게임을 통하여 DQN알고리즘 의사 결정 최적화 연구

SONG SHENGHUI, LI LIUYANG

DQN algorithm study through game

요 약

DQN(Dueling Deep Q-Network)은 딥러닝과 강화학습을 결합한 알고리즘으로, Atari 게임에서 인간 수준의 성능을 보인 선구적인 알고리즘이다. DQN은 Deep Q-Network에서 네트워크 구조를 개선하여 더욱 효율적인 학습을 이루게 하였다. 이를 통해 DQN은 강화학습 분야에서 혁신적인 발전을 이루어냈으며, 실제로도 다양한 분야에서 활용되고 있다. DQN은 게임을 통한 학습에 적합한 알고리즘이기 때문에 게임 분야에서 많이 연구되고 있으며, 그 중에서도 Atari 게임을 활용한 연구는 인공지능 분야에서 큰 주목을 받았다.

1. 서 론

강화학습 분야에서 DQN 알고리즘은 매우 유명한 알고리즘이다. DQN 알고리즘은 딥러닝을 기반으로 한 강화학습 알고리즘으로, 이미지 데이터를 입력으로 사용하여 복잡한 환경에서의 강화학습 문제를 해결할 수 있다.

DQN 알고리즘의 핵심은 Q-learning 알고리즘을 딥러닝 기술과 결합하는 것이다. 이를 통해 이전의 Q-learning 알고리즘에서 발생하는 문제들을 개선할 수 있었다. DQN 알고리즘에서는 기존의 Q-learning 알고리즘에서는 전통적으로 사용되었던 Q테이블이 아닌, 딥러닝 신경망을 사용하여 상태와 행동에 대한 Q값을 예측한다.또한, DQN 알고리즘에서는 Experience Replay와 Target Network 기법이 사용되는데, Experience Replay는 강화학습에서 많이 사용되는 Replay Memory를 개선한 방법으로, 학습 데이터를 랜덤하게 뽑아서 경험을 반복해서 사용하게 된다. 이를 통해 학습 효율을 높일 수 있다.

Target Network 기법은 학습 중에 신경망이 불안정하게 업데이트 되는 것을 방지하기 위한 기법이다. 이를 위해 학습 대상 신경망과 별도로 Target Network를 유지하면서 일정한 주기로 업데이트를 수행하게 된다. 이를 통해 학습이 불안정해지는 현상을 줄이고 학습 효율을 높일 수 있다

최근에는 DQN 알고리즘이 게임 내에서도 활용되고 있습니다. 예를 들어, 퍼즐 게임에서 퍼즐 조각들을 정렬하는 데에 DQN 알고리즘이 적용될 수 있습니다. 또한, 게임 내에서 배열을 정렬하는 등의 과정에서 DNQ 알고리즘에 대한 이해가 필요할 수 있습니다.

따라서 이러한 DQN 알고리즘의 활용과 중요성을 더욱 깊이 이해하기 위해, 게임을 통해 DNQ 알고리즘을 연구하는 것은 의미 있는 시도가 될 수 있습니다. 게임을 통해 DNQ 알고리즘을 더욱 쉽게 이해할 수 있고, 게임을 활용하여 DNQ 알고리즘을 최적화하거나 새로운 방식으로 활용하는 아이디어를 얻을 수도 있을 것입니다.

2.기존 연구

2.1 Deep Q-learning와 DQN

Q-learning은 강화학습의 일종으로, 이산적인 상태와 행동의 공간을 갖는 문제를 해결하는 데 사용됩니다. 이 알고리즘은 최적의 행동을 선택하는 정책을 찾기 위해 에이전트가 취할 수 있는 모든 행동의 가치를 추정합니다. Q-learning은 이 추정된 가치를 사용하여 에이전트가 행동을 선택하고, 선택된 행동의 결과로부터 보상을 받아 경험을 쌓습니다. 이 과정을 반복하면서 점차 보상을 최대화하는 행동을 선택할 수 있도록 최적의 가치함수를 찾아나가는 것이 목표입니다. 이를 위해 Q-learning은 Bellman 최적 방정식을 사용하여 가치함수를 업데이트합니다. Q-learning은 강화학습 분야에서 가장 기본적인 알고리즘 중 하나로, 다양한 환경에서 성능이 입증되어 널리 사용되고 있습니다

.

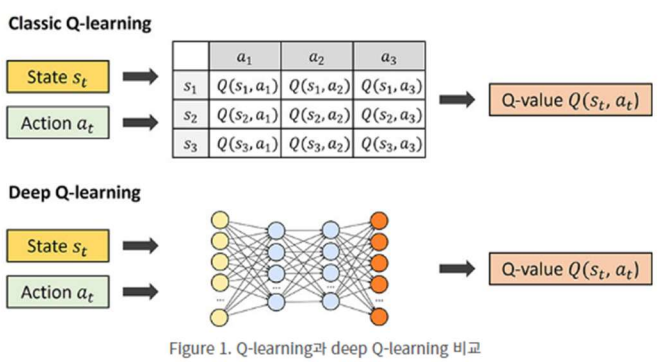


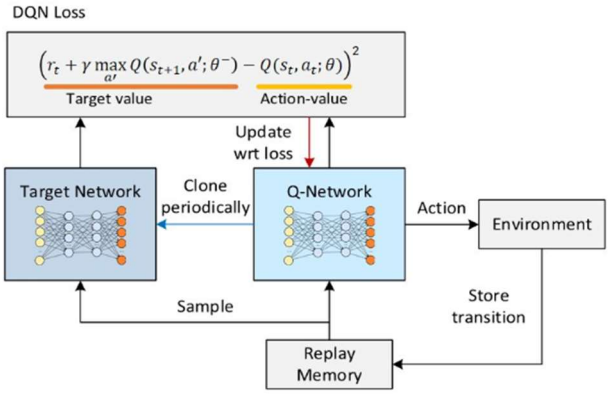
Figure 1. Q-learning과 deep Q-learning 비교

DQN(Deep Q-Network)은 딥러닝과 강화학습을 결합한

알고리즘으로, 딥러닝 모델을 이용해 게임의 상태(State)를 입력으로 받고, 각 가능한 행동(Action)에 대한 Q-value를 출력하는 함수를 학습한다. 이를 이용해 최적의 행동을 선택하고, 이를 통해 게임을 플레이하며 점차 학습한다.

DQN은 Q-learning 알고리즘의 한계를 극복하는데 중요한 역할을 하였다. 기존의 Q-learning 알고리즘은 탐험(exploration)과 이용(exploitation) 간의 균형을 맞추기 어렵다는 단점이 있었는데, 이를 해결하기 위해 DQN은 ε-greedy 기법을 사용한다. 이 기법은 ε 확률로 무작위로 행동을 선택하고, 1-ε 확률로는 모델이 현재 학습한 결과를 바탕으로 가장 높은 Q-value를 가진 행동을 선택하는 방식이다. 이를 통해 무작위 탐험을 하면서 최적의 행동을 찾아나가는 학습이 가능해졌다

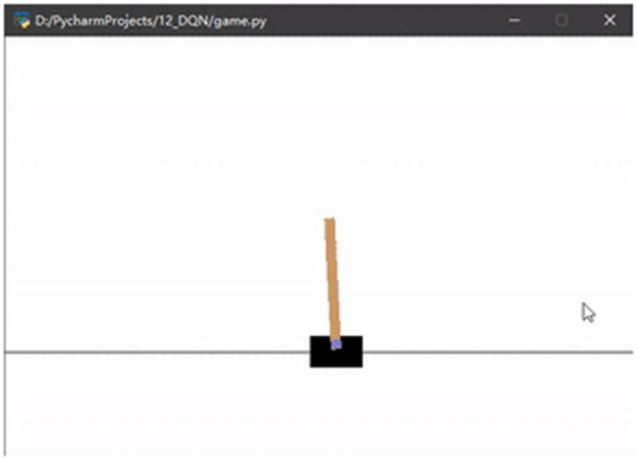
.



2.2DQN으로 게임 노는 원리

DQN은 강화 학습에서 가치 함수를 근사화하는 데 사용됩니다. 이를 위해 DQN은 딥러닝의 신경망을 사용하여 상태와 행동을 입력으로 받고 해당 상태에서의 행동 가치를 출력으로 반환합니다. DQN은 또한 경험 재생(replay memory) 및 고정된 Q-targets를 사용하여 안정적인 학습을 도모합니다. DQN은 비디오 게임에서 상당한 성과를 보여주며, 이를 통해 게임 이외의 여러 응용 프로그램에도 활용될 수 있습니다.

Open Ai라는 인공지능 연구소에서는 인공지능으로 게임을 노는 기록을 볼수있었습니다.레를 들어



이게임은 좌우를 조중하여 박대를 넘어지지 않게 하는 게임 입니다.

3. 문제 정의

'Flappy Bird'는 매우 고전적인 미니 게임입니다. 게임에서 플레이어는 한 마리의 새가 끊임없이 위로 움직이는 배경에서 양쪽의 수도관을 통과하여 가능한 한 높은 점수를 받도록 제어해야 합니다. 하지만 AI의 경우 이 게임에서 높은 점수를 얻기 위해서는 매우 강력한 의사결정 능력이 필요합니다.

따라서 머신러닝 알고리즘을 최적화하고 게임 AI의 훈련 전략을 최적화하여 게임 내 게임 AI의 성능을 빠르게 향상시키는 방법은 현재 해결해야 할 문제 중 하나가 되었습니다.

4. 해결 연구

따이 문제와 관련하여 DQN 알고리즘에 기반한 게임 AI 훈련 전략 최적화 방안이 필요하다.
이 기사는 주로 네트워크 구조의 방향에서 최적화를 연구합니다.

네트워크 구조 최적화는 신경망의 계층 수, 노드 수, 활성화 함수 등을 조정해 모델의 성능을 향상시키는 것을 말한다.DQN에서 네트워크 구조의 최적화는 모델의 성능에 중요한 영향을 미칩니다.네트워크 구조를 최적화하는 방법에는 다음과 같은 단계가 있습니다.

1. 네트워크 깊이 증가: 네트워크 깊이를 증가시켜 모델의 비선형 표현 능력을 증가시켜 모델의 성능을 향상시킬 수 있습니다.그러나 네트워크가 너무 깊으면 구배가 사라지거나 구배가 폭발하는 문제가 발생하기 쉬우므로 적절한 정규화나 특수한 네트워크 구조가 필요하다.

2. 컨볼루션 레이어 사용: 입력 데이터에 이미지 데이터와

같은 공간 구조가 있는 경우 컨볼루션 레이어를 사용하여 특징을 추출할 수 있습니다.컨볼루션 레이어는 모델 매개변수의 수를 줄여 모델의 훈련 속도를 높이는 동시에 입력 데이터의 공간 정보를 유지하고 모델의 성능을 향상시킬 수 있습니다.

'플래피버드'와 같은 게임을 처리할 때 게임 인터페이스의 급격한 변화로 인해 네트워크의 이미지 처리 능력을 향상시키고 게임 상태에 대한 정보를 더 잘 포착할 수 있도록 컨볼루션 레이어와 풀링 레이어를 추가해야 합니다.

컨볼루션 레이어의 역할은 이미지의 가장자리, 선 및 모서리와 같은 특성을 감지하고 더 고급 기능으로 결합할 수 있는 이미지의 특성을 추출하는 것입니다.CNN에서 컨볼루션 레이어는 게임 캡처의 그레이스케일 이미지 또는 RGB 이미지에 적용할 수 있으며 컨볼루션 체크 이미지를 통해 컨볼루션 작업을 수행하여 게임 상태의 특성을 추출할 수 있습니다.

풀링 레이어의 기능은 특성 맵의 크기를 줄이고 모델의 연산 효율을 향상시키는 것입니다.특성 맵을 여러 영역으로 나누고 각 영역을 풀링하고 이러한 영역의 값을 하나의 값으로 결합하여 더 작은 특성 맵을 얻을 수 있습니다.CNN에서 풀링 레이어는 특성 맵의 해상도를 줄이고 네트워크의 계산 속도를 향상시키는 데 사용할 수 있습니다.

컨볼루션 레이어와 풀링 레이어 외에도 CNN 네트워크에 Batch Normalization 레이어와 Dropout 레이어를 추가하여 네트워크의 안정성과 일반화 능력을 향상시킬 수 있습니다.Batch Normalization 계층은 네트워크의 각 계층의 입력 데이터를 표준화하여 입력 데이터의 평균과 분산이 훈련 중에 안정적으로 유지되도록 하여 모델의 수렴을 가속화하고 모델의 견고성을 향상시킬 수 있습니다.Dropout 레이어는 네트워크 과적합을 방지하기 위해 훈련 중에 뉴런의 일부를 무작위로 버릴 수 있습니다.

```
h_conv1 = tf.nn.relu(conv2d(s, W_conv1, 4) + b_conv1)
h_pool1 = max_pool_2x2(h_conv1)

h_conv2 = tf.nn.relu(conv2d(h_pool1, W_conv2, 2) + b_conv2)
#h_pool2 = max_pool_2x2(h_conv2)

h_conv3 = tf.nn.relu(conv2d(h_conv2, W_conv3, 1) + b_conv3)
#h_pool3 = max_pool_2x2(h_conv3)

#h_pool3_flat = tf.reshape(h_pool3, [-1, 256])
h_conv3_flat = tf.reshape(h_conv3, [-1, 1600])

h_fc1 = tf.nn.relu(tf.matmul(h_conv3_flat, W_fc1) + b_fc1)
```

참 고 문 헌

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

[2] Zhang, S., Xu, M., & Yang, J. (2021). Improved Flappy Bird game AI based on deep reinforcement learning. IEEE Access, 9, 13467-13478.