스터디 13주차 정리본

컴퓨터비전 총정리: Q-learning

홍송은

1. 강화학습

- 강화학습(Reinforcement Learning, RL)
  : 인공지능 에이전트가 환경과 상호작용하면서 최적의 행동을 학습하는 알고리즘
- 대표적 알고리즘:
  - Q-learning
    - 가치 기반(Value-based) 접근 방식. Q함수를 업데이트함
  - Policy Gradient
    - 정책 기반(Policy-based) 방식으로, 확률적 정책을 직접 최적화함
  - Actor-Critic
    - 정책 함수와 가치 함수를 동시에 학습
  - PPO (Proximal Policy Optimization)
    - 안정성과 효율성을 개선한 정책 최적화 알고리즘

2. Q-learning의 핵심 개념

> ε-greedy 정책을 이용해 탐험과 활용을 균형 있게 수행하면서, 환경의 명시적 모델 없이도 상태-행동 가치 함수를 업데이트하여 최적 정책을 학습하는 강화학습 기법

- 모델 없는 강화학습(Model-Free Reinforcement Learning) 기법
- 에이전트가 환경의 전이 확률이나 보상 구조를 몰라도 시행착오를 통해 상태-행동 쌍의 장기 누적 보상을 학습(off-policy)
  - 환경을 직접 탐색하며 최적의 행동을 학습하는 방식으로 동작
  - 특정 상태에서 특정 행동을 했을 때 얻을 수 있는 보상을 경험을 통해 예측하고, 이를 활용하여 점진적으로 더 나은 정책을 만들어 나감
- 목표: 최대 기대 보상을 얻는 최적 정책(policy) 찾기
  - Q-learning이 **off-policy**인 이유?
    - Q-learning은 ε-greedy 정책으로 행동(탐험)을 하면서도, 업데이트는 항상 greedy 정책 기준으로 수행함 → 행동 정책 ≠ 업데이트 정책
    - 더 나은 정책으로 학습을 유도할 수 있으며, 여러 다른 행동 데이터(경험)도 학습에 재활용 가능
    - DQN에서도 이 off-policy 구조가 적용되어 replay buffer에 저장된 다양한 과거 행동 데이터를 샘플링하여 업데이트함

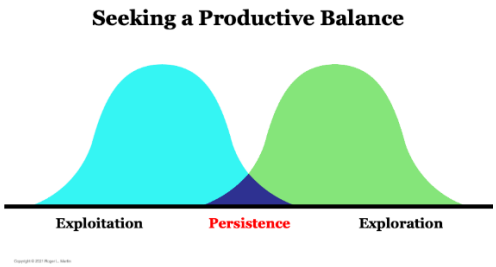
3. Q-learning 구성 요소

- Q함수 $Q(s,a)$
  : 상태 s에서 행동 a를 선택했을 때 얻을 것으로 기대되는 보상의 추정치
- 할인율 $\gamma \in [0,1)$
  : 미래 보상의 현재 가치 반영 정도. 값이 클수록 장기 전략을 중시
- 학습률 $\alpha \in (0,1]$
  : 새 정보가 기존 Q값에 반영되는 비율로, 낮을수록 보수적인 업데이트

4. Q-learning 업데이트 수식

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_t + \gamma \max_{a'} Q\,(s_{t+1},a') - Q(s_t,a_t)\right]$$

5. ε-Greedy 탐험 전략



탐색이 부족하면 최적 행동을 찾지 못할 가능성이 있고, 반대로 탐색이 과하면 최적 정책 수렴이 느려질 수 있음 ➜ ε-Greedy 전략 사용

ε-Greedy: 탐험(exploration)과 활용(exploitation) 사이의 균형을 유지하는 전략

- 확률 ε: 무작위 행동 선택 (탐험)
- 확률 1−ε: 현재 Q값이 가장 큰 행동 선택 (활용)
- Decaying ε-greedy
  시간에 따라 ε을 점차 감소시켜 학습 초반엔 폭넓은 탐색을, 후반엔 안정적인 최적 정책 수렴을 유도

$$\varepsilon_t = max(\varepsilon_m in, \varepsilon_0 \cdot decay^t)$$

  - ε를 decay시키는 이유?
    - 초기에는 다양한 상태와 행동을 탐색하기 위해 큰 ε으로 폭넓은 탐험 수행

- 학습이 진행되며 안정적인 Q값이 형성되면 ε를 줄여서 최선의 행동을 반복적으로 활용
- 탐험(Exploration) → 활용(Exploitation)의 점진적 전환을 유도

○ ε-greedy 정책과 softmax 정책의 차이점?

- ε-greedy는 하나의 최선 행동을 중심으로 무작위 행동을 일정 확률로 추가한다. 간단하지만, 동일한 가치의 행동 간 차이를 구분하지 못함.
- Softmax는 Q값에 따라 행동의 선택 확률을 부드럽게 분포시켜, 가치 차이가 클수록 더 좋은 행동을 선택할 확률이 높아짐
- ε-greedy는 간단한 문제에 유리하고, softmax는 미세한 가치 차이를 구분하는 복잡한 문제에 유리
  ➜ softmax는 policy Gradient 계열 알고리즘, On-policy계열에서 주로 사용

6. Q-learning 학습 과정

① Q 테이블 초기화

○ 각 상태(state)에서 가능한 행동(action)에 대한 예상 보상 값(Q-Value)을 저장하는 테이블

○ 테이블을 점진적으로 업데이트하면서 최적의 행동을 학습

상태(State)	행동 1	행동 2	행동 3	행동 4
S1	0.5	0.2	-0.1	0.0
S2	0.0	0.8	0.3	-0.5
S3	-0.3	0.7	0.5	0.2

② 상태 s_t 관찰

③ ε-greedy 정책으로 행동 a_t 선택

④ 행동 수행 후 보상 r_t, 다음 상태 s_{t+1} 관찰

⑤ Q값 업데이트

⑥ s_{t+1}을 새로운 상태로 설정하고 반복

7. Q-learning 특징

항목	설명
정책 유형	Off-policy (탐색은 ε-greedy, 업데이트는 greedy)
학습 방식	Temporal Difference (TD) 방식
모델 사용	환경 모델 불필요 (Model-Free)
적용 분야	로봇 제어, 게임, 자율 주행, 추천 시스템 등

References

Q-learning 참고 블로그: https://velog.io/@euisuk-chung/%EC%84%A4%EB%AA%85%EC%B6%94%EA%B0%80-Q-Learning-%EA%B0%95%ED%99%94%ED%95%99%EC%8A%B5%EC%9D%98-%ED%95%B5%EC%8B%AC-%EA%B0%9C%EB%85%90%EA%B3%BC-%EC%9D%B4%ED%95%B4

딥러닝 공부 추천 자료:

- 허깅스페이스 – 학습 페이지
  - https://huggingface.co/learn
- 커뮤니티 및 뉴스 레터
  - 딥다이브
    - https://deepdaiv.oopy.io/
  - 모두의 연구소
    - https://moduletter.stibee.com/