Human-Feedback 및 행동 복제를 활용한 심층강화학습 알고리즘의 학습 성능 개선 연구
착수보고서

팀명: 강아지도학습

201924504 심영찬
201624420 김동건
202025221 오현식

목차

1. 과제 배경 및 목적

1.1. 과제 배경

최근 자율주행 자동차, 스마트 팩토리, 이족보행 로봇 등 다양한 분야에서 인공지능의 중요성이 점점 더 커지고 있다. [1] 그중에서도 강화학습 알고리즘은 가장 널리 활용되는 대표적인 방법의 하나이다. 강화학습이란, 인공지능 모델이 특정 알고리즘을 통해 주어진 과제를 해결하는 방법을 스스로 학습해 나가는 과정을 의미한다.

지금까지 다양한 고성능 심층강화학습 알고리즘들이 개발되어 활용되고 있지만 여전히 여러 단점이 존재한다. 그중 가장 치명적인 단점은 강화학습 기반의 인공지능 모델이 온전히 제 기능을 할 수 있는 만큼 학습되기까지 매우 많은 시간이 소모된다는 점이다. [2]

강화학습은 지도학습과는 달리 정답을 주지 않은 상태에서 시행착오를 통해 학습이 이루어진다. 초기 단계에서 모델은 다양한 예외 상황을 마주치며 실패를 경험하고, 그 경험을 바탕으로 점차 더 “올바른 선택”을 할 수 있게 된다. 이러한 시행착오적 접근은 환경과의 상호작용을 통해 스스로 학습해 나가는 강화학습의 특징이다.

그런데 딥러닝 기반의 모델들은 기본적으로 방대한 데이터와 반복 학습을 해야 하므로 학습에 오랜 시간이 소요되며, 강화학습 역시 마찬가지로 학습 시간이 길다는 공통적인 한계를 갖는다. 이처럼 심층 강화학습의 두 방식 모두 시간 소모가 크다는 사전 조사를 통해, 학습 시간을 단축할 방법의 필요성을 확인할 수 있었다.

이에 따라, 본 과제에서는 강화학습중인 모델에게도 지도학습처럼 “올바른 선택”을 함께 제시하는 것으로 이 문제를 해결하려고 한다. 학습중인 모델이 특정 상황을 마주쳤을 때 사용자가 실시간으로 피드백을 제공하거나, 사전에 정의된 “올바른 선택”을 제공함으로써 모델이 단축된 시간으로 학습하여 더 향상된 성능을 보이도록 한다.

1.2. 과제 목표

본 과제의 목표는 스스로 학습해 나가는 강화학습 기반 인공지능 모델이 사용자로부터 피드백을 받아 학습 시간을 단축할 수 있도록, 기존보다 효율적인 방식으로 알고리즘을 개선하고 이를 연구하는 것이다. 사용자의 실시간 피드백을 긍정/부정으로 구분하고 이를 통해 인공지능이 다양한 상황에서 보다 빠르게 “올바른 선택”을 학습할 수 있도록 유도함으로써, 전체 학습 효율을 높이는 것을 지향한다.

기존 학습 방법과 비교해 짧은 시간과 적은 자원을 사용하면서 성능은 유지하는 모델을 생성하는 사용자 실시간 피드백, 행동 복제를 적용한 학습 알고리즘을 개발한다.

구현된 새로운 알고리즘을 유니티 환경에 안정적으로 이식하고, ML-Agents를 gym으로 랩핑하여[3] 새로운 시뮬레이션 환경에서도 원활하게 동작하도록 한다.

사용자가 시뮬레이션 환경에서 모델에게 실시간으로 피드백을 쉽게 제공할 수 있도록 한다. 또한, 사용자 친화적인 인터페이스를 통해 필요한 모델 정보를 안정적이고 효과적으로 전달한다.

2. 요구 사항 분석

2.1 기능적 요구사항

2.1.1. LLM과 행동복제를 기반으로 학습하는 심층강화학습 알고리즘 개발

(1) DQN을 활용한 새로운 알고리즘 구현

기존의 DQN 알고리즘을 개선하여 강화학습 중인 모델이 실시간 피드백을 제공받는 새로운 알고리즘을 구현한다. 실시간으로 제공되는 피드백은 전처리를 거쳐 모델에게 주어지며, 이는 모델의 보상 체계에 영향을 끼쳐 더 뛰어난 학습을 가능하게 한다.

(2) 사용자의 피드백을 변환할 수 있는 LLM 모델 구현

사용자가 음성/자판으로 제공하는 피드백들에 대해 pretrained-BERT모델을 개선하여 긍정/부정 감성분석을 구분해 낼 수 있는 모델을 구현한다.

2.1.2. 시뮬레이터 개발

(1) 유니티엔진을 이용하여 시뮬레이션 환경 개발

Unity엔진의 오픈 소스 강화학습 패키지인 ML-Agents를 이용하여 학습을 진행할 수 있도록 유니티에서 기본적으로 지원하는 3D 렌더링, 물리 시스템을 사용하여 실제 환경과 유사한 시뮬레이션을 개발한다.

(2) ML-Agents를 통해 시뮬레이션과 학습 알고리즘 연결

ML-Agents를 통해 학습 알고리즘을 유니티 시뮬레이션과 연결하여 학습을 진행한다.

(3) UI를 통한 사용자의 시뮬레이터 환경 조정

UI를 통해 사용자가 파라미터를 조정 및 모델 변경시, 변경사항을 적용하여 시뮬레이션을 자동으로 빌드하고 시뮬레이션 할 수 있게한다.

2.1.3. 웹 개발

(1) 시뮬레이터 화면 출력

구동중인 시뮬레이터 화면을 실시간 스트리밍으로 사용자에게 제공한다.

(2) 환경 선택 및 하이퍼파라미터 설정

실험 환경 및 학습 조건을 사용자 입력을 통해 동적으로 조정할 수 있도록 한다.

(3) 모델 저장 및 불러오기

학습 모델의 진행 상황을 저장하고, 필요한 시점에 불러올 수 있는 기능을 제공한다.

(4) 학습 곡선 출력 및 비교

학습 결과를 시각적으로 출력하고, 다양한 조건에서의 학습 성능을 그래프로 비교할 수 있도록 한다.

2.2 비기능적 요구사항

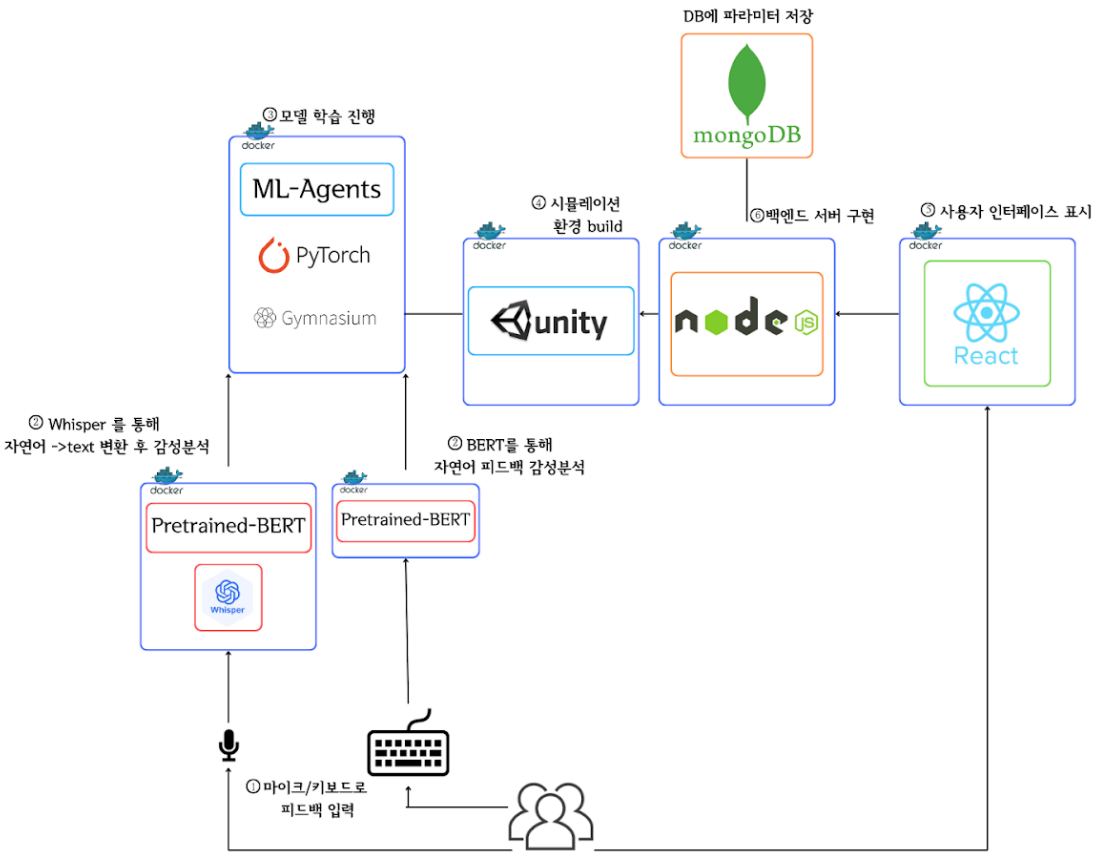
(1) 직관적인 UI를 통해 사용자가 쉽게 사용할 수 있어야 한다.
(2) 실시간 피드백이 중요하므로, 모델이 구동되는 시뮬레이터는 원활하게 실행되어야 한다.
(3) 모든 학습 데이터를 보여주기 보다는 모델 개발에 주요한 데이터 위주로 사용자에게 제공한다.

3. 개발 환경 및 시스템 구조

3.1. 개발환경

1. 강화학습 모델: pyTorch, Gymnasium
2. LLM 모델: Pretrained-BERT
3. 시뮬레이터: unity(ML Agents)
4. web: React, node.js
5. 개발 및 배포 환경 구성 : docker
6. DB: Mongo DB

3.2. 전체 구성도



1. 사용자가 마이크 또는 키보드로 피드백을 입력한다.
2. 자연어로 제공된 피드백이 whisper ai와 pretrained-bert LLM모델을 거쳐 긍정/부정 피드백으로 변환된다.
3. 제공된 피드백은 pyTorch를 기반으로 구현된 강화학습 모델에게 입력된다.
4. 모델의 학습 과은 unity를 통해 사용자에게 시각적으로 제공된다.
5. unity 빌드에서 구동된 모델은 React 기술을 활용한 웹으로 사용자에게 제공된다.
6. React로 구현된 웹은 node.js와 docker기술을 활용해서 서버로부터 제공된다.

4. 개발 일정 및 역할 분담

4.1. 개발 일정

개발구분	세부항목	5	6	7	8	9
기획	주제 선정 및 고도화					
	사전 조사					
인공지능 모델 개발	pretrained-bert 감성분석 fine-tuining					
	human-feedback 구현					
시뮬레이터 설계	시뮬레이터 환경 구현					
	사용자 상호작용 시스템 구현					
	구현된 알고리즘 gym-wrapping					
애플리케이션 개발	웹-시뮬레이션 시스템 연동					
	프론트엔드 개발					
	백엔드 개발					
배포 및 수정	보완사항 수정 및 배포					

4.2. 역할 분담

이름	담당
심영찬	pyTorch활용 DQN개선 알고리즘 개발 pretrained-bert기반 피드백 입력 LLM 모델 개발
김동건	unity 기반 시뮬레이터 개발, Unity - server, 학습 알고리즘간 연결
오현식	React 기반 FE 개발, node.js, docker 기반 BE 개발
공통	관련 논문 분석, 보고서 작성

5. 참고 문헌

[1] 송주상,「알파고 승리 이끈 강화학습, 로봇·자율주행서 재조명」,『IT조선』, 2021년 2월 28일, https://it.chosun.com/news/articleView.html?idxno=2021022602881
[2] Yuxi Li,「Deep Reinforcement Learning: An Overview」, https://arxiv.org/abs/1701.07274
[3] alex-mccarthy-unity,「Unity ML-Agents Gym Wrapper」, ml-agents/docs/Python-Gym-API.md at develop · Unity-Technologies/ml-agents