

신경망 가중치 복제와 가중치 Q-Mix 를 이용한 DQN 학습 성능 개선

서용필¹, 성연식²

¹ 동국대학교 멀티미디어공학과 석사과정

² 동국대학교 컴퓨터·AI 학부 교수

seo104@dgu.ac.kr, sung@dongguk.edu

Improving Learning Performance of DQN Using Neural Network Weight Duplication and Weighted Q-mix

Yongphil Seo¹, Yunsick Sung²

¹Dept. of Multimedia Engineering, Dongguk University-Seoul

²Dept. of Computer Science and Artificial Intelligence, Dongguk University-Seoul

요약

강화학습(Reinforcement Learning, RL)은 복잡하고 동적인 환경에서 효과적인 의사결정을 가능하게 한다. 특히, Deep Q-Network(DQN)는 전통적인 강화학습과 비교할 때 고차원 상태 공간을 처리하는 데 강점을 보이나, 학습 성능의 일관성 부족과 속도 불균형 등의 한계가 발생한다. 본 연구는 DQN에 신경망 가중치 복제와 Q-mix 기법을 통합한 방법을 제안한다. 가중치 복제와 제한된 업데이트로 계산 효율성을 높이고, Q-값 통합을 통해 의사결정의 신뢰성과 정밀도를 강화한다. 실험 결과에서는 전통적인 방법과 비교할 때 80% 이상 향상된 학습 속도와 안정성을 달성하며, DQN의 기존 한계를 개선할 수 있는 잠재력을 확인하였다.

1. 서론

강화학습(Reinforcement Learning, RL)은 에이전트가 환경과의 상호작용을 통해 최적의 행동을 스스로 학습하는 방법으로, 여러 분야에서 활발히 적용되고 있다. 특히, Deep Q-Network(DQN)[1]는 Q-Learning[2]과 심층 신경망을 결합해서 고차원 상태 공간에서도 효과적인 학습을 가능하게 한다. 하지만, 학습 데이터 활용 효율과 안정적인 수렴 면에서 한계를 보인다. 이를 보완하기 위해 Maxmin Q-Learning[3], Averaged DQN[4] 등의 다양한 변형 기법들이 있지만 Q-값 추정의 불확실성을 완전히 해소하지는 못했다. 최근 Random Ensemble Mixture (REM)[6]은 무작위 가중 결합으로 불확실성 완화를 시도했으나, 모델 선택의 어려움과 계산 자원의 소모가 크다는 단점이 있다.

본 논문은 두 DQN 모델의 Q-값을 정규화 된 가중치로 혼합하는 가중치 Q-mix 를 제안한다. 이 방법은 모델 간 네트워크 가중치 복제와 Q-값 혼합을 통해 다양한 탐색을 유도하며, 학습 안정성과 효율성을 향상시킨다.

이 논문은 2 장에서 관련 연구를 소개한다. 3 장에서 제안 기법을 설명한다. 4 장에서 실험 결과를, 5 장에서 논의를, 6 장에서 결론을 제시한다.

2. 관련 연구

DQN은 심층 신경망을 Q-Learning에 접목하여 복잡한 환경에서 효율적 학습을 가능케 했으며, Atari 게임에서 뛰어난 성과를 보인다. 경험 리플레이[5]와 타겟 네트워크 도입으로 학습 안정성을 크게 개선한다. 하지만, 샘플 효율성, 학습 속도, Q-값 과대추정 문제가 여전히 존재한다.

Random Ensemble Mixture(REM)[6]는 다수 개 Q-네트워크를 활용하는 양상을 기반 접근법으로, 다양한 Q-값을 무작위 가중치로 동적 조합하여 불확실성을 효과적으로 다루어 탐색 능력을 강화한다. 단일 네트워크 내 다중 Q-함수를 구현하고 무작위 가중치로 결합해 다양한 Q-값 추정치를 생성한다. REM은 오프라인 강화학습에서 탁월한 성능과 일반화 능력을 보여주지만, 양상을 구성원 간 다양성 제한과 무작위 가중치 선택의 최적성을 보장할 수 없다는 한계도 존재한다.

3. 제안 방법

본 논문은 DQN의 학습 안정성과 속도를 개선하기 위해, 네트워크 가중치(Network weight) 복제와 혼합(Mixing Weights)기반의 Q-값 혼합(Weighted Q-Mix)을 융합한 방법을 제안한다. 제안 방법에서는 두 개의 모델을 사용하며, 하나의 모델만 직접 학습하고 다른 모델은 주기적으로 학습 모델의 네트워크 가중치를 부분적으로 전달받는다(그림 1). 이 과정에서 네트워크 가중치를 복제할 때는 0과 1 사이의 임의로 생성된 복제 비율을 사용하여 복제한다.

두 모델의 Q-값을 결합할 때는 각 모델에서 개별적으로 산출된 Q-값에 무작위로 생성된 정규화 된 혼합 가중치(Weights)를 적용하여 최종 Q-값을 도출한다. 이 방식은 모델 간의 다양성과 안정성을 유지하면서 학습 속도를 향상시킨다. 추가적으로 무작위로 선택된 행동을 One-hot Encoding[7]하여 탐색 항(exploration term)으로 추가 한다. 그래서 무작위성을 유도하여 탐색(exploration)과 이용(exploitation)의 균형을 적절하게 유지한다. 최종 -Q 값 (q_t^X)을 기반으로 최대-Q 값에 해당하는 행동(a_t)을 선택한 후, 탐색률(epsilon)을 갱신한다.

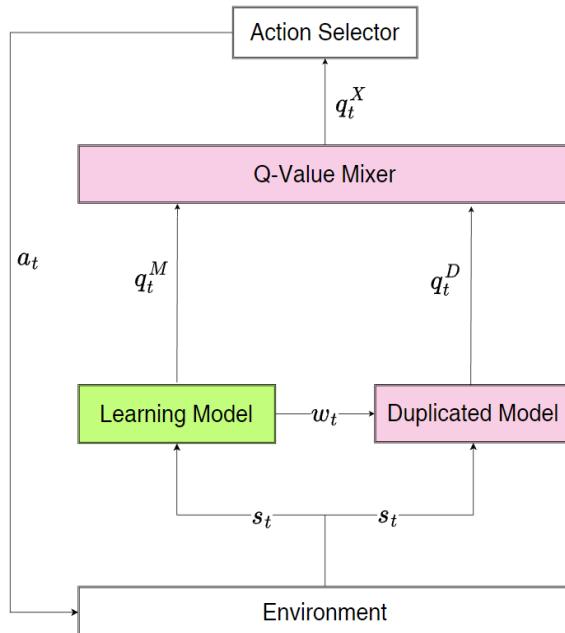


그림 1. 신경망 가중치 복제와 가중치 Q-mix 과정

4. 실험 결과

본 논문은 10×10 크기의 Catch Game을 실험 환경으로 사용하였다. 이 게임은 플레이어가 하강하는 과일을 받아내는 것이 목적이며, 왼쪽 이동, 정지, 오른쪽 이동의 세 가지 행동이 가능하다. 과일을 받으면 +1 점, 놓치면 -1 점, 이동만 하면 0 점의 보상이 주어진다. 학습 성능 평가는 손실함수(Loss)와 승리 비율(Win Rate)을 주요 지표로 설정하였다.

손실 값 분석 결과(그림 2 와 3), 전통적인 DQN 모

델은 약 7,000 Epochs 전후로 수렴하나 이후에도 불안정한 학습 양상을 보였다. 하지만, 제안된 모델은 약 4,000 Epochs 이내에 빠르게 수렴하였으며, 이후에도 낮고 안정적인 손실 값을 유지하였다.

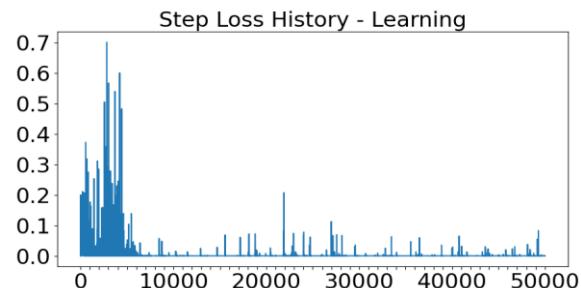


그림 2. 전통적인 DQN - Loss

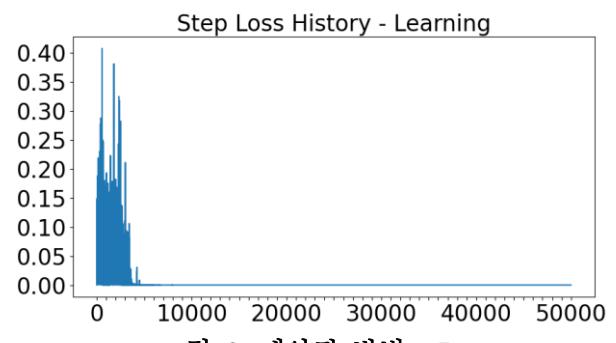


그림 3. 제안된 방법 - Loss

평균 승리 비율을 비교한 결과 (그림 4 와 5), 전통적인 모델은 약 7,000 Epochs 이후에나 80% 이상의 성능을 보였으나, 이후에도 성능 변동이 나타나며 최적 성능(100%)에 안정적으로 도달하지 못하였다. 제안된 모델은 약 4,000 Epochs 이내에 빠르게 승리 비율 100%에 도달하였고, 이후 일관되게 최적 성능을 유지했다.

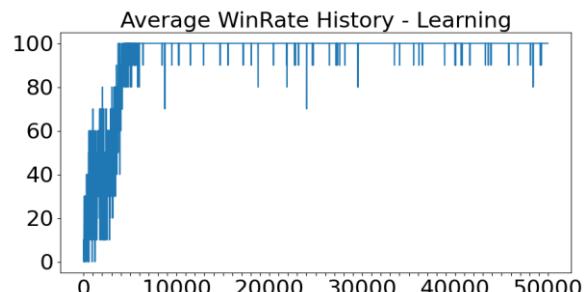


그림 4. 전통적인 DQN - 승률

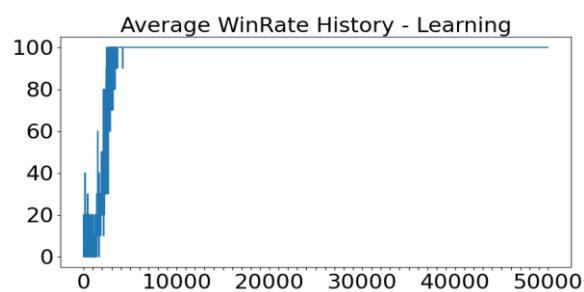


그림 5. 제안된 방법 - 승률

검증 과정(그림 6 과 7)에서도 전통적인 모델은 성능 변동으로 인해 승률 100%를 유지하지 못했지만, 제안된 모델은 일관된 성능을 보이며 안정적인 결과를 나타냈다.

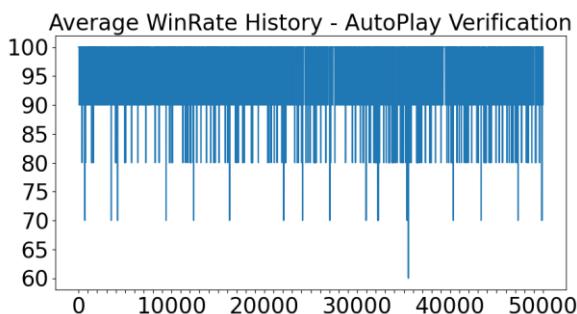


그림 6. 전통적인 DQN – 검증

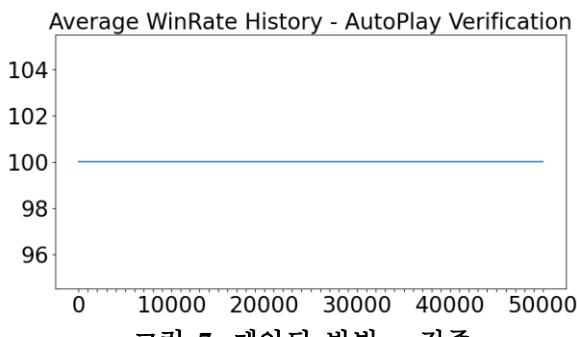


그림 7. 제안된 방법 – 검증

5. 논의

실험 결과를 종합하면, 제안된 모델은 전통적인 모델에 비해 학습 효율성, 안정성에서 우수한 성과를 보였다. 첫 번째, 손실 값이 더 빠르게 수렴하며 적은 학습 횟수로 최적 정책에 도달해 정책 탐색이 효율적이었다. 두 번째, 학습이 진행될수록 손실 값이 안정적으로 감소한 반면, 기본 모델은 후반까지도 불규칙한 손실 변동을 보였다. 전통적인 모델의 학습률, Replay Buffer, 타겟 네트워크 갱신 주기 등이 최적화되지 않았을 가능성을 시사한다. 세 번째, 제안된 모델은 학습 후반과 검증 단계 모두에서 100% 승률을 유지하며 안정적인 성능을 보였고, 전통적인 모델은 일부 구간에서 성능 저하가 나타났다. 본 연구는 10×10 그리드 기반의 제한된 환경에서 수행되었으므로, 향후 복잡한 환경에 대한 확장 연구가 필요하다. 사용된 하이퍼파라미터 설정이 다른 환경에서도 유효하지 확인하고, 이에 대한 최적화 연구가 필요하다.

6. 결론

본 본문은 DQN에 Network weight 복제와 가중치 Q-Mix 기법을 적용하여, Catch Game 환경에서 더 나은 성능을 달성하기 위한 방안을 제시하였다. 제안된 모델은 전통적인 모델에 대비 빠른 수렴 속도와 안정적이고 신뢰성 있는 성능을 보였다. 계산 자원의 효율적 활용과 모델 간 다양성 유지, 학습 시간 단축 등의 이점을 제공했다. 손실 함수 및 승리 비율 분석

결과, 빠르고 안정적인 학습과 높은 최종 성능, 그리고 검증 단계에서도 안정적인 성능을 나타냈다. 제안된 방법이 향후 복잡한 강화학습 문제에도 유용하게 활용될 가능성을 보여준다. 다만, 제한된 환경에서 실험이 이루어진 만큼, 다양한 조건에서의 추가 검증이 필요하다. 향후 연구는 복잡한 환경에서의 성능 검증, 하이퍼파라미터 최적화 와 다른 학습 기법과의 결합, 새로운 학습 전략을 통한 적응성과 일반화 성능 향상에 초점을 둘 수 있다. 본 연구는 향후 강화학습 성능 향상을 위한 후속 연구에 참고자료로 기여할 수 있다.

사사

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업 연구 결과로 수행되었음(IITP-2025-RS-2023-00254592).

참고문헌

- [1] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. "Playing Atari with Deep Reinforcement Learning". Proceedings of the 27th Conference on Neural Information Processing Systems (NIPS) Deep Learning Workshop, Lake Tahoe, USA, 2013, pp. 1–9.
- [2] Jang, H., Hao, S., Chu, P. M., Sharma, P. K., Sung, Y., & Cho, K. "Deep Q-network-based multi-criteria decision-making framework for virtual simulation environment. Neural Computing and Applications", 33(19), 2021, pp. 10657–10671.
- [3] Lan, Q., Pan, Y., Fyshe, A., & White, M. "Maxmin Q-Learning: Controlling the Estimation Bias of Q-Learning". Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 2020, pp. 1–9.
- [4] Anschel, O., Baram, N., & Shimkin, N. "Averaged-DQN: Variance Reduction and Stabilization for Deep Reinforcement Learning". Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 2017, pp. 1–9.
- [5] Gu, B., & Sung, Y. "Enhanced DQN Framework for Selecting Actions and Updating Replay Memory Considering Massive Non-Executable Actions". Applied Sciences, 11(23), 2021, Article 11162.
- [6] Agarwal, R., Schuurmans, D., & Norouzi, M. "An Optimistic Perspective on Offline Reinforcement Learning". Proceedings of the 37th International Conference on Machine Learning (ICML), Vienna, Austria, 2020, pp. 1–9.
- [7] Gu, B., & Sung, Y. "Enhanced Reinforcement Learning Method Combining One-Hot Encoding-Based Vectors for CNN-Based Alternative High-Level Decisions". Applied Sciences, 11(3), 2021, Article 1291.