단단한 강화학습

Ch 01 ~ 02

최우진

2020/03/12

0. 목차

1.1 강화학습

- 강화학습: 주어진 상황에서 어떠한 행동을 취할지를 학습하는 것

  -> 그 행동의 결과는 최대한의 보상(이득)을 가져다 줘야 한다.

  -> 그 보상은 수치적으로 표현

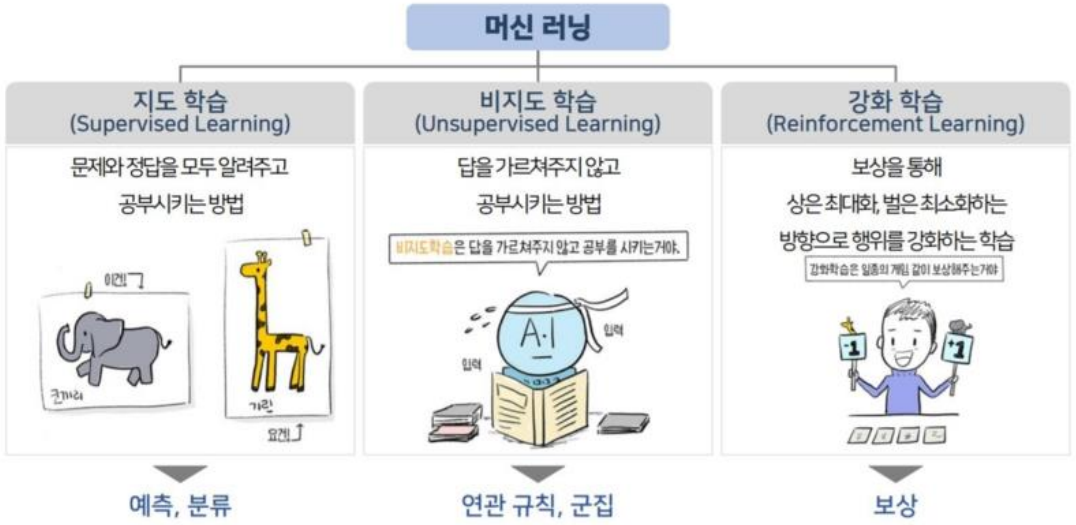
- 따라서, 학습자(agent)는 시행착오를 통해 최대의 보상을 가져다 주는 행동을 찾아야한다.

- 기계학습

  1) 지도학습

  2) 비지도학습

  3) 강화학습



출처: https://steemit.com/kr/@urobotics/5bksow

1.1 강화학습

- 강화학습에서 많은 보상 획득 방법

1) 활용: 학습자는 과거에 보상을 획득하는 데 효과적이었던 행동을 선호해야만 한다.

2) 탐험: 효과적인 행동을 발견하려면 과거에 하지 않았던 행동을 시도해 봐야 한다.

- 강화학습에서 활용과 탐험을 절충하는 점이 어려운 점이다.

1.2 예제

- 숙련된 체스 선수가 말을 옮길 때, 어느 위치로 말을 옮기는 것이 좋을지는 상대의 대응과 그에 대한 재대응을 예상하는 계획, 그리고 즉각적이고 직관적인 판단을 통해 결정된다.
- 새끼 가젤은 태어난 지 몇 분 지나지 않아 어렵게 혼자 힘으로 일어서고, 30분 후에는 시속 30km의 속도로 달릴 수 있다.
- 로봇 청소기는 더 많은 쓰레기를 모으기 위해 새 방을 탐색할지, 아니면 충전 스테이션으로 돌아가야 할지를 결정한다. 이러한 결정은 현재 남아 있는 배터리의 양과 얼마나 쉽고 빠르게 충전 스테이션을 찾을 수 있는지에 대한 과거 경험에 의존한다.

⇒ 모든 예제는 학습자와 그를 둘러싼 주변 환경 사이의 상호작용을 다룬다.

⇒ 주변 환경에는 불확실한 요소들이 있지만, 학습자는 목표를 이루기 위한 방법을 모색한다.

1.3 강화학습의 구성 요소

- 1) 정책(policy)

  -> 학습자가 인지한 주변 환경의 상태에 대해 학습자가 취해야 할 행동을 알려준다.

  ex) 간단한 함수, look up table, 복잡한 계산식, 확률적 선택

- 2) 보상 신호(reward signal) (즉각적인 관점)

  -> 강화학습이 성취해야 할 목표를 정의

  -> 매 시간마다 주변 환경은 학습자에게 보상이라는 수치적 값을 전달

※ 학습자 목표: 장기간에 걸쳐 학습자가 획득하게 되는 보상의 총합을 최대로 만드는 것

1.3 강화학습의 구성 요소

- 3) 가치 함수(value function) (장기적인 관점)

  -> 현 상태의 시작점에서부터 일정시간동안 학습자가 기대할 수 있는 보상의 총량

  -> 행동 결정할 때 주로 사용하지만, 보상보다 계산하기 어렵다.

  -> 보상은 주변환경으로부터 제공되지만, 가치는 학습자의 전 생애주기 동안 학습자가 관찰하는 것으로부터 반복적으로 추정되어야 한다.

- 4) 환경 모델(model) (필수적인 요소는 아님)

  -> 환경이 어떻게 변화해 갈지를 추정하는 도구

  -> 모델은 계획을 위해 사용된다.

※ 계획: 미래의 상황을 실제로 경험하기 전에 가능성만을 고려하여 일련의 행동을 결정하는 방법

1.4 한계와 범위

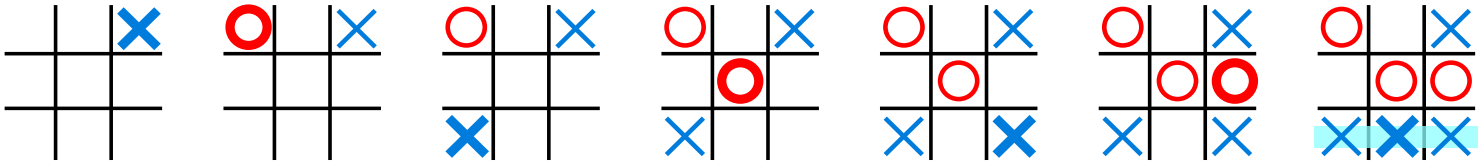
- **상태(state)**
  - 정책과 가치함수의 입력, 모델의 입력 및 출력
  - 학습자가 사용할 수 있는 환경에 대한 모든 정보

- **진화적 방법**
  - 예) genetic algorithm, genetic programming, simulated annealing
  - 동작 방식이 생물학적 진화와 비슷하기 때문에 "진화적 방법"으로 부름.
  - 환경과 오랜 시간 동안 불연속적 시간 간격으로 상호작용하는 **다수의 정적 정책**을 적용한다.
  - 가장 큰 보상을 얻는 정책과 그것의 무작위 변형이 다음 세대의 정책으로 전달되는 일련의 과정을 반복.

1.5 확장된 예제: 틱택토(tic-tac-toe)

- 틱택토: 세개의 행과 세개의 열로 이루어져 있으며, 2명에서 진행하는 게임
- 한 사람은 'X'표시를 다른 한 사람은 'O'표시를 사용한다.
- 둘 중 한 사람이 가로, 세로, 대각선 중 하나의 방향으로 연달아 세 개의 동일한 표시를 하면 승리하고, 두 사람 모두 완성 시키지 못하고 모든 칸을 채우면 무승부.

- 게임 예시



출처:
https://ko.wikipedia.org/wiki/%ED%8B%B1%ED%83%9D%ED%86%A0

1.5 확장된 예제: 틱택토(tic-tac-toe)

- **틱택토를 해결 하기 위한 방법들**

- **1) 전통적인 방법**
  - ‘미니맥스’ 방법(게임이론): 상대방이 특정한 방법으로 게임한다는 것을 가정한 방법
  - 동적프로그래밍(최적화 방법)

  ⇒ 두 방법 모두 사전 정보가 필요하다.

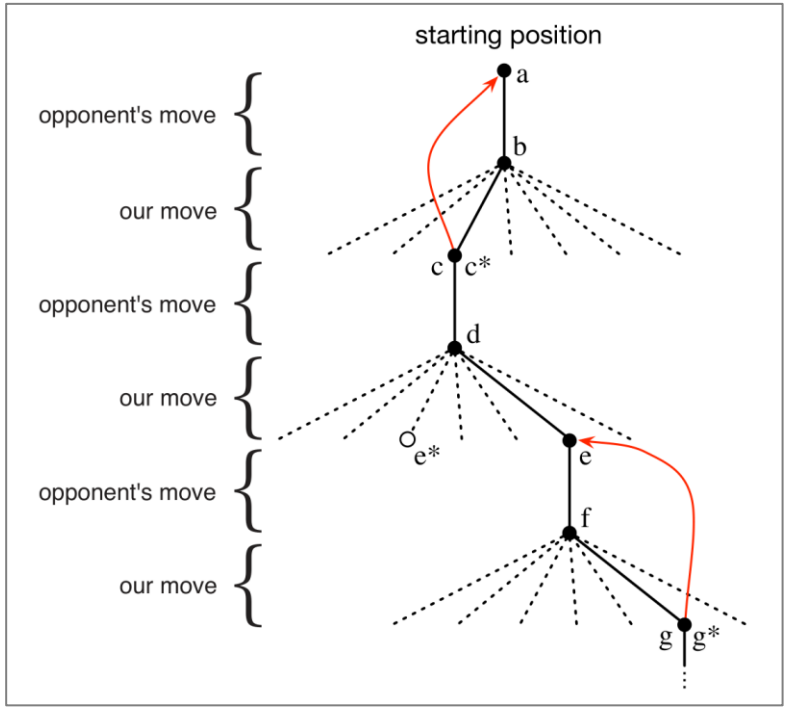
  ⇒ 따라서, 상대방과 많은 게임하여 상대방에 대한 사전 정보를 경험으로 모델을 학습하여 추론할 수 있다.

  ⇒ 학습된 모델로 상대방을 추론하여, 이 값을 동적프로그래밍에 적용하여 최적의 해결책을 제시할 수 있다.

- **2) 진화적 방법**
  - 게임 참여자는 가능한 정책들을 직접 탐색하여 상대방을 이길 확률이 가장 높은 정책을 찾으려고 한다.
  - 정책: 게임 참여자에게 게임의 모든 상황 (즉, 게임판 위에 모든 ‘X’와 ‘O’의 조합에서 어떠한 선택을 해야 할지 알려주는 규칙)

1.5 확장된 예제: 틱택토(tic-tac-toe)

- **3) 가치 함수**
  - 게임에서 나타날 수 있는 모든 상태를 가진 숫자표(가치함수, $V(S)$) 생성(초기값: 0.5)
  - 상대방과 여러 번 게임을 진행
  - S_t: 탐욕스러운 선택 이전의 상태, S_{t+1}: 이후 상태
  - α: 시간 간격 파라미터(step-size parameter)
  - $V(S_t) \leftarrow V(S_t) + \alpha[V(S_{t+1}) - V(S_t)]$
  - α는 시간이 지남에 따라 적절히 줄어든다
  - 따라서, 숫자표의 확률 값이 각각의 상태로부터 승리할 확률이 참값으로 수렴한다는 것을 의미



1.5 확장된 예제: 틱택토(tic-tac-toe)

- **진화적 방법 vs 가치 함수**
- **진화적 방법**
  - 정책을 고정한 채로 상대방과 많은 게임을 시도
  - 승리의 빈도수로 해당 정책을 선택의 기준이 된다.
  - 모든 정책의 수정은 많은 게임을 수행한 후에 이뤄지고, 게임의 최종 결과만 사용된다.
    (즉, 게임에 승리하게 되면 모든 행동이 좋은 평가를 받게 된다.)
- **가치 함수**
  - 개별적인 상태들을 평가한다.
  - 진화적 방법과 가치 함수 모두 정책 탐색을 하지만,
    가치 함수를 학습하면 게임 도중에 발생한 정보를 활용한다.

1.5 확장된 예제: 틱택토(tic-tac-toe)

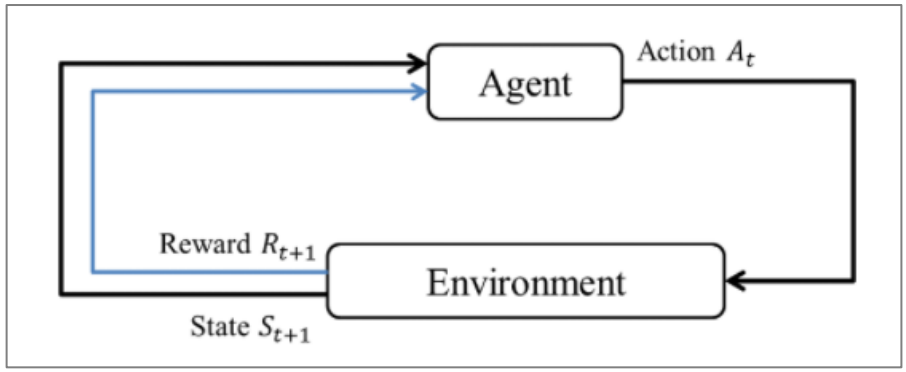
- 강화학습 핵심 특성

1) 주변환경과 상호작용하며 학습하는 것을 강조
   (이번 예제에서는 '상대방 플레이어')

2) 확실한 목표가 있고, 올바른 행동을 위해 학습자가 선택한 행동의 지연된 효과를 고려하는 계획 또는 예측이 필요하다.

1.6 요약

- 다른 기계학습과 달리, 환경과 직접 상호작용하여 학습한다.
- 상태, 행동, 보상의 측면에서 학습자와 환경 사이의 상호작용을 정의하기 위해, 강화학습은 마르코프 결정 과정의 형식적 틀을 사용한다.



출처:
https://untitledtblog.tistory.com/139

- 가치와 가치 함수를 사용하여 정책의 효율적인 탐색이 가능하다.

2.0 다중 선택

강화학습

- 올바른 행동을 알려주는 지시가 아니라, **훈련할 때 행동의 좋고 나쁨을 평가**하는 훈련 정보를 사용한다.
- 따라서, 좋은 행동을 찾기 위해 직접적인 탐색이 필요하다.

피드백

평가적인 피드백

- 예) 강화학습
- 취해진 행동에 대한 평가만 하고, 그 행동이 최상 또는 최악의 행동인지를 알려주지 않는다.

지시적인 피드백

- 예) 지도학습
- 취해진 행동과 상관없이, 취해야 할 올바른 행동을 제시한다.

2.1 다중 선택 문제

- **가정: 하나의 상황에서만 행동을 학습한다.**

- **다중 선택 문제**
  - 1) K개의 서로 다른 옵션이나 행동 중 하나를 반복 선택
  - 2) 선택된 행동에 따라 결정되는 정상(stationary) 확률 분포로부터 얻은 값을 보상으로 제공
    => 그 행동에 대한 '가치'
  - 선택의 목적: 일정 기간동안 얻은 보상의 총량에 대한 기대값 최대화

- **행동의 가치 계산**
  - 임의의 행동 a의 가치 $q_*(a)$는 행동 a가 선택되었을 때 얻는 보상의 기대값
  - $q_*(a) = \mathbb{E}[R_t \mid A_t = a]$, (시간 단계 t에서 선택되는 행동 A_t, 그에 따른 보상 R_t)

2.1 다중 선택 문제

- **하지만, 행동의 가치를 계산할 수 없고 추정만 할 수 있다.**
- **추정한 행동의 가치**
  - $\boldsymbol{Q_t(a)}$: 시간 단계 $\boldsymbol{t}$에서 추정한 행동 $\boldsymbol{a}$의 가치
- **활용**
  - 추정한 가치 $\boldsymbol{Q_t(a)}$가 최대인 행동을 선택
  - 지금 당장 큰 보상 획득
- **탐험**
  - 추정한 가치가 최대인 행동이 아닌 행동을 선택
  - 미래에 큰 보상이 나올 수도 모르는 기대 (불확실성)
- **'활용'과 '탐험'을 적절하게 분배해서 보상을 최대화 해야한다.**

2.2 행동 가치 방법

- **행동 가치 방법: 행동의 가치를 추정하고 추정값을 이용해 행동을 선택하는 방법**

- **간단한 추정 행동의 가치 식**
  - $Q_t(a) = \frac{\text{시각 } t \text{ 이전에 취해지는 행동 } a\text{에 대한 보상의 합}}{\text{시간 } t \text{ 이전에 행동 } a\text{를 취하는 횟수}} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{I}_{A_{i=a}}}{\sum_{i=1}^{t-1} \mathbb{I}_{A_{i=a}}}$
  - $\mathbb{I}_{check}$: $check$가 참이면 1, 거짓이면 0

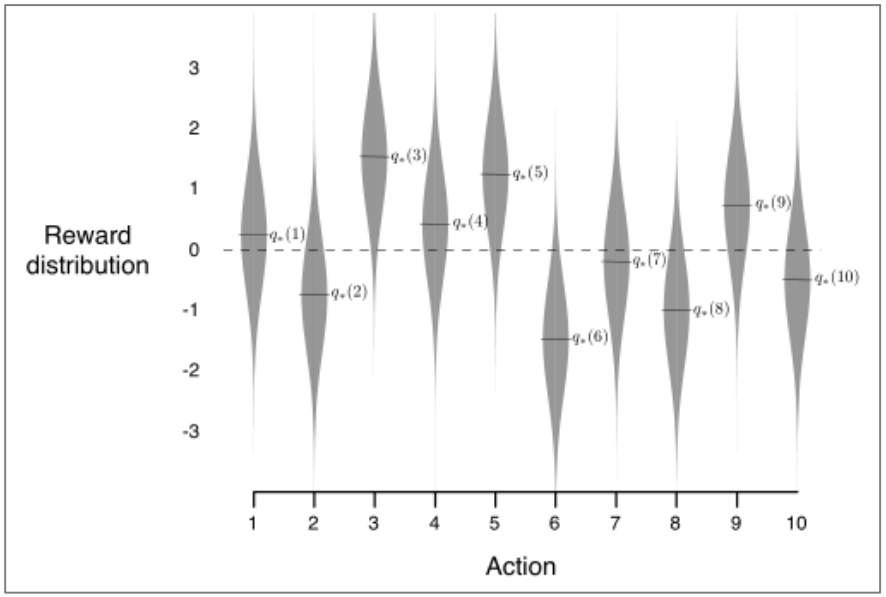
- **간단한 행동 선택 규칙**
  - $A_t = \underset{a}{\operatorname{argmax}}\, Q_t(a)$, ($a$가 여러 개가 나오는 경우, 랜덤으로 행동 선택)

2.2 행동 가치 방법

- **입실론 탐욕적 방법**
  - 탐욕적 방법으로만 행동을 선택하게 되면, 가치가 높은 행동 이외는 선택할 경우가 없다.
  - 이러한 가치가 낮은 행동을 입실론 만큼 선택하여, 탐험을 할 수 있게 해준다.
  - 가치가 낮은 행동을 선택
    - 가치가 낮은 행동들 중에서 랜덤으로 선택
    - 이때는 행동의 가치를 추정과 무관하다.

2.3 10중 선택 테스트

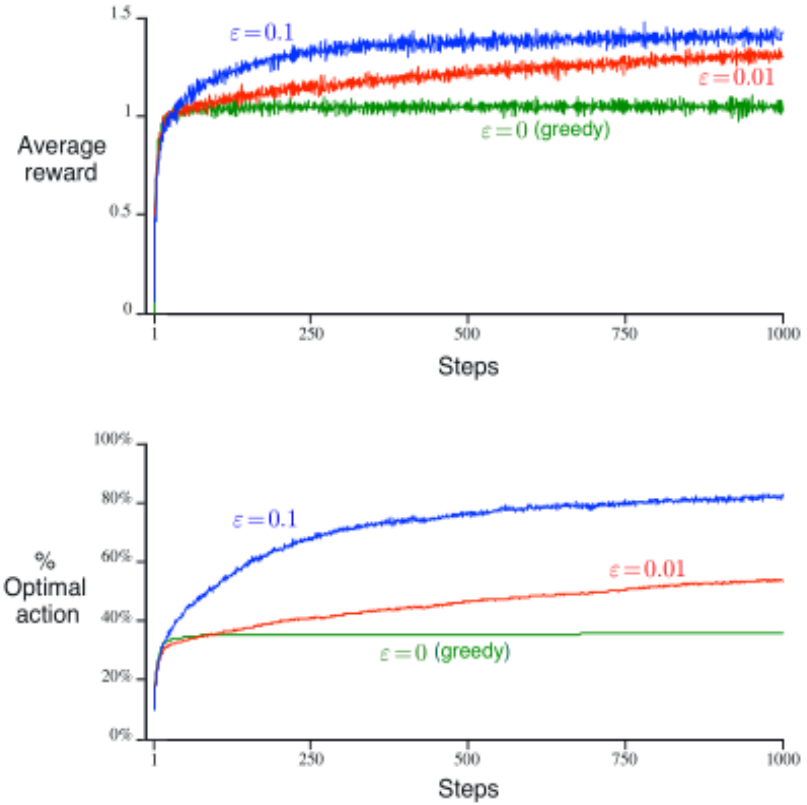
- **탐욕적 방법과 입실론 탐욕적 방법 비교**
- **조건**
  - K = 10, t = 1000, 2000번 수행
  - 시간 단계 $\boldsymbol{t}$에서 행동 $\boldsymbol{A_t}$를 선택할 때,
    실제 보상값 $\boldsymbol{R_t}$가 평균이 $q_*(A_t)$이고 분산이 1인 정규분포에서 선택된다.



2.3 10중 선택 테스트

- 탐욕적 방법과 입실론 탐욕적 방법 비교 결과

- ε = 0.1 인 경우 단위 단계당 보상값이 1.55이지만, ε = 0(greedy)인 경우 1 이다.
- 탐욕적 방법은 대략 30%정도만 최적의 행동을 찾았다.
- 입실론 탐욕적 방법은 탐험을 통해 최적 행동을 식별할 확률을 증가시켰다.
- ε = 0.1인 경우, 최적 행동을 선택한 비율은 대략 90퍼센트 정도이다.
- ε = 0.01인 경우에는 ε = 0.1보다 느리지만, 더 좋은 성능(대략 99퍼센트)을 보일 것이다.



2.3 10중 선택 테스트

- **탐욕적 방법과 입실론 탐욕적 방법 비교**
- **1) 보상의 분산이 더 큰 경우 (1 -> 10)**
  - 보상의 노이즈가 더 많기 때문에, 최적 행동을 찾기 위해 더 많은 탐험이 필요하다.
  - 따라서, 입실론 탐욕적 방법이 탐욕적 방법보다 더 좋다.
- **2) 보상의 분산이 0인 경우**
  - 이 경우는 최적 행동을 바로 찾을 수 있기 때문에 탐험이 필요 없다.
  - 따라서, 탐욕적 방법이 더 좋을 수도 있다.

2.4 점증적 구현

- R_i: 특정 행동이 i번째 선택된 후 받은 보상
- Q_n: 특정 행동이 n - 1번 선택된 이후에 이 행동의 가치 추정값
- $Q_n = \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$,
  이렇게 계산하면 계산량과 메모리가 증가
- 점증적 구현
  - 이전 값을 이후 값을 예측할 때 사용
  - 새로운 추정값 ← 이전 추정값 +
    시간 간격의 크기[목푯값 - 이전 추정값]

$$
\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\Big(R_n + (n-1)Q_n\Big) \\
&= \frac{1}{n}\Big(R_n + nQ_n - Q_n\Big) \\
&= Q_n + \frac{1}{n}\Big[R_n - Q_n\Big],
\end{aligned}
$$

2.4 점증적 구현

- 간단한 다중 선택 알고리즘

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Loop forever:

$$A \leftarrow \begin{cases} \text{argmax}_a\, Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$ $bandit(A)$: 행동 A에 대한 이득, 보상 계산

$N(A) \leftarrow N(A) + 1$

$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\left[R - Q(A)\right]$

Ch 01 ~ 02

감사합니다