Factors of Influence of the Overestimation Bias of Q-Learning

Julius Wagenbach[1] and Matthia Sabatelli[1]

Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence
University of Groningen, The Netherlands

Abstract. We study whether the learning rate α, the discount factor γ and the reward signal r have an influence on the overestimation bias of the Q-Learning algorithm. Our preliminary results in environments which are stochastic and that require the use of neural networks as function approximators, show that all three parameters influence overestimation significantly. By carefully tuning α and γ, and by using an exponential moving average of r in Q-Learning's temporal difference target, we show that the algorithm can learn value estimates that are more accurate than the ones of several other popular model-free methods that have addressed its overestimation bias in the past.

1 Approach

Recall that given a Markov Decision Process with state space $\mathcal{S}$, action space $\mathcal{A}$, transition function P and reward function $\Re(s_t, a_t, s_{t+1})$; Q-Learning learns optimal values $Q^*(s_t, a_t)$ for each state s and action a at time-step t as follows:

$$Q(s_t, a_t) := Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t)]. \tag{1}$$

Due to the maximization operator in its Temporal Difference (TD) target, Q-Learning estimates the expected maximum value of a state, instead of its maximum expected value. Most recent work aimed to reduce Q-Learning's overestimation bias by replacing its max operator [3,4,2,6,1,7]. In this paper, instead of replacing the maximization estimator, we investigate whether overestimation can be prevented by tuning the following parameters: the learning rate α, the discount factor γ and the reward signal r.

2 Main Findings

In the Gridworld environment initially proposed by Hasselt [5] we find that a static learning rate as well as a lower discount factor value significantly prevent the algorithm from overestimating. The same holds when using an exponential moving average in Q-Learning's TD-target instead of the raw reward signal r. We compute this quantity as follows: $\hat{r}(s) \mathrel{+}= \frac{1}{x}(r(t) - \hat{r}(s))$, where x is a static hyperparameter determining the degree of weighting decrease. We also

find that by tuning all three hyperparameters, Q-Learning does not suffer from underestimation either contrary to algorithms such as Double Q-Learning (DQL) [5] and Self-Correcting Q-Learning (SCQL) [7].

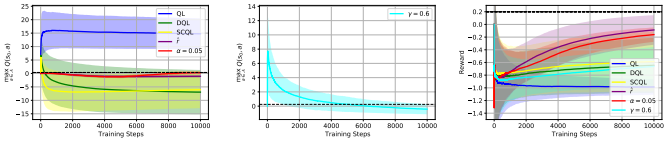

Fig. 1. From left to right our results showing that i) overestimation, as well as underestimation, can be prevented by using a constant value of $\alpha = 0.05$ and by maintaining an exponential moving average estimate $\hat{r}$; ii) $\gamma = 0.6$ allows Q-Learning to not overestimate; and iii) QL trained with either $\alpha = 0.05$, $\gamma = 0.6$ or $x = 70$ outperforms regular QL, DQL and SCQL.

Similar results have been obtained when combining Q-Learning with a multilayer perceptron serving as function approximator. On the popular `Cartpole` environment we found that simply changing the discount factor γ from 0.999 to 0.97 significantly prevents overestimation without harming Q-Learning's performance (Fig. 2).

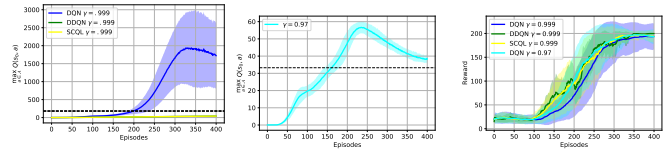

Fig. 2. From left to right our results showing that: i) DQN suffers from severe overestimation when compared to DDQN and SCQL; ii) training a DQN agent with $\gamma = 0.97$ mitigates this overestimation bias significantly; iii) a DQN agent trained with $\gamma = 0.97$ performs just as well as all other algorithms trained with $\gamma = 0.999$.

3 Summary

We have studied the role of the learning rate α, the discount factor γ and the reward signal r under the lens of the overestimation bias that characterizes the popular Q-Learning algorithm and shown that all three factors of influence play a significant role in Q-Learning's value estimations. By considering α, γ and $\hat{r}$ one can prevent overestimation in an easy, computationally not-intensive way as, differently from methods such as Double Q-Learning and Self-Correcting Q-Learning, there is no need to keep track of a second state-action table.

References

1. Abed-alguni, B.H., Ottom, M.A.: Double delayed q-learning. International Journal of Artificial Intelligence **16**(2), 41–59 (2018)
2. Karimpanal, T.G., Le, H., Abdolshah, M., Rana, S., Gupta, S., Tran, T., Venkatesh, S.: Balanced q-learning: Combining the influence of optimistic and pessimistic targets. arXiv preprint arXiv:2111.02787 (2021)
3. Lan, Q., Pan, Y., Fyshe, A., White, M.: Maxmin q-learning: Controlling the estimation bias of q-learning. arXiv preprint arXiv:2002.06487 (2020)
4. Pentaliotis, A., Wiering, M.A.: Variation-resistant q-learning: Controlling and utilizing estimation bias in reinforcement learning for better performance. (2021)
5. Van Hasselt, H.: Double q-learning. In: Advances in neural information processing systems. pp. 2613–2621 (2010)
6. Zhang, Z., Pan, Z., Kochenderfer, M.J.: Weighted double q-learning. In: IJCAI. pp. 3455–3461 (2017)
7. Zhu, R., Rigotti, M.: Self-correcting q-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11185–11192 (2021)