AIL 722: Reinforcement Learning

Lecture 31: Policy Gradient Methods

Raunak Bhattacharyya

ScAI YARDI SCHOOL OF ARTIFICIAL INTELLIGENCE
INDIAN INSTITUTE OF TECHNOLOGY DELHI

Recap & Today's Outline

- Overestimation bias
- Double estimator
- Double Q-Learning
- Policy Gradient methods

Why Does it Work

Lemma 1. *Let $X = \{X_1, \ldots, X_M\}$ be a set of random variables and let $\mu^A = \{\mu_1^A, \ldots, \mu_M^A\}$ and $\mu^B = \{\mu_1^B, \ldots, \mu_M^B\}$ be two sets of unbiased estimators such that $E\{\mu_i^A\} = E\{\mu_i^B\} = E\{X_i\}$, for all i. Let $\mathcal{M} \stackrel{\text{def}}{=} \{j \mid E\{X_j\} = \max_i E\{X_i\}\}$ be the set of elements that maximize the expected values. Let a^* be an element that maximizes μ^A: $\mu_{a^*}^A = \max_i \mu_i^A$. Then $E\{\mu_{a^*}^B\} = E\{X_{a^*}\} \leq \max_i E\{X_i\}$. Furthermore, the inequality is strict if and only if $P(a^* \notin \mathcal{M}) > 0$.*

Proof. Assume $a^* \in \mathcal{M}$. Then $E\{\mu_{a^*}^B\} = E\{X_{a^*}\} \stackrel{\text{def}}{=} \max_i E\{X_i\}$. Now assume $a^* \notin \mathcal{M}$ and choose $j \in \mathcal{M}$. Then $E\{\mu_{a^*}^B\} = E\{X_{a^*}\} < E\{X_j\} \stackrel{\text{def}}{=} \max_i E\{X_i\}$. These two possibilities are mutually exclusive, so the combined expectation can be expressed as

$$
\begin{aligned}
E\{\mu_{a^*}^B\} &= P(a^* \in \mathcal{M})E\{\mu_{a^*}^B | a^* \in \mathcal{M}\} + P(a^* \notin \mathcal{M})E\{\mu_{a^*}^B | a^* \notin \mathcal{M}\} \\
&= P(a^* \in \mathcal{M}) \max_i E\{X_i\} + P(a^* \notin \mathcal{M})E\{\mu_{a^*}^B | a^* \notin \mathcal{M}\} \\
&\leq P(a^* \in \mathcal{M}) \max_i E\{X_i\} + P(a^* \notin \mathcal{M}) \max_i E\{X_i\} \qquad = \max_i E\{X_i\} \ ,
\end{aligned}
$$

Double Q-Learning

- Idea: Don't use same Q estimator for action selection and value estimation

How do we separate action selection and value estimation?

- Use two estimators

$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma Q_2\big(S_{t+1}, \arg\max_a Q_1(S_{t+1}, a)\big) - Q_1(S_t, A_t) \Big]$$

Double Q-Learning

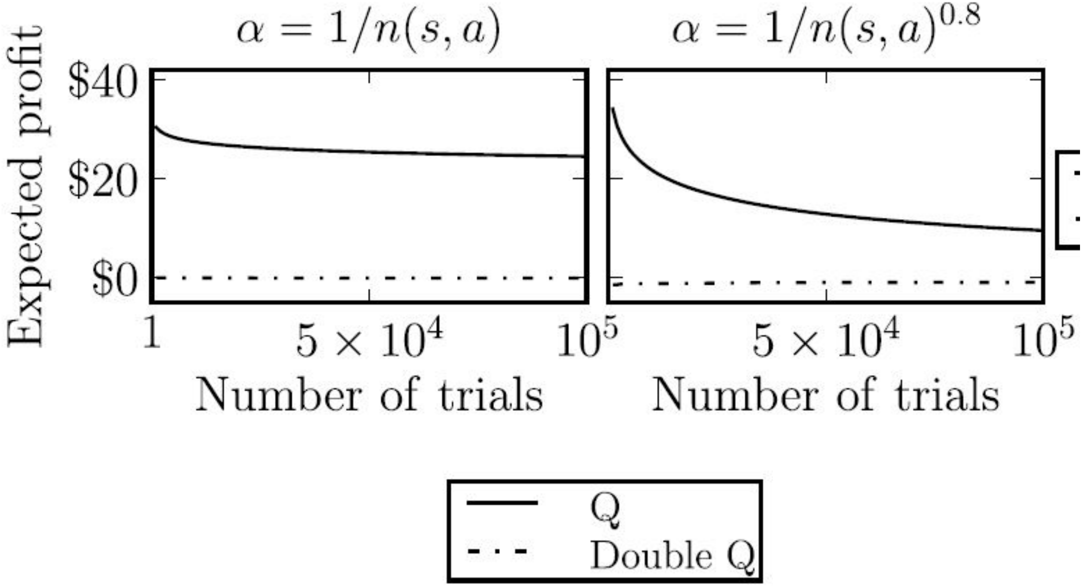
Double Q-learning, for estimating $Q_1 \approx Q_2 \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q_1(s, a)$ and $Q_2(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, such that $Q(terminal, \cdot) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        Choose A from S using the policy ε-greedy in $Q_1 + Q_2$
        Take action A, observe R, S'
        With 0.5 probabilility:
            $Q_1(S, A) \leftarrow Q_1(S, A) + \alpha\Big(R + \gamma Q_2\big(S', \arg\max_a Q_1(S', a)\big) - Q_1(S, A)\Big)$
        else:
            $Q_2(S, A) \leftarrow Q_2(S, A) + \alpha\Big(R + \gamma Q_1\big(S', \arg\max_a Q_2(S', a)\big) - Q_2(S, A)\Big)$
        $S \leftarrow S'$
    until S is terminal

How do we get two estimators in deep Q-Learning?

Overestimation: Roulette Example



- Linear decay
- Polynomial decay

Double Q-Learning, van Hasselt, NeurIPS 2010

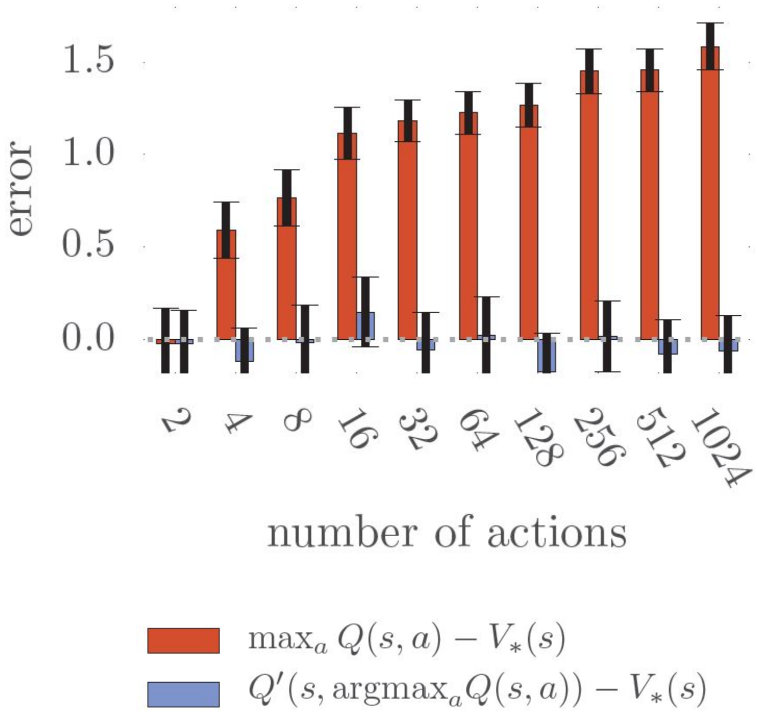
Towards Double DQN

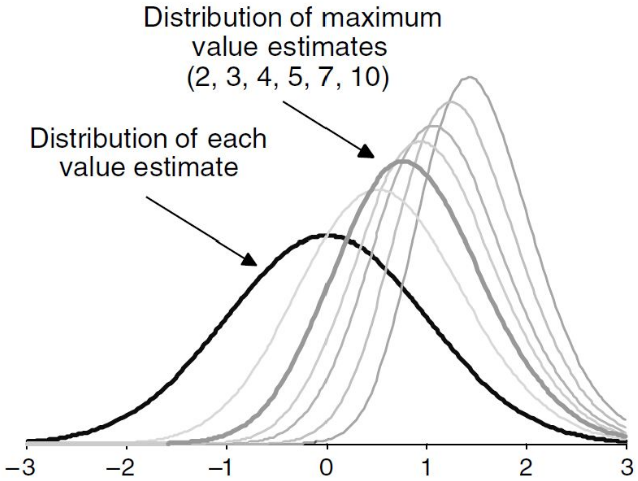
Theorem 1. *Consider a state s in which all the true optimal action values are equal at $Q_*(s,a) = V_*(s)$ for some $V_*(s)$. Let Q_t be arbitrary value estimates that are on the whole unbiased in the sense that $\sum_a (Q_t(s,a) - V_*(s)) = 0$, but that are not all correct, such that $\frac{1}{m}\sum_a (Q_t(s,a) - V_*(s))^2 = C$ 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 in appendix.)*

Even if value estimates are on avg correct, estimation errors of any source can drive the estimates up and away from true optimal values

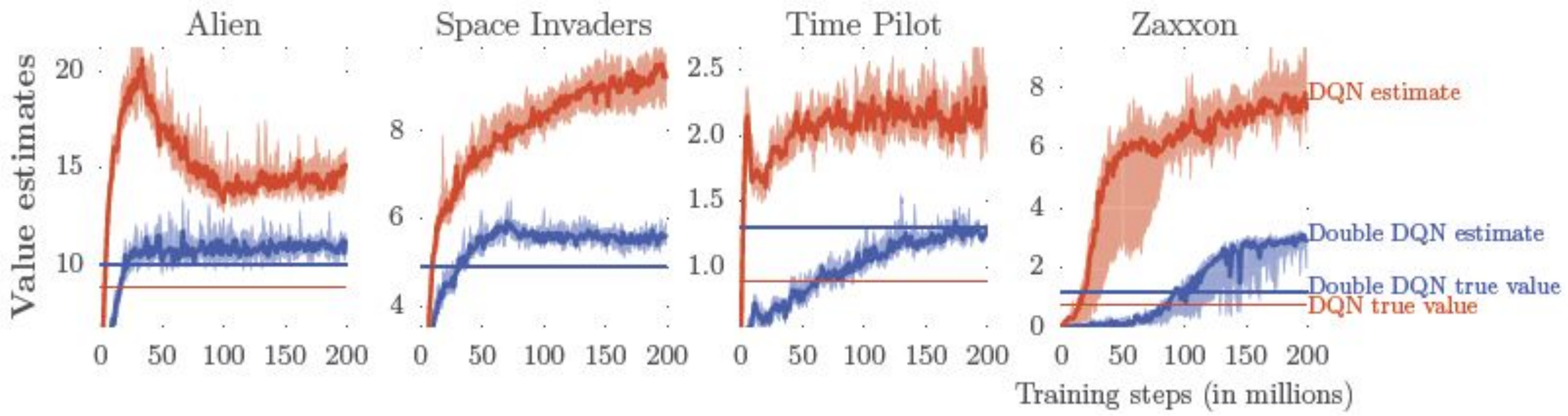
Deep RL with Double Q-Learning, van Hasselt et. al., AAAI 2016

Impact of Double Estimator

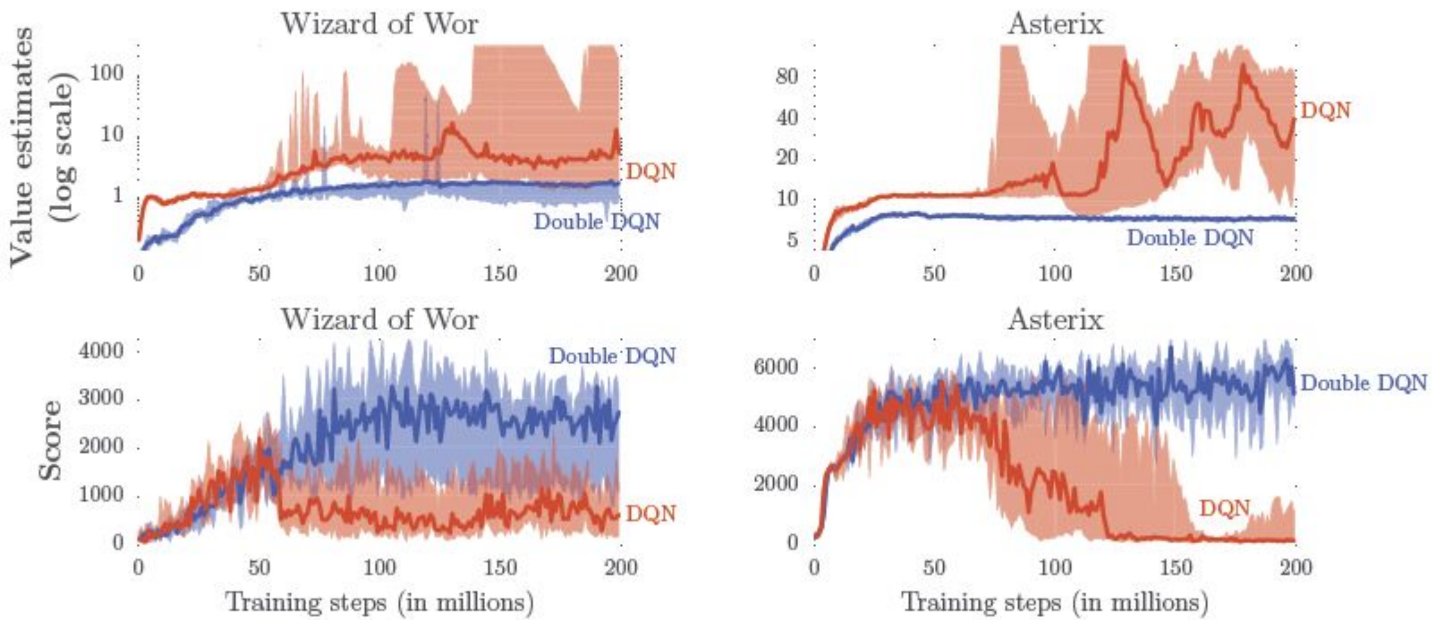




Overestimation: ALE

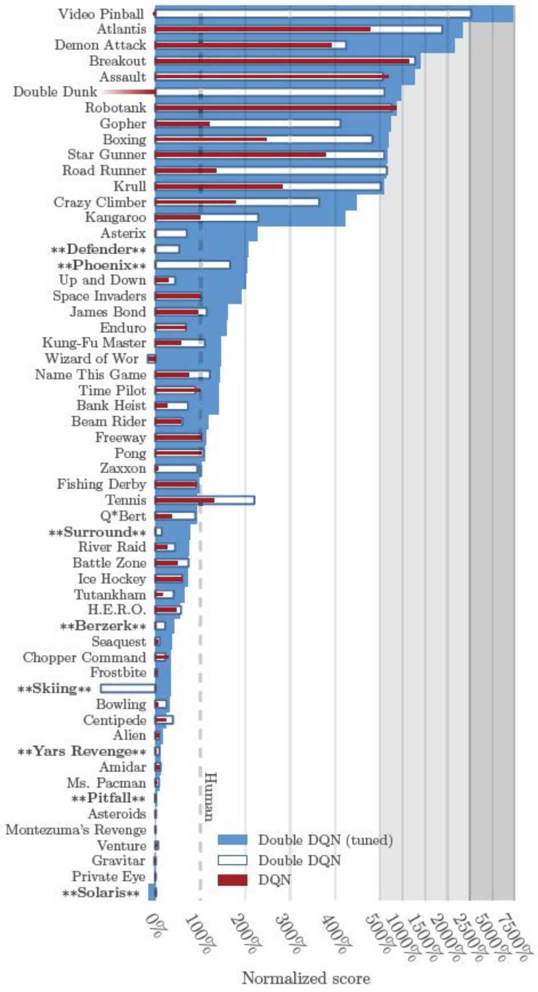


Impact on Performance



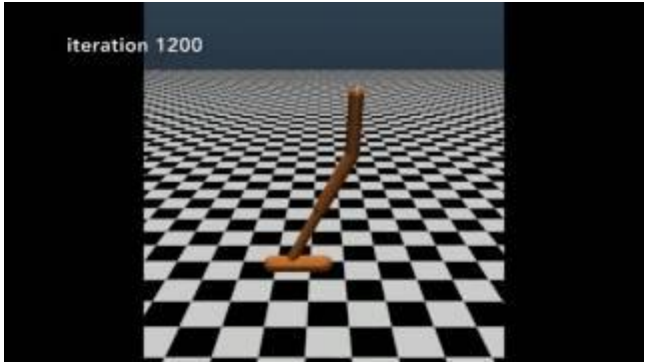
Atari: DQN vs. Double DQN

$$\text{score}_{\text{normalized}} = \frac{\text{score}_{\text{agent}} - \text{score}_{\text{random}}}{\text{score}_{\text{human}} - \text{score}_{\text{random}}}$$



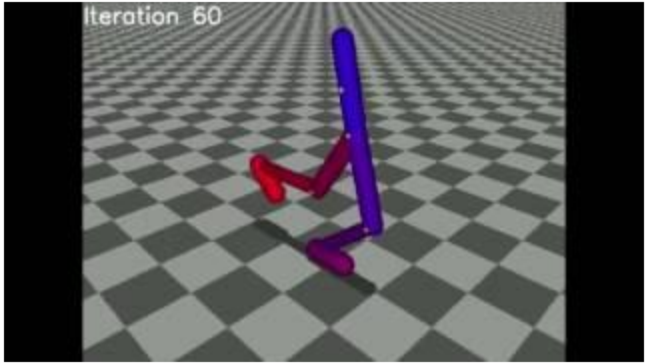
Policy Gradients

Examples



Source: Youtube

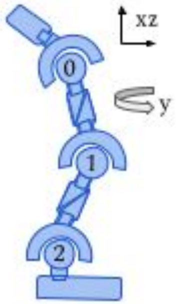
Source: Youtube



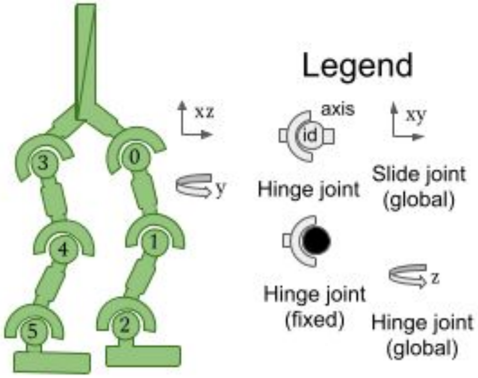
Source: Youtube

Source: Youtube

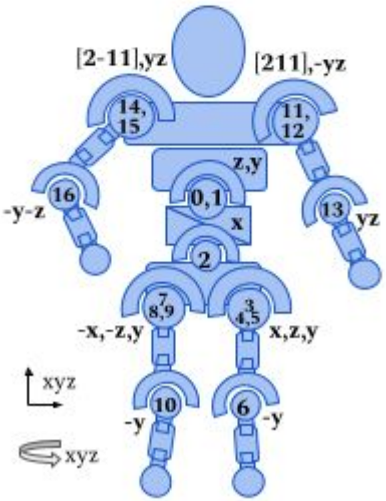
Hopper & Walker



Num	Action	Control Min	Control Max	Name (in corresponding XML file)	Joint	Type (Unit)
0	Torque applied on the thigh rotor	-1	1	thigh_joint	hinge	torque (N m)
1	Torque applied on the leg rotor	-1	1	leg_joint	hinge	torque (N m)
2	Torque applied on the foot rotor	-1	1	foot_joint		



Num	Action	Control Min	Control Max	Name (in corresponding XML file)
0	Torque applied on the thigh rotor	-1	1	thigh_joint
1	Torque applied on the leg rotor	-1	1	leg_joint
2	Torque applied on the foot rotor	-1	1	foot_joint
3	Torque applied on the left thigh rotor	-1	1	thigh_left_joint
4	Torque applied on the left leg rotor	-1	1	leg_left_joint
5	Torque applied on the left foot rotor	-1	1	foot_left_joint



Summary & Announcements

- Summary
  - Double Q-learning and double DQN
  - Policy gradient motivation
  - Policy gradient theorem

- Announcements
  - Assignment 2
    - Demo straw poll
  - Paper presentation (10% weight)
    - To be held week of Nov 4 and/or Nov 11
      - Only for those crediting
    - List of (suggested) papers
    - Paper selection deadline this **Saturday, 26 Oct, 11.55 pm**
    - If no selection, randomly assigned