

THE IMAGINEER PROJECT - FINAL ACCOUNTANT

태양광 발전량 예측 알고리즘 개발

지도 교수 : 이성근

20164368 김민수

20164370 배재한

20154376 김진성

20184304 한유리



ENERGY

Contents



주제 선정 이유

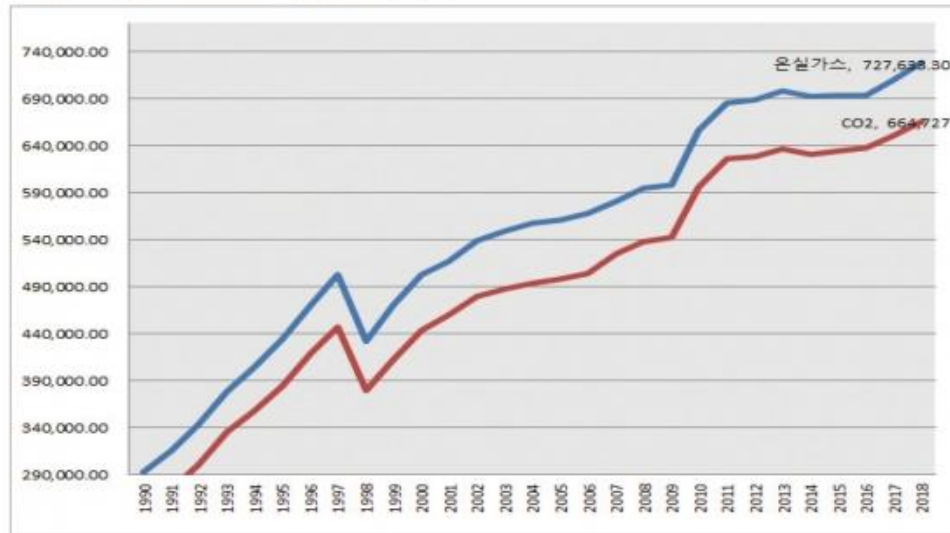
태양광 발전량 예측 알고리즘 개발



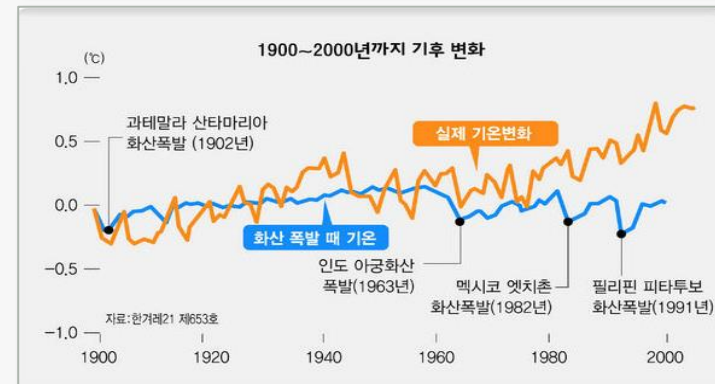
1. 주제

화석연료 사용으로 인한 지구온난화

〈그림 1〉 지난 30년간 탄소배출량 증가 추이



※자료: 환경부, 단위: 천톤



1. 주제

화석연료의 고갈 문제



"LNG도 언젠가는 고갈될 자원...화석연료 문제 반복될 것"

2018.10.16 09:13

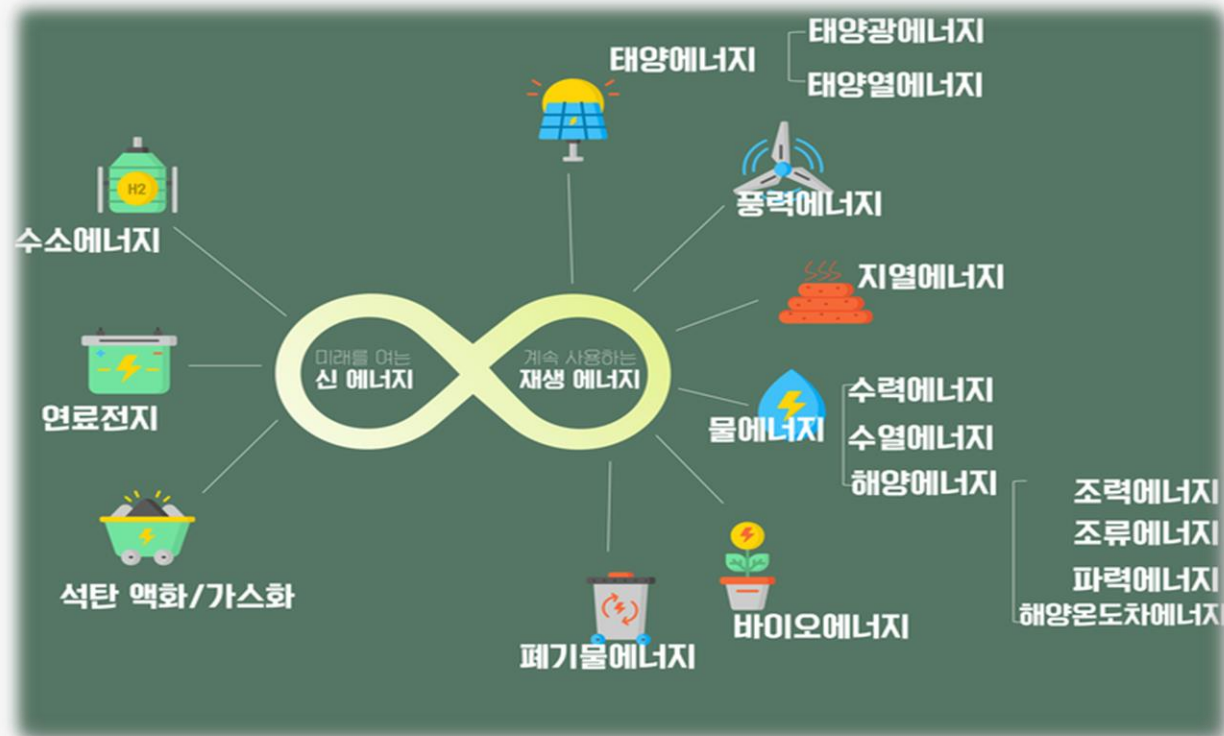
0 버나드 비고 ITER 사무총장 인터뷰

독자가 많이 읽었습니다

- 1 '이제 좀 쉬... 그만, 사람'
- 2 남아공서 '변이까지...'
- 3 "오미크론 불평등이...
- 4 "표준모형 입자물리의
- 5 단계적 일상 위험도 전...

1. 주제

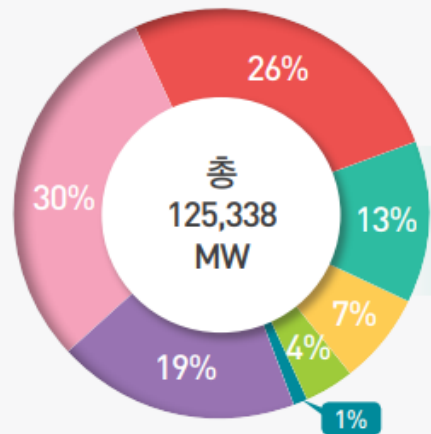
신재생 에너지의 시대



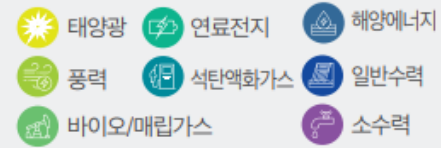
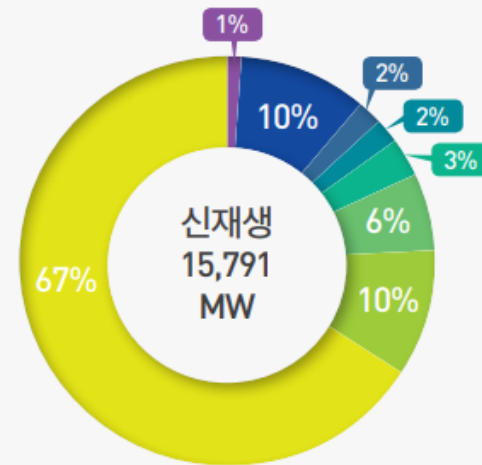
1. 주제

태양광 에너지의 시대

2019년도 발전설비 비중



2019년도 신재생 설비 종류별 비중



2. 목표 설정

연구의 목표

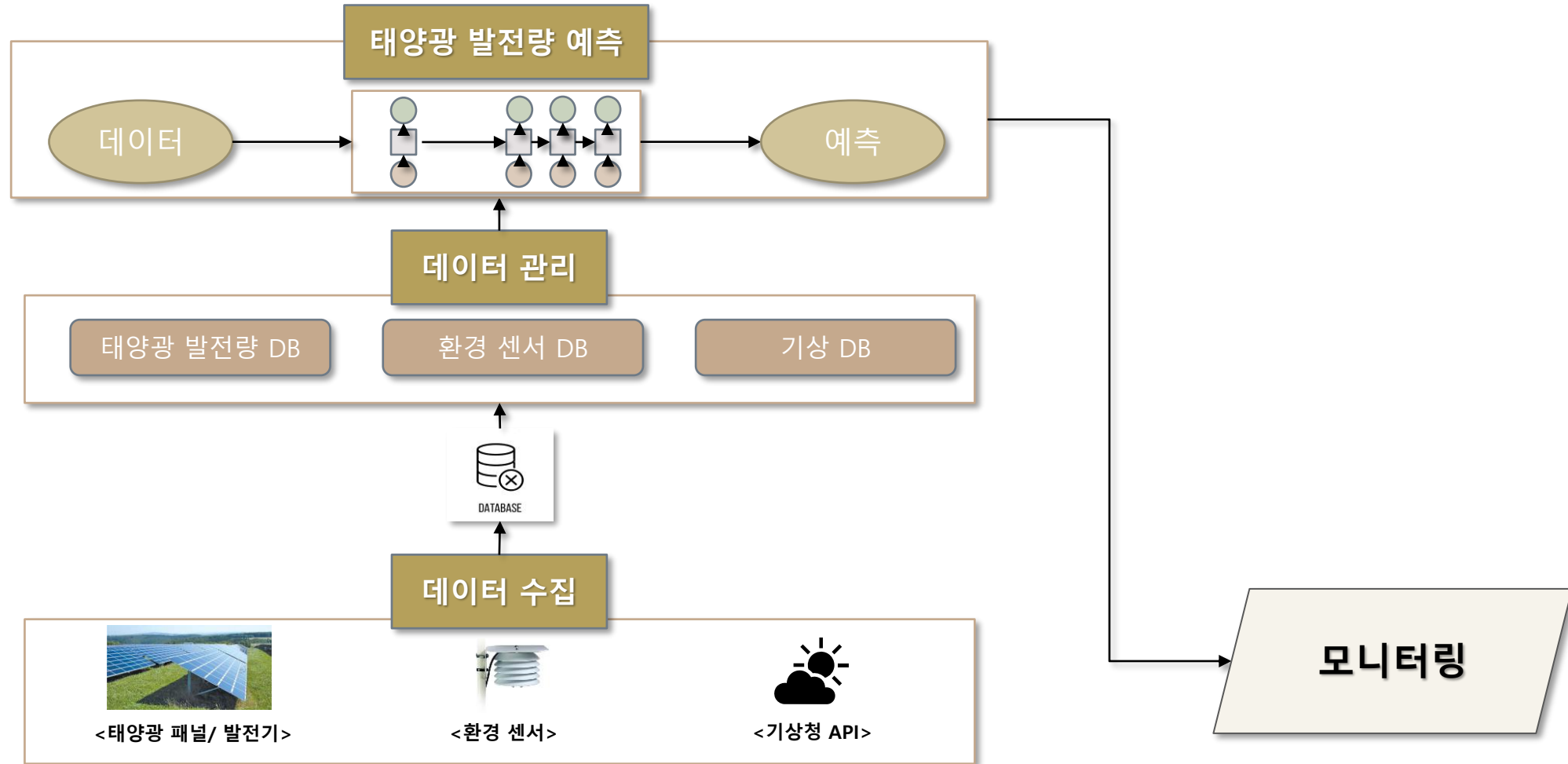
태양광 발전시설에 대한 고찰

- 태양광 발전 시설은 장기간 운용이 보장되어야 하며, 이 과정에서 발전 효율저하를 막고, 고장진단 및 부품 교체 등의 사안에 대해 빠른 대응이 필요
- 태양광 시장에서 모듈가격이 하락하고 있는 상황에서 시스템을 유지보수하기 위한 비용이 상대적으로 증가하는 추세이므로, 지능적이고 효율적인 운용 및 진단 기능의 중요성이 크게 부각

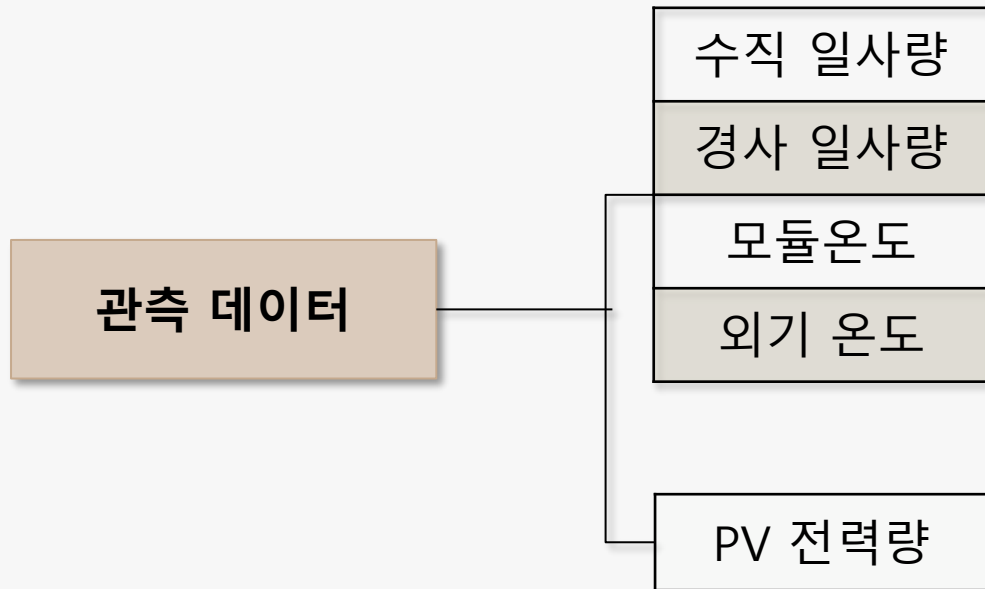
2. 목표 설정



3. 설계 - 태양광 발전량 예측 시스템의 구성도



데이터 셋 – 관측 데이터

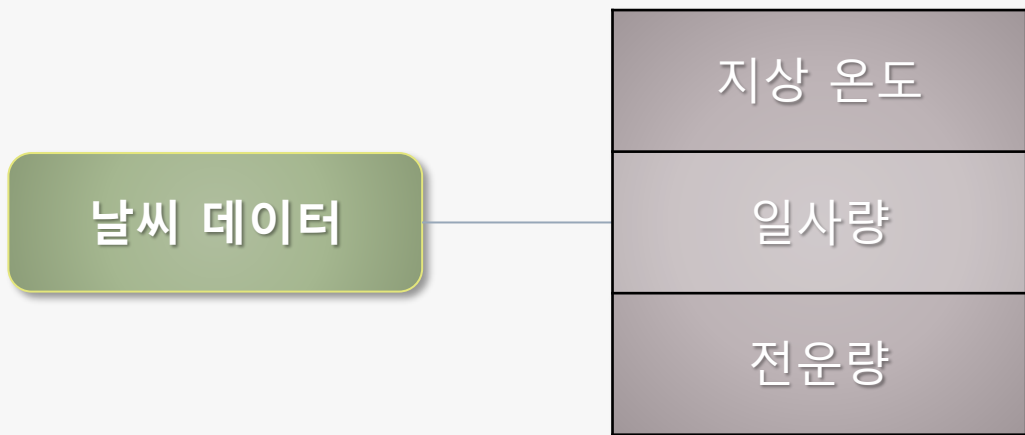


*PV : photovoltaic(태양광 발전)의 약어



*국내 최대 규모인 영암 태양광 발전소

데이터 셋 – 날씨데이터



기상자료개방포털이란?
데이터
기후통계분석
간행물
소통과 참여
ALL

데이터

기상관측
^

| 지상
-

- 종관기상관측(ASOS)
- 방재기상관측(AWS)
- 농업기상관측(AAOS)
- 북한기상관측
- 공공기관 기상관측
- 계절관측
- 항사관측(PM10)
- 낙뢰관측

Home > 데이터 > 기상관측 > 지상 > 종관기상관측(ASOS)

종관기상관측(ASOS) - 자료

자료설명

■ 자료설명

종관기상관측이란 종관규모의 날씨를 파악하기 위하여 정해진 시각에 모든 관측소에서 같은 시각에 실시하는 지상관측을 말합니다. 종관규모는 일기도에 표현되어 있는 보통의 고기압이나 저기압의 공간적 크기 및 수명을 말하며, 주로 매일의 날씨 현상을 뜻합니다.

자료형태	분, 시간, 일, 월, 연	제공기간	1904년~(지점별, 요소별 다름)
제공지점	102개	제공요소	기온, 강수, 바람, 기압, 습도, 일사, 일조, 눈, 구름, 시정, 지면상태, 지면·초상온도, 일기현상, 증발량, 현상번호
유의사항	1회 조회 가능 최대 기간: 분 1일, 시간 1년, 일 10년, 월·연 제한 없음(장기간 자료는 '파일셋 조회' 메뉴 이용) 시간/분 자료에 대해 관측값의 정상 여부를 판단하는 품질검사 플래그(QC FLAG) 정보 제공 * 제공 요소: 기온, 습도, 기압, 지면온도, 풍향, 풍속, 일조 / 플래그 종류(의미): 0(정상), 1(오류), 9(결측)		
비고	10분 또는 1시간 최다강수시각은 최다강수가 나타난 시작 시간으로, (-) 표기가 있는 경우 전날을 뜻함		
지침	요소별 관측방법이나 자료 산출방식에 대한 상세 설명은 ☞ [지상기상관측지침] 참조		

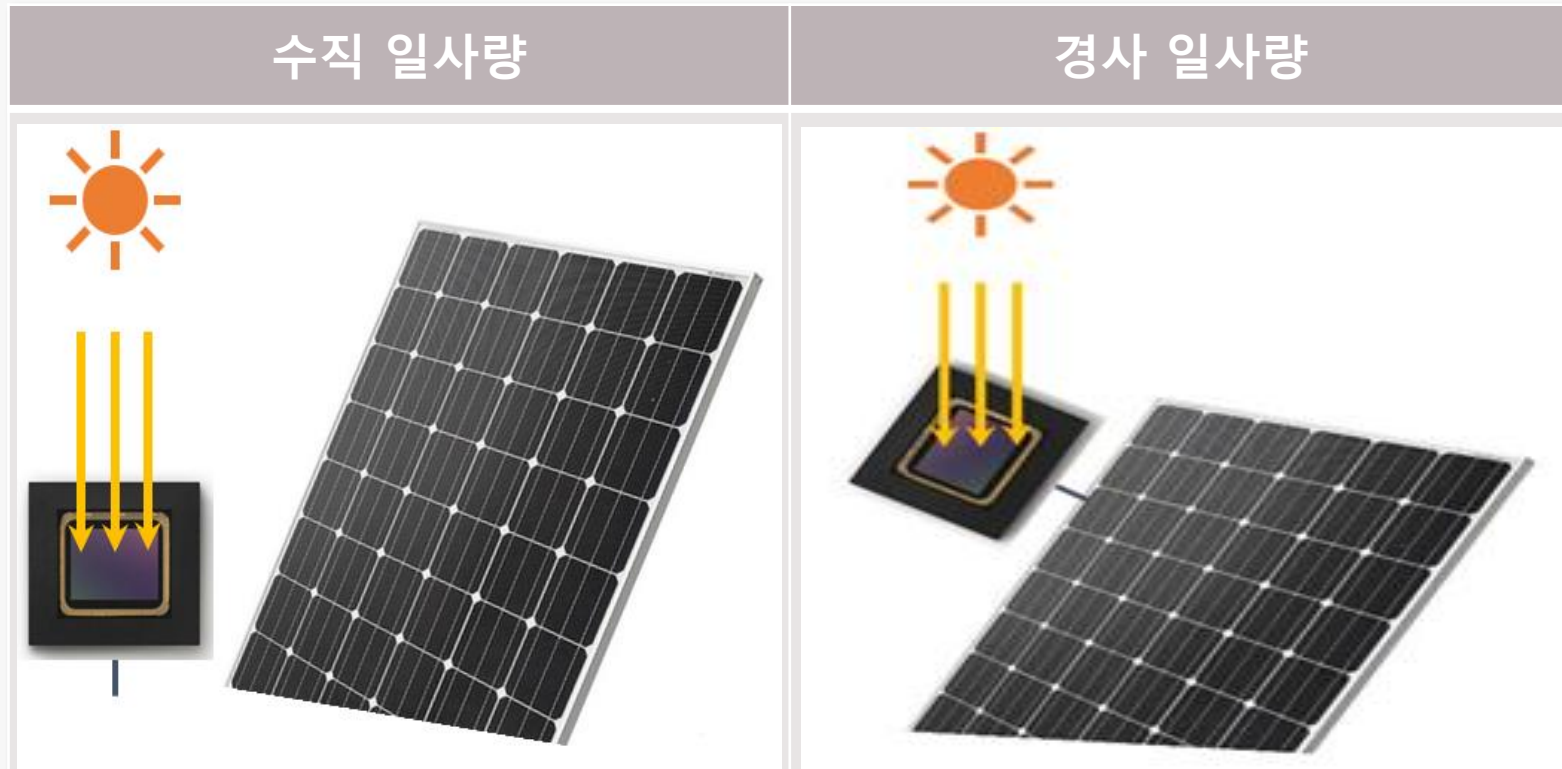
데이터 셋 – 전체데이터

```
df = pd.read_csv('data/solar1.csv')
df.tail()
```

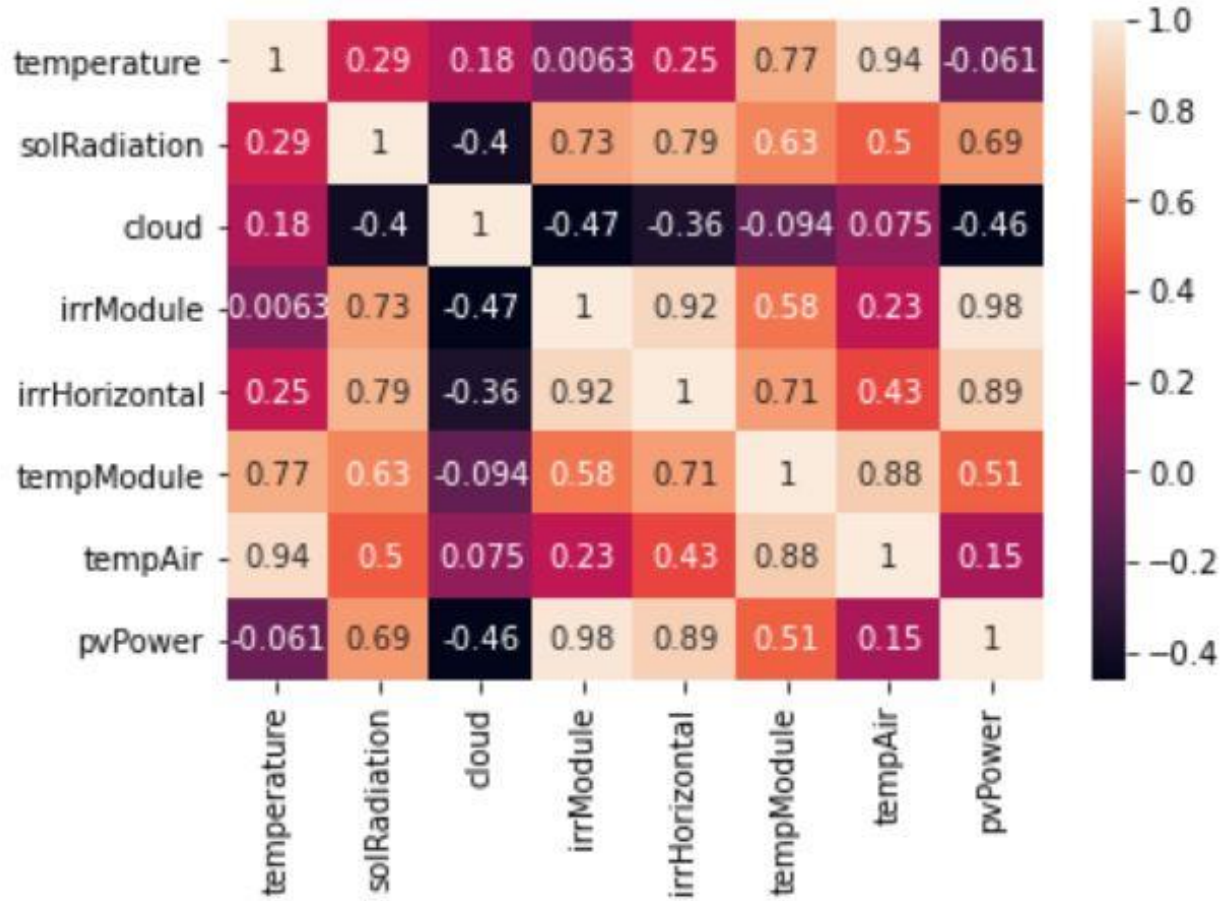
	day	temperature	solRadiation	cloud	irrModule	irrHorizontal	tempModule	tempAir	pvPower
3787	2021-08-02 8:00	25.1	0.22	10	277.62	329.75	35.34	27.91	27.96
3788	2021-08-02 9:00	25.2	0.32	9	430.88	493.25	40.20	30.62	42.75
3789	2021-08-02 10:00	26.2	1.18	9	323.50	381.12	38.67	31.70	30.70
3790	2021-08-02 11:00	26.4	1.09	9	197.57	229.29	33.54	29.43	20.46
3791	2021-08-02 12:00	27.1	1.71	9	398.00	439.00	35.50	29.20	45.60

temperature	지상 온도
solRadiation	일사량
Cloud	전운량
irrModule	경사일사량
irrHorizontal	수직일사량
tempModule	모듈 온도
tempAir	외기 온도
pvPower	태양광 패널 전력량
데이터 총 개수	3792개

데이터 – 일사량 측정 방법



데이터 분석 – 상관관계 그래프



다음 그래프를 통해서 태양광 발전량과 일사량이 밀접한 관계를 갖는다는 것을 알 수 있다.

4. 구현 - 환경

개발 언어	파이썬(python)
프레임워크	파이토치(pytorch)
모델	다중 계층 퍼셉트론(MLP)
정규화	MinMaxScaler()
최적화	Adam()
비용함수	MSELoss()

데이터 전처리 – MinMaxScaler()

```
df = pd.read_csv('data/solar1.csv')  
df.tail()
```

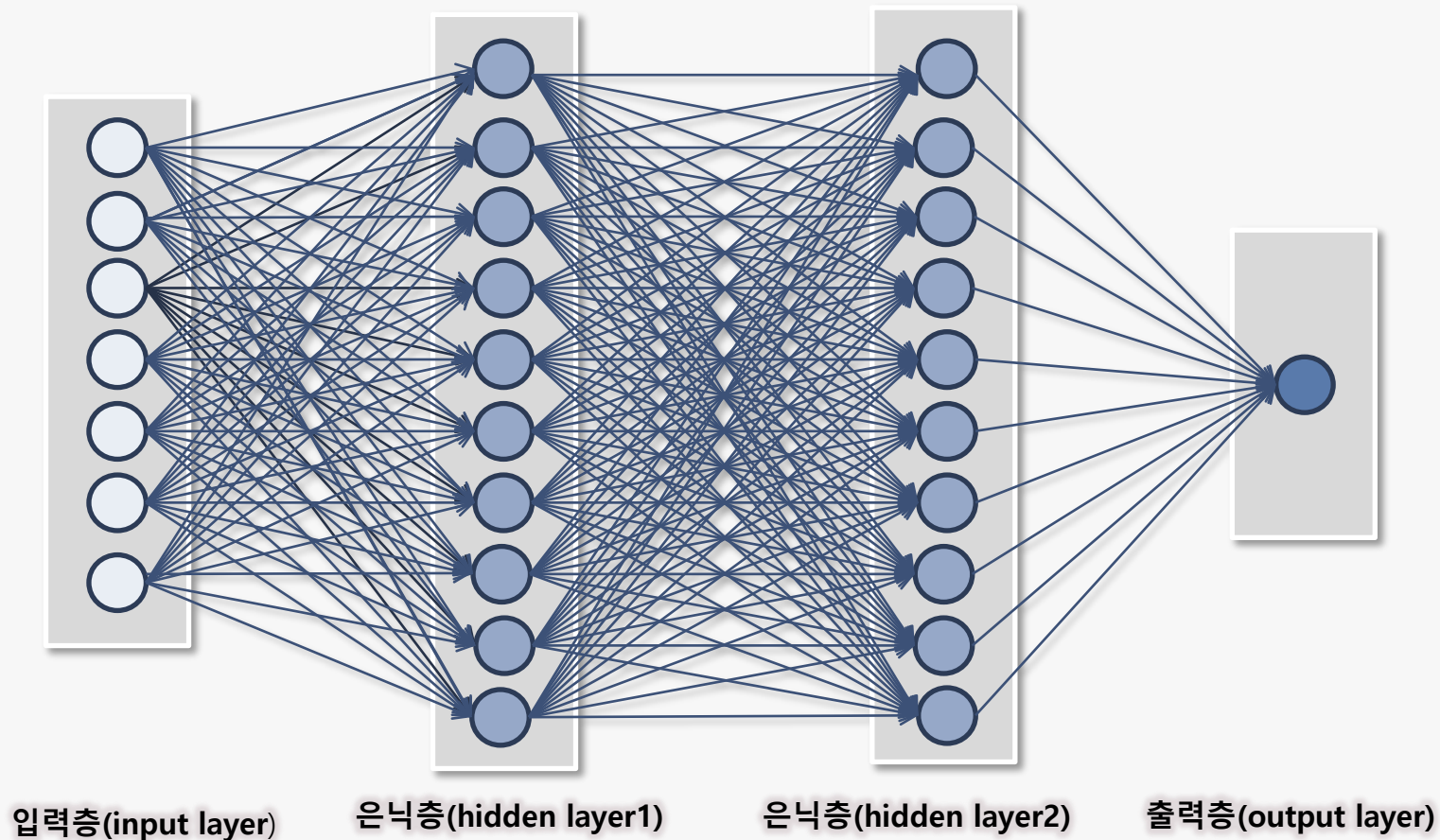
	day	temperature	solRadiation	cloud	irrModule	irrHorizontal	tempModule	tempAir	pvPower
3787	2021-08-02 8:00	25.1	0.22	10	277.62	329.75	35.34	27.91	27.96
3788	2021-08-02 9:00	25.2	0.32	9	430.88	493.25	40.20	30.62	42.75
3789	2021-08-02 10:00	26.2	1.18	9	323.50	381.12	38.67	31.70	30.70
3790	2021-08-02 11:00	26.4	1.09	9	197.57	229.29	33.54	29.43	20.46
3791	2021-08-02 12:00	27.1	1.71	9	398.00	439.00	35.50	29.20	45.60

	temperature	solRadiation	cloud	irrModule	irrHorizontal	tempModule	tempAir	pvPower
3787	0.809302	0.059140	1.0	0.222885	0.291106	0.585486	0.691483	0.200223
3788	0.811628	0.086022	0.9	0.386624	0.473291	0.656023	0.744299	0.365106
3789	0.834884	0.317204	0.9	0.271902	0.348346	0.633817	0.765348	0.230769
3790	0.839535	0.293011	0.9	0.137361	0.179165	0.559361	0.721107	0.116611
3791	0.855814	0.459677	0.9	0.351496	0.412841	0.587808	0.716624	0.396878



정규화

모델



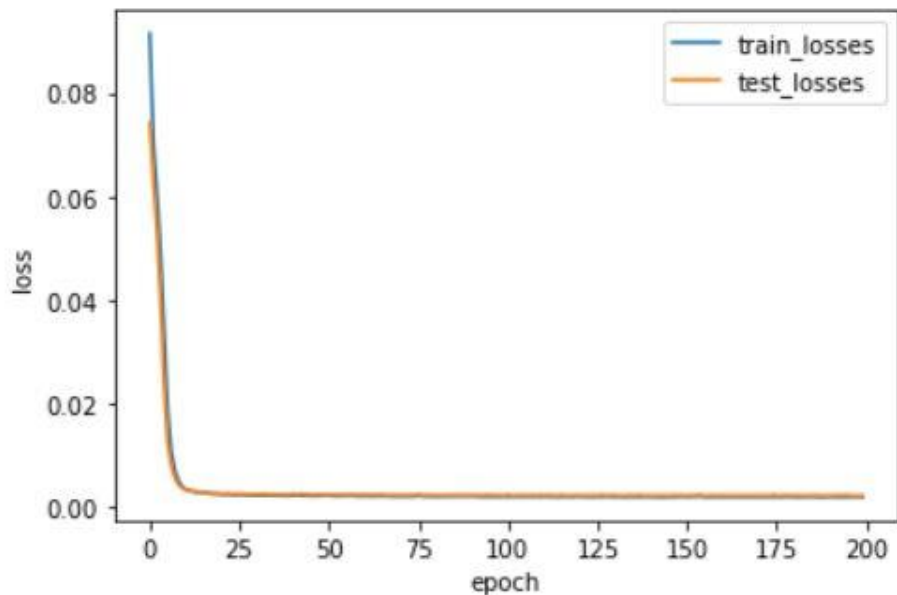
```
class MLP(nn.Module):  
    def __init__(self):  
        super().__init__()  
        self.fc1 = nn.Linear(7, 10, bias=True)  
        self.fc2 = nn.Linear(10, 10, bias=True)  
        self.fc3 = nn.Linear(10, 10, bias=True)  
        self.fc4 = nn.Linear(10, 1, bias=True)  
  
    def forward(self, x):  
        x = F.relu(self.fc1(x))  
        x = F.relu(self.fc2(x))  
        x = F.relu(self.fc3(x))  
        x = F.relu(self.fc4(x))  
        return x
```

4. 구현 절차

- 1 데이터를 가져와서 MinMaxScaler()를 통해 정규화 진행
- 2 전체 데이터 중 훈련 데이터는 70%, 학습 데이터는 30%로 나눔
- 3 훈련 데이터를 구현한 MLP 모델에 적용해 훈련을 진행
- 4 각 파라미터를 Loss 값을 많이 줄이는 방향으로 계속 조정하여 학습을 진행

결과

```
plt.plot(train_losses, label='train_losses')
plt.plot(test_losses, label='test_losses')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend()
plt.show()
```



Batch_size	Learning rate	Epochs	hidden layer	Activation function
128	1e-5	200	2	ReLU

```
print("MSE : %.3f, RMSE : %.3f" % (test_mse, np.sqrt(test_mse)))
```

MSE : 0.002, RMSE : 0.045

Mse 값이 0.002로 높은 정확도를 보였습니다.

그래프에서는 epoch가 많아질수록 loss값이 줄어드는 것을 보아 학습이 잘 되었다는 것을 알 수 있습니다.

*epoch : 데이터셋에 대하여 학습을 한번 완료한 상태

5. 향후 계획

- 데이터가 부족하여 생각한 것 보다 만족스럽지는 않았다.
→ 기회가 된다면 여러 지역의 태양광 발전소들의 데이터를 가지고 연구할 계획
- 가지고 있는 데이터 특성상 시계열 데이터의 특징을 갖기 때문에 시계열 모델 예측에 적합한 RNN과 LSTM 모델을 이용한 구현을 연구할 계획

감사합니다.

