Programming Assignment 1

Purvam Jain EE20B101, Martin Reich ME23F201

November 13, 2023

Contents

1 Abstract **2**

2 Hyper-Parameter tuning **2**
2.1 Termination Criteria 3

3 SARSA **4**
3.1 ϵ-Greedy Policy 4
3.2 Softmax Policy 13

4 QLearning **22**
4.1 ϵ-Greedy Policy 22
4.2 Softmax Policy 31

1 Abstract

This report is structured into three parts. The first part contains general information and an explanation of the approach taken. The second section contains descriptions about the policies learnt with SARSA, while the third part contains the same for Q-Learning.
Both algorithms were trained with eight different combinations of environment parameters, thus leading to a total of 16 combinations as:

Wind	True	False
Start State	[0,4]	[3,6]
Transition probability	0.7	1.0
Policy	ϵ-greedy	Softmax

Table 1: Environmental Parameters

2 Hyper-Parameter tuning

To determine the best set of hyper-parameters for different settings and to avoid a combinatorial explosion of parameters leading to longer computational time, a coarse set of hyper-parameters was chosen with respect to the hyper-parameters characteristics. After this coarse search a fine tuning in the intervals surrounding the best set of hyper-parameters could be done, as these give a tendency towards the actual best hyper-parameters. Specifically, three values were picked each, leading to 27 different combinations as:

Gamma (γ)	0.999	0.75	0.1
Alpha (α)	0.1	0.25	1.0
Epsilon (ϵ)	0	0.01	0.1
Temperature (τ)	0.5	1.0	1.5

Table 2: List of Hyper-parameters for search

We used Weights&Biases platform to perform the tuning for the model. All possible combinations were sweeped over to determine the effect of respective hyper-parameters as well as their correlation. We used three metrics to evaluate the performance, viz., rate of convergence, mean rewards and mean steps. These were averaged over three experimental runs by default for 250 episodes to account for randomness. The following observations were made:

1. **Gamma (γ)**: The discount rate, a parameter in the interval [0, 1], part of both SARSA and Q-Learning and describes the influence of future rewards to the present reward. The bigger the value, the more farsighted the algorithm learns, while the smaller the value, the more emphasis is placed on the immediate reward. We found that irrespective of other variables

a value close to 1, but not equal to 1 always gave better results which is to be expected since if the discount factor is smaller it will only pay attention to immediate rewards while if it's equal to or greater than 1, it acts too optimistic and might also lead to diverging actions. And hence, future runs were performed with a gamma value close to 1 discarding other values.

2. **Alpha (α)**: Learning rate, also a parameter in the range [0, 1], is part of both the learning algorithms. It guides the amount by which the Q-values should be updated. A large value suggests an aggressive learning pattern while a smaller learning rate requires a lot of updates to correct its estimate. Further variation in this parameter is discussed with respective settings. Although ideally, it should be decayed from a higher value to a smaller value as we reach the optimal policy.

3. **Epsilon (ϵ)**: The parameter for the ϵ-greedy policy. It controls the probability of taking a random action instead of the action with the biggest expected reward and is, therefore, part of the exploit and explore dilemma. The smaller the value, the lower the probability of taking another action than the optimal one, while the bigger the value the more random the policy is. Ideally, the epsilon value should be decayed from a higher value with training to take advantage of exploration phase at start and subsequently exploit the learned policy in later stages to reach optimal policy.

4. **Temperature (τ)**: The parameter for softmax policy. The temperature value is inversely proportional to model's confidence. A high temperature penalizes the actions with bigger probabilities more than the ones with smaller probabilities and hence results in a softer distribution to choose action from. While a lower temperature increases the confidence by amplifying probabilities. Empirically it is found that our models converge faster with tau values less than or equal to 1 since it increases the frequency of choosing action decided by the policy.

Weights&Biases report can be accessed through this link:here.

2.1 Termination Criteria

For termination criteria, we look at the rolling average of rewards after a minimum episode length of 1000. We compare the absolute difference in average reward for last 100 and 10 epsiodes to be less than some constant $stop\_crit = 0.01$ to check for convergence of values.

3 SARSA

3.1 ϵ-Greedy Policy

All experiments were run with $\epsilon = 0.01$, as mentioned in above section.

Wind	p_trans	start-state	learning rate	Episodes
True	1.0	[0, 4]	0.25(1.0)	1285
True	1.0	[3, 6]	0.25(1.0)	1028
True	0.7	[0, 4]	0.25	1145
True	0.7	[3, 6]	0.25	1281
False	1.0	[0, 4]	0.25(1.0)	1198
False	1.0	[3, 6]	0.25(1.0)	1626
False	0.7	[0, 4]	0.25	1401
False	0.7	[3, 6]	0.25	1341

Table 3: SARSA with ϵ-greedy policy

So, it's observed that for for most cases a learning rate of 0.25 is better than either 1.0 or 0.1. Although in some cases, where 1 is mentioned in brackets, using a learning rate of 1 leads to faster convergence but at a slightly lower value of reward as compared to 0.25. Hence, it's assumed that a more optimal learning rate can be found between 0.25 and 1.0. And it should be noted that 1.0 is proving to be a better learning rate in cases where transtion probability is 1 and/or wind parameter is set to False, that is the randomness in the environment is decreased hence model can be more confident in leaping with a larger learning rate and converge faster. Plots for different cases are visualised below.

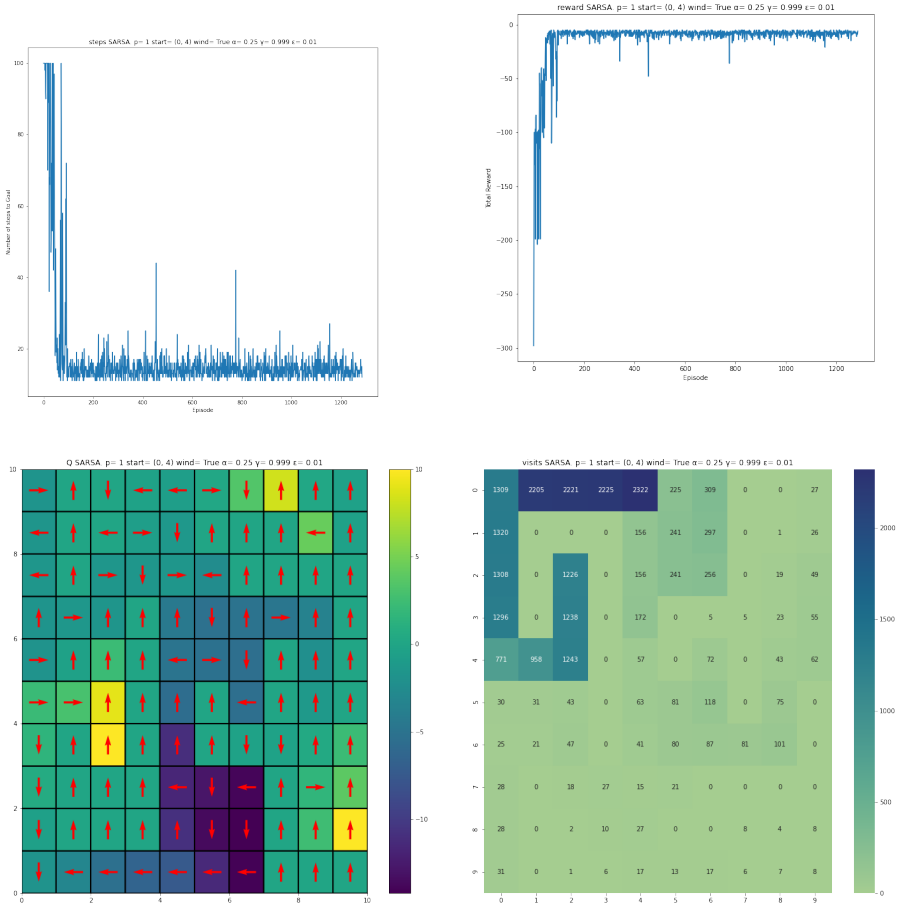

Figure 1: SARSA. p= 1 start= (0, 4) wind= True α= 0.25 γ= 0.999 ϵ= 0.01

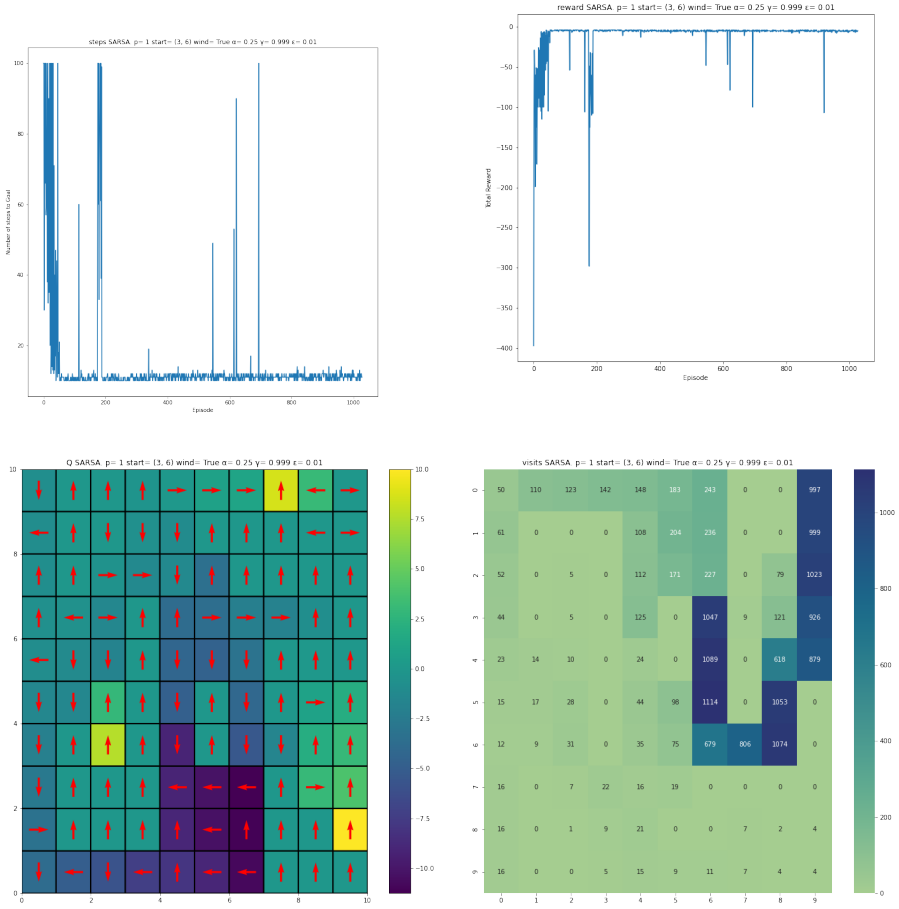

Figure 2: SARSA. p= 1 start= (3, 6) wind= True α= 0.25 γ= 0.999 ϵ= 0.01

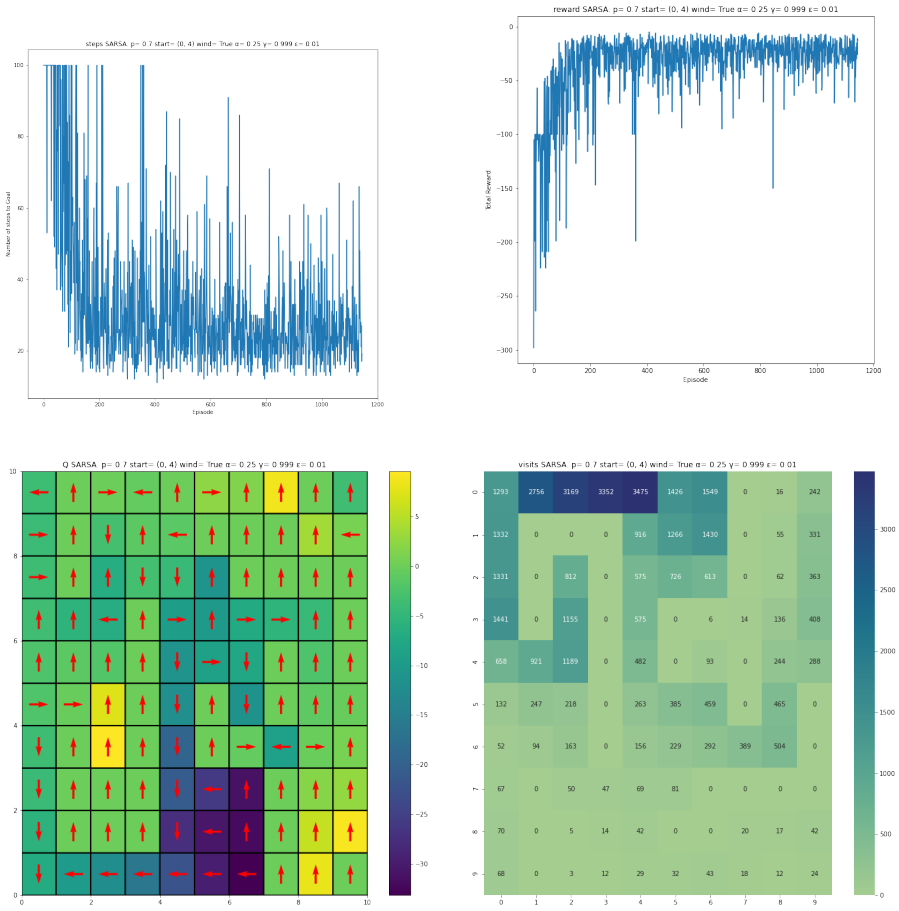

Figure 3: SARSA. p= 0.7 start= (0, 4) wind= True α= 0.25 γ= 0.999 ϵ= 0.01

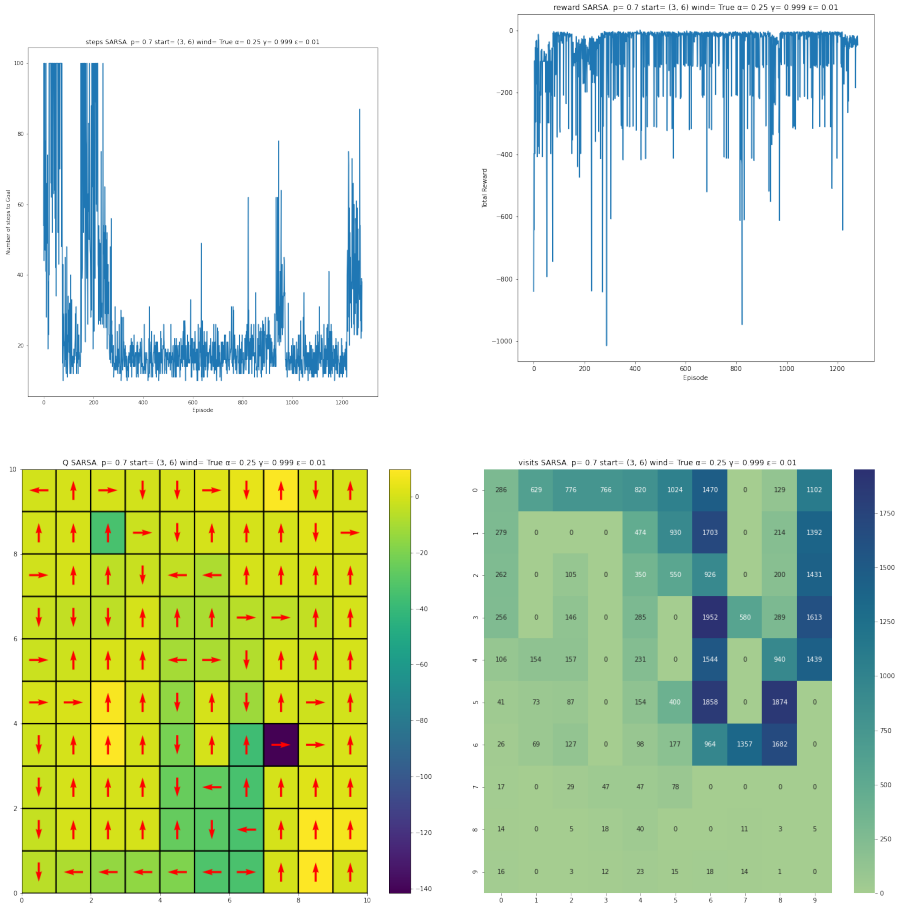

Figure 4: SARSA. p= 0.7 start= (3, 6) wind= True α= 0.25 γ= 0.999 ϵ= 0.01

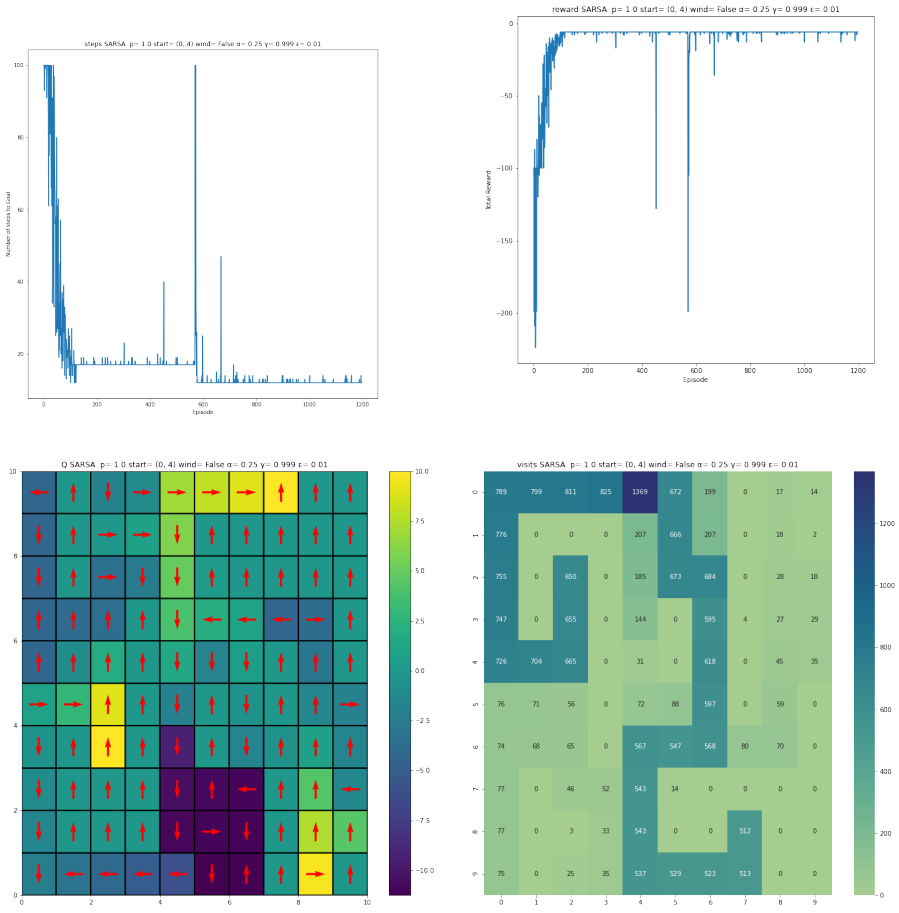

Figure 5: SARSA. p= 1.0 start= (0, 4) wind= False α= 0.25 γ= 0.999 ϵ= 0.01

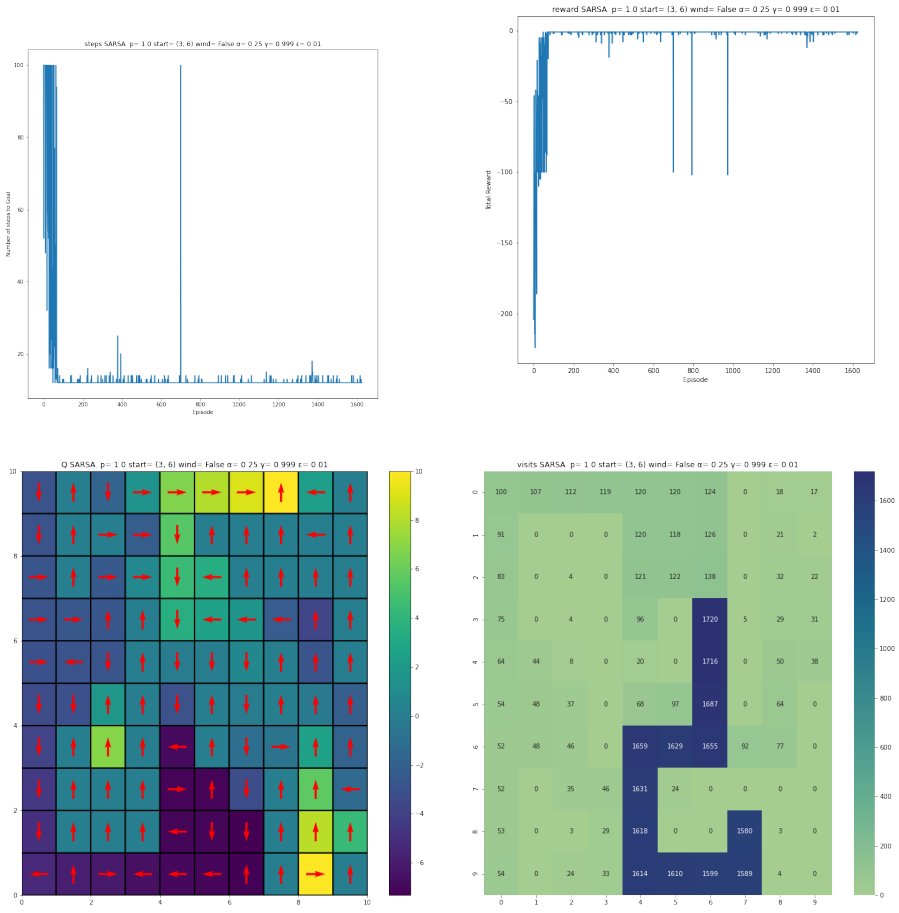

Figure 6: SARSA. p= 1.0 start= (3, 6) wind= False α= 0.25 γ= 0.999 ϵ= 0.01

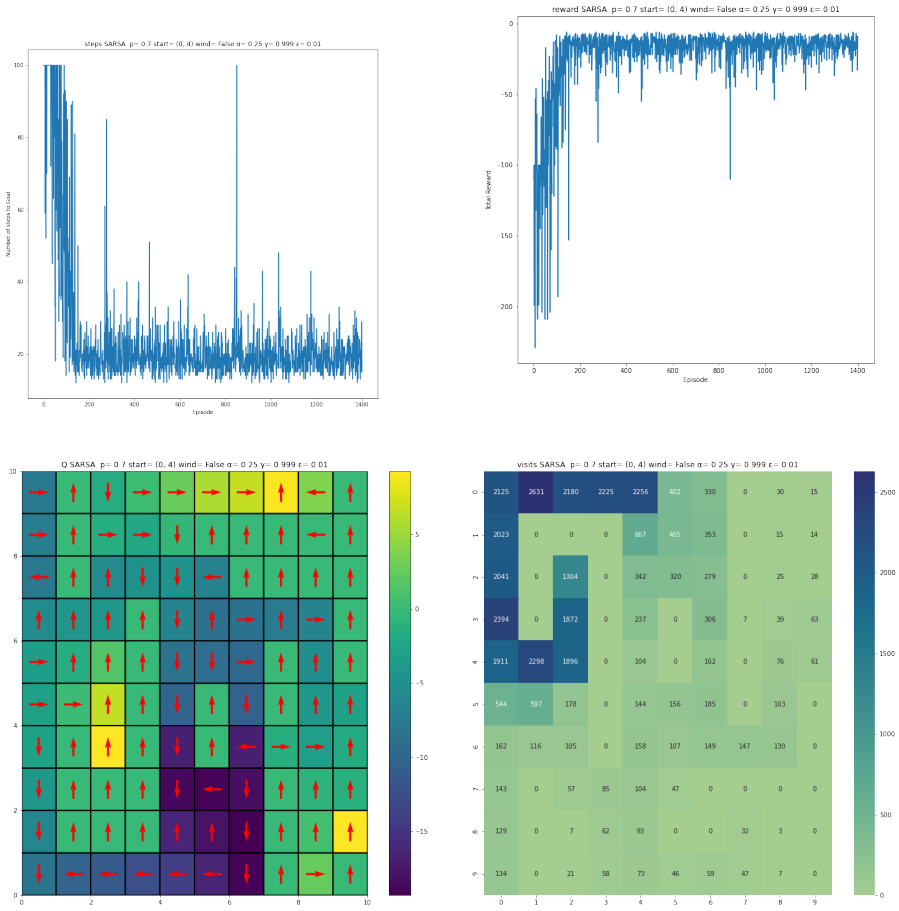

Figure 7: SARSA. p= 0.7 start= (0, 4) wind= False α= 0.25 γ= 0.999 ϵ= 0.01

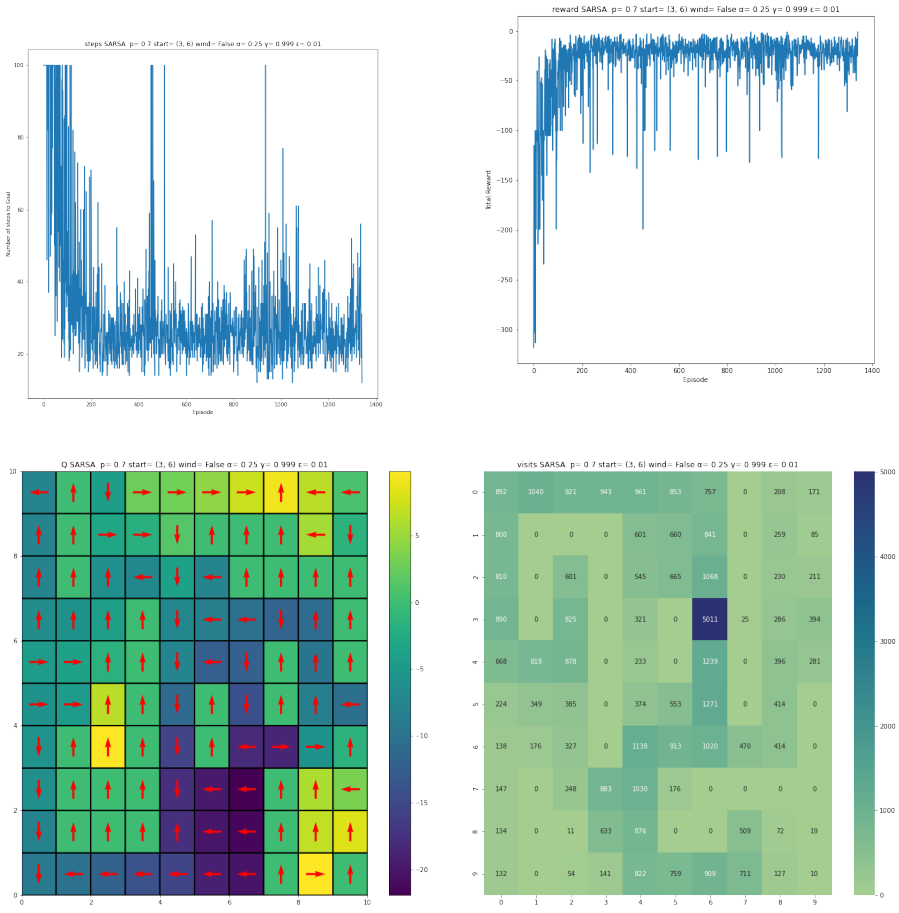

Figure 8: SARSA. p= 0.7 start= (3, 6) wind= False α= 0.25 γ= 0.999 ϵ= 0.01

3.2 Softmax Policy

All experiments were implemented with temperature = 0.5, because it was empirically observed that it lead to faster convergence, which can be understood that we increase model's confidence by increasing probability as discussed in hyperparameters section.

Wind	p_trans	start-state	learning rate	Temperature	Episodes
True	1.0	[0, 4]	1.0	0.5	1088
True	1.0	[3, 6]	1.0	0.5	1010
True	0.7	[0, 4]	0.25	0.5	1817
True	0.7	[3, 6]	0.25	0.5	1333
False	1.0	[0, 4]	1.0	0.5	1001
False	1.0	[3, 6]	1.0	0.5	1001
False	0.7	[0, 4]	1.0	0.5	1073
False	0.7	[3, 6]	0.25	0.5	1396

Table 4: SARSA with softmax policy

Now, although in most instances learning rate of 1 seemed to perform better for softmax policy, for instances where there was induced stochasticity due to either wind and/or transition probability a slower learning rate was preferred which is obvious as model is often taken to a random state than chosen one.

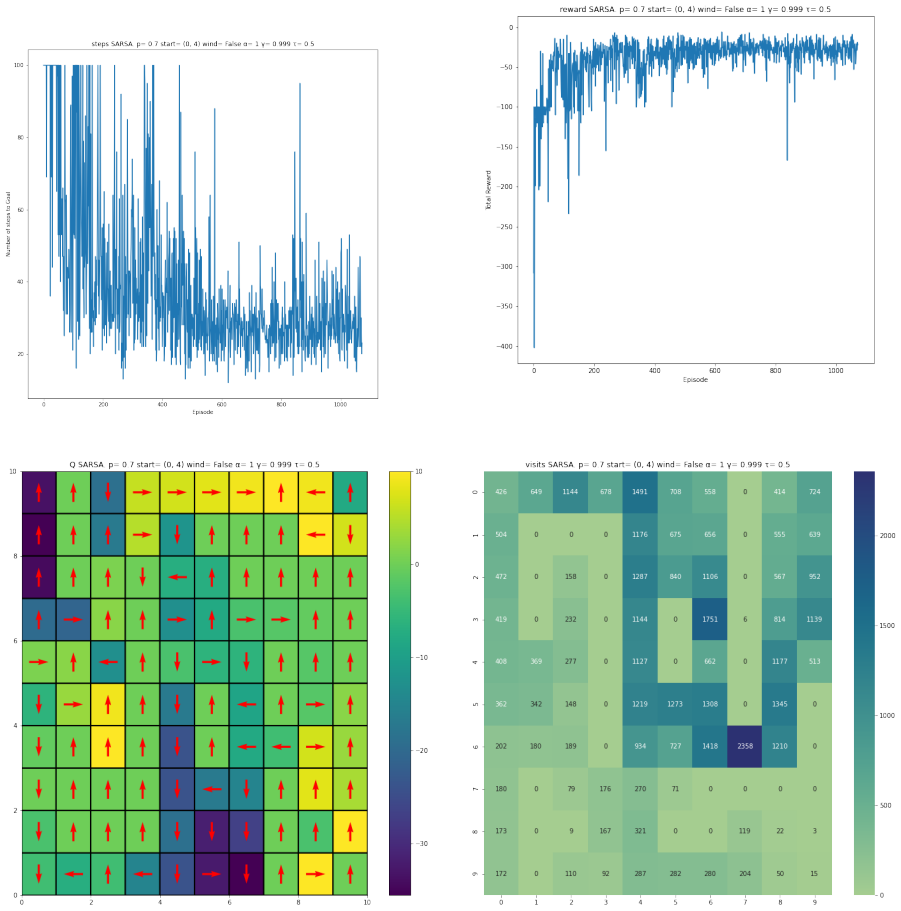

Figure 9: SARSA. p= 0.7 start= (0, 4) wind= True α= 1 γ= 0.999 τ= 0.5

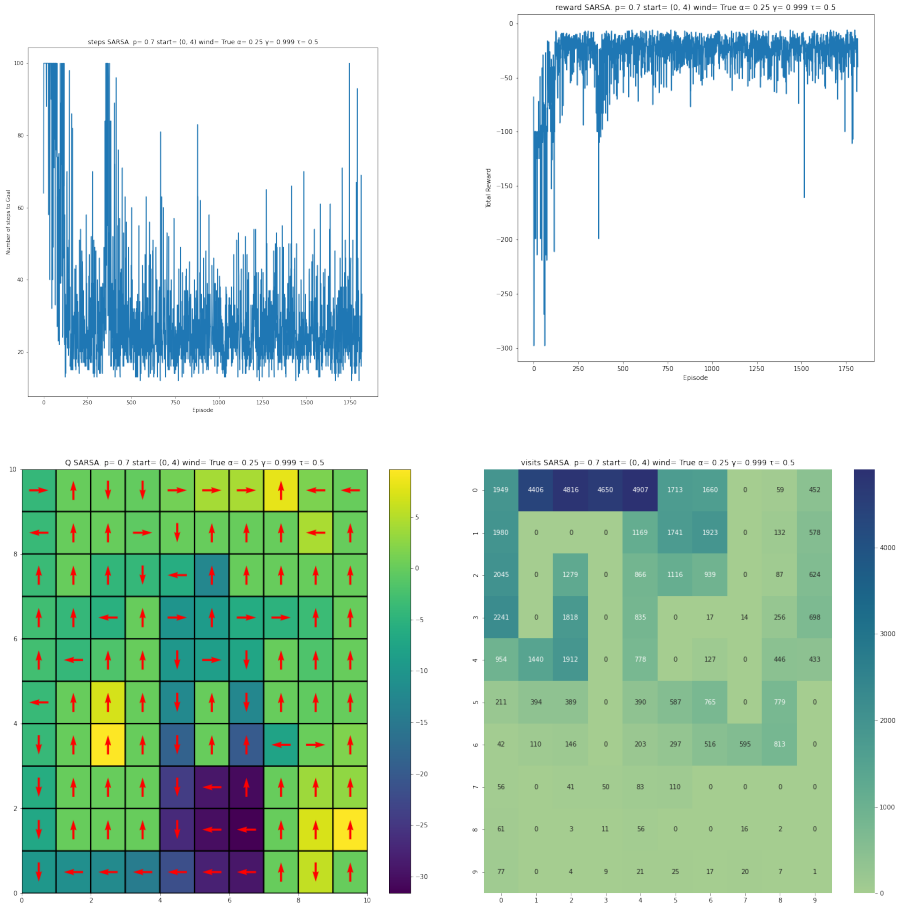

Figure 10: SARSA. p= 0.7 start= (0, 4) wind= True α= 0.25 γ= 0.999 τ= 0.5

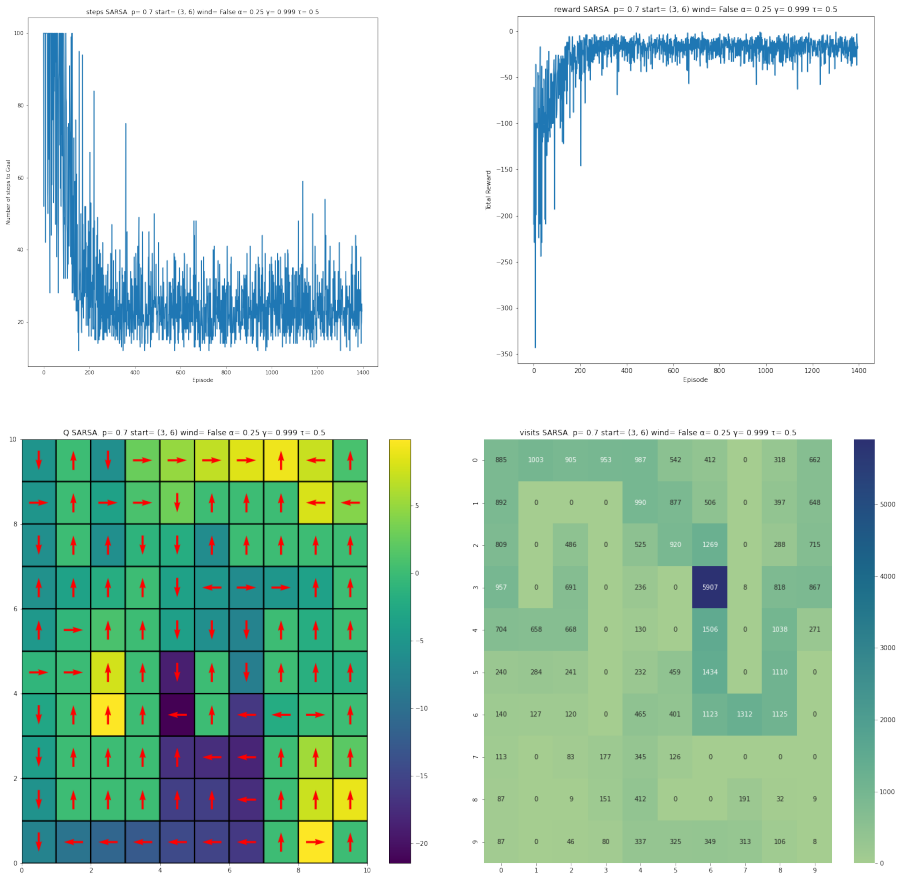

Figure 11: SARSA. p= 0.7 start= (3, 6) wind= False α= 0.25 γ= 0.999 τ= 0.5

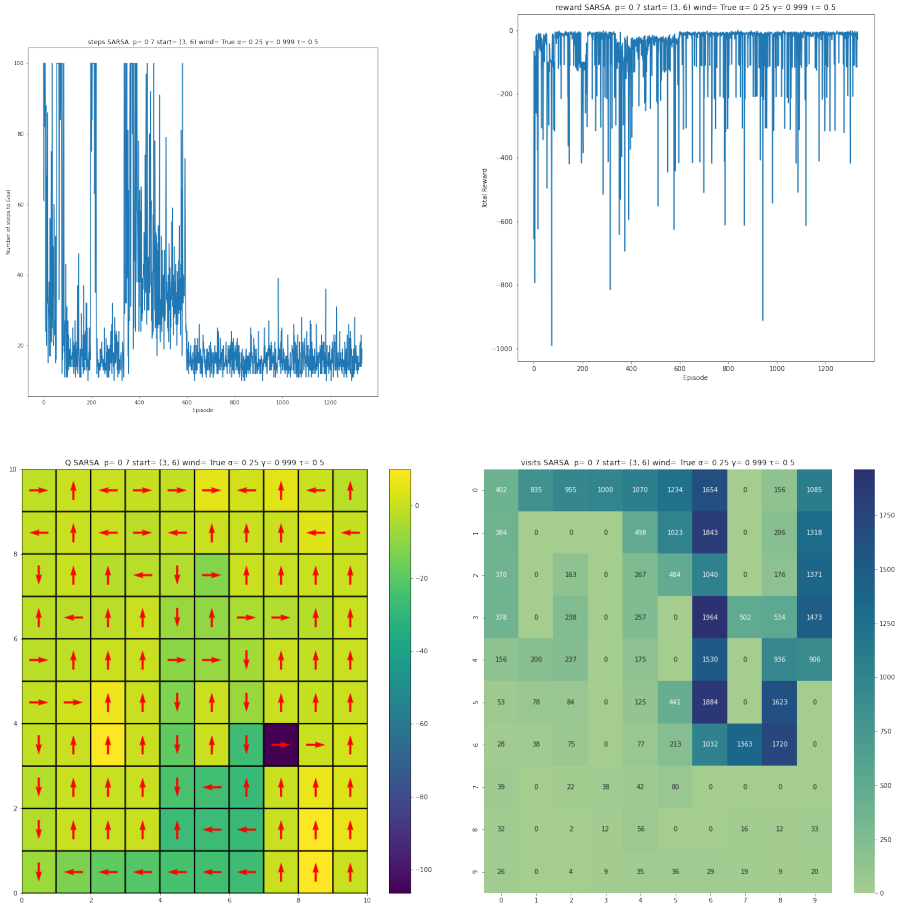

Figure 12: SARSA. p= 0.7 start= (3, 6) wind= True α= 0.25 γ= 0.999 τ= 0.5

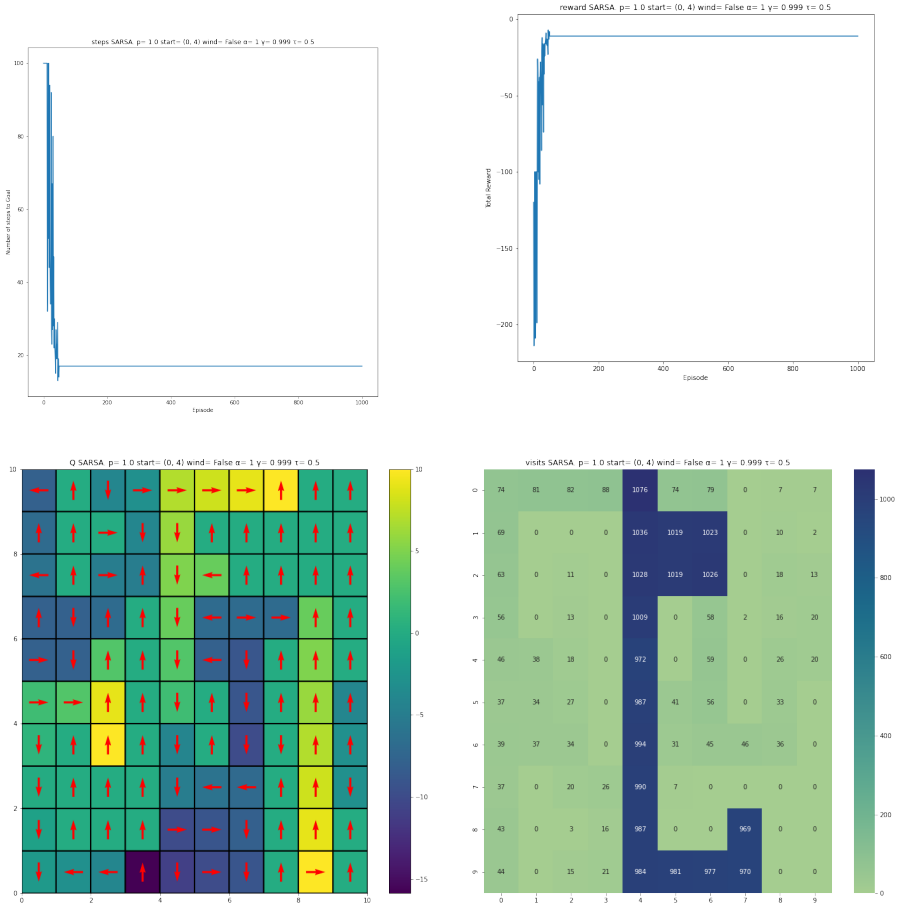

Figure 13: SARSA. p= 1.0 start= (0, 4) wind= False α= 1 γ= 0.999 τ= 0.5

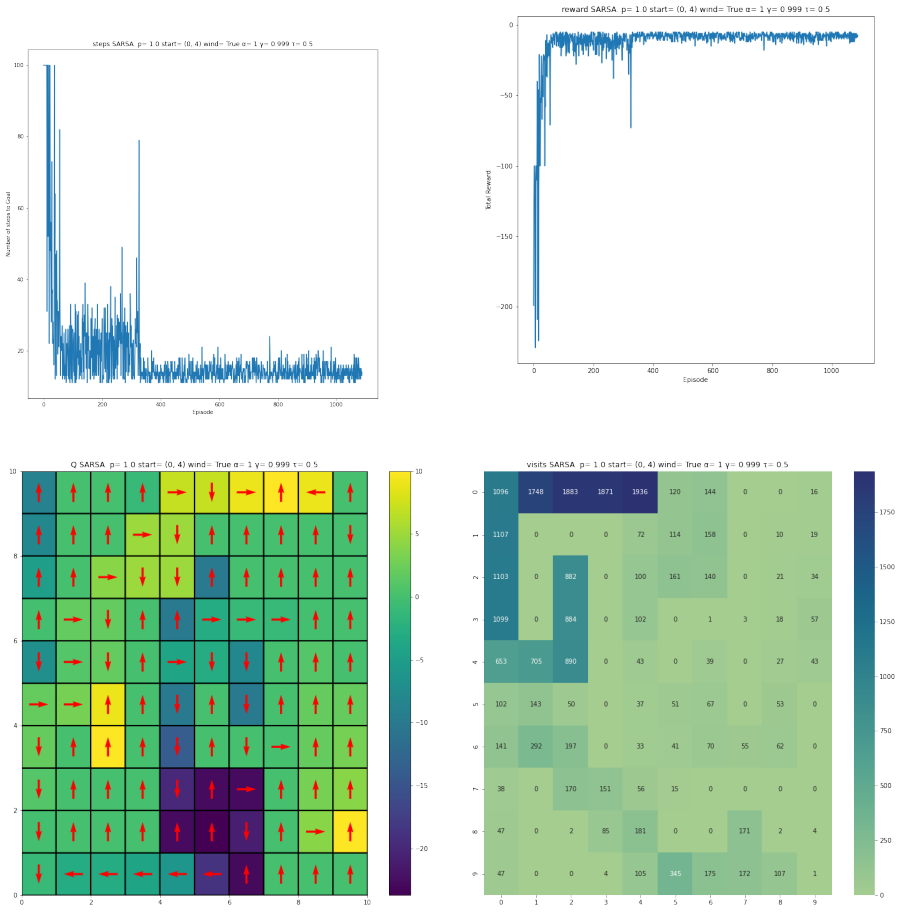

Figure 14: SARSA. p= 1.0 start= (0, 4) wind= True α= 1 γ= 0.999 τ= 0.5

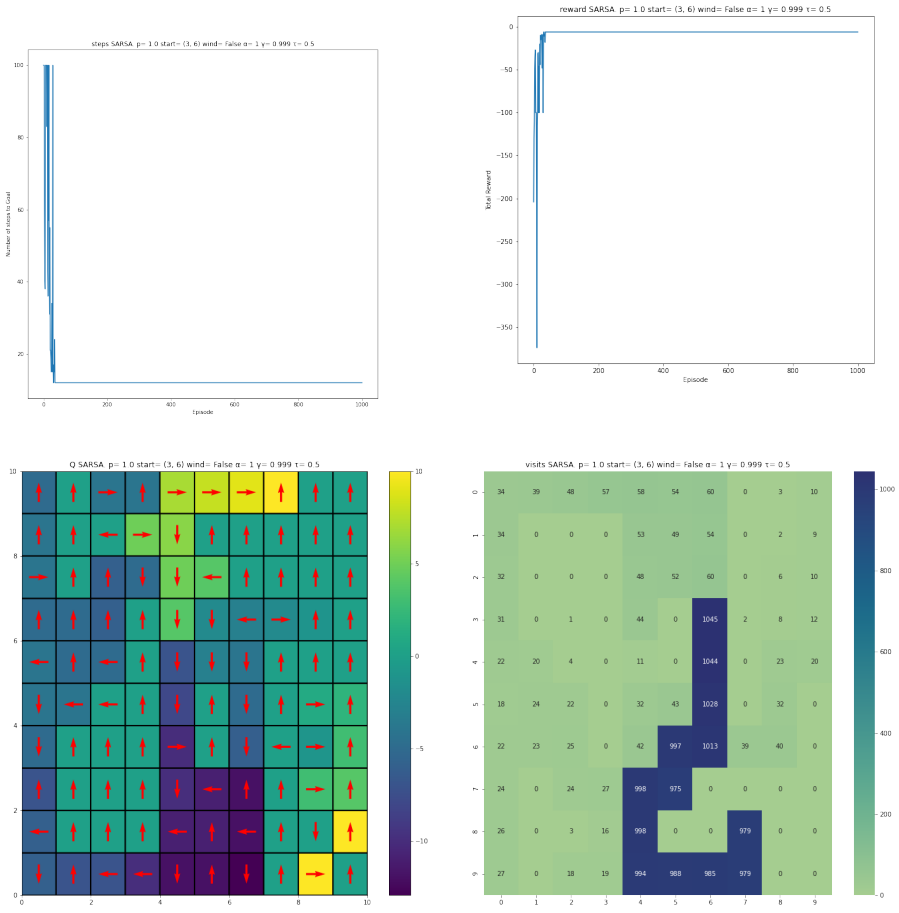

Figure 15: SARSA. p= 1.0 start= (3, 6) wind= False α= 1 γ= 0.999 τ= 0.5

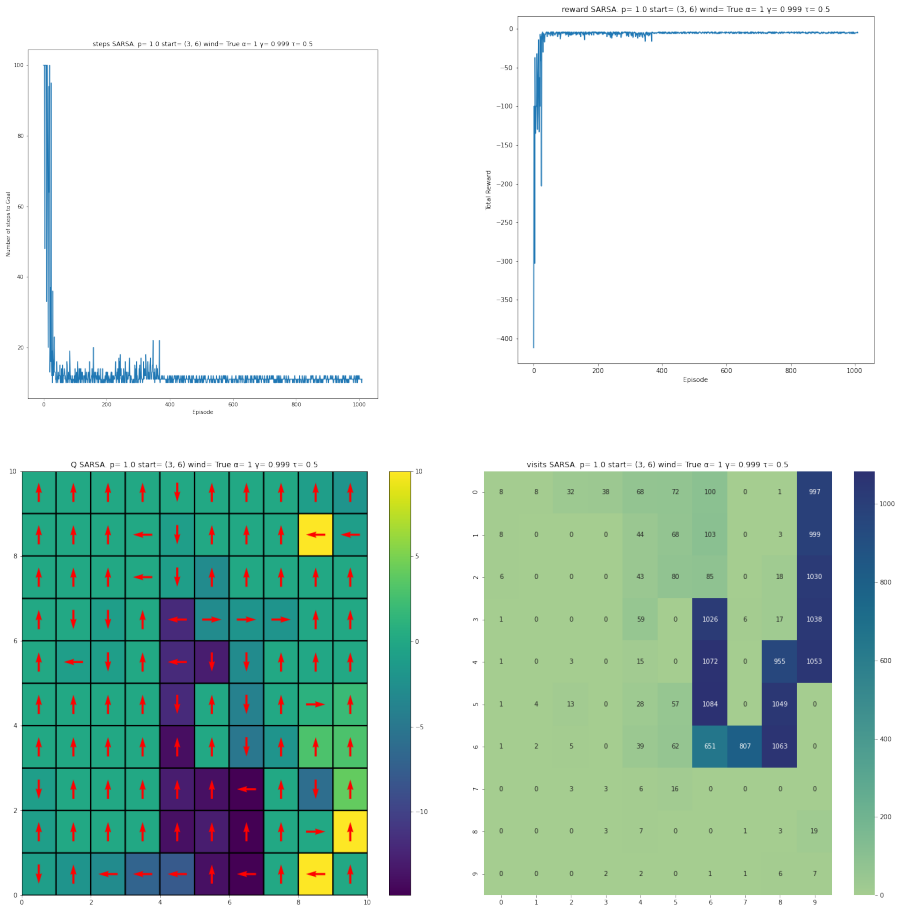

Figure 16: SARSA. p= 1.0 start= (3, 6) wind= True α= 1 γ= 0.999 τ= 0.5

4 QLearning

4.1 ϵ-Greedy Policy

All experiments were run with $\epsilon = 0.01$.

Wind	p_trans	start-state	learning rate	Episodes
True	1.0	[0, 4]	0.25(1.0)	1103
True	1.0	[3, 6]	0.25(1.0)	1007
True	0.7	[0, 4]	0.25	1964
True	0.7	[3, 6]	0.25	7793
False	1.0	[0, 4]	0.25(1.0)	1061
False	1.0	[3, 6]	0.25	1052
False	0.7	[0, 4]	0.25	1183
False	0.7	[3, 6]	0.25	1105

Table 5: Q-Learning with ϵ-greedy policy

Here too, similar to SARSA algorithm same trend is observed in learning rates. Although, one special point to note here is that when conditions as mentioned in row number 4 are trained, QLearning requires a rather high number of episodes to converge. This can attributed to doubly induced randomness due to wind as well as transition probability. Since Qlearning tends to be more greedy as compared to SARSA it which takes a rather conservative approach, this might had led to a rather diverging choice of actions.

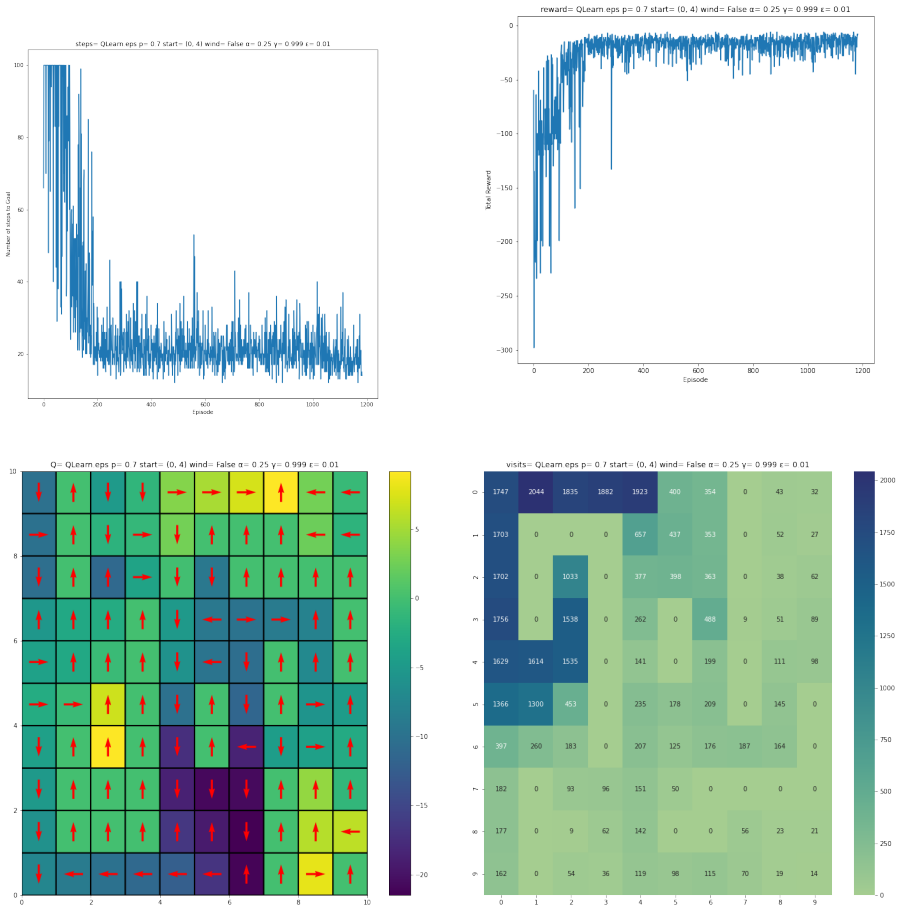

Figure 17: Qlearning with p= 0.7 start= (0,4) wind= False α= 0.25 γ= 0.999 ϵ= 0.01

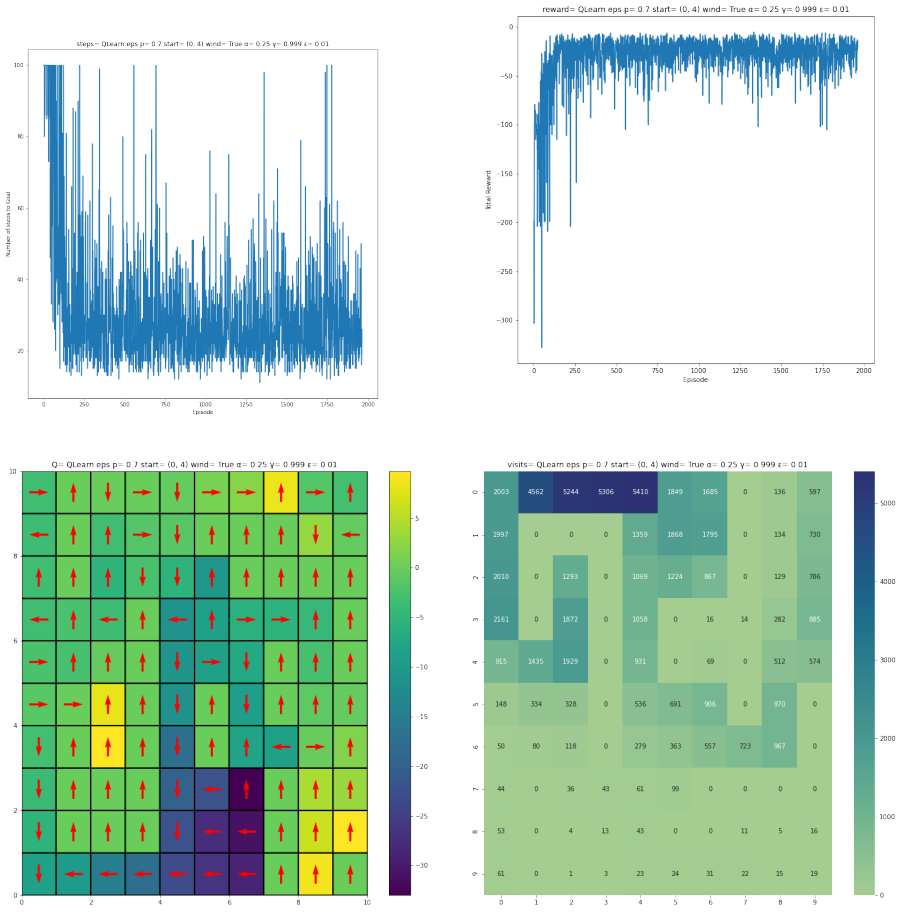

Figure 18: Qlearning with p= 0.7 start= (0,4) wind= True α= 0.25 γ= 0.999 ϵ= 0.01

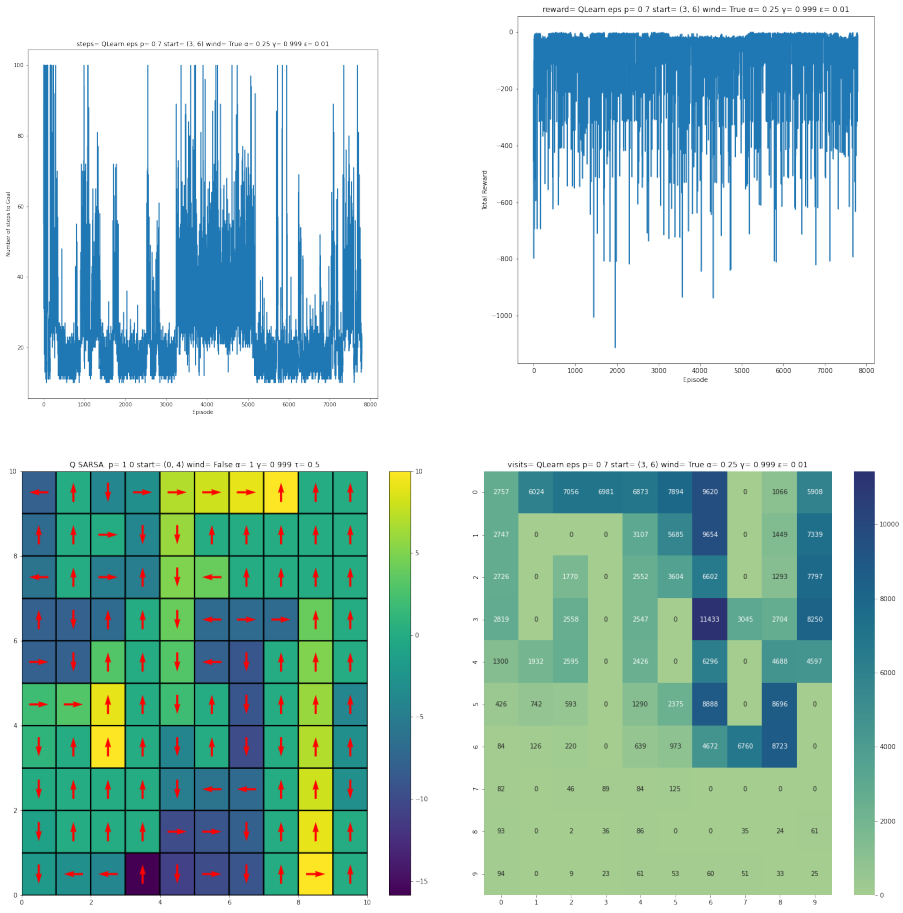

Figure 19: Qlearning with p= 0.7 start= (3,6) wind= True α= 0.25 γ= 0.999 ϵ= 0.01

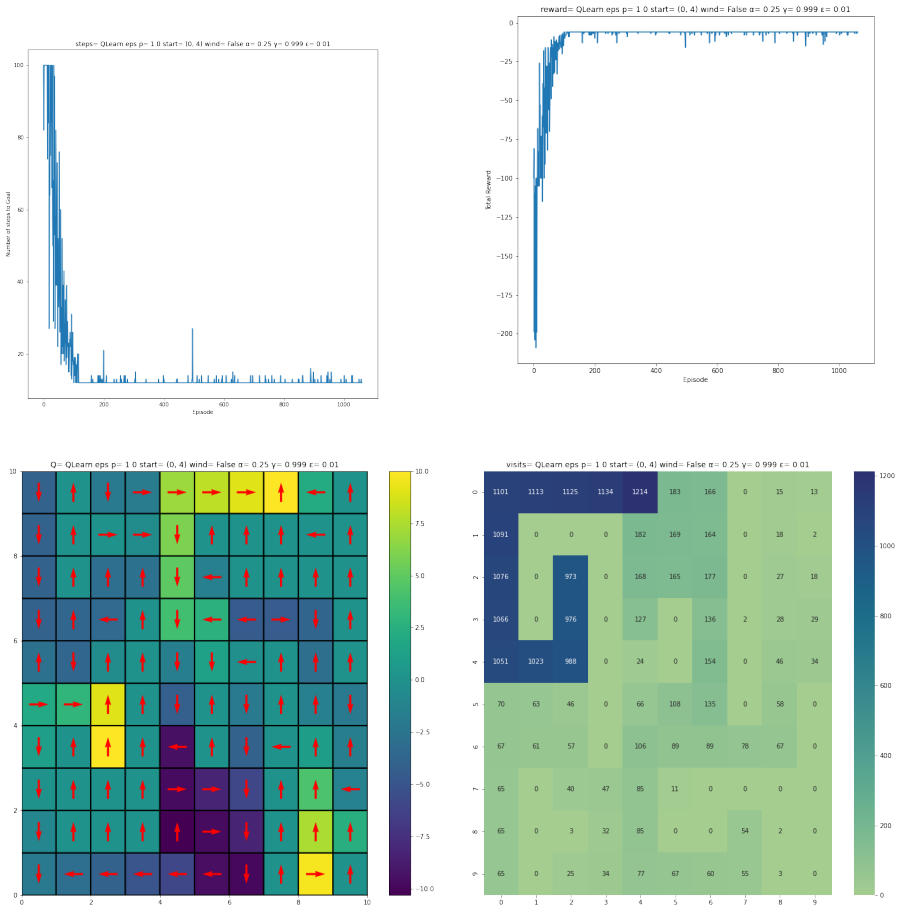

Figure 20: Qlearning with p= 1.0 start= (0,4) wind= False α= 0.25 γ= 0.999 ϵ= 0.01

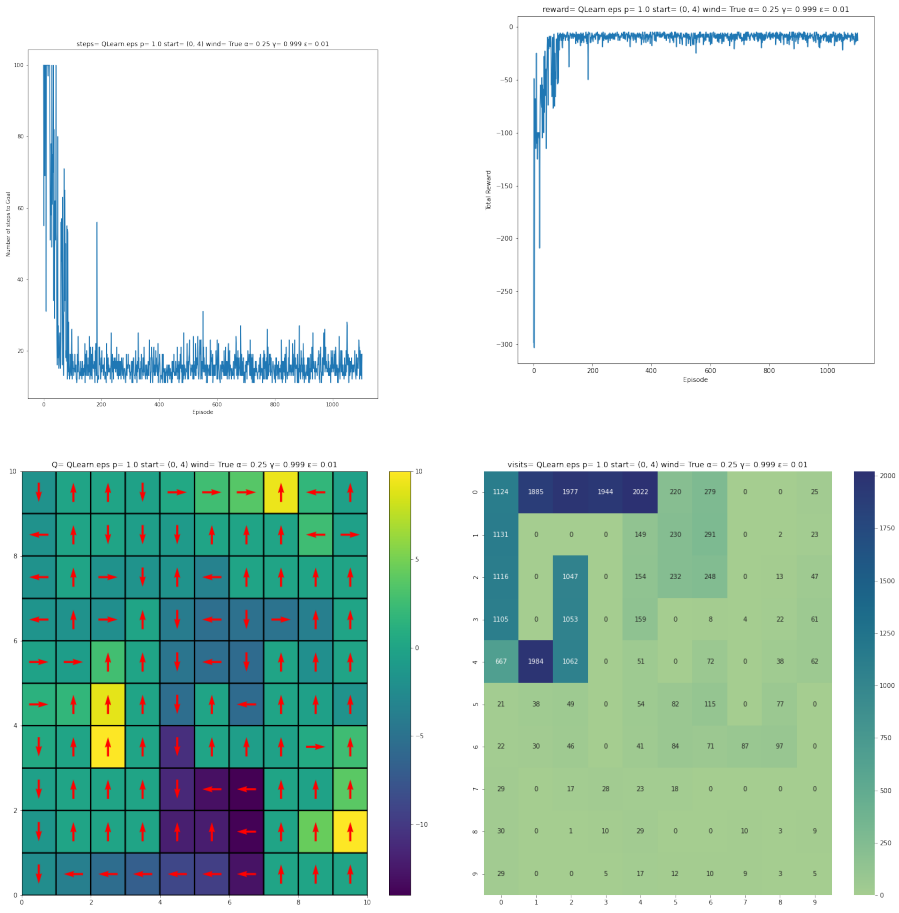

Figure 21: Qlearning with p= 1.0 start= (0,4) wind= True α= 0.25 γ= 0.999 ϵ= 0.01

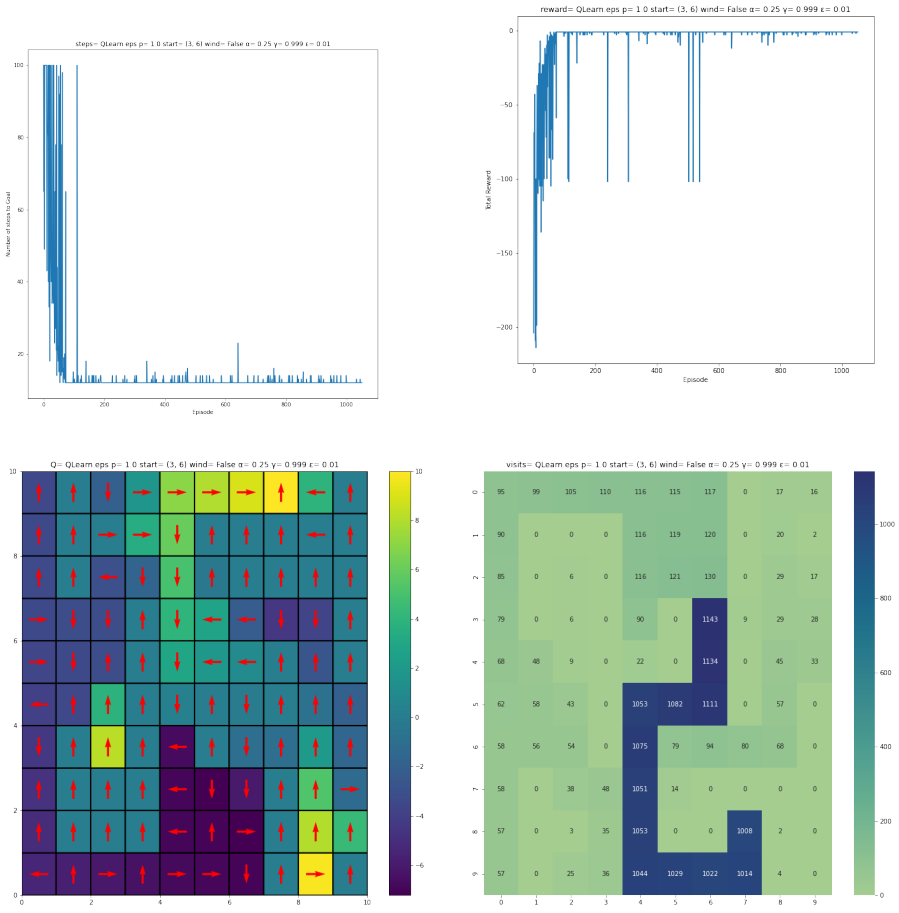

Figure 22: Qlearning with p= 1.0 start= (3,6) wind= False α= 0.25 γ= 0.999 ϵ= 0.01

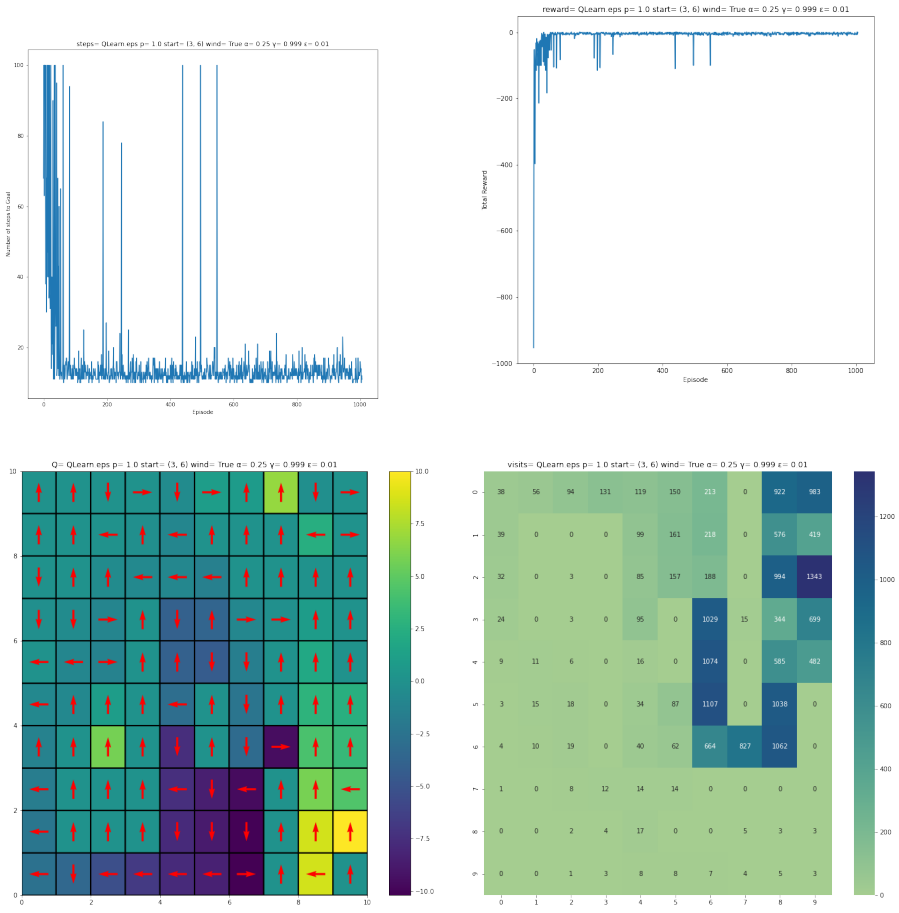

Figure 23: Qlearning with p= 1.0 start= (3,6) wind= True α= 0.25 γ= 0.999 ϵ= 0.01

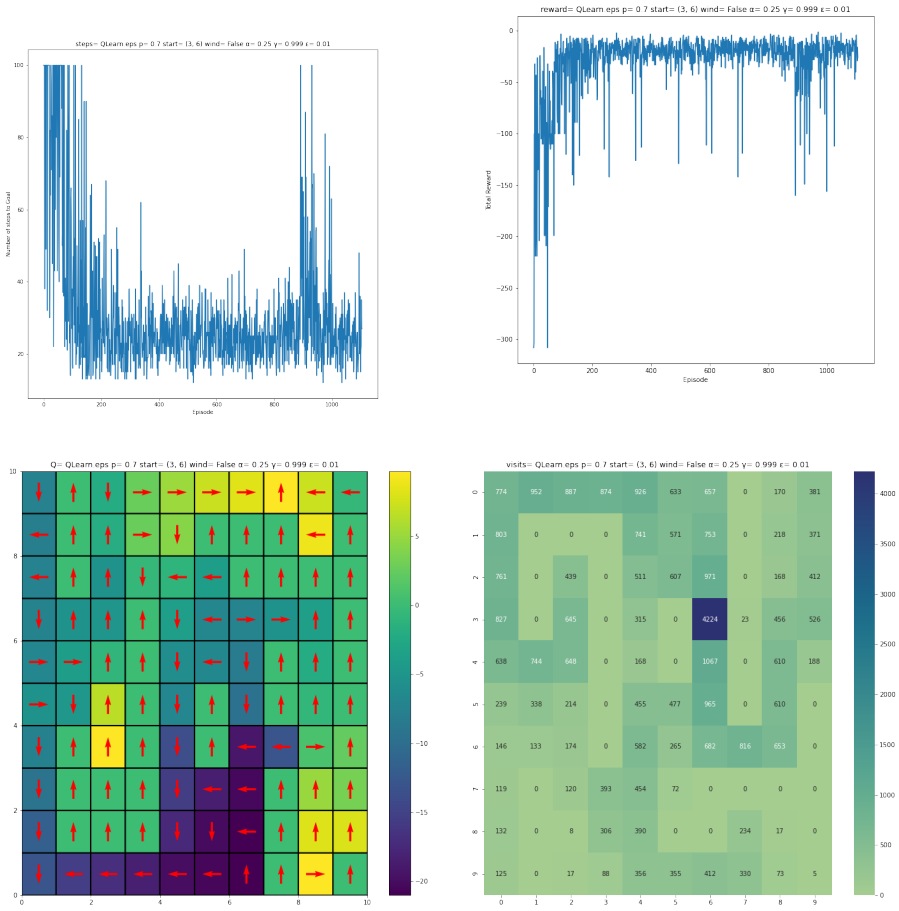

Figure 24: Qlearning with p= 0.7 start= (3,6) wind= False α= 0.25 γ= 0.999 ϵ= 0.01

4.2 Softmax Policy

As mentioned before, we take temperature=0.5. QLearning seems to be act-

Wind	p_trans	start-state	learning rate	Temperature	Episodes
True	1.0	[0, 4]	1.0	0.5	1198
True	1.0	[3, 6]	1.0	0.5	1094
True	0.7	[0, 4]	0.25	0.5	1095
True	0.7	[3, 6]	0.25	0.5	4086
False	1.0	[0, 4]	1.0	0.5	1001
False	1.0	[3, 6]	1.0	0.5	1001
False	0.7	[0, 4]	0.25	0.5	1330
False	0.7	[3, 6]	0.25	0.5	1441

Table 6: SARSA with softmax policy

ing almost like SARSA in terms of hyperparameter tuning and convergence of outputs as well. Except case in row number 4 , for the same reasons of extra stochasticity as explained in subsection 4.1.

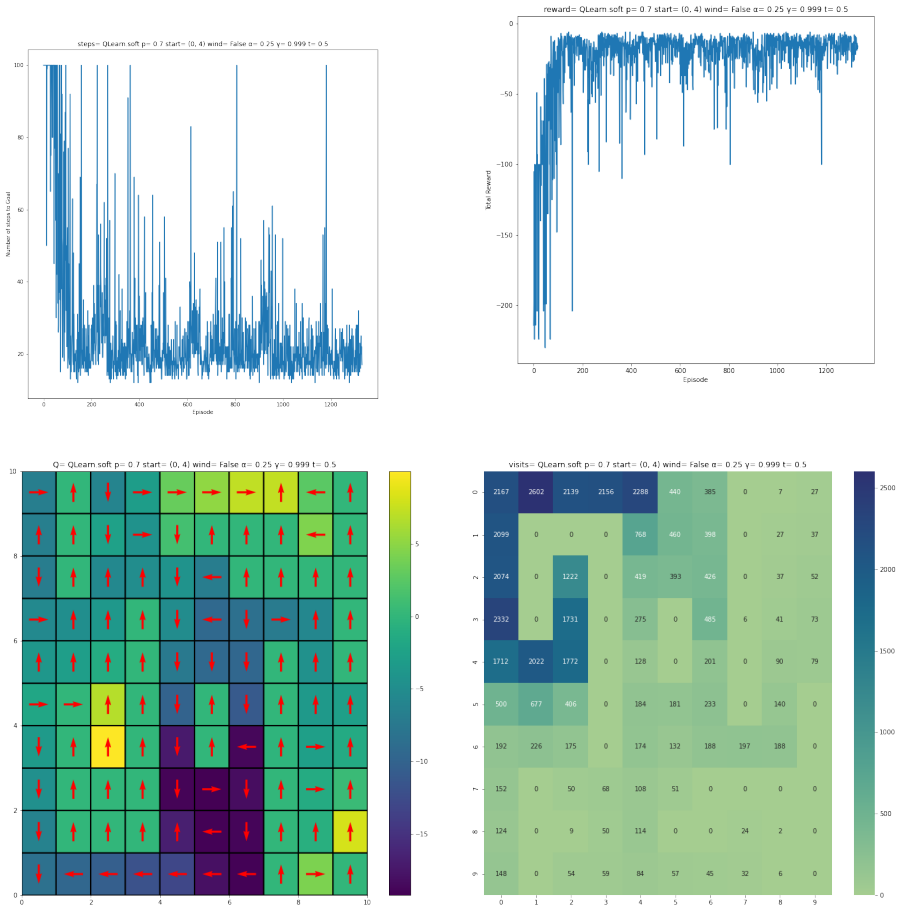

Figure 25: Qlearning with p= 0.7 start= (0,4) wind= False α= 0.25 γ= 0.999 τ= 0.5

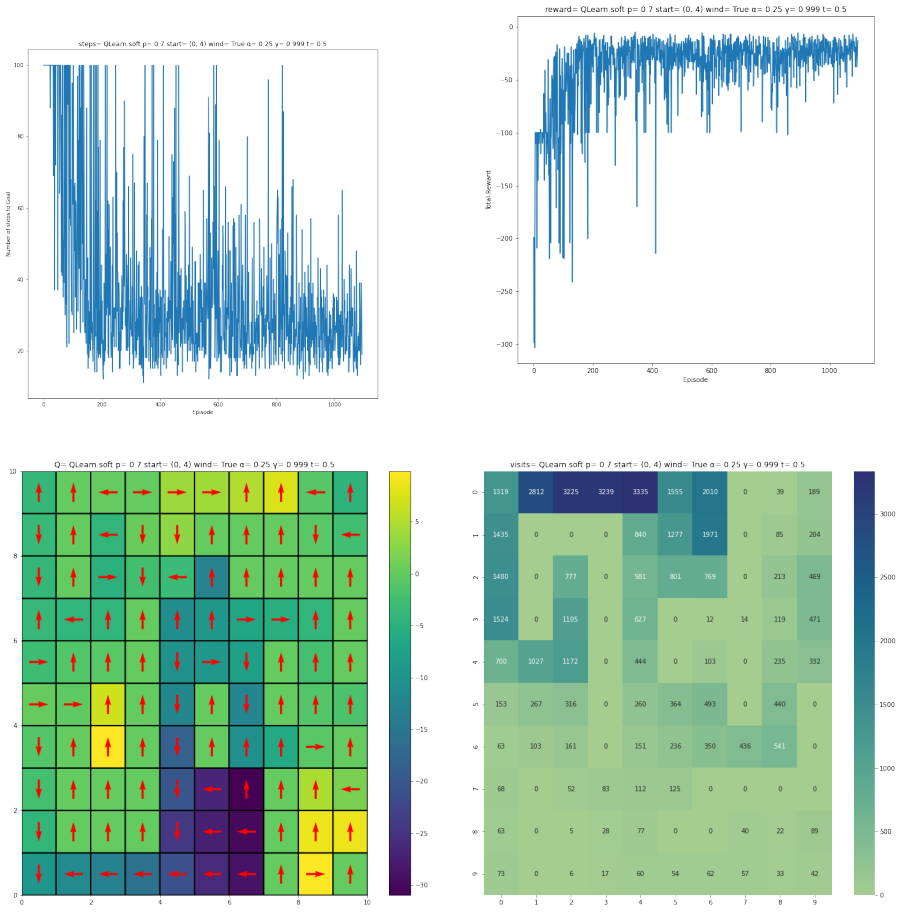

Figure 26: Qlearning with p= 0.7 start= (0,4) wind= True α= 0.25 γ= 0.999 τ= 0.5

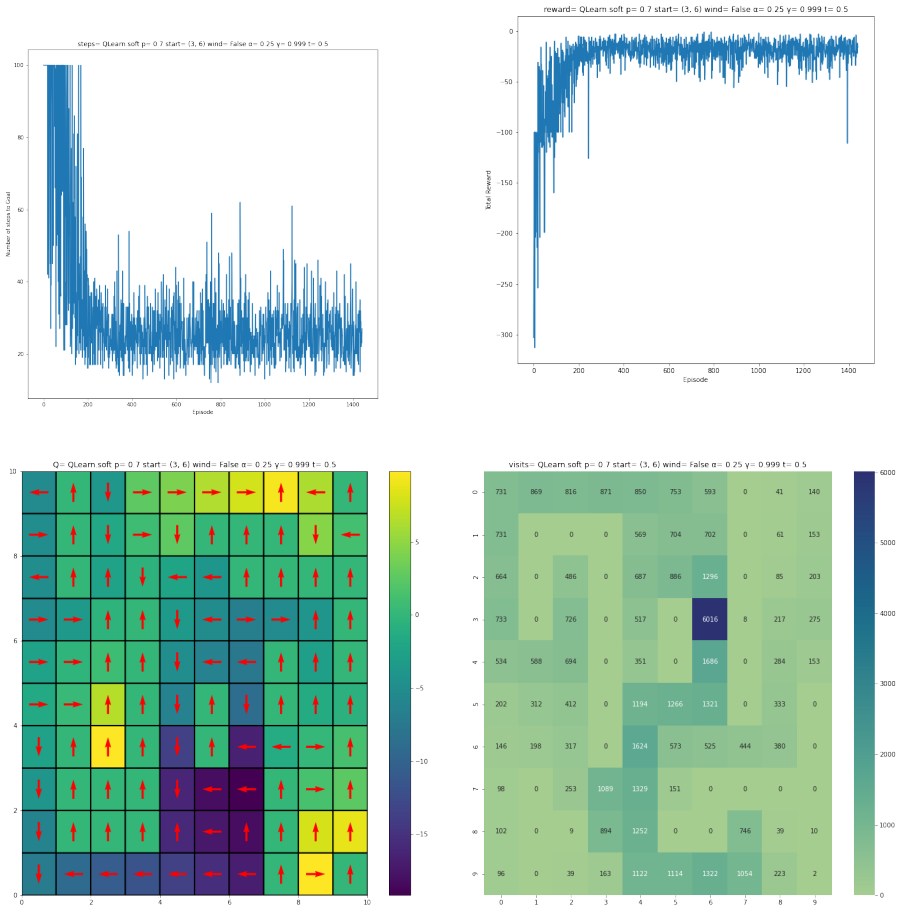

Figure 27: Qlearning with p= 0.7 start= (3,6) wind= False α= 0.25 γ= 0.999 τ= 0.5

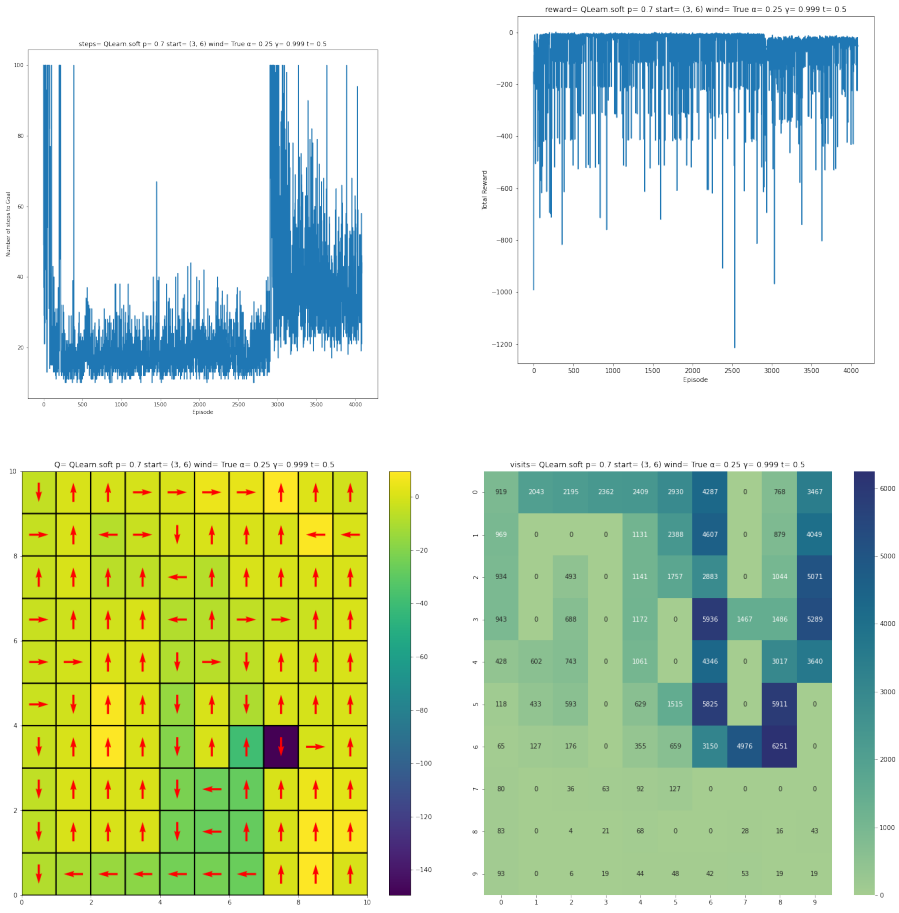

Figure 28: Qlearning with p= 0.7 start= (3,6) wind= True α= 0.25 γ= 0.999 τ= 0.5

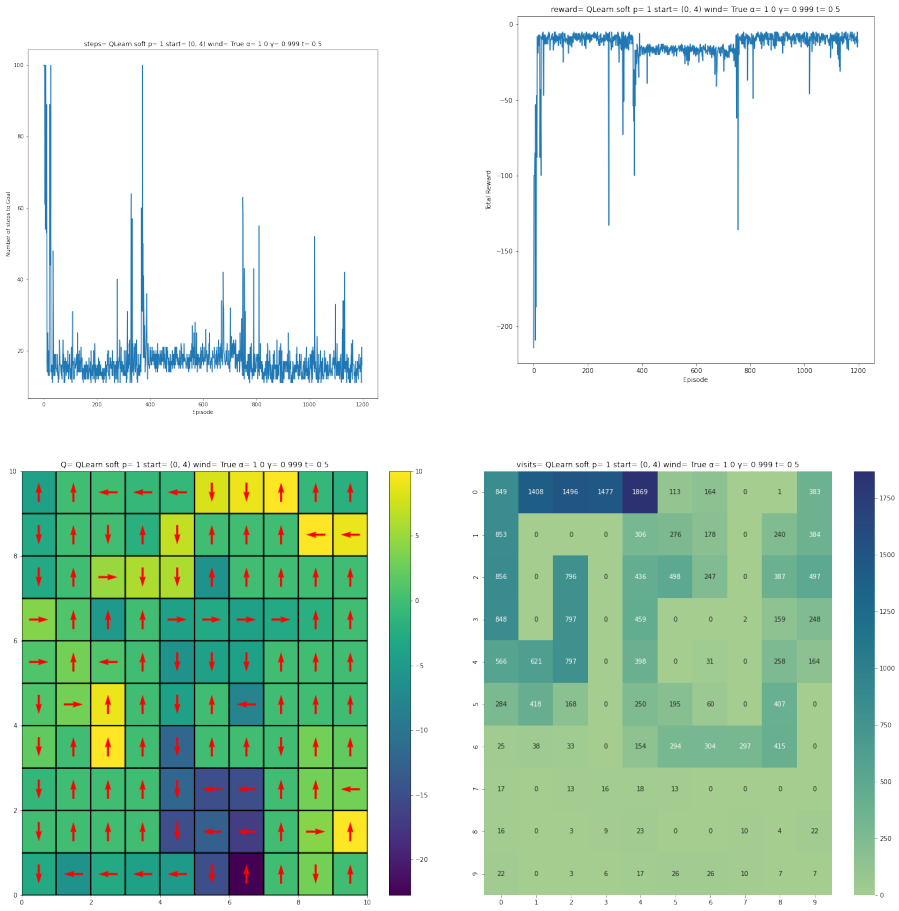

Figure 29: Qlearning with p= 1 start= (0,4) wind= True α= 1 γ= 0.999 τ= 0.5

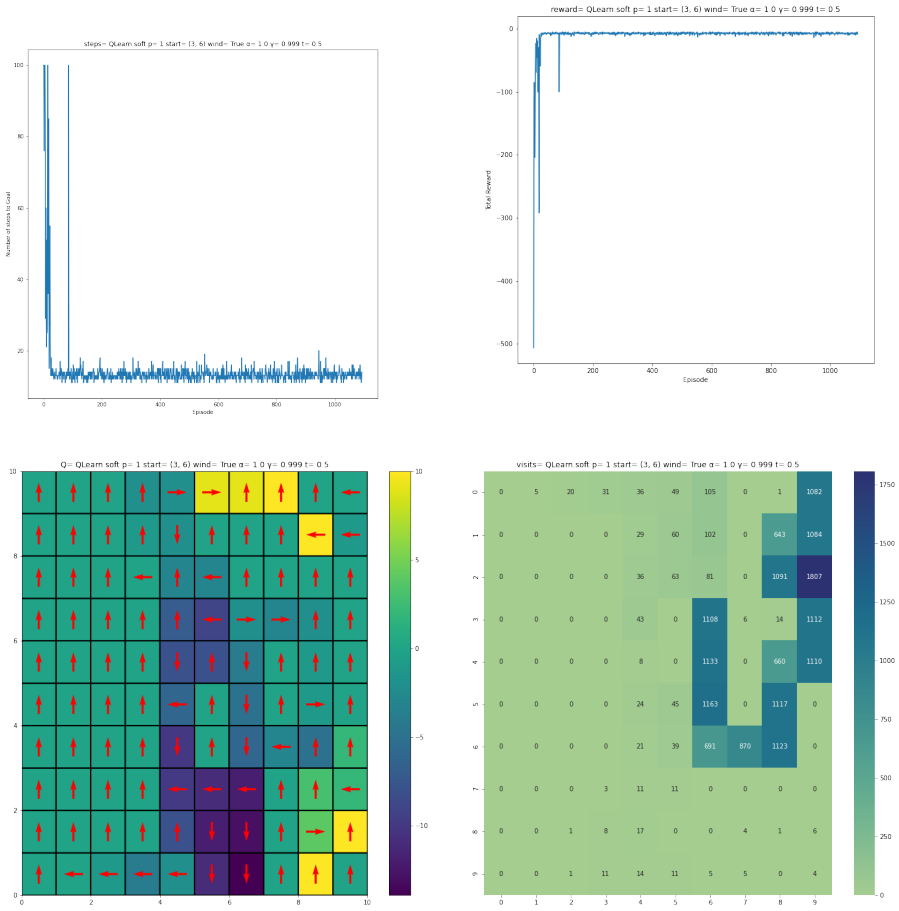

Figure 30: Qlearning with p= 1 start= (3,6) wind= True α= 1 γ= 0.999 τ= 0.5

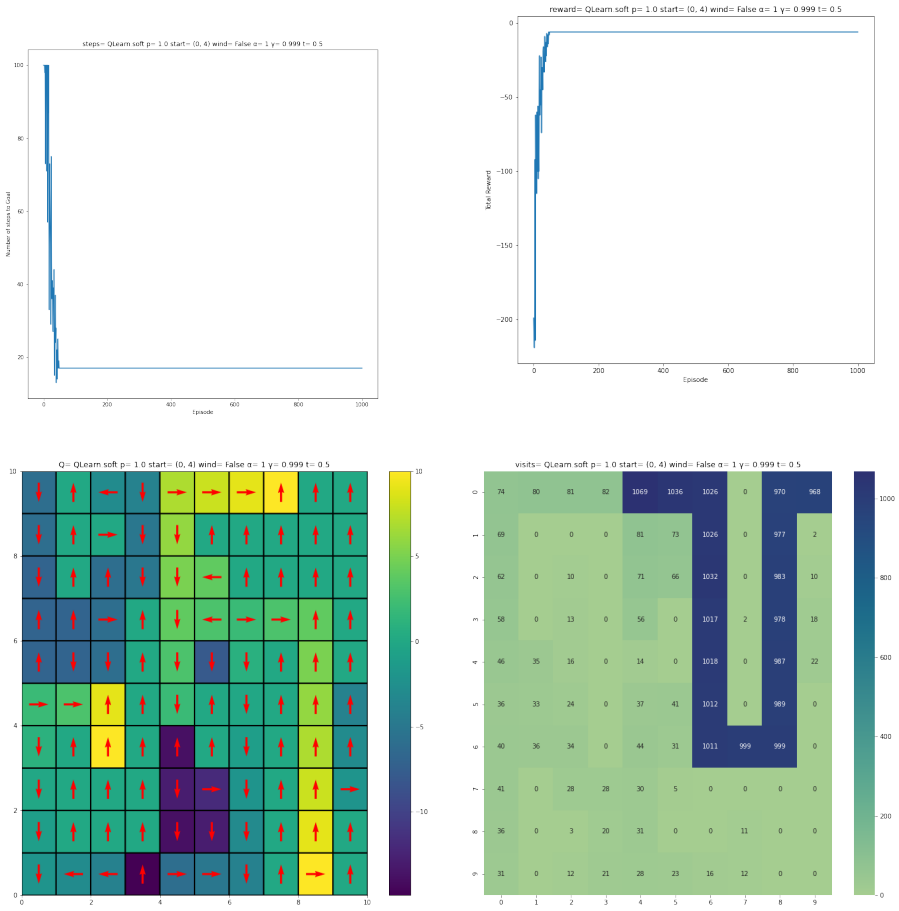

Figure 31: Qlearning with p= 1.0 start= (0,4) wind= False α= 1 γ= 0.999 τ= 0.5

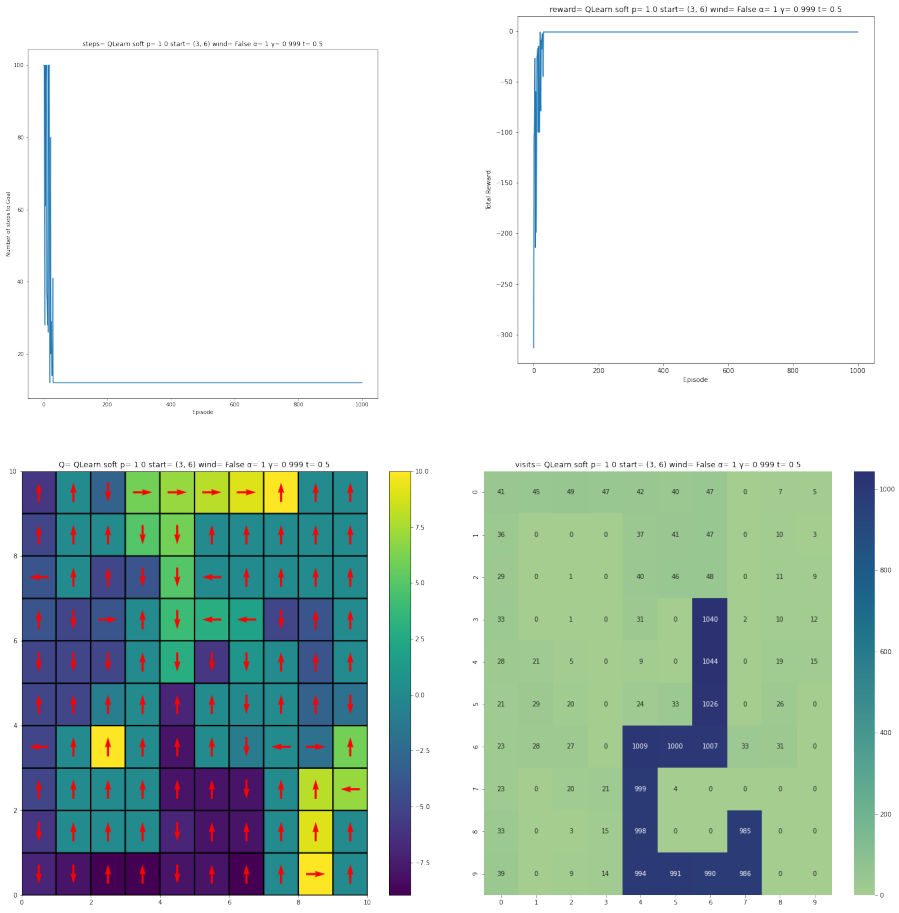

Figure 32: Qlearning with p= 1 start= (3,6) wind= False α= 1.0 γ= 0.999 τ= 0.5