Revisiting the Softmax Bellman Operator: New Benefits and New Perspective

Zhao Song [1] [*] Ronald E. Parr [1] Lawrence Carin [1]

Abstract

The impact of softmax on the value function itself in reinforcement learning (RL) is often viewed as problematic because it leads to sub-optimal value (or Q) functions and interferes with the contraction properties of the Bellman operator. Surprisingly, despite these concerns, and *independent of its effect on exploration*, the softmax Bellman operator when combined with Deep Q-learning, leads to Q-functions with superior policies in practice, even outperforming its double Q-learning counterpart. To better understand how and why this occurs, we revisit theoretical properties of the softmax Bellman operator, and prove that (i) it converges to the standard Bellman operator exponentially fast in the inverse temperature parameter, and (ii) the distance of its Q function from the optimal one can be bounded. These alone do not explain its superior performance, so we also show that the softmax operator can reduce the overestimation error, which may give some insight into why a sub-optimal operator leads to better performance in the presence of value function approximation. A comparison among different Bellman operators is then presented, showing the trade-offs when selecting them.

1. Introduction

The Bellman equation (Bellman, 1957) has been a fundamental tool in reinforcement learning (RL), as it provides a sufficient condition for the optimal policy in dynamic programming. The use of the max function in the Bellman equation further suggests that the optimal policy should be greedy w.r.t. the Q-values. On the other hand, the trade-off between exploration and exploitation (Thrun, 1992) motivates the use of exploratory and potentially sub-optimal actions during learning, and one commonly-used strategy is to add randomness by replacing the max function with the softmax function, as in Boltzmann exploration (Sutton & Barto, 1998). Furthermore, the softmax function is a differentiable approximation to the max function, and hence can facilitate analysis (Reverdy & Leonard, 2016).

The beneficial properties of the softmax Bellman operator are in contrast to its potentially negative effect on the accuracy of the resulting value or Q-functions. For example, it has been demonstrated that the softmax Bellman operator is not a contraction, for certain temperature parameters (Littman, 1996, Page 205). Given this, one might expect that the convenient properties of the softmax Bellman operator would come at the expense of the accuracy of the resulting value or Q-functions, or the quality of the resulting policies. In this paper, we demonstrate that, in the case of deep Q-learning, this expectation is surprisingly incorrect. We combine the softmax Bellman operator with the deep Q-network (DQN) (Mnih et al., 2015) and double DQN (DDQN) (van Hasselt et al., 2016a) algorithms, by replacing the max function therein with the softmax function, in the target network. We then test the variants on several games in the Arcade Learning Environment (ALE) (Bellemare et al., 2013), a standard large-scale deep RL testbed. The results show that the variants using the softmax Bellman operator can achieve higher test scores, and reduce the Q-value overestimation as well as the gradient noise on most of them. *This effect is independent of exploration and is entirely attributable to the change in the Bellman operator.*

This surprising result suggests that a deeper understanding of the softmax Bellman operator is warranted. To this end, we prove that starting from the same initial Q-values, we can upper and lower bound how far the Q-functions computed with the softmax operator can deviate from those computed with the regular Bellman operator. We further show that the softmax Bellman operator converges to the optimal Bellman operator in an exponential rate w.r.t. the inverse temperature parameter. This gives insight into why the negative convergence results may not be as discouraging as they initially seem, but it does not explain the superior performance observed in practice. Motivated by recent work (van Hasselt et al., 2016a; Anschel et al., 2017) targeting bias and insta-

[1]Duke University [*]Work performed as a graduate student at Duke University; now at Baidu Research. Correspondence to: Zhao Song <zsong@alumni.duke.edu>, Ronald E. Parr <parr@cs.duke.edu>, Lawrence Carin <lcarin@duke.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

bility of the original DQN (Mnih et al., 2015), we further investigate whether the softmax Bellman operator can alleviate these issues. As discussed in van Hasselt et al. (2016a), one possible explanation for the poor performance of the vanilla DQN on some Atari games was the overestimation bias when computing the target network, due to the max operator therein. We prove that given the same assumptions as van Hasselt et al. (2016a), the softmax Bellman operator can reduce the overestimation bias, for any inverse temperature parameters. We also quantify the overestimation reduction by providing its lower and upper bounds.

Our results are complementary to and add new motivations for existing work that explores various ways of softening the Bellman operator. For example, entropy regularizers have been used to smooth policies. The motivations for such approaches include computational convenience, exploration, or robustness (Fox et al., 2016; Haarnoja et al., 2017; Schulman et al., 2017; Neu et al., 2017). With respect to value functions, Asadi & Littman (2017) proposed an alternative mellowmax operator and proved that it is a contraction. Their experimental results suggested that it can improve exploration, but the possibility that the sub-optimal Bellman operator could, independent of exploration, lead to superior policies was not considered. Our results, therefore, provide additional motivation for further study of operators such as mellowmax. Although very similar to softmax, the mellowmax operator needs some extra computation to represent a policy, as noted in Asadi & Littman (2017). Our paper discusses this and other trade-offs among different Bellman operators.

The rest of this paper is organized as follows: We provide the necessary background and notation in Section 2. The softmax Bellman operator is introduced in Section 3, where its convergence properties and performance bound are provided. Despite being sub-optimal on Q-functions, the softmax operator is shown in Section 4 to consistently outperform its max counterpart on several Atari games. Such surprising result further motivates us to investigate why this happens in Section 5. A thorough comparison among different Bellman operators is presented in Section 6. Section 7 discusses the related work and Section 8 concludes this paper.

2. Background and Notation

A Markov decision process (MDP) can be represented as a 5-tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition kernel whose element $P(s'|s,a)$ denotes the transition probability from state s to state s' under action a, R is a reward function whose element $R(s,a)$ denotes the expected reward for executing action a in state s, and $\gamma \in (0,1)$ is the discount factor. The policy π in an MDP can be represented in terms of a probability mass function (PMF), where $\pi(s,a) \in [0,1]$ denotes the probability of selecting action a in state s, and $\sum_{a \in \mathcal{A}} \pi(s,a) = 1$.

For a given policy π, its state-action value function $Q^\pi(s,a)$ is defined as the accumulated, expected, discount reward, when taking action a in state s, and following policy π afterwards, i.e., $Q^\pi(s,a) = \mathbb{E}_{a_t \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a\right]$. For the optimal policy π^*, its corresponding Q-function satisfies the following Bellman equation:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q^*(s',a').$$

In DQN (Mnih et al., 2015), the Q-function is parameterized with a neural network as $Q_{\boldsymbol{\theta}}(s,a)$, which takes the state s as input and outputs the corresponding Q-value in the final fully-connected linear layer, for every action a. The training objective for the DQN can be represented as

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \left\| Q_{\boldsymbol{\theta}}(s,a) - [R(s,a) + \gamma \max_{a'} Q_{\boldsymbol{\theta}^-}(s',a')] \right\|^2, \tag{1}$$

where $\boldsymbol{\theta}^-$ corresponds to the frozen weights in the target network, and is updated at fixed intervals. The optimization of Eq. (1) is performed via RMSProp (Tieleman & Hinton, 2012), with mini-batches sampled from a replay buffer.

To reduce the overestimation bias, the double DQN (DDQN) algorithm modified the target that $Q_{\boldsymbol{\theta}}(s,a)$ aims to fit in Eq. (1) as

$$R(s,a) + \gamma\, Q_{\boldsymbol{\theta}^-}\big(s', \arg\max_a Q_{\boldsymbol{\theta}_t}(s',a)\big).$$

Note that a separate network based on the latest estimate $\boldsymbol{\theta}_t$ is employed for action selection, and the evaluation of this policy is due to the frozen network.

Notation The softmax function is defined as

$$f_\tau(\mathbf{x}) = \frac{[\exp(\tau x_1), \exp(\tau x_2), \ldots, \exp(\tau x_m)]^T}{\sum_{i=1}^m \exp(\tau x_i)},$$

where the superscript T denotes the vector transpose. Subsequently, the softmax-weighted function is represented as $g_{\mathbf{x}}(\tau) = f_\tau^T(\mathbf{x})\,\mathbf{x}$, as a function of τ. Also, we define the vector $Q(s,) = [Q(s,a_1), Q(s,a_2), \ldots, Q(s,a_m)]^T$. We further set m to be the size of the action set $\mathcal{A}$ in the MDP. Finally, R_{min} and R_{max} denote the minimum and the maximum immediate rewards, respectively.

3. The Softmax Bellman Operator

We start by providing the following standard Bellman operator:

$$\mathcal{T} Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a'). \tag{2}$$

We propose to use the softmax Bellman operator, defined as

$$\mathcal{T}_{\text{soft}}\, Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \times \underbrace{\sum_{a'} \frac{\exp[\tau\, Q(s',a')]}{\sum_{\bar{a}} \exp[\tau\, Q(s',\bar{a})]}\, Q(s',a')}_{sm_\tau(Q(s',))}, \quad (3)$$

where $\tau \geq 0$ denotes the inverse temperature parameter. Note that $\mathcal{T}_{\text{soft}}$ will reduce to $\mathcal{T}$ when $\tau \to \infty$.

The mellowmax operator was introduced in Asadi & Littman (2017) as an alternative to the softmax operator in Eq. (3), defined as

$$mm_\omega(Q(s,)) = \frac{\log\left(\frac{1}{m}\sum_{a'} \exp[\omega\, Q(s,a')]\right)}{\omega}, \quad (4)$$

where $\omega > 0$ is a tuning parameter. Similar to the softmax operator $sm_\tau(Q(s',))$ in Eq. (3), mm_ω converges to the mean and max operators, when ω approaches 0 and ∞, respectively. Unlike the softmax operator, however, the mellowmax operator in Eq. (4) does not directly provide a probability distribution over actions, and thus additional steps are needed to obtain a corresponding policy (Asadi & Littman, 2017).

In contrast to its typical use to improve exploration (also a purpose for the mellowmax operator in Asadi & Littman (2017)), the softmax Bellman operator in Eq. (3) is combined in this paper with the DQN and DDQN algorithms in an off-policy fashion, where the action is selected according to the same ϵ-greedy policy as in Mnih et al. (2015).

Even though the softmax Bellman operator was shown in Littman (1996) not to be a contraction, with a counterexample, we are not aware of any published work showing the performance bound by using the softmax Bellman operator in Q-iteration. Furthermore, our *surprising* results in Section 4 show that test scores in DQNs and DDQNs can be improved, by solely replacing the max operator in their target networks with the softmax operator. Such improvements are independent of the exploration strategy and hence motivate an investigation of the theoretical properties of the softmax Bellman operator.

Before presenting our main theoretical results, we first show how to bound the distance between the softmax-weighted and max operators, which is subsequently used in the proof for the performance bound and overestimation reduction.

Definition 1. $\widehat{\delta}(s) \triangleq \sup_Q \max_{i,j} |Q(s,a_i) - Q(s,a_j)|$ *denotes the largest distance between Q-functions w.r.t. actions for state s, by taking the supremum of Q-functions and the maximum of all action pairs.*

Lemma 2. *By assuming* $\widehat{\delta}(s) > 0$[1]*, we have* $\forall Q$, $\frac{\widehat{\delta}(s)}{m \exp[\tau\, \widehat{\delta}(s)]} \leq \max_a Q(s,a) - f_\tau^T(Q(s,))\, Q(s,) \leq (m-1)\max\left\{\frac{1}{\tau+2}, \frac{2Q_{max}}{1+\exp(\tau)}\right\}$ [2]*, where* $Q_{max} = \frac{R_{max}}{1-\gamma}$ *represents the maximum Q-value in Q-iteration with* $\mathcal{T}_{soft}$.[3]

Note that another upper bound for this gap was provided in O'Donoghue et al. (2017). Our proof here uses a different strategy, by considering possible values for the difference between Q-values with different actions. We further derive a lower bound for this gap.

3.1. Performance Bound for $\mathcal{T}_{\text{soft}}$

The optimal state-action value function Q^* is known to be a fixed point for the standard Bellman operator $\mathcal{T}$ (Williams & Baird III, 1993), i.e., $\mathcal{T}Q^* = Q^*$. Since $\mathcal{T}$ is a contraction with rate γ, we also know that $\lim_{k\to\infty} \mathcal{T}^k Q_0 = Q^*$, for arbitrary Q_0. Given these facts, one may wonder in the limit, how far iteratively applying $\mathcal{T}_{\text{soft}}$, in lieu of $\mathcal{T}$, over Q_0 will be away from Q^*, as a function of τ.

Theorem 3. *Let* $\mathcal{T}^k Q_0$ *and* $\mathcal{T}^k_{soft} Q_0$ *denote that the operators* $\mathcal{T}$ *and* $\mathcal{T}_{soft}$ *are iteratively applied over an initial state-action value function* Q_0 *for k times. Then,*

(I) $\forall (s,a)$, $\limsup_{k\to\infty} \mathcal{T}^k_{soft}\, Q_0(s,a) \leq Q^*(s,a)$ *and*

$$\liminf_{k\to\infty} \mathcal{T}^k_{soft}\, Q_0(s,a) \geq Q^*(s,a) - \frac{\gamma(m-1)}{(1-\gamma)} \max\left\{\frac{1}{\tau+2}, \frac{2Q_{max}}{1+\exp(\tau)}\right\}.$$

(II) $\mathcal{T}_{soft}$ *converges to* $\mathcal{T}$ *with an exponential rate, in terms of* τ*, i.e., the upper bound of* $\mathcal{T}^k Q_0 - \mathcal{T}^k_{soft} Q_0$ *decays exponentially fast, as a function of* τ*, the proof of which does not depend on the bound in part* (I).

A noteworthy point about part (I) of Theorem 3 is it does not contradict the negative convergence results for $\mathcal{T}_{\text{soft}}$ since its proof and result do not need to assume the convergence of $\mathcal{T}_{\text{soft}}$. Furthermore, this bound implies that non-convergence for $\mathcal{T}_{\text{soft}}$ is different from divergence or oscillation with an arbitrary range: Even though $\mathcal{T}_{\text{soft}}$ may not converge in certain scenarios, the Q-values could still be within a reasonable range [4]. $\mathcal{T}_{\text{soft}}$ also has other benefits over $\mathcal{T}$, as shown in Section 5.

Note that although the bound for mellowmax under the entropy-regularized MDP framework was shown in Lee et al. (2018) to have a better scalar term $\log(m)$ instead of $(m-1)$, its convergence rate is linear w.r.t. τ. In contrast, our softmax Bellman operator result has an exponential rate, as shown in Part (II), which potentially gives insight into why very large values of τ are not required experimentally

[1]Note that if $\widehat{\delta}(s) = 0$, then all actions are equivalent in state s, and softmax = max.

[2]The $\sup_Q$ is over all Q-functions that occur during Q-iteration, starting from Q_0 until the iteration terminates.

[3]The Supplemental Material formally bounds $Q_{\max}$.

[4] The lower bound can become loose when τ is extremely small, but this degenerate case has never been used in experiments.

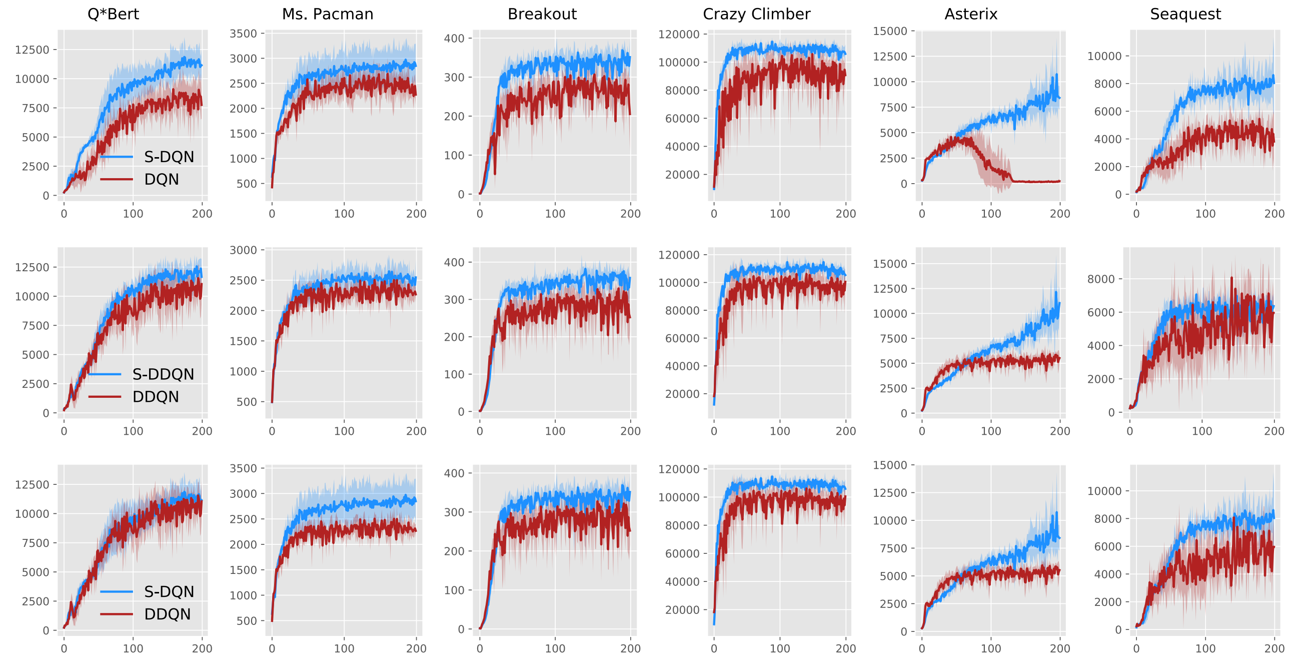

Figure 1. Performance of (top row) S-DQN vs. DQN, (middle row) S-DDQN vs. DDQN, and (bottom row) S-DQN vs. DDQN on the Atari games. X-axis and Y-axis correspond to the training epoch and test score, respectively. All curves are averaged over five independent random seeds.

in Section 4. The error bound in the sparse Bellman operator (Lee et al., 2018) improves upon $\log(m)$, but is still linear w.r.t. τ. We further empirically illustrate the faster convergence for the softmax Bellman operator in Figure 7 of Section 6.

4. Main Experiments

Our theoretical results apply to the case of a tabular value function representation and known next-state distributions, and they bound suboptimality rather than suggesting superiority of softmax. The goal of our main experiments is to show that despite being sub-optimal w.r.t. the Q-values, the softmax Bellman operator is still useful in practice by improving the DQN and DDQN algorithms. Such an improvement is entirely attributable to changing the max operator to the softmax operator in the target network.

We combine the softmax operator with DQN and DDQN, by replacing the max function therein with the softmax function, with all other steps being the same. The corresponding new algorithms are termed as S-DQN and S-DDQN, respectively. Their exploration strategies are set to be ϵ-greedy, the same as DQN and DDQN. Our code is provided at `https://github.com/zhao-song/Softmax-DQN`.

We tested on six Atari games: Q*Bert, Ms. Pacman, Crazy Climber, Breakout, Asterix, and Seaquest. Our code is built on the `Theano+Lasagne` implementation from `https://github.com/spragunr/deep_q_rl/`. The training contains 200 epochs in total. The test procedures and all the hyperparameters are set the same as DQN, with details described in Mnih et al. (2015). The inverse temperature parameter τ was selected based on a grid search over $\{1, 5, 10\}$. We also implemented the logarithmic cooling scheme (Mitra et al., 1986) in simulated annealing to gradually increase τ, but did not observe a better policy, compared with the constant temperature. The results statistics are obtained by running with five independent random seeds.

Table 1. Mean of test scores for different values of τ in S-DDQN (standard deviation in parenthesis). Each game is denoted with its initial. Note that S-DDQN reduces to DDQN when $\tau = \infty$.

τ	1	5	10	∞
Q	**12068.66** (1085.65)	11049.28 (1565.57)	11191.31 (1336.35)	10577.76 (1508.27)
M	2492.40 (183.71)	**2566.44** (227.24)	2546.18 (259.82)	2293.73 (160.50)
B	313.08 (20.13)	**350.88** (35.58)	303.71 (65.59)	284.64 (60.83)
C	107405.11 (4617.90)	**111111.07** (5047.19)	104049.46 (6686.84)	96373.08 (9244.27)
A	3476.91 (460.27)	**10266.12** (2682.00)	6588.13 (1183.10)	5523.80 (694.72)
S	272.20 (49.75)	2701.05 (10.06)	**6254.01** (697.12)	5695.35 (1862.59)

Figure 1 shows the mean and one standard deviation for the average test score on the Atari games, as a function of the training epoch: In the top two rows, S-DQN and S-DDQN generally achieve higher maximum scores and faster learning than their max counterparts, illustrating the promise of replacing max with softmax. For the game Asterix, often used as an example of the overestimation issue in DQN (van Hasselt et al., 2016a; Anschel et al., 2017), both S-DQN and S-DDQN have much higher test scores than their max counterparts, which suggests that the softmax operator can mitigate the overestimation bias. Although not as dramatic as for Asterix, the bottom row of Figure 1 shows that S-DQN generally outperforms DDQN with higher test scores for the entire test suite, which suggests the possibility of using softmax operator over double Q-learning (van Hasselt, 2010) to reduce the overestimation bias, in the presence of function approximation.

Table 1 shows the test scores of S-DDQN with different values of τ, obtained by averaging the scores from the last 10 epochs. Note that S-DDQN achieves higher test scores than its max counterpart, for most of the values of τ. Table 1 further suggests a trade-off when selecting the values for τ: Setting τ too small will move $\mathcal{T}_{\text{soft}}$ further away from the Bellman optimality, since the max operator (corresponding to $\tau = \infty$) is employed in the standard Bellman equation. On the other hand, using a larger τ will lead to the issues of overestimation and high gradient noise, as to be discussed in Section 5.

Table 2. Mean of test scores for different Bellman operators (standard deviation in parenthesis). The statistics are averaged over five independent random seeds.

	Max	Softmax	Mellowmax
Q*Bert	8331.72 (1597.67)	11307.10 (1332.80)	**11775.93** (1173.51)
Ms. Pacman	2368.79 (219.17)	**2856.82** (369.75)	2458.76 (130.34)
C. Climber	90923.40 (11059.39)	**106422.27** (4821.40)	99601.47 (19271.53)
Breakout	255.32 (64.69)	345.56 (34.19)	**355.94** (25.85)
Asterix	196.91 (135.16)	8868.00 (2167.35)	**11203.75** (3818.40)
Seaquest	4090.36 (1455.73)	**8066.78** (1646.51)	6476.20 (1952.12)

To compare against the mellowmax operator, we combine it with DQN in the same off-policy fashion as the softmax operator [5]. Note that the root-finding approach for mellowmax in Asadi & Littman (2017) needs to compute the optimal inverse temperature parameter for every state, and thus is too computationally expensive to be applied here. Consequently, we tune the inverse temperature parameter for both softmax and mellowmax operators from $\{1, 5, 10\}$, and report the best scores. Table 2 shows that both softmax and mellowmax operators can achieve higher test scores than the max operator. Furthermore, the scores from softmax are higher or similar to those of mellowmax on all games except Asterix. The competitive performance of the softmax operator here suggests that it is still preferable in certain domains, despite its non-contraction property.

[5]We tried the mellowmax operator for exploration as in Asadi & Littman (2017), but observed much worse performance.

5. Why Softmax Helps?

One may wonder why the softmax Bellman operator can achieve the surprising performance in Section 4, as the greedy policy from the Bellman equation suggests that the max operator should be optimal. The softmax operator is not used for exploration in this paper, nor is it motivated by regularizing the policy, as in the entropy regularized MDPs (Fox et al., 2016; Schulman et al., 2017; Neu et al., 2017; Nachum et al., 2017; Asadi & Littman, 2017; Lee et al., 2018).

As discussed in earlier work (van Hasselt et al., 2016a; Anschel et al., 2017), the max operator leads to the significant issue of overestimation for the Q-function in DQNs, which further causes excessive gradient noise to destabilizes the optimization of neural networks. Here we aim to provide analysis of how the softmax Bellman operator can overcome these issues, and also the corresponding evidence for overestimation and gradient noise reduction in DQNs. Although our analysis is focused on the softmax Bellman operator, this work could potentially give further insight into practical benefits of entropy regularization as well.

5.1. Overestimation Bias Reduction

Q-learning's overestimation bias, due to the max operator, was first discussed in Thrun & Schwartz (1994). It was later shown in van Hasselt et al. (2016a) and Anschel et al. (2017) that overestimation leads to the poor performance of DQNs in some Atari games. Following the same assumptions as van Hasselt et al. (2016a), we can show the softmax operator reduces the overestimation bias. Furthermore, we quantify the range of the overestimation reduction, by providing both lower and upper bounds.

Theorem 4. *Given the same assumptions as van Hasselt et al. (2016a), where $(A1)$ there exists some $V^*(s)$ such that the true state-action value function satisfies $Q^*(s,a) = V^*(s)$, for different actions. $(A2)$ the estimation error is modeled as $Q_t(s,a) = V^*(s) + \epsilon_a$, then*

(I) the overestimation errors from $\mathcal{T}_{soft}$ are smaller or equal to those of $\mathcal{T}$ using the max operator, for any $\tau \geq 0$;

(II) the overestimation reduction by using $\mathcal{T}_{soft}$ in lieu of $\mathcal{T}$ is within $\left[\frac{\widehat{\delta}(s)}{m \exp[\tau\, \widehat{\delta}(s)]},\ (m-1)\max\{\frac{1}{\tau+2}, \frac{2Q_{max}}{1+\exp(\tau)}\}\right]$;

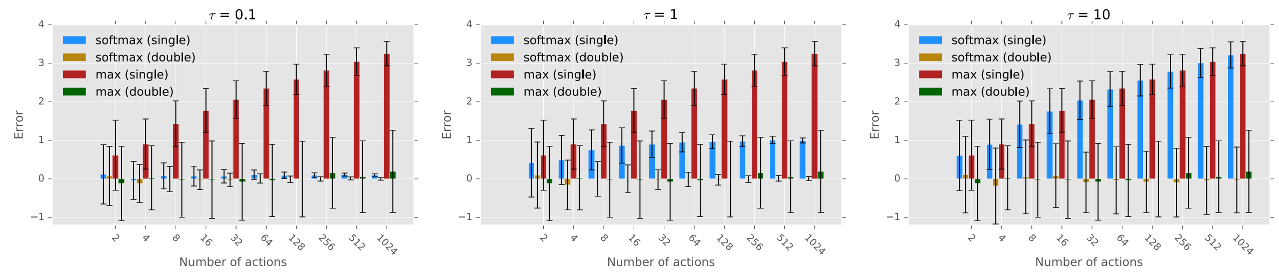

Figure 2. The mean and one standard deviation for the overestimation error for different values of τ.

(III) *the overestimation error for $\mathcal{T}_{soft}$ monotonically increases w.r.t.* $\tau \in [0, \infty)$.

An observation about Theorem 4 is that for any positive value of τ, there will still be some potential for overestimation bias because noise can also influence the softmax operator. Depending upon the amount of noise, it is possible that, unlike double DQN, the reduction caused by softmax could exceed the bias introduced by max. This can be seen in our experimental results below, which show that it is possible to have negative error (overcompensation) from the use of softmax. However, when combined with double Q-learning, this effect becomes very small and decreases with the number of actions.

To elucidate Theorem 4, we simulate standard normal variables ϵ_a with 100 independent trials for each action a, using the same setup as van Hasselt et al. (2016a). Figure 2 shows the mean and one standard deviation of the overestimation bias, for different values of τ in the softmax operator. For both single and double implementations of the softmax operator, they achieve smaller overestimation errors than their max counterparts, thus validating our theoretical results. Note that the gap becomes smaller when τ increases, which is intuitive as $\mathcal{T}_{\text{soft}} \to \mathcal{T}$ when $\tau \to \infty$, and also consistent with the monotonicity result in Theorem 4.

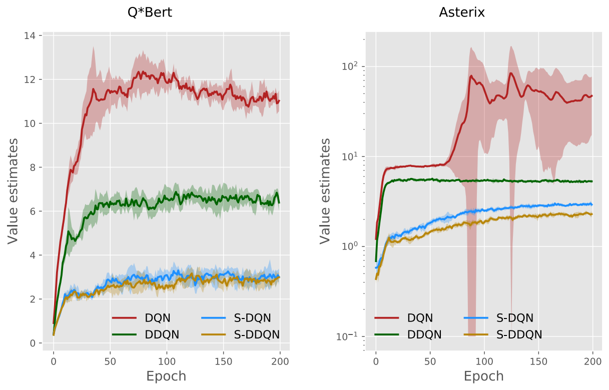

Figure 3. Mean and one standard deviation of the estimated Q-values on the Atari games, for different methods.

We further check the estimated Q-values on Atari games [6], by plotting the corresponding mean and one standard deviation in Figure 3. Both S-DQN and S-DDQN achieve smaller values than DQN, which verifies that the softmax operator can reduce the overestimation bias. This reduction can partly explain the higher test scores for S-DQN and S-DDQN in Figure 1. Moreover, we plot the estimated Q-values of S-DDQN for different values of τ in Figure 4, and the result is consistent with Theorem 4 that increasing τ leads to larger estimated Q-values and, thus, more risk of overestimation. It also shows that picking a value of τ that is very small may introduce a risk of underestimation that can hurt performance, as shown for Asterix, with $\tau = 1$. (See also Table 1.)

[6]Due to the space limit, we plot fewer games from here, and provide the full plots in Supplemental Material.

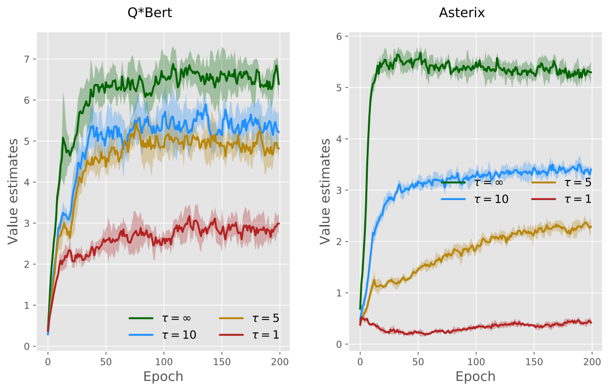

Figure 4. Mean and one standard deviation of the estimated Q-values on the Atari games, for different values of τ in S-DDQN.

5.2. Gradient Noise Reduction

Mnih et al. (2015) observed that large gradients can be detrimental to the optimization of DQNs. This was noted in the context of large reward values, but it is possible that overestimation bias can also introduce variance in the gradient and have a detrimental effect. To see this, we first notice that the gradient in the DQN can be represented as

$$\nabla_{\boldsymbol{\theta}} \mathcal{L} = \mathbb{E}_{s,a,r,s'} \Big\{ \left[Q_{\boldsymbol{\theta}}(s,a) - \mathcal{T} Q_{\boldsymbol{\theta}^-}(s,a) \right] \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(s,a) \Big\}. \tag{5}$$

When the Q-value is overestimated, the magnitude for $\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(s,a)$ in Eq. (5) could become excessively large and so does the gradient, which may further lead to the instability during optimization. This is supported by Figure 3, which shows higher variance in Q-values for DQN vs. others. Since we show in Section 5.1 that using softmax in lieu of max in the Bellman operator can reduce the overestimation bias, the gradient estimate in Eq. (5) can be more stable with softmax. Note that stabilizing the gradient during DQN training was also shown in van Hasselt et al. (2016b), but in the context of adaptive target normalization.

To evaluate the gradient noise on Atari games, we report the ℓ_2 norm of the gradient in the final fully-connected linear layer, by averaging over 50 independent inputs. As shown in Figure 5, S-DQN and S-DDQN achieve smaller gradient norm and variance than their counterparts, which implies that the softmax operator can facilitate variance reduction in the optimization for the neural networks. For the game Asterix, we also observe that the overestimation of Q-value in DQN eventually causes the gradient to explode, while its competitors avoid this issue by achieving reasonable Q-value estimation. Finally, Figure 6 shows that increasing τ generally leads to higher gradient variance, which is also consistent with our analysis in Theorem 4 that the overestimation monotonically increases w.r.t. τ.

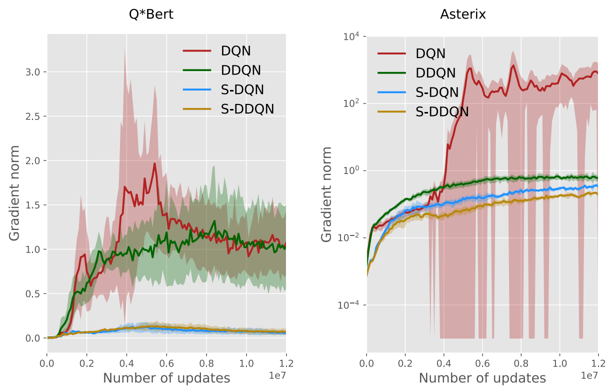

Figure 5. Mean and one standard deviation of the gradient norm on the Atari games, for different methods.

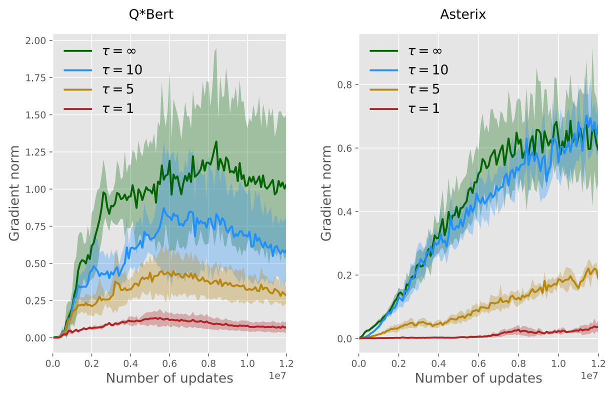

Figure 6. Mean and one standard deviation of the gradient norm on the Atari games, for different values of τ in S-DDQN.

6. A Comparison for Bellman Operators

So far we have covered three different types of Bellman operators, where the softmax and mellowmax variants employ the softmax and the log-sum-exp functions in lieu of the max function in the standard Bellman operator, respectively. A thorough comparison among these Bellman operators will not only exhibit their differences and connections, but also reveal the trade-offs when selecting them.

Table 3. A comparison of different Bellman operators (B.O. Bellman optimality; O.R. overestimation reduction; P.R. policy representation; D.Q. double Q-learning).

	B.O.	Tuning	O.R.	P.R.	D.Q.
Max	Yes	No	-	Yes	Yes
Mellowmax	No	Yes	Yes	No	No
Softmax	No	Yes	Yes	Yes	Yes

Table 3 shows the comparison from different criteria: Bellman equation implies that the optimal greedy policy corresponds to the max operator only. In contrast to the mellowmax and the softmax operators, max also does not need to tune the inverse temperature parameter. On the other hand, the overestimation bias rooted in the max operator can be alleviated by either of the softmax and mellowmax operators. Furthermore, as noted in Asadi & Littman (2017), the mellowmax operator itself cannot directly represent a policy, and needs to be transformed into a softmax policy, where numerical methods are necessary to determine the corresponding state-dependent temperature parameters. The lack of an explicit policy representation also prevents the mellowmax operator from being directly applied in double Q-learning.

For mellowmax and softmax, it is interesting to further investigate their relationship, especially given their comparable performance in the presence of function approximation, as shown in Table 2. First, we notice the following equivalence between these two operators, in terms of the Bellman updates: For the softmax-weighted Q-function, i.e., $g_{Q(s',)}(\tau) = \sum_{a'} \frac{\exp[\tau\, Q(s',a')]}{\sum_{\bar{a}} \exp[\tau\, Q(s',\bar{a})]} Q(s',a')$, there exists an ω such that the mellowmax-weighted Q-function achieves the same value. This can be verified by the facts that both softmax and mellowmax operators (i) converge to the mean and max operators when the inverse temperature parameter approaches 0 and ∞, respectively; (ii) are continuous on τ and ω, with $[0, \infty)$ as the support.[7]

Furthermore, we compare via simulation the approximation error to the max function and the overestimation error, for

[7]The operators remain fundamentally different since this equivalence would hold only for a particular set of Q-values. The conversion between softmax and mellomax parameters would need to be done on a state-by-state basis, and would change as the Q-values are updated.

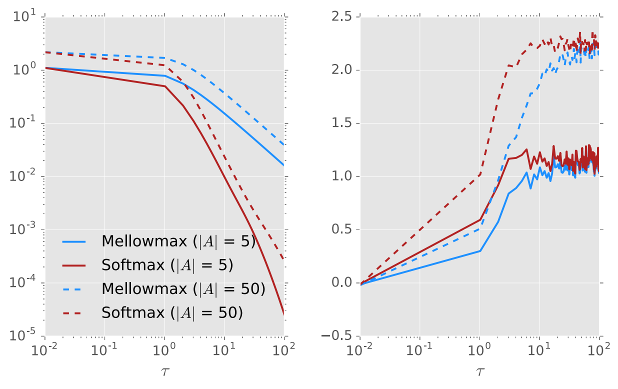

Figure 7. (Left) Approximation error to the max function; (Right) Overestimation error, as a function of τ (softmax) or ω (mellowmax).

mellowmax and softmax. The setup is as follows: The inverse temperature parameter τ is chosen from a linear grid from 0.01 to 100. We generate standard normal random variables, and compute the weighted Q-functions for the cases with different number of actions. The approximation error is then measured in terms of the difference between the max function and the corresponding softmax and mellowmax functions. The overestimation error is measured the same as in Figure 2. The result statistics are reported by averaging over 100 independent trials. Figure 7 shows the trade-off between converging to the Bellman optimality and overestimation reduction in terms of the inverse temperature parameter, where the softmax operator approaches the max operator in a faster speed while the mellowmax operator can further reduce the overestimation error.

7. Related Work

Applying the softmax function in Bellman updates has long been viewed as problematic, as Littman (1996) showed that the corresponding Boltzmann operator could be expansive in a specific MDP, via a counterexample. The primary use of the softmax function in RL has been focused on improving the exploration performance (Sutton & Barto, 1998). Our work here instead aims to bring evidence and new perspective that the softmax Bellman operator can still be beneficial beyond exploration.

The mellowmax operator (Asadi & Littman, 2017) was proposed based on the `log-sum-exp` function, which can be proven as a contraction and has received recent attention due to its convenient mathematical properties. The use of the log-sum-exp function in the backup operator dates back to Todorov (2007), where the control cost was regularized by a Kullback-Leibler (KL) divergence on the transition probabilities. Fox et al. (2016) proposed a G-learning scheme with soft updates to reduce the bias for the Q-value estimation, by regularizing the cost with the KL divergence on policies. Asadi & Littman (2017) further applied the log-sum-exp function to the on-policy updates, and showed that the state-dependent inverse temperature parameter can be numerically computed. More recently, the log-sum-exp function has also been used in the Boltzmann backup operator for entropy-regularized RL (Schulman et al., 2017; Haarnoja et al., 2017; Neu et al., 2017; Nachum et al., 2017). One interesting observation is that the optimal policies in these work admit a form of softmax, and thus can be beneficial for exploration. In Lee et al. (2018), a sparsemax operator based on Tsallis entropy was proposed to improve the error bound for mellowmax. Note that these operators were not shown to explicitly address the instability issues in the DQN, and their corresponding policies were used to improve the exploration performance, which is different from our focus here.

Among variants of the DQN in Mnih et al. (2015), DDQN (van Hasselt et al., 2016a) first identified the overestimation bias issue caused by the max operator, and mitigated it via the double Q-learning algorithm (van Hasselt, 2010). Anschel et al. (2017) later demonstrated that averaging over the previous Q-value estimates in the learning target can also reduce the bias, while the analysis for the bias reduction is restricted to a small MDP with a special structure. Furthermore, an adaptive normalization scheme was developed in van Hasselt et al. (2016b) and was demonstrated to reduce the gradient norm. The softmax function was also employed in the categorical DQN (Bellemare et al., 2017), but with a different purpose of generating distributions in its distributional Bellman operator. Finally, we notice that other variants (Wang et al., 2016; Mnih et al., 2016; Schaul et al., 2016; He et al., 2017; Hessel et al., 2018; Dabney et al., 2018) have also been proposed to improve the vanilla DQN, though they were not explicitly designed to tackle the issues of overestimation error and gradient noise.

8. Conclusion and Future Work

We demonstrate surprising benefits of the softmax Bellman operator when combined with DQNs, which further suggests that the softmax operator can be used as an alternative for the double Q-learning to reduce the overestimation bias. Our theoretical results provide new insight into the convergence properties of the softmax Bellman operator, and quantify how it reduces the overestimation bias. To gain a deep understanding about the relationship among different Bellman operators, we further compare them from different criteria, and show the trade-offs when choosing them in practice.

An interesting direction for future work is to provide more theoretical analysis for the performance trade-off when selecting the inverse temperature parameter in $\mathcal{T}_{\text{soft}}$, and subsequently design an efficient cooling scheme.

Acknowledgements

We thank the anonymous reviewers for their helpful comments and suggestions. This work was partially supported by NSF RI grant 1815300. Opinions, findings, conclusions or recommendations herein are those of the authors and not necessarily those of the NSF.

References

Anschel, O., Baram, N., and Shimkin, N. Averaged-DQN: variance reduction and stabilization for deep reinforcement learning. In *International Conference on Machine Learning*, pp. 176–185, 2017.

Asadi, K. and Littman, M. L. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pp. 243–252, 2017.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458, 2017.

Bellman, R. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

Dabney, W., Rowland, M., Bellemare, M. G., and Munos, R. Distributional reinforcement learning with quantile regression. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Fox, R., Pakman, A., and Tishby, N. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 202–211. AUAI Press, 2016.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361, 2017.

He, F. S., Liu, Y., Schwing, A. G., and Peng, J. Learning to play in a day: Faster deep reinforcement learning by optimality tightening. In *International Conference on Learning Representations*, 2017.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Lee, K., Choi, S., and Oh, S. Sparse Markov decision processes with causal sparse Tsallis entropy regularization for reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3):1466–1473, July 2018.

Littman, M. L. *Algorithms for Sequential Decision Making*. PhD thesis, Department of Computer Science, Brown University, February 1996.

Mitra, D., Romeo, F., and Sangiovanni-Vincentelli, A. Convergence and finite-time behavior of simulated annealing. *Advances in applied probability*, 18(3):747–771, 1986.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 02 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1928–1937, 2016.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2772–2782, 2017.

Neu, G., Jonsson, A., and Gómez, V. A unified view of entropy-regularized Markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

O'Donoghue, B., Munos, R., Kavukcuoglu, K., and Mnih, V. Combining policy gradient and Q-learning. In *International Conference on Learning Representations*, 2017.

Reverdy, P. and Leonard, N. E. Parameter estimation in softmax decision-making models with linear objective functions. *IEEE Transactions on Automation Science and Engineering*, 13(1):54–67, 2016.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

Schulman, J., Chen, X., and Abbeel, P. Equivalence between policy gradients and soft Q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

Thrun, S. and Schwartz, A. Issues in using function approximation for reinforcement learning. In Mozer,

M. C., Smolensky, P., Touretzky, D. S., Elman, J. L., and Weigend, A. S. (eds.), *Proceedings of the 1993 Connectionist Models Summer School*, Hillsdale, NJ, 1994. Lawrence Erlbaum.

Thrun, S. B. The role of exploration in learning control. In White, D. A. and Sofge, D. A. (eds.), *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, pp. 527–559. Van Nostrand Reinhold, New York, NY, 1992.

Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4 (2):26–31, 2012.

Todorov, E. Linearly-solvable Markov decision problems. In *Advances in neural information processing systems*, pp. 1369–1376, 2007.

van Hasselt, H. Double Q-learning. In *Advances in Neural Information Processing Systems*, pp. 2613–2621, 2010.

van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double Q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2094–2100. AAAI Press, 2016a.

van Hasselt, H. P., Guez, A., Hessel, M., Mnih, V., and Silver, D. Learning values across many orders of magnitude. In *Advances in Neural Information Processing Systems*, pp. 4287–4295, 2016b.

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., and De Freitas, N. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1995–2003. JMLR. org, 2016.

Williams, R. J. and Baird III, L. C. Tight performance bounds on greedy policies based on imperfect value functions. Technical report, Northeastern University, 1993.