# Boltzmann exploration assignment

By: Barnabas Katona

**Task 1** **3**
Description 3
Results 3
Implementation 3
Optimal parameters 3
Comparing the two algorithms 4
Slippery on 4
Slippery off 5
**Task 2** **7**
Description 7
Results 7
**Appendix** **9**
No decay of temp 9

# Task 1

## Description

This task consists of three main points:

- Implement the Boltzmann exploration strategy instead of the epsilon greedy algorithm we have been using so far
- Find the optimal parameters for this new code
- Compare the results of the code when using Boltzmann and epsilon-greedy

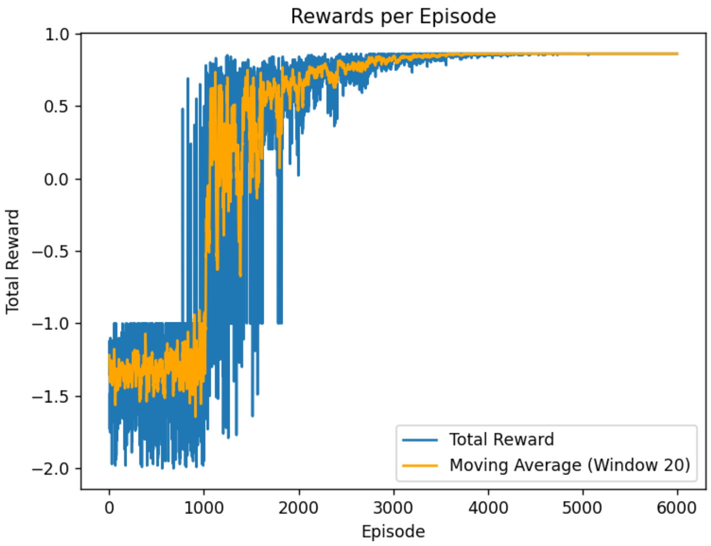
## Results

### Implementation

After researching the Boltzmann strategy I first declared a constant called temperature, this will be used to set the exploration to exploitation rate, then I changed the epsilon greedy algorithm to first setting probabilities for each action using the Boltzmann distribution and using those probabilities for the agent to choose the action. At the end of each episode I reduce the temperature variable, this is crucial in order to emphasise  exploitation once the q-table is properly set up. In case we neglect this last line of code the output will be random, see (1)

### Optimal parameters

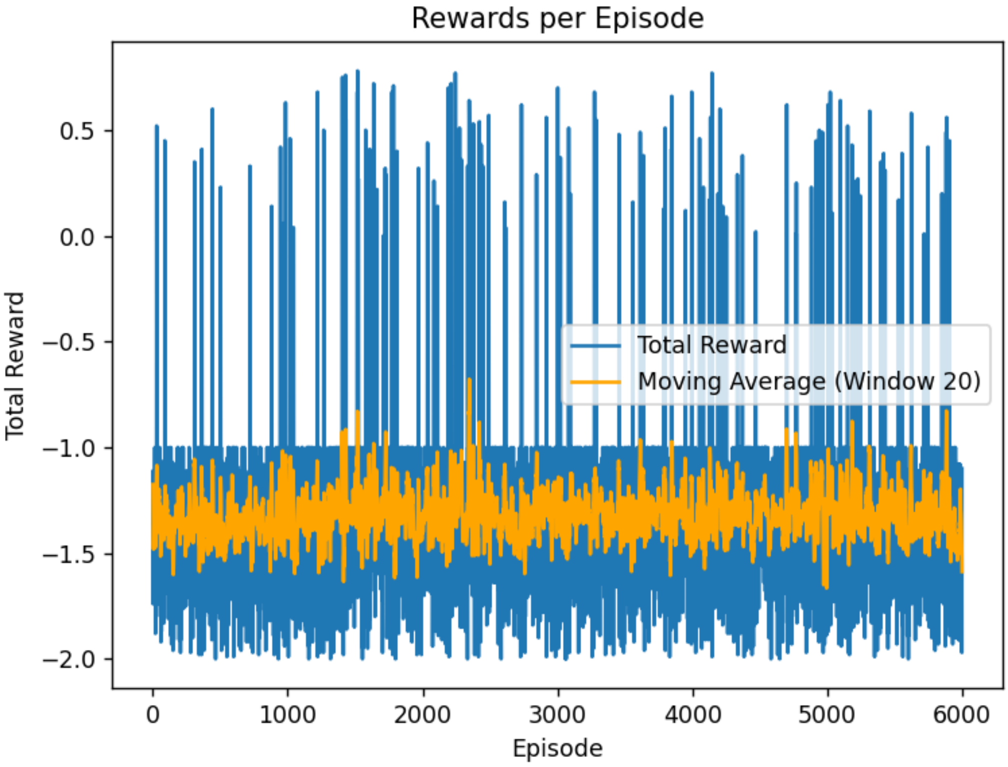
After thorough experimentation I came to the conclusion that having the temperature parameter be .9 and multiplying it by .999 with each episode will bring the most optimal results. See image:



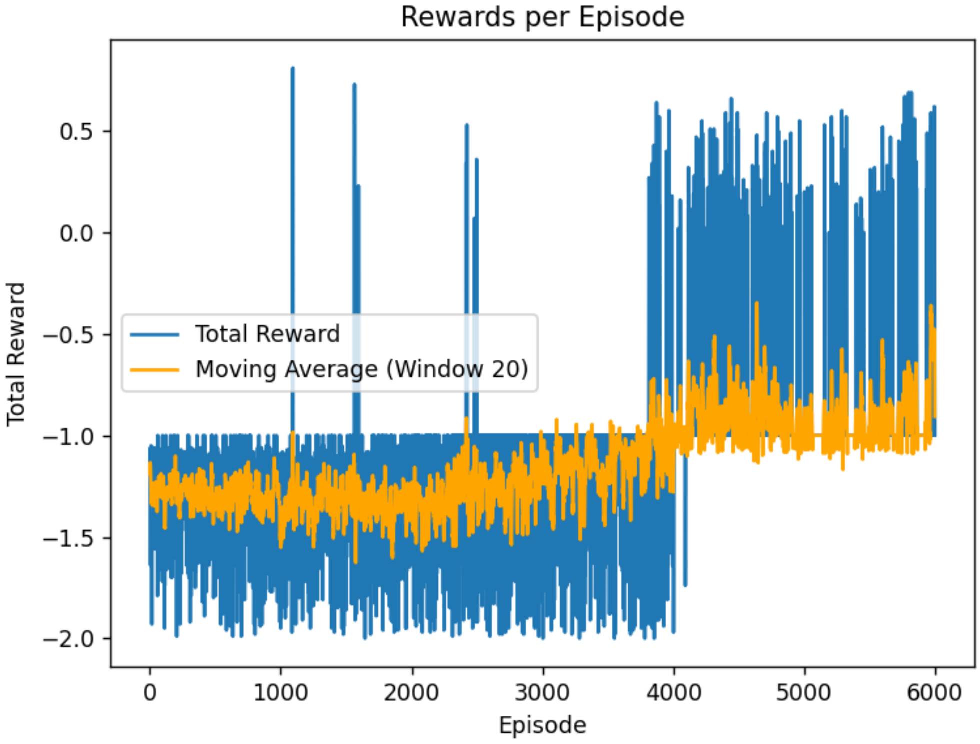
## Comparing the two algorithms

### Slippery on

Epsilon:



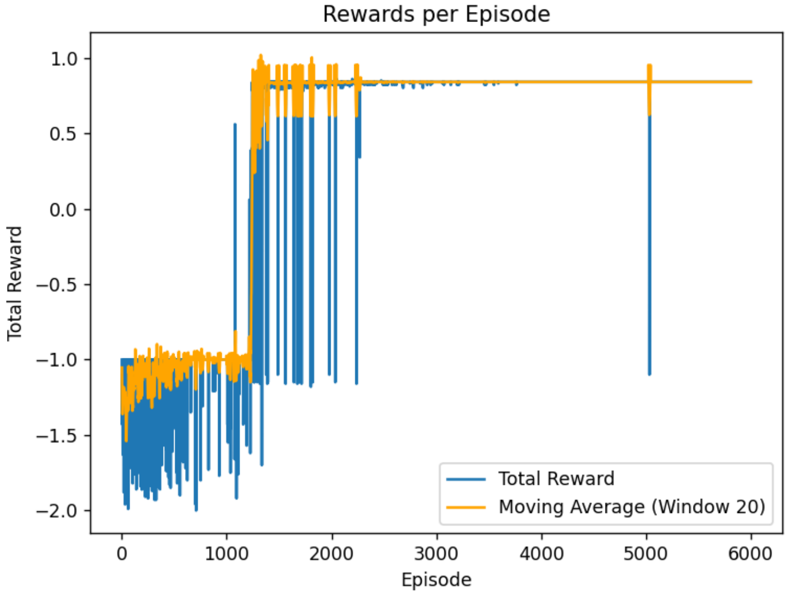
Boltzmann



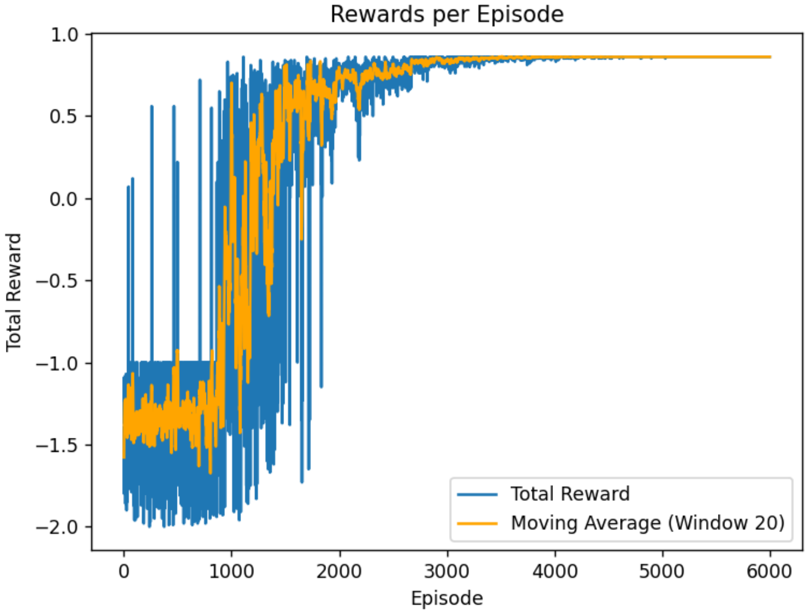
In case the slippery feature is on the Boltzmann strategy outperforms the epsilon greedy approach significantly with higher episode numbers. The high blue total rewards are misleadingly showing that the results are great when using the new approach, however the yellow moving average shows that the moving average is almost identical until episode 4000, however from that point there is (seemingly) an increase of around +0.5 on the average rewards the agent brings back.

## Slippery off

### Epsilon greedy



### Boltzmann



Conclusion:
When slippery is False the epsilon approach has a lot less noise, mainly once it had set up a proper q-table, also the transition from getting consistently negative rewards to positive rewards is from one episode to another unlike in the case of the Boltzmann approach, where the transition from constantly getting negative rewards to constantly getting positive rewards

was over the span of approximately 1000 episodes, thus I conclude that overall the epsilon greedy approach performs better, mainly when extra uncertainty (is_slippery) is not present. It is interesting to note however that when both algorithms have finished exploring the epsilon greedy approach had a few -2 returns over the span of the last 4000 episodes, whilst the Boltzmann approach had 0.

# Task 2

## Description

“

- Turn off the frequent reward and set is_slippery to False. Run a random agent (100% exploration) for a large number of episodes and store the entire episode sequences for the next task.
- Calculate the average number of episodes until the random agent hits the goal for the first time. You can estimate this by running several simulations until you get a few successful episodes (say 10).
- Calculate the confidence interval of your estimate. You can use your statistics course on confidence intervals.

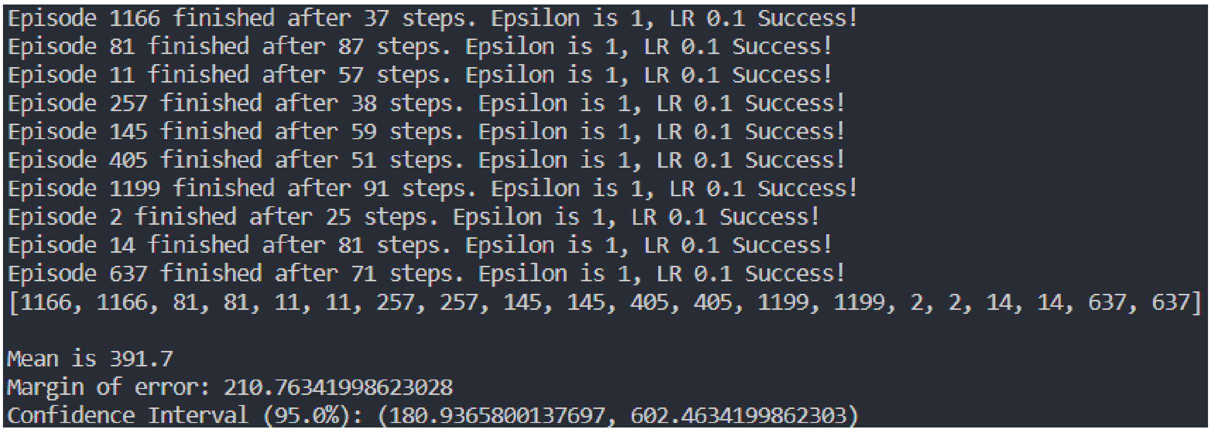
“

## Results

I modified the provided code as described above. I ran 10 tests, each of them until the first time the agent successfully completed the maze at which point I saved the episode number to an array and started over with a new counter.

Afterwards I used Python’s numpy and scipy library to calculate the following:

- Mean
- Standard deviation
- Standard error
- Degrees of freedom
- T values based on 95% confidence
- Margin of error
- Lower and upper cf level



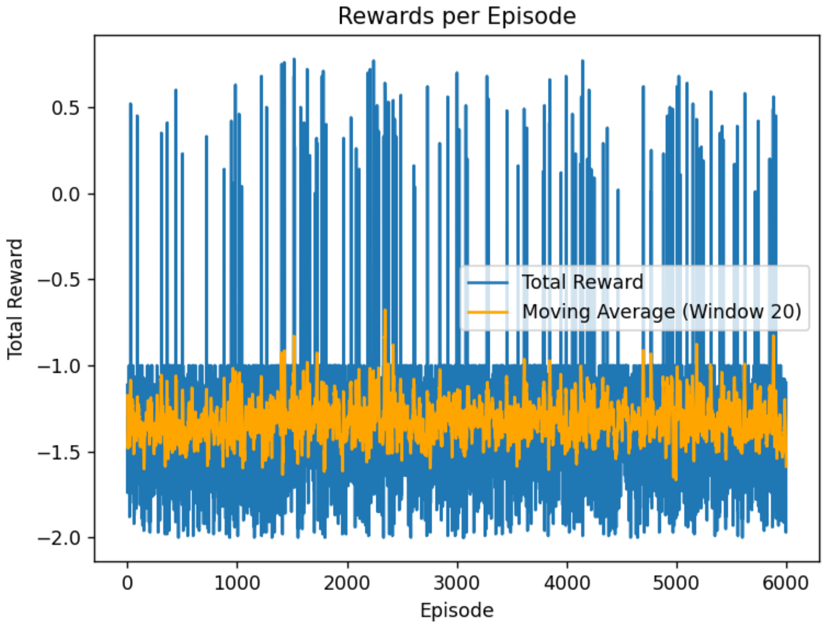
```
Episode 1166 finished after 37 steps. Epsilon is 1, LR 0.1 Success!
Episode 81 finished after 87 steps. Epsilon is 1, LR 0.1 Success!
Episode 11 finished after 57 steps. Epsilon is 1, LR 0.1 Success!
Episode 257 finished after 38 steps. Epsilon is 1, LR 0.1 Success!
Episode 145 finished after 59 steps. Epsilon is 1, LR 0.1 Success!
Episode 405 finished after 51 steps. Epsilon is 1, LR 0.1 Success!
Episode 1199 finished after 91 steps. Epsilon is 1, LR 0.1 Success!
Episode 2 finished after 25 steps. Epsilon is 1, LR 0.1 Success!
Episode 14 finished after 81 steps. Epsilon is 1, LR 0.1 Success!
Episode 637 finished after 71 steps. Epsilon is 1, LR 0.1 Success!
[1166, 1166, 81, 81, 11, 11, 257, 257, 145, 145, 405, 405, 1199, 1199, 2, 2, 14, 14, 637, 637]

Mean is 391.7
Margin of error: 210.76341998623028
Confidence Interval (95.0%): (180.9365800137697, 602.4634199862303)
```

As shown in the picture above the

- mean is 391.7
- Margin of error: 210.76
- Confidence intervals: (180.93, 602.46)

# Appendix

## No decay of temp


Rewards per Episode
Total Reward
Episode
Total Reward
Moving Average (Window 20)