Lecture 11: Deep RL (RL Algorithms based on Neural Networks)

Course: Reinforcement Learning Theory
Instructor: Lei Ying
Department of EECS
University of Michigan, Ann Arbor

Deep Q-learning

- Motivation: Q-learning overestimates action values under certain conditions, because of the maximization step over estimated action values.
- Question: How bad is overestimation?

Deep Q-learning

- Thrun and Schwartz (1993):

Assume under function approximation,

$$\tilde{Q}(s,a) = Q(s,a) + Y_{s,a}$$

where $E[Y_{s,a}] = 0$ (zero mean).

Q-learning: with transition (s, a, s'),

$$Q(s,a) \leftarrow r(s,a) + \alpha \max_{\hat{a}} Q(s', \hat{a})$$

Deep Q-learning

Error from noise:

$$
\begin{aligned}
Z &= r(s,a) + \alpha \max_{\hat{\alpha}} \tilde{Q}(s', \hat{a}) - (r(s,a) + \alpha \max_{\hat{a}} Q(s', \hat{a})) \\
&= \alpha \left(\max_{\hat{a}} \tilde{Q}(s', \hat{a}) - \max_{\hat{a}} Q(s', \hat{a}) \right) \\
&= -\alpha \left(\max_{\hat{a}} Q(s', \hat{a}) - \max_{\hat{a}} (Q(s', \hat{a}) + Y_{s', \hat{a}}) \right)
\end{aligned}
$$

Claim: $E[Y_{s,a}] = 0 \xRightarrow{\text{often}} E[Z] > 0$

- Consider the case $Q(s', \hat{a}_i) = Q(s', \hat{a}_j) \quad \forall i \neq j$
  i.e. Q-values are the same for all actions at state s'.
- Assume $Y_{s', \hat{a}} \sim$ uniform$[-\epsilon, \epsilon]$ and independent across $\hat{a}$

Deep Q-learning

$$
\begin{aligned}
E[Z] &= \alpha E[-Q(s', \hat{a}_1) + Q(s', \hat{a}_1) + \max_{\hat{a}} Y_{s',\hat{a}}] \\
&= \alpha E[\max_{\hat{a}} Y_{s',\hat{a}}]
\end{aligned}
$$

Let n be the number of actions,

$$
= \alpha \int_{-\infty}^{\infty} xnf(x) \Big(\underbrace{\int_{-\infty}^{x} f(z)dz}_{P(\leq x)} \Big)^{n-1} dx
$$

$$
= \alpha n \int_{-\epsilon}^{\epsilon} x \frac{1}{2\epsilon} (\frac{1}{2} + \frac{x}{2\epsilon})^{n-1} dx
$$

Deep Q-learning

- Define $y = \frac{1}{2} + \frac{x}{2\epsilon}$, we have

$$\begin{aligned} E[Err] &= an \int_0^1 (2\epsilon y - \epsilon) y^{n-1} dy \\ &= \alpha n \epsilon \int_0^1 2y^n - y^{n-1} dy \\ &= \alpha n \epsilon \left(\frac{2}{n+1} - \frac{1}{n}\right) \\ &= \alpha \epsilon \frac{n-1}{n+1} \end{aligned}$$

- Remark: This is the worst case. The error is smaller if $Q(s', \hat{a}_1) \neq Q(s', \hat{a}_2)$

Deep Q-learning

- Lower bound (van Hasselt, Guez, Silver (2015)): again, consider $Q(s', \hat{a}_1) = Q(s', \hat{a}_2) \quad \forall \hat{a}_1 \neq \hat{a}_2$
- Define $Q(s', \hat{a}) = V_*(s') \quad \forall \hat{a}$, and assume

$$\sum_{\hat{a}} (\tilde{Q}(s', \hat{a}) - V_*(s')) = 0 \text{ (unbiased as a whole)}$$

and $\frac{1}{n} \sum_{\hat{a}} (\tilde{Q}(s', \hat{a}) - V_*(s'))^2 = c$

Claim:

$$\max_{\hat{a}} \tilde{Q}(s', \hat{a}) \geq V_*(s') + \sqrt{\frac{c}{n-1}}$$

Deep Q-learning

Proof (by contradiction):

- Define $\epsilon_{\hat{a}} = \tilde{Q}(s', \hat{a}) - V_*(s')$
  Suppose there exist $\{\epsilon_{\hat{a}}\}$ such that $\epsilon_{\hat{a}} < \sqrt{\frac{c}{n-1}} \quad \forall \hat{a}$
- Let $\{\epsilon_i^+\}_{i=1,\dots,m}$ be positive ϵ, and $\{\epsilon_j^-\}_{j=1,\dots,n-m}$ be negative ϵ.

$$
\begin{aligned}
\sum_{j=1}^{n-m} |\epsilon_j^-| &= \sum_{i=1}^{m} \epsilon_i^+ \text{ (because } \sum_{i=1}^{m} \epsilon_i = 0) \\
&\leq m \max_{i=1,\dots,m} \epsilon_i^+ < m\sqrt{\frac{c}{n-1}} \\
\implies \sum_{j=1}^{n-m} (\epsilon_j^-)^2 &\leq \left(\sum_{j=1}^{n-m} |\epsilon_j^-| \right)^2 \\
&< m^2 \frac{c}{n-1}
\end{aligned}
$$

Deep Q-learning

$$\begin{aligned}
\implies \sum_{i=1}^{n} \epsilon_i^2 &= \sum_{i=1}^{m} (\epsilon_i^+)^2 + \sum_{j=1}^{n-m} (\epsilon_j^-)^2 \\
&< m\frac{c}{n-1} + m^2\frac{c}{n-1} \\
&= \frac{m(m+1)}{n-1}c \\
&\leq nc \text{ (because } m \leq n-1\text{).}
\end{aligned}$$

In other words,

$$\sum_{i=1}^{n} \epsilon_i^2 < nc \text{ (because } m \leq n-1\text{). Contradiction}$$

Deep Q-learning

- Overestimation leads to the failure of Q-learning
- Example (Thrun and Schwartz (1993)):

1. A set of goal states
2. $r_s = \begin{cases} 1 & s \text{ is a goal state} \\ 0 & \text{otherwise} \end{cases}$
3. state-transition is deterministic

- Suppose $\langle s_i, a_i \rangle$ $(i \in \{0, \ldots, L\})$ is an optimal state-action sequence.
- Then, necessary condition: $Q(s_i, a_i) < Q(s_{i+1}, a_{i+1})$ when $\alpha < 1$ (because the reward is only received at the end).

Deep Q-learning

- Define $c = \epsilon\frac{n-1}{n+1}$ with the same uniform noise.
- Suppose the algorithm overestimates Q values by αc, $c = \epsilon\frac{n-1}{n+1}$ every step. Then,

$$\begin{aligned} q_L &= 1 \\ q_{L-1} &= \alpha + \alpha c = \alpha(1+c) \\ q_{L-2} &= \underbrace{\alpha q_{L-1}}_{\text{true estimation}} + \underbrace{\alpha c}_{\text{error}} \\ &= \alpha(q_{L-1} + c) \\ &= \alpha(\alpha + \alpha c + c) \\ &= \alpha^2 + \alpha^2 c + \alpha c \end{aligned}$$

$$q_i = \alpha^{L-i} + \sum_{k=1}^{L-i} \alpha^k c$$

Deep Q-learning

- Necessary condition:

$$
\begin{aligned}
0 \geq q_i - q_{i+1} &= \alpha^{L-i} + \sum_{k=1}^{L-i} \alpha^k c - \alpha^{L-i-1} - \sum_{k=1}^{L-i-1} \alpha^k c \\
&= \alpha^{L-i-1}(\alpha - 1) + \alpha^{L-i} c
\end{aligned}
$$

$$
0 \geq \alpha - 1 + \alpha c \iff \alpha \leq \frac{1}{1+c}
$$

Otherwise, Q-learning may fail.

Q-learning methods overview

- Q-learning (Watkin '89) tabular method.

$$Q_{k+1}(i,u) = Q_k(i,u) + \beta_k(r(i,u) + \alpha \max_v Q_k(j,v) - Q_k(i,u))$$

- Q-learning (function approximation)

$$\begin{aligned}\theta_{k+1} = &\theta_k + \beta_k \left(r(s_t, a_t) + \alpha \max_u Q(s_{t+1}, u; \theta_t) - Q(s_t, a_t; \theta_t) \right) \times \\ &\nabla_{\theta_t} Q(s_t, a_t; \theta_t)\end{aligned}$$

Q-learning methods overview

- Double Q-learning (van Hasselt 2010)
  Maintain two Q-functions, denoted by their parameters θ and θ',

$$\theta_{k+1} = \theta_k + \beta_k \delta_k \nabla_{\theta_k} Q(s_k, a_k; \theta_k),$$

where

$$\delta_k = r(s_k, a_k) + \alpha Q(s_{k+1}, \arg\max_u Q(s_{k+1}, u; \theta_k); \theta'_k) - Q(s_t, a_t; \theta_t),$$

and

$$\theta'_{k+1} = \theta'_k + \beta_k \delta'_k \nabla_{\theta_k} Q(s_k, a_k; \theta'_k)$$

where

$$\delta'_k = r(s_k, a_k) + \alpha Q(s_{k+1}, \arg\max_u Q(s_{k+1}, u; \theta'_k); \theta_k) - Q(s_t, a_t; \theta'_t).$$

The double estimator to estimate $\max_i \mathbb{E}[X_i]$

(Van Hasselt '2010)
S_i: set of samples from X_i
Single estimator: $\max_i \mathbb{E}[X_i] \approx \max_i \frac{1}{|S_i|} \sum_{s \in S_i} s$
Double estimator:
Split S_i into S_i^A and S_i^B such that $S_i^A \cup S_i^B$ and $S_i^A \cap S_i^B = \varnothing$
Define

$$\mu_i^A = \frac{1}{|S_i|} \sum_{s \in S_i^A} s, \qquad \mu_i^B = \frac{1}{|S_i|} \sum_{s \in S_i^B} s$$

- pick i^* such that $\mu_{i^*}^A = \max_i \mu_i^A$
- approximate $\max_i \mathbb{E}[X_i] = \mu_{i^*}^B$

Remark: Separate the maximizer and evaluation.
Claim: Double estimator is an underestimator.

The double estimator to estimate $\max_i \mathbb{E}[X_i]$

Proof.

If a^* is the maximizer,

$$\mathbb{E}[\mu^B_{a^*}] = \mathbb{E}[X_{a^*}] = \max_i \mathbb{E}[X_i].$$

If not, $\mathbb{E}[\mu^B_{a^*}] < \max_i \mathbb{E}[X_i]$.

$$\begin{aligned}
\Rightarrow \mathbb{E}[\mu^B_{a^*}] =& \mathbb{P}(a^* \in M)\mathbb{E}[\mu^B_{a^*}|a^* \in M] + \mathbb{P}(a^* \notin M)\mathbb{E}[\mu^B_{a^*}|a^* \notin M] && (1)\\
<& \mathbb{P}(a^* \in M) \max_i \mathbb{E}[X_i] + \mathbb{P}(a^* \notin M) \max_i \mathbb{E}[X_i] && (2)\\
=& \max_i \mathbb{E}[X_i] && (3)
\end{aligned}$$

where M is the set of maximizers.
If $\{X_i\}$ are i.i.d., then $\mathbb{E}[\mu^B_{a^*}] = \max_i \mathbb{E}[X_i]$.

□

Q-learning methods overview

- Deep Q-learning (DQN) (Mnih et al. 2015)
  Maintain a target network θ^-,
  $\theta_k^- \leftarrow \theta_k$ every τ steps.

$$Q_{k+1} = \theta_k + \beta_t(Y_t - Q(s_k, a_t; \theta_k))\nabla_{\theta_k} Q(s_k, a_k; \theta_k)$$

$$Y_k = r(s_k, a_k) + \alpha \max_u Q(s_{k+1}, u; Q_k^-)$$

target network

Action selection and evaluation based on the target network.

Q-learning methods overview

- Double DQN (van Hasselt, Guez, Silver 2015)

$$Y_k = r(s_k, a_k) + \alpha Q(s_{k+1}, \arg\max_u Q(s_{k+1}, u; Q_k); \theta_k^-)$$

action selection based on online networks

evaluation based on target network

Q-learning methods overview

- Clipped Double-Q Learning (Fujimoto, van Hoof, David Meger 2018)
  Maintain two Q-functions, denoted by their parameters θ and θ',

$$\theta_{k+1} = \theta_k + \beta_k(Y_k - Q(s_t, a_t; \theta_t))\nabla_{\theta_k} Q(s_k, a_k; \theta_k)$$
$$\theta'_{k+1} = \theta'_k + \beta_k(Y_k - Q(s_t, a_t; \theta'_t))\nabla_{\theta_k} Q(s_k, a_k; \theta'_k),$$

  where

$$Y_k = r(s_k, a_k) + \alpha \min\left\{\max_u Q(s_k + 1, u; \theta_k), \max_u Q(s_k + 1, u; \theta'_k)\right\}$$

References

- Sebastian Thrun and A. Schwartz, *Issues in Using Function Approximation for Reinforcement Learning*, Proceedings of the Connectionist Models Summer School, June, 1993.
- H Hasselt, *Double Q-learning*, NeurIPS, 2010.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, *Playing atari with deep reinforcement learning,* arXi, 2013
- H. van Hasselt. , A. Guez, D. Silver, *Deep Reinforcement Learning with Double Q-Learning*, AAAI. 2016.
- Scott Fujimoto, Herke van Hoof, and David Meger, *Addressing Function Approximation Error in Actor-Critic Methods*, ICML, 2018.

Acknowledgements: I would like to thank Alex Zhao for helping prepare the slides, and Honghao Wei and Zixian Yang for correcting typos/mistakes.