**1. Training Enviroment**

* RTX A6000 x 1
* CUDA 12.1
* Pytorch 2.1
* Python 3.10
* 각 모델 설정값 참고

**2. 파일 저장 구조**

**2.1 지상관측자료 데이터 위치  
root\_path = ‘/home/work/data/dataset/lstm\_model’**

├── code  
│ ├── [Train&Infer]LSTM\_v1.ipynb

│ └── lstm\_v1.pth

├── data2

│ ├── Year

│   │ ├── Test

│   │ │ └── 2021

│   │ ├── Train

│   │ │ ├── 2012

│   │ │ ├── 2013

│   │ │ ├── 2014

│   │ │ ├── 2016

│   │ │ ├── 2017

│   │ │ ├── 2018

│   │ │ └── 2019

│   │ ├── Valid

   ├── 2015

   └── 2020

**인공위성자료 데이터 위치**

|  |  |  |
| --- | --- | --- |
| Base | Data split | 연도별 데이터셋 구성 |
| /home/work/dataset/CNN3D\_model/data/ | Multi\_modal/train | 2012  2013  2014  2016  2017  2018  2019 |
| Multi\_modal/valid | 2015  2020 |
| Test | 2021 |

**3. 모델 실행 방법**

**[모델1] LSTM model**

요약: 지상관측자료 강우 데이터를 활용하여 LSTM 모델을 학습 및 검정하는 코드

파일 위치 및 파일명: /home/work/data/dataset/lstm\_model/code/[Train&Infer] LSTM\_v1.ipynb

모델 입력: 6개의 시퀀스 길이(과거 6시간)와 10개의 지점 강우량을 가진 데이터

모델 출력: 6개의 시퀀스 길이(미래 6시간)와 10개의 지점에 대한 강우량 예측 값

구성:

Import: 라이브러리 불러오기

Config: 데이터셋 위치, 모델 학습 환경, 하이퍼 파라미터 등 실험을 위한 각종 사용자 옵션을 정의함. 실험 재현을 위한 고정된 난수를 설정함.

Data preprocessing: 데이터셋을 불러와 학습을 위한 데이터셋으로 구성하기 위한 전처리 코드. 2012년부터 2020년까지의 데이터 중 2015년, 2020년을 validation set으로 설정하고 나머지는 train set으로 구성함.

Model build: LSTM 모델을 빌드함. 3개의 LSTM layer을 쌓고 뒤에 linear layer 두개를 붙여 모델을 구성함.

Model training: 모델 학습을 위한 함수 설정함. 모델 평가 방법은 MAE로 하지만 학습을 위한 손실함수로는 MSE로 구성함. 이유는 많은 양의 비가 오는 경우에 대한 모델의 학습이 MSE에서 효과적이기 때문임. 학습 과정을 추적하기 위한 그래프를 시각화 함. 모델을 저장하고 불러오는 코드 설정함.

Validation: 학습이 완료된 모델을 불러와서 validation set에 대한 강우 예측력을 그래프로 확인함.

Test on 2021: Test을 위한 2021년도 데이터 중 실제로 예측해야 하는 구간 정보가 담긴 csv 파일을 불러와 데이터셋을 구성함.

MAE & Correlation score on 2021: 테스트셋에 대해 MAE와 correlation score을 산출하고 구간에 따른 예측정도를 시각화 함.

MAE & Correlation by STN: Station 별 MAE와 correlation score를 산출함. 시간에 따라 각 지점별로 예측 분포가 어떻게 나타나는지 확인하는 코드.

MAE & Correlation by Time: 예측하는 미래의 6시간에 대해서 각 시간별로 MAE와 correlation score가 어떻게 나타나는지 확인하는 코드. 예측해야 하는 시점이 현재로부터 가까울수록 MAE와 Correlation score가 높을 것으로 기대.

**[모델2] Conv3D depth-wise pooling UNet (Optional)**

요약: multimodal 데이터를 활용하여 Conv3D depth-wise pooling UNet 모델을 학습 및 검정하는 코드

파일 위치 및 파일명: /home/work/dataset/CNN3D\_model/code/[train & infer].ipynb

모델 입력: T x C x W x H tensor

* T: time sequence, ex) 6
* C: Channel, ex) 9
  + TA: 기온
  + WD: 풍향
  + WS: 풍속
  + RN\_day: 하루 강수량
  + RN\_1h: 1시간 강수량
  + HM: 습도
  + PA: 현지기압
  + PS: 해면기압
* W: width, ex) 128
* H: height, ex) 128

모델 출력: T x 1 x W x H tensor -> 6시간 후의 강우량 예측 분포

구성:

Train.ipynb

* Import library
* Dataset:
  + MultimodalDataset\_3D for 9 channel 3D tensor
  + RadarDataset\_3D for single channel radar 3D tensor
* Test model architecture in CPU
* Visualize the results on the validation set

Test.ipynb

* Import library
* Load trained model
* Run test