RL: model을 모를때 MDP를 푸는 algo

DRL: Deep neural net + RL.

Lecture H1. DRL - Value-based agent 1

Sim, Min Kyu, Ph.D., mksim@seoultech.ac.kr

SEOULTECH 서울과학기술대학교 데이터사이언스학과

1. I. Deep Reinforcement Learning (DRL)
2. II. Value-based agent
3. III. Deep Q-learning (DQN)

I. Deep Reinforcement Learning (DRL)

Agent's space to learn

- RL agent must explore its state and action space. The space can be

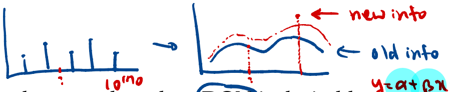
1. Discrete, finite and small set.
   - Skiier's problem has only 8 states and 2 actions.
2. Discrete, finite, but large set.
   - Chess has ~~10^47~~ states. 10^{47}
   - Go has ~~10^170~~ states. 10^{170}
3. Discrete and infinite set.
4. Continuous set (thus infinite).
   - Fine-control problems such as pendulum problem
   - Require policy-based approach. v
   - May discretize it into discrete set.

$\{0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1\}$

Suitable approaches

$q(s.a) \leftarrow q(s.a) + \alpha(Q\text{-}tgt - q(s.a))$

- For 1, $\boxed{\text{tabular approach}}$ is possible. For 2, 3, and 4, states are too many.
  - The storage space itself is burden, if the state has size of $\cancel{10^{170}}$. 10^{180}
  - In tabular approach, (s, a, r, s') updates only a single state's value. Thus, processing time can be inhibitive.
  - **For large or continuous space, functional approximation approach is necessary to search for all spaces.**



- For large space,
  - Part H If the space is discrete, value-based approach such as DQN is desirable.
  - Part I. If the space is continuous, then policy-based approach is required.

$y = a + \beta x + \gamma x^2 + \omega x^3 + \dots + x^8$

- Generalization plays an important role in the acquisition of intelligence by humans and machines
  - Remind that the extension from an action to a policy is a form of **generalization**.
  - Tabular form to functional form has also an implication of **generalization**.

II. Value-based agent

Connecting deep learning with value functions.

- Suppose the a function is parameterized with θ. Then, an approximation function's (e.g. neural net) loss function is typically written as

$$L(\theta) = (\underline{y_{true}} - \underline{y_\theta})^2,$$

θ로 parameterized 된 함수의 추정치.

  where y_θ is an estimated response using the function parameterized by θ.
- If the function approximates state-value function $V(s)$, then

$$L(\theta) = \left(V_{true}(s) - V_\theta(s)\right)^2$$

w. θ.

- If the function approximates state-value function $q(s, a)$, then

action

$$\boxed{L(\theta) = \left(q_{true}(s, a) - q_\theta(s, a)\right)^2}$$

$\min_\theta L(\theta)$

- Find something missing here? How to aggregate the argument s and a so that $L(\theta)$ is a single value?

- Possible approach is to aggregate over all s.
  - Is this possible? Not realistic in current simulated approach because we have to create a stochastic path that goes through all states and only once for each state.
  - Is this desirable? Not really, we should want to focus more on frequent states or frequent trajectory.
- If you agree with the above two points, then see the following alternative.

$$L(\theta) = \mathbb{E}_{\pi}\left[\left(V_{true}(s) - V_{\theta}(s)\right)^2\right]$$

or

$$L(\theta) = \mathbb{E}_{\pi}\left[\left(q_{true}(s,a) - q_{\theta}(s,a)\right)^2\right]$$

- In this approach, the update goes through a stochastic path through π.
- During the agent's learning process, the current π is the best at the moment during the learning. Thus, the visited states following π is most important concern. This learning process is natural in a sense of the Einstein's quote, "Learn from yesterday, live for today.' '

Summary

- $L(\theta) = (y_{true} - y_\theta)^2$
- $L(\theta) = \mathbb{E}_\pi \left[(V_{true}(s) - V_\theta(s))^2 \right]$
- $L(\theta) = \mathbb{E}_\pi \left[(q_{true}(s,a) - q_\theta(s,a))^2 \right]$
- $\theta' \leftarrow \theta - \alpha \nabla_\theta L(\theta)$

목적함수.

θ로 parameterized 된 q함수

"L"의 기울은 낮추는 방향으로 update

learning rate

nabla

편미분하는 것.

DQN

'q-net()'

III. Deep Q-learning (DQN)

Main algorithm

- $q(s, a)$
  - $V(s)$ is value, and $\pi(s)$ is policy. Then, what is really $q(s, a)$?
  - Remind that having an accurate $q(s, a)$ leads to π^*. In this sense, $q(s, a)$ can be regarded as *implicit policy*.
  - Thus, $q(s, a)$ is value and also implicit policy.
- In `pol_eval_Q()`, Q-learning updates $q(s, a)$ in the following way.

$$q(s,a) \leftarrow q(s,a) + \alpha(\overbrace{r_t + \gamma max_{a' \in \mathcal{A}}\, q(s', a')}^{\text{Q-tgt}} - q(s,a)), \ \ \forall s, a$$

- In `pol_eval_DQN()`, DQN updates $q(s, a)$ in the following way.

$$q_\theta(s,a) \leftarrow q_\theta(s,a) + \alpha(\underbrace{r_t + \gamma max_{a' \in \mathcal{A}}\, q_\theta(s', a')}_{\text{Deep Q-tgt}} - q_\theta(s,a)), \ \ \forall s, a$$

Experienced replay

- In "Atari 2013' ' paper, the authors introduced the notion of *replay buffer*.
- This is related to what samples should be used for deep learning updates or so-called `model.fit()`. sample.
- *Replay buffer* stores the recent records of N episodes, shuffle them, and select only some small portion of data to update Q-network. Data structure type of `deque` is popularly used.
- This reduces correlation of the data. Is this desirable?
- Is experienced replay desirable for stationary or non-stationary environment?
- Experienced replay works for off-policy algorithm, because the study material data is not only stemmed from current policy.
- V The implementation is not complex.

- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Target network

- Default setting of DQN has a single neural net for $q(s, a)$. This network is used to setting a Q-target (`q_tgt`) and the value of Q-target (`q_tgt`) is used to update the deep Q-netwrowk (`q_net()`).
- This may deteriorate the stability, resulting in a high variance during the learning.
- The idea is to prepare two separate networks for $q(s, a)$. The two networks are repetitively and interchangeably used for prediction and updating.
- The implementation is not complex.

```
"Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop
questioning. - Albert Einstein"
```