2048 Game with Deep Reinforcement Learning

장 용 수
4학년 여름방학
2018-08-14

- 언어는 C (속도 측면에서 기대, 2048은 좋은 UI 가 필요하지 않는 게임)
- 직접 Neural Network 구현
- Input은 현재 state(3*3, 4*4 혹은 5*5), Output은 각각 action(위,아래,왼쪽,오른쪽)의 Q value
- 액션을 취했을때 상태가 바뀌지 않는 경우가 있다. 그럴 땐 보상 0점, 어떻게든 새로운 환경(state)으로 변했을 땐 기본 점수(알파) 부여.
- e-greedy exploration 기법 사용.
- replay memory에는 <state, action, reward, new state> 을 저장. 업데이트 할때는 replay memory에서 몇개의 샘플을 뽑아(mini batch) 다시 FeedForward & BackPropagation -> 각각 샘플마다 gradient descent 구해서 평균내서 업데이트. FeedForward 과정시 new state에서 최고 q value를 뽑아내야 한다. 즉 한번의 샘플에서 두번의 FeedForward 수행을 해야함.
- 구현 과정 : 신경망 설계 -> 2048 Game 설계 -> 강화학습 알고리즘 설계

1. 신경망 설계
   - Input 입력 노드 수 설정 가능해야 한다.(3*3>9, 4*4->16, 5*5->25 등)
   - replay memory로부터 몇개의 샘플을 가져올 건지 batch size를 정할 수 있어야 한다.
   - activation function 설정이 가능해야 한다. (Relu or Sigmoid func)
   - Regularization 기법(L1, L2)
   - FeedForward 함수, BackPropagation 함수(gradient descent 를 구함), update 함수(모아놓은 g.d를 이용해 파라미터 재설정) 구현.
   - 레이어를 크게 하거나 learning rate을 크게 할 경우 발산하는 성질이 있다. 그때 값이 너무 커져버려서 수를 표현하지 못한다(Nan). 발산하는 경우 사전에 막아줘야 한다.
   - 2018/8/20 구현 완료.

2. 2048 Game 설계
   - Reward 규칙 설계해야한다. 큰수가 합쳐질수록 점수 크게(실제 게임에서도 합쳐지는 양을 보상으로 준다)
   - 2048 실패했을 시 보상은 마이너스. 마이너스를 어느정도까지?
   - action 이 선택되었는데 그 action이 움직이지 못하는 상황일 경우, 그냥 보상 0점. 실제 게임에서도 아무런 상태 변화가 없고, 보상도 없다.
   - 움직이지 못하는 action을 학습할 필요가 있을까? 무작위로 호출되지 않는 이상 - 보상을 주지 않으면 같은 action(상태를 변화시키지 못하는)이 나온다. (실험 필요)
   - 행동을 하였을때 새로 나타나는 수는 무작위이다. (항상 사이드 아님). 2또는 4.(4가 나올 확률은 1/8, 2가 나올 확률을 7/8).
   - 새로운 숫자는 액션이 취해지고 난 뒤 빈 공간(0)에 들어온다.
   - +리워드를 주는 경우 : 숫자가 새로 합쳐졌을 때.
   - -리워드를 주는 경우 : 더 이상 새로운 state으로 넘어가지 못할 상황(끝)
   - 0점 리워드를 주는 경우 : state는 변하였으나 아무런 숫자가 합쳐지지 않을 경우. -> 추후 검토해야함. action을 취해도 새로운 state으로 변하지 않는 상황.
   - 초기 게임 보드에는 랜덤으로 두칸에 2또는 4 점수가 부여됨.
   - 멀티스레드를 이용해서 결과를 받아오고 한번에 학습시키는 것도 방법일 듯.
   - 2018/8/21 구현 완료

3. 강화학습 알고리즘 설계
   - e-greedy 에서 얻은 액션을 취해서 reward와 state를 얻는다.
   - e값을 시간의 흐름에 따라서 조절할 수 있어야 한다.

    - replay 메모리에 저장한다.
    - 신경망을 업데이트 한다.

- 잡다한 생각
    - java와 c로 하는것의 차이는 무엇일까..
    - UI : 학습시키기, 시뮬레이션, 학습 저장하기(레이어 정보 필수), 학습 불러오기(처음 일때)