

클라우드 환경에서 인공지능 모듈 기반 로봇 응용을 위한 증강 지능 모델 공유 기술 개발

장철수, 송병열, 정영숙
한국전자통신연구원 지능로봇연구본부
jangcs@etri.re.kr, sby@etri.re.kr, ysjeong@etri.re.kr

A Development of Augmented Intelligence Model Sharing for AI Modular Robot Application in Cloud Environment

Choulsoo Jang, ByoungYoul Song, YoungSook Jeong
Intelligent Robotics Research Division, Electronics and Telecommunications Research Institute

요 약

본 논문에서는 다양한 인공지능을 모듈화하고 모듈들을 서로 결합하여 서비스를 제공할 수 있는 지능형 서비스 로봇에서, 인공지능 모듈들을 라이브러리 간의 의존성을 해소하기 위한 방법 중 하나인 가상 머신의 일종인 도커(Docker)를 활용하여 컨테이너화하여 사용할 때, 인공지능 모듈 내부에서 사용하는 신경망 데이터에 해당하는 지능 모델에 대해 버전 관리를 수행하면서 클라우드 등 외부 서버를 이용하여 증강시킨 지능 모델을 공유하는 기술 개발에 대해 설명한다.

1. 서론

인간에게 지능적 서비스를 제공하는 서비스 로봇은 다양한 인공지능 소프트웨어들의 조합으로 만들어진다. 서비스 로봇의 운용 환경이 복잡해지고 인간과의 자연스러운 상호작용을 위해서는 인간의 의도를 파악하는 HRI(Human-Robot Interaction)와 환경 인지를 위한 방대한 데이터 처리 능력이 요구된다. 로봇 단독 시스템으로는 방대한 데이터에 대한 처리 능력의 한계가 따를 수 밖에 없으므로 클라우드를 활용한 지능적 서비스 로봇의 개발이 활발히 연구되고 있다[1]. 클라우드와 로봇에서 역할을 나눠 각각 수행되는 인공지능 모듈들은 서로 유기적인 정보 교환을 통해서 로봇 서비스를 수행한다. 이런 인공지능 모듈들은 한번의 학습을 통해 완성되지 않고, 환경과 사용자와의 상호작용을 통해 데이터를 수집하고 피드백을 받으면서 지속적 학습을 통해 지능이 증강되어야 한다[2].

본 논문에서는 클라우드 환경에서 구동되는 컨테이너 기반의 로봇용 인공지능 모듈의 지능 모델이 지속적 학습을 통해 증강되었을 때 이들을 공유하고 배포하기 위한 방법을 제공하기 위해, 증강된 지능 모델의 공유 기술 개발에 관한 내용을 설명한다.

2. 관련 연구

로봇 응용 소프트웨어 개발자들은 인공지능 소프트웨어들을 손쉽게 조합하기 위해 ROS(Robot Operating

System)[3], OpenRTM(Robot Technology Middleware)[4], OPRoS(Open Platform for Robotic Services)[5] 등과 같은 소프트웨어 프레임워크(Software Framework)를 활용하여 인공지능 소프트웨어들의 입출력과 실행 특성을 규격화 하여 모듈화 형태로 개발하고 배포하는 방식을 선호한다. 이를 위해 ISO 와 같은 국제표준화 기구와 지능형로봇표준포럼(KOROS)과 같은 국내표준화 기구에서 로봇 소프트웨어 모듈화를 위한 표준이 개발되고 있다[6,7].

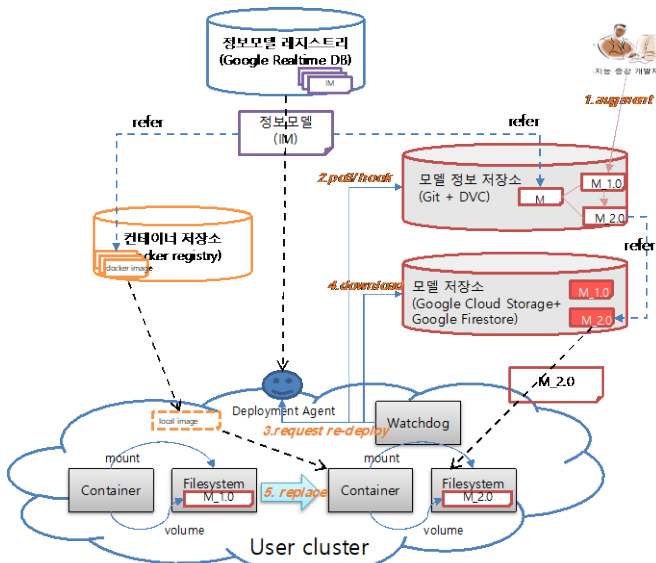
한편 최신의 인공지능 모듈들은 인공 신경망(Artificial Neural Network) 기반 기계학습의 발달을 통해 성능이 크게 향상되고 있으며, 점점 더 많은 신경망 기반의 인공지능 모듈들이 출시되고 있다. 하지만, 여러 인공지능 모듈에서 필요한 인공지능 프레임워크나 외부 패키지들의 버전이 서로 다르거나, 라이브러리 의존 관계가 서로 상충하여 다수의 인공지능 모듈을 하나의 운영체제에서 동시에 구동 시키는데 어려움이 발생한다. 이런 문제를 해결하기 위해, 최근에는 도커(Docker)를 이용하여 소프트웨어 실행에 필요한 운영체제, 런타임, 시스템 라이브러리, 외부 패키지 등 모든 것을 포함하여 가상화 이미지로 만들어 구동시키는 컨테이너(Container) 기술이 개발되었다[8].

컨테이너 기술을 활용하여 만들어진 인공지능 모듈들은 클라우드 환경에서 구동되고 로봇을 활용하여 RaaS(Robotics as a Service)의 형태로 인공지능 서비스

를 제공한다. 로봇에 탑재되는 다양한 인공지능 모듈과 로봇용 소프트웨어 프레임워크를 하나의 패키지로 컨테이너화 시키고, 이들을 로봇 및 클라우드 환경에 배포한 후 로봇용 소프트웨어 프레임워크의 통신 기능을 활용하여 조합하면 로봇이 클라우드 환경에서 인공지능 모듈을 이용하여 사용자에게 서비스를 제공할 수 있게 된다.

3. 증강 지능 모델의 공유를 위한 시스템 구성

증강된 지능 모델의 공유를 위한 시스템 구성은 그림 1 과 같다.



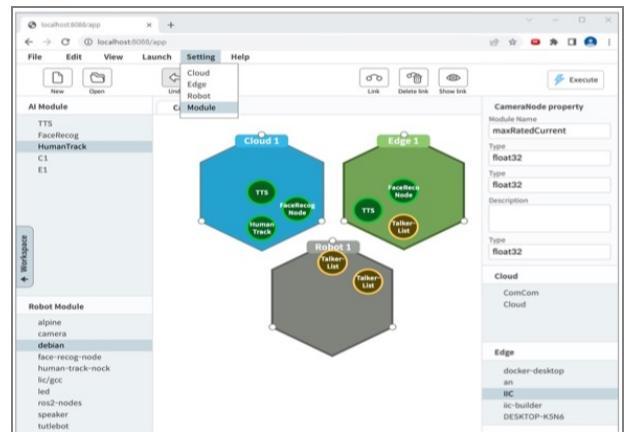
(그림 1) 증강 지능 모델 공유 시스템의 구성도

로봇용 소프트웨어 모듈의 표준 중 KOROS 소프트웨어 모듈의 정보모델(Information Model)을 확장한 인공지능 모듈의 정보모델은 인공지능 모듈을 만들 때 생성하여 모듈 정보모델 레지스트리에 저장해 놓는다. 모듈의 정보모델은 모듈을 실행하는데 필요한 도커 이미지와 인공지능 모델에 대한 참조 정보를 포함하고 있다. 인공지능 모델에 대한 참조 정보는 지속적인 학습을 통해 증강되는 지능 모델에 대한 버전 정보를 가리키고 있다.

모델 정보 저장소에 저장되는 모델 정보는 인공지능 모델의 신경망 구조를 재구축하는데 필요한 클래스의 소스파일과 버전 별 모델 파일에 대한 경로를 저장하고 있는 모델 URL 파일로 구성된다. 모델 정보에 저장된 이들 파일들은 버전 관리 시스템(Version Control System)인 Git[9]을 활용하여 관리한다. Git 과 같은 보통의 버전 관리 시스템은 크기가 큰 바이너리 형태의 모델을 직접 저장하고 관리하는데 있어서 저장용량의 제약이나 성능의 문제가 발생하기 때문에 모델 정보 중 모듈 버전 정보와 같은 핵심 정보만 버전 관리 시스템으로 관리하고 모델 파일 자체는

NAS(Network Attached Storage)와 같은 네트워크 기반 저장장치나 Google Cloud Storage 버킷(bucket)혹은 Amazon s3 버킷과 같은 클라우드 스토리지 서비스를 활용하여 모델 저장소를 구축하여 저장하고, 모델 정보에는 저장한 모델을 접근할 수 있는 경로 URL 정보와 소스 파일 형태로 크기가 작은 모델 구축 클래스만 저장한다. 이를 위해 Git 과 더불어 DVC(Data Version Control)[10]을 활용하여 모델 데이터에 대한 저장 및 버전 관리를 병행하도록 개발하였다.

본 개발에서는 정보모델 레지스트리에 저장되어 있는 모듈들을 그림 2 와 같은 로봇 응용 구성 GUI (Graphic User Interface) 도구를 활용하여 로봇 응용 구성 창에 끌어 놓기를 통해 모듈들을 배치하고 각 모듈 사이의 연결 관계를 설정하는 방식으로 로봇 응용을 만든다.



(그림 2) GUI 기반 로봇 응용 구성 도구

로봇용 컴퓨터 및 클라우드 서버에 탑재된 배포 에이전트는 로봇 응용에 해당하는 응용 구성정보를 해석하고, 로봇 응용에 참여하는 모듈 정보를 참조하여 컨테이너 저장소에 저장되어 있는 모듈들을 로봇에 다운로드한다. 배포 에이전트는 각 모듈이 사용하는 인공지능 모델에 대해 모델 정보 저장소 URL 에 연결하여 인공지능 모델 이름에 해당하는 모델 정보를 복사하여(git clone) 가져온 후, 모델 정보에 포함되어 있고 Git 으로 관리되는 모델 버전으로 체크아웃(git checkout)을 수행하여 모델 정보를 원하는 버전의 스냅샷으로 변경한다. 배포 에이전트는 DVC 를 통해 해당 스냅샷의 모델 정보가 가리키는 모델 저장소 접근 경로 URL 을 이용하여 네트워크 기반 저장장치나 클라우드 스토리지 서비스로부터 모델 파일을 다운로드한다.

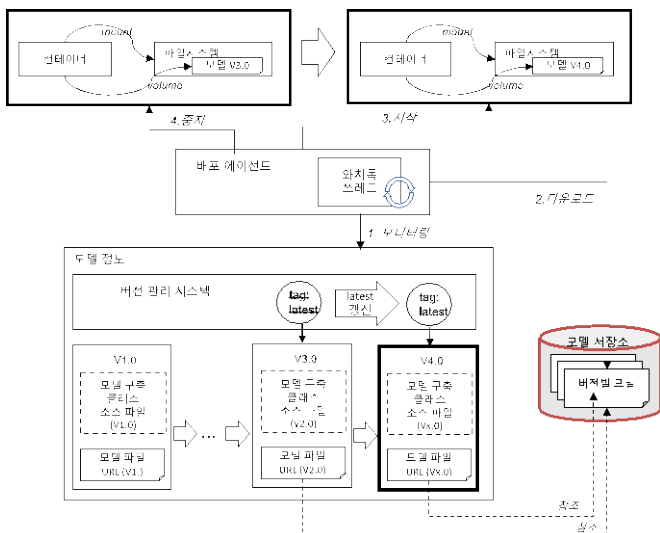
다운받은 컨테이너 이미지를 실행시킬 때 로컬 파일시스템에 저장한 지능 모델을 컨테이너에 볼륨으로 마운트(mount)하여 모듈을 실행시키면, 모듈은 실행 초기화 과정에서 마운트된 볼륨을 마치 로컬 파일시

스택처럼 접근하여 인공지능 모델을 메모리에 적재하여 신경망 데이터를 완성하고 모듈의 알고리즘 수행 과정에 활용한다.

다운로드된 모델 파일들은 내부 캐쉬 저장소에 저장하여 운용된다. 즉, 모듈이 다시 시작하여 기존에 사용하였던 모델을 사용하기 위해 다운로드하려고 할 때 이미 내부 캐쉬에 저장되어 있는 모델 파일이 존재하는 경우에는 다시 다운로드하지 않고 캐쉬 저장소에 있는 모델 파일을 이용한다.

4. 증강된 지능 모델의 롤링 업데이트

인공지능 모델을 위한 응용 구성정보 중 지능 모델이 증강되어 갱신된 경우에 갱신된 모델을 업데이트하기 위한 AutoUpdatePolicy 정책을 추가로 정의한다. AutoUpdatePolicy 가 NoUpdate 로 설정되어 있는 경우에는 다운로드되어 캐쉬에 저장된 모델이 있는 경우 해당 모델을 이용하고, 캐쉬되어 있는 모델이 없는 경우에만 다운로드를 수행한다. 정책이 UpdateOnStart 로 설정되어 있는 경우에는 도커를 시작할 때 만 최신 버전의 모델 여부를 확인하고 캐쉬되어 있는 모델보다 최신 버전이 있는 경우 새로 다운로드하여 실행한다. 정책이 UpdateOnRunning 으로 설정되어 있는 경우에는 도커 컨테이너가 동작 중일 때에도 최신 모델이 있는 경우에는 그림 3 과 같이 와치독(Watchdog)을 통해 항상 최신 모델을 다운로드하고, 순차적으로 롤링 업데이트(Rolling Update) 형태로 도커를 재실행 시킨다.



(그림 3) 증강 지능 모델 공유를 이용한 롤링 업데이트.

5. 결론

본 고에서는 클라우드 환경에서 구동되는 컨테이너 기반의 로봇용 인공지능 모듈을 위한 증강된 지능 모델의 공유 방법을 설명했다. 증강 지능 모델의 공유

를 위해 지능 모델 정보와 데이터를 분리하여 버전별로 저장하고 관리할 수 있는 버전 관리 기술에 대한 개발 내용을 설명하였으며, 증강 지능 모델 공유 기능을 이용하기 위한 인공지능 모듈의 정보 모델과 이들을 조합한 로봇 응용 구성 방법에 대해 설명하였다.

증강 지능 모델 공유 기술은 지속적 학습을 통해 지능 모델이 증강되었을 때, 지능 모델을 로봇 및 클라우드 서버에 자동으로 재배포하여 로봇 응용 소프트웨어에 참여하고 있는 인공지능 모듈의 지능이 업데이트할 수 있고, 최종적으로는 사용자에게 서비스하고 있는 로봇의 지능을 향상시킬 수 있는 메커니즘을 제공한다.

향후, 개발된 증강 지능 모델 공유 기술을 클라우드 환경에서 다양한 로봇 응용에 적용하여 실증할 계획이다.

Acknowledgment

본 연구는 2022 년도 과학기술정보통신부의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00857, 다수 로봇의 지능을 통합 고도화하는 클라우드 로봇 지능 증강·공유 및 프레임워크 기술 개발)

참고문헌

- [1] B. Kehoe et al., "A Survey of Research on Cloud Robotics and Automation," IEEE Transactions on Automation Science and Engineering, 2015.
- [2] G. Parisi et al., "Continual lifelong learning with neural networks: A review," Neural Networks, Volume 113, 2019.
- [3] S. Macenski et al., "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics, Vol.7, No.66, 2022.
- [4] OpenRTM, Available: <http://openrtm.org/>
- [5] 장철수, 송병열, 김성훈, "Windows 용 경성 실시간 확장 플랫폼을 위한 멀티 코어 지원 개방형 로봇 미들웨어," 한국로봇종합학술대회, 2016.
- [6] ISO 22166-1:2021, "Robotics-Modularity for service robots-Part 1: General requirements," ISO, 2021.
- [7] KOROS 1148-2:2021, "서비스로봇을 위한 모듈 - 제 2-2 부: 소프트웨어 모듈을 위한 정보 모델", 한국로봇산업협회, 2021.
- [8] D. Merkel, "Docker: lightweight Linux containers for consistent development and deployment," Linux Journal, Volume 2014, 2014.
- [9] Git, Available: <http://git-scm.com/>
- [10] DVC, Available: <http://dvc.org/>