멀티에이전트 강화학습 기술 동향: 분산형 훈련-분산형 실행 프레임워크를 중심으로

Survey on Recent Advances in Multiagent Reinforcement Learning Focusing on Decentralized Training with Decentralized Execution Framework

신영환 (Y.H. Shin, shinyh1115@etri.re.kr) 정보전략부 Post-Doc.
서승우 (S.W. Seo, seosw@etri.re.kr) 정보전략부 Post-Doc.
유병현 (B.H. Yoo, bhyoo@etri.re.kr) 복합지능연구실 선임연구원
김현우 (H.W. Kim, kimhw@etri.re.kr) 복합지능연구실 책임연구원
송화전 (H.J. Song, songhj@etri.re.kr) 복합지능연구실 책임연구원/실장
이성원 (S. Yi, sungyi@etri.re.kr) 정보전략부 책임연구원/부장

ABSTRACT

The importance of the decentralized training with decentralized execution (DTDE) framework is well-known in the study of multiagent reinforcement learning. In many real-world environments, agents cannot share information. Hence, they must be trained in a decentralized manner. However, the DTDE framework has been less studied than the centralized training with decentralized execution framework. One of the main reasons is that many problems arise when training agents in a decentralized manner. For example, DTDE algorithms are often computationally demanding or can encounter problems with non-stationarity. Another reason is the lack of simulation environments that can properly handle the DTDE framework. We discuss current research trends in the DTDE framework.

KEYWORDS 멀티에이전트 강화학습(MARL), 분산형 훈련-분산형 실행(DTDE)

Ⅰ. 서론

주어진 환경(Environment)에서 에이전트(Agent)를 상태(State)에 따라 보상(Reward)을 극대화하는 방식으로 행동(Action)하도록 학습시키는 것을 강화학습이라고 한다. 멀티에이전트 강화학습(MARL: Multi-Agent Reinforcement Learning)은 다수의 에이전트가 존재하여 각 에이전트의 보상을 극대화하도록 학습시키는 방법을 말한다[1]. 앞서 참고문헌 [2]에서는 중앙집중형 훈련-분산형 실행(CTDE: Centralized

* DOI: https://doi.org/10.22648/ETRI.2023.J.380409
* 본 연구는 한국전자통신연구원 내부연구과제의 일환으로 수행되었음[멀티에이전트 강화학습 탐색, 통신, 학습전략 기술 연구, 22YE1210, 자율성장형 복합인공지능 원천기술 연구, 23ZS1100].

Training with Decentralized Execution) 프레임워크를 중심으로 MARL에서 해결하고자 하는 문제와 관련 알고리즘들을 소개하였다.

본고에서는 참고문헌 [2]에 이어 분산형 훈련-분산형 실행(DTDE: Decentralized Training with Decentralized Execution) 프레임워크를 중심으로 MARL에서 해결하고자 하는 문제와 해당 알고리즘을 분류하여 기술하고자 한다.

참고문헌 [2]에서 서술한 대로 에이전트가 다수 존재하는 MARL 환경에서는 CTDE 프레임워크를 채택한 연구가 주류를 이루고 있다. CTDE와 DTDE 두 가지 프레임워크 중 하나를 선택할 수 있는 상황이라면 CTDE를 선택하는 것이 많은 경우에서 효과적이기 때문이다. 그 이유는 다음과 같다.

DTDE와 CTDE 두 프레임워크는 알고리즘 실행 과정에서는 각각의 에이전트의 관측 정보만 활용한다는 면에서 동일하다. 두 프레임워크는 훈련 방식에서 차이가 있다. DTDE 프레임워크는 훈련 과정에서도 각각의 에이전트의 정보만 활용한다. 반면, CTDE 프레임워크는 각각의 에이전트가 가지고 있는 정보를 공유한다[3]. 따라서 훈련 시에 모든 에이전트의 정보를 한 번에 이용하는 CTDE 프레임워크가 그렇지 않은 DTDE 프레임워크보다 효과적일 수밖에 없다.

그러나, 실생활에서는 DTDE 프레임워크를 사용할 수밖에 없는 경우가 많다. 예를 들면, Self-driving Car 같은 환경에서는 중앙집중형 모델 하나가 모든 에이전트의 관측과 행동 정보에 접근하는 것이 불가능하다. 또한, 개개 에이전트의 프라이버시가 가장 우선순위로 고려되어야 하는 환경에서는 에이전트 간 정보 교류가 없는 DTDE의 사용이 선호된다[4].

본고의 구성은 다음과 같다. Ⅱ장에서는 DTDE 프레임워크에서 발생하는 고유의 문제점인 학습 속도 저하, 비정상성 문제에 관해 설명한다. 1절에서는 에이전트 간 교류할 수 있는 정보에 관해 다루고, 2절과 3절에서는 에이전트 간 정보 교류가 부재하여 발생하는 학습 속도 저하 문제와 비정상성 문제에 관해 자세히 다룬다.

Ⅲ장에서는 DTDE 프레임워크에 해당하는 알고리즘들을 다룬다. 1절에서는 가치 함수 기반 DTDE 알고리즘에 대해 소개하고, 2절에서는 정책 경사 기반 DTDE 알고리즘에 관해 다룬다. 3절에서는 1, 2절에서 다룬 기본 알고리즘들을 기반으로 DTDE에서 발생하는 문제점들을 개선한 알고리즘들에 관해 서술한다.

Ⅳ장에서는 DTDE 프레임워크를 실험할 수 있는 Extended Python MARL(EPyMARL) 환경[5]과 EPyMARL에서 제공하는 시나리오와 기본 알고리즘을 다룬다.

Ⅱ. DTDE 프레임워크에서의 문제

1. 에이전트 간 정보 교류 부재

DTDE 프레임워크와 CTDE 프레임워크의 차이점은 훈련 시 에이전트 간 정보가 부재하다는 것이다. 정보의 종류로는 다음과 같다. CTDE에서는 관찰 정보와 학습 정보를 둘 다 공유할 수도 있다.

가. 관측 정보

CTDE 프레임워크에서는 에이전트 간 관측 정보를 공유하며 더 많은 상태 정보를 갖고 훈련을 진행한다. 에이전트 간 통신을 통해 관측 정보를 알게 되는 경우도 있고, 관측 정보가 글로벌로 공유된다는 가정하에 있는 환경에서 강화학습을 진행하는 경우도 있다.

나. 학습 정보

CTDE 프레임워크에서 학습 정보는 흔히 파라미

터 공유(Parameter-sharing)를 통해 이루어진다[6,7]. 파라미터 공유란 훈련 시에 모든 에이전트가 하나의 강화학습 모델을 훈련시키는 것을 말한다. 대부분의 최근 강화학습 모델은 신경망을 기반으로 구성되어 있는데, 이 경우에는 하나의 신경망을 모든 에이전트가 학습시키는 것을 의미한다.

다. DTDE에서의 정보 교류 부재

반면, DTDE 프레임워크에서는 앞서 살펴본 두 가지 정보를 사용할 수 없다. DTDE 프레임워크에서 이 두 가지 핵심 정보의 부재는 비정상성(Non-stationarity) 문제와 학습 속도의 저하를 야기한다.

2. 비정상성 문제

이 절에서는 흔히 움직이는 표적 문제(Moving Target Problem)라고도 일컬어지는 비정상성 문제에 관해 서술한다[8]. 강화학습의 이론적 바탕은 마르코프 가정(Markov Assumption)이다. 마르코프 가정에 따르면, 에이전트가 학습을 진행함에 따라 환경을 수리통계적으로 이해할 수 있게 된다.

MARL에서 각각의 에이전트는 자신을 제외한 타 에이전트를 환경으로 인식한다. 따라서 에이전트는 타 에이전트들의 행동을 확률적으로 이해하고 있어야 학습 방향을 정할 수 있다. 그러나 MARL에서는 타 에이전트들의 학습과 탐색이 마르코프 가정을 붕괴시킨다.

가. 학습으로 발생한 비정상성 문제

MARL에서 학습을 진행할 때 문제는 모든 에이전트가 학습을 진행하면서 자신의 행동 전략을 계속 수정해 나간다는 점에 있다. MARL 에이전트는 타 에이전트가 행동 전략을 수정하면 환경이 급작스럽게 변화한 것으로 인식한다. 따라서 마르코프 가정에 기반한 수리통계적 환경 이해가 불가능해지는 것이다. 이렇게 되면 지금까지 확인했던 타 에이전트의 행동에 관한 데이터의 통계적 의미가 퇴색된다. 이렇게 타 에이전트들이 학습을 진행하면서 마르코프 가정이 무너지는 문제를 비정상성 문제라고 한다.

나. 탐색으로 발생한 비정상성 문제

강화학습 에이전트는 현재 최적이라고 생각하는 행동을 이용(Exploitation)하기도 하고 새로운 행동을 탐색(Exploration)하며 최적의 행동 전략을 찾아낸다. 탐색에도 입실론-그리디, 엔트로피 등 다양한 전략이 있다. 탐색 전략은 대체로 무작위적 성향을 띠게 된다. MARL 에이전트 입장에서는 타 에이전트가 탐색하고 있는지, 이용하고 있는지 아무런 정보 없이는 구분할 방법이 없다. 따라서 타 에이전트가 탐색하는 것도 이용으로 간주하며 훈련을 진행할 수밖에 없다. 타 에이전트의 탐색 전략이 다변화되고 복잡할수록 MARL 에이전트 입장에서는 환경이 비정상성 성향을 띠게 된다고 느껴진다.

다. 비정상성 문제 예시

MARL에서의 비정상성을 설명할 수 있는 예로는 가위바위보 게임이 있다. 가위바위보 게임에서 두 에이전트는 서로를 이겨야 하므로 경쟁적 멀티에이전트 강화학습에 해당한다. 만약 상대 에이전트의 행동 전략이 고정된 상태라면 게임을 진행하면서 에이전트는 상대 행동 전략에 맞게 학습을 수행할 수 있다.

상대가 주먹만 내는 행동 전략을 취한다면 보자기만 내는 전략을 학습하면 되고, 상대가 1/3의 확률로 주먹, 가위, 보자기를 선택하여 낸다면, 똑같이 1/3 확률로 주먹, 가위, 보자기를 선택하여 내는 전략을 학습하면 될 것이다. 그러나, 상대 에이전트

도 계속 학습을 하며 전략을 바꿔나간다면 과거의 데이터의 수리통계적 의미가 퇴색되므로 마르코프 가정에 기반한 학습을 진행할 수가 없게 된다. 예시로 든 경쟁적 환경뿐만 아니라 협력적 환경에서도 비정상성 문제는 똑같이 발생할 수 있다.

CTDE 프레임워크에서는 서로의 정보를 공유하기 때문에 비정상성 문제가 일어나는 것을 예방할 수 있다. 에이전트 간 정보 공유를 통해 한 에이전트의 수정된 행동 전략도 타 에이전트들이 충분히 인지할 수 있기 때문이다. 반면, 에이전트 간 정보 교류가 원천 차단되어 있는 DTDE 프레임워크에서는 비정상성 문제를 근본적으로 없애는 것은 불가능하다. 따라서 알고리즘적으로 비정상성 문제 저감 조치를 취하는 식으로 많이 연구되고 있다.

3. 학습 속도 저하 문제

정보와 데이터가 집약될수록 학습이 효율적으로 된다는 것은 CTDE 프레임워크가 발전하게 된 이유이기도 하다. 바꾸어 말하면 DTDE에서는 상대적으로 학습 속도가 저하된다는 것을 의미한다. DTDE 프레임워크하의 에이전트들은 타 에이전트들의 행동 전략을 직접 경험해 보며 통계적으로 체득할 수밖에 없다. 아울러, 타 에이전트가 경험해 봤던 상태, 관측 정보들을 이용할 수 없으니 똑같은 상황이나 최소한 비슷한 상황이라도 직접 경험해 봐야 한다. 에이전트 입장에서는 탐색 공간이 굉장히 넓어진다는 것을 의미한다.

또한, 파라미터 공유를 활용하지 않으니 학습해야 하는 파라미터 수가 늘어난다. 파라미터를 공유할 때는 모든 에이전트가 하나의 신경망을 협력적으로 학습시킬 수 있다. 반면, DTDE처럼 각각의 에이전트가 개별 신경망을 가지고 있는 경우에는 파라미터 수가 에이전트 수에 비례하여 늘어난다.

Ⅲ. DTDE 알고리즘

이 장에서는 가장 기본적인 DTDE 알고리즘 세 가지와 기본 알고리즘을 기반으로 DTDE의 문제점을 개선한 알고리즘들을 소개한다. 다음에서 소개하는 DTDE 알고리즘들은 파라미터 공유를 하지 않는 경우를 상정한다. 앞으로 언급할 알고리즘들도 파라미터 공유를 할 때 훈련 정보를 공유하는 것이기 때문에 CTDE 알고리즘에 해당한다.

1. 가치 함수 기반 방법

가장 기본이 되는 가치 함수 기반 DTDE 알고리즘은 Independent Q-Learning(IQL)이다[9]. 다양한 버전의 Q 네트워크를 구성할 수 있으나 현재 대부분 IQL은 각각의 에이전트가 Deep Q-Network(DQN)[10]를 지니고 있도록 설계된 버전을 베이스라인으로 사용하고 있다. 에이전트 간 단절된 정보에 대한 고려가 없는 알고리즘이므로 언급한 DTDE에서의 문제점을 그대로 지니고 있다.

2. 정책 경사 기반 방법

가장 기본이 되는 정책 경사 함수 기반 알고리즘에는 Independent synchronous Advantage Actor-Critic(IA2C)[11]과 Independent Proximal Policy Optimisation(IPPO)이 있다[12].

IA2C 알고리즘은 각각의 에이전트가 하나 존재하는 강화학습에서 흔히 사용하는 A2C 알고리즘[13]을 탑재하도록 설계되었다. IPPO 알고리즘 역시 각각의 에이전트가 강화학습에서 일반적으로 사용하는 PPO 알고리즘[14]을 탑재하도록 설계되었다. IA2C와 IPPO 모두 에이전트 간 단절 된 정보 교류에 대한 고려가 없기 때문에 DTDE에서 발생하

는 문제점을 그대로 가지고 있다.

3. 개선된 알고리즘

DTDE 프레임워크에서 비정상성 문제를 가장 잘 억제하는 방법은 한 에이전트가 정책을 수정할 때 타 에이전트들은 정책을 수정하지 않는 것이다. 이렇게 한 에이전트씩 정책을 학습해 나가면 다른 에이전트도 변경된 정책을 충분히 참고할 수 있게 된다. 최근 연구에서 이렇게 순차적 학습을 진행하면 각 에이전트는 최적의 정책을 찾을 수 있다는 것이 수학적으로 증명되었다[15]. 그러나 이런 순차적 학습의 가장 큰 문제점은 DTDE 프레임워크의 학습 속도가 더 느려진다는 것이다. 따라서 순차적 학습과 평행적 학습의 중간에서 타협을 보는 알고리즘이 많이 연구되고 있다.

참고문헌 [16]에서는 Hysteric Q-Learning(HQL) 알고리즘을 제안하였다. IQL을 기반으로 이 알고리즘은 느린 학습률과 빠른 학습률의 두 가지 학습률을 채택한다. 가치(Q-value)가 작을 것으로 예상되는 부정적인 상황에서는 느린 학습률을 채택하고, 가치가 클 것으로 예상되는 긍정적인 상황에서는 빠른 학습률을 채택한다. 느린 학습률로 학습하는 에이전트가 생기기 때문에 일반적인 IQL보다 비정상성 문제가 완화되는 효과를 가진다.

참고문헌 [17]에서는 Lenient Deep Q-Network (LDQN) 알고리즘을 제안하였다. 이 알고리즘 또한 IQL을 기반으로 하는 알고리즘이다. 일정한 확률로 가치가 작은 상황에서는 업데이트하지 않는 알고리즘이다. 이 확률은 시간에 따라 점점 감소하므로 훈련 후반부에는 일반적인 IQL과 똑같아진다. LDQN 또한 학습을 안 하는 경우가 있기 때문에 비정상성 문제가 완화된다.

참고문헌 [15]에서는 완화된 버전의 순차적 학습 방법을 제안하였다. 이른바 단계별 학습(Staged Learning) 개념을 제안하며, 상대적으로 많이 학습하는 에이전트와 적게 학습하는 에이전트를 번갈아 가며 선택할 것을 제안했다. 일반적 순차적 학습에서는 정책을 학습하지 않는 에이전트가 있는 반면, 단계별 학습(Staged Learning)에서는 정책을 학습하지 않는 것이 아니라 적게 학습한다는 면에서 차이가 난다. 따라서 Staged Learning 개념을 탑재한 Staged IPPO(SIPPO)와 Staged IQL(SIQL) 모두 IPPO와 IQL보다 좋은 성능을 보였다.

참고문헌 [18]에서는 Ideal Independent Q-Learning (I2Q) 알고리즘을 소개하였다. 본 알고리즘에서는 비정상성 문제를 해결하기 위해 이상적 전이 확률(Ideal Transition Probability) 개념을 제안하였다. 일반적인 가치(Q-value)는 다음 행동의 기대치로 계산된다. I2Q 알고리즘에 따르면 다음 상태 가치를 기반으로 이상적 전이 확률을 계산할 때 다른 에이전트가 행동 전략을 수정해도 비정상성 문제를 적게 겪는다.

기존의 IQL에서의 데이터를 저장하는 방식인 리플레이 버퍼를 DTDE에 맞게 활용하는 방법도 있다. 참고문헌 [19]에서는 Concurrent Experience Replay Trajectories(CERT) 개념을 제안하며, 에이전트들이 동시에 경험을 쌓아나갈 때 비정상성을 최대한 억제하며 데이터를 다루는 방법을 서술하였다.

IA2C 기반 알고리즘에 에이전트 간 효율적 통신을 고려한 F2A2 같은 알고리즘도 있다[20]. 이렇게 되면, CTDE에서의 에이전트 간 정보 교류와 비슷한 맥락이 있어 완전한 DTDE라고 할 수는 없지만, 최소한의 정보 교류를 목표로 하므로 정보 교류가 완전히 제한되지 않고 어느 정도 제한된 부분적 DTDE 환경에서 사용할 수 있도록 한다는 점에서 의미가 있다.

표 1 EPyMARL에서 제공하는 실험 환경

환경명	행동	목표
MPE	동, 서, 남, 북 및 시나리오에 따른 특수한 행동	MPE 내에 시나리오에 따라 목표 변화
LBF	None, 동, 서, 남, 북, 음식 적재	흩어져 있는 음식을 모두 적재하기
RWARE	좌향좌, 우향우, 직진, 선반 적재, 선반 하역	흩어진 선반을 목표 지점으로 옮기기
SMAC	동, 서, 남, 북, 유닛 유형에 따라 공격 혹은 치료	교전 승리

출처 Reproduced from [5].

Ⅳ. 실험 환경

최근 DTDE 프레임워크 기반 MARL 알고리즘을 실험해 볼 수 있는 강화학습 환경으로는 EPyMARL이 각광 받고 있다[5]. 기존에 참고문헌 [2]에서 소개한 StarCraft Multi-Agent Challenge(SMAC) 환경은 PyMARL 기반이다[21]. 참고문헌 [2]에서 소개했던 Multi-Agent Particle Environment(MPE) 환경[22]은 PyMARL과 분리되어 있어 연구자가 실험해 보고 싶은 알고리즘을 원하는 환경에 맞춰 일일이 코드를 수정해야 하는 불편함이 있었다.

또한, PyMARL 실험 환경에는 대체로 가치 함수 기반 알고리즘이 주로 탑재되어 있었다. 게다가 대부분 파라미터 공유가 디폴트로 내재되어 있어 파라미터 공유를 하지 않은 채 실험을 진행하려면 연구자가 스스로 알고리즘을 수정해야 하는 불편함이 있었다.

표 1에서 볼 수 있듯이, EPyMARL에서는 SMAC 환경뿐만 아니라 기존에 MARL에서 자주 쓰는 환경인 MPE, Level-Based Foraging(LBF), Multi-Robot-Warehouse(RWARE) 환경[23]을 통합하여 제공하고 있다.

또한, 파라미터 공유를 간단한 명령어로 자유롭게 해제할 수 있어 DTDE 프레임워크 실험에 적합하다. 파라미터 공유를 해제하기 위해선 알고리즘마다 파라미터 비공유(Non-sharing)를 뜻하는 -NS를 입력하면 된다. 예를 들어, IPPO의 경우 ippo-ns라고 입력하면 된다. 표 2에서 언급된 IPPO, IA2C, IQL는 파라미터를 비공유할 때만 DTDE로 실험할 수 있다는 것을 주의해야 한다. CTDE로 분류된 알고리즘들은 파라미터를 공유하지 않더라도 관측 상태를 공유하기 때문에 CTDE로 분류된다.

참고문헌 [2]에서 설명한 SMAC과 MPE 환경은 EPyMARL에서도 제공하고 있으므로, 본고에서는 LBF와 RWARE 환경을 조금 더 다루어 본다.

1. Level-Based Foraging

LBF 환경은 MPE와 비슷하게 그리드 세계를 탐색하는 환경이다. 다른 에이전트와 협력하여 흩어져 있는 음식을 모두 적재하는 것이 목표이다. LBF 환경에는 레벨 개념이 추가적으로 더해져 단순히 음식을 수집할 수 없고 레벨에 맞춰 음식을 수집해야 한다.

표 2 EPyMARL에 탑재되어 있는 알고리즘

프레임워크	유형	알고리즘명
DTDE	정책 경사 기반	IPPO, IA2C
DTDE	가치 함수 기반	IQL
CTDE	정책 경사 기반	MADDPG, COMA, MAA2C, MAPPO
CTDE	가치 함수 기반	VDN, QMIX

출처 Reproduced from [5].

보다 구체적으로는 에이전트에게 레벨이 무작위적으로 지정되며, 그리드 세계에 흩어져 있는 음식들에도 레벨이 무작위적으로 지정된다. 에이전트는 동, 서, 남, 북 불연속적 행동을 수행할 수 있으며, 음식 옆에서 음식 적재를 시도할 수 있다. 이때, 에이전트가 음식보다 레벨이 같거나 높으면 혼자서 음식을 적재할 수 있다. 그러나, 음식의 레벨이 더 높은 경우 타 에이전트와 협력해야 한다. 타 에이전트와 자신의 레벨을 더해서 음식보다 더 높거나 같은 레벨을 갖는다면 음식을 적재할 수 있다. 셋 이상의 에이전트가 있는 경우 어떤 에이전트와 협업해야 음식을 적재할 수 있을지 판단하는 것 또한 이 환경의 도전 과제이다.

2. Multi-RobotWarehouse

RWARE 환경은 다수의 로봇이 요청받은 상품들을 이동시켜야 하는 창고 시뮬레이션 환경이다. RWARE 환경 역시 그리드 세계를 기반으로 만들어졌다. 강화학습의 시각에서 로봇이 에이전트이다. LBF와 비슷한 점은 에이전트가 목표물인 상품이 놓인 선반을 찾아 적재해야 한다는 것이다. LBF와의 차이점은 RWARE 환경에는 레벨 개념이 존재하지 않아 어떤 로봇도 어떤 선반이든 적재할 수 있다. 또한, 선반을 적재하는 것에서 끝나는 것이 아니라 선반을 이동시켜야 하는 곳까지 가서 선반을 하역해야 한다. 선반을 가진 있는 로봇은 이동경로가 제한되어 복도로만 이동해야 하며, 선반이 없는 경우는 어디든 이동할 수 있다.

새로운 물건 배송 요청이 들어오면, 기존에 배달된 선반이 있을 때 다시 반환하고 요청된 물건을 배송해야 한다. 에이전트 입장에서는 어떤 선반을 어디로 이동시킬지, 기존에 있는 선반을 어느 위치에 반환할지 등을 판단하여 물건을 올바르게 배송하는 것이 도전 과제이다.

V. 결론

본고에서는 DTDE 프레임워크 기반 강화학습에 대해 소개하고, CTDE와 DTDE의 차이점, DTDE에서 발생할 수 있는 문제점을 기술하고 그에 따른 해결 방법 및 관련 알고리즘을 소개하였다. 또한, DTDE 프레임워크와 관련된 실험 환경에 대하여 살펴보았다.

DTDE 프레임워크는 CTDE 프레임워크에 비해 비효율적인 경우가 많고, 실험 환경도 다양하지 않아 환경 구성에 어려움이 있었으며, 학습 속도 저하로 인해 실험 자체도 오래 걸리는 경우가 많아 비교적 연구가 덜 되어 있었다. 그러나, EPyMARL 환경의 등장으로 DTDE 연구에 활기가 생기고 있다.

CTDE 프레임워크에서 얻을 수 있는 아이디어들이 DTDE에서 발생하는 문제점들을 해결하는 데 어느 정도 도움이 될 수 있다. 실제로, DTDE에서 발생하는 문제점에 대한 솔루션이 CTDE와 어느 정도 겹치는 경향이 있다.

CTDE 프레임워크가 더 효율적인 경우가 많음에도 불구하고 실생활에서는 개인 정보, 이해 당사자 간 보안 문제, 환경 및 지형적 요소로 인한 정보 공유 수단 부재 등 다양한 이유로 DTDE 프레임워크 사용이 강제되는 경우가 많다.

현재까지는 DTDE 연구들은 DTDE 프레임워크에서 발생하는 문제들에 대한 해답을 명쾌하게 제공하고 있지 못한 상태이다. 따라서 본고와 같이 DTDE 문제 분류를 통해 각 문제의 어려움과 그에 대한 해결방식에 대한 DTDE의 연구동향을 살펴보는 것은 중요하다고 생각된다. DTDE 프레임워크 연구는 실생활에 적용될 때 필수적으로 고려되어야 할 중요한 분야이므로, 본고에서 다룬 문제점들이

온전히 해결될 때 향후 인공지능 및 강화학습 분야에서의 핵심적인 실생활 적용 기술로써 자리 잡을 것으로 기대된다.

용어해설

MARL(Multi-Agent Reinforcement Learning) 다수의 에이전트가 존재하는 강화학습

CTDE(Centralized Training with Decentralized Execution) 에이전트 간 정보를 교류하며 학습하고 실행 시 정보 교류를 하지 않는 멀티에이전트 강화학습

DTDE(Decentralized Training with Decentralized Execution) 에이전트 간 정보 교류 없이 학습하고 실행 시에도 정보 교류를 하지 않는 멀티에이전트 강화학습

약어 정리

CTDE	Centralized Training with Decentralized Execution
DTDE	Decentralized Training with Decentralized Execution
HQL	Hysteric Q-Learning
IA2C	Independent synchronous Advantage Actor-Critic
IPPO	Independent Proximal Policy Optimization
IQL	Independent Q-Learning
LBF	Level-Based Foraging
LDQN	Lenient Deep Q-Network
MARL	Multi-Agent Reinforcement Learning
MPE	Multi-Agent Particle Environment
RWARE	Multi-RobotWAREhouse
SIPPO	Staged IPPO
SIQL	Staged IQL
SMAC	StarCraft Multi-Agent Challenge

참고문헌

[1] L. Busoniu et al., "A Comprehensive survey of multiagent reinforcement learning," IEEE Trans. Syst. Man, Cybern. C, Appl. Rev., vol. 38, no. 2, 2008, pp. 156-172.

[2] 유병현 외, "멀티 에이전트 강화학습 기술 동향," 전자통신동향분석, 제35권 제6호, 2020, pp. 137-149.

[3] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: A survey," Artif. Intell. Rev., vol. 55, no. 2, 2022, pp. 895-943.

[4] H. Nekoei et al., "Dealing with non-stationarity in decentralized cooperative multi-agent deep reinforcement learning via multi-timescale learning," arXiv preprint, CoRR, 2023, arXiv: 2302.02792.

[5] G. Papoudakis et al., "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks," in Proc. Neural Inform. Process. (Virtual), 2021.

[6] J. Foerster et al., "Counterfactual multi-agent policy gradients," Proc. AAAI Conf. Artif. Intell., vol. 32, no. 1, 2020.

[7] T. Rashid et al., "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in Proc. Int. Conf. Mach. Learn., (Stockholm, Sweden), July 2018.

[8] P. Hernandez-Leal et al., "A survey of learning in multiagent environments: Dealing with non-stationarity," arXiv preprint, CoRR, 2016, arXiv: 1707.09183.

[9] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in Proc. Int. Conf. Mach. Learn., (Amherst, MA, USA), 1993, pp. 330-337.

[10] V. Mnih et al., "Playing atari with deep reinforcement learning," arXiv preprint, CoRR, 2013, arXiv: 1312.5602.

[11] T. Chu et al., "Multi-agent deep reinforcement learning for large-scale traffic signal control," IEEE Trans. Itell. Transportation Syst., vol. 21, no. 3, 2020, pp. 1086-1095.

[12] C. Witt et al., "Is independent learning all you need in the StarCraft multi-agent challenge?," arXiv preprint, CoRR, 2020, arXiv: 2011.09533.

[13] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in Proc. Int. Conf. Mach. Learn., (New York, NY, USA), June 2016,

[14] J. Schulman et al., "Proximal policy optimization algorithms," arXiv preprint, CoRR, 2017, arXiv: 1707.06347.

[15] H. Nekoei et al., "Staged independent learning: Towards decentralized cooperative multi-agent reinforcement learning," in Proc. Int. Conf. Mach. Learn., (Virtual), Apr. 2022.

[16] L. Matignon et al., "Hysteretic q-learning: An algorithm for decentralized reinforcement learning in cooperative multi-agent teams," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., (San Diego, CA, USA), Oct. 2007, pp. 64-69.

[17] G. Palmer et al., "Lenient multi-agent deep reinforcement learning," in Proc. Int. Conf. Auton. Agents MultiAgent Syst., (Stockholm, Sweden), July 2018, pp. 443-451.

[18] J. Jiang et al., "I2Q: A fully decentralized Q-learning algorithm," in Proc. Neural Inform. Process., (New Orleans, LA, USA), Nov. 2022.

[19] S. Omidshafiei et al., "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in Proc. Int. Conf. Mach. Learn., (Sydney, Australia), Aug. 2017, pp. 2681-2690.

[20] W. Li et al., "F2A2: Flexible fully-decentralized approximate actor-critic for cooperative multi-agent reinforcement learning," arXiv preprint, CoRR, 2020, arXiv: 2004.11145.

[21] M. Samvelyan et al., "The starcraft multi-agent challenge," arXiv preprint, CoRR, 2019, arXiv: 1902.04043.

[22] G. Brockman et al., "OpenAI gym," arXiv preprint, CoRR, 2016, arXiv: 1606.01540.

[23] F. Christianos et al., "Shared experience actor-critic for multi-agent reinforcement learning," in Proc. Neural Inform. Process. Syst., (Vancouver, Canada), Dec. 2020, pp. 10707-10717.