Reinforcement Learning Report

Curriculum Learning for Reinforcement Learning Agents

김건형
214249, 인공지능학부

***Abstract*—본 보고서에서는 Proximal Policy Optimization (PPO)를 활용하여 체스를 학습하는 에이전트를 구현하고, Curriculum Learning 기법을 적용하여 성능을 향상시킨 프로젝트에 대해 설명한다. Curriculum Learning은 학습 과정에서 점진적으로 난이도를 높여가는 방법으로, 에이전트가 더 효과적으로 학습할 수 있도록 돕는다. 본 프로젝트에서는 Random하게 두는 상대, Stockfish 레벨 0 순으로 난이도를 설정하였으며, 이를 통해 에이전트의 학습 성능을 관찰하였다. 또한 초기 학습 단계에서의 안정성을 높이기 위해 Stockfish 레벨 0의 대국 기록을 활용하여 행동 클로닝(Behavior Cloning) 기법을 적용하였다. 실험 결과, Curriculum Learning과 행동 클로닝을 결합한 접근법이 에이전트의 성능 향상에 긍정적인 영향을 미쳤음을 확인할 수 있었다. https://github.com/edenkim9741/Reinforcement-Learning/tree/main/cleanrl**

I. Introduction

체스는 8개의 폰, 2개의 룩, 2개의 나이트, 2개의 비숍, 1개의 퀸, 1개의 킹으로 구성된 체스판에서 두 명의 플레이어가 번갈아 가며 말을 움직여 상대방의 킹을 체크메이트하는 전략 보드 게임이다. 체스는 복잡한 전략과 전술이 요구되며, 다양한 상황에서 최적의 수를 찾는 것이 중요하다. 이러한 특성 때문에 체스는 인공지능 연구에서 중요한 역할을 해왔다. 본 프로젝트에서는 Proximal Policy Optimization (PPO) 알고리즘을 활용하여 체스를 학습하는 에이전트를 구현하고, Curriculum Learning 기법을 적용하여 에이전트의 학습 성능을 향상시키고자 한다.

PPO 알고리즘을 선택한 이유는 안정적이고 효율적인 정책 업데이트를 제공하며, 다양한 환경에서 좋은 성능을 보여주기 때문이다. 비교 대상이 되었던 DQN은 마찬가지로 이산적인 행동 공간을 가지는 체스에 적합하지만 학습이 중단되어도 이어서 학습이 가능한 PPO의 특성이 본 프로젝트에 더 적합하다고 판단되었다.

II. Preliminaries

A. Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) [1] 알고리즘은 On-Policy 계열의 강화학습 알고리즘으로, 정책의 업데이트를 안정적으로 수행하기 위해 설계되었다. PPO는 클리핑(clipping) 기법을 사용하여 정책의 변화가 너무 크지 않도록 제한함으로써, 학습 과정에서 발생할 수 있는 불안정성을 줄인다. 이를 통해 에이전트는 더 효율적으로 최적의 정책을 학습할 수 있다.

현재의 행동이 평균보다 얼마나 더 나은지를 나타내는 어드밴티지 함수 A_t를 사용하여 정책을 업데이트한다. 이 때 TD Error를 활용하여 어드밴티지 함수를 계산한다. PPO의 목표 함수는 다 [1]음과 같이 정의된다:

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon) A_t \right) \right]$$

여기서 $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ 는 현재 정책과 이전 정책의 비율을 나타내며, ϵ는 클리핑 범위를 조절하는 하이퍼파라미터이다. 이 목표 함수를 최대화함으로써, PPO는 정책의 급격한 변화를 방지하면서도 효과적으로 학습을 진행할 수 있다.

B. Environment of Chess in PettingZoo

PettingZoo [2]는 다양한 멀티에이전트 강화학습 환경을 제공하는 라이브러리로, 체스 환경도 포함되어 있다. PettingZoo의 체스 환경은 두 명의 플레이어가 번갈아 가며 말을 움직이는 전통적인 체스 게임을 시뮬레이션한다. 각 플레이어는 자신의 차례에 가능한 모든 합법적인 수 중에서 하나를 선택하여 말을 이동시킨다. 환경은 관찰 공간, 행동 공간, 보상 구조 등을 정의하며, 세부 내용은 아래와 같다.

1) Observation Space: 체스 환경의 상태 공간은 체스판의 현재 상태를 나타내는 8x8 격자 형태로 구성된다. 각 격자 칸은 111개의 채널로 표현되며, 각 채널은 체스에서 사용되는 다양한 말과 특수 상태(예: 캐슬링 가능 여부, 앙파상 가능 여부 등)를 나타낸다. 이를 통해 에이전트는 체스판의 전체 상태를 정확하게 인식할 수 있다. 각 채널의 세부 구성은 부록 VI-A에 설명되어 있다.

2) Action Space: 행동 공간 또한 8x8의 격자의 의미를 담고 있지만, 3차원 Tensor를 flatten하여 0∼4672의 정수로 표현한다. 각 칸마다 73개의 가능한 행동이 존재하며, 이는 해당 칸에 있는 말을 이동시킬 수 있는 모든 합법적인 수를 나타낸다. 예를 들어, 폰의 경우 앞으로 한 칸 이동, 두 칸 이동, 대각선으로 상대 말을 잡는 등의 행동이 포함된다. 이러한 표현 방식을 통해 에이전트는 체스판에서 가능한 모든 행동을 선택할 수 있다. 각 행동의 세부 구성은 부록 VI-B에 설명되어 있다.

3) Reward Structure: 이기는 경우 +1, 지는 경우 -1, 무승부인 경우 0의 보상을 받는다. 본 프로젝트에서는 빠른 승리를 유도하기 위해 Reward Shaping 기법을 적용하였다.

C. Stockfish

Stockfish [3]는 오픈 소스 체스 엔진으로, 높은 수준의 체스 플레이 능력을 가지고 있다. 다양한 난이도 레벨을 제공하며, 본 프로젝트에서는 Stockfish 레벨 0을 사용하여 에이전트의 학습 상대방으로 활용하였다. Stockfish는 미니맥스 알고리즘과 알파-베타 가지치기 기법을 사용하여 최적의 수를 계산하며, 다양한 체스 전략과 전술을 구현하고 있다. 이를 통해 에이전트는 강력한 상대와의 대국을 통해 체스 실력을 향상시킬 수 있다.

III. Method

A. Reward Shaping

기존의 PettingZoo 체스 환경에서는 승리 시 +1, 패배 시 -1, 무승부 시 0의 보상을 제공한다. 그러나 이러한 보상 구조

는 에이전트가 게임의 승패에만 집중하게 하여, 중간 단계에서의 전략적 움직임을 학습하는 데 어려움을 겪을 수 있다. 따라서 본 프로젝트에서는 Reward Shaping 기법을 도입하여 에이전트가 더 효과적으로 학습할 수 있도록 하였다.

에이전트가 상대 기물을 잡았을 때에는 해당 기물의 가치에 0.1을 곱한 값을 보상으로 제공하고, 기물을 잃었을 때에는 동일한 값을 페널티로 부여하였다. 기물의 가치는 일반적으로 체스에서 사용되는 표준 가치를 기준으로 설정하였다. 0.1을 곱한 이유는 너무 큰 보상은 승패보다도 기물 교환에만 집중하게 만들 수 있기 때문이다. 이를 통해 에이전트는 단순히 승패에만 집중하는 것이 아니라, 중간 단계에서의 전략적 움직임을 학습할 수 있게 된다.

체크에 대해서 추가적인 보상을 제공하였다. 체크는 체스에서 상대방의 수를 제한하고 압박하는 중요한 전략적 요소이므로, 에이전트가 체크 상황을 적극적으로 활용하도록 유도하기 위해 체크 시 0.02의 보상을 추가로 제공하였다.

B. Pretraining with Behavior Cloning

초반 학습 단계에서 에이전트의 안정성을 높이기 위해 행동 클로닝(Behavior Cloning) 기법을 적용하였다. 행동 클로닝은 전문가의 행동 데이터를 활용하여 에이전트가 초기 정책을 학습하도록 돕는 방법이다. 본 프로젝트에서는 Stockfish 레벨 0의 대국 기록을 수집하여, 에이전트가 전문가의 행동을 모방할 수 있도록 하였다. 더 높은 레벨의 Stockfish를 사용하지 않은 이유는, 최종적인 목표인 Stockfish 레벨 0을 상대하는 데에 초점을 맞추었기 때문이다.

C. Curriculum Learning

Curriculum Learning은 학습 과정에서 점진적으로 난이도를 높여가는 방법으로, 에이전트가 더 효과적으로 학습할 수 있도록 돕는다. 본 프로젝트에서는 체스의 전략과 전술을 에이전트가 하나도 학습히자 못한 상태에서 시작하여 랜덤 상대와의 대국 이후에 Stockfish 레벨 0과의 대국을 통해 점진적으로 난이도를 높여가는 방식으로 Curriculum Learning을 적용하였다.

D. Agent Architecture

에이전트는 체스 판이 8x8 크기의 격자 형태로 표현되기 때문에, Convolutional Neural Network (CNN) 아키텍처를 사용하여 상태를 처리한다.

입력으로 들어온 Observation은 3x3 Convolution Layer와 ReLU 활성화 함수를 거친 후, Residual Block을 통과한다. Residual Block은 두 개의 3x3 Convolution Layer와 ReLU 활성화 함수로 구성되며, 입력과 출력이 더해지는 스킵 연결을 포함한다.

이후 Policy Head와 Value Head를 각각 통과하여 행동 확률 분포와 상태 가치 값을 출력한다. Policy Head와 Value Head도 Convolution Layer와 Fully Connected Layer로 구성되어 있다.

E. Implementation Details

학습 관련 주요 하이퍼파라미터는 Table I에 정리되어 있고, PPO 관련 하이퍼파라미터는 Table II에 정리되어 있다. Stockfish 레벨 0과 학습하는 경우에는 두 배의 환경을 사용하였다.

learning rate	# envs	# steps	# minibatches	update_epochs
3×10^{-4}	16(32)	256	8	4

TABLE I
학습 관련 주요 하이퍼파라미터

gamma	lambda	clip_coef	ent_coef	vf_coef
0.99	0.95	0.2	0.01	0.5

TABLE II
PPO 관련 주요 하이퍼파라미터

PPO 관련 하이퍼 파라미터 중 gamma는 discount factor로, 미래 보상의 현재 가치를 결정하는 역할을 한다. 본 프로젝트는 20수 이상의 장기적인 전술이 있는 체스의 특성을 고려하여 0.99로 설정하였다. lambda는 GAE(Generalized Advantage Estimation) 계산에 사용되는 파라미터로, 편향-분산 트레이드오프를 조절한다. 본 프로젝트에서는 0.95로 설정하여 적절한 균형을 유지하였다. clip_coef는 PPO의 클리핑 범위를 결정하는 하이퍼파라미터로, 정책 업데이트의 안정성을 높이는 역할을 한다. 본 프로젝트에서는 0.2로 설정하였지만 하한은 clip하지 않았다. 이는 에이전트가 초기 학습 단계에서 나쁜 행동을 했을 경우에는 PPO가 급격하게 정책을 업데이트하여 잘못된 행동을 수정할 수 있도록 하기 위함이다. ent_coef는 정책의 엔트로피 보상 계수로, 탐험을 촉진하는 역할을 한다. 본 프로젝트에서는 0.01로 설정하여 적절한 탐험을 유도하였다. vf_coef는 가치 함수 손실의 가중치로, 가치 함수의 학습에 대한 중요도를 조절한다. 본 프로젝트에서는 0.5로 설정하여 정책과 가치 함수의 균형을 맞추었다.

Stockfish 레벨 0과 대국하는 것은 연산량이 많아 병렬 처리를 통해 학습 속도를 높이는 것이 중요했다. 각 환경에서의 대국이 독립적이기 때문에, 여러 개의 환경을 동시에 실행하여 에이전트가 더 많은 경험을 쌓을 수 있도록 하였다. 이를 통해 학습 시간을 단축시키고, 에이전트가 더 빠르게 성능을 향상시킬 수 있었다.

IV. Result

A. Training Performance

Fig. 1. lose rate during training

학습결과는 Figure 1, 2, 3에 나타나 있다. 학습 초반에는 에이전트가 랜덤 상대와 대국을 하였기 때문에 패배율이 높았으나, 점차 학습이 진행됨에 따라 패배율이 감소하고 무승부 비율이 증가하는 경향을 보였다. 학습이 충분히 진행된 후에

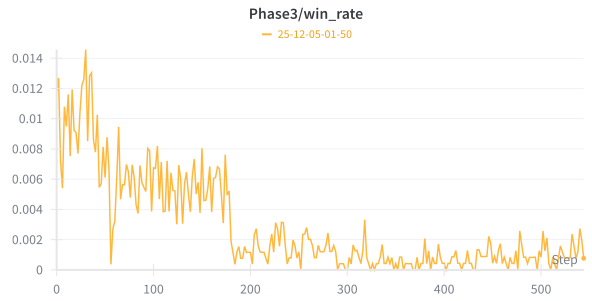

Fig. 2. win rate during training

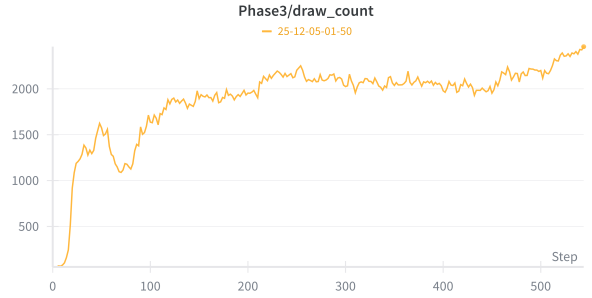

Fig. 3. draw count during training

는 lose rate가 약 0.048까지 감소하였으며, 대부분의 승부가 무승부로 나타났다.

다만, 승률은 크게 증가하지 않았는데, 이는 에이전트가 Stockfish 레벨 0과 대국하는 과정에서 Stalemate나 반복 무승부 상황이 발생하기 쉬웠기 때문으로 보인다.

B. Ablation Study

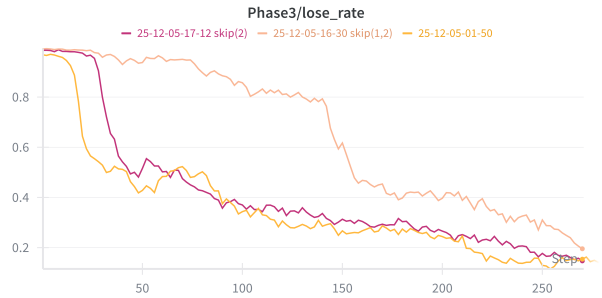

Fig. 4. Ablation Study Results

curriculum learning과 behavior cloning 기법의 효과를 평가하기 위해 ablation study를 수행하였다. 각 기법을 제거한 경우의 학습 성능을 비교한 결과, 두 기법 모두 에이전트의 학습 성능 향상에 긍정적인 영향을 미쳤음을 Figure 4에서 확인할 수 있었다.

behavior cloning과 curriculum learning을 모두 적용한 경우가 가장 빠르게 lose rate가 감소하는 경향을 보였으며, 그 다음으로 behavior cloning만 적용한 경우, 마지막으로 아무 기법도 적용하지 않은 경우가 가장 느리게 감소하는 경향을 보였다.

stockfish 레벨 0과 대국하며 학습하는 경우에는 stockfish의 연산으로 인해 학습 속도가 느려지기 때문에, behavior cloning과 curriculum learning 기법을 통해 초기 학습 속도를 높이는 것이 유의미했음을 알 수 있었다.

V. Conclusion

체스 환경에서 PPO 알고리즘을 활용하여 에이전트를 학습시키고, Curriculum Learning과 behavior cloning 기법을 적용하여 빠르게 성능을 향상시키는 방법에 대해 분석하였다. 실험 결과, 제안한 방법들이 에이전트의 학습 성능에 긍정적인 영향을 미쳤음을 확인할 수 있었다.

프로젝트 진행 기간이 짧아 추가적인 실험이 부족했으나, 본 코드를 더 오랜 시간 동안 학습시킨다면 더 나은 성능을 기대할 수 있을 것이다.

References

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[2] J. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. S. Santos, C. Dieffendahl, C. Horsch, R. Perez-Vicente *et al.*, "Pettingzoo: Gym for multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 032–15 043, 2021.

[3] The Stockfish developers (see AUTHORS file), "Stockfish," Software, version 1.2.0 (based on CFF version). [Online]. Available: https://stockfishchess.org/

VI. Supplementary Material

A. Observation Space Details

111개의 채널은 다음과 같이 구성된다:

- 0-3:
  - 백이 퀸 사이드 캐슬링이 가능하다면 모두 1
  - 백이 킹 사이드 캐슬링이 가능하다면 모두 1
  - 흑이 퀸 사이드 캐슬링이 가능하다면 모두 1
  - 흑이 킹 사이드 캐슬링이 가능하다면 모두 1
- 4: player가 백인지 흑인지
- 5: 50수 동안 폰이 움직이지 않았거나 기물이 잡히지 않은 경우 무승부 판정이 가능함. 이를 나타내기 위해 8x8을 flatten하여 현재까지 폰이 움직이지 않고 기물이 잡히지 않은 채 몇 수가 지났는지를 해당 index를 1로 하여 표현
- 6: Neural Network가 보드 edge를 인식할 수 있도록 모든 칸을 1로 채운 채널
- 7-18: 룩, 나이트, 비숍, 퀸, 킹, 폰 각각 백과 흑에 대해 2채널씩 총 12채널 각 채널에서는 해당 기물의 위치를 1로 표현. 앙파상 될 수 있는 폰의 경우 특별히 8행에 1로 표현
- 19: 현재 보드가 이전에 나타난 적이 있는지 여부를 나타내는 채널. 반복 무승부를 방지하기 위해 사용
- 20-111: 지난 7턴의 보드를 쌓아올린 채널. 가장 최신 보드가 20-32으로 표현되고, 채널이 증가할수록 과거의 보드 상태를 나타냄. 나타내는 보드의 상태는 7-19의 채널과 동일

B. Action Space Details

행동 공간의 각 행동은 다음과 같이 구성된다:

- 0-55: 퀸 이동에 대한 채널(비숍과 룩, 킹은 퀸의 제한된 버전으로 생각할 수 있음). 퀸이 이동할 칸 수 (1∼7)와 방향(8방향)을 조합하여 표현
- 56 63: 나이트 이동에 대한 채널. 나이트가 이동할 수 있는 8가지 경우를 각각 표현
- 64 73: 폰 이동에 대한 채널. 폰이 앞으로 1칸, 2칸 이동하거나 대각선(좌, 우)으로 상대 말을 잡는 경우, 프로모션하는 경우(퀸, 룩, 비숍, 나이트로 각각 프로모션) 등을 표현