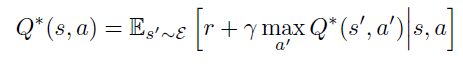
논문 리뷰: Playing Atari with Deep Reinforcement Learning

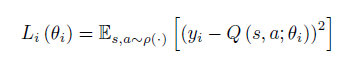
14기 김상현

Vision과 Speech 같은 고차원 입력을 이용해 직접적으로 agent 통제를 학습하는 것은 강화학습의 도전 과제였다. 이 논문 이전 대부분의 강화학습 모형은 수작업 특징에 의존해왔다. 여러 지도 및 비지도 학습 모델들이 발달해 왔지만 강화학습에서 기존의 딥러닝 모델들의 기술들을 사용은 어려웠다. 왜냐하면 첫째로 기존의 모델들은 label이 된 학습데이터를 이용한 반면 강화학습은 reward를 이용해 학습해야 하는데 양질의 reward를 얻는 것이 어려웠다. 또한 기존 모델들의 데이터는 독립인 것을 가정하는 반면에 강화학습에 사용되는 데이터는 높은 상관관계를 갖고, 새로운 행동에 따라 데이터의 분포가 달라진다. 이러한 문제들을 해결하기 위해 이 논문에서는 Q-learning 알고리즘을 사용해서 가중치를 업데이트하고, experience reply mechanism을 사용했다.

Agent가 높은 reward를 받는 방향으로 학습을 진행시켜야 한다. 이때 reward는 즉시 보상과 미래의 보상들의 합을 이용해서 정의한다. 이때 미래의 보상들은 time-step에 따라 할인율(γ:gamma)를 등비수열 꼴로 곱해줘서 구한다. Q\*(s,a)(optimal action-value function)를 상태 s와 행동 a 일 때 정책 pi에 대해 보상 기대값의 최대값으로 한다.



Q\*(s’,a’)을 이용해 Q(s,a)를 표현하는 점화식이 되고, 이는 bellman equation을 따른다. 이를 가치 반복 알고리즘을 이용해 Q\_i가 Q\*로 수렴하게 해서 최적의 값을 찾는다. 하지만 이는 실용적이지 못하다. 해당 논문에서는 Q-network로써 신경망 함수를 approximator로 사용해서 loss function을 정의하고, 확률적 경사 하강법을 사용하여 최적화를 수행한다.

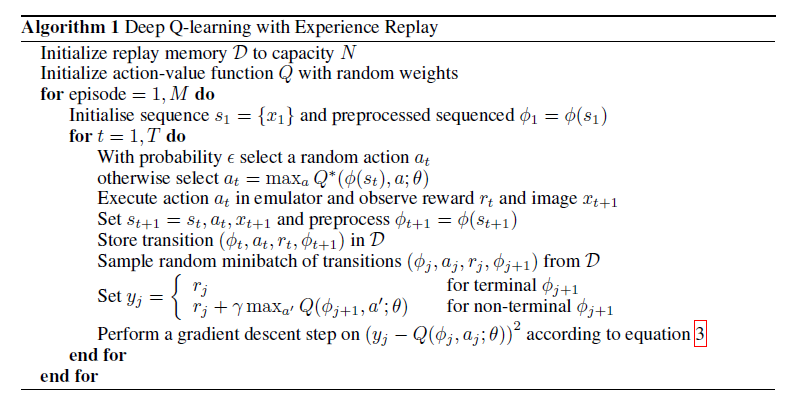




θ를 학습해가면서 Q\*(optimal action-value function)를 찾는다.

Environment의 추정치를 명시적으로 구성하지 않고, emulator environment의 샘플들을 사용하여 직접 강화학습을 해결하는 model-free 알고리즘이다. 또한, epsilon-greedy 알고리즘을 활용해 특정 확률에 따라 랜덤 action을 하거나 greedy strategy를 이용한 action(reward를 가장 크게 하는 action)을 하는 off-policy이다. No-policy를 통해 Exploration(탐사)과 Exploitation(개척)을 같이 수행해 나간다.

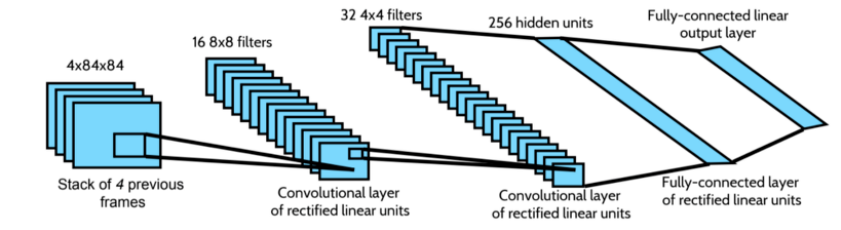
따라서 정리하자면 model-free하고, no-policy한 알고리즘으로 확률적 경사 하강법을 이용해 최적의 θ를 찾아서 Q\*를 찾아나간다.



DQN 알고리즘을 살펴보면 위에 설명한 알고리즘에서 Experience Replay를 추가한 것이다.

먼저 replay memory D를 N의 수용 공간을 갖게 생성하고, Q를 random weights로 생성한다. for문이 두 번 등장하는데 첫번째 loop는 episode에 대한 것이고, 두번째 loop는 각 episode마다 time-step에 대한 것이다. 이렇게 반복을 순회할 때 먼저 각 episode마다 s1을 생성하고, 파이함수를 이용해 전처리를 한다. 이 sequence를 갖고, 두번째 반복을 들어간다. 먼저 epsilon-greedy에 따라 random 또는 greedy action을 정한다. 그 action을 수행한 후 reward와 image를 얻는다. 이들을 sequenced 추가해주고 파이함수를 이용해 전처리 한다. D에 저장시켜준 후 D에서 mini-batch만큼 random sample을 추출한다. 이후 목표지점 도달여부에 따라 y\_j를 결정한다. 위에 나온 경사 하강법을 실행한다. 위 과정을 명시된 loop들 만큼 순회한다. 이 알고리즘을 통해 처음에 생성된 random weights가 optimal weights로 변하게 되면서 Q\*를 찾게 된다. Replay memory라는 저장 공간과 이 공간에서 random sample을 뽑아서 해당 과정을 수행하는 것이 experience replay이다. Experience replay를 통해 parameter의 divergence문제와 local minimum에 빠지는 문제를 해결할 수 있다. 수용 공간을 N에 대해 생각해보면 N이 너무 작을 경우 local minimum 문제 즉 근방의 sequence만을 고려하여 잘못된 학습을 할 것이다. 또한 N이 너무 큰 경우 학습이 오래 걸릴 것이다.

파이함수(전처리 함수)는 210x160pixel를 흑백처리와 down-sampling을 거친 후 cropping을 통해 84x84의 이미지로 변환시킨다. 이후 최근 4 frames를 이용해 input값을 만든다. 이렇게 얻은 84x84x4 input을 이용해 CNN(Convolutional Neural Network)구현하여 Deep-Q-Networks를 이용해 학습시킨다. 해당 모델을 확인해보자



Input을 이용해 합성곱연산층을 거친 후 fc(fully connected)층을 거쳐 각각 action에 따른 Q-value값을 반환한다. Action은 고려하는 game에 따라 달라진다.

‘google deepmind’ 팀은 hyperparameter(batch size, optimizer, epsilon확률 등)을 지정한 후 해당 논문의 알고리즘을 이용해서 뛰어난 강화학습 결과를 얻어냈다. 이전의 ‘Sarsa’(state-action-reward-state-action)와 sarsa에서 augmentation을 이용한 ‘contingency’ 보다 뛰어난 성능을 보였고, 몇몇 게임에서는 사람을 능가하는 성능을 보여준다.

참조

[논문번역]Playing Atari with Deep Reinforcement Learning 논문설명/요약: <https://mangkyu.tistory.com/60>

DQN(Deep Q-Networks) : <https://sumniya.tistory.com/18?category=781573>

Pytorch를 활용한 강화학습/심층강화학습 실전 입문(오가와 유타로 지음)