Q-Learning의 이론적 배경

> Author : nemo
> 2020-08-31

Contents

지난 세미나에서는 Q-Learning의 구현과 동작에 대한 내용을 다루었다.

DQN에 대한 이론을 설명하기 앞서, Q-Learning의 수렴성에 대한 이해도를 높여보자.

DQN 논문의 Background로 여러 수식들이 던져져 있는데 그 수식들에 대해 잘 알기 위함이다.

Q-Learning의 근간이 되는 Bellman Optimality Equation에서부터 출발해 알아볼 것이다.

---

1. Review

Bellman Optimality Equation에 들어가기 앞서 살짝 복습하자.

1.1. Reward

각 시각마다 환경은 나(Agent)에게 내 행동의 보상이 얼마인지 알려줄 것이다.

현재 행동은 미래에 영향을 주므로, 내가 현재 행동으로 받은 보상을 계산할 때에는 미래에 받은 보상까지 고려해야 한다.

> $R_t = \Sigma_{t'=t}^{T} \gamma^{t'-t} r_{t'}$

γ이란 파라미터가 있다. 이는 미래보상을 감가한 것으로, 0이면 현재보상만 고려한다는 뜻이고, 1이면 미래보상과 현재보상을 동일하게 고려한다는 뜻이다. 보통 0.99 ~ 0.99999 등의 수를 쓴다.

이 γ는 현재보상과 미래보상을 각각 얼마나 중요하게 볼 것인가에 대한 의미도 있으며 이후 Bellman Equation의 수렴성을 증명할 때 중요하게 쓰인다. γ가 1보다 작기 때문에 수렴하는데, 에피소드의 종료가 보장되는 경우에는 1을 사용할 수도 있다.

1.2. Q(s,a)

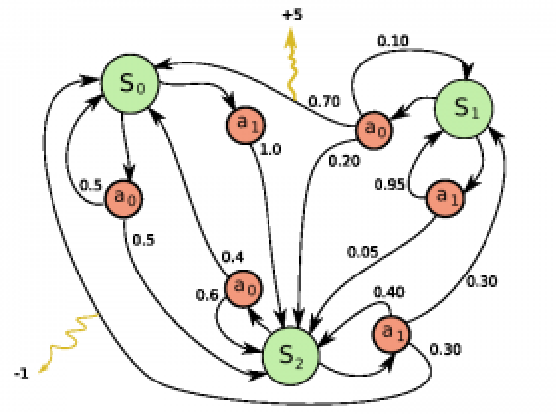
가치 함수의 한 종류에는 상태와 행동을 입력으로 받는 행동 가치 함수(action value function)가 있다.

최적 행동 가치 함수는 행동 가치 함수의 일종으로 ϵ-greedy 기법을 사용한다.

> $Q^*(s,a) = max_{\epsilon} \mathbb{E}_{s' \sim \epsilon}(R_t | s_t = s, a_t = a)$

현재 상태(state)에서 행동(action)을 선택했을 때 얻을 수 있는 보상의 기댓값을 나타내는 함수이다.

그럴 수 있도록 좋은 정책을 찾아 Q를 최대화한 것이 Q^*(Optimal policy)이다.



2. Bellman Optimality Equation & Convergence

이번 절에서는 다음과 같은 내용을 다루어보자.

1) Q와 r을 잘 정리해서점화식 형태의 Bellman Optimality Equation을 얻을 수 있다.

2) Bellman Optimality Equation으로 계속 가치 반복을 수행하면 Q는 수렴한다. (Q-Learning)

2.1. Bellman Optimality Equation

Q에 R을 대입해보면 다음과 같다.

$Q^*(s,a) = max_{\epsilon}\mathbb{E}_{s'\sim\epsilon}(\Sigma_{t'=t}^{T}\gamma^{t'-t}r_{t'}|s_t = s, a_t = a)$

$= max_{a'}\mathbb{E}_{s'\sim a'}(\Sigma_{t'=t}^{T}\gamma^{t'-t}r_{t'}|s_t = s, a_t = a)\ (\because \epsilon \to 0)$

미래 보상에 대한 정보가 필요하다는 점이 너무 복잡하고 구하기 어렵다. 밸만 최적 방정식(Bellman Optimality Equation)은 이를 점화식 형태로 유도한 것이다.

> **Bellman Optimality Equation**
>
> $Q^*(s,a) = r_t + max_{a'}\gamma Q^*(s',a')$
>
> $pf)$
>
> $Q^*(s,a) = max_{a'}\mathbb{E}_{s'\sim a'}(\Sigma_{t'=t}^{T}\gamma^{t'-t}r_{t'}|s_t = s, a_t = a)$
>
> $= max_{a'}\mathbb{E}_{s'\sim a'}(r_t + \Sigma_{t'=t+1}^{T}\gamma^{t'-t}r_{t'}|s_t = s, a_t = a)$
>
> $= max_{a'}\mathbb{E}_{s'\sim a'}(r_t + \gamma\left[\mathbf{\Sigma_{t'=t+1}^{T}\gamma^{t'-(t+1)}r_{t'}}\right]|s_t = s, a_t = a)$
>
> $= \mathbb{E}_{s'\sim a'}(r_t + \gamma max_{a'}Q^*(s',a')|s_t = s, a_t = a)\ (\because \epsilon \to 0)$
>
> $= r_t + max_{a'}\gamma Q^*(s',a')$

좋은 점화식이다.

에피소드를 돌리는 과정에서 우리는 현재의 보상(r)과 변한 환경에 따른 다음 상태(s')를 안다. Q함수(Q_learing에서는 배열, DQN에서는 신경망)에 대입하면 다음 행동(a')에 따른 Q를 추측할 수 있다. 우변의 식을 아니 Q*(s,a)을 바로 다음 상태만 이용해서 업데이트하면 된다.

그런데 이 식은 수렴할까. 여러 상태에 대해 저 식을 계속 돌리다보면, 항상 최적 정책 함수를 구할 수 있을까.

2.2. Value Iteration

밸만 에러에는 비선형 함수인 최댓값 함수가 쓰인다. 간결한 수식으로 해를 계산할 수 없지만 가치 반복, 정책 반복 등의 방식으로 해를 구할 수 있다.

Q-Learning은 가치 반복 기법을 사용한다. 가치 반복이란, 위의 Bellman Optimality Equation을 계속 반복하는 것을 의미한다.

2.3. Convergence

가치 반복을 통해 최적해를 계산할 수 있을까. 가치 반복을 무한히 반복하면 밸만 에러는 수렴할까.

> **Bellman Optimality Equation의 수렴성**
>
> $Q(s,a) \Leftarrow r_t + max_{a'} \gamma Q(s', a')$
>
> > **L-infinity norm**
> >
> > $||\mathbf{x}||_\infty = lim_{t \to \infty} (\Sigma_i |x_i|^p)^{\frac{1}{p}} = max_i x_i$
> >
> > $pf)$
> >
> > $Show\ ||\mathbf{x}||_\infty \le max_i x_i$
> >
> > $||\mathbf{x}||_p = (\Sigma_i |x_i|^p)^{\frac{1}{p}} \le (n\ max_i |x_i|^p)^{\frac{1}{p}}$
> >
> > $Since\ p \to \infty, n\ is\ a\ fixed\ number$
> >
> > $= max_i x_i$
> >
> > $Show\ ||\mathbf{x}||_\infty \ge max_i x_i$
> >
> > $||\mathbf{x}||_p = (\Sigma_i |x_i|^p)^{\frac{1}{p}} \ge (max_i |x_i|^p)^{\frac{1}{p}} = max_i x_i$
> >
> > $Hence, ||\mathbf{x}||_\infty = lim_{t \to \infty} (\Sigma_i |x_i|^p)^{\frac{1}{p}} = max_i x_i \square$
>
> > **Fixed Point**
> >
> > $x = f(x)$
> >
> > **Banach Fixed Point Theorem**
> >
> > $\forall x, \tilde{x} \in Banach\ Space, 0 \le r < 1,$

> $||T(x) - T(\tilde{x})|| \leq r||x - \tilde{x}|| \Rightarrow \exists!\ \alpha\ s.t.\ T(\alpha) = \alpha$
>
> $pf)$
>
> $Show\ T\ is\ Continuous$
>
> $\forall \epsilon > 0$
>
> $||T(x) - T(\tilde{x})|| \leq r||x - \tilde{x}|| < 2\delta < \epsilon$
>
> $Take\ \delta = \frac{\epsilon}{2}$
>
> $Show\ Existence\ of\ \alpha$
>
> $Let\ \{x_n\} := \{x_{n+1} := T(x_n)\}$
>
> $\forall n, m \in \mathbb{N}\ s.t.\ n \geq m$
>
> $let\ n = m + k$
>
> $||x_n - x_m|| = ||x_{m+k} - x_m|| \leq ||x_{m+1} - x_m||(1 + r + \cdots + r^k) \leq ||x_{m+1} - x_m||\frac{1}{1-r} \leq \frac{r^{m-1}}{1-r}||x_1 - x_0||$
>
> $\therefore x_n\ is\ Cauchy\ Sequence$
>
> $Show\ Uniqueness\ of\ \alpha$
>
> $Let\ a, b \in X\ s.t.\ \ f(a) = a, f(b) = b$
>
> $||a - b|| \leq ||T(a) - T(b)|| \leq r||a - b|| \Rightarrow (1 - r)||a - b|| <= 0$
>
> $\therefore a = b$

> $Q(s, a) \Leftarrow r_t + max_{a'}\gamma Q(s', a')$
>
> $pf)$
>
> $Let\ T(Q) = r + \gamma max_{a'}Q, x_1 = Q_a, y_1 = Q_b$
>
> $p \rightarrow \infty$
>
> $||T(x) - T(y)||_p = ||\gamma(x - y)||_p$
>
> $Hence\ t \rightarrow \infty \Rightarrow x_t = y_t = Q$

2.4. Bellman Error

Bellman Optimality Equation을 다시 불러오자.

$Q^*(s, a) = r_t + max_{a'}\gamma Q^*(s', a')$

당연히 좌변이 우변이길 바라겠지만, 우리는 처음에 Q(s, a)를 랜덤으로 초기화했기 때문에 오차가 존재할 것이다.

> $r(s, a) + \gamma max_a Q(s', a) - Q(s, a)$

이를 벨만 에러(Bellman error)라 하는데, 오차니까 최소화하는 방향이 좋으며, 추후 DQN할 때 손실함수로 사용할 수 있다.

2.5. Exploration-Exploitation Tradeoff (Q-Learning)

모험심을 위해 $min\ \epsilon = 0$으로 설정하는 경우는 드물다.

Q-Learning은 방문한 상태에 대해서만 Bellman Operation을 수행하므로, 모험심이 너무 낮으면 잘 수렴하지 못할 수도 있고(bias↑), 모험심이 너무 높으면 오차항이 커진다(variance↑).

Reference

[Bellman Error tronto csc321]

[L-infinity norm proof]

[Banach Fixed Point Theorem Proof]