Deep Reinforcement Learning with Double Q-learning

Jacob, Kawin & Sasanka

Indian Institute of Science, Bengaluru

April 14, 2022

Overview

1. Motivation
2. Q-learning: Background
3. Deep Q-Networks
   - Experience Replay
   - Fixed Targets
   - Algorithm
4. Double Q-learning
5. Overestimation Problem
6. Double Deep Q-Network
7. Results
8. Limitations of DDQN
9. Conclusion
10. References

Motivation

- Deep Q-Networks (DQNs) and Q-learning methods are known to be overestimating the Q value. Performances are lowered due to it.
- How bad is this error? How does it affect the model performance ?
- Double Q-learning is known to be a solution for this overestimation problem. Can we combine this idea with DQN ?

Q-learning with Value Function Approximation

- Use a function approximator (linear or non-linear) to estimate the action-value function.

$$Q_\pi(s, a) \approx Q(s, a; \theta)$$

- Standard update rule:

$$\theta_{t+1} = \theta_t + \alpha_t[Y_t^Q - Q(S_t, A_t; \theta_t)]\nabla_{\theta_t}(Q(S_t, A_t; \theta_t))$$

where,

$$Y_t^Q = R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta)$$

Deep Q-Networks [Mnih et al., 2015]

- Non-linear function approximation.
- A multi-layered neural network that for a given state s outputs a vector of action values $Q(s, \cdot\,; \theta)$, where θ are the parameters of the network.
- Q-learning with VFA can diverge because of two issues:
  - Correlation between samples
  - Non-stationary targets
- DQN addresses both these challenges by:
  - Experience Replay
  - Fixed targets using a target network

DQNs: Experience Replay

- Store dataset $\mathcal{D}$ (called replay buffer) from prior experience:

s_1, a_1, r_2, s_2
s_2, a_2, r_3, s_3
...
$s_t, a_t, r_{t+1}, s_{t+1}$

- To apply experience replay:
  - $(s, a, r, s') \sim \mathcal{D}$. Uniformly sample an experience tuple from $\mathcal{D}$.
  - Apply Q-learning updates on samples (or minibatches) of experiences.

DQNs: Fixed Targets

- To help improve stability, fix the **target network** weights used in the target calculation for multiple updates.
- Use a different set of weights to compute target than is being updated
- Let parameters θ^- be the set of weights used in the target, and θ be the weights that are being updated.
- Resulting SGD rule:

$$\theta_{t+1} = \theta_t + \alpha_t[Y_t^{DQN} - Q(S_t, A_t; \theta_t)]\nabla_{\theta_t}(Q(S_t, A_t; \theta_t))$$

where,

$$Y_t^{DQN} = S_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta_t^-)$$

Double Q-learning

- The max operator in standard Q-learning and DQN, uses the same value function both to select and to evaluate an action. This makes it more likely to select overestimated values, resulting in overoptimistic value estimates.

$$Y_t^Q \equiv R_{t+1} + \gamma \max_{\boldsymbol{a}} Q(S_{t+1}, a; \theta_t)$$

Idea: Decouple the selection from the evaluation (**Double Q-learning**)

- In Double Q-learning, two value functions are learned by assigning each experience randomly to update one of the two value functions, such that there are two sets of weights, θ and θ'.
- For each update in Double Q-learning, one set of weights is used to determine the greedy policy and the other to determine its value.

Double Q-learning

Q-learning Target

$$Y_t^Q \equiv R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \boldsymbol{\theta_t})$$

Untangling the selection and evaluation in Q-learning

$$Y_t^Q \equiv R_{t+1} + \gamma Q(S_{t+1}, \underset{a}{argmax}\, Q(S_{t+1}, A_{t+1}; \boldsymbol{\theta_t}); \boldsymbol{\theta_t})$$

Double Q-learning Target

$$Y_t^{DoubleQ} \equiv R_{t+1} + \gamma Q(S_{t+1}, \underset{a}{argmax}\, Q(S_{t+1}, A_{t+1}; \boldsymbol{\theta_t}); \boldsymbol{\theta'_t})$$

Overestimation Problem

- Q-learning methods are known to be overestimating the Q-value.
- In this paper, it is shown more generally that estimation errors of any kind can induce an upward bias, regardless of the source of the error.

Overoptimism due to estimation errors

Theorem (Lower Bound on Estimation Errors)

Consider a state s in which all the true optimal action values are equal at $\mathbb{Q}_*(s,a) = \mathbb{V}_*(s)$ for some $\mathbb{V}_*(s)$. Let Q_t be arbitrary value estimates that are on the whole unbiased in the sense that $\sum_a (Q_t(s,a) - \mathbb{V}_*(s)) = 0$, but that are not all correct, such that $\frac{1}{m}\sum_a (Q_t(s,a) - \mathbb{V}_*(s))^2 = C$ for some $C > 0$, where $m \geq 2$ is the number of actions in s.

Under these conditions, $max_a Q_t(s,a) \geq \mathbb{V}_*(s) + \sqrt{\frac{C}{m-1}}$. This lower bound is tight.

Under the same conditions, the lower bound on the absolute error of the Double Q-learning estimate is **zero**.

Overestimation Problem

- In real cases, the Q-learning's estimation error grows as number of actions increases, while the Double Q-learning is unbiased.

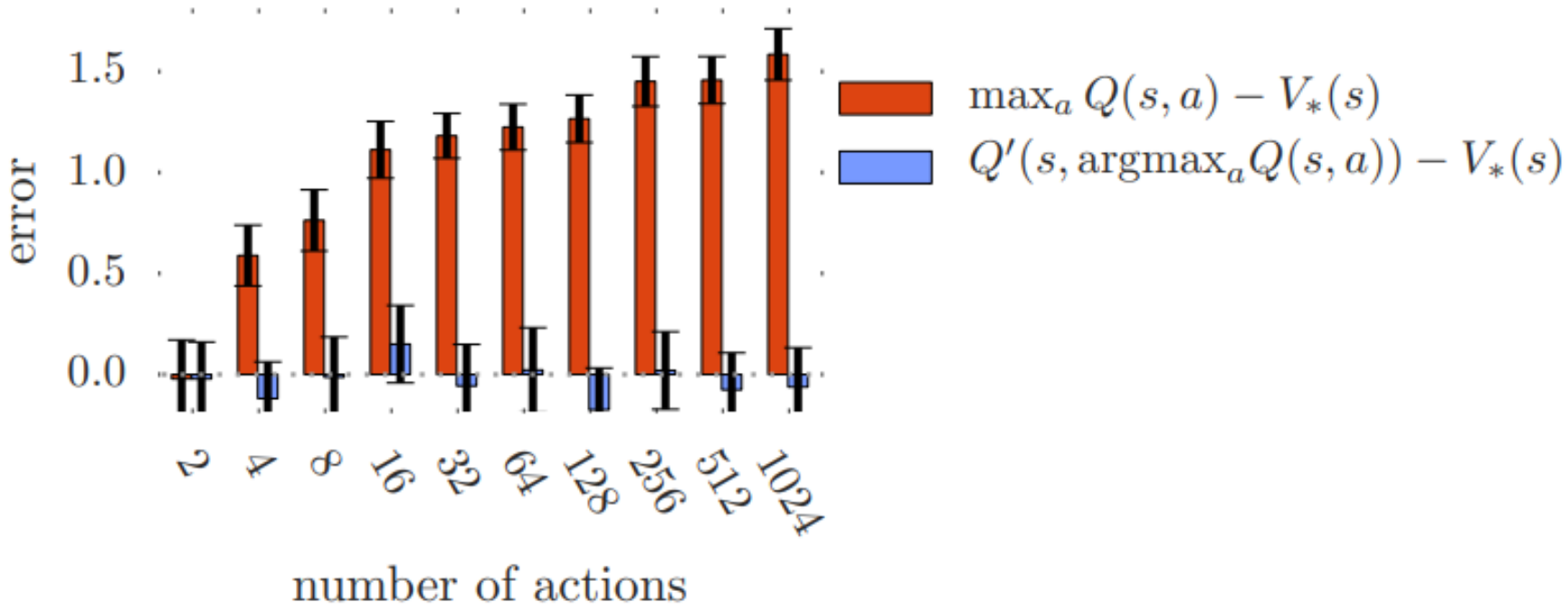
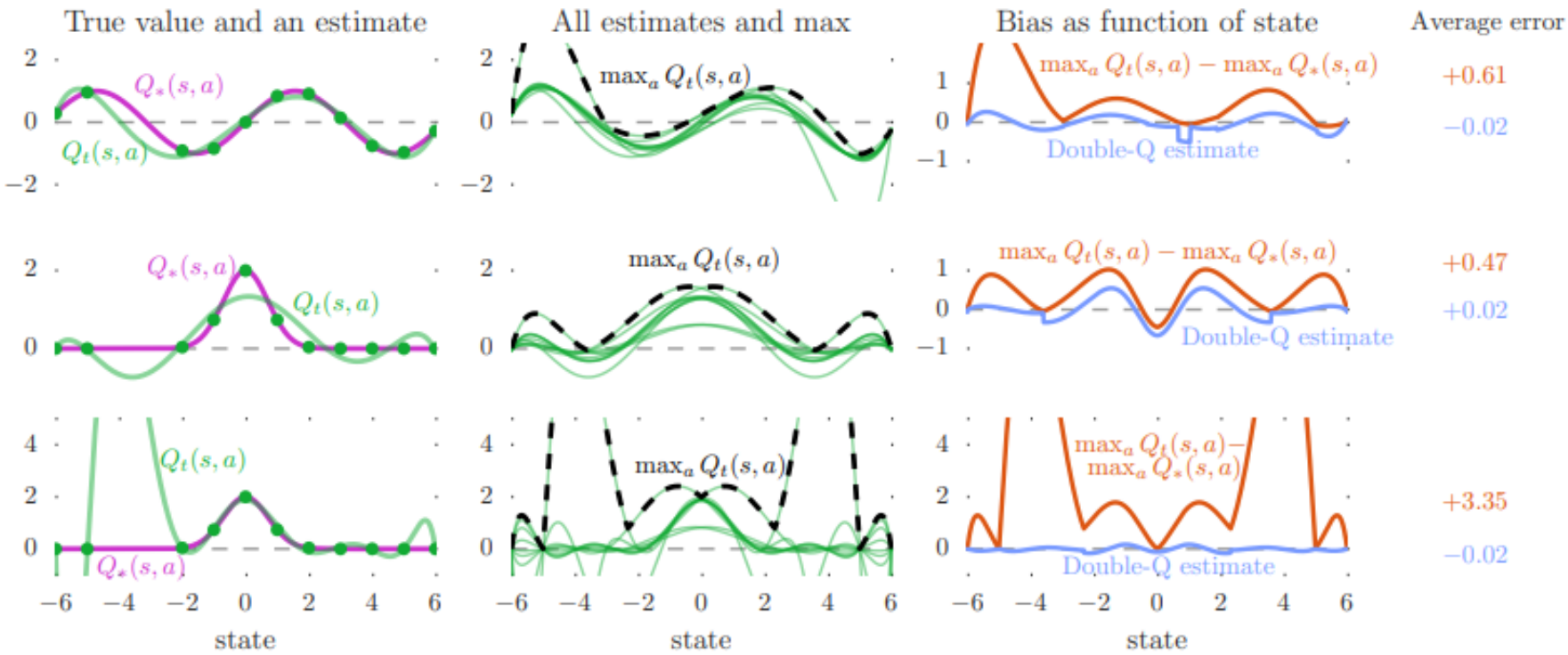


Figure: Estimation Error vs Number of Actions

Overestimation Problem

- Over-estimations occur even when assuming we have samples of the true action value at certain states and also occur irrespective of the capacity of the fitted function.



Double Deep Q-Network

Double Deep Q-Network combines the ideas of Deep Q-Network with Q Learning. DQN Target value

$$Y_t^{DQN} = R_{t+1} + \gamma \max_a Q_t(S_{t+1}, a; \boldsymbol{\theta}^-)$$

Rewriting it to Double Q-form

$$Y_t^{DQN} = R_{t+1} + \gamma Q_t(S_{t+1}, \underset{a}{argmax}\, Q(S_{t+1}, a, \boldsymbol{\theta_t^-}); \boldsymbol{\theta_t^-})$$

Double DQN Target value

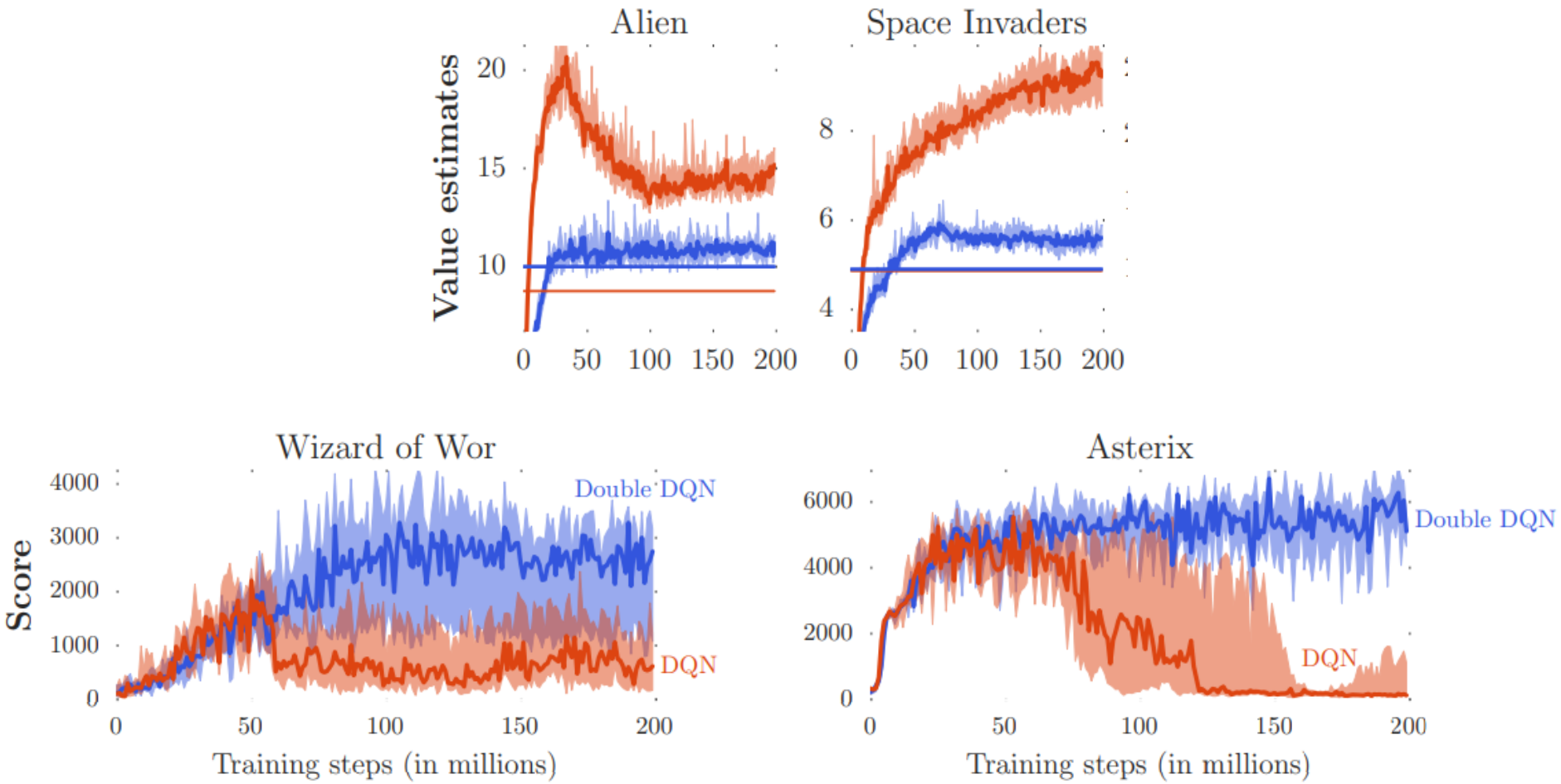
$$Y_t^{DoubleDQN} = R_{t+1} + \gamma Q_t(S_{t+1}, \underset{a}{argmax}\, Q(S_{t+1}, a, \boldsymbol{\theta_t}); \boldsymbol{\theta_t^-})$$

In Double DQN, the max operation in Y_t is replaced by an action selection followed by an action evaluation (concept used in Double Q Learning).

DDQN Results

Evaluation over Atari 2600

Results on Overoptimisim.



DDQN Results

Quality of Learned policy

	DQN	Double DQN
Median	93.5%	114.7%
Mean	241.1%	330.3%

- The learned policies were evaluated for a certain number of episodes and the scores were collected in each episode
- The Normalized performance was calculated as

$$score_{normalized} = \frac{score_{agent} - score_{random}}{score_{human} - score_{random}} * 100$$

- Comparitive results show that Double DQN is able to learn better polices compared to DQN.

Simulation Results

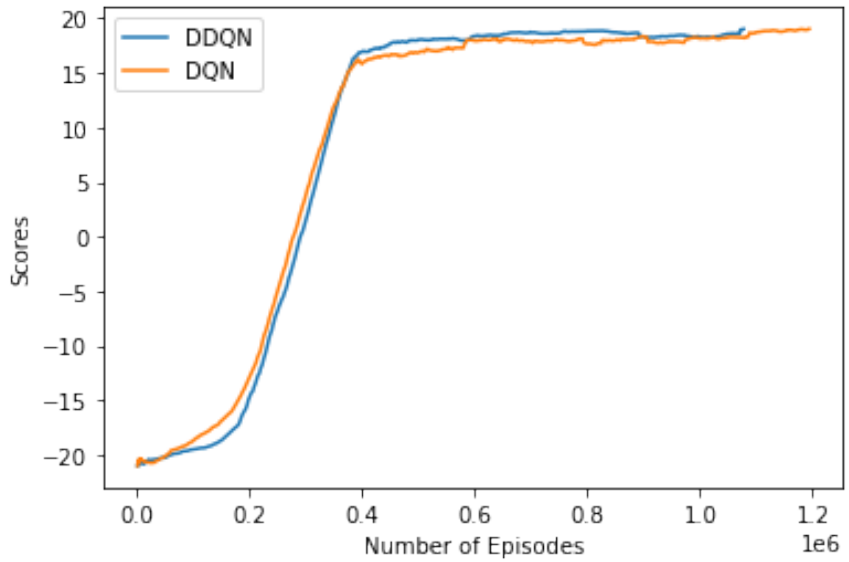


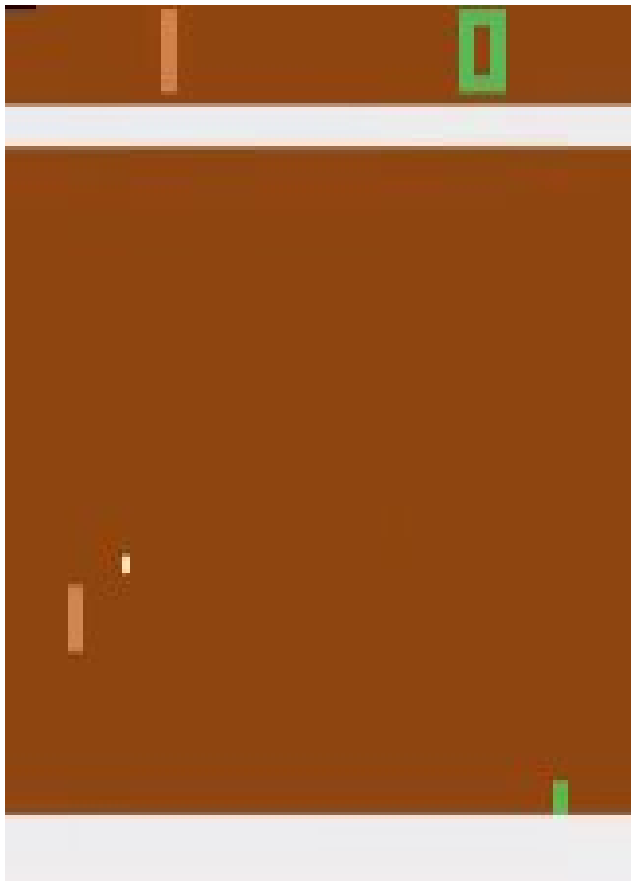
Figure: Evolution of Double DQN and DQN Scores in the Atari Game Pong

Game	Random	Human	DQN	Double DQN
Pong	-20.70	9.30	18.90	21.00

Figure: Scores attained in Pong (Paper)

Limitations of DDQN

- The experiences are sampled randomly from the replay buffer. Sometimes the experience sample chosen may be rarely encountered after training. If we can prioritize samples from the replay buffer for selection we can attain better scores faster during the evaluation process.



- The

Conclusion

- Q Learning can be over optimistic due to estimation errors during the learning process
- Double Q Learning can be used to reduce this over optimism
- The paper introduces Double DQN having which uses the same architecture as that of DQN but incorporating principles from Double Q Learning to reduce overoptimizations.

References

van Hasselt, Hado, Guez, Arthur, and Silver, David

Deep Reinforcement Learning with Double Q learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016. URL http://arxiv.org/abs/1509.06461.

Mnih et al., 2013 V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, et al.

Playing Atari with deep reinforcement learning. Technical report Deepmind Technologies (2013)

Mnih, V., Kavukcuoglu, K., Silver, D. et al.

Human-level control through deep reinforcement learning. Nature 518, 529–533 (2015). https://doi.org/10.1038/nature14236

Hasselt, Hado V.

Double q-learning. In Advances in Neural Information Processing Systems, pp. 2613–2621, 2010.

Thank You

Appendix

Theorem 1. *Consider a state s in which all the true optimal action values are equal at $Q_*(s,a) = V_*(s)$ for some $V_*(s)$. Let Q_t be arbitrary value estimates that are on the whole unbiased in the sense that $\sum_a (Q_t(s,a) - V_*(s)) = 0$, but that are not all zero, such that $\frac{1}{m}\sum_a (Q_t(s,a) - V_*(s))^2 = C$ for some $C > 0$, where $m \geq 2$ is the number of actions in s. Under these conditions, $\max_a Q_t(s,a) \geq V_*(s) + \sqrt{\frac{C}{m-1}}$. This lower bound is tight. Under the same conditions, the lower bound on the absolute error of the Double Q-learning estimate is zero.*

Proof of Theorem 1. Define the errors for each action a as $\epsilon_a = Q_t(s,a) - V_*(s)$. Suppose that there exists a setting of $\{\epsilon_a\}$ such that $\max_a \epsilon_a < \sqrt{\frac{C}{m-1}}$. Let $\{\epsilon_i^+\}$ be the set of positive ϵ of size n, and $\{\epsilon_j^-\}$ the set of strictly negative ϵ of size $m-n$ (such that $\{\epsilon\} = \{\epsilon_i^+\} \cup \{\epsilon_j^-\}$). If $n = m$, then $\sum_a \epsilon_a = 0 \implies \epsilon_a = 0 \ \forall a$, which contradicts $\sum_a \epsilon_a^2 = mC$. Hence, it must be that $n \leq m-1$. Then, $\sum_{i=1}^n \epsilon_i^+ \leq n \max_i \epsilon_i^+ < n\sqrt{\frac{C}{m-1}}$, and therefore (using the constraint $\sum_a \epsilon_a = 0$) we also have that $\sum_{j=1}^{m-n} |\epsilon_j^-| < n\sqrt{\frac{C}{m-1}}$. This implies $\max_j |\epsilon_j^-| < n\sqrt{\frac{C}{m-1}}$. By Hölder's inequality, then

$$\begin{aligned}\sum_{j=1}^{m-n} (\epsilon_j^-)^2 &\leq \sum_{j=1}^{m-n} |\epsilon_j^-| \cdot \max_j |\epsilon_j^-| \\ &< n\sqrt{\frac{C}{m-1}} n\sqrt{\frac{C}{m-1}}.\end{aligned}$$

We can now combine these relations to compute an upper-bound on the sum of squares for all ϵ_a:

$$\begin{aligned}\sum_{a=1}^{m}(\epsilon_a)^2 &= \sum_{i=1}^{n}(\epsilon_i^+)^2 + \sum_{j=1}^{m-n}(\epsilon_j^-)^2 \\ &< n\frac{C}{m-1} + n\sqrt{\frac{C}{m-1}}n\sqrt{\frac{C}{m-1}} \\ &= C\frac{n(n+1)}{m-1} \\ &\leq mC.\end{aligned}$$

This contradicts the assumption that $\sum_{a=1}^{m}\epsilon_a^2 < mC$, and therefore $\max_a \epsilon_a \geq \sqrt{\frac{C}{m-1}}$ for all settings of ϵ that satisfy the constraints. We can check that the lower-bound is tight by setting $\epsilon_a = \sqrt{\frac{C}{m-1}}$ for $a = 1, \ldots, m-1$ and $\epsilon_m = -\sqrt{(m-1)C}$. This verifies $\sum_a \epsilon_a^2 = mC$ and $\sum_a \epsilon_a = 0$.

The only tight lower bound on the absolute error for Double Q-learning $|Q'_t(s, \operatorname{argmax}_a Q_t(s,a)) - V_*(s)|$ is zero. This can be seen by because we can have

$$Q_t(s, a_1) = V_*(s) + \sqrt{C\frac{m-1}{m}},$$

and

$$Q_t(s, a_i) = V_*(s) - \sqrt{C\frac{1}{m(m-1)}}, \text{ for } i > 1.$$

Then the conditions of the theorem hold. If then, furthermore, we have $Q'_t(s, a_1) = V_*(s)$ then the error is zero. The remaining action values $Q'_t(s, a_i)$, for $i > 1$, are arbitrary. □