

강화학습을 이용한 대전 격투 게임 AI 구현

신희상^o, 박승보^{*}

^o인하대학교 소프트웨어융합공학과,

^{*}인하대학교 소프트웨어융합공학과

e-mail: hec@inha.edu, molaal@inha.ac.kr

Implementation Fighting Game AI using Reinforcement Learning

Hee-Sang Shin^o, Seung-Bo Park^{*}

^oDept. of Software Convergence Engineering, Inha University,

^{*}Dept. of Software Convergence Engineering, Inha University

● 요약 ●

본 논문에서는 대전 격투 게임에서의 AI 개발을 위한 강화학습 사용 방법을 제안한다. 이 방법은 학습 모델에 상대방의 다양한 패턴을 학습시켜 적은 코드로 효율적인 AI 개발을 할 수 있어 개발 시간을 최소화 할 수 있다. 또한, 이 방법은 복잡한 코드를 추가 또는 제거할 필요 없이 보상과 액션을 조정하여 다양한 종류의 AI를 원하는 만큼 생성할 수 있다는 장점이 있다. 본 논문에서는 Unity 사에서 제공하는 머신러닝 툴인 ML-Agents를 활용하여 강화학습을 통한 대전 격투 게임 AI의 가능성을 보인다.

키워드: 강화학습(Reinforcement Learning), 게임(Game), 인공지능(AI)

I. Introduction

강화학습은 주어진 환경 속에서 에이전트의 행동에 따라 보상이나 벌점을 주도록 해 보상을 최대화하는 방법을 탐색하는 것을 반복하는 기계학습 방법이다. 마르코프 결정 과정(Markov Decision Process)과 Q-러닝을 사용하여 상태와 모델을 학습시키기에 Q-테이블에 저장된 상태에 대한 정보에 의존하는 단점이 있었다. 하지만 DNN(Deep Neural Network)과 강화학습이 결합한 심층 강화학습 방법이 연구되어 여러 문제에서의 해결책을 제시할 수 있게 되었다.

본 논문에서는 대전격투 게임을 플레이할 수 있는 AI를 만드는 것이 목표이다. Unity를 이용하여 대전격투 게임을 모방한 환경을 제작한 뒤, Unity의 툴인 ML-Agents(Machine Learning Agents)를 활용해 강화학습을 진행하였다. 본 연구에서는, ML-Agent에서 제공하는 알고리즘인 PPO(Proximal Policy Optimization) 알고리즘을 사용해 학습을 진행하였다.

II. Preliminaries

1. Related works

2016년에 개최된 Google DeepMind Challenge Match는 인공지능계에 큰 파장을 일으켰다. 사진에서 공을 받아치기에만 급급하던

AI가 4시간의 학습을 거치고 승리하는 가장 효율적인 방법인 한 곳만 뚫는 법을 터득한다. 이 외에도 다른 논문의 결과에 따르면, 49개의 고전 게임 중 29개의 게임이 사람이 수행하는 것보다 점수가 75%가량 높았다. [1]

게임에 있어서 인공지능은 게임플레이에 있어서 난이도, 조작 등에 빠져서는 안 되는 중요한 주제이다.

Google은 DQN이라는 AlphaGo에 사용된 알고리즘을 Atari사의 고전 게임들에 적용하여 좋은 성과를 얻었다. 다만, 이러한 고전 게임들의 조작은 현대의 게임들과는 차이가 있기에 대전격투 게임이라는 특수한 환경에서 학습을 진행해 결과를 관찰하고자 한다.

III. AI Design and Implementation

1. ML-Agents / Environment

ML-Agent는 세 가지 구성 요소로 이루어진 Unity의 플러그인이다 [2]. 그 구성은 다음과 같다.

1) 학습환경 - Agent 개체가 포함된 Unity의 Scene으로 대전격투 게임의 게임환경을 모방하여 구현한다. Agent 개체란 실질적으로 강화학습이 이루어지는 개체이며, Unity 내 GameObject 객체에

할당된다.

2) 파이썬 API - 훈련에 적용될 알고리즘이 위치하며, 학습 환경과는 다르게 파이썬 API는 Unity와는 별도로, 소켓 통신을 통해 유니티의 외부 통신 모듈과 연결한다.

3) 외부 통신 모듈 - 학습 환경과 파이썬 API를 연결한다.



Fig. 1. Block Diagram of ML-Agent

Agent 개체는 다음의 데이터를 관측하여 행동을 수행하고 행동에 따라 적절한 보상을 받는다.

- 상대방과의 거리(Float),
- 상대방의 Action State (Integer) [Stand, Sit, Jump]
- 자신과 상대방의 체력 (Integer)

위의 관측 데이터를 바탕으로 행동을 결정하며 행동의 성공에 따른 보상은 다음과 같다.

Table 1. Agent Reward & Punishment Info

	피격	공격	방어
Punch	-100.0	+200.0	+100.0
Kick	-150.0	+300.0	+100.0

위와 같은 환경에서 게임 내 캐릭터가 상대방으로부터 피격당하거나 공격하여 둘 중 한 캐릭터의 HP가 0이 될 때까지의 한 과정을 Episode라고 할 때, Episode가 끝날 때마다, 해당 Episode에서 얻은 보상을 기존 행위와 비교하여 다음 진행할 Episode가 기존 Episode보다 높은 보상을 얻을 수 있도록 행동순서를 조정한다.

2. Reward Issue

연구 초에 공격 방어 성공 시의 보상을 +300.0으로 설정하고 학습을 진행하였다. 그러나 특수 동작이라고 하여 보상을 지나치게 크게 만든 결과, AI는 좌우 이동 행동만 빠르게 반복하여 공격하지 않고도 방어만 하여 높은 점수를 얻는 데 성공했다. 이는 공격하여 상대방의 체력을 0으로 만들어 승리하는 것 보다, 시간제한 동안 공격을 막아내는 방법이 보상이 더 크기에 문제가 발생하였다. 이 문제는 보상 수치 조정을 통해 해결하였다.

3. Result

본 연구를 통해 제작된 AI는 상대 캐릭터와의 공격/피격 범위를 인지해 스스로 거리를 벌리거나, 좁히면서 공격을 근소한 차이로 피하거나 공격을 막은 뒤 성공시키는 모습을 보였다.

4. Limitations

학습을 지속할수록 학습 환경에서 AI가 상대방의 Action State 변수를 활용하여 상대방의 상태에 따른 다양한 패턴(점프 중 공격, 앉은 채 공격)을 활용해 공격하기보다, 공격 성공률과 보상치가 높은 발차기 위주로 공격하며 과적합 되는 모습을 보였다.

IV. Conclusions

Unity와 ML-Agent를 활용하여 강화학습에 필요한 학습환경을 구현하고, ML-Agent를 활용해 대전격투 게임을 플레이할 수 있는 AI를 만들었다. 기존의 큰 시간과 비용이 발생하는 작업을 강화학습으로 비교적 간단히 대체할 수 있다. 본 연구는 제한된 액션과 상태로 진행되었지만, 일부 과적합의 모습을 띠게 되었다. 반면, 실제 게임 환경은 더욱 복잡한 환경을 지니게 된다. 따라서 더 많은 액션과 상태가 존재할 때 과적합 되지 않고 성공적으로 학습시키는 방법에 관한 연구의 필요성이 있다.

ACKNOWLEDGMENT

이 논문은 2021년도 정보(교육부)의 재원으로 한국연구재단 4단계 두뇌한국(BK)21 사업 대학원 혁신지원을 받아 수행된 연구임.

REFERENCES

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg & D. Hassabis, "ARTIFICIAL INTELLIGENCE Human-level control through deep reinforcement learning," Nature, Vol 518, pp.529-533, 2015.

[2] H.-B. Choi, C.-M. Kim, J.-B. Kim, Y.-H. Han. "Design and Implementation of Reinforcement Learning Environment Using Unity 3D-based ML-Agents Toolkit," Journal of the KIPS, Vol. 26, No. 1, pp.548-551, 2019.