KIPS Trans. Softw. and Data Eng.
Vol.10, No.7 pp.263~270
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2021.10.7.263

The Design and Practice of Disaster Response RL Environment Using Dimension Reduction Method for Training Performance Enhancement

Sangho Yeo† · Seungjun Lee†† · Sangyoon Oh†††

ABSTRACT

Reinforcement learning(RL) is the method to find an optimal policy through training. and it is one of popular methods for solving lifesaving and disaster response problems effectively. However, the conventional reinforcement learning method for disaster response utilizes either simple environment such as. grid and graph or a self-developed environment that are hard to verify the practical effectiveness. In this paper, we propose the design of a disaster response RL environment which utilizes the detailed property information of the disaster simulation in order to utilize the reinforcement learning method in the real world. For the RL environment, we design and build the reinforcement learning communication as well as the interface between the RL agent and the disaster simulation. Also, we apply the dimension reduction method for converting non-image feature vectors into image format which is effectively utilized with convolution layer to utilize the high-dimensional and detailed property of the disaster simulation. To verify the effectiveness of our proposed method, we conducted empirical evaluations and it shows that our proposed method outperformed conventional methods in the building fire damage.

Keywords : Reinforcement Learning Environment, Disaster Response Simulation, Dimension Reduction Method, PCA

학습 성능 향상을 위한 차원 축소 기법 기반 재난 시뮬레이션 강화학습 환경 구성 및 활용

여 상 호† · 이 승 준†† · 오 상 윤†††

요 약

강화학습은 학습을 통해 최적의 행동정책을 탐색하는 기법으로써, 재난 상황에서 효과적인 인명 구조 및 재난 대응 문제 해결을 위해 많이 활용되고 있다. 그러나, 기존 재난 대응을 위한 강화학습 기법은 상대적으로 단순한 그리드, 그래프와 같은 환경 혹은 자체 개발한 강화학습 환경을 통해 평가를 수행함에 따라 그 실용성이 충분히 검증되지 않았다. 본 논문에서는 강화학습 기법을 실세계 환경에서 사용하기 위해 기존 개발된 재난 시뮬레이션 환경의 복잡한 프로퍼티를 활용하는 강화학습 환경 구성과 활용 결과를 제시하고자 한다. 본 제안 강화학습 환경의 구성을 위하여 재난 시뮬레이션과 강화학습 에이전트 간 강화학습 커뮤니케이션 채널 및 인터페이스를 구축하였으며, 시뮬레이션 환경이 제공하는 고차원의 프로퍼티 정보의 활용을 위해 비-이미지 피쳐 벡터(non-image feature vector)에 이미지 변환방식을 적용하였다. 실험을 통해 본 제안 방식이 건물 화재 피해도를 기준으로 한 평가에서 기존 방식 대비 가장 낮은 건물 화재 피해를 기록한 것을 확인하였다.

키워드 : 강화학습 환경, 재난 대응 시뮬레이션, 차원 축소 기법, PCA

1. 서 론

강화학습은 환경에 대한 효율적인 행동정책을 학습하기 위한 학습 방법론으로써, 게임[1], 자율주행[2], 네트워크 라우팅[3], 재난 대응[4-6]과 같은 다양한 도메인에서 활용되어진다. 재난 대응 도메인 분야에서 강화학습, 특별히 심층 강화학습은 재난 상황의 상태에 대한 행동을 스스로 학습하여 최적의 행동정책을 결정하고 이에 따라 효율적인 인명 구조를 할 수 있다는 점에서 널리 활용되고 있다.

기존 강화학습을 활용한 재난 대응 방식의 경우 강화학습 에이전트가 학습되는 환경이 그리드, 그래프와 같은 단순한 환경에 제한되거나, 자체적으로 개발된 시뮬레이션 환경에서 학습하기 때문에 강화학습으로 학습된 에이전트의 실세계에서의 실용성에 대한 검증이 되지 않았다. 이에 대한 해답으로 실제 세계와 유사한, 복잡한 환경 정보를 제공할 수 있는 기존 재난 대응 시뮬레이션 환경[7, 8]은 개체의 구체적인 프로퍼티 정보를 현재

※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터 지원사업의 연구결과로 수행되었음(IITP-2020-2018-0-01431).
† 비 회 원 : 아주대학교 인공지능학과 박사과정
†† 비 회 원 : 아주대학교 인공지능학과 석사과정
††† 종신회원 : 아주대학교 소프트웨어학과 교수
Manuscript Received : December 17, 2020
Accepted : April 24, 2021
* Corresponding Author : Sangyoon Oh(syoh@ajou.ac.kr)

의 상태 정보로 제공하여 실용성에 대한 검증 문제를 해결할 수 있다. 그러나, 이들 환경이 제공하는 구체적인 프로퍼티 정보를 활용하기 위해 고차원의 피쳐 벡터(feature vector)를 신경망의 입력 데이터로써 정의해야 하며, 고차원 피쳐 벡터는 일반적인 완전 연결(fully-connected) 레이어 기반의 DNN(Deep Neural Network)으로는 학습이 어려운 문제가 있다[9].

이로 인해 기존 연구에서는 고차원의 피쳐 벡터를 신경망의 입력데이터로 활용해야하는 경우 이를 이미지 데이터로 변환하여 컨볼루션(convolution) 레이어를 통해 학습을 하는 다양한 기법들이 제안되었다[10-13]. 특히, 최근 차원 축소 기법 을 적용하여 피쳐 벡터를 이미지로 변환하는 방식[12]은 고차원의 유전정보를 이미지로 변환하여 효율적으로 학습시켰다. 그러나, 차원 축소 기법에 기반한 비-이미지 피쳐 벡터(non-image feature vector)의 이미지 변환 과정은 고차원 피쳐 벡터를 요구되는 유전자 게놈 정보의 이미지 변환 및 게놈 정보 분류에 활용되었으며, 아직 강화학습에서 활용된 사례는 없다.

본 논문에서는 재난 대응 시뮬레이션의 구체적 프로퍼티를 활용한 강화학습을 위해 1) 재난 대응 시뮬레이션과 강화학습 에이전트 간 연동과정을 정의하며, 2) 시뮬레이션에서 제공되는 개체의 프로퍼티 정보로 구성된 고차원의 피쳐 벡터를 효율적으로 활용하기 위한 정규화가 적용된 차원축소기법을 통한 이미지로의 변환 방식을 제안한다. 또한, 제안 방식의 적용을 확인하기 위하여 RCRS 시뮬레이션을 타겟 시뮬레이션으로 지정하여 연동 및 비-이미지 피쳐 벡터의 변환과정을 수행한다.

실험 결과에서 본 연구팀의 제안 방식은 기존 피쳐 벡터를 활용하는 기존 방식들 대비 가장 높은 에피소드 별 보상(reward) 점수에 도달하였으며, 시뮬레이션에서 산출된 에피소드 별 건물 피해도에서 가장 낮은 건물 피해도에 도달함을 확인하였다.

이후 논문의 구성은 다음과 같다. 2장에서는 본 연구의 관련 연구를 세 부류로 나누어 분석한다. 3장에서는 본 연구의 제안 기법인 차원 축소 기법을 활용한 재난 대응 시뮬레이션 기반 강화학습 환경 구성에 대해 설명한다. 4장에서는 본 연구의 제안 기법을 기존 연구의 방식과 비교하여 평가하였다. 5장에서는 본 연구를 기반으로 한 후속연구 계획 및 결론에 대해 설명한다.

2. 관련 연구

본 장에서는 제안 방법과 관련된 세 부류의 관련 연구에 대해 정리한다.

2.1 재난 대응을 위한 강화학습 연구

강화학습은 정의된 환경에서의 최적의 행동정책을 탐색하는 학습 방법으로써 재난 환경에서의 빠른 대피, 구조를 위해 활용되어왔다. 응우옌의 연구(2018)[4]는 강화학습 환경을 통해 홍수상황에서의 구조자와 피-구조자 간의 효율적인 매칭을 수행한다. 이 연구에서는 학습을 하는 대상은 구조자(volunteer)이며, 구조자와 피-구조자 간 거리를 휴리스틱한 알고리즘을 통해 계산한 후에 거리에 따라 구조 우선 순위를

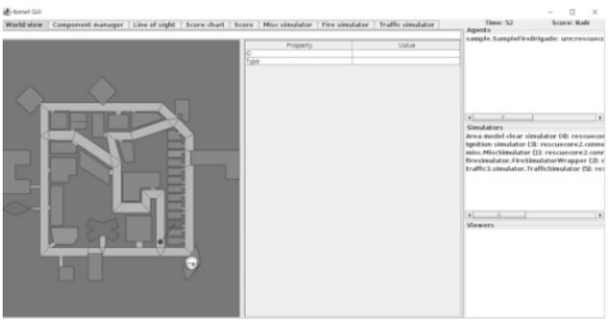


Fig. 1. RCRS(RoboCup Rescue Simulation)[7]

정의하여 구조자-피 구조자 간 매칭(즉, 액션(action))을 효율적으로 수행한다. 샤르마의 연구(2020)[5]에서는 건물을 방, 복도 간의 연결관계를 그래프로 단순화하여 표현하였으며, 다수의 에이전트(즉, 대피자)의 대피 행동 정책을 최적화하였다. 이현록의 연구(2020)[6]에서는 인천대교 버스 사고의 실제 사고 데이터를 활용한 환경에서 기존 정의된 행동 규칙을 활용하기 위해 행동 복제(behavior cloning) 기반의 효율적인 멀티-에이전트 강화학습을 정의하여 EMS(Emergency Medical Service)를 위한 행동을 학습하였다.

2.2 재난 대응을 위한 시뮬레이션 환경

앞서 설명한 강화학습을 통한 재난 대응의 행동정책은 효율적인 행동정책을 제시하였지만, 재난 대응에서 발생할 수 있는 복잡하고 확률적인 재난 환경 내 변수를 고려하지 않았다. 재난 대응을 위한 시뮬레이션 환경은 실세계의 특성을 반영하여 재난 대응에 대한 구체적이며, 실용적인 결과를 제시한다.

Fig. 1은 스키너의 연구(2010)[7]로써 실제 도시 지도에 기반한 화재 재난 대응 시뮬레이션을 제공한다. 화재 재난 대응과 관련된 네 유형의 에이전트(즉, 시민, 경찰, 소방차, 앰뷸런스)를 정의하며, 구체적으로 정의된 건물과 에이전트의 프로퍼티 및 OpenStreetMap 데이터를 시뮬레이션에 활용할 수 있는 GML 데이터로 변환할 수 있는 기능을 제공하여 실제 다양한 도시에서 일어날 수 있는 화재 재난 시나리오에 시뮬레이션을 적용할 수 있다. 코르호넨의 연구(2009)[5]는 화염 유체 역학 시뮬레이션인 FDS(Fire Dyanmics Simulator)에 연동된 탈출(evacuation) 모듈을 활용한 건물 내 화재 대피 시뮬레이션이다. 이 연구는 화염와 연기를 실제와 유사한 수학적 계산을 통해 시뮬레이션을 수행하는 특징을 가지고 있다.

2.3 비-이미지 피쳐 벡터에서 이미지로의 변환을 통한 심층신경망 학습 연구

2.2의 시뮬레이션 환경들은 실세계와 유사한 재난 시뮬레이션을 제공함에도 불구하고, 2.1의 연구들에는 활용되지 않았다. 이는 시뮬레이션 환경이 제공하는 다양한 재난 프로퍼티를 학습 활용을 위한 입력 데이터로 변환하기 어려운 문제가 주된 원인으로 판단된다.

샤르마의 연구(2020)[12]에서는 고차원의 피쳐 벡터를 학습에 효율적으로 활용하기 위하여 차원 축소 기법(즉, PCA,

tSNE)을 활용하여 이미지로 변환하였다. 이러한 피쳐 벡터의 이미지 변환을 통해 신경망은 더 깊게 레이어를 적층할 수 있으며, 이를 통해 기존의 완전 연결 레이어를 활용할 때보다 정확한 판단을 수행할 수 있다.

또한, 이러한 차원 축소 기법을 기반으로 한 방법 이외에 표형식의 피쳐 정보를 이미지로 변환하기 위해 표의 각 행을 이미지 커널로 활용하여 방식[11], 피쳐 벡터를 막대 그래프, 피쳐 간의 유클리디안 거리를 표현하는 이미지로 변환하는 방식[13], 고차원의 피쳐 벡터를 두 개씩 짝을 지은 후, 해당 짝을 2D 좌표축에 표현하여 이미지화 하는 방식[10] 등이 제안되었다.

3. 차원 축소 기법을 활용한 재난 대응 시뮬레이션 기반 강화학습 환경 구성

실 세계에서 강화학습으로 학습된 재난 대응 모델의 효율적인 활용을 위해 강화학습 환경은 기존 재난 대응 시뮬레이션과 같이 실제 세계의 구체적 특성을 반영할 수 있어야 한다.

이를 위해 본 연구팀은 RCRS 시뮬레이션[7]을 타겟 재난 대응 시뮬레이션으로 정하여 강화학습 커뮤니케이션 채널 정의 및 시뮬레이션 환경-강화학습 에이전트 간 인터페이스를 구성을 통해 시뮬레이션 환경을 강화학습 에이전트와 연동하며, 연동된 시뮬레이션 환경이 제공하는 매 스텝 별 피쳐 정보를 효율적으로 활용하기 위해 정규화 및 PCA 기법에 기반한 비-이미지 피쳐 벡터의 이미지 변환 과정을 정의한다.

3.1 재난 대응 강화학습 환경 구성을 위한 시뮬레이션 연동

재난 대응 시뮬레이션 환경을 강화학습의 환경으로써 활용하기 위해 재난 대응 시뮬레이션 환경과 강화학습 에이전트 간 연동이 필요하다. 연동 과정의 정의를 위해 본 연구팀은 1) 강화학습 에이전트 - 시뮬레이션 환경 간 강화학습 커뮤니케이션 채널 정의, 그리고 2) 정의된 커뮤니케이션 채널에 기반한 강화학습 에이전트 – 시뮬레이션 환경 간 연동 인터페이스 구성하고자 한다.

강화학습의 커뮤니케이션 채널은 상태(state), 액션(action), 보상(reward)로 나뉘어진다. 이 중 상태는 시뮬레이션으로부터 받은 변경정보(RCRS 시뮬레이션에서 changeset으로 정의됨)로부터 받은 정보를 활용하여, 개체들의 정보를 표현하도록 구성하였다. RCRS 시뮬레이션의 경우, 개체의 유형은 크게 건물 개체와 에이전트 개체(강화학습 에이전트가 아닌 시뮬레이션 상에서 행동을 수행하는 개체임)로 나눠지며, 각 유형의 개체는 각각 다른 차원의 피쳐를 통해 표현된다. 그러나, 피쳐 정보를 신경망의 입력값으로 활용하기 위해서는 모든 개체를 동일한 차원의 피쳐 벡터로 정의해야 하며, 이를 위해 건물을 표현하는 피쳐 벡터들과 에이전트를 표현하는 피쳐 벡터들을 통합하여 개체의 피쳐 벡터를 표현하였다.

Table 1은 개체에 대한 피쳐 벡터들의 정의이다. 앞서 설명한 바와 같이 개체의 피쳐 벡터들은 에이전트를 정의하는 피쳐(damage, hp, stamina, water quantity)와 건물을 정의하는 피쳐(fireyness, brokeness, temperature)를 모두 포함한다. 개체는 항상 건물 유형 혹은 에이전트 유형이기 때문에, 건물일 경우, 에이전트의 피쳐 정의는 불필요하며, 에이전트 개체일 경우, 건물 피쳐의 정의는 불필요하다. 따라서, 개체의 유형에 따라 불필요한 피쳐 벡터의 경우 시뮬레이션을 통해 정의될 수 없는 피쳐 값인 -1을 정의하여 각 개체의 유형이 구분될 수 있도록 하였다. 예외로 개체의 위치를 정의하는 x, y 좌표 위치 값은 건물 개체와 에이전트 개체가 모두 보유한 피쳐 정보로써 개체 유형에 관계없이 정의된다.

Table 1. The Definition of Feature Vector

Feature	Description
x	X coordinate on the map
y	Y coordinate on the map
damage	Loss of health point at each step
hp	Health point of the agent if health point is zero, agent is dead
waterquantity	Amount of water which is carried by the agent(i.e, fire brigade)
fieryness	Degree of fire damage in the building
brokeness	Structural damage to the building before fire disaster is happened
temperature	Temperature of the building. if temperature drops below certain threshold, the fire is extinguished

액션의 경우 기존 고얄의 연구(2020)[14]와 동일한 방식인 시뮬레이션 환경 내 화재 진압을 수행할 수 있는 건물들을 액션 공간(action space)로 정의하였다. 보상은 신속한 학습을 위하여 RCRS 시뮬레이션 내 전체 건물의 피해량의 변화에 기반한 부정적 보상과 함께, 에이전트의 행동에 따른 긍정적인 보상을 합하여 정의하였다.

$$reward_{agent} = \begin{cases} 1.0 \text{ if} \begin{cases} agent's\, water \geq water_{\min} \\ \text{and} \\ building.isOnFire \end{cases} \\ 0.1 \; else \text{ if} \begin{cases} agent's\, water < water_{\min} \\ \text{and} \\ agent\, position \equiv refuge \end{cases} \end{cases} \quad (1)$$

Equation (1)은 에이전트의 행동에 따른 보상을 정의하기 위한 수식이다. 에이전트의 보상의 경우, 에이전트가 충분한 양의 물을 보유하고 있을 때, 화재가 발생한 빌딩의 화재를 진압하는 경우 1의 긍정적 보상가 부여된다. 또한, 에이전트가 충분한 양의 물이 보유하고 있지 않을 때, 시뮬레이션 상에서 물을 보충해주는 역할을 하는 건물인 피난처(refuge)로 위치를 변경하는 경우 화재 진압 대비 상대적으로 적은 크기의 긍정적 보상인 0.1을 부여하도록 정의하였다.

$$reward_{\sim ulator} = previousBuildingDamage - currentBuildingDamage \quad (2)$$

Equation (2)는 시뮬레이션 환경에서 화재로 인한 건물 피해량의 변화를 보상에 반영하기 위해 활용된다. 이를 위해 기존 RCRS 시뮬레이션에서 건물의 피해량을 산출하기 위해 사용되는 점수기능을 이용한다. 시뮬레이션 상의 점수 기능을 사용하여 이전 스텝의 건물의 화재 피해량과 현재 스텝의 건물의 화재 피해량을 비교하여 이전 스텝 대비 증가한 화재 피해량을 부정적 보상으로 활용한다.

$$reward = reward_{agent} + reward_{\sim ulator} \times \alpha \quad (3)$$

Equation (1)과 (2)에서 계산된 보상을 활용하여 equation (3)을 통해 최종적인 보상을 계산한다. equation (1)과 (2)에서 계산된 $reward_{agent}$와 $reward_{\sim ulator}$를 더할 때, $reward_{\sim ulator}$의 부정적인 보상이 $reward_{agent}$의 긍정적인 보상의 비해 상대적으로 낮기 때문에, 상대적인 보상의 차이를 조절하기 위한 하이퍼파라미터 값인 α를 설정한다.

앞서 정의된 강화학습 커뮤니케이션 채널을 활용하여 RCRS 시뮬레이션 환경과 강화학습 에이전트 간 연동을 위한 인터페이스를 구성하여야 한다. 이를 위해 본 연구팀은 시뮬레이션 환경의 모든 정보를 중개하고 다루는 커널 클래스와 강화학습 에이전트 간에 Fig. 2와 같이 인터페이스를 정의하였다.

인터페이스 구성을 위해 RCRS의 시뮬레이션 환경 중 일부분(하늘색 부분)이 변경된다. 또한, 기존에 널리 활용되는 강화학습 환경 인터페이스인 OpenAI Gym[15]을 참고하여 정의된 step 메소드를 통해 강화학습 에이전트는 시뮬레이션 환경으로부터 상태와 보상 정보를 받을 수 있으며, 시뮬레이션 환경은 강화학습 에이전트로부터 현재 상태에 대한 선택된 액션을 전달받을 수 있다.

3.2 차원 축소 기법 기반의 비-이미지 피쳐 벡터의 이미지 변환

고차원의 피쳐 벡터로 구성된 입력데이터를 활용해야 하는 경우, 완전 연결 레이어로 구성된 DNN을 활용한 학습은 어렵다. 이는 고차원의 피쳐 벡터를 입력데이터로 활용할 때, 입력데이터를 받기 위해 완전 연결 레이어의 파라미터의 수가 증가되며. 파라미터의 수가 증가함에 따라 학습 시간도 늘어나기 때문이다.

따라서, 고차원의 피쳐 벡터를 학습을 효율적으로 하기 위해 피쳐 벡터를 이미지 형식의 데이터로 변환해야한다. 이미지 데이터 형식으로 변환하는 경우, 컨볼루션 레이어를 통해 효율적으로 레이어의 깊이를 늘릴 수 있으며 완전 연결 레이어 비해 입력 데이터의 크기가 증가하더라도 적은 파라미터 수의 증가로 입력데이터를 학습시킬 수 있다. 따라서, 크기가 큰 입력데이터를 비교적 적은 수의 파라미터로 학습 시킬 수 있기 때문에 학습의 효율성도 증가된다.

그러나, 강화학습에서 환경과의 소통에서 발생하는 부하는 학습 시간을 늘리는 주요 요소이므로, 환경과의 소통과정에서 추가적인 부하를 야기하는 피쳐 벡터의 이미지 변환과정은 연산 부하가 작아야 하며, 효과적으로 피쳐 벡터를 이미지로 표현할 수 있어야 한다.

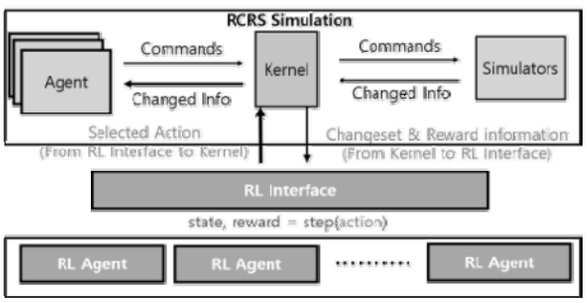
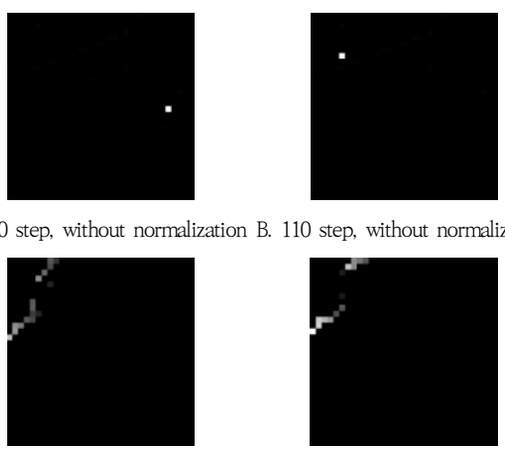


Fig. 2. Overview of Intergration between RCRS Simulation Environment and RL Agent



A. 100 step, without normalization B. 110 step, without normalization

C. 100 step, with normalization D. 110 step, with normalization

Fig. 3. Comparing Transformed Image With/without Normalization

따라서 본 연구팀은 모델의 학습 과정과 학습된 모델의 활용 과정을 나눌 수 있는 PCA 기반의 차원 축소 기법을 사용하였다. 또한, 각 피쳐 간에 존재할 수 있는 비선형적인 연관관계를 분석하기 위하여 PCA 기법 중 kPCA(kernel PCA)[16]를 활용한 차원 축소 기법을 활용하였다.

그러나, 차원 축소 기법의 적용에서 피쳐 별로 최소값과 최대값 사이의 범위에 대하여 그 편차가 매우 크며, kPCA 기반의 차원 축소에서는 최소값과 최대값의 차이가 큰 특정 피쳐 값에 의해서 축소된 벡터 값이 결정되기 때문에, 차원 축소 기법만으로는 효율적인 이미지 변환을 수행할 수 없다. 즉, 이 과정을 통해 축소된 벡터 값은 최소값과 최대값의 차이가 적은 피쳐 값의 변화를 제대로 반영하지 못한다.

Fig. 3A와 3B는 피쳐 벡터들 중 상대적으로 최소값과 최대값의 차이가 큰 피쳐 벡터가 있을 때 발생하는 문제에 따른 실제 비-이미지 피쳐 벡터들의 변환된 이미지를 나타낸다. RCRS 시뮬레이션에서 최소, 최대값의 크기 차이가 가장 큰 피쳐는 개체의 위치를 나타내는 x축 값과 y축 값이다. 이때, 건물의 경우 항상 동일한 x축과 y축의 값을 지니게 되어, 차원 축소 시 이동할 수 있는 에이전트의 x축 및 y축의 값이 주요한 영향을 미치게 된다. 이는 결과적으로 Fig. 3A, B에서 에이전트의 x축 및 y축의 변화만이 반영된 하나의 흰색점으로 나타나게 된다.

이 문제를 해결하기 위해 상태 표현(state representation)

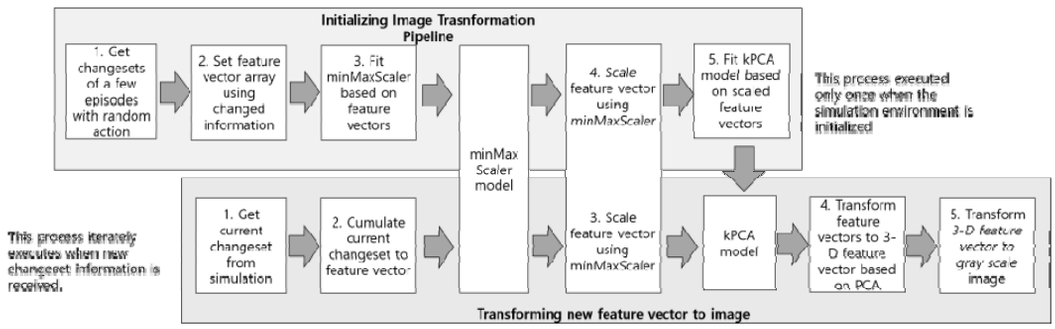


Fig. 4. Image Transformation Pipeline for RL Agent

의 변환과정에서 추가로 피쳐 값의 최소값과 최대값의 차이를 동일하게 맞추는 정규화 과정이 요구된다. 이를 위해 kPCA 기법을 통해 차원 축소를 수행하기 전에 minMaxScaler[17]를 활용하여 피쳐의 값을 0~1 사이로 정규화하였다.

Fig. 3C와 Fig. 3D는 앞서 설명한 정규화 과정을 거친 피쳐 벡터들을 kPCA 모델로 변환시킨 이미지이다. 앞서 설명한 Fig. 3A와 Fig. 3B와 달리 다수의 점의 위치 및 색깔이 10 스텝 동안 변화했음을 알 수 있다. 다만, 이러한 점들이 한쪽으로 치우친 문제가 확인되었으며, 이는 minMaxScaler의 정규화 과정으로 생성되는 피쳐 벡터 중 분포가 0~1 사이에서 특정 값에 분포가 집중되기 때문인 것으로 가정된다. 이 문제는 학습의 성능을 저하시키는 요인이 될 수 있어 피쳐 벡터의 분포를 0~1 사이에 균등하게 나눌 수 있는 정규화 기법의 적용이 필요할 것으로 예상된다.

Fig. 4는 강화학습 수행을 위한 피쳐 벡터의 이미지 변환 파이프라인을 나타낸다. 본 그림은 위쪽의 파이프라인의 초기화 과정과 아래쪽의 학습된 모델을 활용한 피쳐 벡터 값의 이미지 변환 파이프라인으로 나뉘어진다. 해당 변환과정은 샤르마의 연구[12]에서 제시한 파이프라인 과정과 유사하나, 본 연구팀은 변환과정의 부하를 줄이기 위해 PCA를 활용한 이미지 변환과정만을 수행하였다.

Fig. 4의 윗부분에 정의된 파이프라인 초기화 과정은 RCRS 시뮬레이션 기반 강화학습 환경을 초기화할 때 구성된다. 초기화 단계에서 kPCA 모델과 minMaxScaler를 위한 각 피쳐의 최소값과 최대값을 얻기 위하여 임의의 액션을 수행하는 소수의 에피소드들의 변경 정보를 기반으로 피쳐 벡터의 배열을 구성하였다. 이 배열을 기반으로 minMaxScaler의 모델(피쳐 별 최소, 최대값)을 파악하였으며, minMaxScaler를 통해 조정된 피쳐 벡터의 값들을 바탕으로 kPCA 모델을 학습시켰다.

Fig. 4의 아랫부분의 학습과정에서는 앞서 설명한 초기화 과정에서 생성된 minMaxScaler와 kPCA 모델을 활용하여 정규화된 피쳐 벡터 값의 차원을 축소시킨다. 이 과정은 이미 학습된 모델을 통해 수행되며 추가적인 학습 과정이 불필요하므로 변환과정에서 작은 부하를 야기한다. 이후, kPCA를 통해 피쳐 벡터들의 값은 3차원 벡터들로 표현된다. 이때, 두 개의 차원은 이미지 내에 해당 벡터가 표시될 위치를 나타내며, 한 개의 차원은 그레이스케일(gray scale)로 나타낼 때의 색깔을 정의한다.

4. 실 험

4.1 실험 환경

본 연구팀이 제안하는 비-이미지 피쳐 벡터의 이미지 변환을 기반으로 구성된 재난 시뮬레이션 강화학습 환경을 평가하기 위해 키스티의 뉴런 슈퍼컴퓨팅 환경[18]을 활용하였다. 키스티의 뉴런 슈퍼컴퓨팅 환경은 고성능의 GPU 클러스터 환경으로써 본 연구팀은 Table 2의 하드웨어 성능을 지닌 키스티 뉴런 클러스터의 단일 GPU 노드에서 학습을 수행하였다.

또한, 심층 강화학습 에이전트의 모델을 정의하고, kPCA 기반의 차원 축소 기법을 활용하기 위하여 Table 3과 같은 딥러닝 프레임워크 및 머신러닝 라이브러리를 활용하였으며, 제시된 라이브러리와 프레임워크를 슈퍼컴퓨팅 환경에서 활용하기 위하여 Singularity[19] 기반의 이미지를 구성하였다.

본 제안 방식의 비-이미지 피쳐 벡터를 위한 이미지 변환 과정을 평가하기 위하여 Table 4와 같이 비교군을 설정하여 기존 방법과의 비교 실험을 진행하였다.

첫 번째 방법론은 고얄의 방법론 (2020) [14]으로써, 완전 연결 레이어 기반의 DNN에서 효과적으로 학습할 수 있는 소수의 피쳐 벡터를 선택하는 방식이다. 다만 첫 번째 방법론의 구현 시, [14]의 연구에서 피쳐 벡터로 정의되었던 에이전트의 액션 수행 여부는 시뮬레이션에서 기본적으로 제공되는 정보가 아니므로 제외시켰다.

두 번째 방법론인 large feature vector는 피쳐 벡터를 이미지 변환없이 그대로 학습하는 방법이며, 세 번째인 image

Table 2. Hardware Configuration

Type	Specification
CPU	Intel Xeon Cascade Lake(Gold 6226R)
RAM	384GB DDR4
GPU	Nvidia V100

Table 3. Software Configuration

Type	Specification
DL Framework	PyTorch[20] 1.4
ML Library	scikit-learn 0.23.2 [21]
Container Framework	Singularity v3.1.0

Table 4. Compare Group for Evaluation

Name	Description
Small feature vector	Feature vector which applied the feacture selection of [14]
Large feature vector	Feature vector as shown in table 1
Image	Image which obtained by converting the feature vector of table 1
Greedy	Algorithm which decide the fire extinguish priority of building by greedy search strategy

Table 5. Configuration of RL Agent and Environment

Type	Description
DRL Algorithm	DQN[1]
Replay buffer size	1,000,000
Frame history length	4
Learning frequency	4
Optimizer	RMSProp[22](learning rate=0.0001, alpha=0.95, eps=0.01, weight_decay=10^{-5})
Total training step	2,100,000(300 step per episode 7000 episode)
Learning start step	1000
Epsilon decay rate scheduling	Linear decay from 1 to 0.1 during 10000 steps
The number of building	37
The number of agent	1
The number of refuge	1

는 본 연구팀이 제안한 이미지 변환 방식이다. 마지막으로 greedy는 [14]의 연구에서 비교군으로 설정한 그리디(greedy) 검색 알고리즘으로써, 에이전트-건물 간의 거리, 건물의 화재 레벨 및 건물에 인접한 다른 건물 수를 기반으로 화재 진압 우선순위를 설정하는 방법론이다.

또한, 본 강화학습 에이전트의 학습을 위하여 Table 5와 같은 강화학습 에이전트 및 시뮬레이션 환경을 설정하였다. 하이퍼파라미터 설정의 경우, 기존 아타리 게임 환경에서의학습 하이퍼파라미터 설정[1]과 유사하다. 하지만, 시뮬레이션의 종료 및 재시작을 위하여 에피소드 당 300 스텝을 설정하였으며, 매 에피소드마다 고정된 건물에서 시작된 화재가 제한된 스텝 동안 전파되는 양상이 시뮬레이션되며, 이 과정에서 시뮬레이션을 통해 전달받은 건물피해도 정보를 통해 매 스텝에 대한 보상 및 건물 피해도 정보를 구성하여 각 에피소드 별 점수를 산정한다. 또한, 시뮬레이션 환경 내에는 건물이 37개, 에이전트(본 시뮬레이션 환경에서는 화재를 진압하는 소방차를 의미함)가 하나 있는 작은 규모의 시뮬레이션 환경에서 학습을 수행하였다.

4.2 실험 결과

본 제안 방식의 보상 값을 통해 정의되는 에피소드 별 점수(즉, 에피소드 별 보상의 합)는 본 연구팀이 신속한 학습을 위해 정의한 equation (1)의 보상 값이 포함되었다. 따라서, 에피소드 별 점수는 건물의 피해도를 평가하는데 정확한 지표가 될 수 없다. 따라서, 본 실험 결과는 제안 강화학습 환경의 학습 효율성의 판단을 위한 1) 에피소드 별 점수 비교와 실제 학습된 모델의 화재 대응력을 판단하는 2) 에피소드 별 화재에 대한 건물 피해도 비교로 수행하고자 한다.

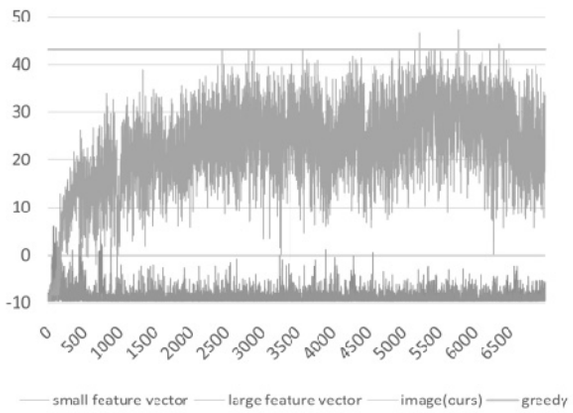


Fig. 5. On-line Scores of the Compare Groups

1) 에피소드 별 점수 비교

본 연구팀이 정의한 보상에 기반한 에피소드 별 점수 비교는 본 연구팀이 구성한 강화학습 환경에서 본 제안 방식이 기존 방식 대비 얼마나 효율적으로 학습을 하였는지 확인할 수 있다.

Fig. 5의 그래프는 제안 방식 및 비교군의 학습 그래프이다. 그래프에서 확인할 수 있듯이 피쳐 벡터를 그대로 활용하는 두 방식의 경우, 학습이 진행됨에 따라 점수의 상승이 거의 일어나지 않음을 확인할 수 있다. 다만, smalll feature vector의 경우, 학습 초반에 점수의 상승이 유의미하게 발생되었으나, 과적합(overfitting)에 의해 점수가 다시 감소하는 경향을 확인할 수 있다.

반면에 제안 방식인 image의 경우, 학습이 진행됨에 따라 점수의 상승이 일어났으며, 기존의 피쳐 벡터를 활용하는 강화학습 대비 효율적인 학습을 수행하였다. 다만, greedy와 점수와 비교할 경우, 평균적인 에피소드의 점수가 greedy의 점수에 미치지 못하는 결과를 확인할 수 있었다. 또한, 학습 후반에 갈수록 과적합에 의해 에피소드 별 점수가 감소하는 경향을 확인할 수 있다.

Table 6은 학습과정에서 도달한 에피소드 별 최고 점수를 기록한 표이다. 앞서 학습 그래프에서 제안 기법이 greedy 방식 대비 낮은 평균점수를 기록한 것과 달리, 에피소드 별 최고 점수의 비교에서, 본 제안 방식은 가장 높은 최고 점수에 도달하였다.

각 비교군을 분석하자면, small feature vector는 학습 초반에 유의미한 학습으로 인해 large feature vector를 활용한 강화학습 대비 높은 최고 점수에 도달하였다. 그러나, 피쳐 벡터를 활용한 두 기법 모두 제안 기법 및 greedy 방식 대비 상대적으로 도달한 최대 점수가 낮다.

Table 6. Maximum Score Comparison

Compared Group.	Maximum Score
Small feature vector	10.208
Large feature vector	0.582
Image(Ours)	**47.218**
Greedy	43.184

제안 기법과 greedy 방식과 비교 시, 제안 기법이 특정 에피소드의 학습에서 greedy 기법을 상회하는 점수에 도달함을 확인하였다. 이는 앞서 확인한 학습 그래프에서 볼 수 있듯이 평균적인 양상은 아니지만, 본 제안의 이미지 변환 기법 및 활용한 강화학습에 있어 다른 기법의 적용을 통해 최적화를 추가적으로 수행할 수 있기 때문에, 이후, 결론 및 향후 계획 장에서 설명하는 차후 연구에서 greedy 알고리즘을 상회하는 강화학습 모델을 제안할 수 있을 것으로 기대한다.

2) 에피소드 별 건물 화재 피해도 비교

건물 화재 피해도 비교를 통해 본 연구팀이 구성한 강화학습 모델이 얼마나 효율적으로 건물의 화재 피해에 대응하였는지 확인할 수 있다. Table 7은 에피소드 별 화재에 의해 발생한 건물 피해를 비교한 결과이다. 비교 결과, 본 제안 방식은 다른 기존 기법 대비 가장 낮은 건물 피해에 도달하였다.

건물 피해도의 비교에서 본 제안 방식이 기존 greedy 방식 대비 낮은 건물 피해도를 기록한 원인은 두 방식 간의 행동정책의 차이로 인하여 발생한다. greedy 방식의 경우, 건물과 에이전트 간의 거리 및 화재 피해도를 판단하여 화재 진압을 수행하는 반면, 강화학습 기반의 본 제안 방식은 가장 긍정적 보상을 얻을 수 있는 방향으로 행동정책이 학습된다. 따라서, 강화학습을 통해 학습된 행동정책의 경우 물을 공급받을 수 있는 피난처 근처의 건물들을 주기적으로 소화하는 양상을 보이는 반면, greedy 방식은 이와 관계없이 건물과 에이전트 간의 거리 및 건물의 화재 피해에 기반하여 화재를 진압할 건물을 선택하게 된다.

결과적으로 전체 건물들의 화재 피해도를 비교하게 될 때, 본 제안 방식의 강화학습 기법이 학습한 행동정책이 건물 피해도를 낮추는데 있어 가장 효율적임을 확인할 수 있다. 다만, 제안 기법을 통해 학습된 행동정책의 경우, 단일 에이전트가 화재 진압을 할 때 유효한 방법론이며, 소방차가 다루어야 할 지역이 더 넓은 실제 규모의 환경에서는 효율적이지 못한 방법론일 수 있다. 따라서, 다수의 에이전트 및 도시 규모의 환경에서 본 제안방법론을 재평가가 요구된다.

5. 결론 및 향후 계획

본 논문이 제안하는 비-이미지 피쳐 벡터의 이미지 변환을 활용한 재난 대응 시뮬레이션 강화학습 환경은 기존 기법을 활용한 환경 대비 효율적인 학습을 가능하게 하였다. 이로 인해 건물 화재 피해도를 기반으로 한 평가에서 제안 방식은 가장 낮은 건물 화재 피해도를 기록하였다.

Table 7. Total Building Damage Comparison

Compared Group.	Total Building Damage
Small feature vector	0.9511
Large feature vector	0.9578
Image(Ours)	**0.9250**
Greedy	0.9416

본 연구의 최종목표는 실제 도시 지도 데이터에 기반한 화재 재난의 취약점 분석이다. 이 목표에 도달하기 위해 본 논문의 제안 기법은 다양한 사항들에 대해 더 발전시켜야 한다. 본 연구팀은 이 사항들에 대해 후속 연구를 통해 본 논문의 제안보다 더 높은 성능을 달성하고자 한다.

a) 효율적인 DRL 모델 활용: 본 연구팀이 활용한 DQN 모델의 경우, 심층 강화 학습의 대표적인 모델이다. 그러나, DQN 알고리즘 이후 제안된 심층강화학습 모델들은 다양한 환경에서 DQN 모델을 상회하는 높은 학습 성능을 보여주었다. 따라서, 기존 제시된 효율적인 심층강화학습 모델의 적용이 요구된다.

b) 멀티 에이전트 강화학습: 본 연구팀의 학습환경은 단일 에이전트로 구성되어있어 다수의 에이전트가 재난 대응을 하는 실제 환경과 매우 다르다. 따라서, 실제 환경과 유사한 학습을 수행하기 위해 다수의 에이전트들을 효율적으로 학습시킬 수 있는 기법의 연구가 필요하다.

c) 효율적인 차원 축소 알고리즘 및 정규화 알고리즘 적용: 차원 축소 알고리즘의 경우 kPCA 대비 더 효율적으로 차원 축소를 수행할 수 있는 신규 알고리즘이 제안되었다. 또한, 본 연구에서 활용한 minMaxScaler를 활용한 정규화는 정규화된 벡터가 균등하게 분포되지 않는 문제가 있었다. 이 문제의 해결을 위한 후속 연구가 필요하다.

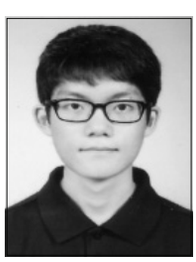
d) 도시-규모의 재난 대응 및 이를 위한 최적화: 본 논문의 실험은 소규모 환경에서 수행되었다. 실용적인 모델의 활용을 위해선 이보다 훨씬 큰 도시-규모의 맵 환경에서의 학습이 필요하다. 또한 이러한 도시 규모의 맵을 구성하기 위해 기존 한국 도시의 맵 데이터를 GML 데이터로 변환시켜야 하며, 액션 공간를 연속 공간(continuous space)에 매핑하는 방법이 추가적으로 필요하다. 또한, 도시 규모의 환경에서는 시뮬레이션의 부하가 더 커지므로, GPU 기반의 차원 축소 기법의 적용 및 POMDP(Partial Observable Markov Decision Process) 기법의 적용하여 상태의 크기를 제한하는 방법 등이 필요하다.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” arXiv preprint arXiv: 1312.5602, 2013.

[2] B. R. Kiran, I, Sobh, V. Talpaert., P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," arXiv preprint arXiv:2002.00444, 2020.
[3] J. Boyan and M. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," *Advances in Neural Information Processing Systems*, pp.671-678, 1994.
[4] L. Nguyen, Z. Yang, J. Zhu, J. Li, and F. Jin, "Coordinating disaster emergency response with heuristic reinforcement learning," arXiv preprint arXiv:1811.05010, 2018
[5] J. Sharma, P. A. Andersen, O. C. Granmo, and M. Goodwin, "Deep Q-Learning with Q-Matrix transfer learning for novel fire evacuation environment," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.
[6] H. R. Lee and T. Lee, "Multi-agent reinforcement learning algorithm to solve a partially-observable multi-agent problem in disaster response," *European Journal of Operational Research*, Vol.291, No.1, pp.296-308, 2021.
[7] C. Skinner and S. Ramchurn, "The robocup rescue simulation platform," In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pp.1647-1648, May 2010.
[8] T. Korhonen and S. Hostikka, "Fire dynamics simulator with evacuation: Fds+ Evac: Technical reference and user's guide," 2009.
[9] P. I. Wójcik and M. Kurdziel, "Training neural networks on high-dimensional data using random projection," *Pattern Analysis and Applications*, Vol.22, No.3, pp.1221-1231. 2019.
[10] B. Kovalerchu, B. Agarwal, and D. C. Kalla, "Solving Nonimage Learning Problems by Mapping to Images," *International Conference Information Visualization*, pp.264-269, 2020.
[11] L. Buturovic and D. Miljkovic, "A novel method for classification of tabular data using convolutional neural networks," *BioRxiv*, 2020.
[12] A. Sharma, E. Vans, D. Shigemizu, K. A. Boroevich, and T. Tsunoda, "DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture," *Scientific Reports*, Vol.9, No.1, pp.1-7, 2019.
[13] A. Sharma and D. Kumar, "Non-image data classification with convolutional neural networks," arXiv preprint arXiv: 2007.03218, 2020.
[14] A. Goyal, "Multi-agent deep reinforcement learning for robocup rescue simulator," The Graduate School of The University of Texas at Austin, May 2020.
[15] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," arXiv preprint arXiv:1606.01540, 2016.
[16] S. Mika, B. Schölkopf, A. Smola, K. R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," *Advances in Neural Information Processing Systems*, Vol.11, pp.536-542, 1998.
[17] scikit-learn, sklearn.preprocessing.MinMaxScaler [Internet], https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[18] KISTI, NEURON computing environment [Internet], https://www.ksc.re.kr/mobile/ggspcpt/neuron
[19] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PloS one*, Vol.12, No.5, e0177459, 2017.
[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, and A. Desmaison, "Pytorch: An imperative style, high-performance deep learning library," In *Advances in Neural Information Processing Systems*, pp.8026-8037, 2019.
[21] O. Kramer, "Scikit-learn," In *Machine Learning for Evolution Strategies*, pp.45-53. Springer, Cham, 2016.

여 상 호
https://orcid.org/0000-0002-9194-7552
e-mail : soboru963@ajou.ac.kr
2017년 아주대학교 소프트웨어학과(학사)
2017년 ~ 현 재 아주대학교 인공지능학과 박사과정
관심분야 : 분산딥러닝, 강화학습



이 승 준
https://orcid.org/0000-0003-0385-0724
e-mail : henry174@ajou.ac.kr
2021년 아주대학교 소프트웨어 및 컴퓨터공학전공(학사)
2021년 ~ 현 재 아주대학교 인공지능학과 석사과정
관심분야 : 클라우드 컴퓨팅

오 상 윤
https://orcid.org/0000-0001-5854-149X
e-mail : syoh@ajou.ac.kr
2006년 미국 인디애나대학교 컴퓨터공학과(박사)
2006년 ~ 2007년 SK텔레콤 전략기술부문
2007년 ~ 현 재 아주대학교 소프트웨어학과 교수
관심분야 : 분산딥러닝, 고성능컴퓨팅, 빅데이터 처리