

졸업 프로젝트 데이터 정보

2조

1. 경로 종류

- 가. 동해→포항(dp)
- 나. 부산→목포(bm)
- 다. 울산→동해(ud)
- 라. 여수→울산(yu)
- 마. 인천→제주(ij)

2. 테이블(DBMS: PostgreSQL)

가. Input Data: 모델의 Input으로 사용되는 데이터로, MinMaxScaler¹⁾에 의해 미리 정규화 처리 완료. 최종 Input으로 사용될 데이터의 shape 으로는 ²⁾(num_samples, 40, 5)가 되어야 한다. 각 테이블의 열의 개수는 총 200개이며, 각 열의 이름은 c1, c2, ... c200이다. 각 경로의 데이터가 저장된 테이블의 이름은 다음과 같다.

- 1) donghae_to_pohang
- 2) busan_to_mokpo
- 3) ulsan_to_donghae
- 4) yeosu_to_ulsan
- 5) incheon_to_jeju

나. Output(=Label) Data: 모델이 예측해야 될 값을 평가하기 위해 사용되는 데이터로, Input Data와 동일한 Scaler에 의해 미리 정규화 처리 완료. 최종 Label로 사용될 데이터의 shape 으로는 (num_samples, 1, 4)³⁾가 되어야 한다. 각 테이블의 열의 개수는 총 4개이며, 각 열의 이름은 c1, c2, c3, c4이다. 각 경로의 데이터가 저장된 테이블의 이름은 다음과 같다.

- 1) donghae_to_pohang_target
- 2) busan_to_mokpo_target
- 3) ulsan_to_donghae_target
- 4) yeosu_to_ulsan_target
- 5) incheon_to_jeju_target

3. 테스트 데이터

가. 테스트 데이터는 각 경로당 하나의 csv파일로 설정. 차우 모델과 자가외귀 알고리즘에 의해 그려진 경로와, 실제 vessel의 경로를 비교하여 성능 비교.

1) 위도 범위: 33~38, 경도 범위: 124~132, sog 범위: 0~100, cog 범위: 0~360

2) input_seq_len=40(20분), column=5(위도, 경도, sog, cog, 목적지와의 유클리디안 거리)

3) output_seq_len=1(30초), column=4(위도, 경도, sog, cog)

4. 예시 코드(csv 파일 기준)

가. 각 csv 파일로부터 데이터 가져오기 및 reshape

```
## 데이터 불러올때 참고 코드!
# 불러오기
input_df = pd.read_csv("bm_input_seqs.csv", encoding='cp949', header=None)
label_df = pd.read_csv("bm_label_seqs.csv", encoding='cp949', header=None)

# numpy 변환
input_seqs = input_df.to_numpy().reshape(-1, 40, 5)
label_seqs = label_df.to_numpy().reshape(-1, 1, 4)
```

나. 모든 데이터 hstack 및 최종 input/output data 생성

```
input_csv_files = [
    'bm_input_seqs',
    'dp_input_seqs',
    'ij_input_seqs',
    'ud_input_seqs',
    'yu_input_seqs'
]
output_csv_files=[
    'bm_label_seqs',
    'dp_label_seqs',
    'ij_label_seqs',
    'ud_label_seqs',
    'yu_label_seqs'
]
input_seqs = []
label_seqs = []

for file in input_csv_files:
    file_path = "rou/" + file + ".csv"
    if os.path.exists(file_path):
        df = pd.read_csv(file_path, header=None)
        data = df.values
        data = np.array(data).reshape(-1, 40, 5)
        input_seqs.append([data])

for file in output_csv_files:
    file_path = "rou/" + file + ".csv"
    if os.path.exists(file_path):
        df = pd.read_csv(file_path, header=None)
        data = df.values
        data = np.array(data).reshape(-1, 1, 4)
        label_seqs.append([data])

input_seqs = np.hstack(input_seqs)
label_seqs = np.hstack(label_seqs)
input_seqs = np.array(input_seqs)
label_seqs = np.array(label_seqs)

input_seqs = input_seqs.reshape(-1, 40, 5)
label_seqs = label_seqs.reshape(-1, 1, 4)
```

5. 서버 연결

가. ip주소: 121.137.181.45(변동 가능)

나. dbname: ais_data

다. user: postgres

라. port: 5432

마. password: ky76018500