**환경구축 및 학습 코드 실행**

\* 본 학습은 도커 기반의 환경 설정을 활용하였기 때문에 도커 계정이 있으며, 로컬 환경에 도커가 설치되었다는 것을 가정합니다. (windows10 docker 설치 방법: https://goddaehee.tistory.com/251)

1. 도커 이미지 파일 위치로 이동

cd D:/NIA\_Emergency

# 기본 상위 폴더(D:/NIA\_Emergency)로 이동

1. 도커 이미지파일 불러오기

docker load -i er\_process\_cls.tar

# requiremnets.txt 파일에 명시된 학습에 필수적으로 필요한 라이브러리로 구성된 도커 이미지 파일

1. 도커 이미지 목록 확인

docker images

# 해당 명령어 실행 시 출력 결과가 다음과 같을 경우(REPOSITORY, TAG, IMAGE ID, SIZE 일치) 성공한 것입니다.



1. 도커 이미지로 컨테이너 만들기 실행

docker run -it --gpus all --name er\_process\_cls -v ${pwd}:/home/workspace sjw0614/er\_process\_cls:v0

# 컨테이너 내부의 home/workspace 경로에 현재 폴더 경로(D:/NIA\_Emergency) 마운트

1. Java 파일 설치

apt install default-jdk -y

# 활용하는 라이브러리 중 konlpy의 경우 java를 활용하기 때문에 추가 설치가 필요합니다.

1. 컨테이너 내부 코드 실행 경로로 이동

cd /home/workspace/code

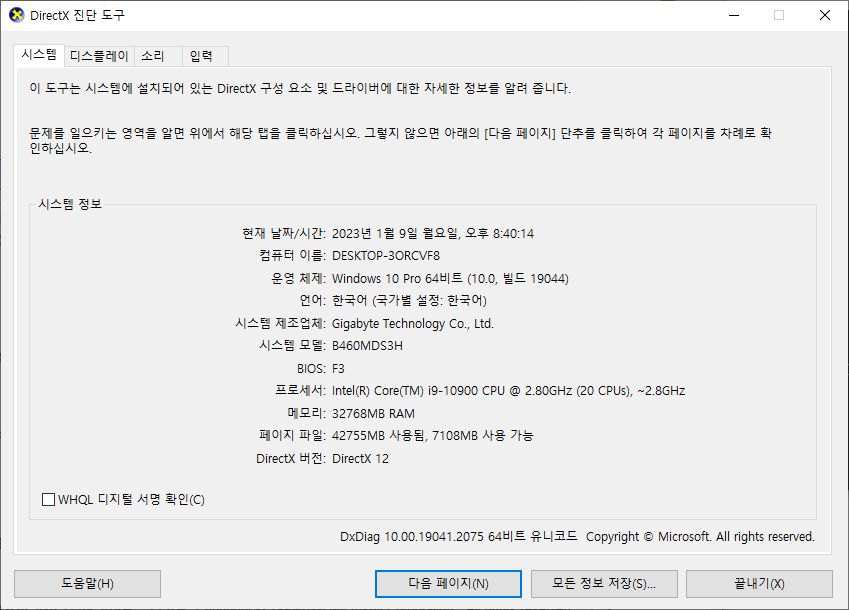
1. 학습 코드 실행

python train.py

# 학습 및 추론을 진행하며, 결과 파일은 /home/workspace/result 및 로컬환경(D:/NIA\_Emergency)에 저장됩니다.

**시험 환경**

* OS: windows
* Python: 3.9
* Tensorflow-gpu 2.10.1
* CPU: intel-i9-10900
* Ram: 32GB
* SSD: 465GB
* GPU: nvidia-RTX 3070 8GB
* CUDA Toolkit: 11.2
* cuDNN: 8.1.0



## 