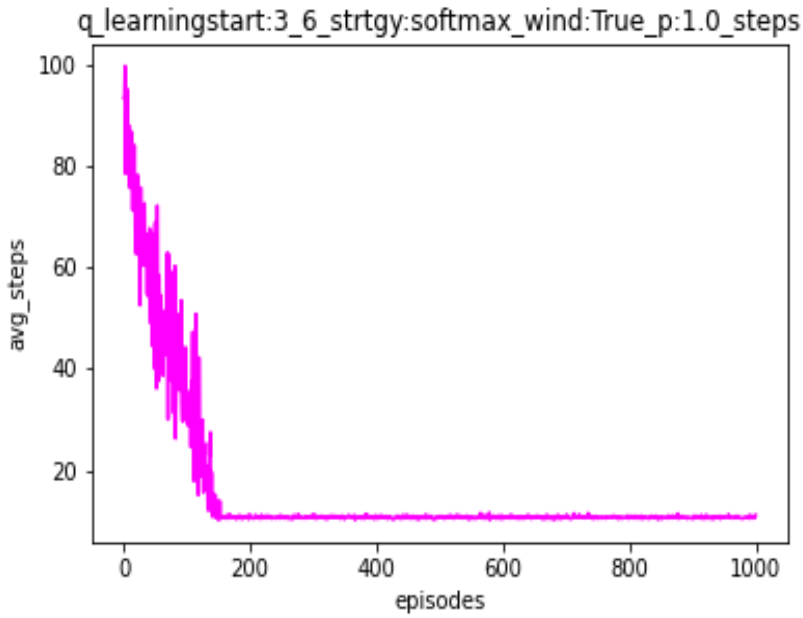



– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach both upper right corner goal may be because wind has moved agent in a rightward direction.

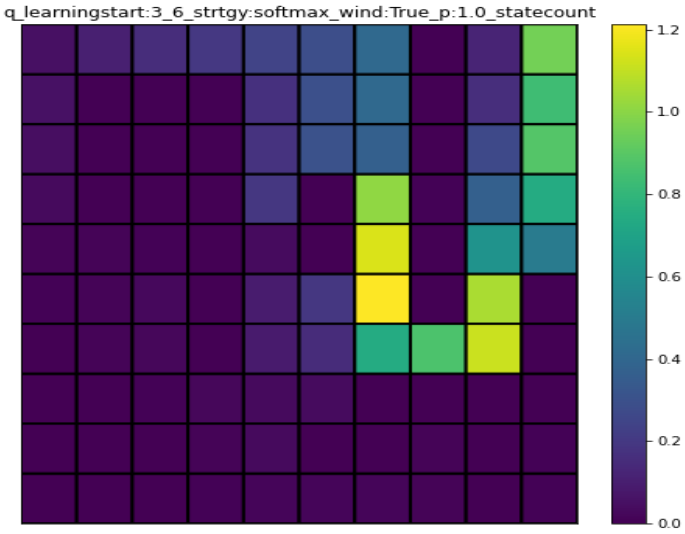
– Total reward after 1000 episodes averaged over 20 runs = -4.05 , Reward curve:



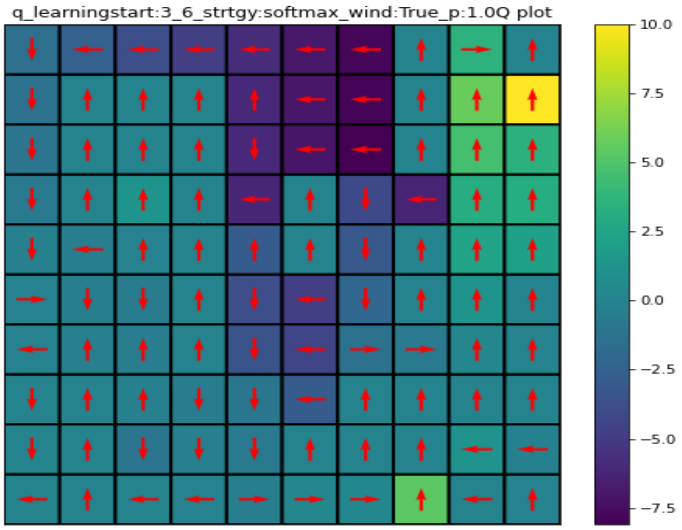
– Number steps to reach goal after 1000 episodes average over 20 runs = 10.3, Step curve:




– Heatmap of the grid with state visit count:

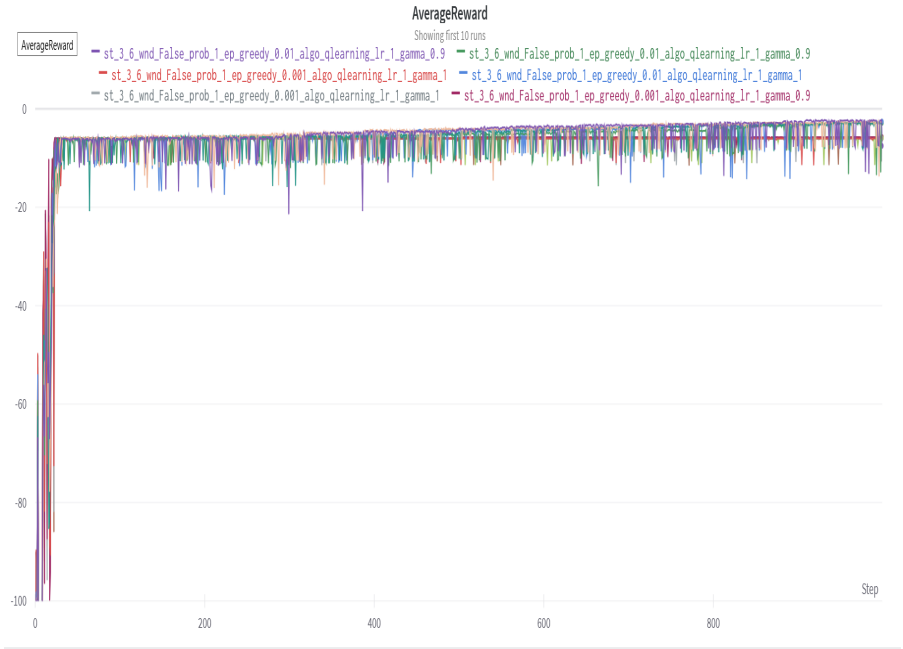



– Heatmap of the grid with Q values after training is complete:




11. **strategy=$\epsilon$-greedy, start_state=(3,6), wind=False, p=1.0**

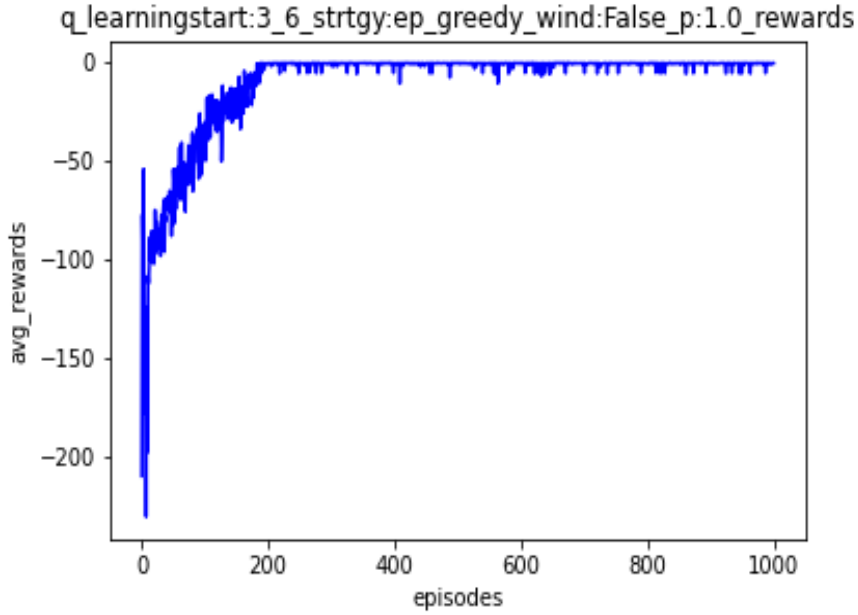
    – From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\epsilon = 0.01$ gave better performance.Following plot shows some of the best performing hyper-parameters.



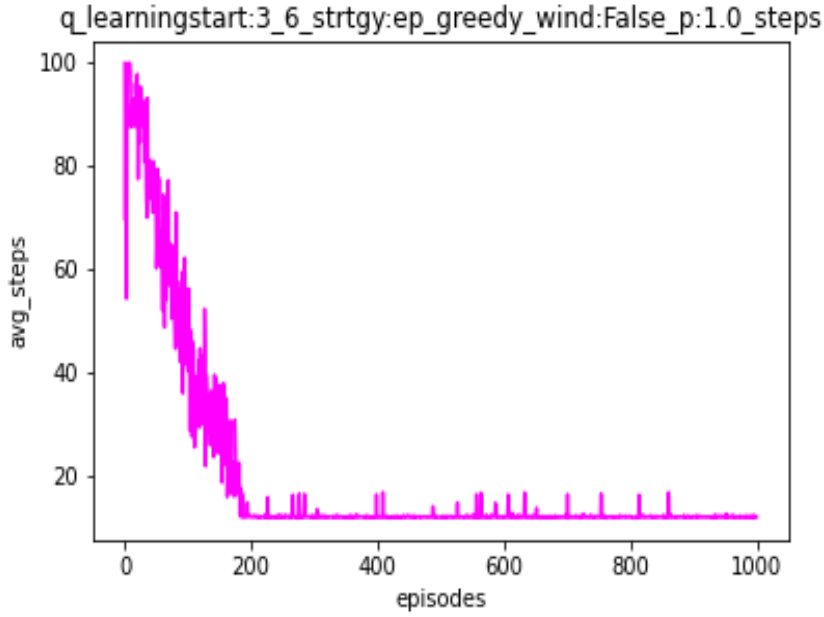

    – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach lower right corner goal.

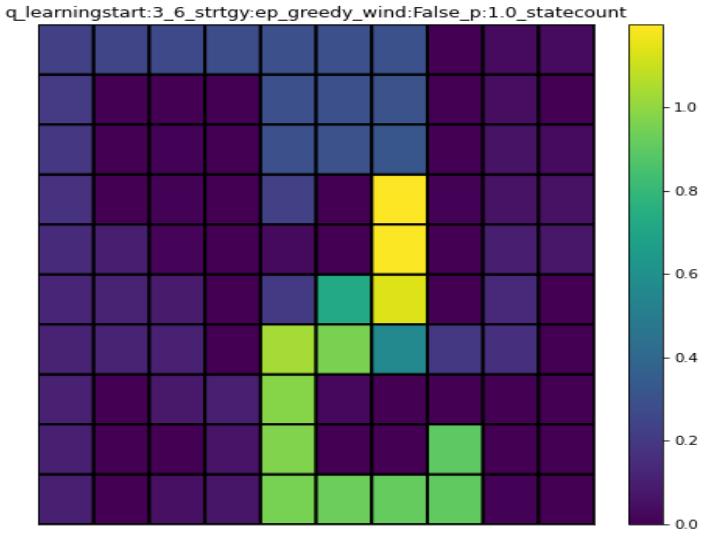
– Total reward after 1000 episodes averaged over 20 runs = -1 , Reward curve:



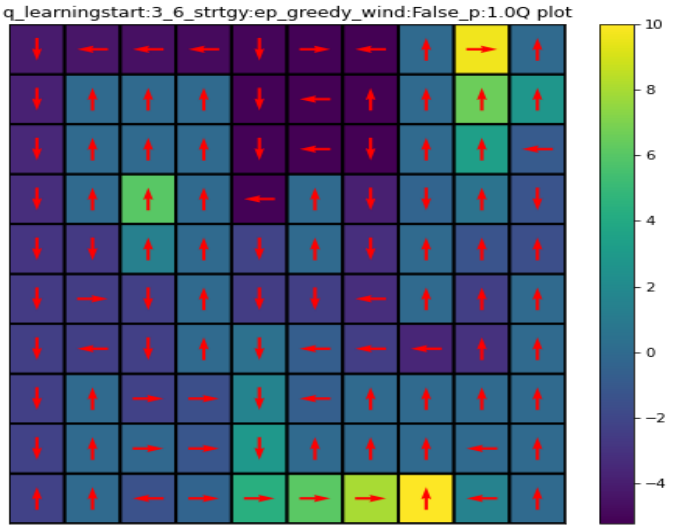

– Number steps to reach goal after 1000 episodes average over 20 runs = 12 , Step curve:




– Heatmap of the grid with state visit count:

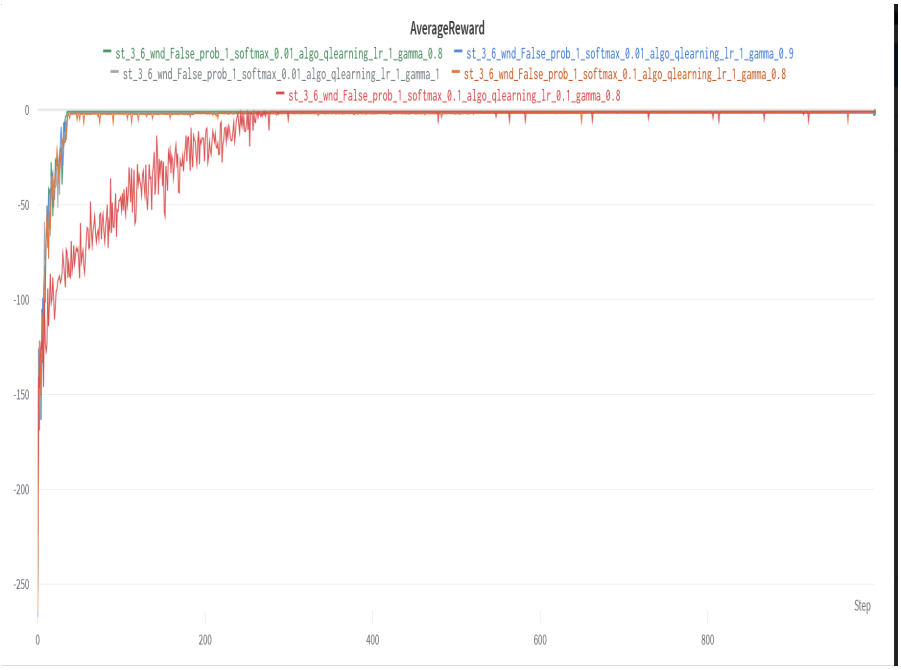



– Heatmap of the grid with Q values after training is complete:

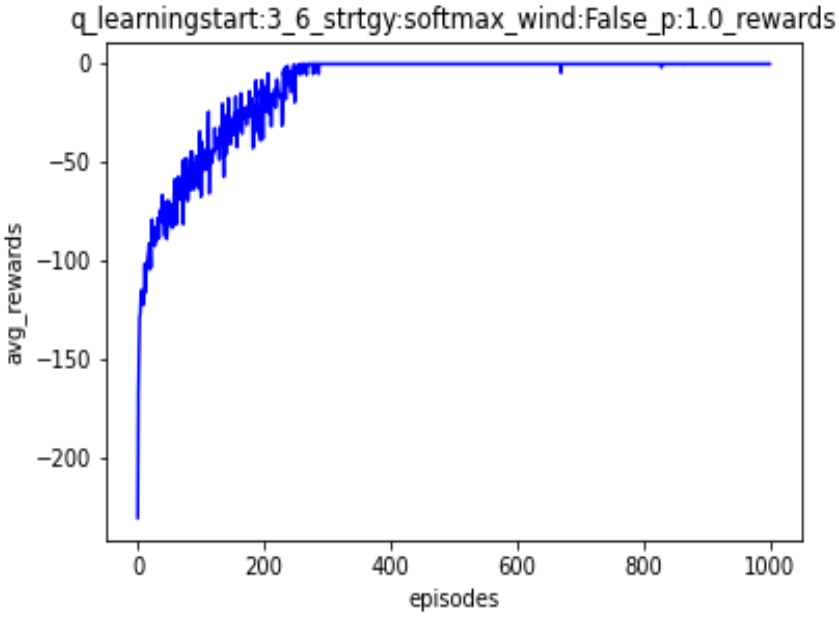



12. **strategy=softmax , start_state=(3,6), wind= False, p=1.0**

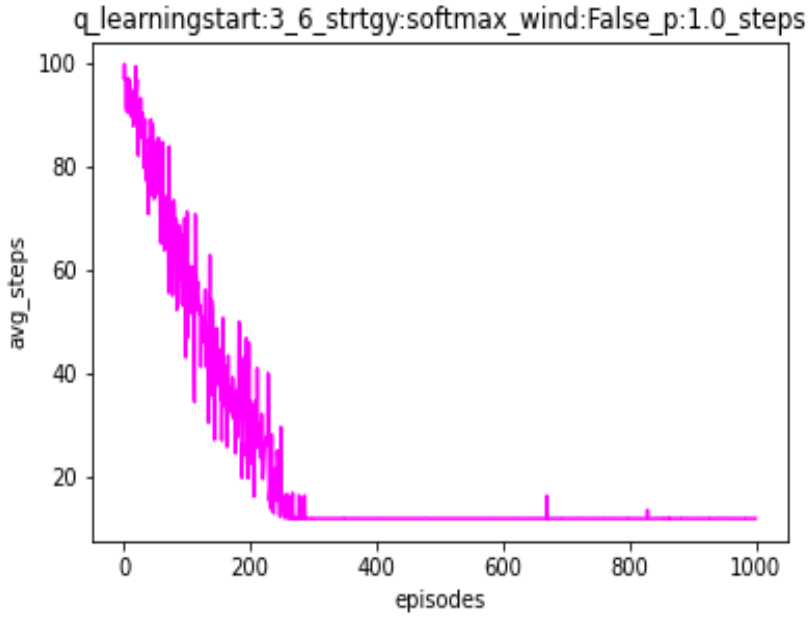
– From the experiments we performed we found that $\gamma = 0.8$, $\alpha = 0.1$, $\beta = 0.1$ gave better performance. Following plot shows some of the best performing hyper-parameters.



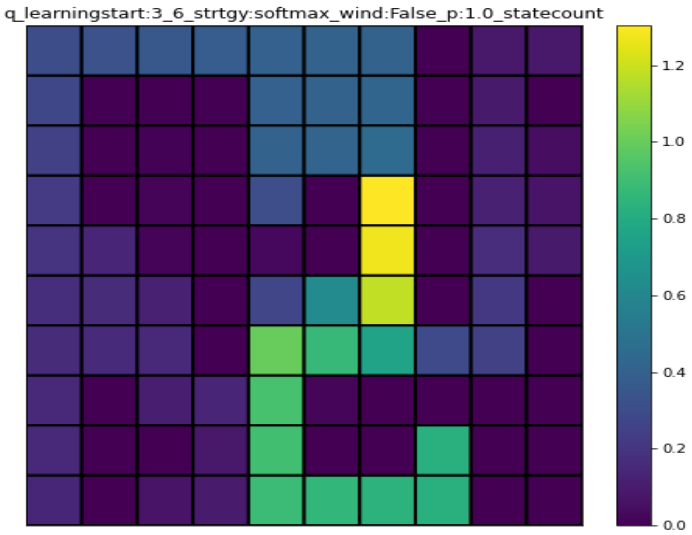

- Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach lower right corner goal.
- Total reward after 1000 episodes averaged over 20 runs = -1.05 , Reward curve:



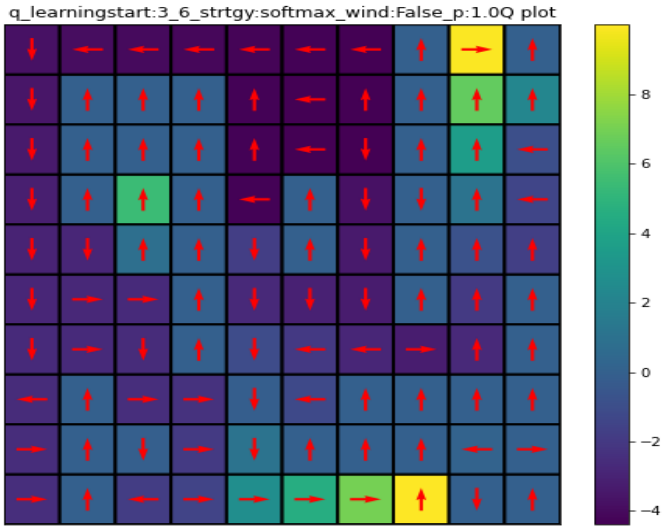

- Number steps to reach goal after 1000 episodes average over 20 runs = 12.05, Step curve:




– Heatmap of the grid with state visit count:

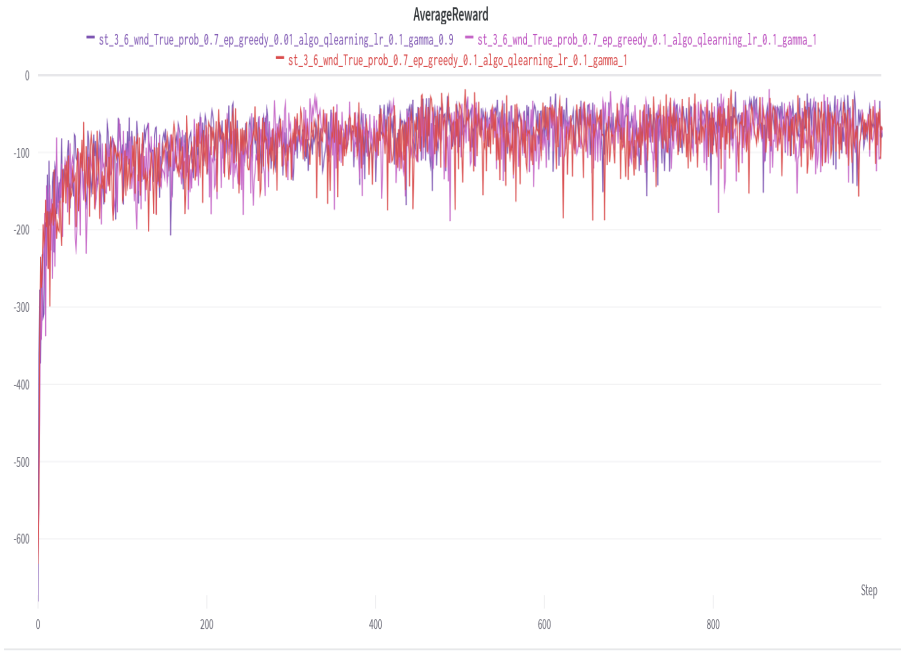



– Heatmap of the grid with Q values after training is complete:




13. **strategy=$\epsilon$-greedy, start_state=(3,6), wind=True, p=0.7**

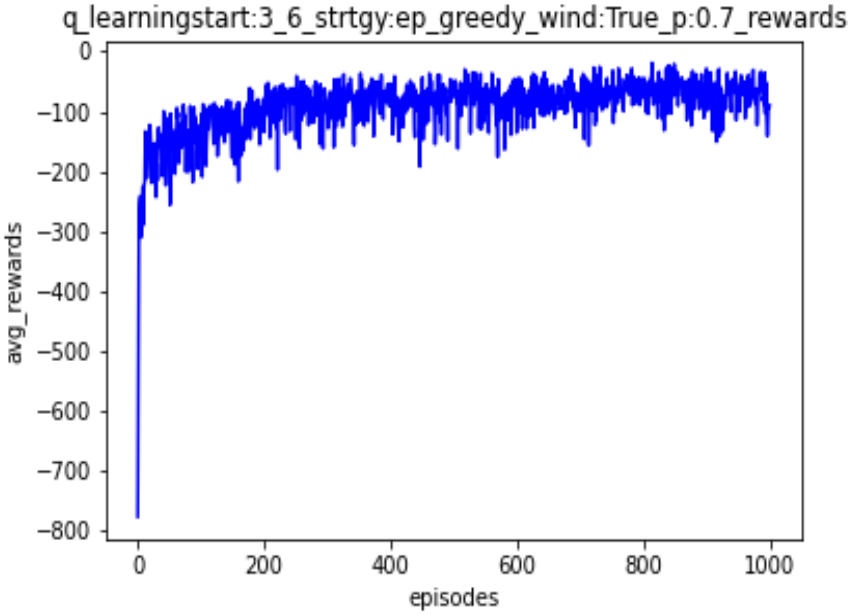
    – From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\epsilon = 0.1$ gave better performance. Following plot shows some of the best performing hyper-parameters.



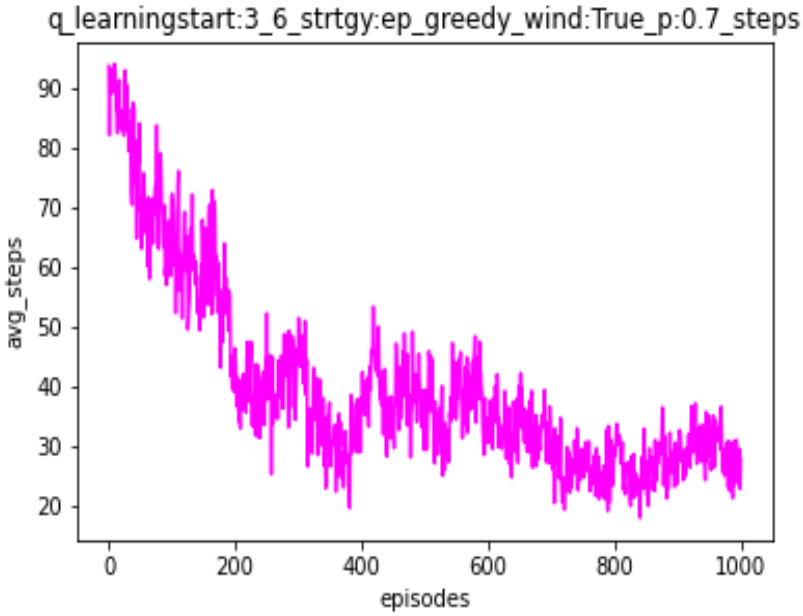

    – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach both upper right corner goal may be because of wind and upper left corner goal.

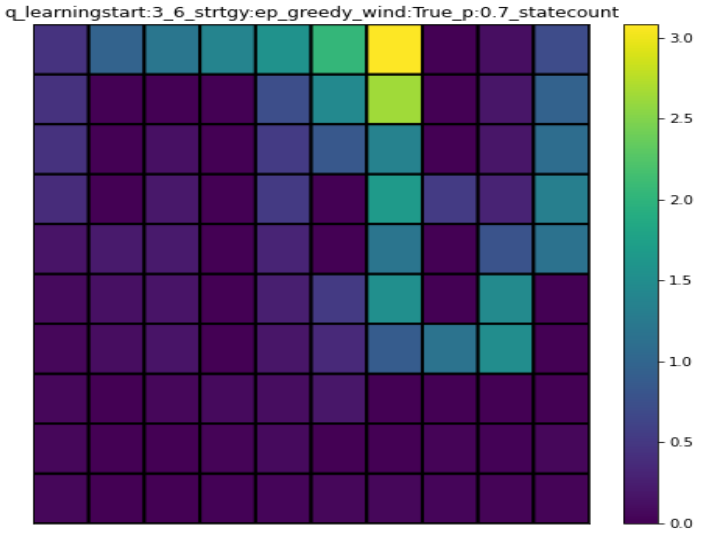
– Total reward after 1000 episodes averaged over 20 runs = -69.35 , Reward curve:



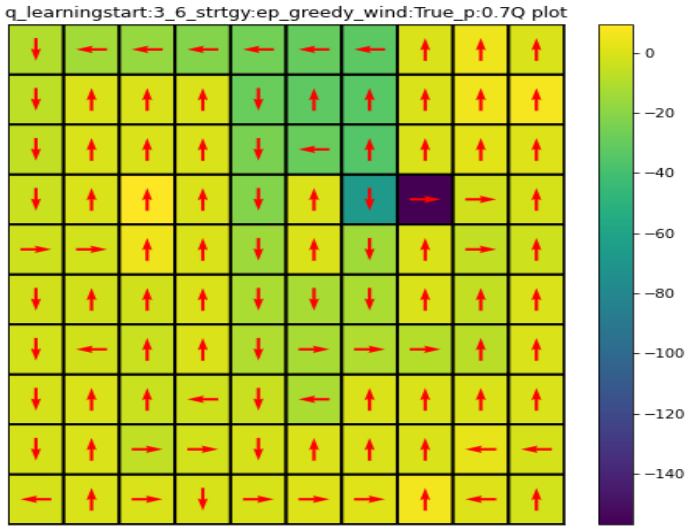

– Number steps to reach goal after 1000 episodes average over 20 runs = 22.85 , Step curve:



– Heatmap of the grid with state visit count:



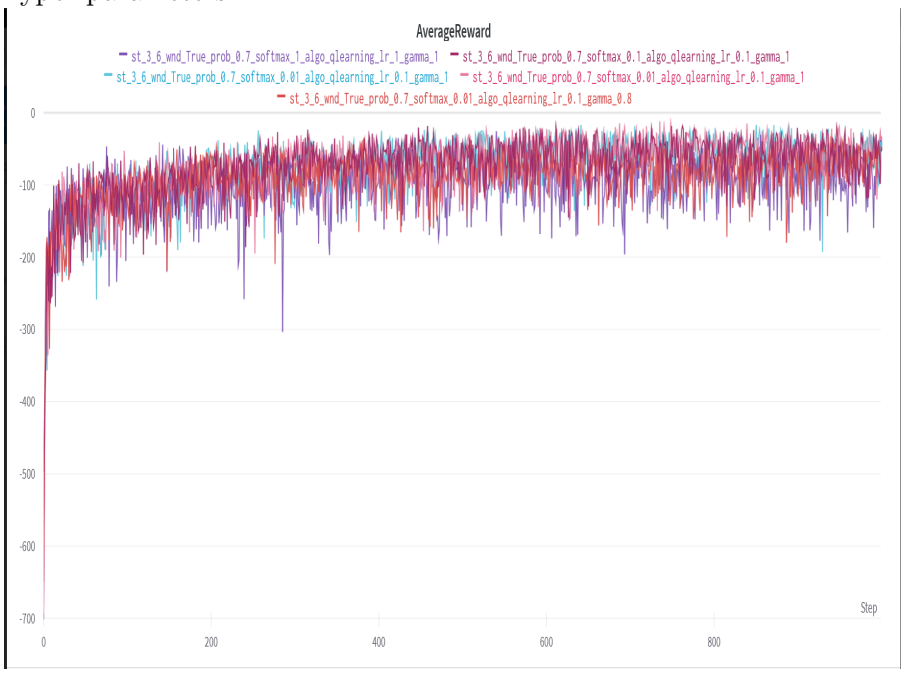

– Heatmap of the grid with Q values after training is complete:




14. **strategy=softmax , start_state=(3,6), wind= True, p=0.7**

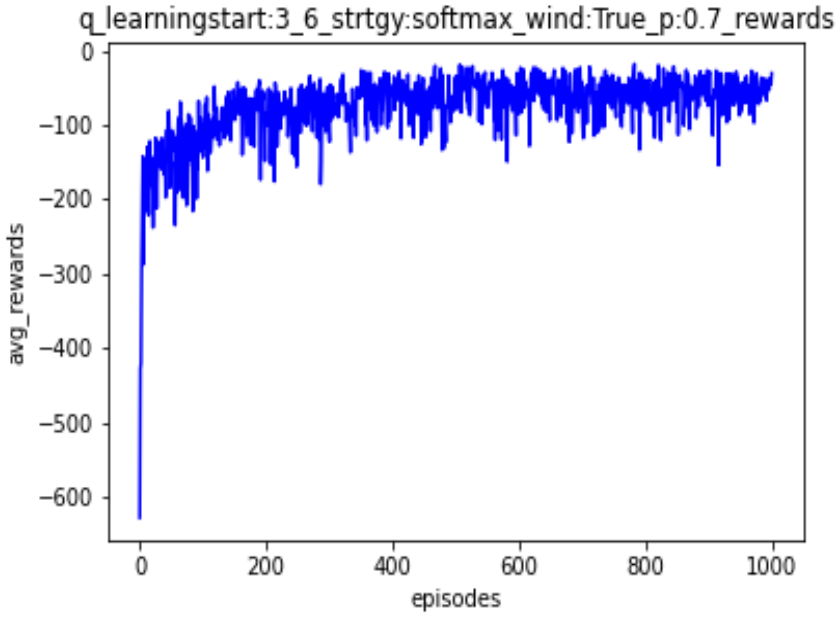
– From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\beta = 0.01$ gave better performance.Following plot shows some of the best performing

hyper-parameters.

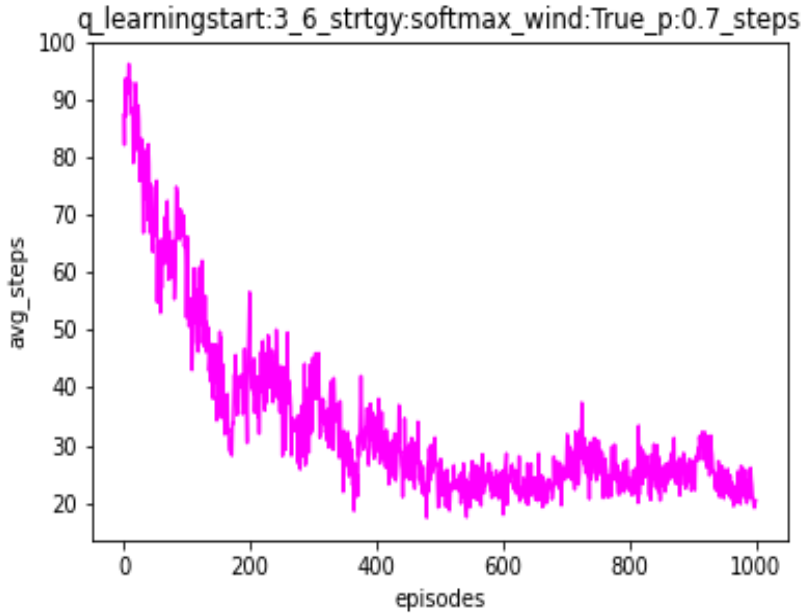



– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach both upper right corner goal may be because of wind and upper left corner goal.

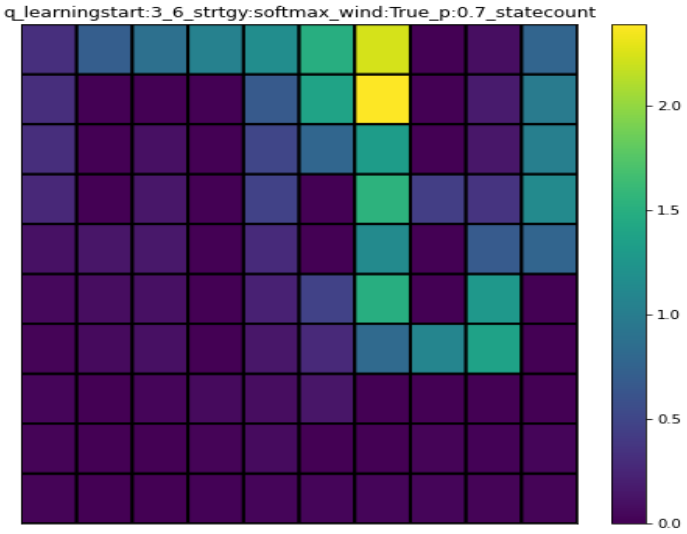
– Total reward after 1000 episodes averaged over 20 runs = -35.75, Reward curve:



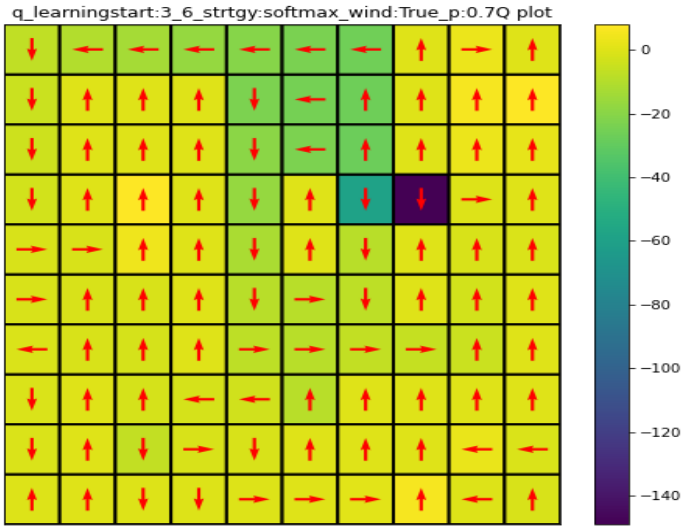
– Number steps to reach goal after 1000 episodes average over 20 runs = 24.15, Step curve:




– Heatmap of the grid with state visit count:

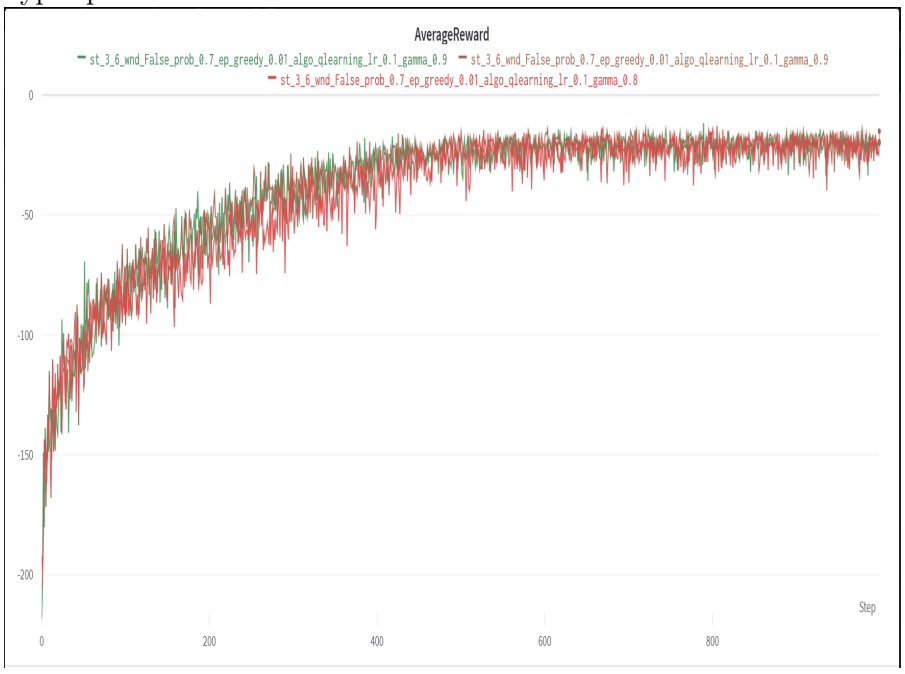



– Heatmap of the grid with Q values after training is complete:




15. **strategy=$\epsilon$-greedy, start_state=(3,6), wind=False, p=0.7**

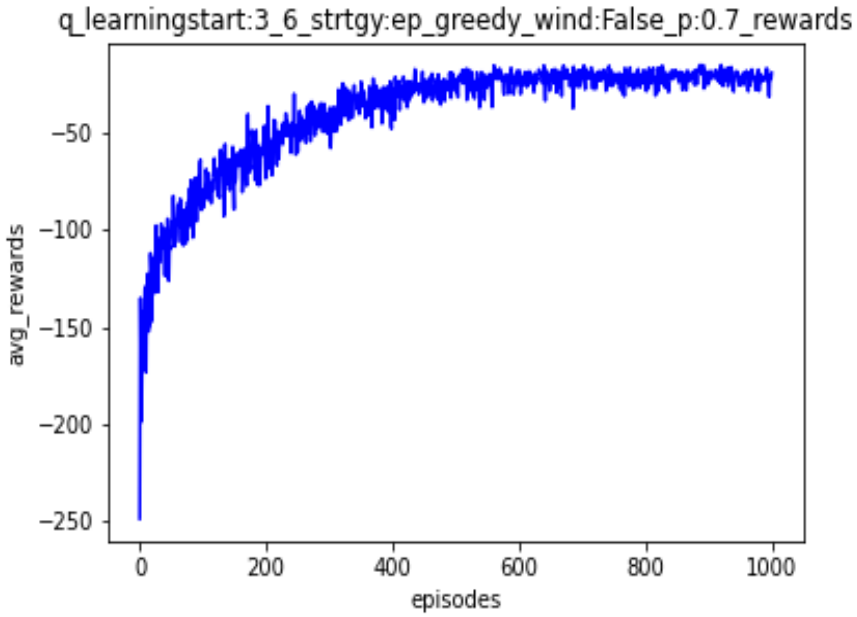
   – From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\epsilon = 0.01$ gave better performance.Following plot shows some of the best performing hyper-parameters.



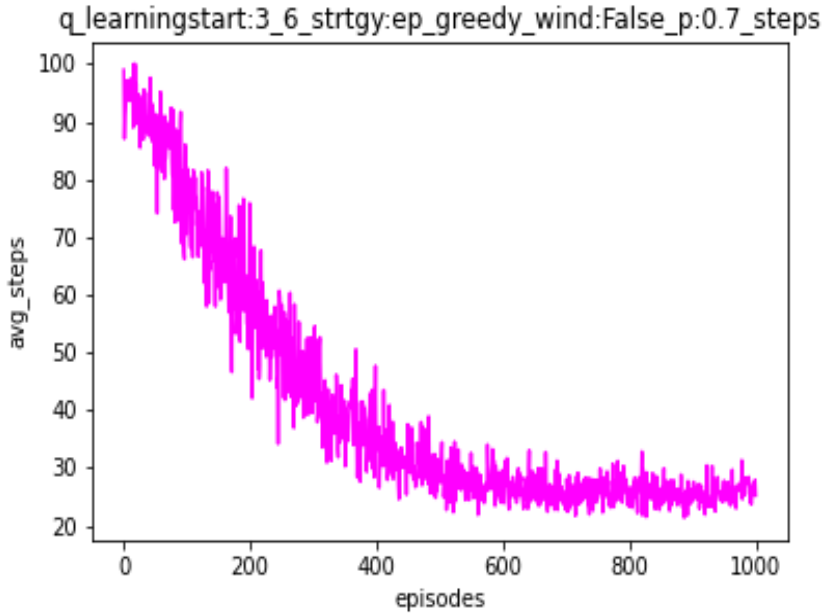

   – Policy learnt: Using the heatmap of state visit count and heatmap of Q

values, we notice that agent has learnt to reach both upper right corner goal and upper left corner goal.

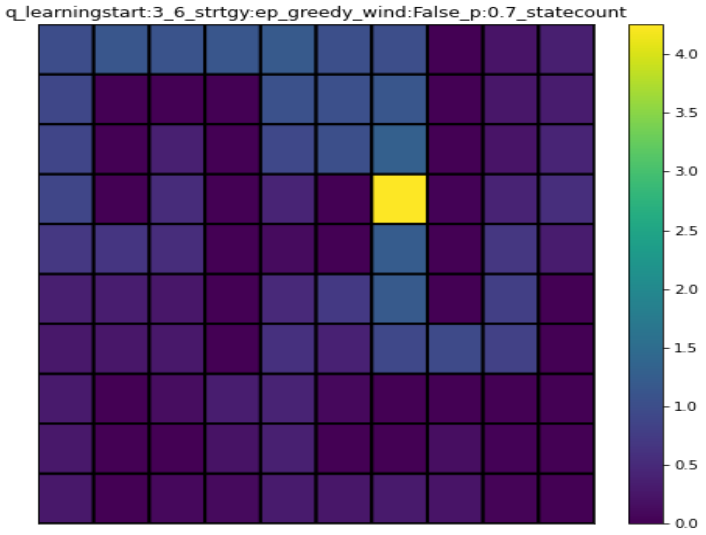
– Total reward after 1000 episodes averaged over 20 runs = -15.1 , Reward curve:



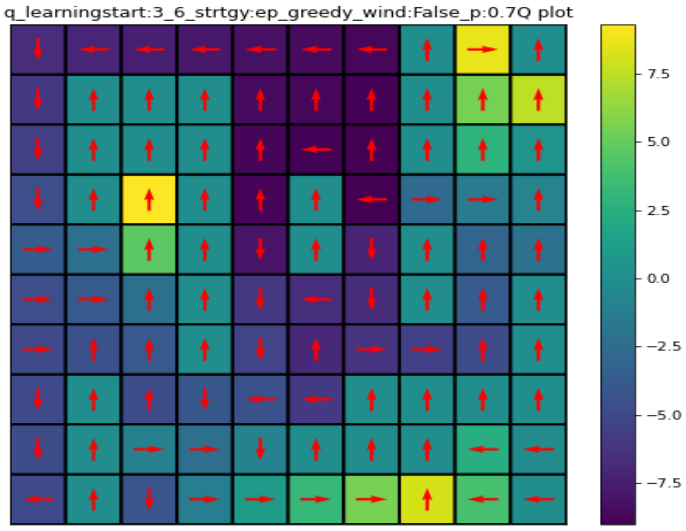

– Number steps to reach goal after 1000 episodes average over 20 runs = 23.35 , Step curve:



– Heatmap of the grid with state visit count:



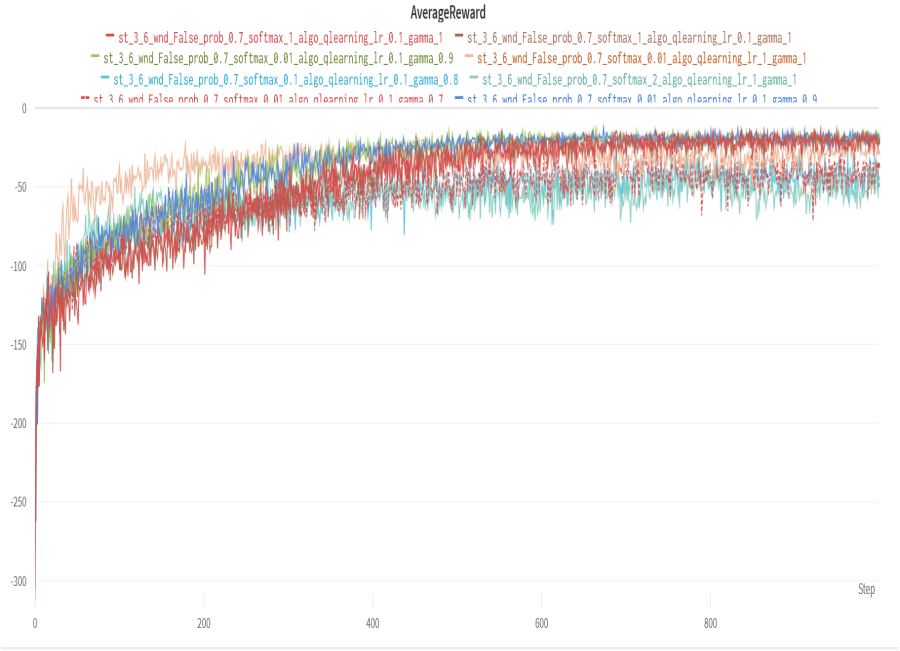

– Heatmap of the grid with Q values after training is complete:




16. **strategy=softmax , start_state=(3,6), wind=False, p=0.7**

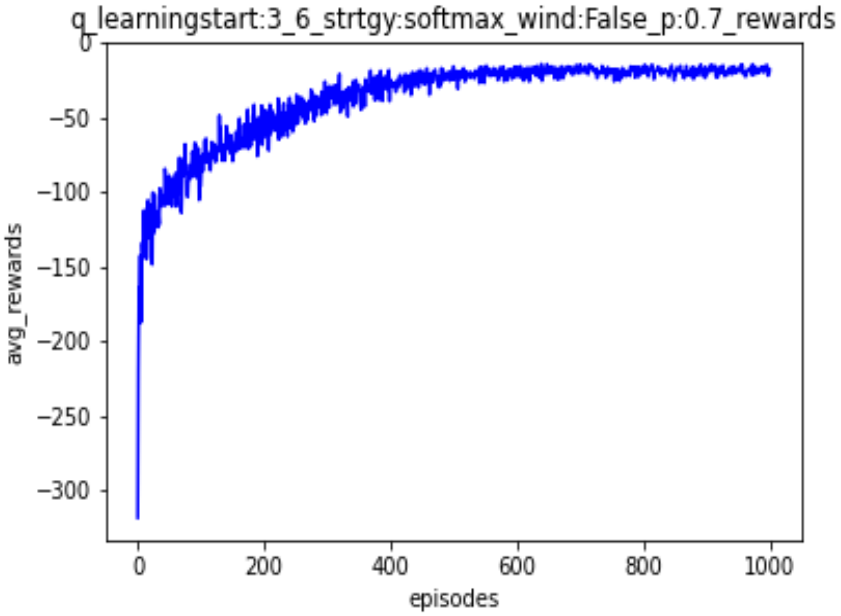
– From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\beta = 0.01$ gave better performance. Following plot shows some of the best performing

hyper-parameters.



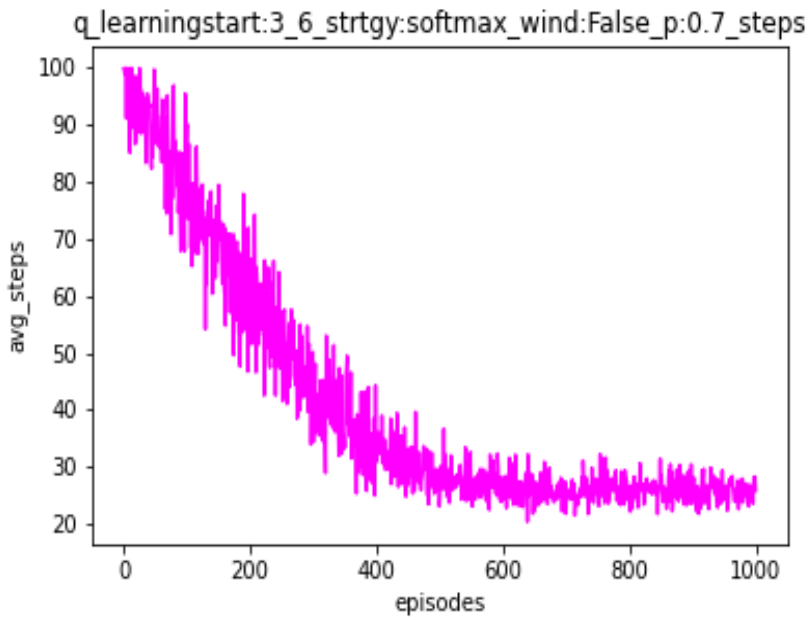

– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach both upper right corner goal and upper left corner goal.

– Total reward after 1000 episodes averaged over 20 runs = -17.9, Reward curve:

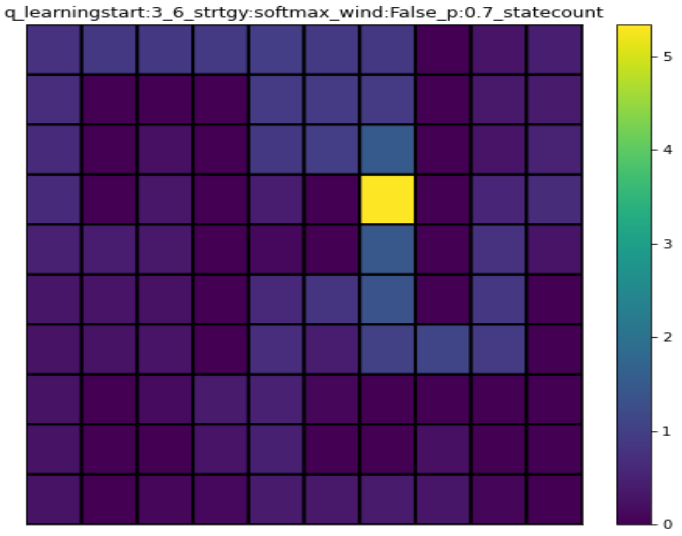



– Number steps to reach goal after 1000 episodes average over 20 runs = 24.65,

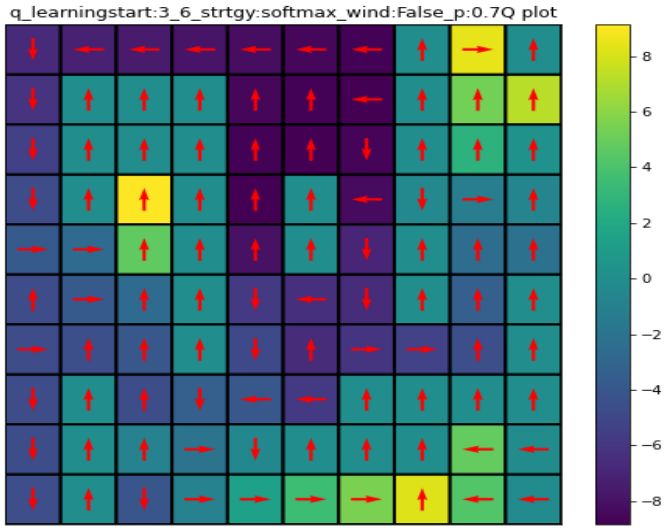
Step curve:




– Heatmap of the grid with state visit count:



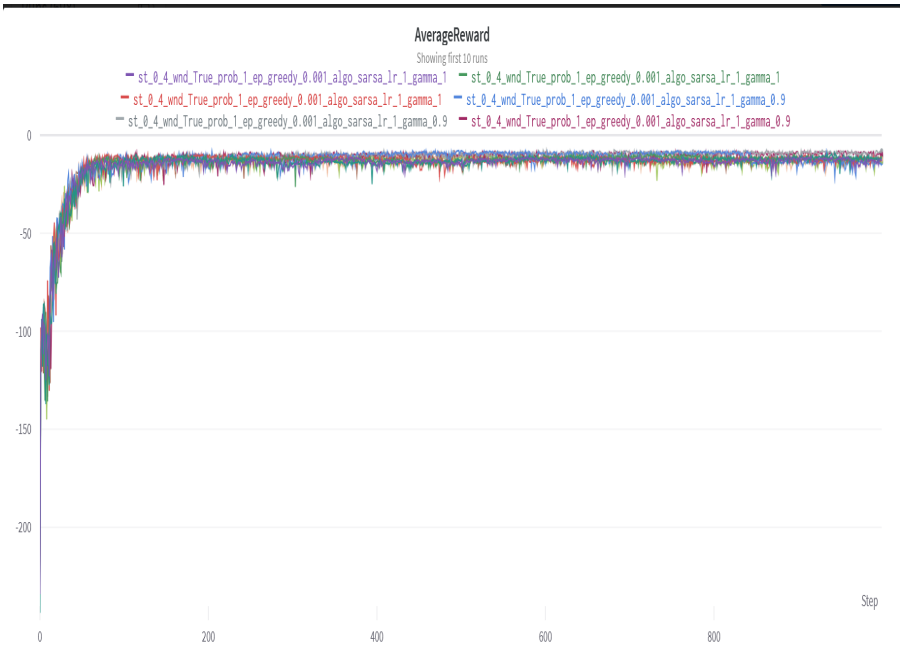

– Heatmap of the grid with Q values after training is complete:




- **SARSA Learning-**

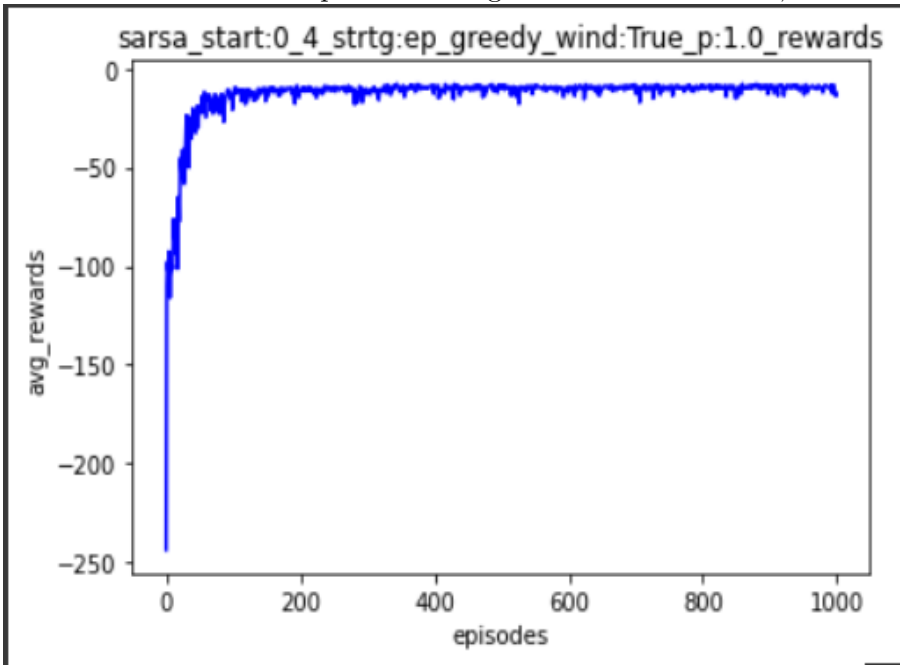
**Solution:**

1. **strategy=$\epsilon$-greedy , start_state=(0,4), wind=True, p=1.0**
   - From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 1.0$, $\epsilon = 0.001$ gave better performance. Following plot shows some of the best performing hyper-parameters.

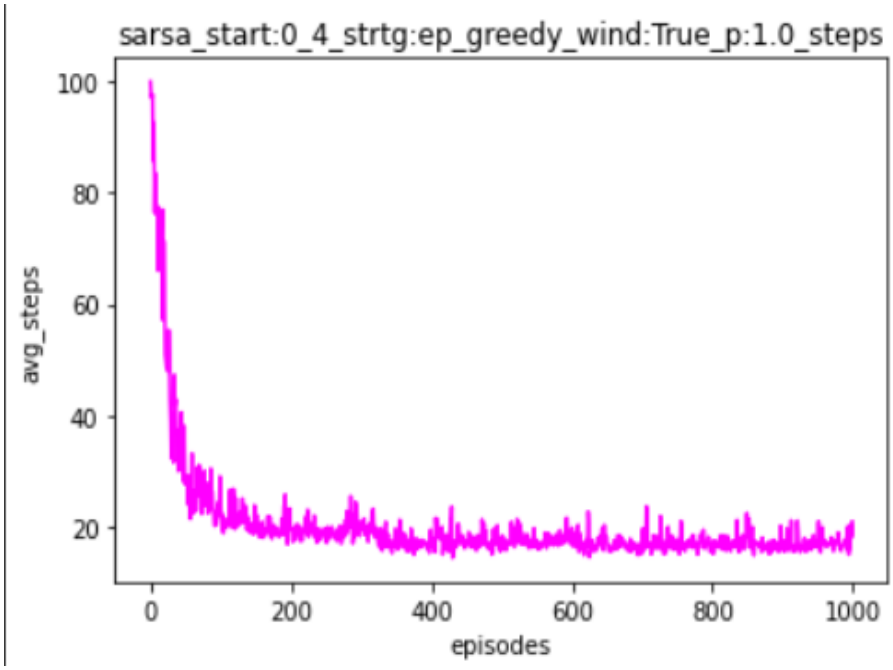



– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach the upper left corner goal and lower right corner goal.

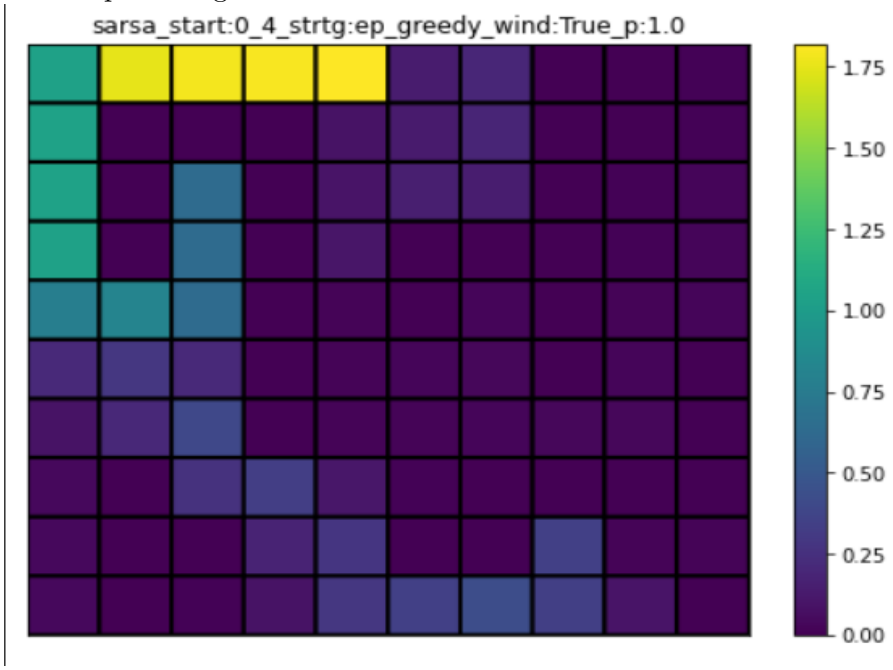
– Total reward after 1000 episodes averaged over 20 runs = -8.2, Reward curve:



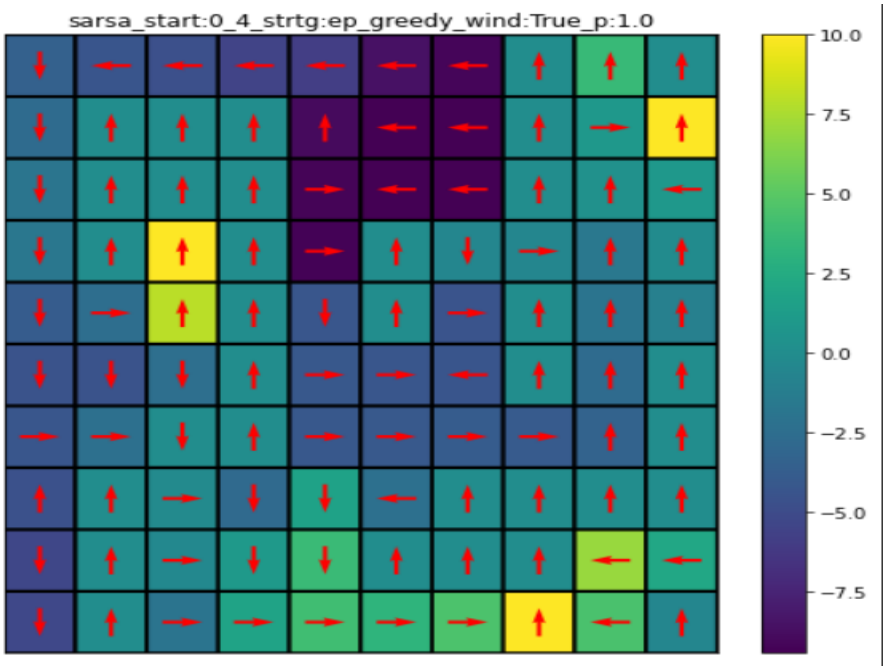

– Number steps to reach goal after 1000 episodes average over 20 runs = 14.95, Step curve:




– Heatmap of the grid with state visit count:

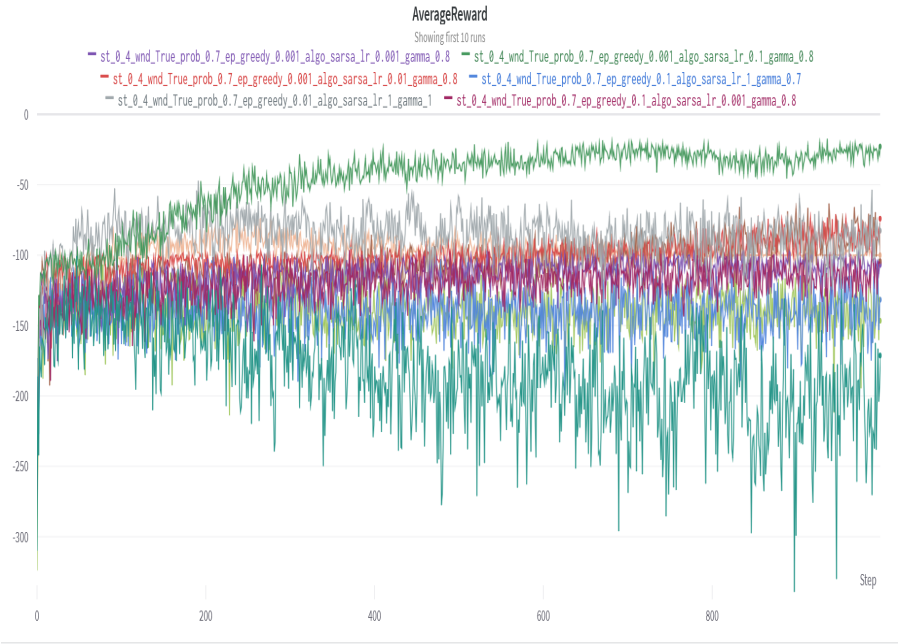



– Heatmap of the grid with Q values after training is complete:




2. **strategy=$\epsilon$-greedy , start_state=(0,4), wind=True, p=0.7**

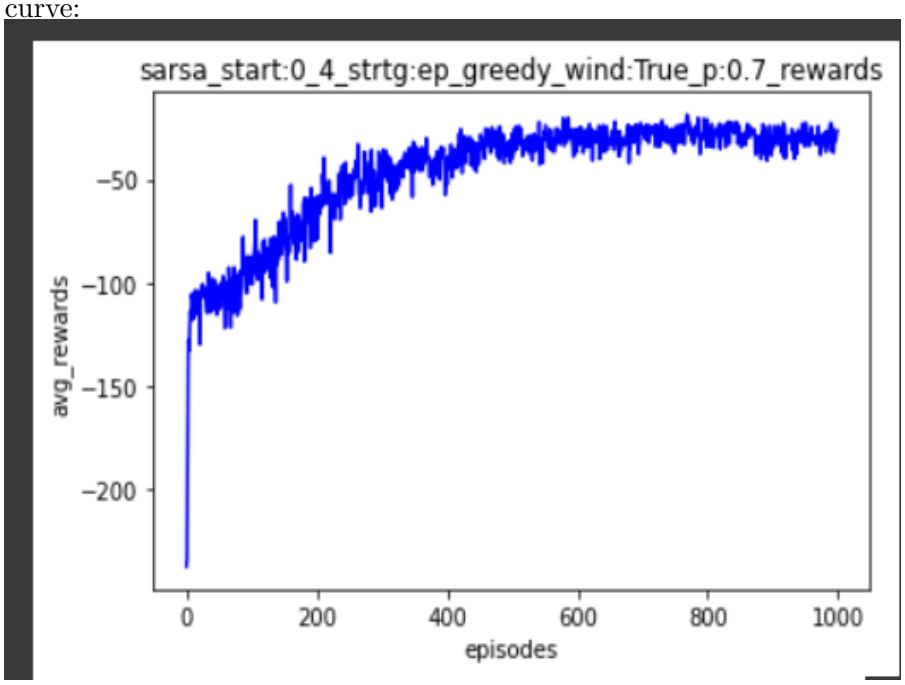
   – From the experiments we performed we found that $\gamma = 0.8$, $\alpha = 0.1$, $\epsilon = 0.001$ gave better performance. Following plot shows some of the best performing hyper-parameters.



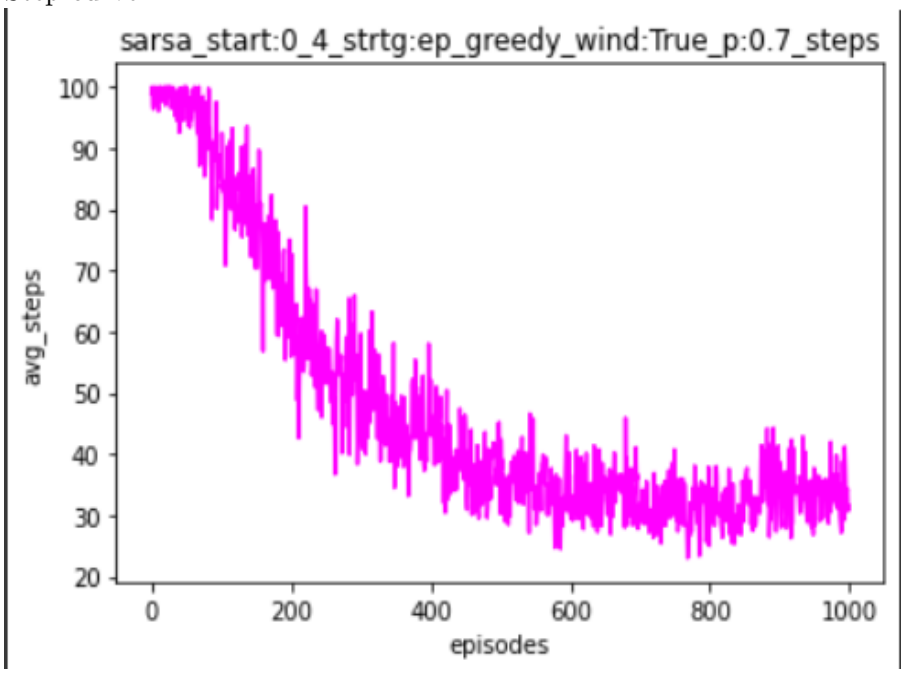

   – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach the upper left corner goal.

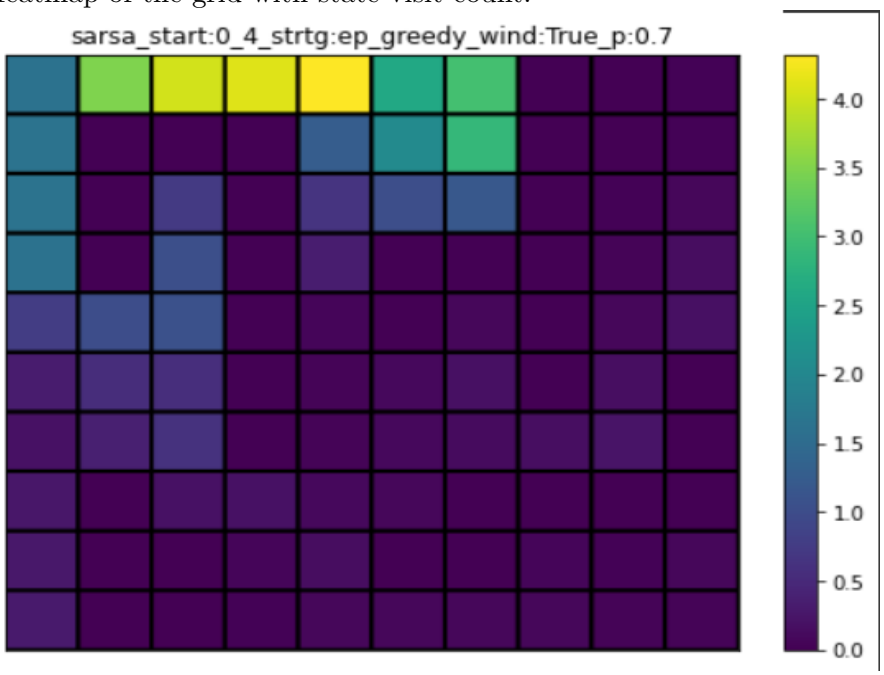
– Total reward after 1000 episodes averaged over 20 runs = -22.85, Reward curve:



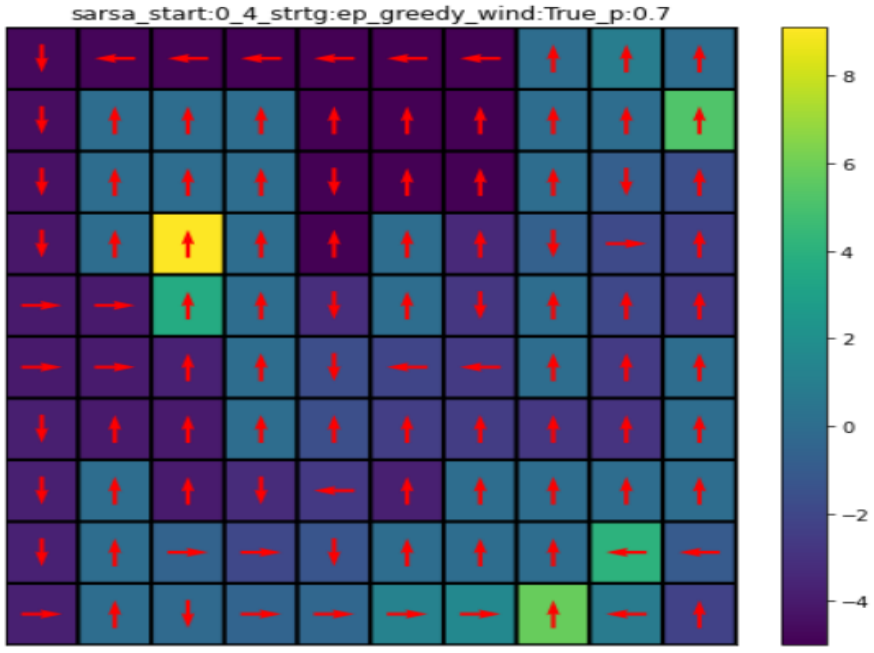

– Number steps to reach goal after 1000 episodes average over 20 runs = 26.85, Step curve:



– Heatmap of the grid with state visit count:




– Heatmap of the grid with Q values after training is complete:



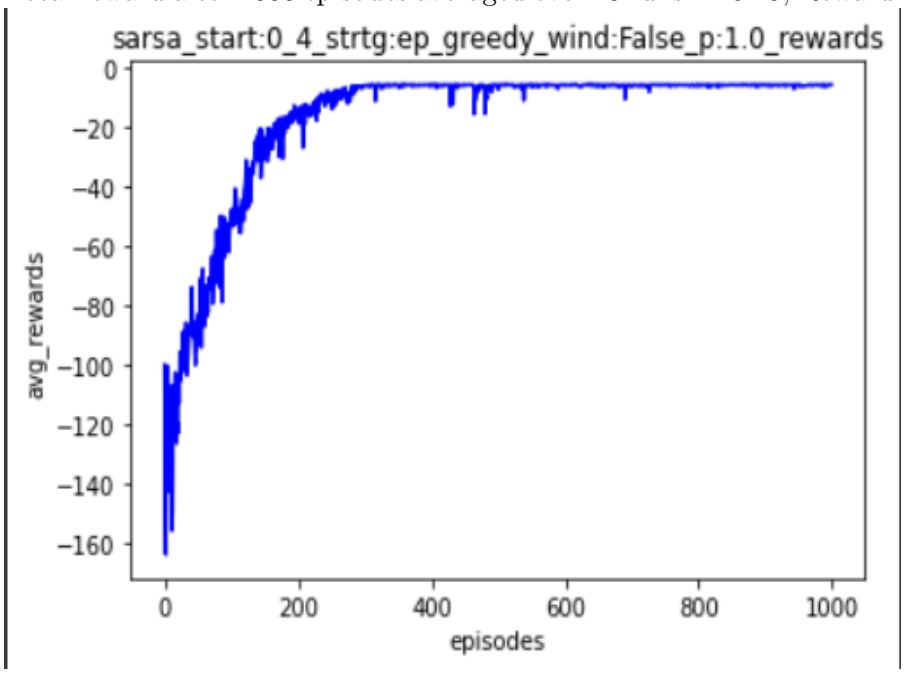

3. **strategy=$\epsilon$-greedy , start_state=(0,4), wind=False, p=1.0**

– From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\epsilon = 0.01$

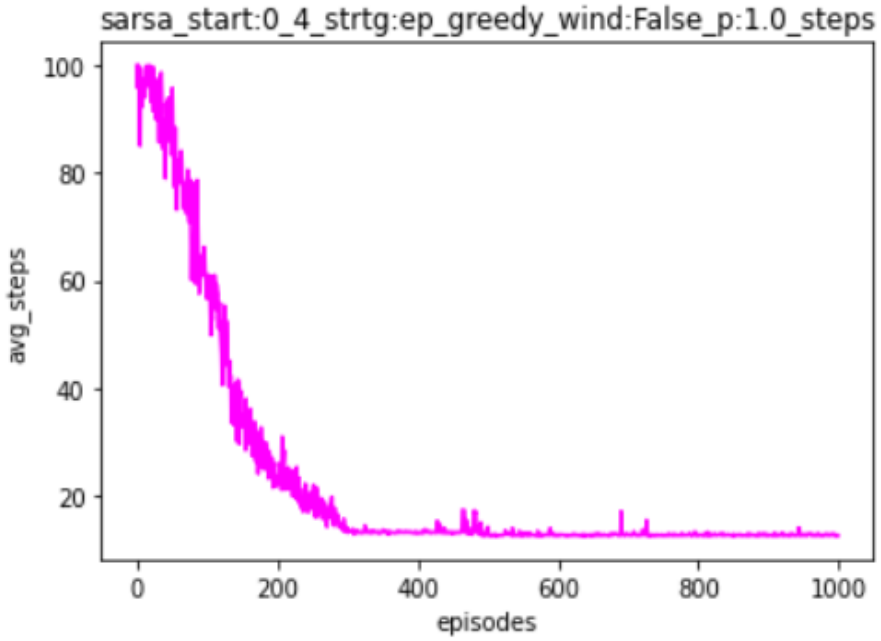
gave better performance. Following plot shows some of the best performing hyper-parameters.


– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach all upper left corner, upper right corner, lower right corner goals.

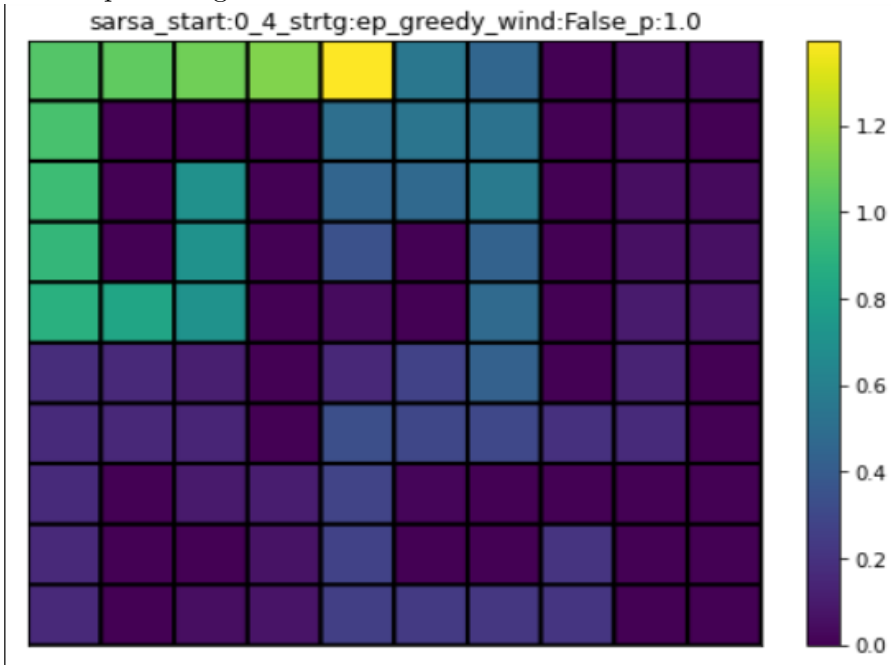
– Total reward after 1000 episodes averaged over 20 runs = -6.15, Reward curve:



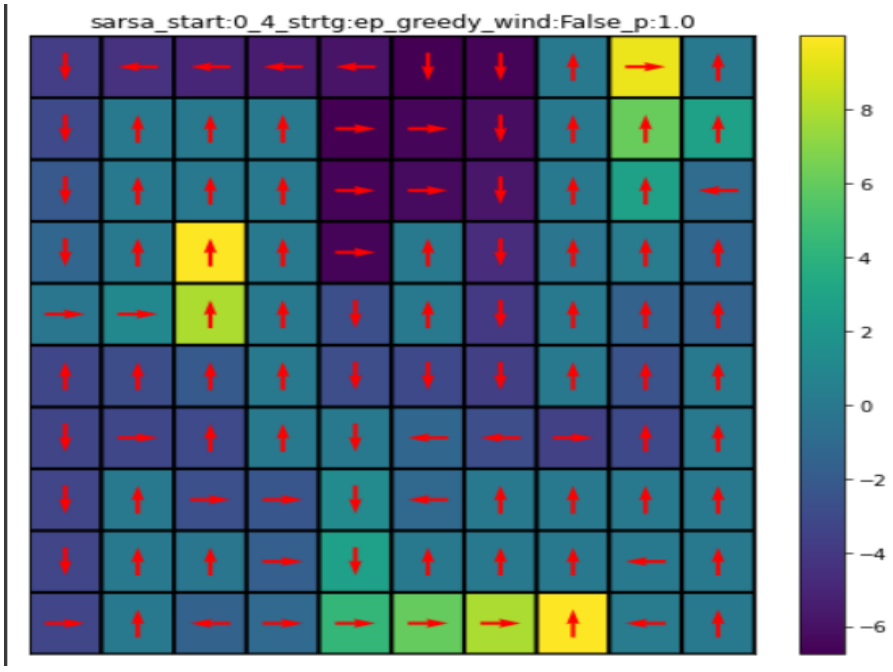
– Number steps to reach goal after 1000 episodes average over 20 runs = 12.65, Step curve:




– Heatmap of the grid with state visit count:




– Heatmap of the grid with Q values after training is complete:



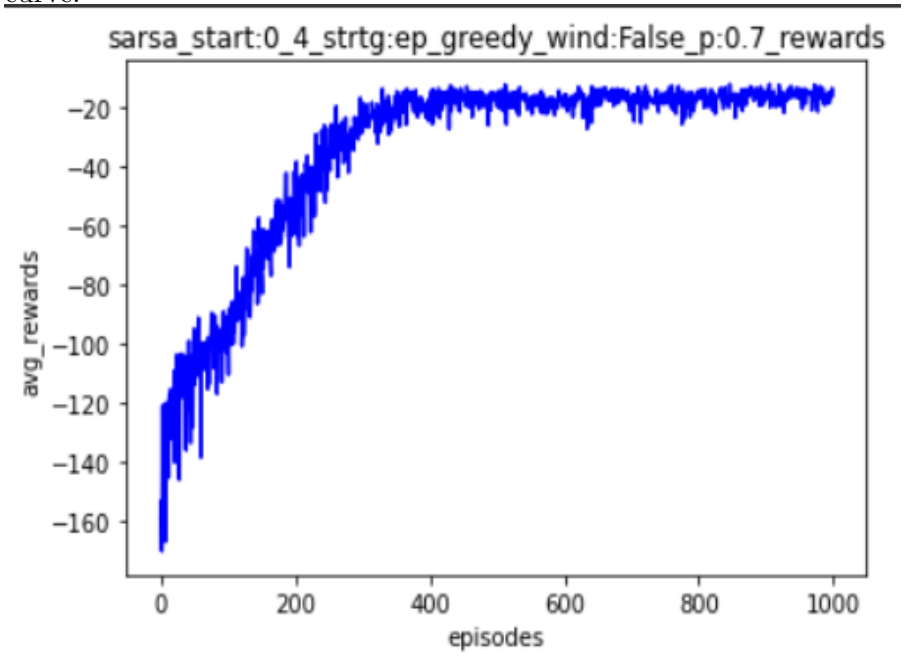

4. **strategy=$\epsilon$-greedy , start_state=(0,4), wind=False, p=0.7**

   – From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\epsilon = 0.01$ gave better performance. Following plot shows some of the best performing hyper-parameters.

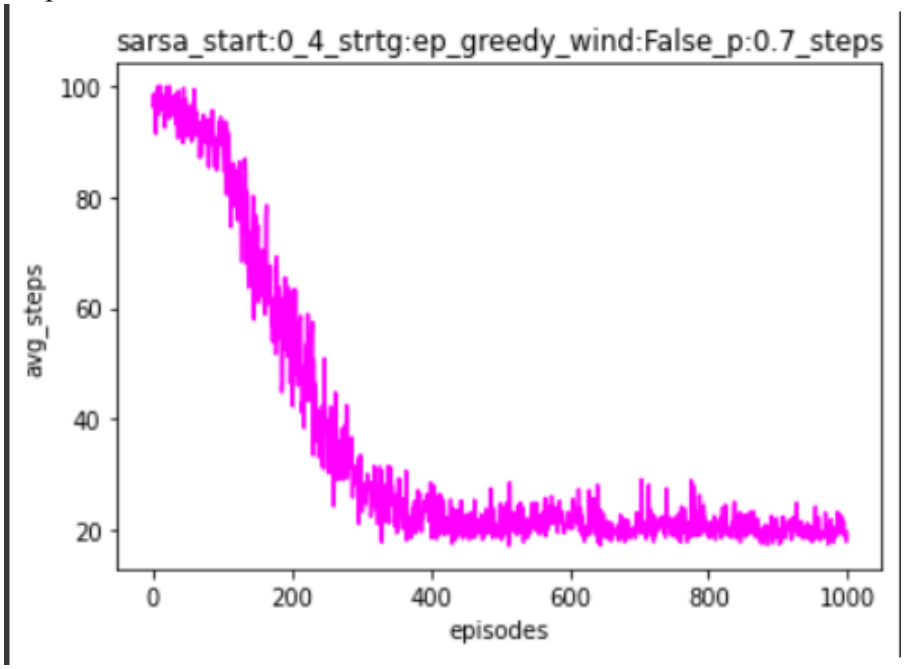

   – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach the upper left corner goal and due to stochastic behaviour of the environment it has also learnt to reach upper right corner goal.

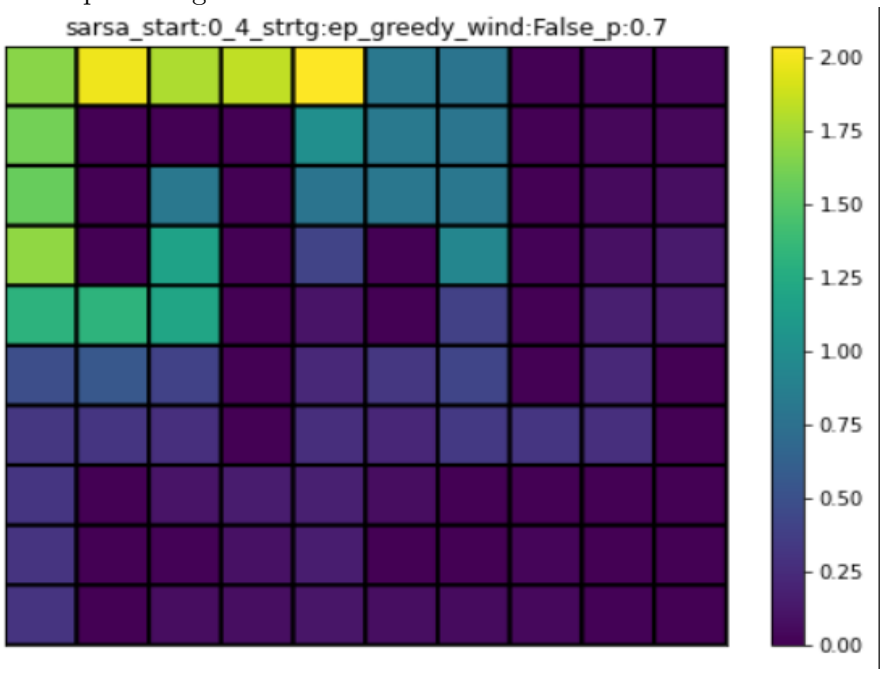
- Total reward after 1000 episodes averaged over 20 runs = -13.75, Reward curve:



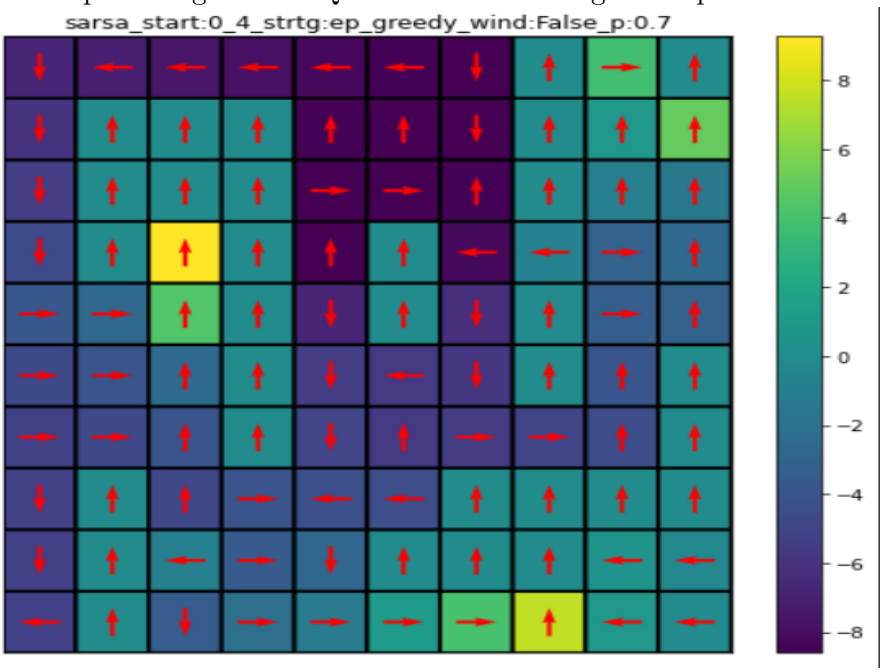

- Number steps to reach goal after 1000 episodes average over 20 runs = 18.75, Step curve:



– Heatmap of the grid with state visit count:




– Heatmap of the grid with Q values after training is complete:



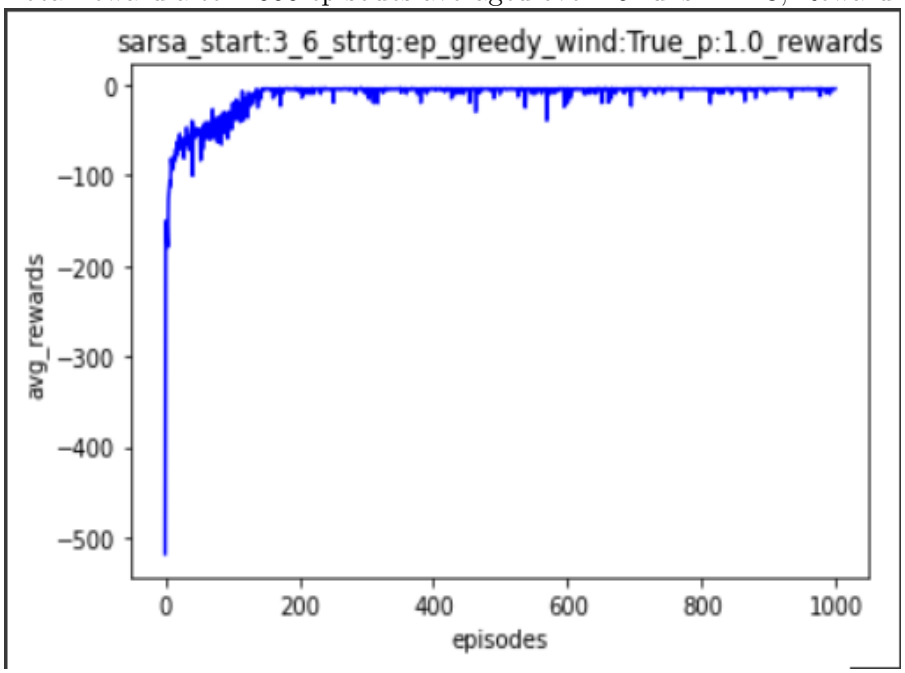

5. **strategy=$\epsilon$-greedy , start_state=(3,6), wind=True, p=1.0**

   – From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\epsilon = 0.01$

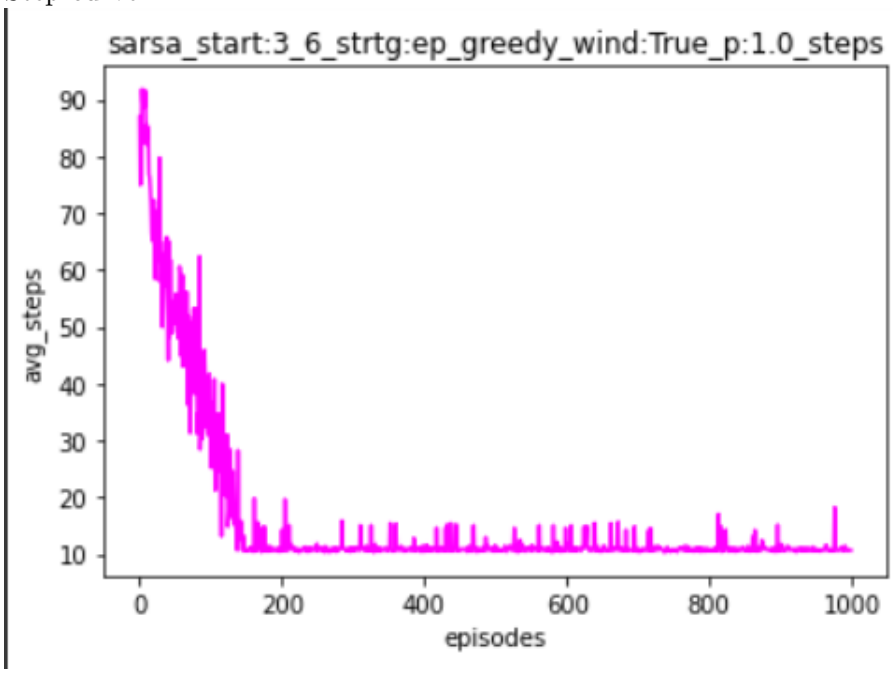
gave better performance. Following plot shows some of the best performing hyper-parameters.


– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach the upper right corner goal may be because of the wind as it tries to move the agent rightward.

– Total reward after 1000 episodes averaged over 20 runs = -4.3, Reward curve:

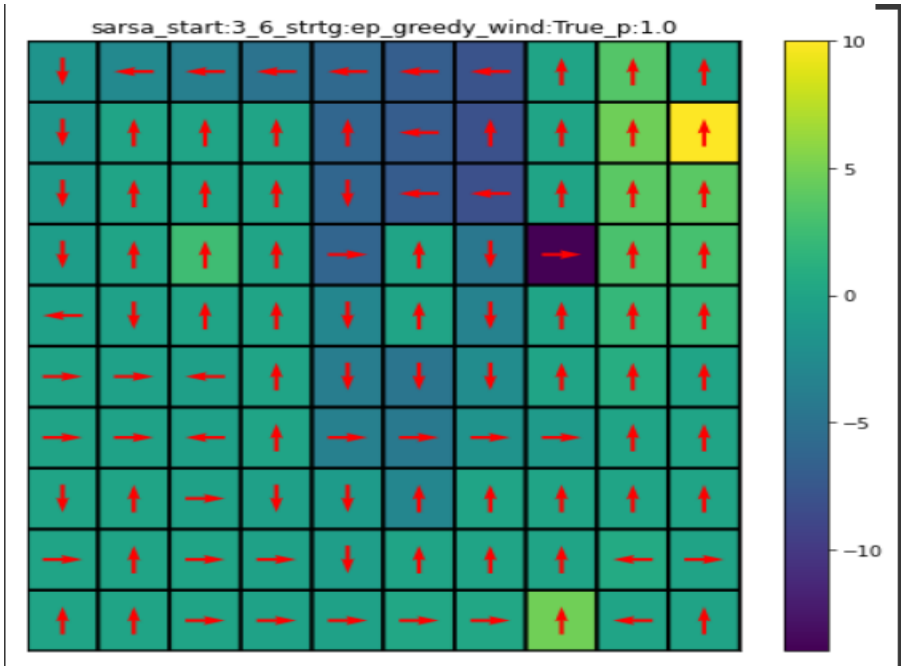


– Number steps to reach goal after 1000 episodes average over 20 runs = 10.8, Step curve:




– Heatmap of the grid with state visit count:


– Heatmap of the grid with Q values after training is complete:



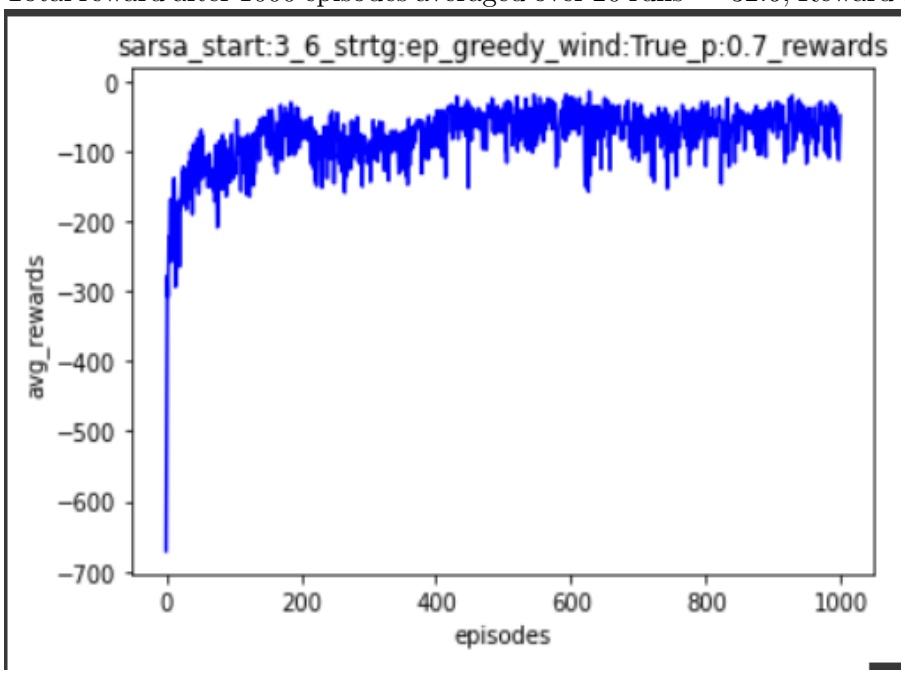

6. **strategy=$\epsilon$-greedy , start_state=(3,6), wind=True, p=0.7**

   – From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\epsilon = 0.001$ gave better performance. Following plot shows some of the best performing hyper-parameters.

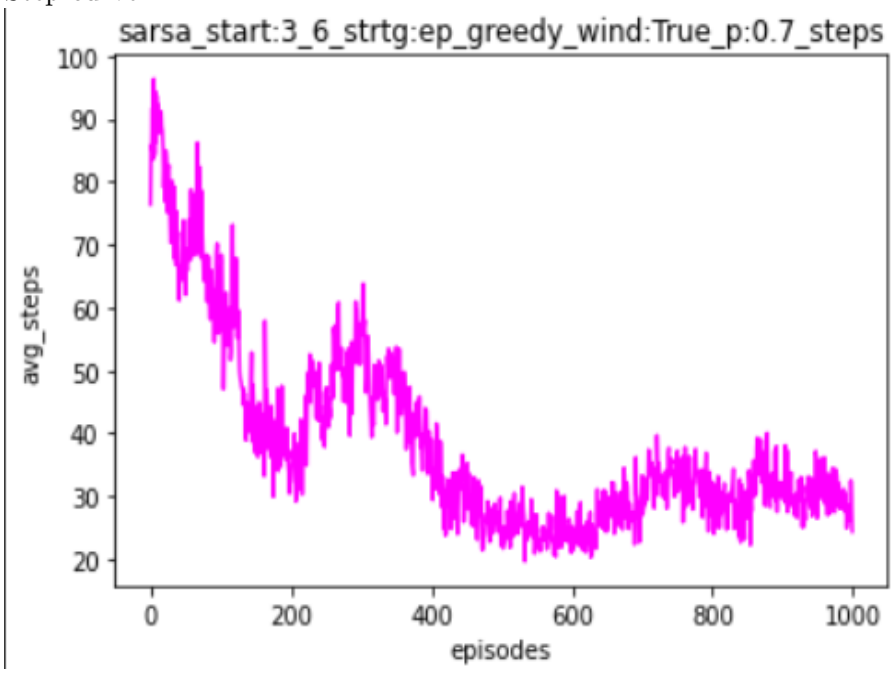

   – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach both upper right corner goal may be beacause of the wind and upper left corner goal.

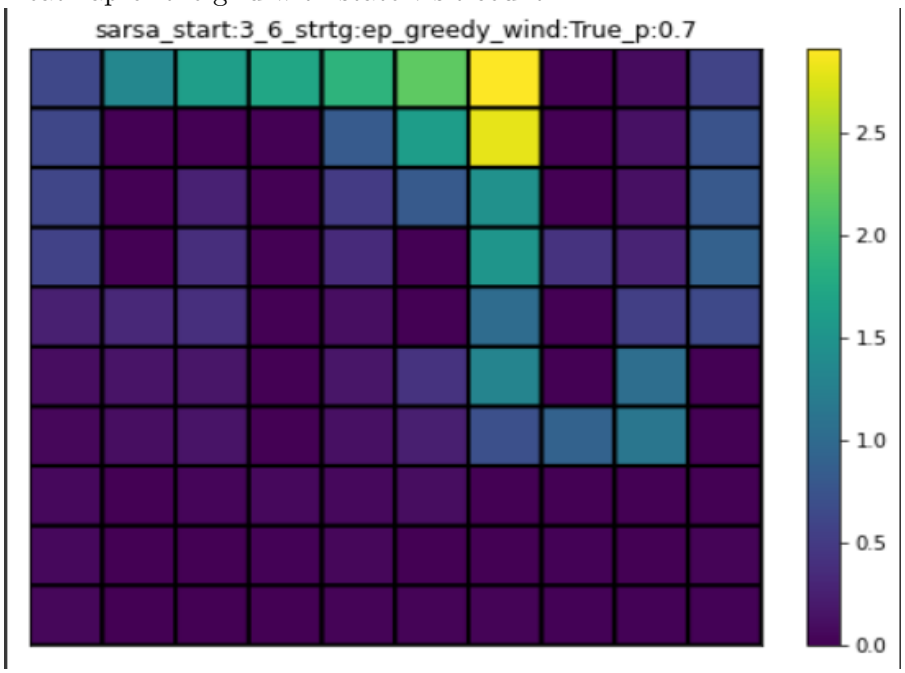
– Total reward after 1000 episodes averaged over 20 runs = -32.6, Reward curve:



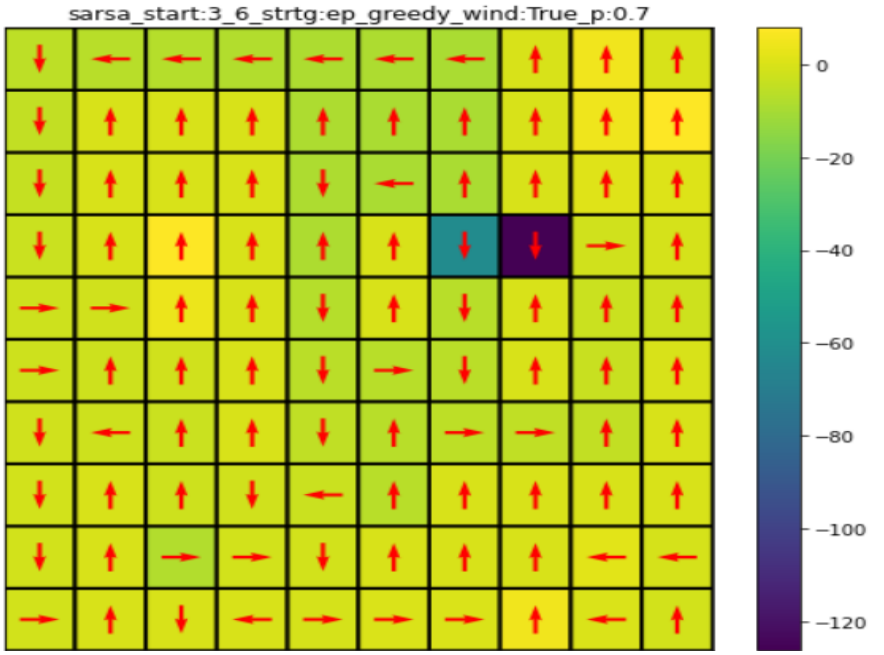

– Number steps to reach goal after 1000 episodes average over 20 runs = 27.95, Step curve:



– Heatmap of the grid with state visit count:



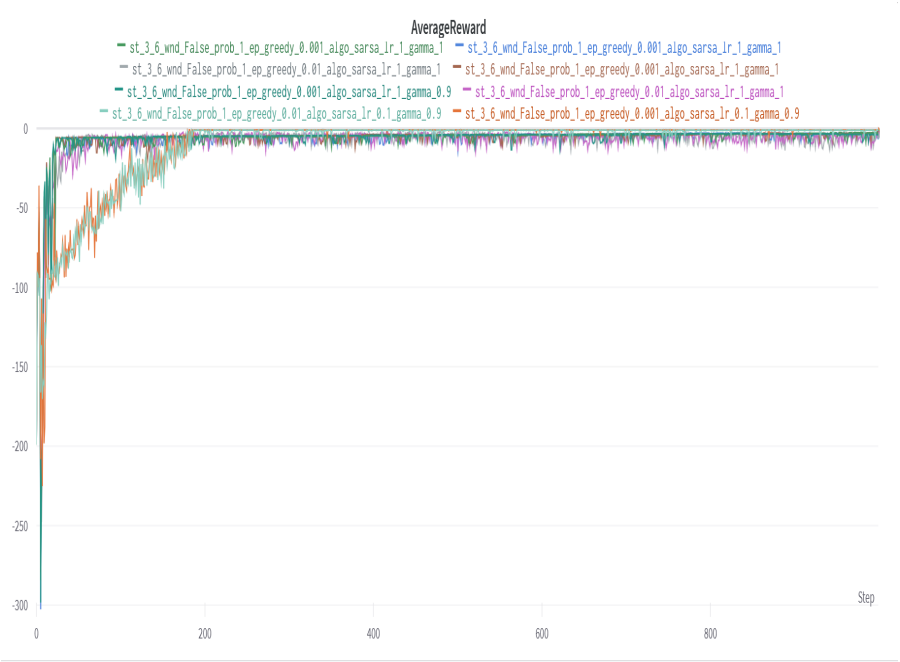

– Heatmap of the grid with Q values after training is complete:




7. **strategy=$\epsilon$-greedy , start_state=(3,6), wind=False, p=1.0**

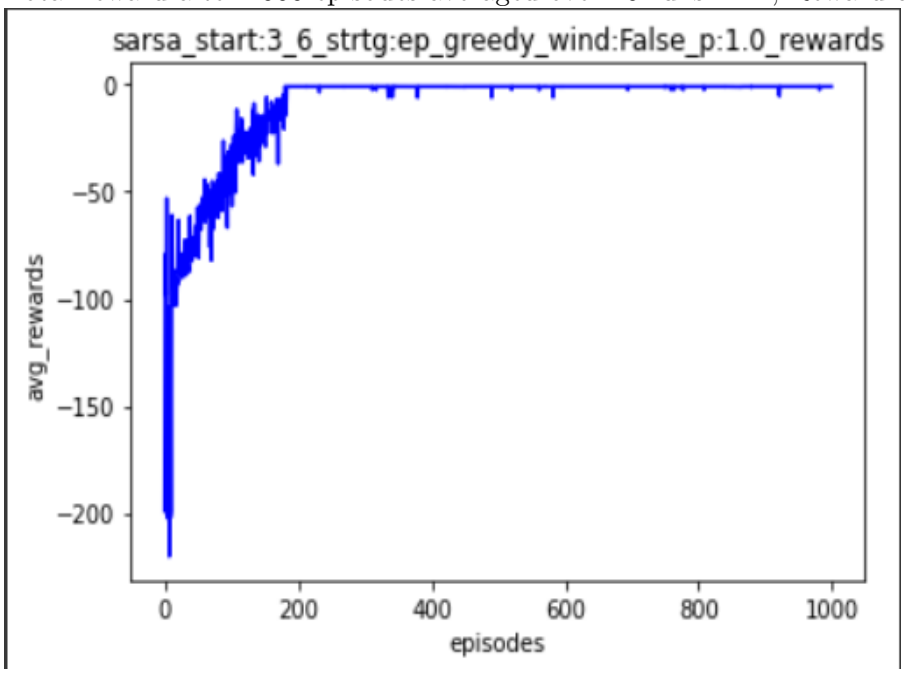
– From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\epsilon = 0.001$

gave better performance. Following plot shows some of the best performing hyper-parameters.

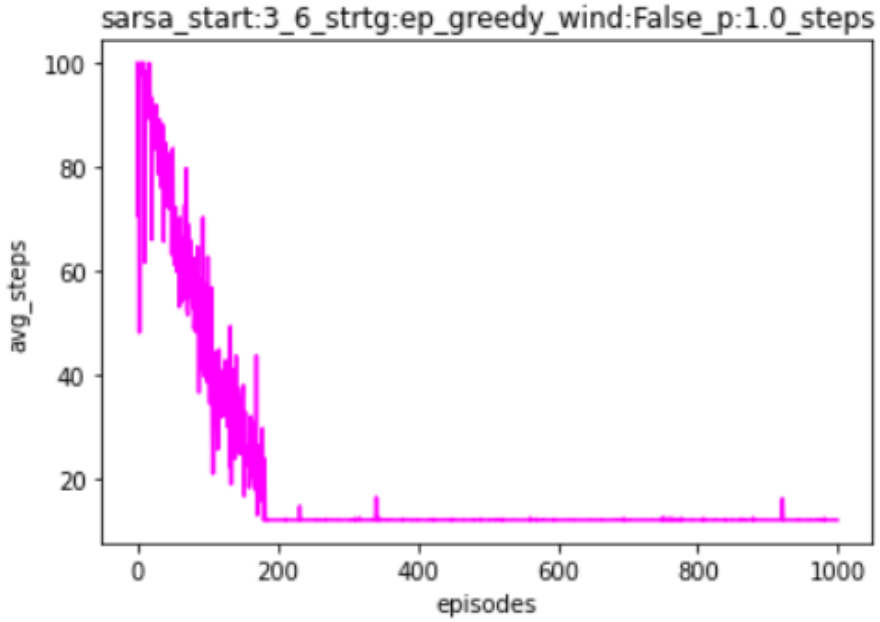



– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach the lower right corner goal. And due to exploration algorithm agent has also learnt to reach other goal state.

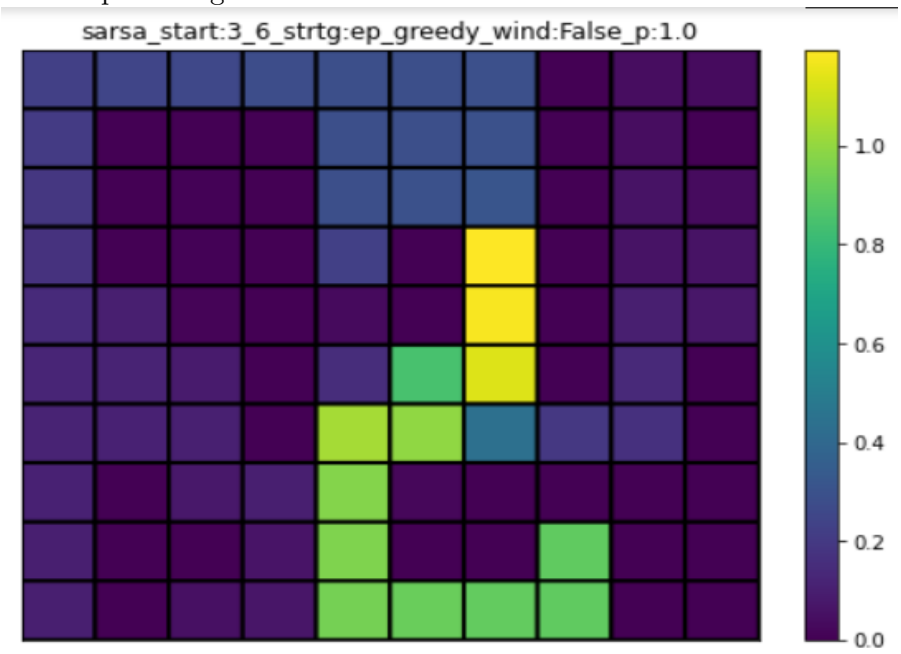
– Total reward after 1000 episodes averaged over 20 runs = -1, Reward curve:



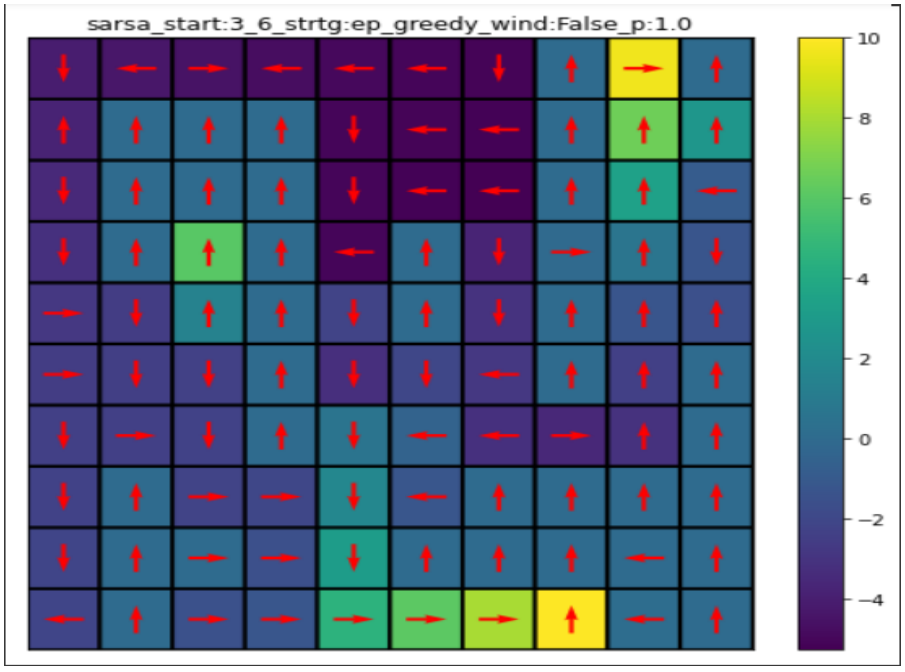
– Number steps to reach goal after 1000 episodes average over 20 runs = 12, Step curve:




– Heatmap of the grid with state visit count:

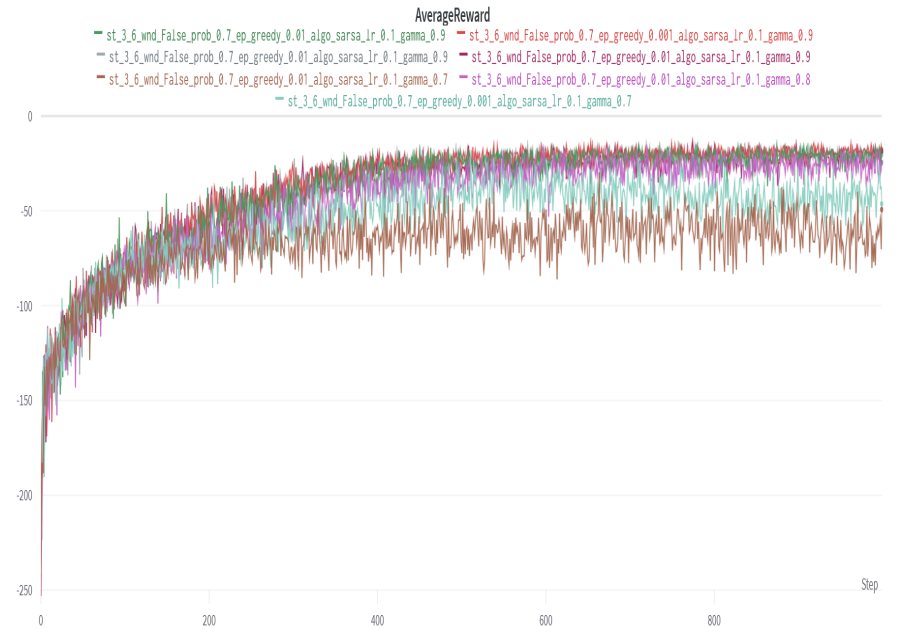



– Heatmap of the grid with Q values after training is complete:




8. **strategy=$\epsilon$-greedy , start_state=(3,6), wind=False, p=0.7**

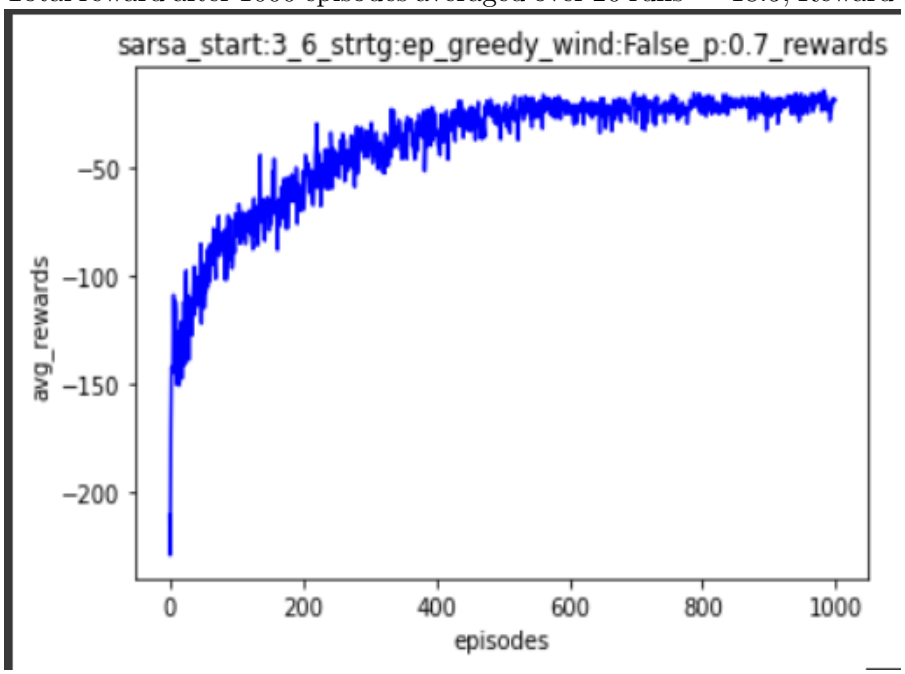
   – From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\epsilon = 0.01$ gave better performance. Following plot shows some of the best performing hyper-parameters.



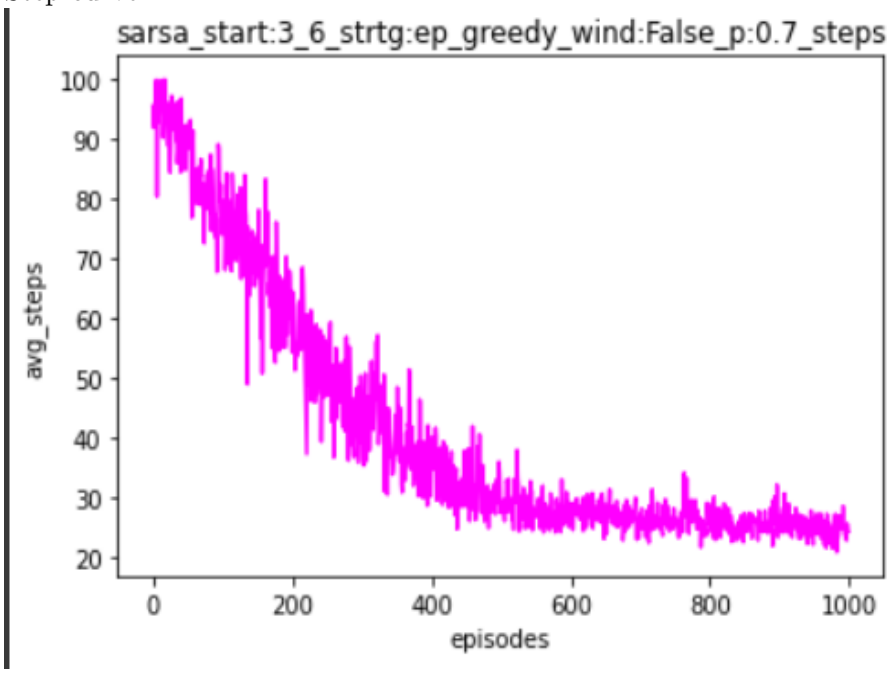

   – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach the upper left corner goal and upper right corner goal.

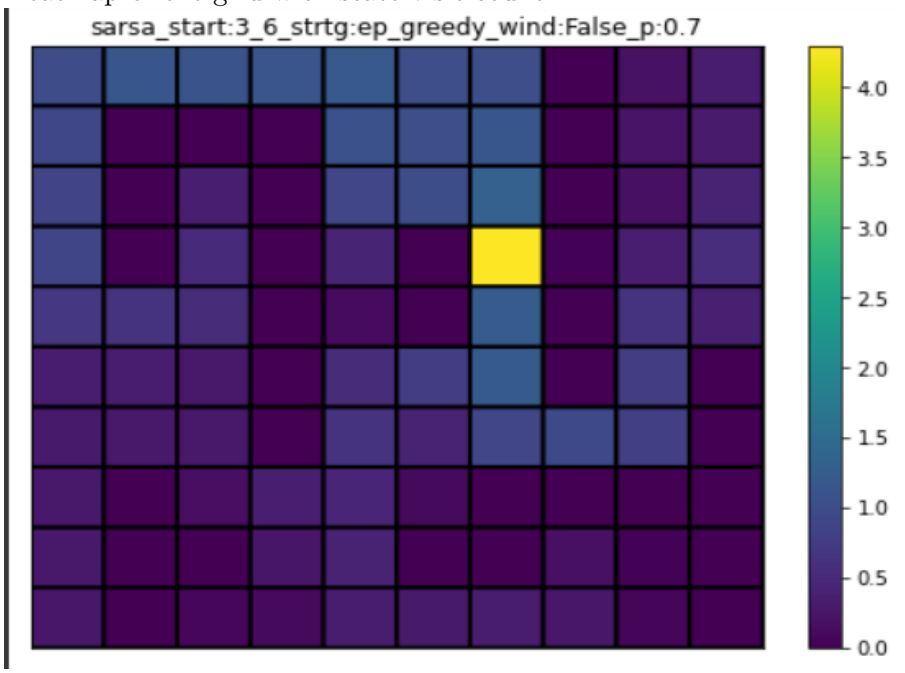
– Total reward after 1000 episodes averaged over 20 runs = -18.5, Reward curve:



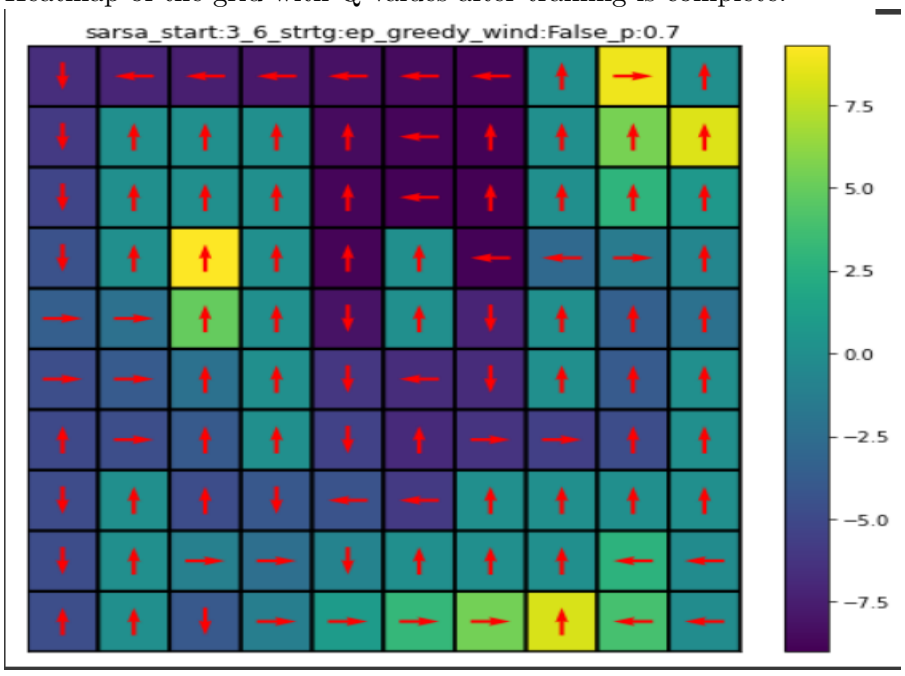

– Number steps to reach goal after 1000 episodes average over 20 runs = 25, Step curve:



– Heatmap of the grid with state visit count:



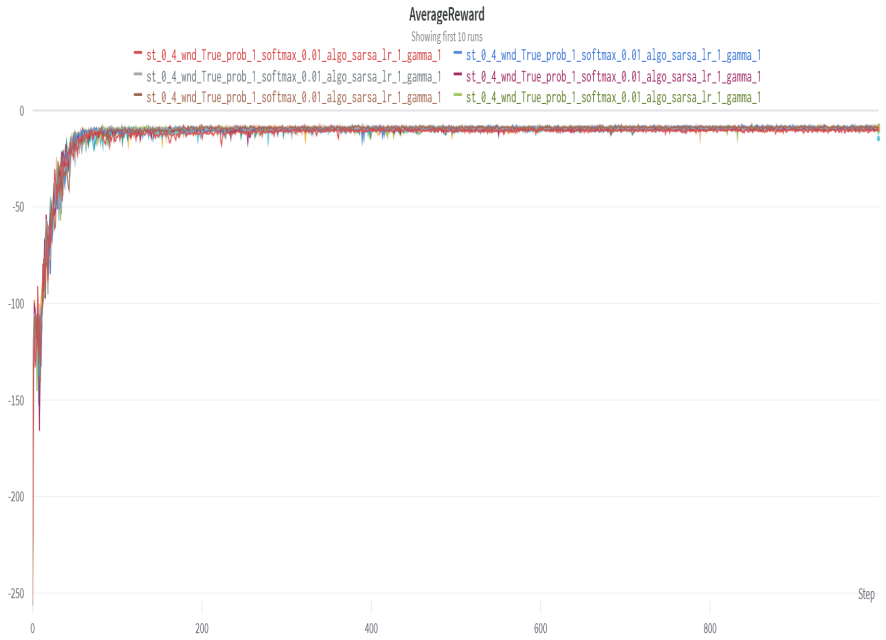

– Heatmap of the grid with Q values after training is complete:




9. **strategy=$softmax$ , start_state=(0,4), wind=True, p=1.0**

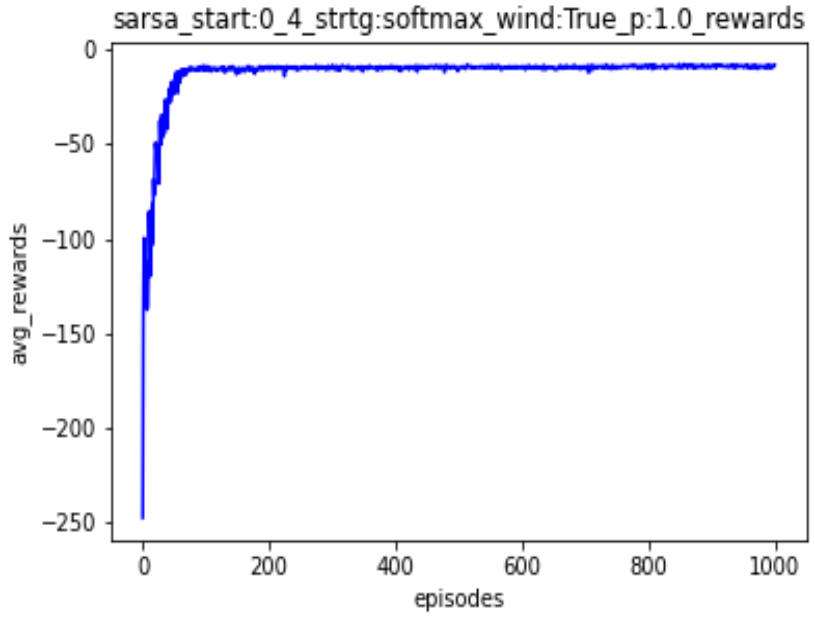
– From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 1.0$, $\beta = 0.01$

gave better performance. Following plot shows some of the best performing hyper-parameters.

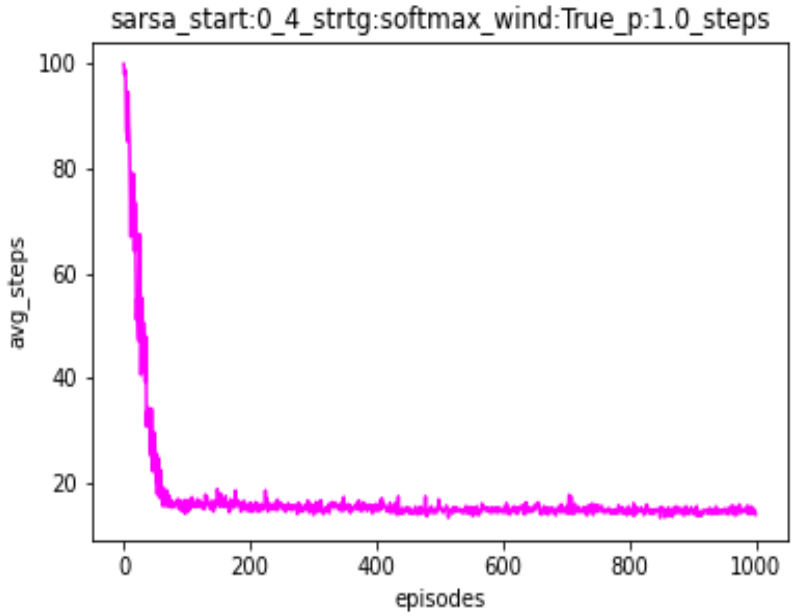



– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach the upper left corner goal. And due to wind, agent has also learnt to reach upper right corner goal.

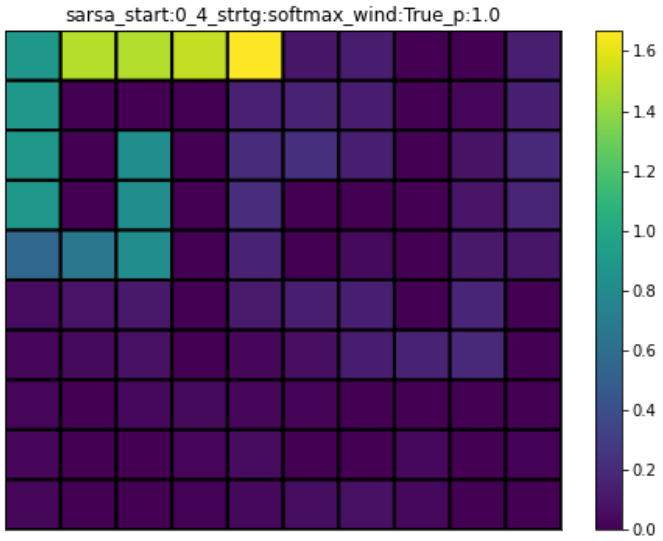
– Total reward after 1000 episodes averaged over 20 runs = -8.25, Reward curve:



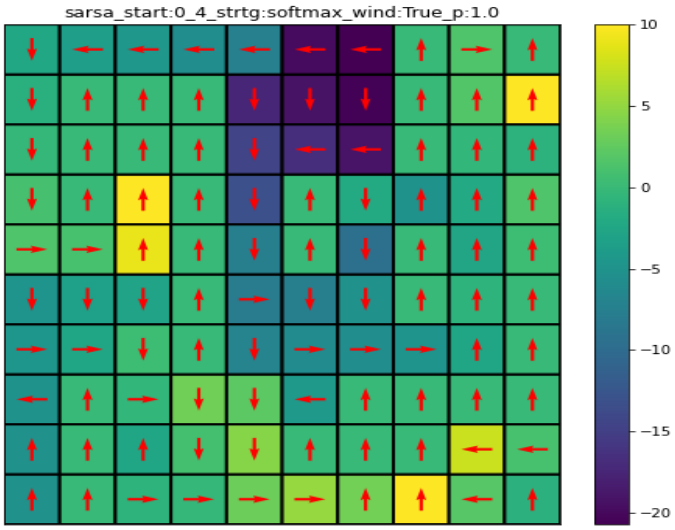
– Number steps to reach goal after 1000 episodes average over 20 runs = 14.25, Step curve:




– Heatmap of the grid with state visit count:

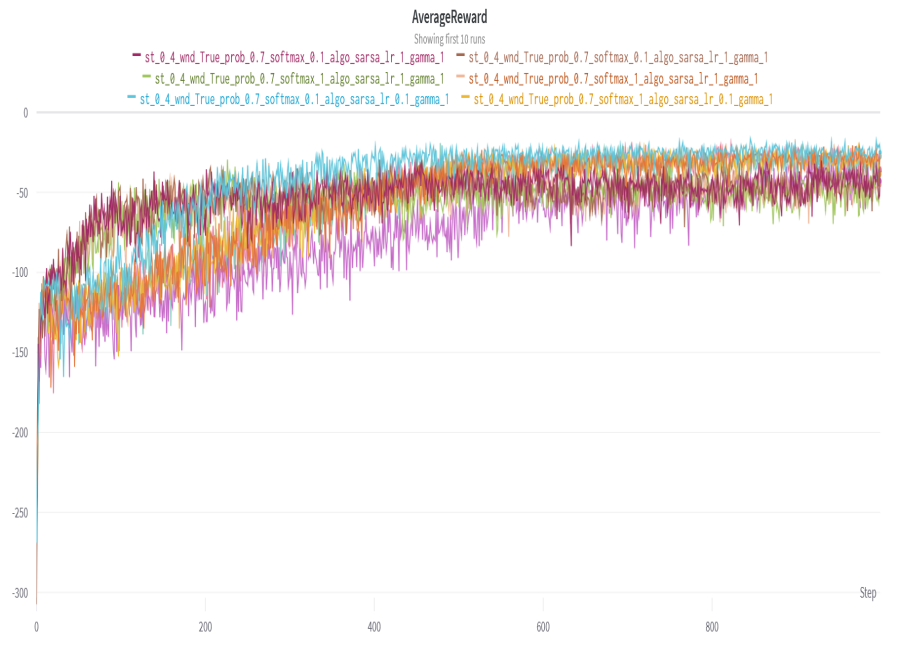



– Heatmap of the grid with Q values after training is complete:




10. **strategy=$softmax$ , start_state=(0,4), wind=True, p=0.7**

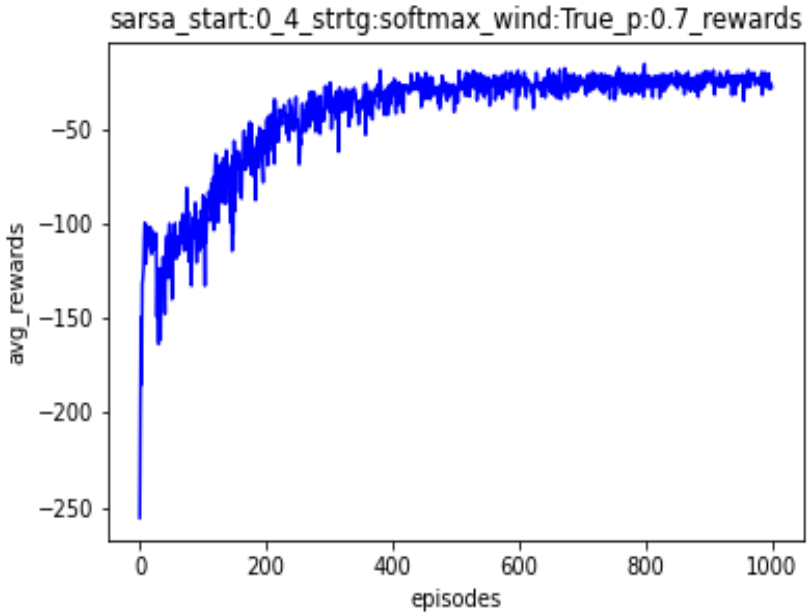
    – From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\beta = 0.1$ gave better performance. Following plot shows some of the best performing hyper-parameters.



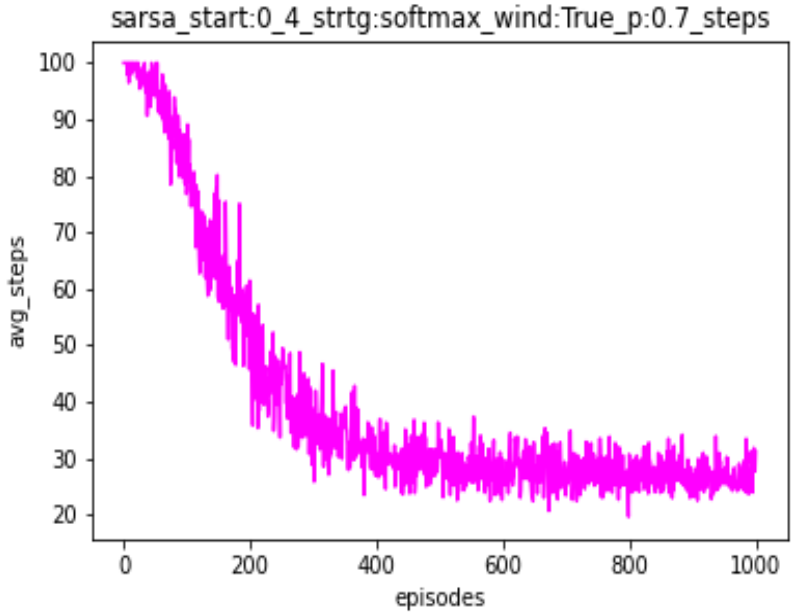

    – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach the upper left corner goal and upper right corner goal.

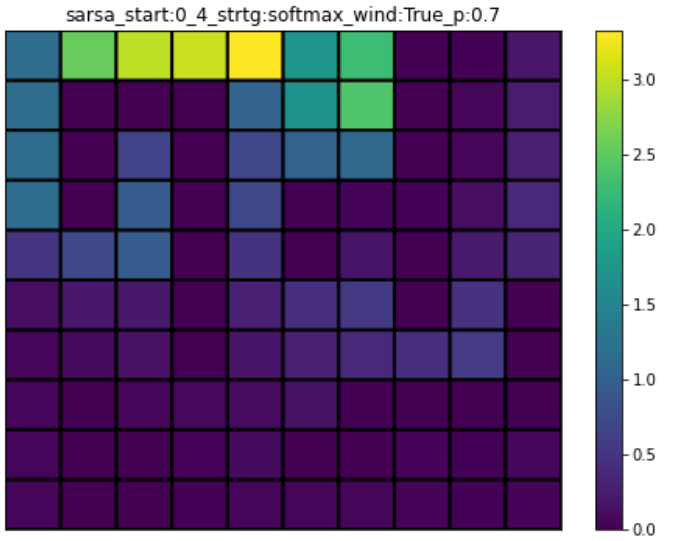
– Total reward after 1000 episodes averaged over 20 runs = -25.15, Reward curve:



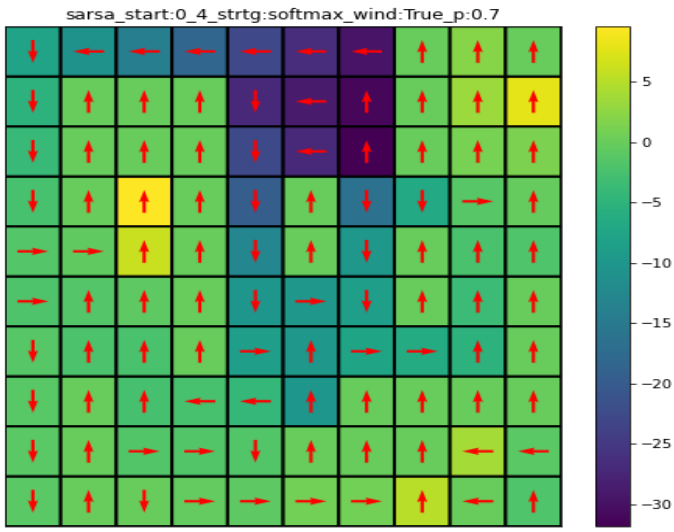

– Number steps to reach goal after 1000 episodes average over 20 runs = 27.15, Step curve:



– Heatmap of the grid with state visit count:




– Heatmap of the grid with Q values after training is complete:



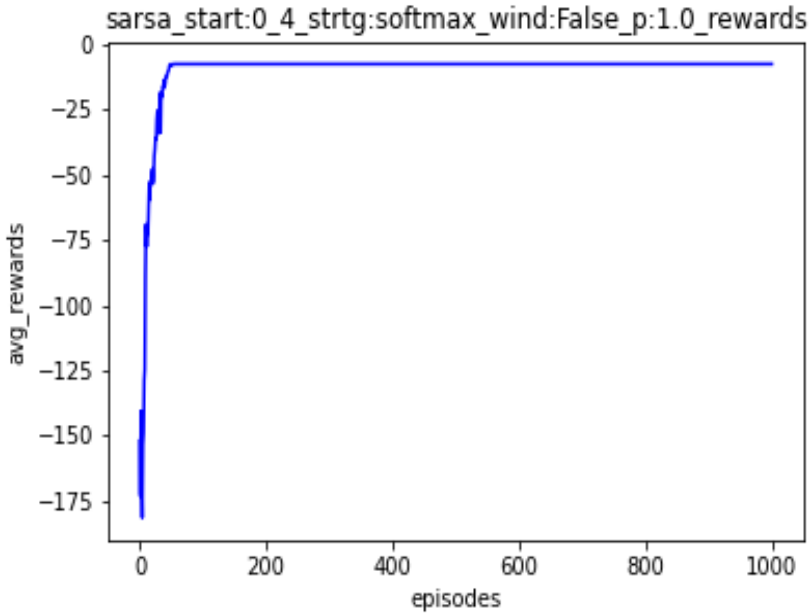

11. **strategy=*softmax* , start_state=(0,4), wind=False, p=1.0**

– From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 1$, $\beta = 0.01$ gave better performance. Following plot shows some of the best performing

hyper-parameters.

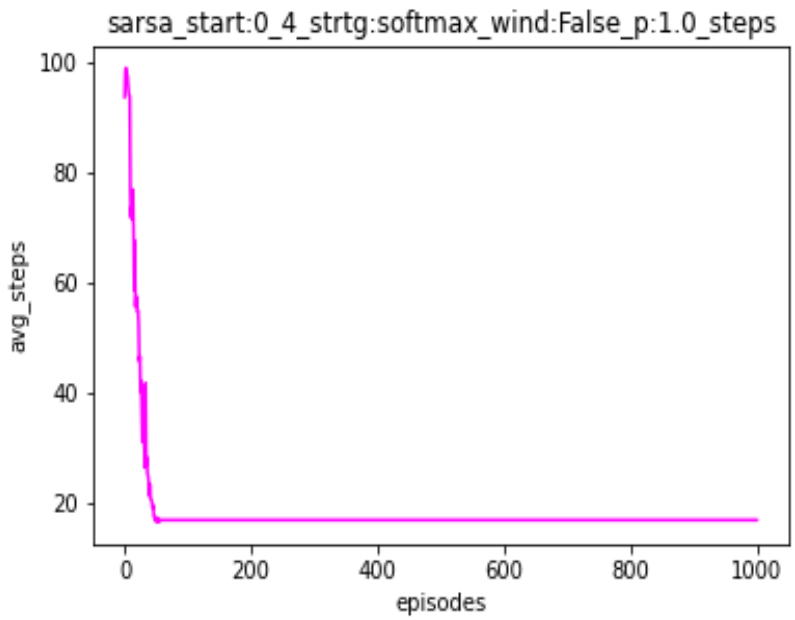

– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach lower right corner goal and due to exploration algorithm agent has also learnt to reach other goal state.

– Total reward after 1000 episodes averaged over 20 runs = -6.35, Reward curve:

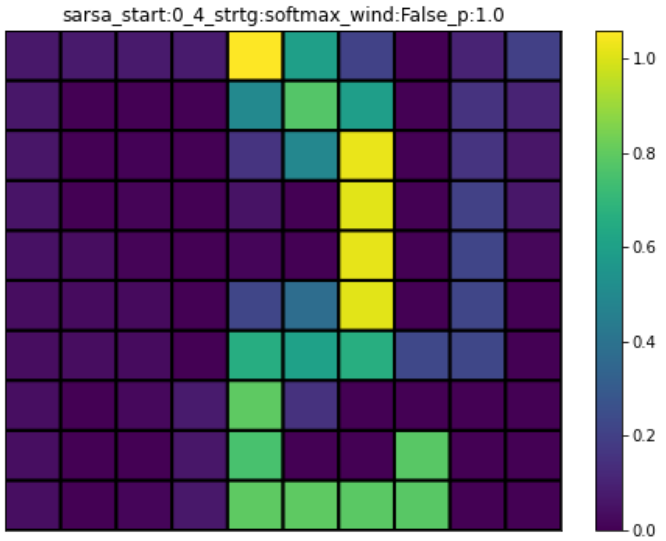



– Number steps to reach goal after 1000 episodes average over 20 runs = 16.6,

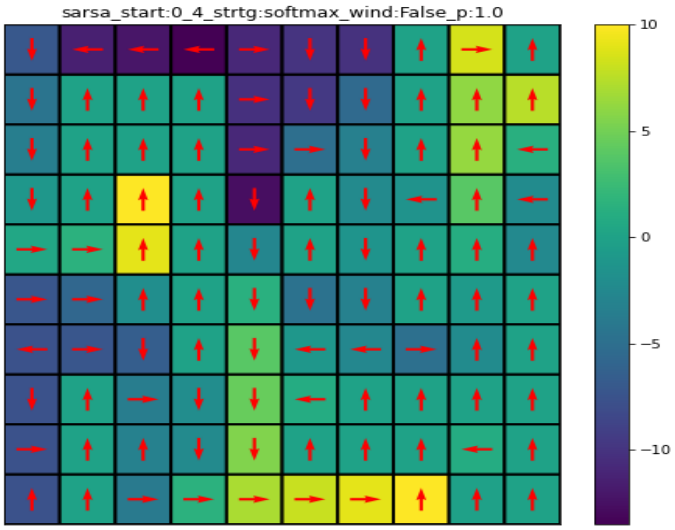
Step curve:




– Heatmap of the grid with state visit count:

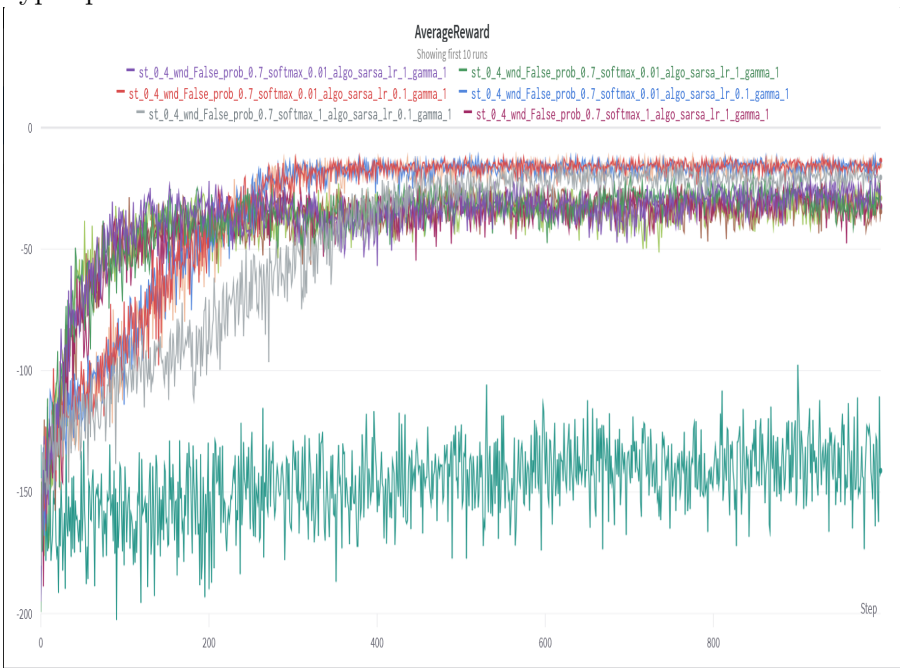



– Heatmap of the grid with Q values after training is complete:




12. **strategy=$softmax$ , start_state=(0,4), wind=False, p=0.7**

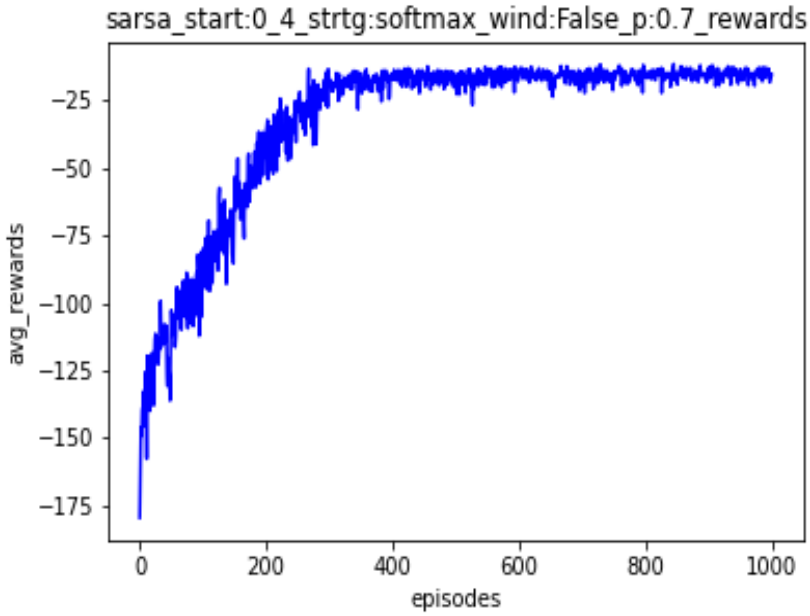
    – From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\beta = 0.01$ gave better performance. Following plot shows some of the best performing hyper-parameters.



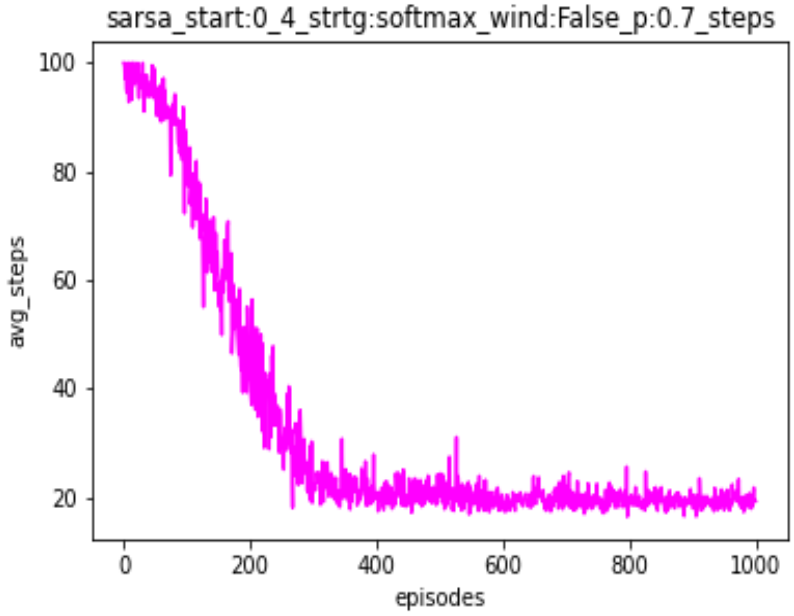

    – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach upper left corner goal and due to stochasticity in action agent has also learnt to reach upper right goal.

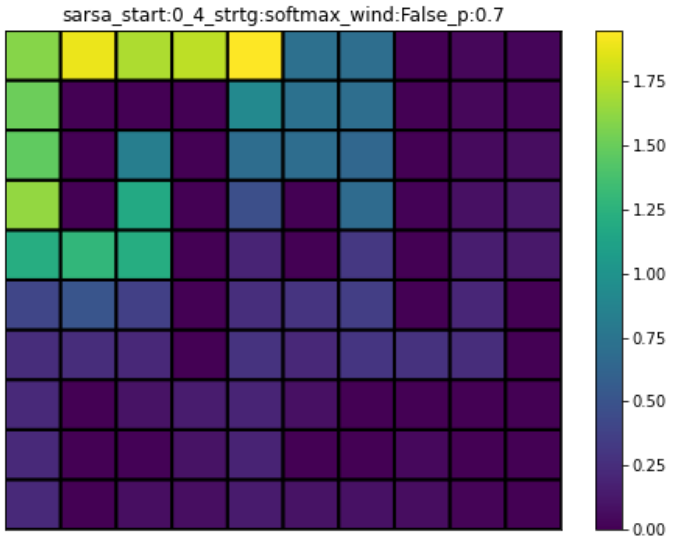
– Total reward after 1000 episodes averaged over 20 runs = -13.55, Reward curve:



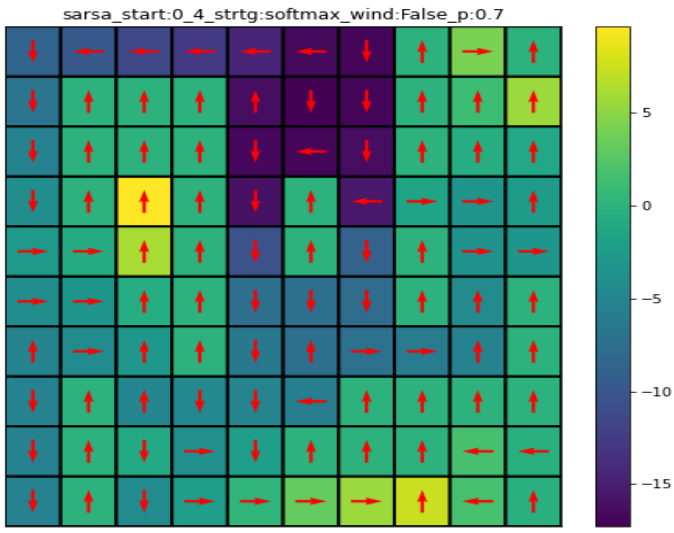

– Number steps to reach goal after 1000 episodes average over 20 runs = 18.8, Step curve:



– Heatmap of the grid with state visit count:



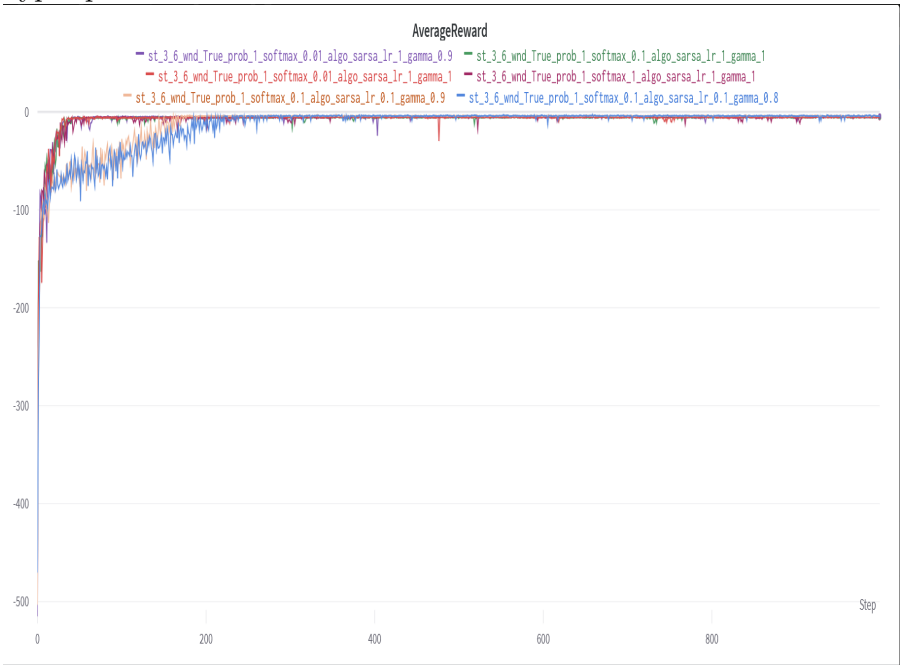

– Heatmap of the grid with Q values after training is complete:




13. **strategy=*softmax* , start_state=(3,6), wind=True, p=1.0**

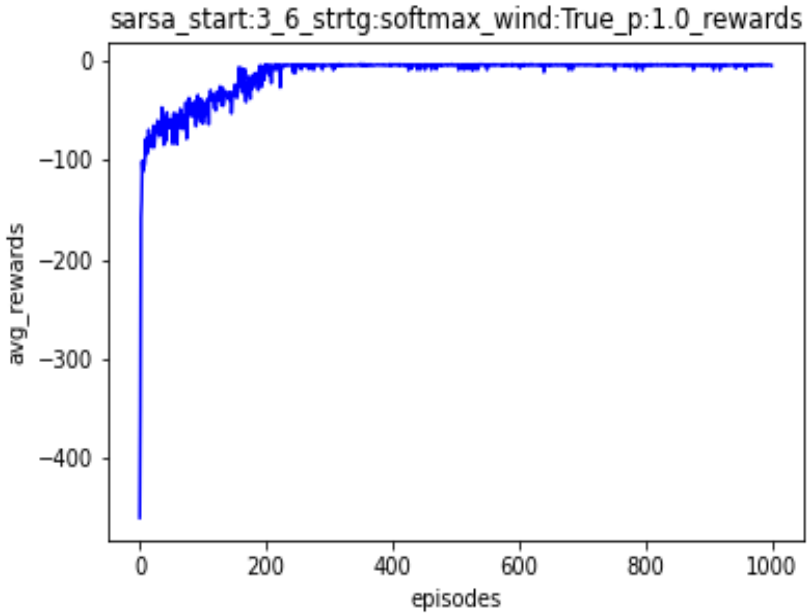
– From the experiments we performed we found that $\gamma = 0.8$, $\alpha = 0.1$, $\beta = 0.1$ gave better performance. Following plot shows some of the best performing

hyper-parameters.



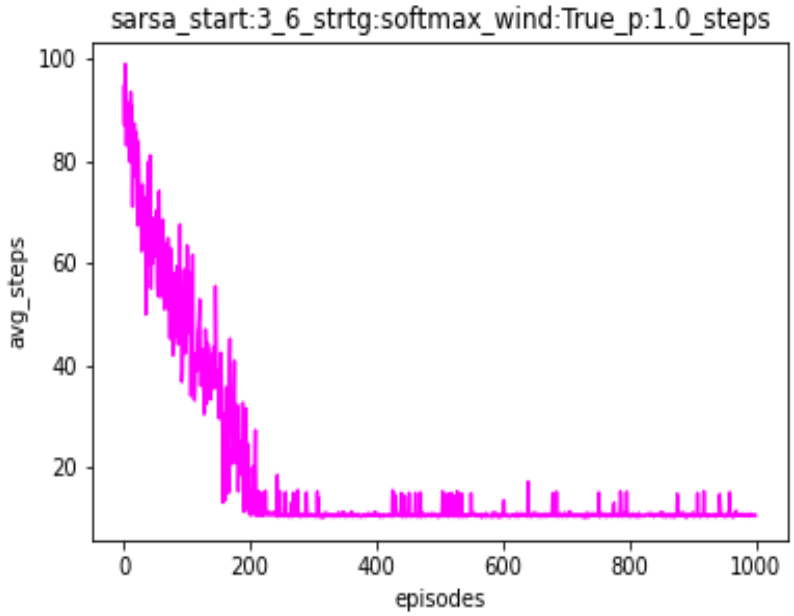

– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach upper right corner goal may be because of the wind.

– Total reward after 1000 episodes averaged over 20 runs = -4.3, Reward curve:

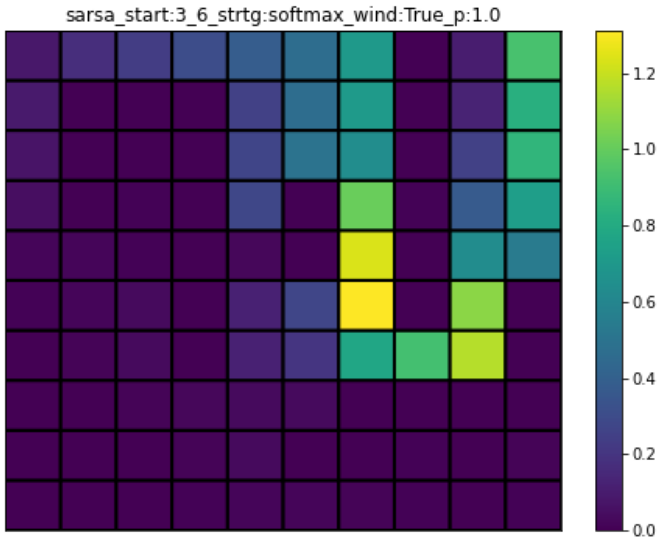



– Number steps to reach goal after 1000 episodes average over 20 runs = 10.8,

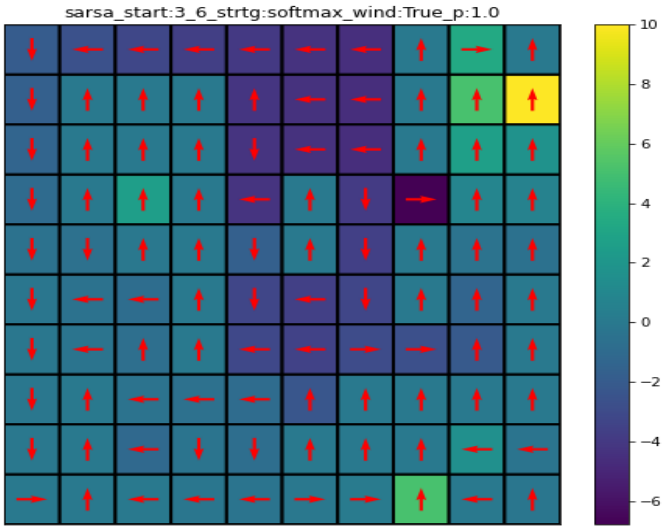
Step curve:




– Heatmap of the grid with state visit count:




– Heatmap of the grid with Q values after training is complete:



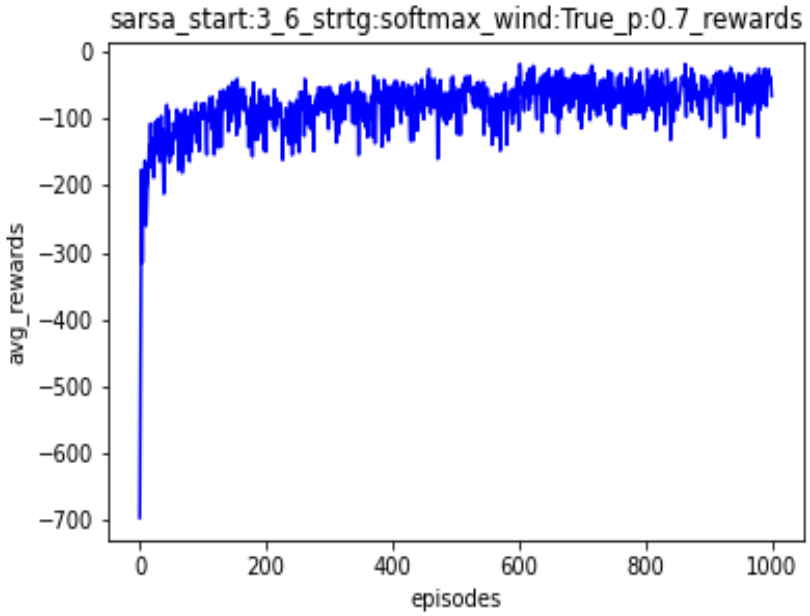

14. **strategy=$softmax$ , start_state=(3,6), wind=True, p=0.7**

    – From the experiments we performed we found that $\gamma = 0.9$, $\alpha = 0.1$, $\beta = 0.01$ gave better performance. Following plot shows some of the best performing hyper-parameters.

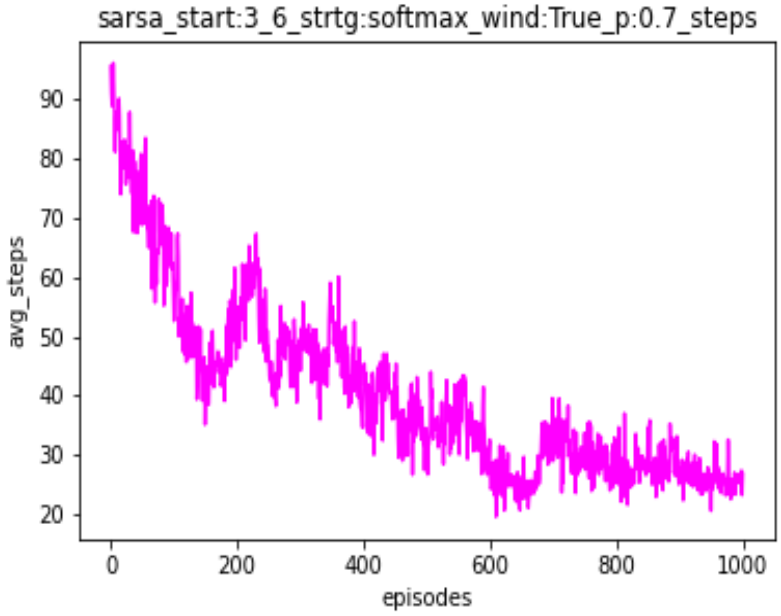

    – Policy learnt: Using the heatmap of state visit count and heatmap of Q values,

we notice that agent has learnt to reach both upper left corner goal and upper right corner goal.

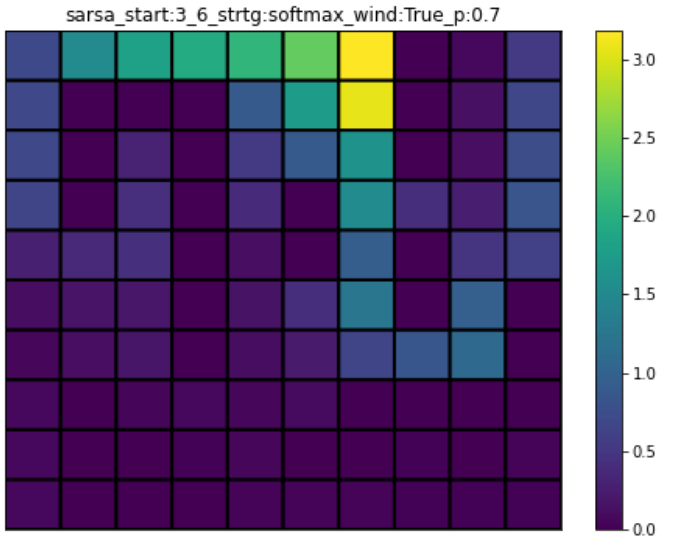
- Total reward after 1000 episodes averaged over 20 runs = -63.25, Reward curve:



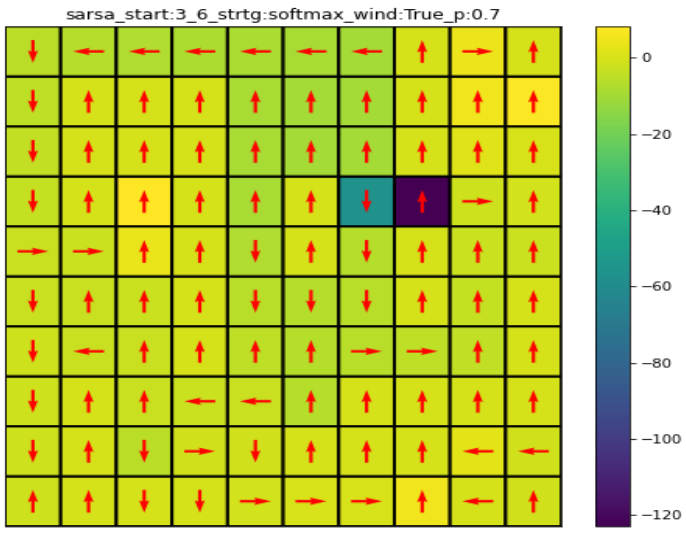

- Number steps to reach goal after 1000 episodes average over 20 runs = 33.5, Step curve:



– Heatmap of the grid with state visit count:



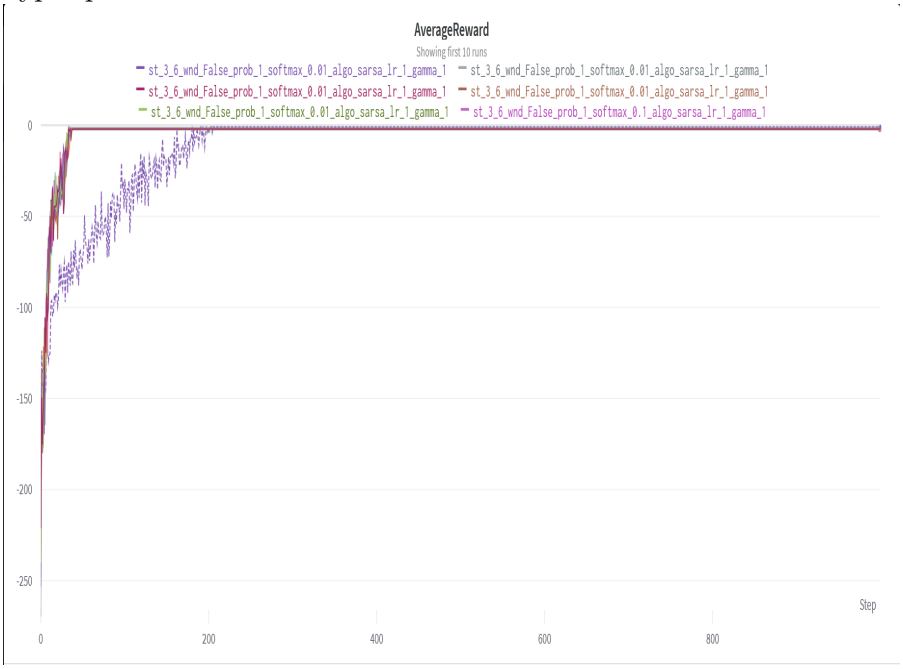

– Heatmap of the grid with Q values after training is complete:




15. **strategy=*softmax* , start_state=(3,6), wind=False, p=1.0**

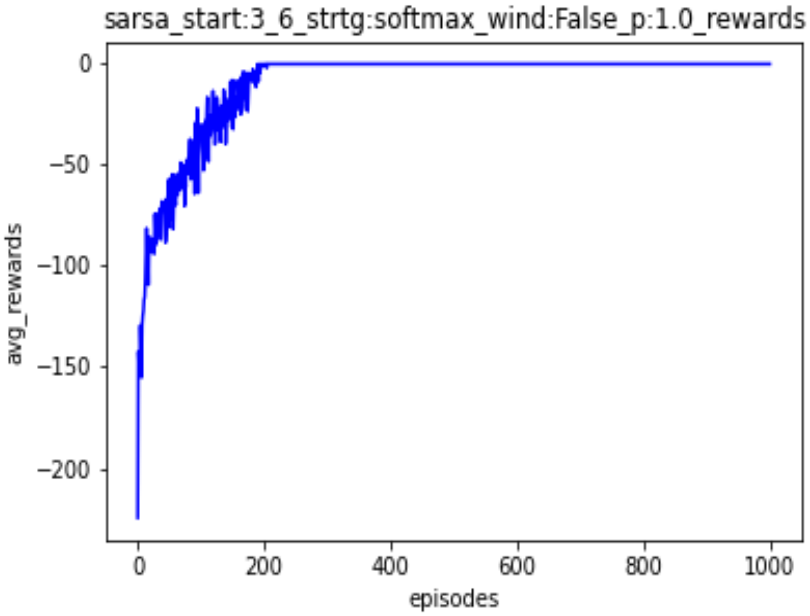
– From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\beta = 0.1$ gave better performance. Following plot shows some of the best performing

hyper-parameters.



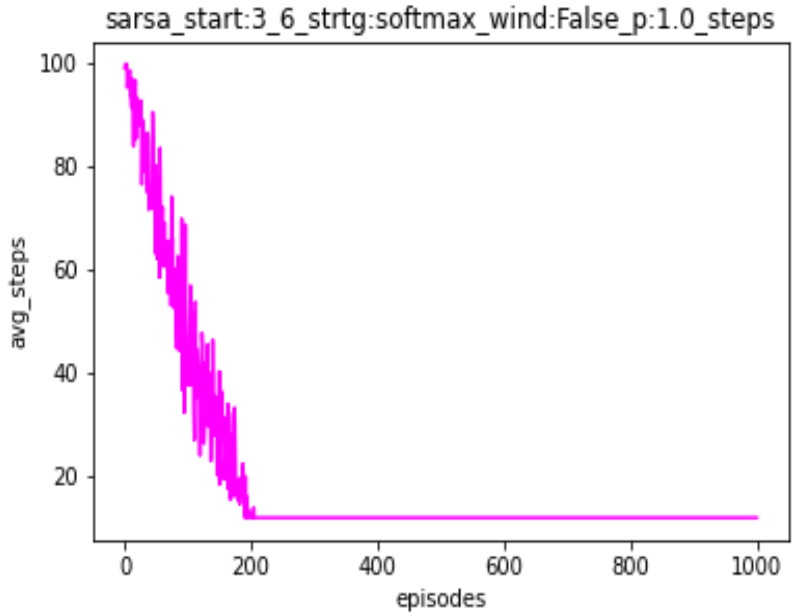

– Policy learnt: Using the heatmap of state visit count and heatmap of Q values, we notice that agent has learnt to reach lower right corner goal and due to exploration strategy agent has learnt to reach other goal state too.

– Total reward after 1000 episodes averaged over 20 runs = -1, Reward curve:

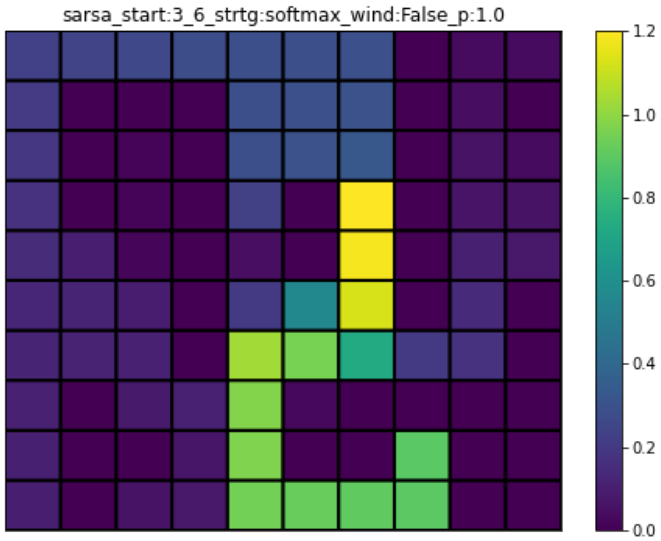



– Number steps to reach goal after 1000 episodes average over 20 runs = 12,

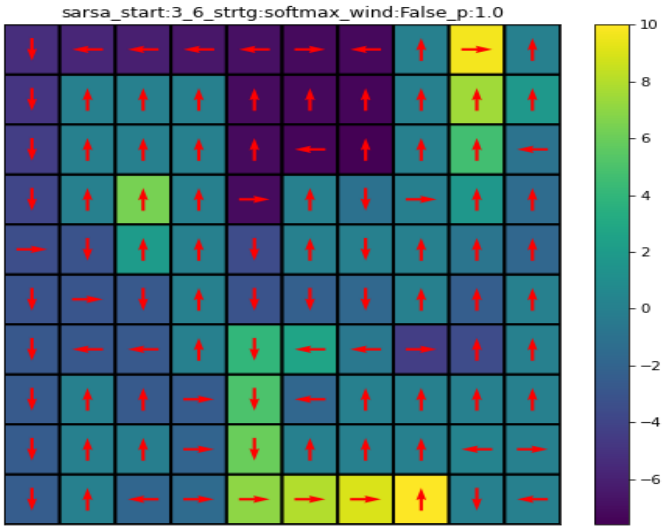
Step curve:




– Heatmap of the grid with state visit count:

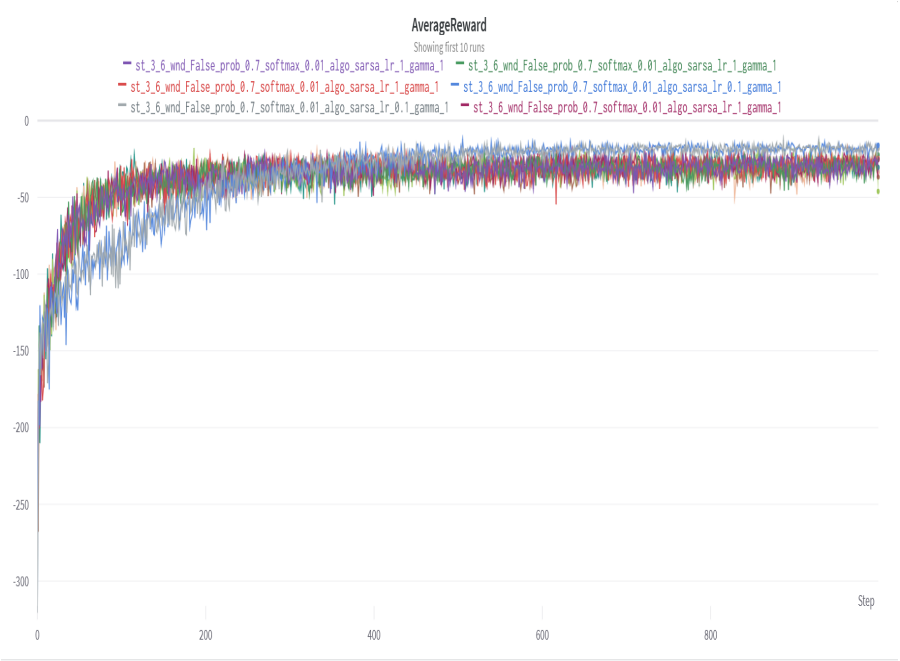



– Heatmap of the grid with Q values after training is complete:




16. **strategy=$softmax$ , start_state=(3,6), wind=False, p=0.7**

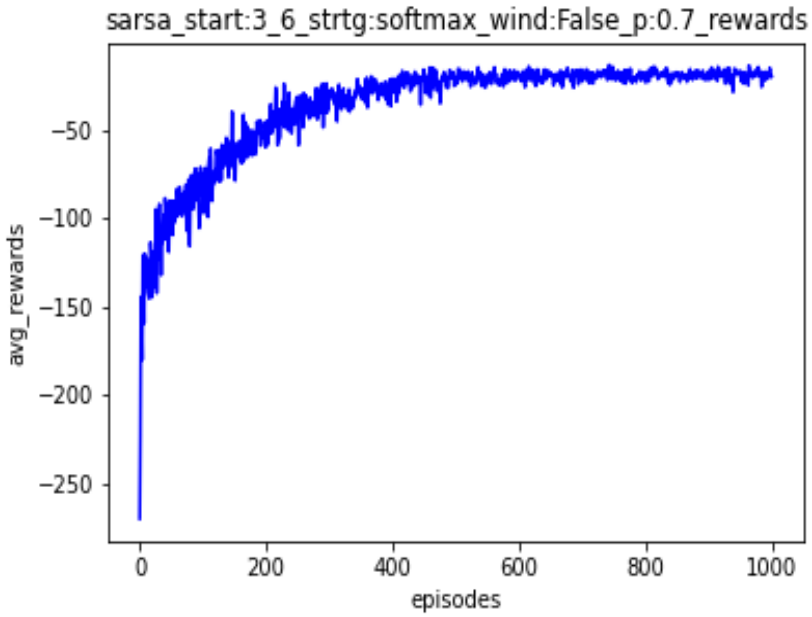
    – From the experiments we performed we found that $\gamma = 1.0$, $\alpha = 0.1$, $\beta = 0.01$ gave better performance. Following plot shows some of the best performing hyper-parameters.



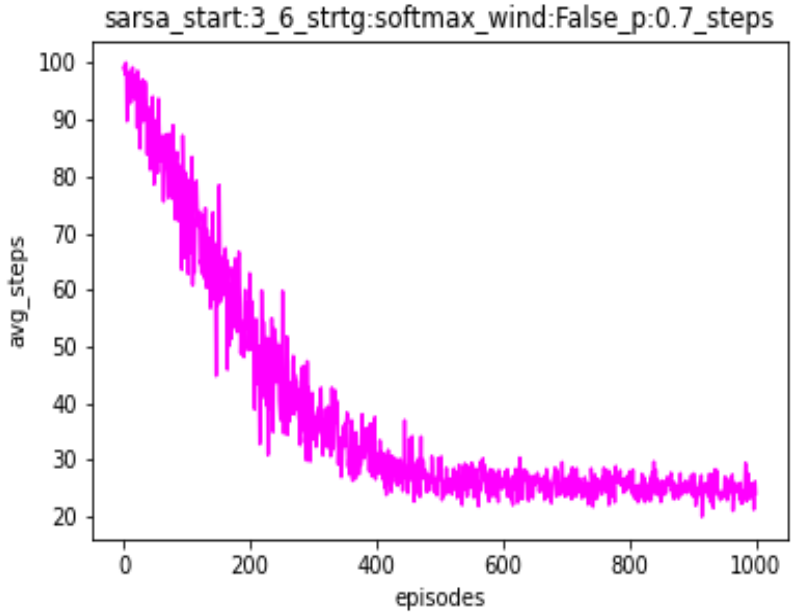

    – Policy learnt: Using the heatmap of state visit count and heatmap of Q

values, we notice that agent has learnt to reach all the goal state may be due to stochasticity in the action has enabled agents to learn different policies to reach goal state.

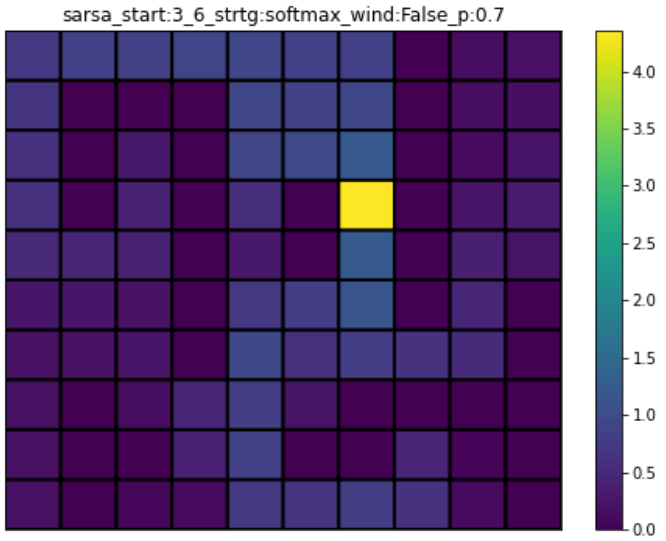
– Total reward after 1000 episodes averaged over 20 runs = -16.15, Reward curve:




– Number steps to reach goal after 1000 episodes average over 20 runs = 21.65, Step curve:



– Heatmap of the grid with state visit count:




– Heatmap of the grid with Q values after training is complete: