Q-Learning and Blackjack

Benjamin Allévius[1], Sebastian Rosengren[1], and Erik Thorsén[1]

[1]Department of Mathematics, Stockholm University, Sweden

Abstract

Sutton and Barto (2018)

1 Results

In this section we perform let the Q-learning algorithm act on the previously defined environments. To ensure that the learning algorithm explores the state space we will use an ϵ-greedy algorithm, if not stated otherwise. The decay of the epsilon greedy algorithm will be linear with the number of episodes has passed one tenth of the number of simulations. As an example, if we perform 10^6 simulations then after a fixed point the proability of taking a random action will decay by 1/episode. Also, the initial values of $Q(S, a)$ will be zero for all states S and actions a.

Two state space representations will be used in this section. These are $(\mathbf{X}_p, \sum_d)$ and $(\sum_p, UA, , \sum_d)$ which henceforth will be called "hand"- and "sum"- environments or state spaces. For visualisations we will aggregate the results of the Q-learning algorithm on the hand environment to a sum representation.

In Figure 1 we can see the average return by episodes, when the Q-learning algorithm is acting on different environments. Both algorithms seem to have a negative trend in the first episodes. This is to be expected since the ϵ greedy assures that we explore the state space in the first iterations. In ϵ number of cases we aim

In Figure , something happens where something is supposed to happen.

References

Sutton, R. S. and Barto, A. G. (2018). Reinforcement learning: An introduction. preprint available at `http://incompleteideas.net/book/bookdraft2018mar21.pdf`.

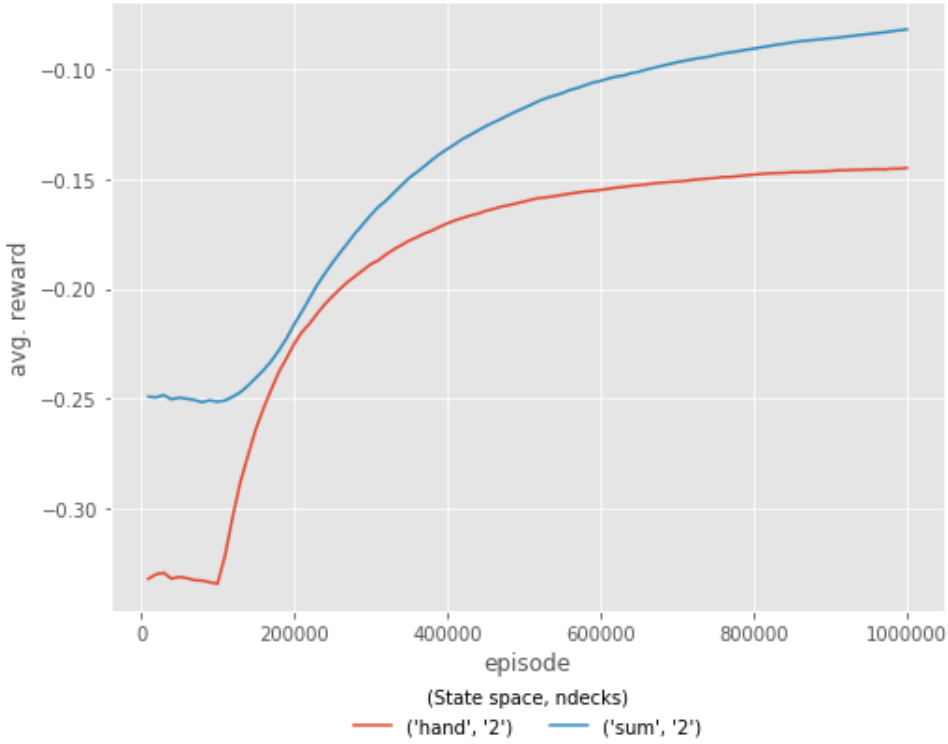

Figure 1: The average return of an ϵ-greedy Q-learning algorithm for TEST different state spaces.

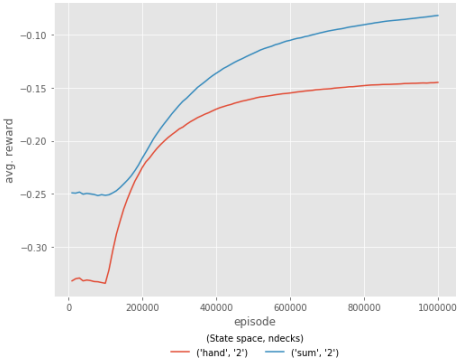

Figure 2: The average return of an ϵ-greedy Q-learning algorithm for TEST different state spaces.

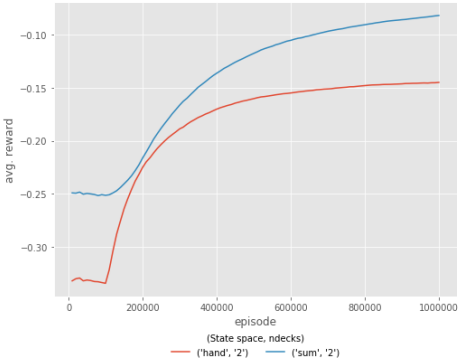

Figure 3: The average return of an ϵ-greedy Q-learning algorithm for TEST different state spaces.