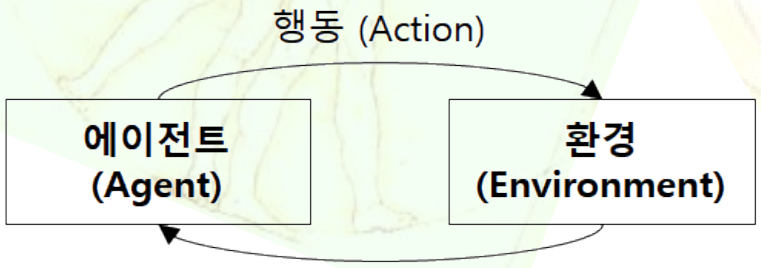
Deep Q-Network

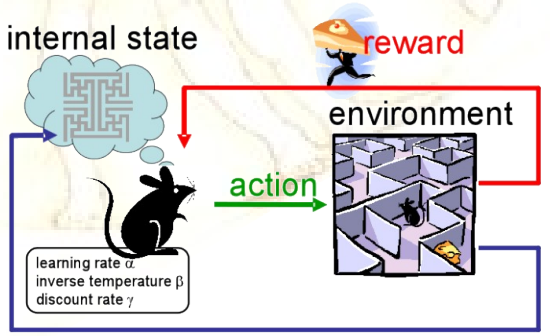
Pattern Recognition & Machine Learning Laboratory
Hyeon-Woo Bae
July 29, 2021

강화 학습

- 강화학습의 특징
  - 에이전트가 주어진 환경에서 행동을 하고 그 행동에 따라 상태가 바뀌고 행동에 대한 보상을 받는 과정의 반복을 통해 학습
  - 음의 보상을 받는 행동은 최대한 피하고 양의 보상을 받는 행동을 강화시켜 보상값을 최대로 얻을 수 있는 학습이 목표
- 용어
  - 상태 (State) : 현재 시점에서의 상황이 어떤지 나타내는 값의 집합
  - 행동 (Action) : 우리가 취할 수 있는 선택지
  - 보상 (Reward) : 에이전트가 행동을 했을 때 따라오는 이득 또는 비용
  - 에이전트 (Agent) : 주변 상태에 따라 어떤 행동을 할 지 판단을 내리는 주체
  - 정책 (Policy) : 에이전트가 판단하는 방식
  - 환경 (Environment) : 주어진 환경



강화학습의 모식도



강화학습의 예시

Q-Learning (1/3)

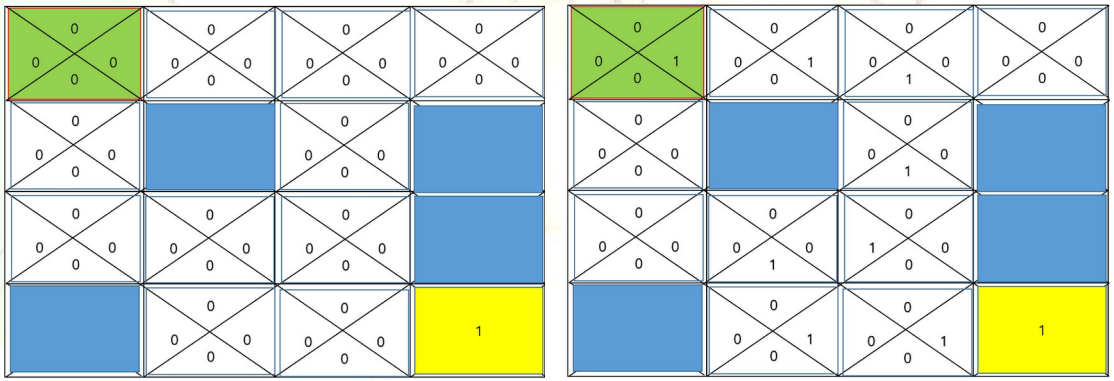
- Q-Learning 의 특징
  - 대표적인 강화학습 알고리즘
  - 특정 상태에서 어떤 결정을 내리는 것이 미래의 보상을 가장 높여줄 것인지에 대한 정책 데이터를 지속적으로 업데이트하는 알고리즘
- 마코프 결정 과정 (Markov decision process)
  - 마코프 상태 : 모든 상태는 오직 그 직전의 상태와 그 상태에서 한 행동에만 의존함
  - 직전 상태보다 더 이전의 상태들에 대해서는 고려하지 않아도 됨
- Discounted future reward
  - : 누적 보상 , : 보상
  - : 할인 요인 (Discounting factor),
  - 미래의 보상은 현재의 보상보다 확실한 값이 아니므로 를 곱해줘서 보정
  - 의 값이 크다면 미래의 보상에 큰 가치를 두는 것
  - 의 값이 작다면 미래의 보상에 작은 가치를 두는 것

Q-Learning (2/3)

- 상태 가치 함수 (state value function)
  - 상태 에서 정책 를 따랐을 때 획득할 총 보상의 기대값
    - : 상태 에서 정책 를 평가하는 가치 함수
- 상태 행동 가치 함수 (state-action value function) (Q-function)
  - 상태 에서 정책 를 따라 행동 를 수행했을 때 획득할 총 보상의 기대값
    - : 상태 에서 행동 를 수행하는 정책 를 평가하는 가치 함수
  - 활용 & 탐험 (Exploit & Exploration)
    - Q-Learning 은 가장 높은 누적 보상을 얻을 수 있는 행동을 취하므로 탐욕 알고리즘
    - Exploration 이 충분하게 이루어지지 못하므로 항상 최선의 결과가 나오지 않을 수도 있음
    - -greedy 알고리즘
      - 인 값을 이용해서 의 확률로는 기존의 탐욕적인 행동을 취하고 의 확률로 랜덤으로 행동을 취하면서 새로운 학습이 가능
    - Decaying -greedy 알고리즘
      - 학습의 초기와 후반부를 구분
      - 학습이 진행될수록 새로운 학습을 하기가 어려워지므로 값을 점점 작게 하여 기존의 행동을 더 많이 취할 수 있도록 하는 알고리즘

Q-Learning (3/3)

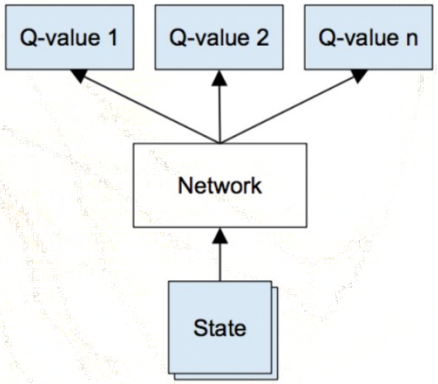
- 상태 행동 가치 함수 (state-action value function) (Cont.)
  - 학습률 (Learning rate)
    - Q-Learning 의 환경이 확률론적인 성질을 가지고 있기 때문에 동일한 매개변수 값과 초기 조건으로 설정이 되어도 다른 출력이 발생될 가능성 존재
    - Learning rate 를 사용하여 기존의 Q 값을 어느 정도 유지하면서 새로운 학습의 결과를 받아들이면 더 좋은 학습이 가능
    - 많은 학습을 반복하게 되면 근사치 가 로 수렴
- Q-Table
  - Q 값이 최대인 방향으로 Agent 가 Action 을 취함
  - 학습을 통해 Table 을 채움
  - 파란색 : 보상없이 종료
  - 초록색 : 시작점
  - 노란색 : 목표 지점



학습 전 상태 학습 후 상태

Q-Network

- **Q-Network 의 등장**
  - Q-Learning 에서 상태와 행동의 수가 커지면 Q-Table 의 크기가 기하급수적으로 커지기 때문에 Q-Table 대신에 Neural network 를 사용하게 됨
- **Q-Network 의 특징**
  - 현재 State 를 입력하면 취할 수 있는 모든 Action 들에 대한 Q 값을 반환하는 형태
  - 학습방법
    - Linear Regression 과 유사함
    - 입력값 : (현재 State 의 정보)
    - 출력값 : (, : 가중치)
    - Cost Function
      - Target 값 :
      - 값을 최소화

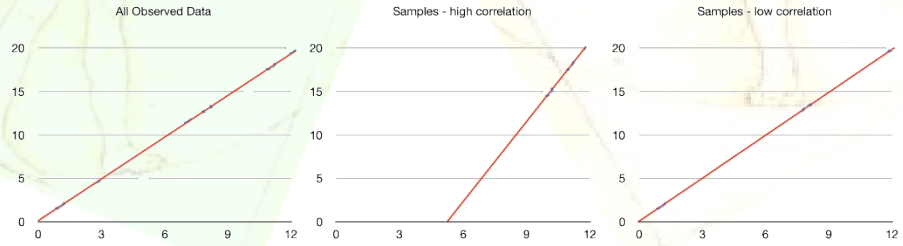


Q-Network 구조

  - Q-Network 알고리즘은 샘플들 사이의 연관성과 Target 값이 불안정한 문제점이 있어 학습이 잘 되지 않음
  - 두 가지 문제점을 해결한 Network 가 Deep Q-Network (DQN)

Deep Q Network (DQN) (1/2)

- **Deep Q Network (DQN) 의 등장**
  - Q-Network 가 학습이 잘 되지 않는 문제점을 Google Deep Mind 팀에서 제대로 학습시킬 수 있는 아이디어를 제시하며 등장
- **Q-Network 의 문제점**
  - 샘플 데이터들 간의 연관성
    - Q-Learning 은 연속된 유사한 State 들을 사용해서 학습이 진행되기 때문에 잘못된 방향으로 학습이 진행될 가능성이 높음
  - 비교해야하는 Target 값의 불안정성
    - ,
    - 현재 State 에서의 Q 값과 다음 State 에서의 Q 값이 같은 값으로 학습
    - (예측값) 이 학습을 통해 업데이트 되면서 (Target 값) 도 변경



샘플 데이터들 간의 연관성 문제

Deep Q Network (DQN) (2/2)

- **Q-Network 문제의 해결방안**
  - **Go deep**
    - Neural Network 의 깊이가 깊어지면 학습 효과가 증가함
  - **Capture and replay**
    - 샘플 데이터 간의 연관성 문제를 해결하는 방법
    - Buffer 에 State, Action, Reward 와 다음 State 에 대한 정보를 저장한 후 무작위로 데이터를 샘플링해서 학습하는 방법
    - 여러 데이터를 학습한 후 각각의 결과들을 수렴해 가장 좋은 하나의 결과를 만들어 해결
  - **Separate networks**
    - Target 값의 불안정성 문제를 해결하는 방법
    - 현재 State 에서 사용되는 Network (가중치 :) 와 다음 State 에서 사용되는 Network (가중치 :) 를 분리한 후 현재 State 의 Network 를 먼저 변화시키면서 학습한 후 다음 State 의 Network 를 현재 State 와 동일한 것으로 바꿔주어 해결