LSTM을 활용한 공급망 최적화

산업경영공학과 2018100930 전계범

목차

- ✓ 분석 목적
- ✓ 현재 기술 수준 및 동향
- ✓ 적용가능한 딥러닝 기법 및 모델 설계 제시
- ✓ 데이터 수집 방안
- ✓ 딥러닝 모형 개발 및 실험
- ✓ 실험 결과 분석
- ✓ 개선 방안 제시

분석 목적

- 모기업에 부품을 납품하는 협력기업은 납품된 부품의 사용량을 알 수 없기 때문에 경험을 기반으로 생산 계획을 수립
- 이로 인해 생산 계획 및 재고 관리에 어려움 발생(발주 수량 예측 실패로 인한 재고량 초과 및 부족)
- 이러한 비효율적 생산 체계로 인해 모기업의 생산량 및 발주 수량의 잦은 변동에 대처하는 것에 어려움 발생
- 결과적으로 협력사(공급사)에 재고 부담을 발생시켜 경영 악화를 야기

따라서, 발주 수량을 미리 예측할 수 있는 AI 기반 수량 예측 시스템이 필요

현재 기술 수준

VMI(Vendor Management Inventory)

- 납품업자가 모기업을 대신해 최소 및 최대 재고 수준에 기초해 재고를 모니터링하고 계획하는 공급망 관리 기법
- 실수에 민감하거나 경쟁이 심하고 이윤이 적은 산업이나 수요를 예측하기 어려운 제품 제조 공정에 효과적으로 활용
- 공급망 관점에서는 전체 공급망 수준 대비 낮은 수준의 재고 유지가 가능
- 작업자의 자료 입력에 따른 오류 감소
- 공급자 관점에서는 제품 항목 관리 및 재고 보충 시간의 감소 등으로 인한 전략적 공급망 관계 구축 가능

최신 기술 동향

- 과거에는 재고관리나 생산량 예측에는 ARIMA(Auto-Regressive Integrated Moving Average)와 같은 시계열 분석법과 베이지안 선형 회귀 같은 회귀 분석법을 활용
- 하지만, 최근 의류 제조사, 카드사 등의 업체에서 인공신경망을 도입함에 따라 시계열 데이터 분석을 통해 수요 예측이 가능해짐으로써 예측 성능을 크게 향상 시키는 것이 가능
- 이와 같이 AI 기반 분석모델을 모기업의 발주 수량 데이터에 적용할 경우 VMI 및 재고 관리 최적화에 큰 도움

RNN(Recurrent Neural Network)

- 다층 퍼셉트론(MLP)와 달리 여러 층의 완전 연결층을 가지지 않고 하나의 은닉층을 가지며, 은닉 노드 간 에지(연결)가 존재
- 순환신경망은 시계열적 특성을 지닌 데이터를 활용해 예측하거나 분류하는 문제를 해결하는 데에 사용
- 순환신경망의 특징은 특정 시점에 받은 입력이 은닉상태에 영향을 주는데, 이전 시점의 은닉상태는 다음 시점의 은닉상태에 영향
- 순환신경망은 은닉층은 모든 시점의 입력을 같은 비중으로 기억
- 하지만, 은닉층과 노드 수는 한정되어 있어 기억력에 한계
- 시간의 흐름에 따라 초기 입력에 대한 기억이 희미해지고 기울기 소실 및 폭발 문제 발생

LSTM(Long-Short Term Memory)

- LSTM은 순환신경망의 일종으로서, 순환신경망의 한계점과 문제점을 극복하기 위해 입력, 출력, 망각 게이트를 도입해 선별적으로 특징을 기억

- 망각 게이트 : 이전 단계의 메모리 셀의 과거 정보를 얼마나 삭제할지 결정
- 입력 게이트 : 현재 단계에서 새로운 정보를 얼마나 추가할지 결정
- 출력 게이트 : 계산된 메모리 셀의 정보를 활용해 최종 출력인 은닉 상태를 제어

알고리즘 구축 절차

1. 데이터 전처리 과정

: 수집된 데이터에 대해 정제 및 필터링 진행

2. 모델 정의

: 적합한 초매개변수(hyper parameter)를 설정

3. 모델 학습

: 데이터셋을 학습(Train), 검증(Validation), 평가(Test) 데이터로 분리해 학습

4. 모델 평가

: 최종적으로 학습이 종료된 모델을 저장하고 성능 평가

모델 설계 및 구조

Input shape : (3, 4)

- 2개의 LSTM 은닉층 사용
- 출력층은 Fully-connecter Layer(완전 연결층)인 Dense 사용
- 은닉층의 활성함수는 ReLU함수 사용
- 각 은닉층은 8개의 은닉 노드로 구성
- 출력층의 활성함수는 Linear함수 사용
- 분류문제가 아닌 회귀문제이기 때문에 출력의 결과는 발주 수량의 예측값

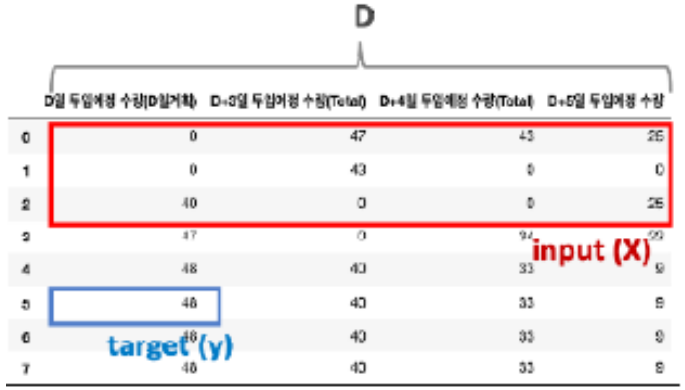
데이터 수집 방안

VMI 재고 관리 방식을 사용하는 협력기업과 모기업의 발주 수량 정보 등을 일자 및 시간별로 수집한 데이터

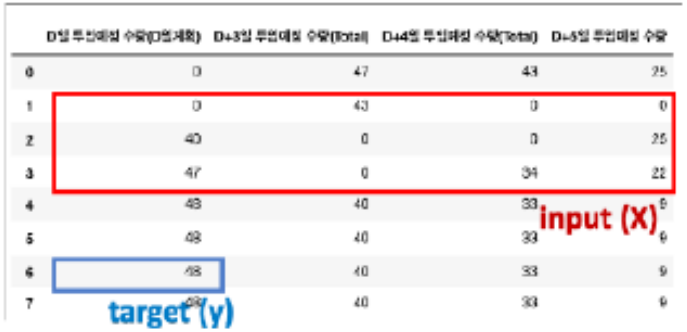
- ERP(Enterprise Resource Planning) 시스템이 21년 9월 13일부터 21년 11월 1일까지 평균 10시간 간격으로 수집한 데이터
- 84개 column, 17364개 row, 총 1458576개의 데이터
- 데이터셋에는 일자, 시간 및 부품 식별자와 다양한 발주 수량 정보가 포함

딥러닝 모형 개발 및 실험

- D(당일 발주 수량)과 D+3, D+4, D+5일의 투입 예정 수량을 활용해 재고 관리 최적화를 위한 발주량 예측 모델을 설계
- 과거 3일 치에 해당하는 수량 데이터(D+3 ~ 5일 투입 예정 수량)를 활용하여 3일 후의 발주 수량을 예측
- 전체 시계열 데이터를 LSTM의 입력을 위해 3D 텐서 형태로 변환
- Input X : (44, 3, 4), target y : (44,)



<첫번째>



<두번째>

딥러닝 모형 개발 및 실험

- 총 44개의 데이터셋을 training(70%), validation(10%), test(20%) 데이터로 분할

  training : 모델 학습 시에 사용

  validation : 훈련된 모델의 성능 측정에 사용(overfitting 방지)

  test : 모델의 성능을 최종적으로 판단

- 모델 학습 시 값의 범위가 달라 변수 별 편향 발생 방지 및 학습 속도 개선을 위해 전체 데이터를 표준화함으로써 평균이 0, 분산이 1인 정규분포를 따르도록 변환 (scikit-learn에서 제공하는 표준화 스케일러를 활용)

딥러닝 모형 개발 및 실험

<분석 모델 구축>

- 2개의 LSTM 층과 1개의 완전연결층 사용
- dropout rate(0.2) : 확률 0.2만큼의 은닉 노드를 사용하지 않게 되고, 역전파 알고리즘 수행 시 사용하지 않은 노드에 대한 gradient가 반영되지 않는다. 모델의 학습 과정에서 매 연산마다 은닉 노드 중 어느 하나에 전적으로 의존하지 않게 되므로 overfitting 문제를 해결
- activation function(ReLU) : 입력된 데이터의 가중 합을 출력 신호로 변환하는 함수
- 기울기 소실 문제를 해결 가능한 ReLU 함수 사용

딥러닝 모형 개발 및 실험

```
model = Sequential()
model.add(LSTM(8, dropout=0.2, activation='relu', input_shape=(3,4), return_sequences=True))
model.add(LSTM(8, dropout=0.2, activation='relu'))
model.add(Dense(1, activation='linear'))

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 3, 8)              416
_________________________________________________________________
lstm_1 (LSTM)                (None, 8)                 544
_________________________________________________________________
dense (Dense)                (None, 1)                 9
=================================================================
Total params: 969
Trainable params: 969
Non-trainable params: 0
_________________________________________________________________
```

딥러닝 모형 개발 및 실험

- loss function : 실제 값과 예측 값의 차이를 계산
- 평균 절대 오차(Mean Absolute Error, MAE) 사용

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

- optimizer : Adam(Adaptive Momentum Estimation) 사용 – RMSProp 방법에 모멘텀을 추가 적용
- Epoch(100) : 모델이 전체 데이터를 순회하며 학습한 횟수
- batch size(4) : 1번의 parameter update 시 사용하는 데이터의 개수
- Early stopping : validation loss가 최저인 시점부터 patience(15)의 수 만큼 학습이 진행되는 동안 개선되지 않으면 stop

```
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

model_path = './model/{epoch:02d}-{val_loss:.4f}.hdf5'
callbacks = [EarlyStopping(monitor='val_loss', patience=15),
             ModelCheckpoint(filepath=model_path, monitor='val_loss', verbose=0, save_best_only=True)]

history = model.fit(X_train_94, y_train_94, epochs=100, batch_size=4, validation_data=(X_val_94, y_val_94),
callbacks=callbacks)
```

딥러닝 모형 개발 및 실험

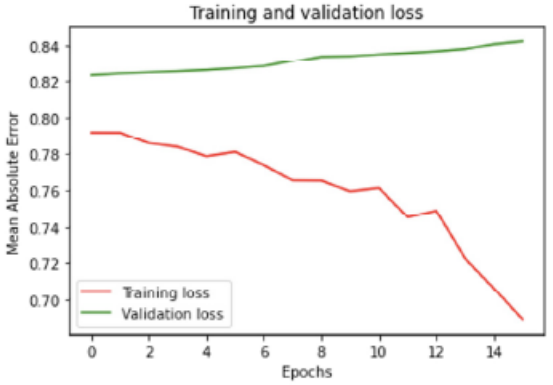
<학습 결과 확인>

validation loss는 epoch이 증가함에 따라 증가하지만, training loss는 epoch이 증가함에 따라 감소한다.
즉 validation loss가 최소인 지점 이후로 early stopping이 일어나기 전까지는 overfitting 되었다고 판단할 수 있다.

```
loss = history.history['loss']
val_loss = history.history['val_loss']
epoch = history.epoch

plt.figure()
plt.plot(epoch, loss, 'r', label='Training loss')
plt.plot(epoch, val_loss, 'g', label='Validation loss')

plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Mean Absolute Error')
plt.legend()
plt.show()
```



실험 결과 분석

< 최종 성능 확인>

- predict() 메서드를 사용해 모델의 예측 결과 확인
- 최종 출력 형태 확인을 위해 표준화한 값을 복원(inverse scaling)
- 최종 성능 출력을 위해 손실 함수로 쓰였던 MAE를 계산
- 처음 과정이 부품 94에 대한 발주 수량 모델이라면, 부품 95라는 94와 연관성이 있는 두 개의 부품의 발주량 데이터를 활용해 동일한 과정으로 부품 94의 발주 수량 모델 구축

예측 성능	부품 94의 발주데이터만 사용	부품 94와 부품 95의 발주데이터 모두 사용
MAE Loss	1.2037	1.0878
MAE (inverse scaling)	5.4498	4.9255

실험 결과 분석

- 위의 결과를 보면 동일한 제품 생산에 사용된 부품들이 서로 발주 수량에 연관성이 있을 것이라는 가정 하에 해당 부품들의 데이터를 모두 활용해 예측하는 것이 더 좋은 결과를 도출할 수 있을 것이라고 예상된다.
- 따라서 수집된 데이터가 많을수록 정확한 예측 모델을 구축할 수 있다.

개선 방안 제시

- 일반적으로 데이터를 충분히 확보했을 때 성능이 향상된다.
- 활용된 데이터는 총 44개로 적은 편이다. 따라서, 좀 더 많은 데이터를 수집해 모델 학습에 활용한다면 향상된 결과를 예측하는 것이 가능하다.
- 또한, 발주에 영향을 미치는 다양한 변수 데이터를 활용한다면 좀 더 정확한 예측 모델 구축이 가능할 것이다.

감사합니다.