



빅데이터를 이용한 전력 수요 예측

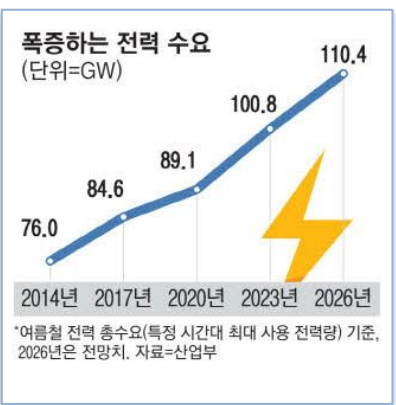
요약 설명

시간이 지날 수록 전력에 대한 수요는 계속 늘어난다. 전력수요가 늘면 발전소에서 전력을 더 많이 생산해야하는데 전력을 많이 만들고 사용하지 않으면 전력을 보관하기가 힘들어 손실되는 전력이 생기게 된다. 남은 전력량이 많을 수록 더 많은 손실이 일어난다. 그렇기에 전력을 정확히 필요한 양만큼 만들어야 손실전력 없이 에너지를 절약하며 비용 절감, 환경 보호의 효과를 얻을 수 있다. 그래서 본 팀은 이러한 장점을 얻기 위해서 기온, 강수량, 습도, 건물의 종류 등을 고려하여 일별, 월별, 년별로 시간에 따른 한 도시에 필요한 전력의 양을 예측해보려 한다.

출품자:
이○○(23○○)
이○○(23○○)
이○○(23○○)

연구 내용

1. 주제선정 및 목적



전력 손실 **다**

전력량 예측

생산량 조절 → 저장전력 최소화

에너지 효율성 향상
비용절감, 환경보호

2. 연구 내용 및 방법

1. 기존 데이터 확인



-전력 소비량 분포

2. 전력 수요 예측

1. LSTM 모델 설계

*시계열 예측 모델

```
class LSTM(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, output_size):
        super(LSTM, self).__init__()
        self.hidden_size = hidden_size
        self.num_layers = num_layers
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers, batch_first=True)
        self.fc = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        h0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(x.device)
        c0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(x.device)
        out, _ = self.lstm(x, (h0, c0))
        out = self.fc(out[-1, :])
        return out
```

-LSTM 모델 설계

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print('current device: ', device)
model = LSTM(input_size, hidden_size, num_layers, output_size).to(device)
criterion = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
```

```
for epoch in range(num_epochs):
    for i, (inputs, labels) in enumerate(train_loader):
        inputs = inputs.to(device)
        labels = labels.unsqueeze(1).to(device)

        # Forward
        outputs = model(inputs)
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if (i+1) % 300 == 0:
            print('Epoch [{}/{}], Step [{}/{}], Loss: {:.4f}'.format(epoch+1, num_epochs, i+1, len(train_loader), loss.item()))
```

-LSTM 모델 학습

3. 연구 결과

Epoch [1/5], Step [300/3188], Loss: 0.0005
Epoch [1/5], Step [600/3188], Loss: 0.0006
Epoch [1/5], Step [900/3188], Loss: 0.0003
Epoch [1/5], Step [1200/3188], Loss: 0.0019
Epoch [1/5], Step [1500/3188], Loss: 0.0165
Epoch [1/5], Step [1800/3188], Loss: 0.0013
Epoch [1/5], Step [2100/3188], Loss: 0.0000
Epoch [1/5], Step [2400/3188], Loss: 0.0001
Epoch [1/5], Step [2700/3188], Loss: 0.0001
Epoch [1/5], Step [3000/3188], Loss: 0.0026
Epoch [2/5], Step [300/3188], Loss: 0.0005
Epoch [2/5], Step [600/3188], Loss: 0.0002
Epoch [2/5], Step [900/3188], Loss: 0.0001
Epoch [2/5], Step [1200/3188], Loss: 0.0009
Epoch [2/5], Step [1500/3188], Loss: 0.0072
Epoch [2/5], Step [1800/3188], Loss: 0.0002
Epoch [2/5], Step [2100/3188], Loss: 0.0000
Epoch [2/5], Step [2400/3188], Loss: 0.0002
Epoch [2/5], Step [2700/3188], Loss: 0.0001
Epoch [2/5], Step [3000/3188], Loss: 0.0006
Epoch [3/5], Step [300/3188], Loss: 0.0004
Epoch [3/5], Step [600/3188], Loss: 0.0001
Epoch [3/5], Step [900/3188], Loss: 0.0000
Epoch [3/5], Step [1200/3188], Loss: 0.0003
Epoch [3/5], Step [1500/3188], Loss: 0.0015
Epoch [3/5], Step [1800/3188], Loss: 0.0001
Epoch [3/5], Step [2100/3188], Loss: 0.0000
Epoch [3/5], Step [2400/3188], Loss: 0.0000
Epoch [3/5], Step [2700/3188], Loss: 0.0000
Epoch [3/5], Step [3000/3188], Loss: 0.0001
Epoch [4/5], Step [300/3188], Loss: 0.0001
Epoch [4/5], Step [600/3188], Loss: 0.0001
Epoch [4/5], Step [900/3188], Loss: 0.0001
Epoch [4/5], Step [1200/3188], Loss: 0.0002
Epoch [4/5], Step [1500/3188], Loss: 0.0010
Epoch [4/5], Step [1800/3188], Loss: 0.0000
Epoch [4/5], Step [2100/3188], Loss: 0.0000
Epoch [4/5], Step [2400/3188], Loss: 0.0000
Epoch [4/5], Step [2700/3188], Loss: 0.0000
Epoch [4/5], Step [3000/3188], Loss: 0.0001
Epoch [5/5], Step [300/3188], Loss: 0.0001
Epoch [5/5], Step [600/3188], Loss: 0.0001
Epoch [5/5], Step [900/3188], Loss: 0.0001
Epoch [5/5], Step [1200/3188], Loss: 0.0002
Epoch [5/5], Step [1500/3188], Loss: 0.0008
Epoch [5/5], Step [1800/3188], Loss: 0.0000
Epoch [5/5], Step [2100/3188], Loss: 0.0000
Epoch [5/5], Step [2400/3188], Loss: 0.0000
Epoch [5/5], Step [2700/3188], Loss: 0.0000
Epoch [5/5], Step [3000/3188], Loss: 0.0000
Epoch [5/5], Step [3000/3188], Loss: 0.0001

num_date_time	answer
0	1_20220825 00 492.777998
1	1_20220825 01 508.540358
2	1_20220825 02 507.235912
3	1_20220825 03 520.526870
4	1_20220825 04 579.702946

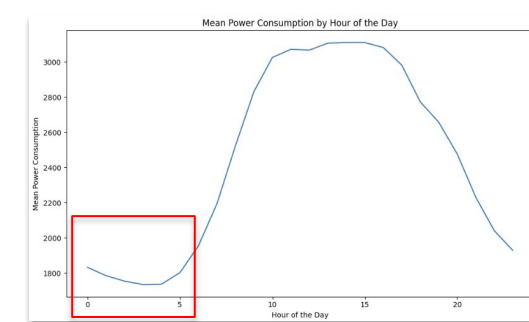
8월 25일 예측 결과

시간	전력소비량 (kWh)
0	422.88
1	386.16
2	363.6
3	353.28
4	356.88

실제 8월 25일 전력

아쉽다!!

4. 결론 및 적용



0시부터 4시는 전력 사용량이 감소하는 것을 알 수 있다.

위 시간 구간에서 전력 수요 증가한다고 예측

아쉽다!!

효과 및 전망

전력 사용량 예측 모델을 이용함으로써 효율적인 인공지능 알고리즘을 바탕으로 안정적인고 효율적인 에너지 공급을 위한 최소한의 전력 생산량에 대해 정확하게 예측을 할 수 있다. 이를 통해 각 도시별로 특정 건물의 수와 구조, 그 도시의 평균 강수량, 평균 기온 등을 분석하여 한 도시의 평균 소비전력량을 구할 수 있을텐데 이를통해 우리가 생각했던 에너지 절약 및 비용절감, 환경보호 측면에서 좋은 효과를 보일 수 있을 것이다. 그리고 전망 면에서는 스마트 그리드 기술이라는 전력 공급과 에너지 수요를 효율적으로 관리하기 위해 정보 및 통신 기술을 활용하는 혁신적인 에너지 공급체계가 발전하여 실시간 전력 관리 및 최적화가 가능하게 되어 전체적인 전력 시스템의 효율성을 높일 수 있을 것이다. 또 인공지능 및 빅데이터에 대한 활용도 증가하여 정확한 결과값을 모르는 상황에서 평균값을 통해 분석할 수 있는 알고리즘을 이용해 우리 주제와 또 다른 여러 분야의 산업도 적용 가능할 것이라 생각하고, 이를 적용한다면 에너지 측면에서 큰 영향이 있을 것이라 생각한다.