RL: Deep

Double DQN

Marius Lindauer

Leibniz Universität Hannover

Winter Term 2021

Recall: Double Q-Learning

- Initialization:
  - $Q_1(s, a)$ and $Q_2(s, a)$ for $\forall s \in S, a \in A$
  - $t = 0$
  - initial state $s_t = s_0$
- Loop
  - Select a_t using ϵ-greedy $\pi(s) \in \arg\max_{a \in A} Q_1(s_t, a) + Q_2(s_t, a)$
  - Observe (r_t, s_{t+1})
  - With 50-50 probability either
    1. $Q_1(s_t, a_t) \leftarrow Q_1(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q_2(s_{t+1}, a) - Q_1(s_t, a_t))$
       or
    2. $Q_2(s_t, a_t) \leftarrow Q_2(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q_1(s_{t+1}, a) - Q_2(s_t, a_t))$
  - $t = t + 1$

Recall: Double Q-Learning

- Initialization:
  - $Q_1(s,a)$ and $Q_2(s,a)$ for $\forall s \in S, a \in A$
  - $t = 0$
  - initial state $s_t = s_0$
- Loop
  - Select a_t using ϵ-greedy $\pi(s) \in \arg\max_{a \in A} Q_1(s_t, a) + Q_2(s_t, a)$
  - Observe (r_t, s_{t+1})
  - With 50-50 probability either
    1. $Q_1(s_t, a_t) \leftarrow Q_1(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q_2(s_{t+1}, a) - Q_1(s_t, a_t))$
       or
    2. $Q_2(s_t, a_t) \leftarrow Q_2(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q_1(s_{t+1}, a) - Q_2(s_t, a_t))$
  - $t = t + 1$

$\rightsquigarrow$ reduces maximization bias

Double DQN Hasselt et al. 2015

- Extend this idea to DQN
- Current Q-network $\vec{w}$ is used to select actions
- Older Q-network $\vec{w}^-$ is used to evaluate actions
- TD-error:

$$r + \gamma \overbrace{\hat{Q}(s', \underbrace{\arg\max_{a' \in A} \hat{Q}(s', a'; \vec{w})}_{\text{Action selection: } \vec{w}}; \vec{w}^-)}^{\text{Action evaluation: } \vec{w}^-} - Q(s, a; \vec{w})$$

Double DQN Hasselt et al. 2015

- Extend this idea to DQN
- Current Q-network $\vec{w}$ is used to select actions
- Older Q-network $\vec{w}^-$ is used to evaluate actions
- TD-error:

$$r + \gamma \overbrace{\hat{Q}(s', \underbrace{\arg\max_{a' \in A} \hat{Q}(s', a'; \vec{w})}_{\text{Action selection: } \vec{w}}; \vec{w}^-)}^{\text{Action evaluation: } \vec{w}^-} - Q(s, a; \vec{w})$$

- Allows flipping between both weight sets frequently
  - alternatively, Polyak averaging:

$$w' \leftarrow \tau w + (1 - \tau) w'$$

  - τ is fairly small, e.g, 0.01
- Faster propagation of information compared to original DQN

Clipped Double DQN Fujimoto et al. 2018

- Extend this idea to DQN
- Again having two independent Q-networks with $\vec{w}_1$ and $\vec{w}_2$
- Take minimum action value for successor state
- TD-error:

$$r + \gamma \min_{i=\{1,2\}} Q(s', \arg\max_{a' \in A} Q(s', a'; \vec{w}); \vec{w}_i) - Q(s, a; \vec{w})$$

  - Less overestimation of Q-values
  - More stable learning targets