



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

박사학위 청구논문

지도교수 김경환

딥러닝 및 관련 기법을 활용한 철근
가격 장단기예측

2022년 2월

건국대학교 대학원

건축학과

이용성

딥러닝 및 관련 기법을 활용한 철근 가격 장단기 예측

Long and Short Term Prediction of Rebar
Price Using Deep learning and Related
Techniques

이 논문을 공학 박사학위 청구논문으로 제출합니다.

2021년 11월

건국대학교 대학원

건축학과

이용성

이용성의 공학 박사학위 청구논문을 인준함.

심사위원장 _____ (인)

심사위원 _____ (인)

심사위원 _____ (인)

심사위원 _____ (인)

심사위원 _____ (인)

2021년 12월

건국대학교 대학원

목차

표 목차	iv
그림 목차	vi
ABSTRACT	viii
제1장 서론	1
1.1 연구의 배경 및 목적	1
1.2 연구의 범위 및 방법	4
1.3 국내외 연구 현황	6
1.3.1 딥러닝 관련 연구 동향	6
1.3.2 공사비 예측 관련 연구 동향	12
제2장 딥러닝 관련 이론적 고찰	15
2.1. 딥러닝 모델 관련 고찰	15
2.1.1 딥러닝	15
2.1.2 딥러닝 학습방식	19
2.2 딥러닝을 활용한 시계열 예측 관련 고찰	20
2.2.1 시계열 데이터	20
2.2.2 데이터 조합탐색	20
2.2.3 하이퍼 파라미터 최적화	21
2.2.4 과소적합 및 과대적합	21
2.2.5 드롭아웃	23
2.2.6 순환신경망 모델의 성능검증 방법	23
2.2.7 데이터 정규화	25
2.2.8 모델 성능평가	26
제3장 철근 가격예측모델 환경구성	27
3.1 데이터셋 구성	27
3.2 순환신경망 모델 구성	32

제4장 순환신경망을 활용한 철근 가격 단기예측	35
4.1 철근 가격데이터에 적합한 순환신경망 모델 선정	35
4.1.1 개요	35
4.1.2 순환신경망의 단기 예측방법	35
4.1.3 예측 및 검증 방법	37
4.1.4 예측결과 및 고찰	37
4.2 데이터 조합탐색 및 딥러닝 관련 기법을 활용한 단기예측	39
4.2.1 개요	39
4.2.2 Random search	39
4.2.3 예측 및 검증 방법	40
4.2.5 예측결과 및 고찰	43
4.3 소결	47
제5장 순환신경망을 활용한 철근 가격 장기예측	49
5.1 정답데이터의 시점 조정을 통한 예측범위 확대	49
5.1.1 개요	49
5.1.2 순환신경망의 예측범위 확대를 위한 정답데이터의 시점 조정	49
5.1.3 예측 및 검증 방법	51
5.1.4 예측결과 및 고찰	52
5.2 데이터 시간 단위 조정을 통한 철근 가격예측 범위 확대	55
5.2.1 개요	55
5.2.2 데이터 시간 간격 재구성	55
5.2.3 예측 및 검증 방법	57
5.2.4 예측결과 및 고찰	60
5.2.4.1 데이터셋에 적합한 딥러닝 모델 선정	60
5.2.4.2 데이터조합탐색	60
5.2.4.3 예측범위 확대를 위한 데이터의 시간 단위 재구성	61
5.3 소결	65

제6장 Recursive 예측방식을 활용한 철근 가격 장기예측	67
6.1 개요	67
6.2 Recursive 예측방법	67
6.3 데이터 선정	68
6.4 예측 및 검증 방법	68
6.5 예측결과 및 고찰	70
6.6 소결	77
제7장 Recursive 예측방식과 시계열 교차검증을 활용한 장기예측	78
7.1 개요	78
7.2 시계열 교차검증 적용	78
7.3 조기 종료	79
7.4 예측 및 검증 방법	81
7.5 예측결과 및 고찰	82
7.8 소결	90
제8장 결론	91
참고문헌	95
부록	101
국문초록	105

표 목차

<표 1-1> 딥러닝 관련 연구 동향	9
<표 1-2> 공사비 예측 관련 연구 동향	14
<표 3-1> 데이터셋 구성	29
<표 3-2> 피어슨 상관계수 해석표	30
<표 3-3> 시스템 구성 사양 및 사용 패키지 정보	32
<표 3-4> 하이퍼 파라미터 및 설명	33
<표 4-1> 신경망 모델 선정에 사용되는 하이퍼 파라미터 입력범위	37
<표 4-2> 순환신경망 모델탐색 결과	38
<표 4-3> 데이터 특성 조합별 데이터셋 수	40
<표 4-4> 선별식 추정을 위한 예측결과	42
<표 4-5> 하이퍼 파라미터 입력범위	43
<표 4-6> 데이터 조합탐색 결과	44
<표 5-1> 하이퍼 파라미터 입력범위	52
<표 5-2> 정답데이터 시점 변경 결과	52
<표 5-3> 데이터셋 구성	57
<표 5-4> 데이터조합 개수	58
<표 5-5> 하이퍼 파라미터 입력범위	59
<표 5-6> 데이터셋에 적합한 모델탐색 결과	60
<표 5-7> 데이터조합 탐색결과	61
<표 5-8> 데이터 시간 간격 재구성에 따른 예측범위 확대결과	62
<표 6-1> Recursive 예측의 하이퍼 파라미터 입력범위	69
<표 6-2> 모델탐색 결과(실제 예측 시점, 2020. 4)	70
<표 6-3> Recursive 장기예측 결과(학습 및 테스트 성능, RMSE)	72
<표 6-3> Recursive 장기예측 결과(테스트 성능, MAPE)	73
<표 6-4> Recursive 장기예측 결과 (예측값의 정확도, %)	74

<표 7-1> Recursive 예측의 하이퍼 파라미터 입력범위	82
<표 7-2> 철근 가격 단기예측 결과	83
<표 7-3> 수입물가지수 5개월 예측결과	85
<표 7-4> 생산자물가지수 5개월 예측결과	86
<표 7-5> 철근 가격 5개월 예측결과	87

그림 목차

<그림 1-1> 일반 철근 SD 300 가격변화 추이	2
<그림 1-2> 연구 프로세스	5
<그림 2-1> 생물학적 뉴런과 인공신경망 구조	15
<그림 2-2> 인공지능 상호관계 벤다이어그램	16
<그림 2-3> RNN 구조	17
<그림 2-4> 양방향 순환신경망 구조	18
<그림 2-5> LSTM 구조	18
<그림 2-6> 합성 곱 신경망 구조	19
<그림 2-7> 데이터조합 예시	21
<그림 2-8> 모델 복잡도에 따른 훈련과 테스트 정확도 변화	22
<그림 2-9> 신경망 드롭아웃 사용 전후 차이	23
<그림 2-10> K-겹 교차검증 예시	25
<그림 3-1> 일반 철근 가격(SD300), 전년 대비 연평균 변화율(%)	27
<그림 3-2> 철근 가격과 관련한 예측변수 선정을 위한 인과지도	28
<그림 3-3> 피어슨 상관관계 분석결과 시각화	31
<그림 3-4> 딥러닝 모델 구성	34
<그림 4-1> 데이터 재구성과정	36
<그림 4-2> 데이터 재구성 예시	36
<그림 4-3> Bi-LSTM 테스트 구간 시각화	38
<그림 4-4> 드롭아웃 적용 전과 후의 차이	41
<그림 4-5> BFGI 조합 테스트 시각화	45
<그림 4-6> ABCDEFGHI 조합 테스트 시각화	45
<그림 5-1> 정답데이터의 시점 변경	50
<그림 5-2> 데이터 배치 재구성과정 변경(month 추가)	50
<그림 5-3> 정답데이터의 시점 변경 데이터 재구성 예시	51

<그림 5-4> 예측범위 1-month 테스트 구간	53
<그림 5-5> 예측범위 2-month 테스트 구간	53
<그림 5-6> 예측범위 3-month 테스트 구간	54
<그림 5-7> 데이터 시간 재구성 방법	55
<그림 5-8> 데이터 시간 간격 재구성 적용 전과 후	56
<그림 5-9> 3개월 간격 데이터 시간 재구성 결과	57
<그림 5-10> ABF 데이터 1개월 예측 테스트2	61
<그림 5-11> 4개월 예측결과	62
<그림 5-12> ABF 2개월 예측 테스트2 시각화	63
<그림 5-13> ABF 3개월 예측 테스트2 시각화	64
<그림 5-14> ABF 4개월 예측 테스트2 시각화	64
<그림 6-1> 수입물가지수 예측결과 시각화	75
<그림 6-2> 생산자물가지수 예측결과 시각화	75
<그림 7-1> 시계열 교차검증	78
<그림 7-2> 시계열 교차검증적용 예시	79
<그림 7-3> 누적 학습량 증가에 따른 학습과 테스트 성능변화	80
<그림 7-4> 수입물가지수 5개월 예측결과 시각화	88
<그림 7-5> 생산자물가지수 5개월 예측결과 시각화	88
<그림 7-6> 철근 가격 5개월 예측결과 시각화	89

ABSTRACT

Long and Short Term Prediction of Rebar Price Using Deep learning and Related Techniques

Lee, Yong Seung

Department of Architecture

Graduate School of Konkuk University

The price of the building materials such as steel, wood, and insulation have recently soared alongside rises in raw material prices. In particular, the rebar price's frequent fluctuations due to the macroeconomic influences have again hit the highest record of the "rebar shock" in 2008. The increased price causes the same issues as in the past to recur, such as delays and interruptions in construction. It can have an especially adverse impact on small and midsize construction companies that purchase materials under unit price contracts. It impacts small companies more heavily than large enterprises that directly enter into supply contracts with steelmakers and that have various procurement routes. According to the data provided by the Korean Iron & Steel Association, it is believed that the demand for rebar decreased last year due to social distancing caused by the COVID-19 pandemic, thus rebar production plummeted. On the contrary, the real-world rebar price has drastically surged due to supply shortages as the construction industry is recovering faster than expected. However, most of the cause analysis in this was conducted after the occurrence of the situations and thus lacked

predictive capabilities; unpredicted circumstances may continuously arise with unforeseen variables. It is no exaggeration to say that the construction industry is directly exposed to unknown risks as project periods lengthen and are more heavily affected by various social and economic factors than other industries. Therefore, a study was conducted to utilize deep learning that shows better results than human judgment, speed, and accuracy for efficient cost planning in the context of construction project management. Unfortunately, the amount of data that can be used to predict the price of rebar with deep learning is absolutely insufficient compared to other big data-based machine learning. Additionally, such predictions are difficult because rebar prices reflect social and economic circumstances rather than natural and physical understandings of these materials. Therefore, research was conducted to evaluate short-term rebar price predictions through deep learning and to improve prediction accuracy. In addition, considering the duration of the construction project, various methodological attempts were made step-by-step to expand the prediction range. This approach was used because it was necessary to predict the long-term perspective through deep learning. The first attempt to expand the prediction range was to reconstruct a time unit in time series data based on the fact that the short-term prediction range varies according to the time interval of the input data seen in deep learning. However, the prediction range has expanded to a limited extent as input data is lost in proportion to the range increase. Consequently, only a three-month prediction was carried out. The second attempt was a recursive prediction in which short-term predictions were performed by setting all data characteristics in a dataset as the prediction targets and the generated value was then added to the original dataset to predict the next point. This methodology has a feature that the number of objects to be predicted

increases as the number of data characteristics increases; thus, the dataset was configured to include the Producer Price Index (PPI) for rebar, Import Price Index (IPI) for rebar, and rebar price, indicating a high correlation in the data characteristics based on the Pearson Correlation Analysis. The prediction errors started to increase after five months of PPI forecasting, and consequently, the errors in the values of IPI and rebar price also increased after six months. Therefore, it is concluded that the effective prediction up to five months ahead had been achieved. As described above, this study aims to develop a process model for predicting values with deep learning based on small-scale rebar price data and to demonstrate the prediction results. It is expected that the results of this study can be used as basic data for future follow-up studies in related fields.

Keyword : rebar price prediction, deep learning, bidirectional LSTM, recursive prediction

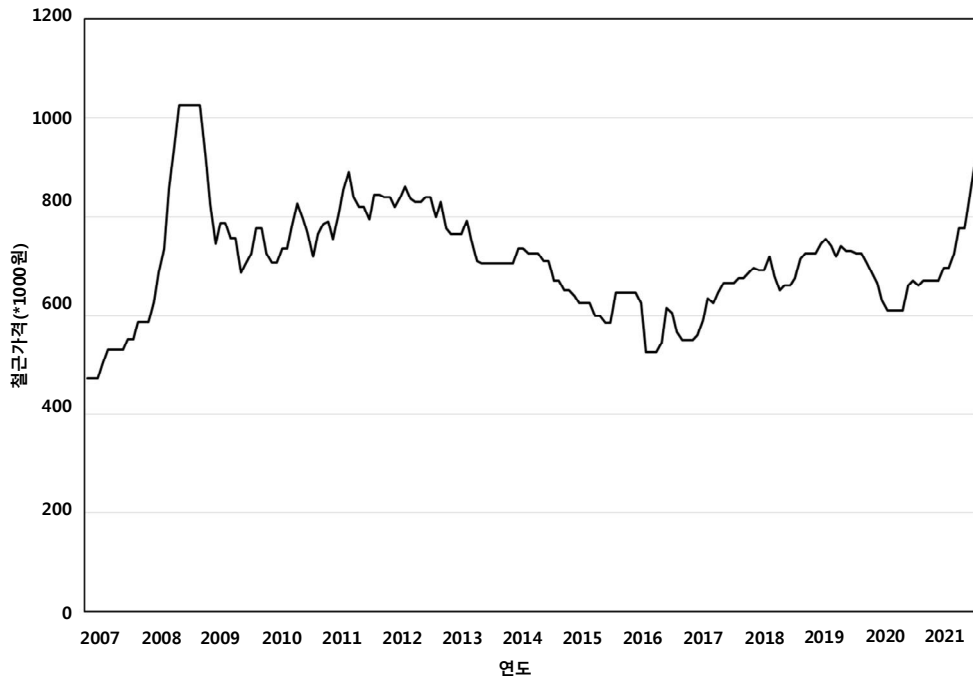
제1장 서론

1.1 연구의 배경 및 목적

건설 프로젝트는 기획부터 시공 완료까지 수차례에 걸친 공사비 예측 및 확인, 정산이 이뤄진다. 기획단계에서는 소요예산을 책정하고, 설계단계에서는 예산의 범위 내에서 효율과 실용적인 대안을 찾기 위한 노력이 이뤄지며, 시공단계에서는 생산활동 이전에 설계도면을 바탕으로 투입될 자재의 물량에 단가의 곱으로 예상되는 비용을 산정한다(지세현 외, 2008). 이 중 시공단계에서 자재에 대한 비용을 산정하는 방식은 계획 당시의 시세를 반영하여 결정하기 때문에 자재의 실제 투입 시점에서 시세의 변동으로 예상했던 가격과 차이가 발생할 수 있다. 특히 재료의 투입 시기가 결정된 이후 물량확보를 위해 선택한 자재의 구매 시기에 시세의 변동으로 단가가 증가하면 계획된 비용 이상으로 지출이 증가할 수 있으며, 이 차이가 클 경우 책정된 예산에 상당한 부담으로 작용할 수 있다. 그리고 이 문제는 시공품질 저하, 공기 지연 등을 유발하며, 공사의 규모와 면적이 클수록 그 부담도 같이 커지게 된다.

특히 건축재료 중 구조체 제작에 필수적으로 사용되는 철근은 국제유가, 환율, 철근 수입물가 등과 같은 거시경제의 영향으로 시세가 달라진다(최열외, 2009). 그리고 <그림 1-1>과 같이 가격변동 폭이 크고 가격인상요인이 있으면, 이를 즉시 시장에 반영할 수 있는 시장 지배력을 가지기 때문에 단가변동이 자주 발생한다. 따라서 채산성을 고려해야 하는 건설기업에서는 철근과 같은 가격 변동성이 큰 자재를 수급하기 위한 계획이 필수적이다(최민수 & 권오현, 2008).

하지만 거시경제의 영향으로 시세가 달라지는 철근의 단가를 반영한 지출비용을 예측하는 기존의 방식은 투입되는 해당 시점 이전 가격의 단순 견적 비교로 결정되기 때문에 기업이윤과 손실에 대한 판단이 어렵다. 그리고 변동성이 큰 자재 단가를 예상하여 비용을 계획하는 일련의 예측과정들은 주문량, 주문 시기에 책정된 단가를 대상으로 관련 비용함수를 최적화하여



<그림 1-1> 일반 철근 SD 300 가격변화 추이

기업이윤을 남기고자 하는 의사결정의 한 범주로 이해할 수 있는데, 결정권자의 경험과 직관에 의존도가 높기 때문에 예측 체계성이 떨어질 수 있다. 따라서 기존의 방식보다 정확한 주문 시기와 이에 책정된 단가의 예측이 일부 가능하다면 불확실성을 가지는 기존의 방식에 예측 체계성을 보완할 수 있을 것으로 사료된다(이재명 외, 2008).

한편 인공지능·경망 기술은 최근 4차 산업혁명의 핵심 기술로 언급되며, 관심이 높아지고 있다. 이러한 계기는 다보스포럼으로부터 시작되었다고 해도 무방하다¹⁾. 2016년 개최된 제46회 다보스포럼에서는 ‘4차 산업혁명의 이해 (Mastering the Fourth Industrial Revolution)’라는 주제로 개최되었다. 이는 글로벌 경제적 위기상황들을 극복할 수 있는 대안으로써 4차 산업혁명으로 인한 사회구조의 혁명적 변화에 주목하였기 때문이다. 그간의 산업혁명은 자동화와 연결성을 발전시켜온 과정으로 축약할 수 있는데, 4차 산업혁명은

1) 다보스포럼은 세계경제포럼(World Economic Forum, WEF)의 클라우스 슈밥(Klaus Schwab) 회장에 의하여 1971년부터 시작된 국제포럼이다. 세계가 직면한 정치·경제·사회적 문제의 해결을 위해 세계 각국의 정상, 장관, 국제기구 수장, 재계 및 금융계 최고 경영자들이 모여 의견을 공유하는 장으로 활용되어왔다.

인공지능에 의해 자동화와 연결성이 극대화되는 단계로 3차 산업혁명의 연장선에 위치하면서 기존 산업혁명과 차별화된 사람의 두뇌를 대체하는 시대의 도래를 포함한다(장필성, 2016). 그리고 인공지능 기술의 한 종류인 딥러닝은 빅데이터를 기반으로 언어, 이미지, 가격예측, 로봇틱스, 무인 차량 등 다양한 분야에서 범용적으로 활용되고 있으며, 이를 통해 인간의 판단과 속도, 정확도를 뛰어넘는 결과를 일부의 영역에서 얻고 있다(Jang & Choi, 2020). 따라서 딥러닝의 적용은 경험과 직관을 기반으로 하는 건설 사업관리 의 일부 전통적인 가격예측방법을 대체할 수 있는 새로운 방법으로 활용 가능할 것으로 사료된다. 또한, 딥러닝은 기존의 회귀 분석모델인 Lasso(Least Absolute Shrinkage and Selection Operator), 서포트 벡터 머신(Support Vector Machine)등과 같은 기계학습 모델보다 예측성능이 우수해, 정확한 예측이 가능할 것으로 판단된다(윤종문, 2019; 이동훈 & 김관호, 2019; 신동하 & 김창복, 2016).

하지만 철근 가격예측을 위해 사용할 수 있는 데이터양은 한국물가정보에서 2000년대부터 월 단위로 제공되기에 대략 240개(20년 * 12개월) 정도로, 다른 빅데이터 관련 연구와 비교해볼 때 많이 부족하다. 그리고 부족한 데이터에 추가로 철근 가격은 사회, 경제적 특성의 반영으로 정확한 예측의 어려움은 가중된다. 또한, 기존의 딥러닝을 활용한 가격예측과 관련한 연구에서 제시하는 예측 시점은 대부분 사용데이터의 시간 간격만큼을 다음 시점으로 예측해 결과로 제시하는데, 건설 프로젝트의 전반적 기간을 고려할 때 철근 가격은 장기적인 시점에 대한 예측이 필요하다.

이에 본 연구에서는 철근의 예상 비용산정과정에서 이뤄지는 결정권자의 경험과 직관을 기반으로 하는 전통적인 예측방식을 대체할 새로운 방법으로 딥러닝을 활용하여 미래 가격변동을 예측하는 방법을 제시하고자 한다. 이와 아울러 확인 가능한 범주에서 제시한 방법을 통해 철근 미래 가격을 장단기예측하고, 실제 가격과 비교하여 제시한 방법의 성능과 예측 가능한 범위를 제시함으로써, 신규 건설사업 수행 시 향후 건축공사에 소요되는 예상 공사비 산정에 도움을 주고자 한다.

1.2 연구의 범위 및 방법

본 연구에서는 거시경제 영향과 가격 인상요인으로 인해 달라지는 철근의 미래 가격을 예측하여 결정권자의 경험과 직관을 기반으로 하는 건설사업관리의 효율적인 자재 비용계획 및 수행을 위해 딥러닝기술을 활용하고자 아래와 같은 절차로 연구를 수행하였다.

(가) 딥러닝을 활용한 가격예측에 관한 기존의 연구 현황

딥러닝을 활용한 가격예측에 관한 연구의 현황을 조사하여, 사용 모델과 시계열 데이터 및 예측성능 향상을 위한 관련 기법에 대해 문헌 고찰을 수행하였다. 또한, 기존의 연구를 통해 딥러닝의 예측범위 확대를 위한 방법에 관하여 문헌 고찰을 수행하였다.

(나) 딥러닝 환경구성

문헌 고찰을 통해 조사된 이론을 토대로 본 연구에서 사용할 딥러닝 모델 알고리즘을 구현하고, 예측을 위한 데이터 특성을 선정하였다.

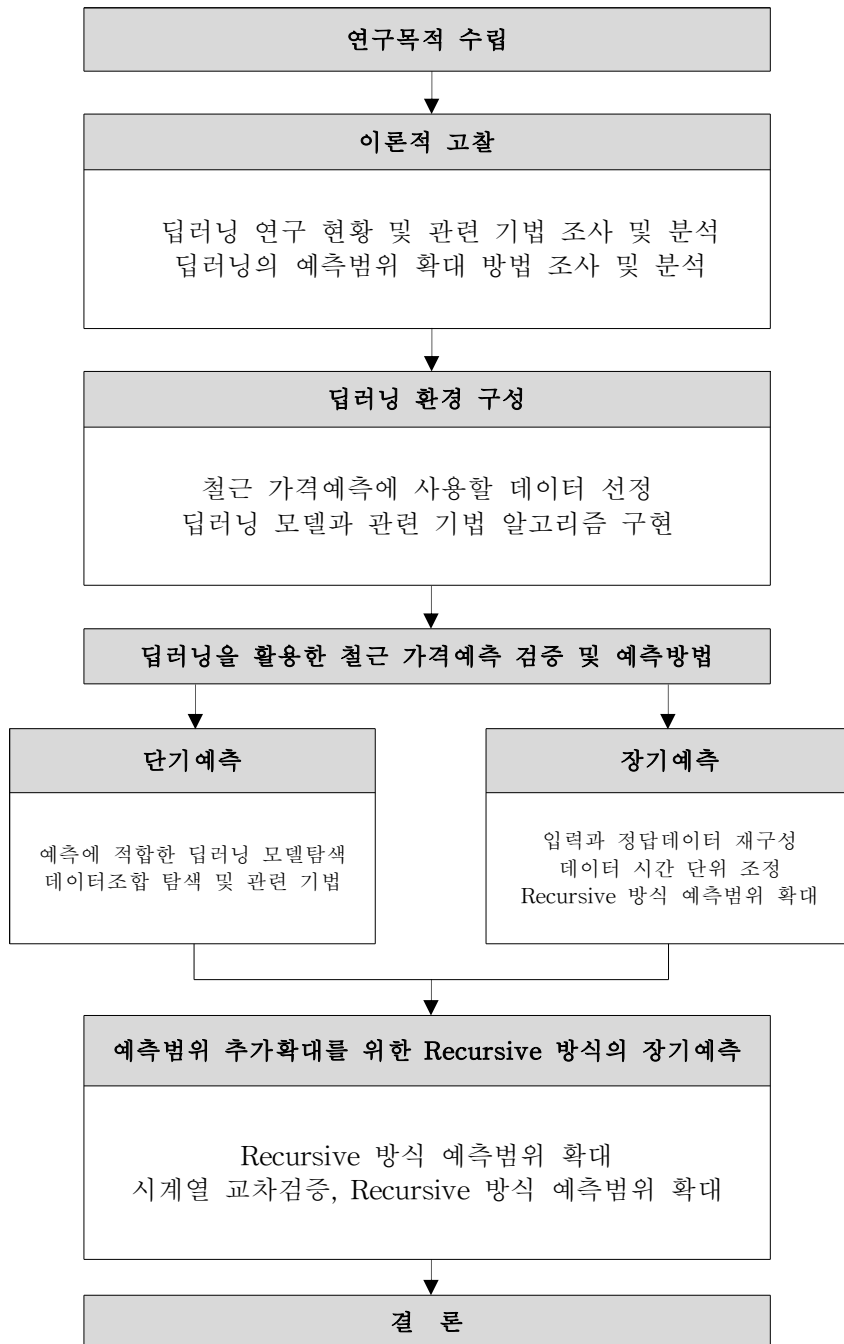
(다) 철근 가격예측에 적합한 딥러닝 모델탐색 및 단기예측

선정된 데이터 특성을 순환신경망 계열의 딥러닝 모델 입력데이터로 사용하여 구성된 데이터 특성에 적합한 딥러닝 모델탐색과 미래의 철근 가격 단기예측을 진행하고, 예측된 가격과 실제값을 비교하여 예측성능을 측정 및 평가하였다.

(라) 철근 가격 장기예측

딥러닝을 활용한 철근 가격예측의 예측범위를 확대하기 위한 여러 방법적 시도를 통해 철근 가격을 장기예측하고, 예측된 가격과 실제값을 비교하여 예측성능을 측정 및 평가하였다.

본 연구의 프로세스를 정리하여 <그림 1-2>와 같다.



<그림 1-2> 연구 프로세스

1.3 국내외 연구 현황

1.3.1 딥러닝 관련 연구 동향

국내외 딥러닝 관련 연구는 특정 값을 예측하기 위해 사용한 데이터와 순환신경망 계열의 딥러닝 모델을 사용하여 구성된 데이터의 시간 간격과 동일한 만큼의 미래값을 단기간 예측하고 결과로 제시하는 연구가 주를 이루고 있다. 그리고 기록된 데이터의 시간 단위에 따라 예측 기간을 일, 월, 연 단위로 제시하는 연구와 기계학습의 여러 모델과 딥러닝에 동일한 데이터를 학습하여 성능을 비교하는 연구도 찾아볼 수 있었다. 또한, 데이터의 시간 단위가 제한적으로 확보 가능할 경우 예측범위를 확대하기 위해 예측된 결과를 데이터에 추가하여 이를 통해 다음의 시점을 예측하는 방식으로 예측범위를 확대하기 위한 관련 연구도 찾아볼 수 있었다.

딥러닝을 활용한 국내외 기존의 연구로 Lahari 외 (2018)과 Cen 외 (2019)는 국제유가인 Brent crude oil, West Texas Intermediate(WTI) 일일 가격, 종가, 개시가 데이터와 딥러닝 모델인 Recurrent Neural Network(RNN), Long Short Term Memory(LSTM) 으로 기름과 원유가격을 예측하였다.

이발엽 외 (2020)과 주일택 외 (2018), Pawar 외 (2019)는 상하이선전 300주식 중 기업 재무지표 분석을 통해 선정된 일 단위 주식데이터와 시가, 고가, 저가, 거래량 등의 주식데이터를 딥러닝 모델인 RNN, LSTM, Bidirectional LSTM으로 주식가격을 예측하여 모델별 예측성능을 비교 제시하였다.

Zhang 외 (2018)과 Mou 외 (2019)는 고속도로에서 수집된 차량 이동데이터와 베이징의 이스트 4차 순환 도로의 남쪽 구간 교통 데이터를 딥러닝 모델인 LSTM을 활용해 도시고속도로에서 이동시간과 단기교통 흐름을 예측하였다.

Chen 외 (2017)은 Beijing, Shanghai, Guangzhou, and Shenzhen의 주택가격 데이터와 딥러닝 모델인 LSTM, Stack LSTM으로 주택가격을 예측하였고, Jang 외 (2020)은 건설시장 및 거시경제 변수와 딥러닝 모델인

RNN, LSTM을 활용해 건설산업 비즈니스 실패율을 연 단위로 예측하였다.

서양모 외 (2019)는 대기오염 측정망을 통하여 측정된 도심지역의 일 단위 데이터와 딥러닝 모델인 LSTM으로 입력된 데이터 시점의 다음 시점인 하루 후를 예측하는 방식으로 미세먼지의 농도를 예측하였다.

배성완 외 (2018)은 기계 학습모델과 LSTM에 한 달 간격의 거시경제 변수를 학습시켜 부동산 가격지수를 예측하고, 모델별 예측성능 비교를 통해 LSTM의 우수성을 입증하였다.

이우창 외 (2020)은 한 달 간격의 거시경제 데이터와 다층신경망 (MLP : Multi layer Perceptron), RNN, 그리고 LSTM을 활용해 입력된 데이터 시점의 다음 시점인 한 달 후를 예측하는 방식으로 철광석 가격을 단기예측하고, 모델별 예측성능 비교를 통해 LSTM의 우수성을 입증하였다.

김준영 외 (2020)은 1달 단위의 콘크리트의 침하데이터와 딥러닝 모델인 LSTM으로 입력된 데이터의 다음 시점을 예측하고 예측값을 다음의 순환신경망의 입력데이터로 재사용하여, 다음 시점을 예측하는 방식으로 콘크리트의 침하 거동을 장기 예측하였다.

전병기 외 (2019)는 1시간 단위 누적에너지 사용량 데이터와 LSTM을 이용하여 재실유무에 대한 정보가 확보되었을 때 다음날의 전기 수요를 예측하였다.

김백천 외 (2020)은 1시간 단위인 태양광 발전량과 기상 데이터, 딥러닝 모델인 LSTM을 이용하여 현시점에서 예측된 출력값을 다음 시점을 예측하기 위한 모델의 입력값으로 사용하는 방식의 재귀과정을 반복해 태양광 발전량을 장기 예측하였다.

이동훈 외 (2019)는 월별, 계절별, 연도별로 정리된 기상요소 데이터로 데이터의 시간 간격에 따라 입력된 데이터의 다음 값을 예측하는 단기예측방식으로 태양광 발전량을 단기, 중기, 장기로 나눠 예측하였다.

최준영 외 (2020)은 시간별 지역 풍력 발전량 데이터와 기상청에서 제공하는 기온, 풍속, 풍향, 습도 등의 데이터와 LSTM을 이용하여 초단기 풍력 발전량을 1시간 단위로 예측하였다.

그리고 딥러닝의 예측 정확도를 높이기 위해 딥러닝 관련 기법을 활용한 연구도 진행되었다. 기존의 연구로는 Zhang 외 (2018)은 딥러닝 모델의

예측성능을 향상을 위해 사용자가 직접 지정하는 하이퍼 파라미터를 자동으로 교차조합하고 최적의 결과를 탐색하는 Grid search를 이용하였다.

Srivastava 외 (2014)는 신경망에 뉴런들을 확률적으로 사용하지 않게 하여 과대적합을 방지하고, 이를 통해 예측성능을 성능을 높이는 드롭아웃(Dropout)을 제안하였다.

또한, 우수한 예측성능을 나타내는 데이터를 선정하기 위해 Jang 외 (2020)과 이종혁 (2019) 그리고 Mou 외 (2019)는 딥러닝 모델에 사용할 후보 데이터를 선정하고, 이를 몇 가지로 조합하여 가장 우수한 예측성능을 나타낸 데이터셋 조합과 결과를 제시하였다.

이상의 선행연구를 분석 요약하면 <표 1-1>과 같다. 분석결과の内容으로는 주택가격, 주식, 국제유가 등과 같이 변동성이 큰 가격을 예측하여 결과로 제시하는 기존의 연구가 있어 철근 가격예측을 위한 방법으로 딥러닝의 활용이 가능할 것으로 판단된다. 그리고 데이터의 시간 단위가 월(month)이나 연(year)일 경우 데이터의 양이 매우 작을 수 있지만, 이를 통해 기계학습과 딥러닝의 학습이 가능함을 알 수 있었다.

또한, 데이터 특성을 조합하는 방법과 Grid search, 드롭아웃 등과 같은 관련 기법을 순환신경망 계열의 딥러닝 모델에 적용하여 구현하면 철근 가격의 단기예측정확도 향상이 가능할 것으로 사료된다. 그리고 순환신경망의 단기 예측범위는 입력데이터의 시간 간격에 따라 결정되는 것으로 파악되며, 예측범위를 확대하기 위해서 딥러닝 학습에 사용할 데이터의 시간 간격을 재구성하는 방식과 확보 가능한 데이터의 시간 단위가 제한적일 경우 예측된 결과를 원 데이터에 추가하고, 이를 통해 다음 예측을 수행하는 반복적인 과정을 통해 예측범위를 확대할 수 있을 것으로 추정된다.

하지만 기존의 연구에서 제시된 방법들을 그대로 반영하여 사용하는 것은 예측대상과 예측을 위해 사용하는 데이터 특성, 딥러닝 모델, 하이퍼 파라미터, 관련 기법 적용방법 등이 모두 상이하기에 우수한 예측결과를 얻을 수 있다고 확신하긴 어렵다. 또한, 이 방법들을 포괄적으로 조합하여 모든 경우를 고려하는 것이 우수한 예측결과를 탐색하기 위한 가장 좋은 방법일 수 있으나, 위에 언급된 방법들을 모두 반영하면, 고려대상 수의

곱만큼 하이퍼 파라미터와 결과를 탐색하기 위한 시간이 급격히 늘어나기 때문에 이를 효율적으로 줄 일 방법이 필요할 것으로 판단된다.

<표 1-1> 딥러닝 관련 연구 동향

분류	연구자	연구 주제	내용
기름 및 원유	Lahari 외 2 (2018)	RNN을 이용한 연료가 격 예측	국제유가인 브렌트유 데이터와 딥러닝 모델인 RNN으로 기름 가격을 예측
	Cen 외 1 (2019)	사전 지식 데이터 전 달을 기반으로 한 장 기 단기 기억 딥러닝 을 이용한 원유 가격 예측 모델	서부텍사스유와 브렌트유의 일 일 가격, 종가, 개시가 데이터 와 딥러닝 모델인 LSTM으로 원유가격 예측
	주일택 외 1 (2018)	양방향 LSTM 순환신 경망 기반 주가예측모 델	시가, 고가, 저가, 거래량 등의 주식데이터를 딥러닝 모델인 LSTM, Bidirectional LSTM으 로 주식을가격을 예측하여 모델 별 예측성능을 비교 제시
주식 가격	Pawar 외 2 (2019)	LSTM RNN을 이용한 주가 예측	주식의 시가, 고가, 저가, 거래 량 등의 데이터와 딥러닝 모델 인 RNN, LSTM을 이용하여 주식가격 예측
	이종혁 (2019)	LSTM과 양방향 순환 신경망을 이용한 주가 예측모델 비교연구	주식가격, 거래량 추세등의 데 이터를 조합하여 데이터셋을 구성하고 양방향 순환신경망을 이용하여 주식가격을 예측
	이발엽 외 1 (2020)	LSTM 인공신경망을 이용한 중국 주가 예 측	주식데이터와 딥러닝 모델인 LSTM을 이용하여 다음 날의 주식가격을 예측

<표 1-1> 딥러닝 관련 연구 동향 (계속)

분류	연구자	연구 주제	내용
풍력 및 태양광 발전량	이동훈 외 1 (2019)	기후 및 계절정보를 이용한 딥러닝 기반의 장기간 태양광 발전량 예측 기법	월별, 계절별, 연도별로 정리된 기상요소 데이터로 데이터의 시간 간격에 따라 입력된 데이터의 다음 값을 예측하는 단기 예측방식으로 태양광 발전량을 단기, 중기, 장기로 나눠 예측
	김백천 외 5 (2020)	기상상태 변동성을 고려한 LSTM 기반 태양광 발전량 예측	1시간 단위인 태양광 발전량, 기상 데이터, 딥러닝 모델인 LSTM을 이용하여 태양광 발전량을 예측
	최준영 외 1 (2020)	시간별 LSTM 기법을 이용한 단기 풍력발전 예측	시간별 지역 풍력 발전량 데이터와 기상청에서 제공하는 기온, 풍속, 풍향, 습도 등의 데이터와 LSTM을 이용하여 초단기 풍력 발전량을 1시간 단위로 예측
고속도로 이동시간 및 교통흐름	Zhang 외 3 (2018)	도시고속도로의 다단계 이동시간 예측을 위한 장기단기역신경망 적용방안 연구	고속도로에서 수집된 차량 이동데이터와 딥러닝 모델인 LSTM, Grid search를 이용하여 도시 고속도로에서 이동 시간을 예측
	Mou 외 2 (2019)	단기교통 흐름 예측: 시간 정보에 의해 향상된 LSTM 모델	베이징의 이스트 4차 순환 도로의 남쪽 구간 교통 데이터와 딥러닝 모델인 LSTM을 이용하여 우수한 예측성능을 나타낸 데이터셋과 예측결과를 탐색하기 위해 선정된 데이터를 몇 가지로 조합하여 가장 우수한 예측성능을 나타낸 결과를 제시

<표 1-1> 딥러닝 관련 연구 동향 (계속)

분류	연구자	연구 주제	내용
콘크리트 침하거동	김준영 외 3 (2020)	순환신경망을 활용한 콘 크리트케도의 장기 침하 거동 예측	1달 단위 콘크리트 침하데이 터와 딥러닝 모델인 LSTM으 로 입력된 데이터 콘크리트 침하 거동 예측
부도율	Jang 외 2 (2020)	LSTM과 RNN을 이용 한 건설업체의 회계, 건 설시장, 거시경제변수를 고려한 사업실패 예측	건설시장 및 거시경제 변수와 딥러닝 모델인 RNN, LSTM 을 이용해 건설산업 비즈니스 실패율을 연 단위로 예측하기 위해 데이터의 특성을 조합하 여 가장 우수한 예측성능을 나타낸 결과를 제시
철광석 가격	이우창 외 3 (2020)	머신러닝을 이용한 철광 석가격 예측에 대한 연 구	한 달 간격의 거시경제 데이 터와 다층신경망, RNN, LSTM을 활용해 철광석 가격 을 단기예측하고, 모델 별 예 측성능 비교를 통해 LSTM의 우수성을 입증
부동산 가격지수	배성완 외 1 (2018)	머신 러닝 방법과 시계 열 분석 모형을 이용한 부동산 가격지수 예측	시계열 모형과 기계학습, LSTM을 통해 부동산가격지 수를 예측하고, 모델별 예측 성능 비교를 통해 LSTM의 우수성을 입증

<표 1-1> 딥러닝 관련 연구 동향 (계속)

분류	연구자	연구 주제	내용
미세먼지 농도	서양모 외 1 (2019)	기상 데이터를 활용한 LSTM 기반 미세먼지 농도 예측 방법 비교	대기오염 측정망을 통하여 측정된 도심지역의 일 단위 데이터와 딥러닝 모델인 LSTM으로 입력된 데이터 시점의 다음 시점인 하루 후를 예측하는 방식으로 미세먼지의 농도를 예측
전기 수요	전병기 외 4 (2019)	딥러닝을 이용한 재실 정보 기반 건물의 전기 수요 예측 모델	1시간 단위 누적에너지 사용량 데이터와 딥러닝 모델인 LSTM을 이용하여 재실유무에 대한 정보가 확보되었을 때 다음날의 전기 수요를 예측
딥러닝 관련 기법	Srivastava 외 4 (2014)	Dropout: 신경망이 과적합되는 것을 방지하는 간단한 방법	신경망에 뉴런들을 확률적으로 사용하지 않게 하여 과적합을 방지하고, 이를 통해 예측성능을 높일 방법인 드롭아웃 제안

1.3.2 공사비 예측 관련 연구 동향

건설 프로젝트는 프로젝트의 기획부터 시공 완료까지 수차례에 걸친 공사비 예측이 이뤄진다. 그리고 이와 같은 공사비 예측과정은 건설공사의 이윤과 손실에 대한 성패를 좌우하는 중요한 작업이기 때문에 예측의 정확성을 위해 여러 방식의 예측모델이 제안되어왔다.

기존의 연구로 Hira 외 (1988)은 실적자료를 이용한 기획·설계단계에서의

단위면적당(m^2) 공사비를 예측하여 상세 단가 견적과 $\pm 10\%$ 의 차이를 나타내는 결과를 제시하였다. 박재형 (2018)은 실적자료를 활용하여 공공건축물 유형별 공사비를 예측하기 위해 공종 별로 평균 공사비를 산출하여 연 면적을 활용한 공사비 예측모델과 콘크리트 물량을 활용한 공사비 예측모델을 통해 공사비를 추정하였다. 예측 정확도는 공종별로 3.53% ~ 36.82%까지 다소 불규칙하게 측정되었으며, 특히 토목, 조경, 콘크리트와 같은 일부 공종들에 포함된 불확실성의 요소로 인해 예측 정확도가 낮게 측정되는 것으로 나타났다.

김진원 외 (2011)은 교육 시설의 공사비를 예측하기 위해 임의 선정된 독립변수와 예측대상인 종속변수를 피어슨 상관관계 분석으로 유의성이 높은 독립변수들을 선택한 뒤, 이를 다중 회귀분석으로 예측하였다. 그리고 입력 방식에 따라 단계적 입력방법을 통해 구축된 회귀모델은 14.29%의 오차를 나타냈고, 모든 변수를 입력하는 방식을 사용한 회귀모델은 11.61%의 오차율을 나타냈다. 박우열 외 (2007)은 프로젝트 초기 단계에서 공동주택 공사비를 예측하기 위해 기계학습의 한 종류인 서포트 벡터 회귀분석을 이용하였다. 사용한 데이터로는 연 면적, 층수, 공사 기간, 공사비 지수 등을 활용하였으며, 오차율은 약 4.85%를 나타냈다.

정승현 외 (2018)은 교육 시설 공사비를 분석 및 예측하기 위해 교육 시설 실적데이터와 딥러닝 모델인 ANN(Artificial Neural Network), DNN(Deep Neural Network), DBN(Deep Belief Network)를 활용하였다. 예측정확도는 ANN, DNN, DBN 순으로 약 5.54%, 3.94%, 2.30%를 나타냈다.

이상의 선행연구를 분석한 결과를 요약하면 <표 1-2>와 같다. 분석결과의 내용으로는 전통적으로 공사비 예측에는 실적자료를 통해 비슷한 규모의 평(m^2) 단가 예측이 주로 이뤄졌다. 하지만 지역과 각 사업의 특징을 무시한 평균 공사비 산출이기 때문에 상세 단가 견적과의 오차율은 상당히 큰 것으로 파악된다. 그리고 기계학습과 딥러닝의 활용으로 이러한 공사비 예측 정확도가 크게 개선된 것으로 파악된다.

따라서 본 연구에서 제안하고자 하는 방법이 공사비를 구성하는 항목 중 가격 변동성이 크고 건축물 생성에 필수적으로 사용되는 철근 가격을

예측하고자 하는 것이기 때문에, 이를 통해 기존의 공사비 예측 체계성 보완 가능한 방법이 될 수 있을 것으로 판단된다.

<표 1-2> 공사비 예측 관련 연구 동향

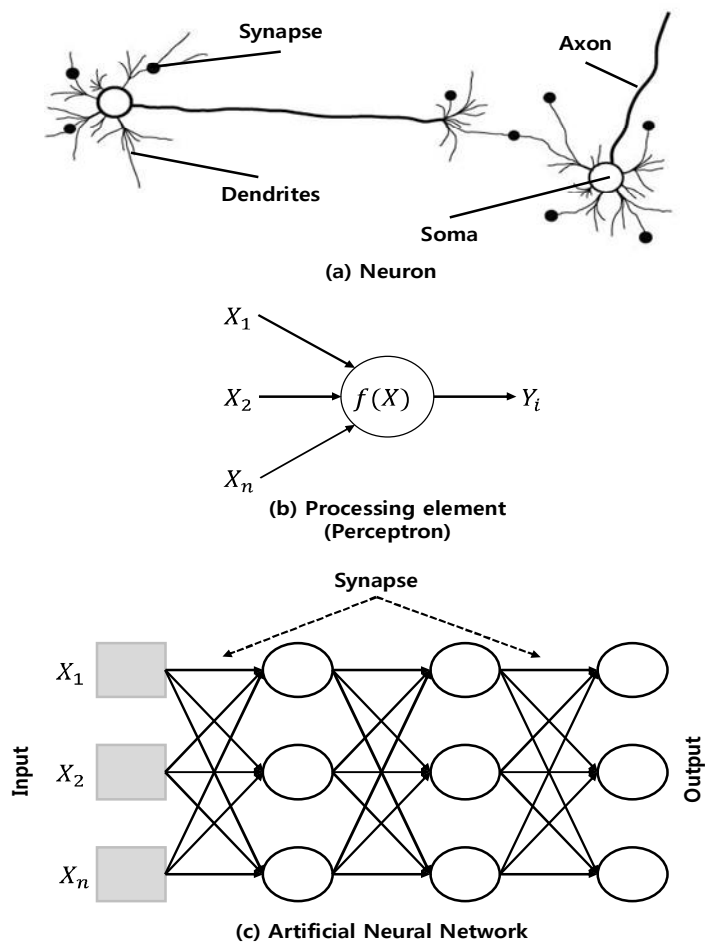
연구자	연구 주제	내용
박재형 (2018)	초기 설계단계의 공사비 예측 및 관리에 대한 실증분석	실적자료와 연면적, 콘크리트 물량을 활용한 공공건축물 공종별 공사비를 예측
정승현 외 2 (2018)	딥러닝을 이용한 스마트 교육 시설 공사비 분석 및 예측	딥러닝 모델인 ANN, DNN, DBN을 활용한 교육 시설 공사비 분석 및 예측
김진원 외 3 (2011)	회귀분석을 이용한 교육 시설의 공간계획에 따른 공사비 예측모델에 관한 연구	다중회귀 분석을 활용한 교육 시설 공사비 예측
박우열 외 1 (2007)	서포트 벡터 회귀분석을 이용한 공동주택 공사비 예측에 관한 연구	SVR을 활용한 프로젝트 초기 단계에서 공동주택 공사비 예측
Hira 외 1 (1988)	Estimating: from Concept to Completion	실적자료를 이용한 기획·설계단계에서의 단위면적당(㎡) 공사비를 예측

제2장 딥러닝 관련 이론적 고찰

2.1. 딥러닝 모델 관련 고찰

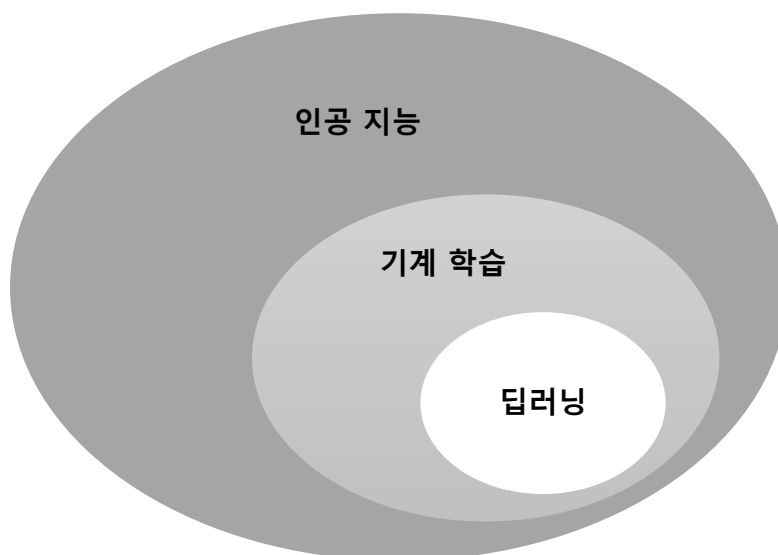
2.1.1 딥러닝

인공지능의 한 분야인 기계학습(ML: Machine Learning)은 인간의 뇌를 모델로 하여 여러 정보를 처리하는 <그림 2-1>과 같은 인공신경망(ANN: Artificial Neural Network)에 기반을 둔다.



<그림 2-1> 생물학적 뉴런과 인공신경망 구조 (마상완, 2007)

기계학습은 컴퓨터가 데이터의 특징을 학습하는 하나의 방법으로, Samuel(1959)은 “코드로 정의되지 않은 동작을 데이터로부터 기계가 학습하여 실행할 수 있도록 알고리즘을 개발하는 연구 분야”로 정의하였다. 그리고 이와 같은 기계학습에 특정한 한 분야에 <그림 2-2>와 같이 딥러닝이 속해있다.



<그림 2-2> 인공지능 상호관계 벤다이어그램²⁾

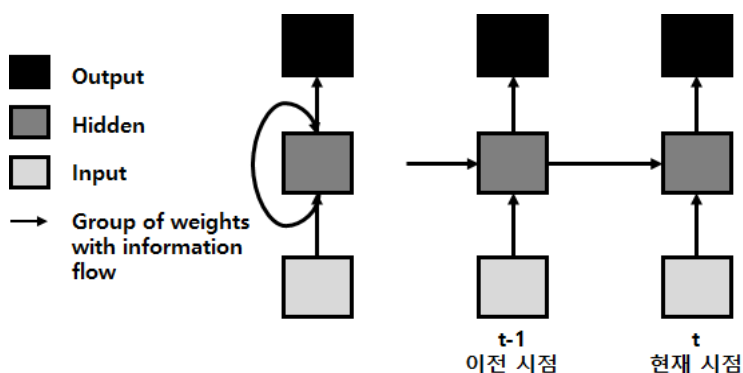
딥러닝은 기계학습과 같이 인간의 뇌를 모델로 하여 여러 정보를 처리하는 인공신경망에 기반을 둔다. 인공신경망은 인간의 신경세포인 뉴런과 유사한 처리 인자들의 병렬 분산에 의해 학습 및 예측을 수행한다. 그리고 지속적인 학습을 통해 연결 강도를 조정하여 새로운 정보를 추가하거나 변경할 수 있다. 또한, 각 신경세포는 완전히 독립된 기능을 갖기 때문에 정보의 분산 표현 및 처리가 가능하다(이철욱, 1993; 손창우, 2004). 그리고 1개에서 2개의 데이터 표현 층을 주로 학습하는 일반적인 기계학습과 달리 딥러닝은 수십개에서 수 백개의 연속 층을 통해 표현을 학습하는 것을 차이점으로 볼 수 있다. 딥러닝은 층 기반 표현학습(Layered

2) Tensorflow. org, 딥러닝이란 무엇인가?, March 16, 2021, accessed <https://tensorflow.blog/케라스-딥러닝/1-딥러닝이란-무엇인가/>.

representations learning) 또는 계층적 표현 학습(Hierarchical representations learning)으로 불리기도 하며 번역, 음성인식, 자율주행차, 로봇틱스 등 다양한 분야에서 활용되고 있다(정동규, 2017). 대표적인 모델로는 순환신경망, 합성곱 신경망 등이 있다.

2.1.1.1 순환신경망

순환신경망(RNN: Recurrent Neural Networks)은 은닉계층에 이전 정보를 기억시킬 수 있는 <그림 2-3>과 같은 순환구조를 이룬다. 그리고 이전 출력의 영향으로 현재까지 계산된 결과를 메모리에 기억한다. 학습은 입력 데이터를 비선형 관계로 변환하여 다음 단계의 입력으로 전달하며 출력값은 $-1 \sim 1$ 사이의 작은 값으로 출력한다(Rumelhart & Williams, 1985; Schuster & Paliwal, 1997). 하지만 순환신경망은 출력값이 곱해지는 단계가 반복되면 기울기 소실(vanishing gradient)이 발생하여 이전 출력값이 소실되어 학습이 이뤄지지 않는다(Gao et al., 2019). 그리고 이로 인해 과거의 학습 결과가 사라지는 장기 의존성(long-term dependencies) 문제를 발생시킨다.

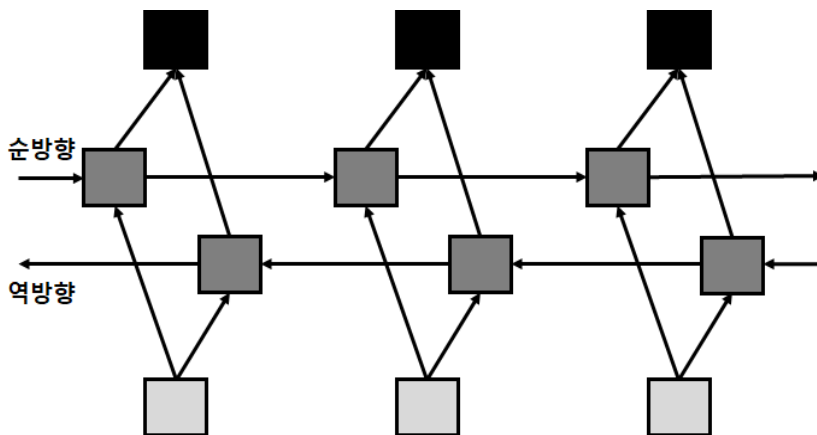


<그림 2-3> RNN 구조 (Schuster & Paliwal, 1997)

(1) 양방향 순환신경망

양방향 순환신경망(BRNN: Bidirectional Recurrent Neural Networks)은 순환신경망의 출력결과가 이전 패턴을 기반으로 하는 경향을 나타내는 한계를 보여, 이를 보완하기 위해 제안되었다(Schuster & Paliwal, 1997). 양방향 순환신경망은 <그림 2-4>와 같이 서로 연결되지 않은 순방향과

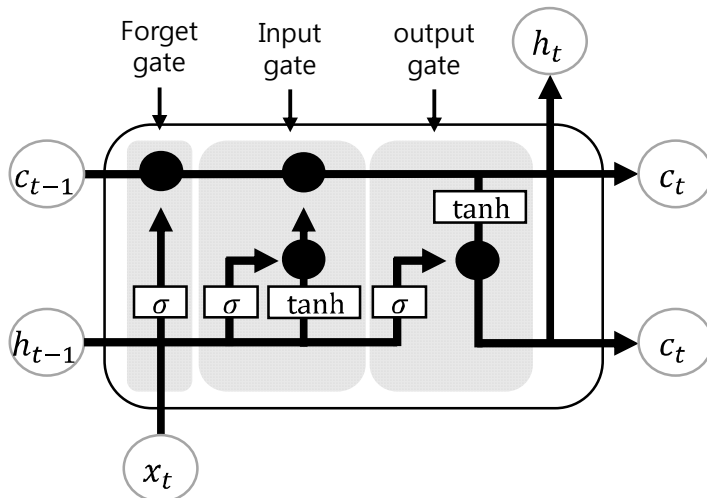
역방향 은닉계층을 가지며, 입력값이 두 개의 은닉계층으로 전달되고 출력계층에서 두 은닉계층의 출력을 하나로 결합해 최종 출력을 한다.



<그림 2-4> 양방향 순환신경망 구조 (Schuster & Paliwal, 1997)

(2) 장단기 메모리

장단기 메모리(LSTM: Long Short-Term Memory)는 순환신경망의 장기 의존성 문제를 극복하기 위해 <그림 2-5>와 같은 구조로 제안되었다(Hochreiter & Schmidhuber, 1997).



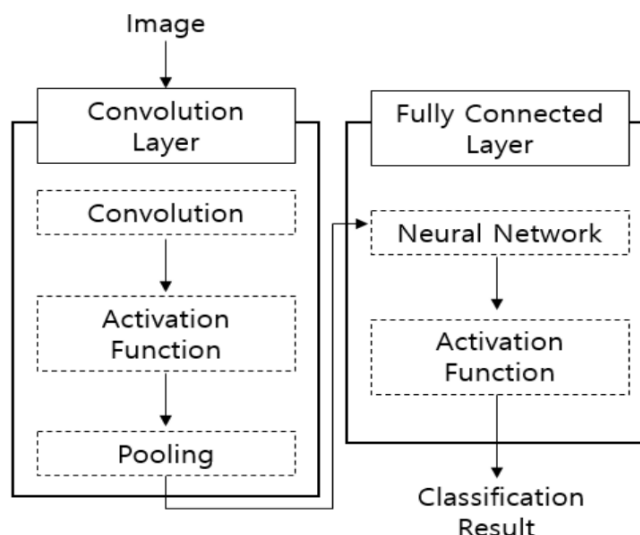
<그림 2-5> LSTM 구조 (권성혁, 2021)

LSTM은 셀 상태(Cell state)를 유지하면서 Input, Output, Forget 게이트라 불리는 구조에 의해 제어된다. 그리고 이를 통해 순환신경망의

장기의존성 문제를 방지하여 시계열 데이터와 같은 방대한 크기의 데이터의 처리에도 기울기 소실 문제를 발생시키지 않는다.

2.1.1.2 합성곱 신경망

합성곱 신경망(CNN: Convolution Neural Network)은 <그림 2-6>과 같이 컨볼루션 층(Convolution layer)과 완전연결계층(Fully connected layer)으로 구성되어있으며, 데이터의 입력을 받아 출력층으로 값을 내보낸다(LeCun et al., 1998). 그리고 다른 신경망 기법들보다 매개변수를 적게 사용하는 점과 쉽게 훈련 가능하다는 장점으로 최근 영상, 음성 분야에서 좋은 성능을 나타내는 것으로 알려져 있다.



<그림 2-6> 합성곱 신경망 구조 (지세현 외, 2019)

2.1.2 딥러닝 학습방식

딥러닝은 지도학습(Supervised learning)과 비지도 학습(Unsupervised learning)으로 구분할 수 있다.

(1) 지도학습

지도학습은 기계학습의 학습방식 중 컴퓨터에 입력된 훈련 데이터를 통해 명시적인 정답이 주어진 상태에서 데이터의 특성을 토대로 컴퓨터를

학습 시킨다(Olson & Delen, 2008). 그리고 세부적으로 연속적인 값을 추출하는 것을 회귀분석(Regression)이라 하며, 입력된 값을 구분하는 것을 분류(Classification)라 한다(김동근, 2020). 지도학습의 대표적 알고리즘은 회귀분석(Regression), 서포트 벡터 머신, 나이브 베이즈 분류(NBC: Naive Bayes Classification) 등이 있다.

(2) 비지도 학습

비지도 학습은 지도학습과 달리 정답이 없는 데이터를 비슷한 특성끼리 군집화(Clustering)하여 새로운 데이터에 대한 결과를 기계가 스스로 학습하는 방법이다. 그리고 세부적으로 분할 기법(Partitioning methods)과 계층적 기법(Hierarchical methods)로 나뉜다(원승현, 2018). 비지도 학습의 대표적 알고리즘은 군집분석(Cluster analysis), 벡터 양자화(Vector quantization), 데이터 차원 축소(Data dimensionality reduction) 등으로 구분한다.

2.2 딥러닝을 활용한 시계열 예측 관련 고찰

2.2.1 시계열 데이터

시계열 데이터는 데이터가 $x_1, x_2, x_3, \dots, x_n$ 까지 있다고 할 때, 각각의 데이터가 특정한 시간 간격으로 측정 및 기록된 데이터 집합을 말한다. 즉 기록 및 관측치가 시간적 순서를 가진다. 시계열 데이터는 과거에 관측된 값을 통해 현재와 미래의 분석 및 예측에 사용되며, 과학, 비즈니스, 경제, 금융, 의료 등 다양한 분야에서 광범위하게 이용되고 있다(Liao, 2005).

2.2.2 데이터 조합탐색

데이터 조합탐색은 측정된 예측결과의 성능개선을 목적으로 사용한다. 방법으로는 <그림 2-7>과 같이 선정된 데이터 특성에 새로운 변수를 추가하거나, 추가한 변수를 조합하는 형태로 데이터셋을 구성한다. 그리고 구성된 여

러 조합의 데이터셋을 사용해 예측결과를 비교한다.

Combination	Input variable groups (size)
Comb_A	Accounting variables (12)
Comb_AC	Accounting variables (12) + construction market variables (3)
Comb_AM	Accounting variables (12) + macroeconomic variables (3)
Comb_ACM	Accounting variables (12) + construction market variables (3) + macroeconomic variables (3)

<그림 2-7> 데이터조합 예시 (Jang & Cho, 2020)

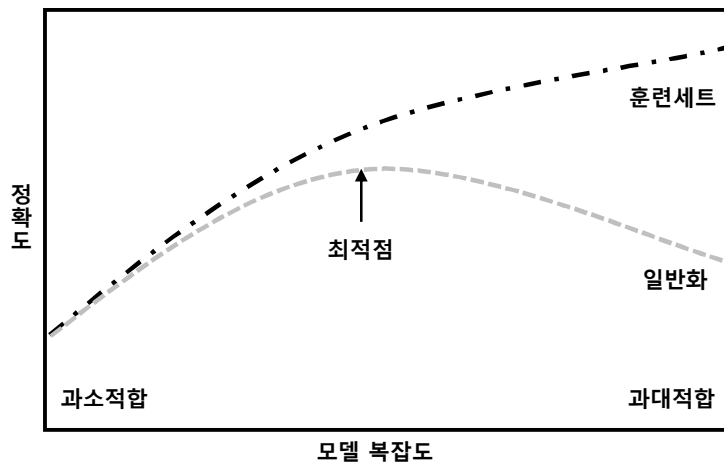
2.2.3 하이퍼 파라미터 최적화

하이퍼 파라미터는 기계학습에서 사용자가 직접 설정하는 모든 값을 지칭한다. 예를 들어 입력데이터 크기(Sequence length), 학습량(Iteration), 학습률(Learning rate), Cell 내부의 은닉층 수(Hidden layer) 등이 있다. 그리고 이 값들은 사용자가 직접 조절하여 설정하는 값의 크기에 따라 모델성능이 달라진다(Claesen & De Moor, 2015). 따라서 모델의 성능을 최대로 끌어올리기 위해서는 하이퍼 파라미터 조합의 탐색이 필요하며, 이 과정을 하이퍼 파라미터 최적화(Hyper parameter optimazation)라 한다. 대표적인 방법들로는 Manual search, Grid search, Random search 등과 같은 방법들이 있다(Bergstra & Bengio, 2012).

Manual search는 시도할 하이퍼 파라미터 후보 입력값을 선정하고, 이 입력값들을 사용하여 실험을 통해 예측결과를 몇 차례 측정한 후 가장 우수한 예측성능을 나타낸 입력값을 채택하는 방법이다. Grid search는 입력값들을 범위로 설정한 후 일정한 간격으로 설정된 범위 내에서 모든 조합을 탐색하는 방법이다. Random search는 Grid search와 마찬가지로 입력값들을 범위로 설정한 후 무작위로 설정된 범위 내에서 조합하여 탐색하는 방법이다.

2.2.4 과소적합 및 과대적합

사용자는 딥러닝 모델의 학습 결과에서 테스트데이터와 실제 데이터를 비교한 성능이 우수할 경우 새로운 데이터에 대한 성능 또한 우수할 것이라 기대한다. 이때 모델이 새로운 데이터에 대해 정확하게 예측한다면 모델의 예측성능이 일반화(Generalization)가 잘 되었다고 판단한다. 하지만 실제 데이터의 예측결과를 측정해보면 예상과 다른 결과가 출력되는 경우가 빈번하다. 이러한 문제는 <그림 2-8>과 같이 과소적합, 과대적합에 의해 발생할 수 있다³⁾.



<그림 2-8> 모델 복잡도에 따른 훈련과 테스트 정확도 변화⁴⁾

(1) 과소적합

과소적합은 모델이 너무 단순하여 입력데이터의 면면과 다양성을 잡아내지 못해 제대로 학습이 이뤄지지 않는 경우 발생한다. 그리고 이는 학습성능이 크게 좋지 않을 경우를 예로 들 수 있다. 이를 해결하기 위해서는 입력데이터에 적합한 딥러닝 모델의 선택이 필요하다.

3) Tensorflow.org. Overfitting and underfitting. accessed March 16, 2021, https://www.tensorflow.org/tutorials/keras/overfit_and_underfit.stand

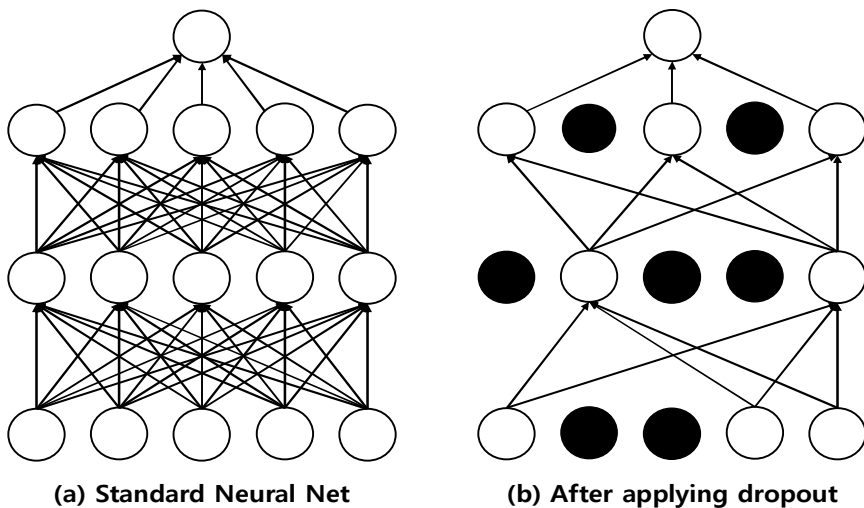
4) Tensorflow.org. 일반화, 과대적합, 과소적합. accessed March 16, 2021, https://www.tensorflow.org/tutorials/keras/overfit_and_underfit.stand

(2) 과대적합

과대적합은 딥러닝 모델이 데이터를 너무 상세하게 학습하는 경우와 학습데이터가 부족한 경우, 데이터의 특성에 비해 모델이 너무 복잡한 경우 발생할 수 있다. 특히 모델이 복잡해지는 경우는 학습성능은 우수하지만, 학습데이터 이외에 다양한 변수에 대한 대응력이 떨어져 일반화된 모델의 예측성능을 감소시킨다. 이 문제를 해결하기 위해서는 과대적합이 되기 전 시점까지 충분히 학습시키는 방법, 드롭아웃, 하이퍼 파라미터 입력값을 낮추는 방법 등의 활용이 필요하다.

2.2.5 드롭아웃

딥러닝 모델을 조합하여 서로 다른 구조를 가지거나, 여러 학습데이터를 통해 모델을 학습시키면 성능을 개선할 수 있다. 그러나 네트워크 훈련에 소비되는 연산 시간이 증가 되어 결과 출력에 많은 시간이 소요되는 문제가 발생한다. 드롭아웃은 여러 모델을 생성하지 않고 모델결합이 여러 형태를 가지게 하여 앞서 설명한 문제와 과대적합을 해결하기 위해 개발되었다. 드롭아웃의 사용방법은 <그림 2-9>와 같이 신경망의 뉴런을 확률적으로 동작하지 않도록 한다.



<그림 2-9> 신경망 드롭아웃 사용 전후 차이 (Srivastava et al., 2014)

2.2.6 순환신경망 모델의 성능검증 방법

순환신경망 모델에 입력데이터는 비율에 따라 학습데이터와 테스트데이터로 분할된다. 이는 모델이 새로운 데이터에 얼마나 일반화가 잘되었는지 판단할 방법으로 테스트 구간을 평가함으로써 오차에 대한 추정값(Estimation)을 얻는다⁵⁾. 이 값은 이전에 본 적이 없는 새로운 데이터에 모델이 얼마나 잘 작동하는지 그리고 과소적합과 과대적합이 발생하지는 않았는지를 판단할 수 있게 한다. 그리고 이와 같은 정확도 추정을 위한 방법으로 가장 널리 사용되는 홀드 아웃(Hold-out) 검증 방법과 K-겹 교차검증(Kfold cross validation)이 있다(Kohavi, 1995).

(1) 홀드 아웃 검증

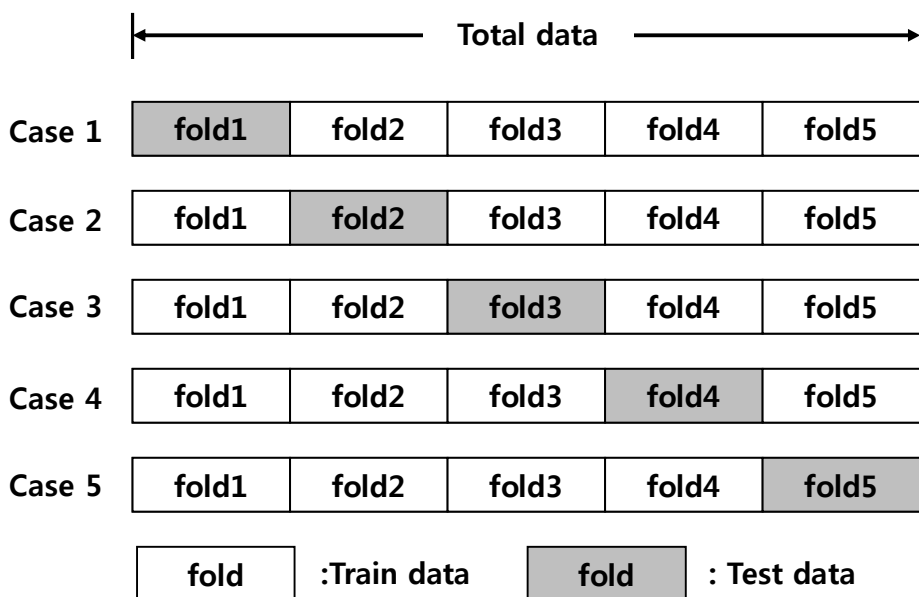
홀드 아웃 검증은 덤퍼닝으로 입력할 데이터셋을 학습과 테스트, 검증 구간으로 나누어 평가한다. 모델의 학습은 학습데이터를 이용하여 이뤄지고, 테스트 데이터셋은 학습된 모델의 성능을 추정하기 위해 사용한다. 검증을 위한 데이터는 모델의 평가를 위해 사용한다. 하지만 평가로 사용될 데이터가 너무 많으면, 전체 데이터에 대해 학습하지 못해 모델의 성능에 대해 비관적으로 예측하는 경향이 증가할 수 있다. 반대의 경우로 너무 적은 양으로 나누면 성능 추정값의 신뢰구간이 커지게 되는 트레이드 오프 관계가 있다(임창남, 2021).

(2) k-겹 교차검증

k-겹 교차검증은 <그림 2-10>과 같이 전체 데이터를 k개의 동등한 부분으로 분리하여 Case 별로 하나의 fold를 제외한 나머지 fold를 통해 학습을 진행한다. 그리고 제외한 나머지 하나의 fold를 통해 학습된 모델을 평가한다(이한승 외, 2019). 또한, Case 별로 앞서 설명한 방식으로 분할된 k개의 데이터셋을 모두 한 번씩 테스트데이터로 사용하기 위해 k번 반복하여 모델을 학습하고 평가한다(Yadav & Shukla, 2016). 이 방법은 모델의 학습과 평

5) Tensorflow.org, 1-5-테스트와-검증, March 16, 2021, accessed <https://www.tensorflow.org/해즈온-머신러닝-1장-2장/1-5-테스트와-검증/>

가에 모든 데이터를 사용할 수 있다는 장점 때문에 데이터의 크기가 작을 경우 사용된다.



<그림 2-10> K-겹 교차검증 예시

2.2.7 데이터 정규화

딥러닝의 학습방법으로 사용되는 경사 하강법(Gradient descent method)은 손실함수 값이 낮아지는 방향으로 변수에 학습률을 곱해 최소 손실함수 값을 갖도록 변수의 값을 찾는다. 이때 손실함수는 실제값과 예측값의 차를 수치화 해주는 함수이다(강민제, 2020). 하지만 데이터의 특성 간 편차가 크면 손실함수의 가중치가 값이 큰 데이터로 편향될 수 있고, 학습률이 불필요하게 크게 사용될 수 있다. 따라서 딥러닝으로 사용될 입력데이터는 데이터 특성 간의 편차를 줄이기 위해 정규화(Normalization)를 진행하여 입력된다. 본 연구에서 데이터의 정규화로 사용하는 방법으로는 입력데이터의 값을 0 ~ 1 사이의 값으로 변환하는 Minmaxscaler를 적용하며, 식(2.1)과 같다.

$$X = (x - x_{\min}) / (x_{\max} - x_{\min}) \dots\dots\dots(2.1)$$

여기서, x = 데이터

x_{\min} = 데이터 특성의 최소값

x_{\max} = 데이터 특성의 최대값

X = 스케일 변환데이터

2.2.8 모델 성능평가

시계열 데이터를 통해 값을 예측하는 순환신경망 모델의 예측성능을 평가하기 위해 평균 제곱근 오차(RMSE: Root Mean Square Error)를 사용한다. RMSE는 모델의 실제값과 예측값의 차이를 다룰 때 사용하는 척도로 식(2.2)과 같다. 그리고 RMSE는 크기 의존적 에러(Scale-dependent error)로 예측대상의 크기에 영향을 받기 때문에 스케일이 다른 예측대상의 RMSE가 비슷한 수치를 나타내는 경우 직관적 해석이 어렵다(Willmott & Willmott, 2005). 따라서 예측대상 수가 하나 이상일 경우 식(2.3)인 평균 절대 백분율 오차 (MAPE: Mean Absolute Percentage Error)를 추가로 사용한다. MAPE가 5일 경우를 예로, 예측값이 실제값에서 평균 5% 벗어남을 의미하기 때문에 직관적 해석이 용이하다.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \tilde{y}_t)^2} \dots\dots\dots(2.2)$$

$$MAPE(\%) = \frac{1}{n} \sum_{i=1}^n \left(\left| \frac{y_i - \tilde{y}_i}{y_i} \right| \right) * 100 \dots\dots\dots(2.3)$$

여기서, y_i = 실제값

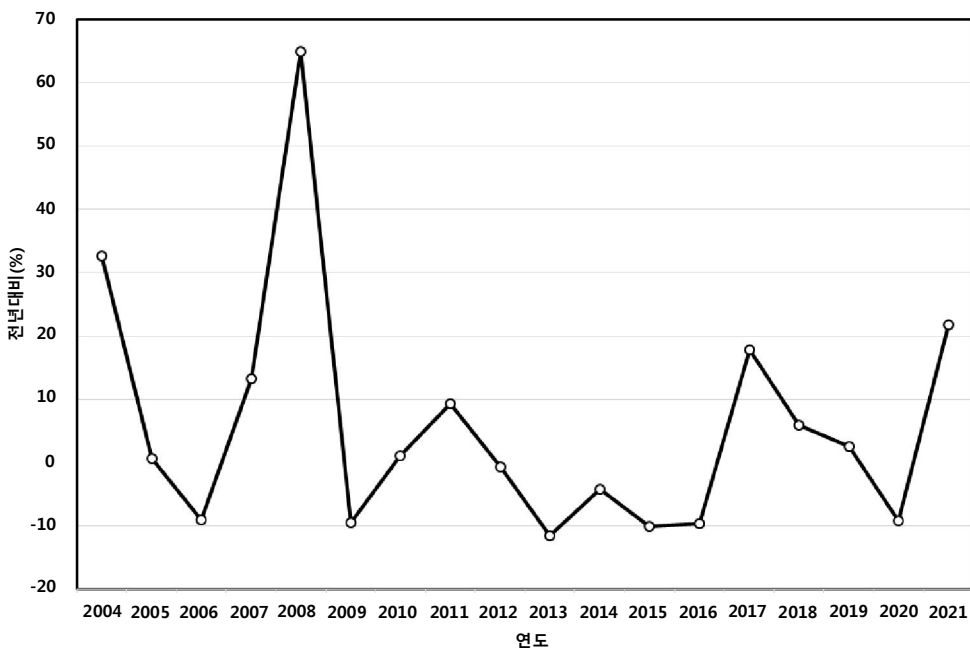
\tilde{y}_i = 예측값

n = 데이터의 개수

제3장 철근 가격예측모델 환경구성

3.1 데이터셋 구성

철근 가격은 <그림 3-1>과 같이 가격 변동성이 크고, 자연적, 물리적 특성이 아닌 사회적, 경제적 특성들의 영향으로 시세가 달라진다. 그리고 다른 재료보다 가격 인상요인이 있으면, 이를 즉시 시장에 반영할 수 있는 시장 지배력으로 가격변동이 자주 발생하는 건축자재이다.

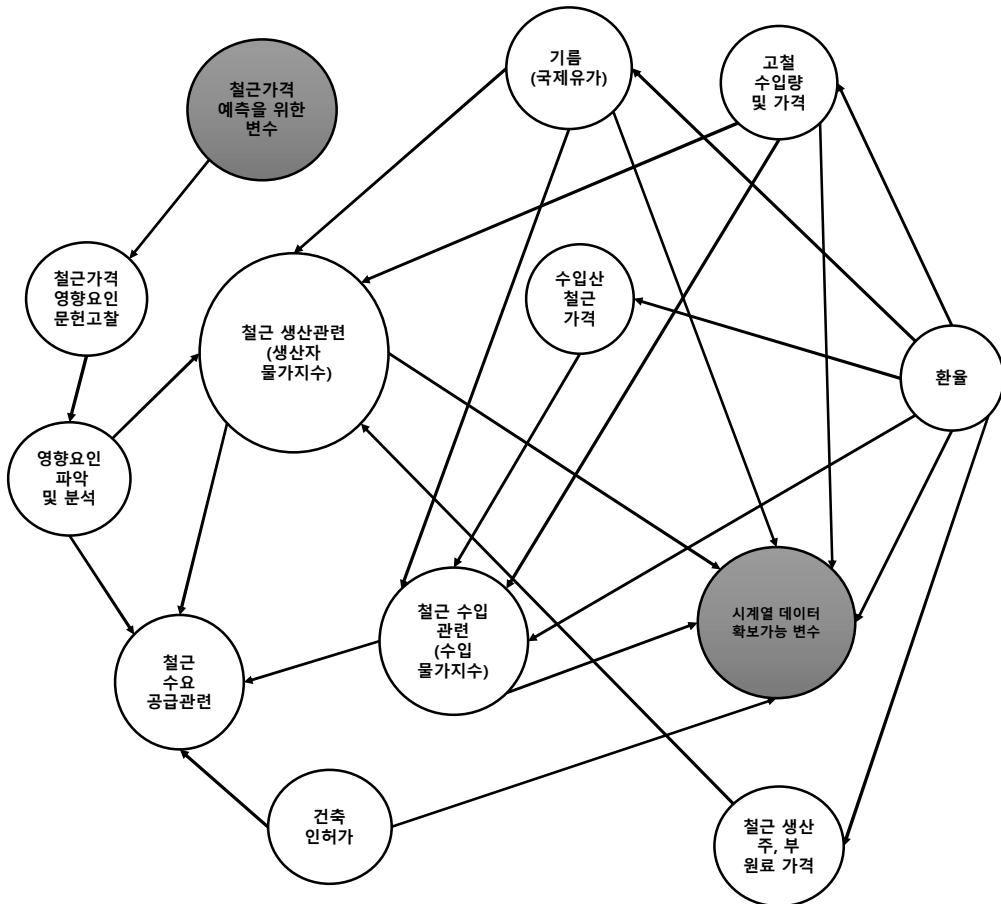


<그림 3-1> 일반 철근 가격(SD300), 전년 대비 연평균 변화율(%)

가격 인상의 요인으로서는 주원료인 철광석, 고철 스크랩의 가격 인상이나, 합금철 등과 같은 부원료의 가격상승과 철근 수요 증가에 따른 공급이 원활하지 않을 경우를 예로 들 수 있다(최민수 & 권오현, 2008). 또한, 철근 생산과 관련해서 전기로의 가동으로 사용되는 기름은 산유국이 아닌 국내에서 이를 전량 수입에 의존하기 때문에 수입 시 책정되는 기름 가격과 환율에도 민감하다(최열 외, 2009).

이와 같은 조사된 내용을 기초로, 철근 가격예측으로 사용할 데이터 특성

변수를 선정하기 위해 인과지도(Influence diagram)를 활용하였다. 인과지도는 여러 변수 간의 인과관계를 동시에 조망하도록 유도하는 상관관계에 있는 변수를 연결한 도식으로 <그림 3-2>와 같다(Howard, R., & Matheson, 2005).



<그림 3-2> 철근 가격과 관련한 예측변수 선정을 위한 인과지도

인과지도를 통해 변수들의 상관관계를 살펴보면, 생산 및 수입과 관련한 변수가 여러 변수의 영향을 받는 것을 알 수 있다. 환율의 경우는 수입과 관련한 변수에 영향을 미치는 변수임을 확인할 수 있다. 이에 따라 인과지도를 통해 수입물가지수, 생산자물가지수, 국제유가, 고철 수입량 및 가격, 수입 철근 가격, 철근 생산량, 철근 생산과 관련한 주, 부원료, 건축인허가와 같은 변수가 철근 가격에 영향을 미치는 인자로 특정하였다.

그리고 이를 토대로 철근 생산과 수요에 직·간접적 연관성이 있으며, 시계

월 데이터로 확보 가능한 변수를 철근 가격예측으로 사용할 데이터 특성으로 선정하였다. 특성으로 사용하는 항목을 ID로 표기하여 <표 3-1>과 같이 정리하였다.

<표 3-1> 데이터셋 구성

ID	데이터 명칭	데이터 시작 날짜	출처
A	수입물가지수(철근)	1971년 1월	통계청
B	생산자물가지수(철근)	1965년 1월	통계청
C	두바이유 (\$/Ton)	2002년 11월	e-나라지표
D	브렌트유 (\$/Ton)	2002년 11월	e-나라지표
E	서부텍사스유 (\$/Ton)	2002년 11월	e-나라지표
F	고철 스크랩 단가(\$/Ton)	2002년 11월	e-나라지표
G	고철 스크랩 수입량	2003년 1월	e-나라지표
H	철근 생산량	2001년 1월	통계청
I	건축 착공현황 (동수별 합계)	2000년 8월	국토교통부 통계누리
I	철근 가격 (일반, SD300)	2000년 1월	한국물가정보(Kpi)

딥러닝으로 학습할 데이터는 시점, 시간 간격이 모두 같아야 한다. 그러나 확보 가능한 철근 가격데이터는 월 단위로 제한적이다. 이에 따라 모든 데이터의 시간 간격을 철근 가격데이터와 같은 월 단위로 정리하였다. 또한, 모든 데이터의 시작 지점은 고철 스크랩 수입량의 시작 월인 2003년 1월을 기준으로 정리하였다.

환율 데이터는 인과지도를 통해 확인한 바와 같이 수입과 관련한 항목들

에 영향을 미치는 것으로 확인된다. 따라서 확보 가능한 데이터 특성 항목 중 \$/Ton으로 표시되는 데이터에 곱하여 원화 단위로 변환 적용하였다.

추가로 철근 가격데이터와 선정된 변수 간의 상관관계를 수치로 확인하기 위해 피어슨 상관관계 분석을 활용하였다. 피어슨 상관계수는 -1 ~ +1의 사이 값으로 출력되며, 식(3.1)과 같다. 그리고 결과에 대한 해석은 <표 3-2>와 같다.

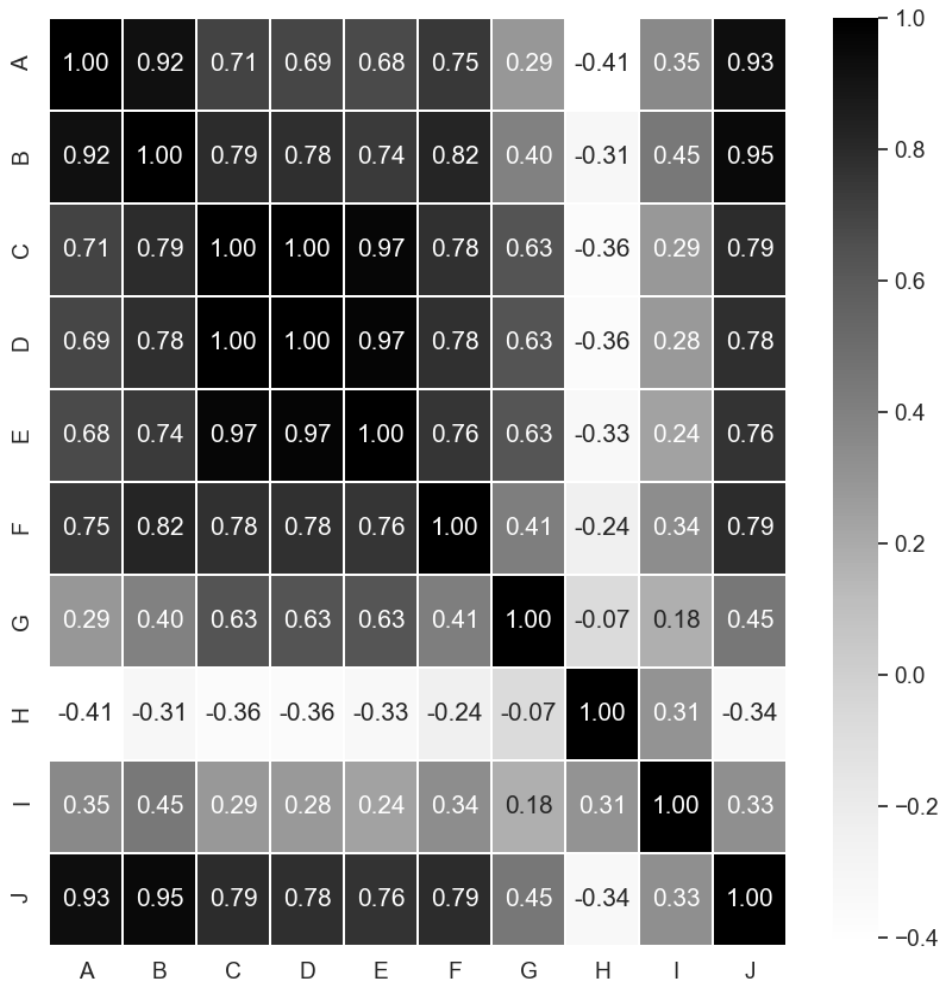
$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \dots\dots\dots(3.1)$$

여기서, \bar{x} = 표본집단 X의 평균
 \bar{y} = 표본집단 Y의 평균
 s_x = 표본집단 X의 표본 표준편차
 s_y = 표본집단 Y의 표본 표준편차
 n = 표본집단의 개체 수
 r = 피어슨 상관계수

<표 3-2> 피어슨 상관계수 해석표

상관계수 r 의 범위	해석
$-1.0 \leq r \leq -0.7$	강한 음의 선형관계
$-0.7 \leq r \leq -0.3$	뚜렷한 음의 선형관계
$-0.3 \leq r \leq -0.1$	약한 음의 선형관계
$-0.1 \leq r \leq +0.1$	무시 될 수 있는 선형관계
$+0.1 \leq r \leq +0.3$	약한 양의 선형관계
$+0.3 \leq r \leq +0.7$	뚜렷한 양의 선형관계
$+0.7 \leq r \leq +1.0$	강한 양의 선형관계

피어슨 상관계수는 값이 -1 또는 +1에 근접할수록 각각 음 또는 양의 상관관계가 높은 것으로 해석할 수 있다(지세현 외, 2019). 그리고 10개의 데이터 특성에 이를 적용하여 직관적 이해가 쉽도록 색상으로 표현할 수 있는 다양한 정보를 이미지 위에 열 분포 형태로 나타내는 히트맵으로 <그림 3-3>과 같이 시각화하였다.



<그림 3-3> 피어슨 상관관계 분석결과 시각화

<그림 3-3>을 살펴보면, 철근 가격 ‘J’와 선정된 데이터 특성은 모두 뚜렷한 강한 양 또는 음 이상의 선형관계를 나타내는 것으로 파악된다. 따라서 본 연구에서 선정한 데이터 특성은 철근 가격과 모두 상관관계의 유의성이 있는 것으로 판단된다.

3.2 순환신경망 모델 구성

본 연구에서는 순환신경망을 활용하여 철근 가격예측 시 과소적합, 과대적합이 발생하지 않은 적합한 모델을 탐색하기 위해 RNN, LSTM 그리고 BRNN과 LSTM을 결합한 Bi-LSTM(Bidirectional LSTM)을 각각 사용하는 딥러닝 모델을 구현하였다. 그 외에 시스템 구성과 사용된 패키지 정보는 <표 3-3>과 같다.

<표 3-3> 시스템 구성 사양 및 사용 패키지 정보

분류	프로그램 정보
Operation system	Window 10
Programming Tool	Visual Studio Code
General-Purpose computing on Graphics Processing Units	Cuda 10.0
Graphics Processing Unit	RTX 2080 Ti
Language	Python 3.6.5
Deep Learning Library	Tensorflow - GPU 1.15.0
Data Visualization	matplotlib 3.1.1
Array	Array numpy 1.16.5
RandomizedSearchCV Train_test_split	sklearn 0.23

딥러닝 관련 기법 중 드롭아웃은 cell의 다음에 위치시켜 출력(output)에만 적용하였다. 모델은 호출하여 사용할 수 있도록 함수로 정의하고 하이퍼 파라미터 탐색기인 Random search와 연결하였다. 딥러닝 모델 내부에서 사용되는 하이퍼 파라미터는 <표 3-4>와 같다.

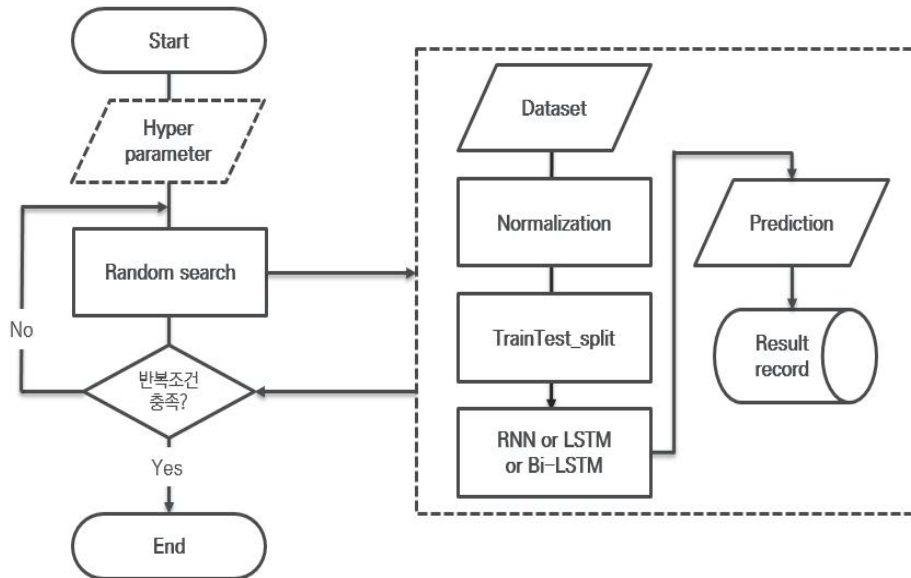
<표 3-4> 하이퍼 파라미터 및 설명

하이퍼 파라미터	설명
Sequence length	입력데이터의 행 길이
Learning rate	학습 속도
Hidden layer	은닉층 수
Iterations	학습량
Data_number	데이터조합에 따른 데이터셋 수
Keep_prob	드롭아웃의 적용비율
Split	학습과 테스트로 데이터 분할비율
Month	예측 기간 조절

Random search에서 학습률과 드롭아웃 비율은 소수점으로 설정된 입력범위 내에서 무작위로 선택되어 입력된다. 마찬가지로 Split을 제외하고, 나머지 하이퍼 파라미터는 정수 타입으로 입력범위 내에서 무작위로 조합되어 입력된다. Split은 분할비율을 소수로 리스트에 저장하여, 저장된 리스트의 순서 내에서 정수 타입의 숫자를 전달받으면 리스트 내에 저장된 소수를 선택하여 sklearn의 'Train_test_split'으로 전달한다.

사용자가 모델 알고리즘을 시작하면, 하이퍼 파라미터 입력값이 저장되어있는 함수에서 무작위로 입력값 조합을 선정한다. 그리고 이 입력값을 모델을 호출하여 전달한다. 모델 내부에서는 전달받은 입력값들을 통해 데이터셋을 불러오고 0 ~ 1 사이의 값으로 각각의 데이터 특성별 최소, 최대값과 sklearn 패키지의 'MinMaxScaler'에 의해 정규화가 진행된다. 이후 sklearn의 'Train_test_split'에 의해 학습과 테스트로 비율에 따라 분할된다. 그리고 이를 통해 학습이 진행되고, 모델에서 출력되는 결과는 텍스트 파일로 기록한다.

기본 모델 구성은 <그림 3-4>와 같다. 그리고 예측방법론에 따라 <그림 3-4>인 기본 모델에 딥러닝 관련 기법, 예측범위 확대방법을 추가 구현하여 적용한다



<그림 3-4> 딥러닝 모델 구성

제4장 순환신경망을 활용한 철근 가격 단기예측

4.1 철근 가격 데이터에 적합한 순환신경망 모델 선정

4.1.1 개요

순환신경망은 예측을 위해 사용하는 데이터양과 특성변수에 따라 과소적합과 과대적합이 발생할 수 있다. 이는 딥러닝의 예측성능을 떨어뜨리거나, 학습이 이뤄지지 않는 문제를 발생시킨다. 또한, 본 연구의 대상인 철근 가격은 한국물가정보에서 2000년 1월을 시작으로 현시점까지 월 단위의 시계열 데이터를 제공한다. 하지만 다른 빅데이터 기반의 기계학습 관련 연구와 비교하였을 때, 본 연구의 대상인 철근 가격은 예측을 위해 사용할 수 있는 데이터양이 절대적으로 부족하다. 데이터양은 학습 및 테스트로 사용할 수 있는 최대량이 대략 240개(20년 * 12개월) 정도이기 때문에 과소적합과 과대적합이 발생할 수 있다. 따라서 과소적합과 과대적합이 발생하지 않는 사용데이터에 적합한 딥러닝 모델탐색이 필요하다.

4.1.2 순환신경망의 단기 예측방법

순환신경망의 시계열 예측방법은 학습데이터와 이에 따른 정답데이터가 필요하다. 따라서 딥러닝 모델에서는 예측에 사용할 데이터셋을 불러들인 후 Cell에 입력할 데이터와 이에 대한 정답데이터를 재구성하는 과정이 진행된다. 이 과정은 <그림 4-1>과 같다.

<그림 4-1>은 딥러닝의 단기 시계열 예측방법으로 사용되는 데이터 재구성과정으로 정의된 함수의 입력값으로 사용되는 'data'는 전체 데이터를 의미하며, 'seq_length'는 전체 데이터에서 부분적으로 읽어올 데이터의 행 크기를 결정하는 하이퍼 파라미터, 'data_dimension'은 입력데이터의 특성 수를 의미한다. 정의된 함수 안에서는 모델의 Cell로 입력되는 데이터의 배열

```
def train_build_dataset(data, seq_length, data_dimension):
    x_shape = (len(data)-seq_length, seq_length, data_dimension)
    x_batch = np.zeros(shape=x_shape, dtype=np.float32)

    y_shape = (len(data)-seq_length, 1)
    y_batch = np.zeros(shape=y_shape, dtype=np.float32)

    for i in range(0, len(data) - seq_length):
        x_batch[i] = data[i:i + seq_length,:]
        y_batch[i] = data[i + seq_length, [-1]]

    return x_batch, y_batch
```

<그림 4-1> 데이터 재구성과정

형태를 나타내는 'x_shape, y_shape'로 'x_batch, y_batch'를 통해 0으로 채워진 배열을 생성하게 된다. 그리고 반복문의 i는 0을 시작으로 전체 데이터의 행수에서 'seq_length'를 뺀 수의 직전까지의 범위를 가지고 이를 0번째 행부터 'seq_length'의 크기만큼 읽어와 'x_batch'에 순서대로 빈 배열을 채운다. 'y_batch'도 x_batch의 다음 행의 마지막 값인 예측대상을 정답으로 순서대로 배열을 채운다. 그리고 이 방식을 통해 데이터를 재구성하면 가장 마지막 행을 포함하는 'x_batch'의 마지막 배열에는 이후의 정답에 해당하는 데이터가 없기 때문에 이 구간의 직전까지만 데이터를 재구성한다. 그리고 이 제외된 마지막 구간은 실제 예측을 위해 사용된다. 이러한 과정을 직관적으로 이해하기 쉽도록 시퀀스 길이 4를 사용할 때를 예로 <그림 4-2>와 같이 나타냈다.

a	b	c
1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18

a	b	c
1	2	3
4	5	6
7	8	9
10	11	12
		15
4	5	6
7	8	9
10	11	12
13	14	15
		18
7	8	9
10	11	12
13	14	15
16	17	18
		?

<그림 4-2> 데이터 재구성 예시

4.1.3 예측 및 검증 방법

앞서 선정한 10가지 데이터 특성으로 구성된 데이터셋에 적합한 딥러닝 모델을 선정하기 위해 RNN, LSTM, Bi-LSTM 3가지 모델을 통해 예측성능을 측정한다. 검증에 사용하는 하이퍼 파라미터의 입력범위는 <표 4-1>과 같다.

입력데이터의 전체기간은 2003년 1월부터 2020년 3월까지, 분할은 학습 80%, 나머지 20%는 테스트로 사용된다. 이에 따라 학습으로 사용되는 데이터의 기간은 2003년 1월부터 2018년 3월까지, 나머지 기간은 테스트로 사용한다. 학습과 테스트 성능은 RMSE에 의해 측정되고, 실제 측정한 2020년 4월의 예측값은 실제값과 비교하여 정확도로 나타낸다. 그리고 모델별 성능 비교를 통해 가장 우수한 성능을 나타내는 1개의 모델을 선택한다.

<표 4-1> 신경망 모델 선정에 사용되는 하이퍼 파라미터 입력범위

하이퍼 파라미터	범위
시퀀스 길이	6
학습률	0.01
은닉층 수	50
학습량	2000

4.1.4 예측결과 및 고찰

본 연구에서 선정한 데이터에 적합한 딥러닝 모델 선택을 위한 각각의 모델별 단기예측 결과는 <표 4-2>와 같다.

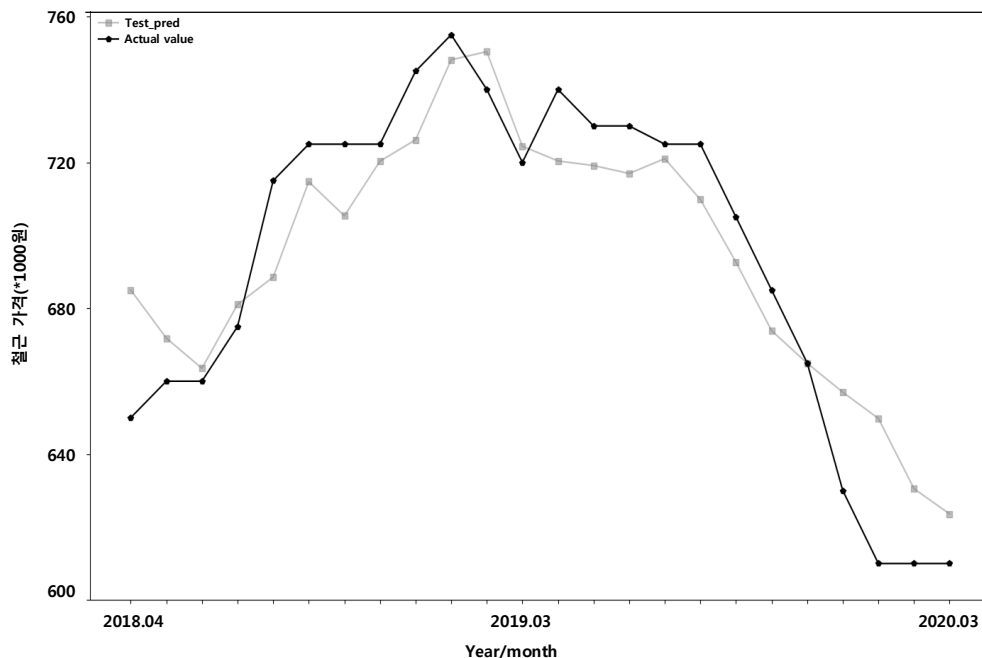
<표 4-2>는 모델별로 측정된 학습과 테스트 성능, 예측값을 정리한 것으로 Bi-LSTM이 학습과 테스트 RMSE 모두 우수한 결과를 나타냈다. 2003년 1월부터 2020년 3월까지의 입력데이터를 통해 측정된 2020년 4월의 예측값도 Bi-LSTM이 약 97.21%의 정확도로 RNN, LSTM의 측정된

<표 4-2> 순환신경망 모델탐색 결과

딥러닝 모델	학습 RMSE	테스트 RMSE	예측값 (원)	실제값 (원)	정확도(%)
RNN	0.03006	0.03578	657,000	610,000	92.30
LSTM	0.03216	0.03548	636,000		95.74
Bi-LSTM	0.02490	0.02883	627,000		97.21

정확도보다 약 4.91%, 1.47% 높은 정확도를 나타냈으며, 측정된 테스트 구간 예측값과 실제값을 시각화한 결과는 <그림 4-3>과 같다.

이를 살펴보면 예측된 결과가 최근의 시점에 가까워질수록 오차가 커지는 것으로 확인된다. 따라서 최근 값의 예측도 이로 인해 오차가 커지는 것으로 추정되며, 이를 개선하기 위한 딥러닝 관련 기법의 적용이 필요하다.



<그림 4-3> Bi-LSTM 테스트 구간 시각화

4.2 데이터 조합탐색 및 딥러닝 관련 기법을 활용한 단기에측

4.2.1 개요

본 절에서는 딥러닝을 활용한 철근 가격 단기에측의 예측 정확도 향상을 위해 데이터조합 및 관련 기법을 활용하고자 한다. 철근 가격은 사회적, 경제적 특성이 반영되기 때문에 딥러닝의 학습으로 사용할 데이터 특성 선정에 어려움이 있다. 그리고 딥러닝을 통해 예측으로 사용할 데이터 특성을 선정하더라도 일부 특성이 예측성능의 저해 요소로 작용할 수 있다. 또한, 사용자에게 의해 결정되는 하이퍼 파라미터는 입력하는 값에 따라 모델에서 측정되는 예측성능 달라진다. 따라서 예측성능을 최대로 끌어올리기 위한 데이터의 조합과 하이퍼 파라미터 최적값의 조합탐색이 필요하다.

4.2.2 Random search

학습을 수행하기 위해 사전에 설정해야 하는 입력값인 하이퍼 파라미터는 조합에 따라 모델의 예측성능이 크게 달라진다. 따라서 모델의 예측성능을 끌어올리기 위해서는 사용하는 하이퍼 파라미터의 조합탐색이 필요하다. 이를 위해 본 연구에서는 탐색 대상 구간 내의 후보 하이퍼 파라미터값을 무작위로 선택하여 입력값의 조합을 탐색하는 Random search를 딥러닝 모델과 연결하여 사용한다.

Random search는 Grid search와 방법적 큰 맥락은 유사하다. 하지만 Grid search보다 불필요한 반복 수행 횟수를 대폭 줄이고, 정해진 간격 사이의 위치한 값들에 대해서도 확률적 탐색이 가능하기 때문에 최적화가 더 빠르게 가능하다(Bergstra & Bengio, 2012).

4.2.3 예측 및 검증 방법

딥러닝을 활용한 단기예측의 정확도 향상을 위해 본 연구에서 선정한 2003년 1월부터 2020년 3월까지 1개월 시간 단위로 기록된 9개의 데이터를 조합하여 데이터셋을 생성한다. 그리고 Random search로 각각의 데이터조합에서 무작위로 선택된 데이터셋과 하이퍼 파라미터를 동시에 탐색한다. 데이터의 조합의 수는 <표 4-3>과 같다.

<표 4-3> 데이터 특성 조합별 데이터셋 수

조합	9C ₂	9C ₃	9C ₄	9C ₅	9C ₆	9C ₇	9C ₈	9C ₉	합계
데이터셋 수	36	84	124	126	84	36	9	1	500

그리고 이 각각의 과정에는 테스트 구간의 실제값과 예측값에 의한 RMSE, 입력데이터에 포함되지 않는 1달 후의 가격, 즉 2020년 4월을 예측하여 출력한다. Random search의 탐색 횟수는 각각의 조합별로 100번씩 총 800번 반복한다. 그리고 각각의 조합에서 탐색한 100개의 결과는 식(4.1)에 의해 우선 선별기록 된다.

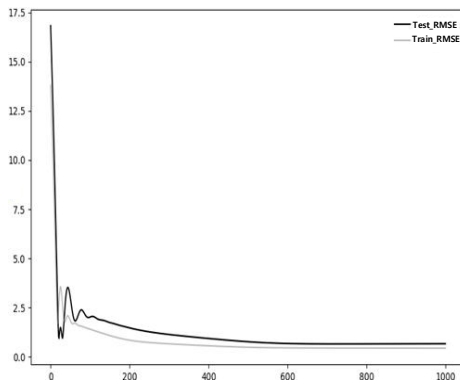
$$P = 0.999 \leq 1 - (TrainR - TestR) < 1 \dots \dots \dots (4.1)$$

여기서, P = Random search의 측정결과 선별 기록조건

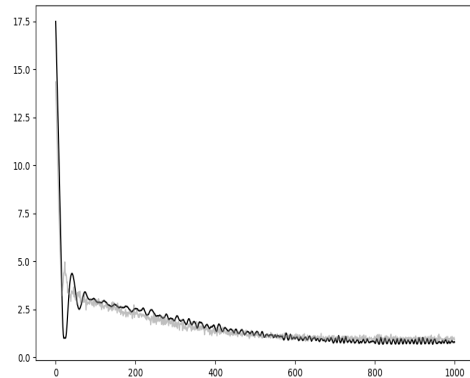
$TrainR$ = 학습성능

$TestR$ = 테스트성능

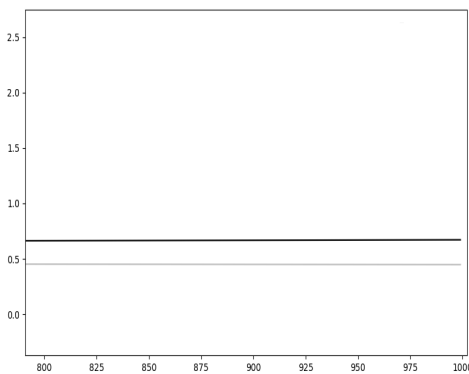
식(4.1)은 학습과 테스트 RMSE가 1과 0.999 이내의 근접한 결과를 선별하기 위해 사용한 식으로, 이에 대한 설명을 위해 <그림 4-4>를 추가하였다. <그림 4-4>는 드롭아웃의 적용 전후로 학습량 1부터 1000까지 학습량이 1씩 증가 시 학습과 테스트 RMSE를 측정하여 기록한 결과를 나타낸다. 드롭아웃을 사용하지 않으면 <그림 4-4>의 (A)와 이를 부분적으로 확대한 (C)와 같이 테스트 RMSE가 학습 RMSE 이상으로



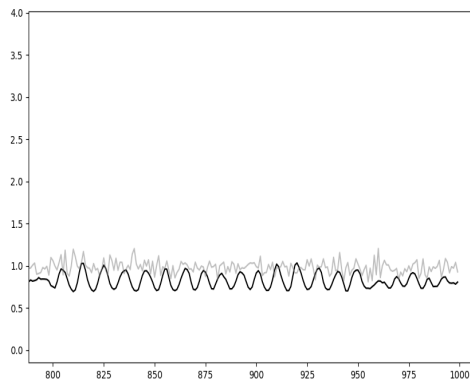
(A) 드롭아웃 비적용



(B) 드롭아웃 적용



(C) 그림확대
(드롭아웃 비적용, 학습량 800 ~ 1000)



(D) 그림확대
(드롭아웃 적용, 학습량 800 ~ 1000)

<그림 4-4> 드롭아웃 적용 전과 후의 차이

감소하지 못한다. 하지만 드롭아웃의 적용으로 <그림 4-4>의 (B)와 이를 확대한 (D)와 같이 학습량이 증가할수록 과동의 형태로 서서히 감소하며, 학습능력 이상으로 테스트 성능이 감소하는 것을 확인할 수 있다. 따라서 드롭아웃의 적용으로 테스트 성능은 학습능력보다 감소할 수 있다. 하지만 학습능력과 테스트 성능 차이가 크면 과대적합에 의한 모델이 일반화되지 못한 상태이기 때문에 입력데이터의 범위를 벗어난 값을 예측할 경우 전혀 다른 값을 예측할 가능성이 있다. 따라서 본 연구에서 사용하는 Random search에는 드롭아웃의 입력값을 동시에 탐색하기 때문에 탐색결과 내에서 학습과 테스트 성능 차이가 작은 결과를 식별할 수 있도록 식(4.1)을 추가하였다. 또한, 본 장의 연구에서 사용하는 신뢰구간의 정확도를

추정하는 방식은 홀드 아웃 검증방식을 사용하기 때문에 성능 추정에 민감하다는 단점이 있어 모델에서 <표 4-4>와 같이 실제값과 출력된 예측값을 확인하는 경험적인 추론을 통해 가정하였다.

<표 4-4> 선별식 추정을 위한 예측결과

학습 RMSE	테스트 RMSE	P	예측값 (원)	실제값 (원)	정확도 (%)
0.03504	0.03498	0.99994	621,000	610,000	98.19
0.03458	0.03417	0.99959	618,100		98.67
0.02555	0.02465	0.99910	603,200		98.87
0.03234	0.02573	0.99339	632,800		96.26
0.03234	0.02592	0.99358	631,800		96.42

<표 4-4>를 살펴보면 P 값이 0.999 이상에 해당하는 경우에서 예측 정확도가 98% 이상을 나타낼 수 있다. 또한, 이에 해당하는 결과에서 학습과 테스트 RMSE가 낮을수록 정확도가 증가하는 것을 확인할 수 있다. 따라서 식(4.1)을 통해 선별 및 기록된 결과 내에서는 테스트 RMSE가 낮은 순으로 단계적으로 선별하여 최종선택하며, 최종선택된 결과에 기록된 입력 데이터의 범위를 벗어난 이후의 1달 가격을 실제값과 비교하여 모델의 예측 성능을 평가한다.

사용데이터는 철근 가격이 2020년 1월부터 3월까지 610,000원으로 가격변화가 없는 상태를 나타낸다. 따라서 단기에측 시 가격의 등락을 올바로 예측하는지에 대한 확인이 어렵다. 그래서 이에 대한 추가적인 확인을 위해 기존의 데이터셋에서 데이터의 마지막 기간을 삭제하는 방식으로 데이터셋을 추가생성 한다. 추가 생성하는 데이터셋은 가격변화의 상황을 가격이 상승, 하락, 유지의 세 가지 변화에 대해 특징하여 생성하였다. 이에 따라 생성된 데이터의 기간은 마지막 철근 가격이 오르는 추세를 나타내는 2003년 1월부터 2019년 11월까지, 하락하는 추세를 나타내는 2003년 1월부터 2017년 7월까지로 저장하였다.

그리고 딥러닝에 이를 각각 입력데이터로 사용하여 철근 가격의 등락을 올바르게 예측하는지 확인한다.

검증과정에서 사용할 하이퍼 파라미터의 탐색 범위는 <표 4-5>와 같다. 하이퍼 파라미터는 최소, 최대값의 범위 내에서 Random search에 의해 무작위로 선택된다.

<표 4-5> 하이퍼 파라미터 입력범위

하이퍼 파라미터	범위
Random search 탐색 수	100
시퀀스 길이	2 ~ 40
학습률	0.001 ~ 0.1
은닉층 수	10 ~ 500
학습량	100 ~ 7000
데이터 조합번호	500
드롭아웃 비율	0.1 ~ 1.0

4.2.5 예측결과 및 고찰

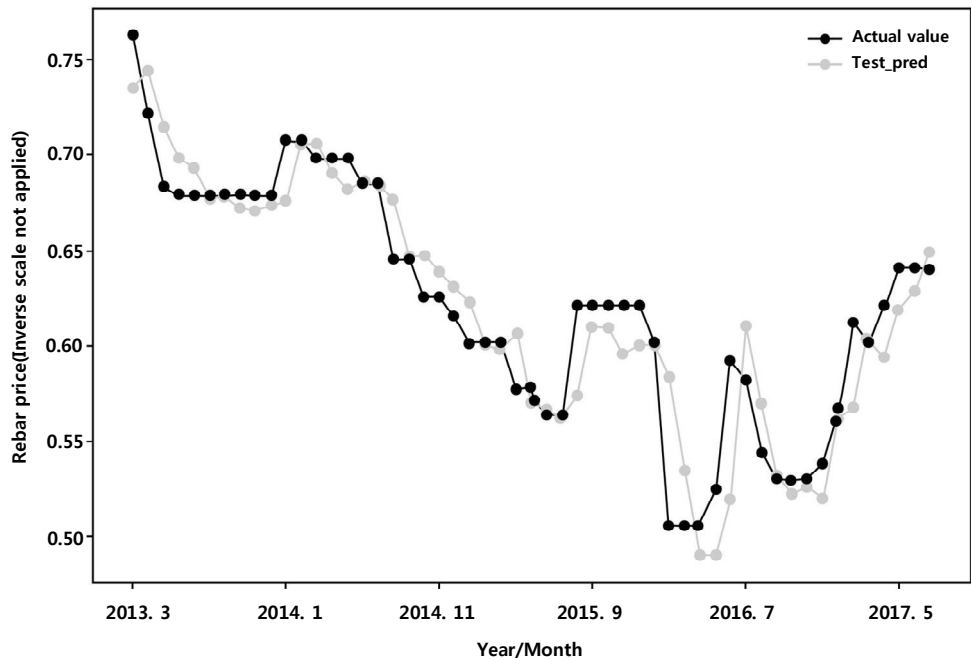
10개의 데이터를 조합하여 Random search로 탐색한 각각의 데이터 조합 결과는 <표 4-6>과 같다. 예측 기간별로 우수한 결과를 나타낸 데이터조합의 테스트 구간은 <그림 4-5>, <그림 4-6>, <그림 4-7>과 같다.

<표 4-6>을 살펴보면 데이터의 마지막 철근 가격이 오르는 추세인 2003년 1월부터 2017년 6월까지의 데이터셋을 사용한 결과에서 가장 우수한 예측성능을 나타낸 ${}_9C_4$ 의 데이터 BFGI는 테스트 RMSE는 0.02375를 나타냈으며, 이를 통해 측정된 2017년 7월의 예측값은 668,400원으로 실제값인 675,000과 약 6,600원 차이를 나타냈고, 실제값에 대한 정확도는 약 99.02%를 나타냈다. 데이터의 마지막 철근 가격이 하락하는 추세인 2003년 1월부터 2019년 10월까지의 데이터셋을 사용한 결과에서 가장 우수한 예측성능을 나타낸 ${}_9C_9$ 의 데이터 ABCDEFGHI는 테스트 RMSE 0.02527를 나타냈으며, 이를 통해 측정된 2019년 11월의 예측값은 649,000원으로 실제값인 630,000과 약 19,000원 차이를 나타냈으며, 예측정확도는 약 97.0

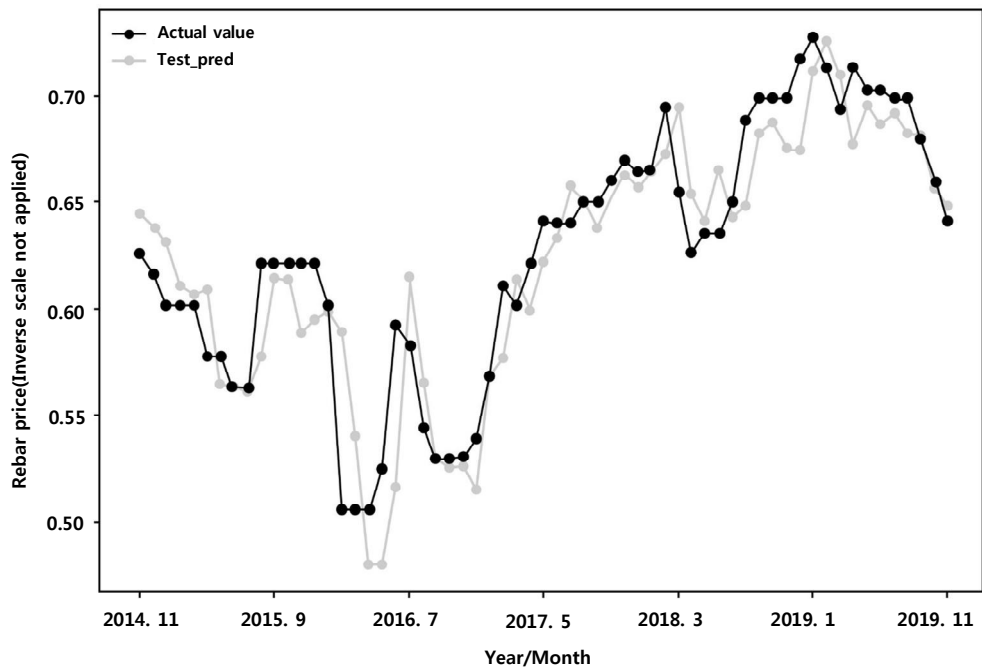
7%를 나타냈다. 2003년 1월부터 2020년 3월까지의 데이터셋을 사용한 결과는 가장 우수한 예측성능을 나타낸 ${}_9C_5$ 의 ACEGH가 테스트 RMSE 0.2466을 나타냈으며, 측정된 2020년 4월의 예측값은 603,200원으로 실제값과 6,800원의 차이를 나타냈다. 예측 정확도는 약 98.89%를 나타냈다.

<표 4-6> 데이터 조합탐색 결과

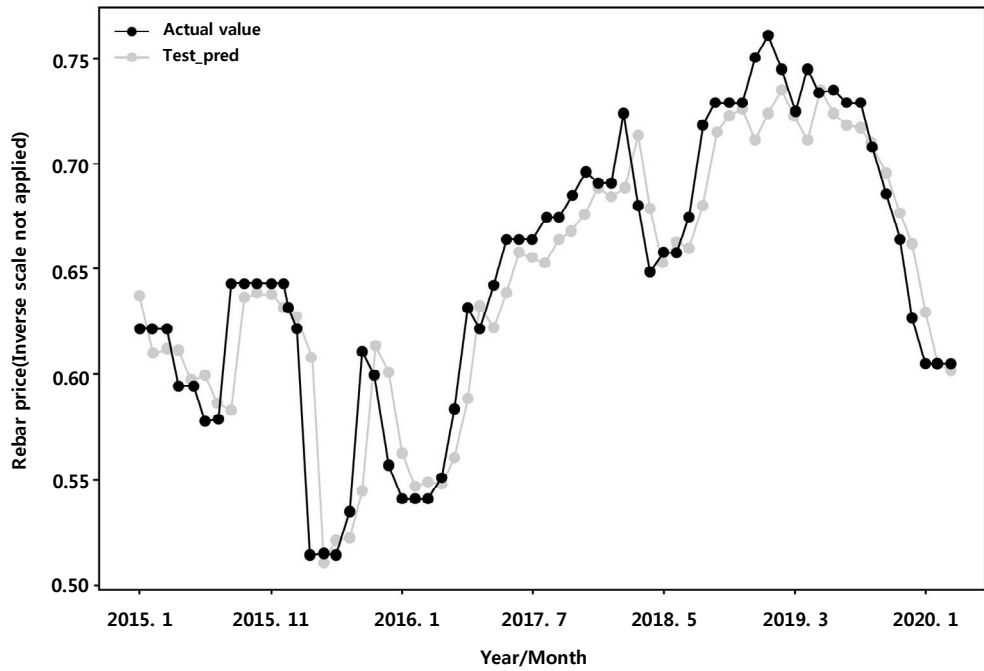
예측 월	데이터 조합	데이터셋	학습 RMSE	테스트 RMSE	RMSE 표준화 (원)	<i>P</i>	테스트 RMSE Rank	예측값 (원)
2017.8	${}_9C_2$	EG	0.02853	0.02803	29,200	0.99950	8	645,400
	${}_9C_3$	CDG	0.02775	0.02700	28,100	0.99925	6	650,700
	${}_9C_4$	BFGI	0.02419	0.02375	24,700	0.99956	1	668,400
	${}_9C_5$	AEFGI	0.02698	0.02675	27,900	0.99978	5	645,000
	${}_9C_6$	BCEFGI	0.02471	0.02471	25,700	0.99999	2	672,500
	${}_9C_7$	ACEFGHI	0.02669	0.02642	27,500	0.99972	4	684,700
	${}_9C_8$	ABCDEFGH	0.02629	0.02618	27,300	0.99988	3	665,800
	${}_9C_9$	ABCDEFGHI	0.02775	0.02771	28,800	0.99996	7	680,300
2019.12	${}_9C_2$	BI	0.03134	0.03111	23,600	0.99979	7	661,900
	${}_9C_3$	DGI	0.02917	0.02845	25,000	0.99927	6	657,500
	${}_9C_4$	EFGI	0.02688	0.02630	27,400	0.99942	3	654,700
	${}_9C_5$	BDEGH	0.02753	0.02715	28,300	0.99961	4	650,600
	${}_9C_6$	ACDFGI	0.03299	0.03278	26,300	0.99980	8	655,200
	${}_9C_7$	ACDEFGI	0.02591	0.02571	26,800	0.99980	2	644,300
	${}_9C_8$	ABCDEGHI	0.02881	0.02763	25,800	0.99903	5	655,500
	${}_9C_9$	ABCDEFGHI	0.02543	0.02527	26,300	0.99984	1	649,000
2020.4	${}_9C_2$	BG	0.02822	0.02735	25,500	0.99912	6	624,500
	${}_9C_3$	FGH	0.02711	0.02696	28,500	0.99984	5	616,200
	${}_9C_4$	CDEI	0.02708	0.02667	21,900	0.99958	4	601,600
	${}_9C_5$	ACEGH	0.02554	0.02466	23,000	0.99911	1	603,200
	${}_9C_6$	ADEFGH	0.02530	0.02468	25,700	0.99937	2	627,500
	${}_9C_7$	BCDFGHI	0.02774	0.02767	25,900	0.99994	7	628,000
	${}_9C_8$	ABDEFGHI	0.02658	0.02596	27,000	0.99938	3	628,700
	${}_9C_9$	ABCDEFGHI	0.03193	0.03103	29,000	0.99910	8	619,900



<그림 4-5> BFGI 조합 테스트 시각화



<그림 4-6> ABCDEFGHI 조합 테스트 시각화



<그림 4-7> ACEGH 조합 테스트 시각화

4.3 소결

본 장에서는 딥러닝을 활용하여 철근 가격을 단기 예측하기 위해 철근 가격예측에 사용하는 데이터셋에 적합한 딥러닝 모델 선정과 단기에측의 성능 향상을 위한 데이터조합 및 관련 기법을 적용한 연구를 진행하였다. 결과를 요약하면 다음과 같다.

1) 딥러닝은 예측을 위해 사용하는 입력데이터 특성에 따라 과소적합과 과대적합이 발생할 수 있으며, 이로 인한 문제는 학습이 제대로 이뤄지지 않거나 예측성능의 저하가 발생한다. 따라서 철근 가격 단기에측에 적합한 딥러닝 모델을 선정하기 위해 RNN, LSTM, Bi-LSTM으로 구성된 3가지 딥러닝 모델에 2003년 1월부터 2020년 3월까지의 철근 가격데이터를 포함하는 10개의 데이터 특성으로 구성된 데이터셋을 적용하여 예측성능과 예측값을 측정하였다. 측정결과 Bi-LSTM이 나머지 두 모델보다 학습과 테스트 성능이 모두 우수하였으며, 2020년 4월의 예측값은 실제값의 약 97.21%의 정확도로 예측하였다.

2) 딥러닝의 단기에측 정확도 향상을 위해 예측대상인 철근 가격데이터를 제외하고 9개의 데이터 특성을 조합하여 하이퍼 파라미터 최적화기인 Random search를 통해 동시에 탐색을 진행하였다. 그리고 사용 데이터는 철근 가격이 2020년 1월 ~ 4월까지 610,000원으로 가격변화가 없는 상태를 나타내기 때문에 데이터의 기간을 조정하여 마지막 철근 가격이 오르는 추세인 2003년 1월부터 2019년 10월까지, 하락하는 추세인 2003년 1월부터 2017년 6월까지를 데이터셋으로 추가 생성하여 본 연구에서 구현한 알고리즘이 철근 가격의 등락을 올바르게 예측하는지 확인하였다. 예측결과는 3가지 데이터셋의 대한 평균 예측 정확도 약 98.32%를 나타냈다.

이상과 같이 본 연구에서 선정한 데이터셋에는 Bi-LSTM을 사용하는 딥러닝 모델이 적합하고, 한가지 데이터셋의 하이퍼 파라미터를

집중탐색하는 방식보다 데이터조합을 사용하여 하이퍼 파라미터를 동시에 탐색하는 방식이 예측성능 개선에 효과적인 것을 알 수 있었다. 그리고 본 연구에서 탐색된 결과의 선별을 위해 사용한 식(4.1)을 통해 학습과 테스트 성능 차가 작은 과대적합이 발생하지 않은 결과의 선별이 가능하였다.

제5장 순환신경망을 활용한 철근 가격 장기예측

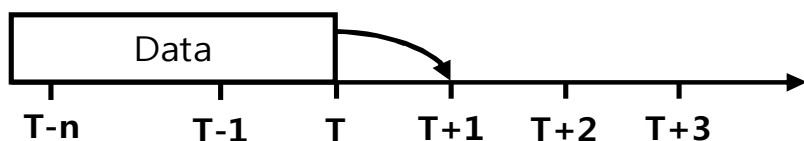
5.1 정답데이터의 시점 조정을 통한 예측범위 확대

5.1.1 개요

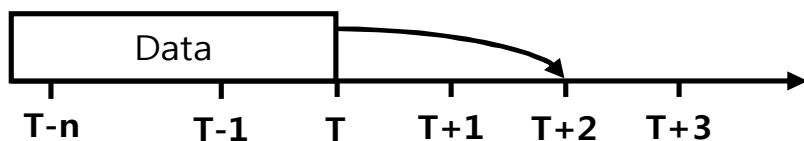
철근의 구매단계에서는 사용계획에 따른 구매 시기를 자재투입 시점 이전에 결정한다. 하지만 구매 적기를 판단하는 기존의 방식은 결정권자의 판단에 의존적이기 때문에 이윤과 손실에 대한 불확실성을 내포한다. 그리고 이 불확실성으로 인해 발생할 수 있는 문제를 건설사업관리 측면에서 가정해보면 자재의 투입 시기가 결정된 이후, 시세의 변동과 자재구매 시기에 대한 판단에 따라 계획된 비용보다 실투입비가 많아질 수 있다. 이에 따른 결과로는 책정된 예산에 상당한 부담으로 작용할 수 있으며, 공사의 규모와 면적이 커질수록 가중된다. 따라서 비용을 산정하는 기존의 방식은 주문량, 주문 시기에 책정된 단가를 대상으로 관련 비용함수를 최적화하여 기업이윤을 남기고자 하는 의사결정의 한 범주로 이해할 수 있으며, 기존의 방식보다 정확한 주문 시기의 판단이 가능하다면, 이러한 불확실성을 가지는 일련의 예측과정들의 체계성을 보완할 수 있을 것으로 사료된다. 이에 구매 시기와 책정된 단가를 파악하기 위해서는 앞서 제시된 순환신경망을 활용한 철근 가격을 단기예측의 예측범위를 확대할 방법이 필요하다.

5.1.2 순환신경망의 예측범위 확대를 위한 정답데이터의 시점 조정

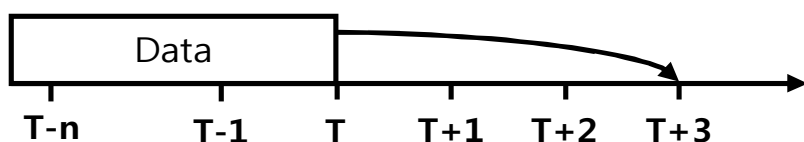
순환신경망 단기예측은 모델 내부에서 데이터셋을 불러들인 후, Cell에 시퀀스 길이 만큼의 입력데이터와 입력데이터 마지막 값의 다음 값을 정답으로 재구성하는 과정이 진행된다. 그리고 예측범위를 확대하기 위해서는 정답데이터의 시점을 <그림 5-1>과 같이 변경해야 한다. 이를 위해 앞서 설명한 <그림 4-1>의 데이터 재구성과정에 정답데이터의 시점 변경을 위한 하이퍼 파라미터 'month'를 추가하여 <그림 5-2>와 같이 변경하였다.



(a) 예측시점 1달



(b) 예측시점 2달



(c) 예측시점 3달

<그림 5-1> 정답데이터의 시점 변경 (최준영 & 이송근, 2020)

```
def train_build_dataset(data, seq_length, data_dimension, month):
    x_shape = (len(data)-seq_length + 1 - month, seq_length, data_dimension)
    x_batch = np.zeros(shape=x_shape, dtype=np.float32)

    y_shape = (len(data)-seq_length + 1 - month, 1)
    y_batch = np.zeros(shape=y_shape, dtype=np.float32)

    for i in range(0, len(data) - seq_length + 1 - month):
        x_batch[i] = data[i:i + seq_length,:]
        y_batch[i] = data[i + seq_length -1 + month, [-1]]

    return x_batch, y_batch
```

<그림 5-2> 데이터 배치 재구성과정 변경(month 추가)

<그림 5-2>는 앞서 설명한 <그림 4-1> 단기예측방식의 데이터 재구성과정에 ‘month’를 추가하여 ‘x_batch’의 마지막 행부터 ‘month’의 입력값만큼 떨어진 행의 마지막 값인 예측대상을 정답으로 변경하도록 설정한 것이다. 예를 들어 현시점을 T라할 때 ‘month’의 입력값으로 1이 사용되면, 단기예측의 데이터 재구성과정과 같은 T+1을 예측하기 위한

데이터 재구성 과정이 되고, 2가 사용되면 T+2를 예측하기 위한 데이터를 재구성한다. 이 과정을 직관적으로 이해하기 쉽도록 시퀀스 길이 4를 사용할 때를 예로 <그림 5-3>과 같이 나타냈다.

Data			시점
1	2	3	T-4
4	5	6	T-3
7	8	9	T-2
10	11	12	T-1
13	14	15	T
	18		T+1
	21		T+2
	24		T+3
	27		T+4

T+1(Pred month=1)			시점
1	2	3	T-4
4	5	6	T-3
7	8	9	T-2
10	11	12	T-1
13	14	15	T
	18		T+1

T+2(Pred month=2)			시점
1	2	3	T-4
4	5	6	T-3
7	8	9	T-2
10	11	12	T-1
13	14	15	T
	21		T+2

T+3(Pred month=3)			시점
1	2	3	T-4
4	5	6	T-3
7	8	9	T-2
10	11	12	T-1
13	14	15	T
	24		T+3

T+4(Pred month=4)			시점
1	2	3	T-4
4	5	6	T-3
7	8	9	T-2
10	11	12	T-1
13	14	15	T
	27		T+4

<그림 5-3> 정답데이터의 시점 변경 데이터 재구성 예시

5.1.3 예측 및 검증 방법

순환신경망의 예측범위를 확대하기 위해 본 연구에서는 월별 가격정보를 포함하는 2003년 1월부터 2020년 3월까지 1개월 간격으로 구성된 10개의 데이터 특성을 데이터셋으로 사용하여 앞서 설명한 정답데이터의 시점 변경에 사용되는 ‘month’의 예측범위를 1씩 늘리면서 예측성능을 측정한다. 각각의 과정은 Random search에 의해 하이퍼 파라미터 최적화가 진행된다. 측정된 결과는 월별로 실제값과 비교하여 정확도로 나타낸다. 순환신경망 모델은 Bi-LSTM을 사용하고, Random search의 하이퍼 파라미터 탐색 범위는 Manual search를 통해 한정하여 <표 5-1>로 정리하였다. 그리고 앞장의 단기예측에서 사용된 학습량의 경우 하이퍼 파라미터 탐색 범위 내에서 특정 값이 선택되면 선택된 값만큼의 학습이 진행되고 종료 시 결과를 출력하였으나, 이번 절의 예측모델에는 학습량의 최대값까지 1씩 증가할 때마다 측정된 결과를 기록하게 변경하였다. 이는 앞장의 <그림 2-8>에 설명되어있는 과소, 과대적합이 발생하지 않은 최적점을 더 면밀하게 탐색하기 위함이다. 출력결과는 식(4.1)에 의해 선별되며, 선별된 결과에서는 테스트 RMSE가 가장 낮은 결과를 선택한다.

<표 5-1> 하이퍼 파라미터 입력범위

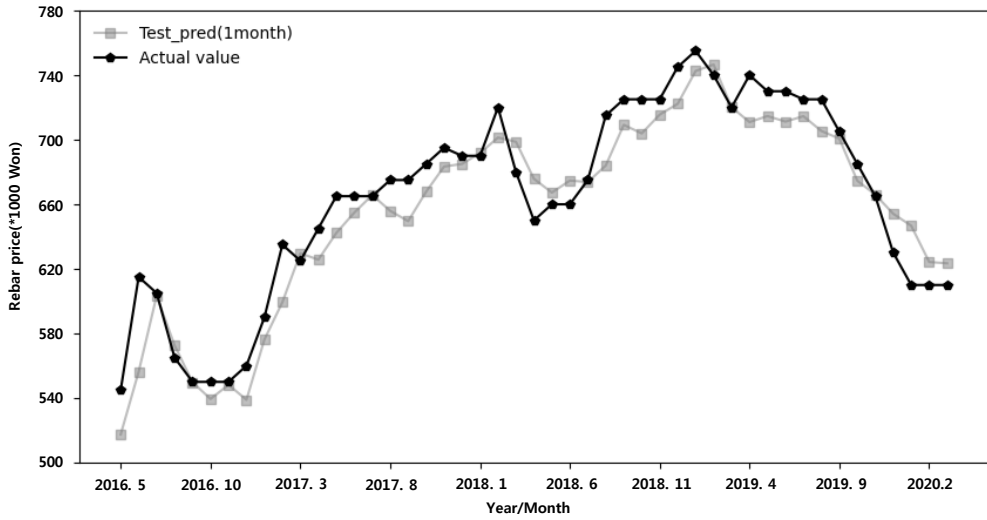
하이퍼 파라미터	설명
Random search의 탐색 수	50
시퀀스 길이	2 ~ 20
학습률	0.001 ~ 0.1
은닉층 수	10 ~ 100
학습량	5000
드롭아웃 비율	0.1 ~ 1.0

5.1.4 예측결과 및 고찰

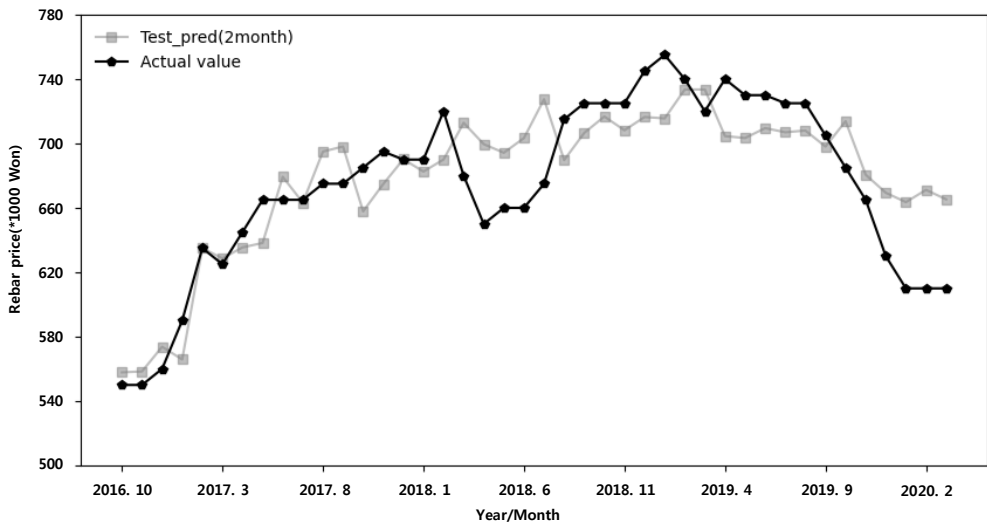
순환신경망을 활용한 철근 가격예측의 예측범위 확대를 위해 정답데이터의 시점을 1씩 증가시키며 Random search로 탐색한 결과는 <표 5-2>와 같다. 그리고 각각의 예측범위별 테스트 구간을 시각화한 결과는 <그림 5-4>, <그림 5-5>, <그림 5-6>과 같다.

<표 5-2> 정답데이터 시점 변경 결과

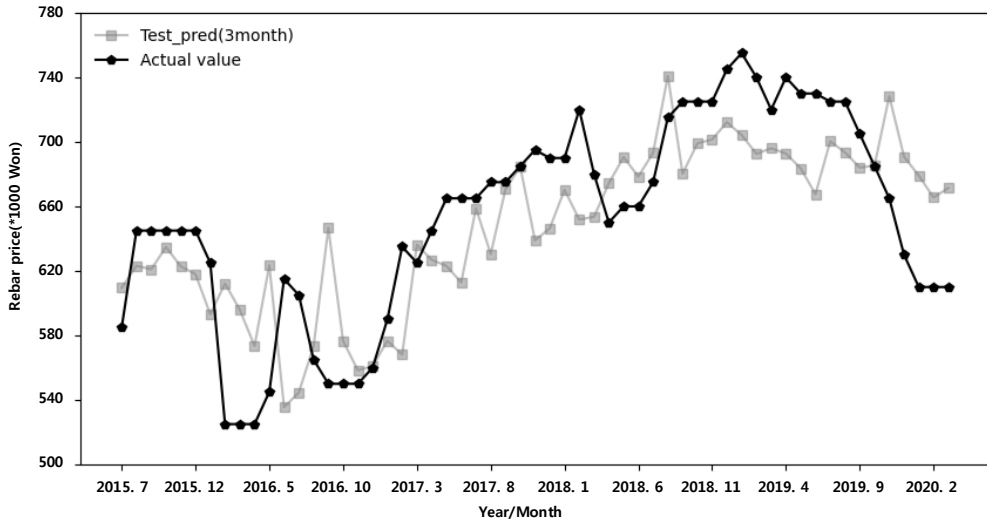
예측범위	학습 RMSE	테스트 RMSE	P	예측값 (원)	실제값 (원)	정확도 (%)
1month	0.03008	0.02908	0.999	628,000	610,000	97.05
2month	0.03644	0.03621	0.999	638,000	660,000	96.55
3month	0.06534	0.06533	0.999	649,000	670,000	95.37



<그림 5-4> 예측범위 1-month 테스트 구간



<그림 5-5> 예측범위 2-month 테스트 구간



<그림 5-6> 예측범위 3-month 테스트 구간

<표 5-2>를 살펴보면 예측범위를 확대하기 위해 정답데이터의 시점을 1씩 늘려 예측범위가 확대되면 학습과 테스트 성능, 예측 정확도가 커지는 범위에 따라 감소함을 알 수 있다. 그리고 각각의 예측범위별 시각화 결과를 살펴보면 예측범위가 늘어남에 따라 테스트 구간에서 측정된 각각의 예측값들이 실제값과의 차가 커지는 것 또한 확인할 수 있다. 특히 <그림 5-6>인 예측범위 3개월부터는 테스트 구간의 결과가 다른 예측범위의 결과보다 더 많은 차이를 나타낸다. 따라서 순환신경망을 통해 철근의 미래 가격을 장기예측하기 위해서는 다른 방식의 적용이 필요할 것으로 판단된다.

5.2 데이터 시간 단위 조정을 통한 철근 가격예측 범위 확대

5.2.1 개요

철근의 구매 시기와 책정된 단가를 파악하기 위해서는 순환신경망의 장기적 시점 예측이 필요하다. 하지만 모델의 단기예측에서 예측범위를 확대하기 위해 입력과 정답데이터 시점을 변경하면, 예측 시점이 확대됨에 따라 성능이 저하되는 문제를 나타낸다. 따라서 순환신경망의 예측범위를 확대할 수 있는 다른 방식의 적용이 필요하다.

5.2.2 데이터 시간 간격 재구성

단기 시계열 예측모델은 입력데이터의 시간 간격에 따라 예측범위가 결정된다. 예로 1달 간격의 데이터 사용은 1달 이후를 예측하는 것과 같다. 따라서 월, 분기, 반기, 또는 연 단위로 확보가 가능한 시계열 데이터를 통해 원하는 예측 기간을 선택하여 예측할 수 있다. 하지만 한국 물가정보에서 제공하는 철근 가격정보는 월 단위로만 제공된다. 그래서 예측 기간에 대한 선택이 제한적이다. 따라서 예측범위를 확대하기 위해서는 데이터의 시간 간격 조정이 필요하다. 이를 위해 본 연구에서는 데이터의 시간 간격을 재구성하기 위한 <그림 5-7>과 같은 방식을 코드로 구현하였다.

```
def rebuild_dataset(dataset(dataset, month):  
    dataset = dataset[::1]  
    dataset = dataset.ix[::month, :]  
    dataset = dataset[::1]  
  
    return dataset
```

<그림 5-7> 데이터 시간 재구성 방법

<그림 5-7>의 정의된 함수를 살펴보면 입력값인 데이터셋은 재구성이 필요한 원본 데이터셋이고, month는 재구성할 데이터의 시간 간격을

결정한다. 그리고 파이썬 패키지인 pandas의 'ix'는 레이블의 위치 정수를 사용하여 데이터를 인덱싱하는데, 이 위치 정수 입력값이 하이퍼 파라미터인 month이다. 이를 적용한 예로 3 month를 입력값으로 사용하여 데이터의 시간 간격을 재구성하면 <그림 5-8>의 (B)와 같은 결과를 얻을 수 있다.

	a	b		a	b
0	50.24	51.11	0	50.24	51.11
1	50.71	51.11	3	54.41	56.28
2	52.55	51.11	6	53.25	56.25
3	54.41	56.28	9	53.62	56.25
4	53.73	56.28	12	58.71	62.14
...
202	102.88	109.47	192	100.45	115.77
203	102.85	108.61	195	102.40	112.80
204	101.57	107.97	198	103.32	112.30
205	104.16	108.49	201	104.35	110.56
206	106.44	108.56	204	101.57	107.97

(A) Before

(B) After

<그림 5-8> 데이터 시간 간격 재구성 적용 전과 후

<그림 5-8> (A)와 (B)를 살펴보면 데이터 시간 간격 재구성 적용 전후로 3의 배수에 해당하는 값들만 존재하는 것을 알 수 있다. 따라서 단순히 ix로 데이터를 재구성할 경우 최신의 마지막 데이터가 반영되지 않기 때문에 <그림 5-7>의 정의된 함수 내에 'dataset[::1]'을 통해 전체 데이터의 순서를 뒤집어 맨 앞의 0번 데이터가 206번이 되도록 하여 제일 오래된 앞 데이터가 반영되지 않도록 설정하였다. 그리고 ix를 적용하여 데이터의 시간 간격을 조정 후 'dataset[::1]'를 다시 적용하고, 시간 순서를 원상태로 되돌린다. 이 방식의 적용을 통해 1개월의 시간 간격을 나타내는 데이터는 <그림 5-9>와 같이 3개월의 시간 간격을 나타내는 데이터로 재구성된다. 그리고 이 데이터를 단기 시계열 예측모델의 입력데이터로 사용하면 3개월 이후의 미래의 값을 예측하게 된다.

	a	b	c	d	e	f	g	h	i	j
2	52.55	51.11	31114.08	35254.27	39018.06	217045.8	562000	993	13751	365000
5	53.42	56.25	32971.50	35298.90	39565.80	195243.0	559000	1023	18765	402000
8	52.69	56.25	29212.54	30592.66	32432.82	210468.3	461000	877	9825	402000
11	55.44	56.31	32915.76	35658.74	38282.46	258794.2	70000	913	9349	402000
14	63.7	74.31	35200.62	38755.08	42080.22	355446.0	552000	1015	12975	493000
..
194	100.8	112.62	75938.19	76051.70	66062.82	355286.3	633300	955	17351	720000
197	103.79	111.70	71360.46	72746.10	63162.09	386824.5	567000	906	16444	730000
200	103.94	111.84	73087.82	74523.26	68183.40	377999.2	425000	704	14924	705000
203	102.85	108.61	75050.36	75397.28	69152.72	367735.2	519000	790	14485	630000
206	106.44	108.56	41026.38	41026.38	37130.70	427307.4	491000	764	17704	610000

<그림 5-9> 3개월 간격 데이터 시간 재구성 결과

5.2.3 예측 및 검증 방법

본 절의 예측과정은 데이터조합과 예측범위 확대에 따른 많은 기간에 대한 예측이 진행되기 때문에 데이터 조합탐색 수와 학습, Random search의 탐색 소요시간의 절감이 필요하다. 따라서 <그림 3-10>에서 선정한 10가지 데이터 특성 중 몇 가지를 제외하기 위해 식(3.1)인 피어슨 상관계수를 활용하였다. <그림 3-10>을 살펴보면 10가지 데이터 특성 중 고철 수입량, 철근 생산량, 건축 착공현황 데이터가 다른 데이터 특성보다 피어슨 상관계수 값이 다소 낮음을 알 수 있다. 따라서 본 절에서 사용하는 데이터셋에는 고철 수입량, 철근 생산량, 건축 착공현황 데이터 특성을 제외하였으며, 예측에 사용하는 데이터 특성은 <표 5-3>과 같다.

<표 5-3> 데이터셋 구성

ID	데이터 명칭	기간	출처
A	수입물가 지수 (철근)	2003.1 ~ 2020.3	통계청
B	생산자물가 지수 (철근)		통계청
C	두바이유 (\$/Ton)		e-나라지표
D	브렌트유 (\$/Ton)		e-나라지표
E	서부텍사스유 (\$/Ton)		e-나라지표
F	고철 스크랩 단가(\$/Ton)		e-나라지표
G	철근 가격(일반, SD300)		한국물가정보(Kpi)

예측은 사용데이터의 특성이 달라짐에 따라 3가지 방식의 과정을 통해 진행한다.

첫 번째는 사용데이터의 특성변경으로 데이터셋에 적합한 모델을 선택하기 위해 RNN, LSTM, Bi-LSTM 모델에 각각 새로 구성된 데이터셋을 적용하여 출력결과와 예측성능을 비교하고 이후의 예측과정에 사용할 한 가지 모델을 선택한다.

두 번째는 예측대상인 철근 가격데이터를 제외하고 나머지 6가지의 데이터 특성을 조합하여 이를 모델의 입력데이터로 사용하고, 출력된 결과 중에 학습과 예측성능이 우수한 데이터조합을 탐색한다. 데이터조합은 예측대상인 철근 가격을 제외하고 나머지 입력변수를 조합하여 데이터셋을 생성해 이를 사용하여 예측결과를 각각 출력하며, 탐색에 사용하는 데이터조합 수는 <표 5-4>와 같다.

<표 5-4> 데이터조합 개수

데이터조합	${}_6C_2$	${}_6C_3$	${}_6C_4$	${}_6C_5$	${}_6C_6$	합계
데이터셋 수	15	20	15	6	1	57

세 번째는 탐색한 데이터조합 중 가장 우수한 성능을 나타내는 데이터셋의 시간 간격을 재구성하여 모델에 적용한 후, 조정된 시간 간격에 따른 단기예측결과를 출력한다. 그리고 측정된 예측값과 한국물가정보에서 제공하는 해당 시점의 실제값과 비교하여 정확도(%)로 평가한다.

본 절의 모델에서 사용하는 하이퍼 파라미터의 탐색 범위는 <표 5-5>와 같다. 하이퍼 파라미터의 탐색 범위는 Manual search를 통해 일부 한정하였다. 입력데이터는 전체 데이터의 70%는 학습에 나머지 30%는 각각 15%씩 테스트 1, 테스트 2로 나눈다. 이는 데이터의 시간 단위가 재구성되면 늘어나는 시간 간격의 배수만큼 전체데이터양이 감소하기 때문에 테스트 구간의 성능 추정을 명확하게 하기 위함이다. 그리고 모델에서 출력되는 결과는 학습량이 5000까지 1씩 증가할 때마다 출력되고, 식(5.2), 식(5.3)에 의해 선별된다.

<표 5-5> 하이퍼 파라미터 입력범위

하이퍼 파라미터	설명
Random search의 탐색 수	50
시퀀스 길이	2 ~ 15
학습률	0.0001 ~ 0.1
은닉층 수	50 ~ 100
학습량	5000
드롭아웃 비율	0.1 ~ 1.0

$$0 < Test1R - TrainR < 0.01 \dots\dots\dots (5.2)$$

$$TrainR < Test2R < Test1R \dots\dots\dots (5.3)$$

여기서, $TrainR$ = 학습성능

$Test1R$ = 테스트1 성능

$Test2R$ = 테스트2 성능

식(5.2), 식(5.3)은 현재 과대적합을 판단할 수 있는 기준이 명확하지 않기 때문에 학습과 테스트1, 테스트2 성능 간의 관계를 특정하여 설정하였다. 그리고 식(5.2)는 앞서 설명한 식(4.1)과 반대의 경우로, 이에 대한 테스트가 필요할 것으로 판단되어 설정하였다. 출력결과는 모델의 하이퍼 파라미터와 학습, 테스트1, 2 RMSE, 예측값, 테스트 구간의 시각화가 포함된다. 기록된 결과에서는 학습성능이 가장 우수한 한 개의 결과를 선택한다.

5.2.4 예측결과 및 고찰

5.2.4.1 데이터셋에 적합한 딥러닝 모델 선정

본 연구에서 선정한 데이터셋에 적합한 딥러닝 모델 선택을 위한 각각의 모델별 단기예측 결과는 <표 5-6>과 같다.

<표 5-6> 데이터셋에 적합한 모델탐색 결과

딥러닝 모델	학습 RMSE	테스트1 RMSE	테스트2 RMSE	예측값 (원)	실제값 (원)	정확도 (%)
RNN	0.03487	0.03551	0.03549	637,000	610,000	95.76
LSTM	0.04847	0.04938	0.04894	657,000		92.84
Bi-LSTM	0.02857	0.02950	0.02935	609,000		99.83

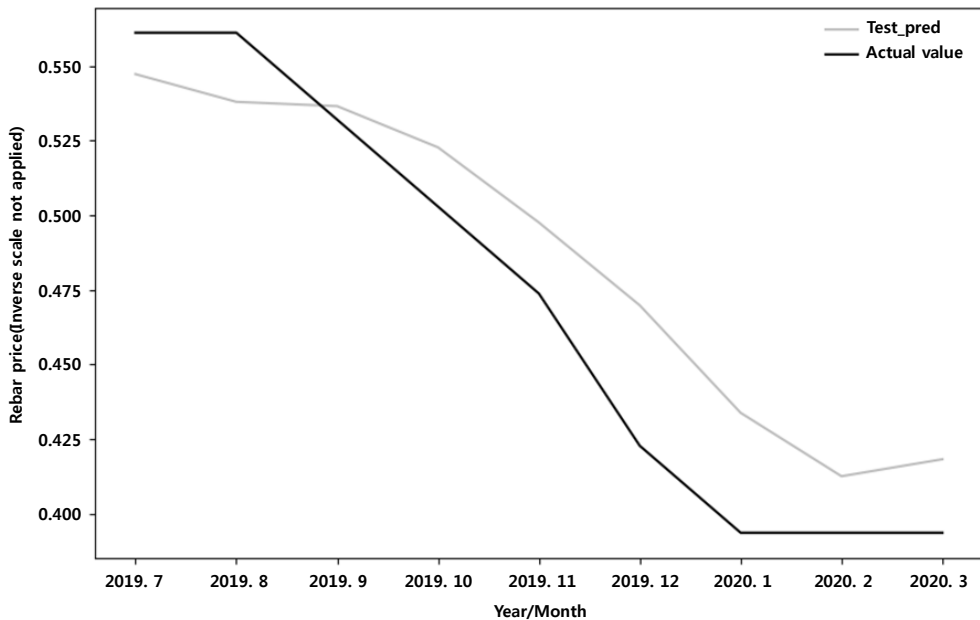
<표 5-6>은 철근 가격예측을 위해 6가지 데이터 특성으로 구성된 데이터셋을 3가지 모델에 적용하여 각각의 모델별로 1개월을 예측하였을 때 가장 우수한 예측성능을 나타낸 단기예측 결과를 정리한 것으로 Bi-LSTM이 모델별 학습, 테스트1, 2에 대한 RMSE 모두 우수한 결과를 나타냈다. 그리고 입력데이터의 범위를 벗어난 1달 후의 예측결과도 Bi-LSTM이 약 99.83%로 나머지 두 모델보다 우수하였다.

5.2.4.2 데이터조합탐색

선정된 6개의 데이터 특성을 조합하여 데이터셋으로 구성하고, 이를 통해 1개월의 단기 예측성능이 우수한 데이터조합을 탐색을 진행하였으며, 결과는 <표 5-7>과 같다. <표 5-7>을 살펴보면 학습 RMSE가 가장 우수한 데이터조합은 ${}_6C_3$ 에 ABF로 나타났으며, 테스트2 구간을 시각화한 결과는 <그림 5-10>과 같다. 그리고 2020년 4월 예측값의 정확도는 실제값의 약 98.85%를 나타냈다. 실제값에 대한 예측값의 정확도는 ${}_6C_6$ 의 ABCDEF가 약 99.83%로 가장 높게 나타났으나 학습, 테스트1, 2의 성능이 ${}_6C_6$ 의 ABCDEF 보다 우수한 ${}_6C_3$ 의 ABF 데이터셋을 본 연구에서는 최종 선택하였다.

<표 5-7> 데이터조합 탐색결과

데이터 조합	데이터셋	학습 RMSE	테스트1 RMSE	테스트2 RMSE	예측값 (원)	실제값 (원)	정확도 (%)
${}_6C_2$	AE	0.02989	0.03035	0.03016	604,000	610,000	99.33
${}_6C_3$	ABF	0.02475	0.02554	0.02536	617,000		98.85
${}_6C_4$	ADEF	0.03287	0.03351	0.03317	619,000		98.52
${}_6C_5$	ABCDE	0.03308	0.03338	0.03319	622,000		98.03
${}_6C_6$	ABCDEF	0.02857	0.02950	0.02935	609,000		99.83



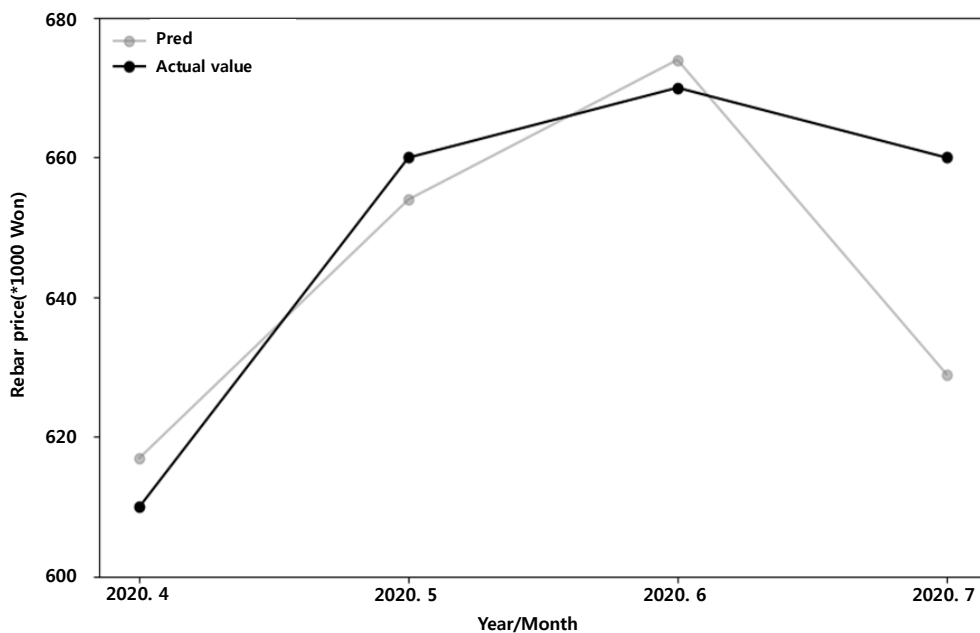
<그림 5-10> ABF 데이터 1개월 예측 테스트2

5.2.4.3 예측범위 확대를 위한 데이터의 시간 단위 재구성

예측범위를 확대하기 위한 데이터조합 탐색에서 선택된 ABF 데이터셋의 시간 간격을 재구성하여 예측한 결과는 <표 5-8>과 같다. 그리고 각각의 예측 월별 추정값을 실제값과 시각화한 결과는 <그림 5-11>과 같다.

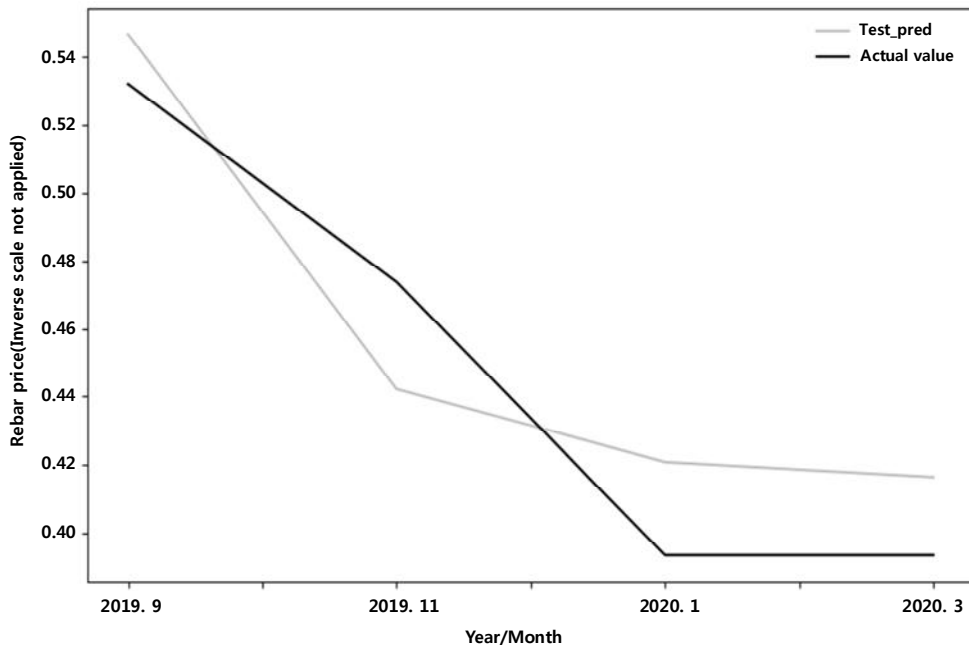
<표 5-8> 데이터 시간 간격 재구성에 따른 예측범위 확대결과

예측 월	학습 RMSE	테스트1 RMSE	테스트2 RMSE	예측값 (원)	실제값 (원)	정확도 (%)
2 (2020.5)	0.03270	0.03298	0.03187	654,000	660,000	99.09
3 (2020.6)	0.05498	0.06286	0.06197	674,000	670,000	99.40
4 (2020.7)	0.00785	0.01124	0.00983	629,000	660,000	95.30



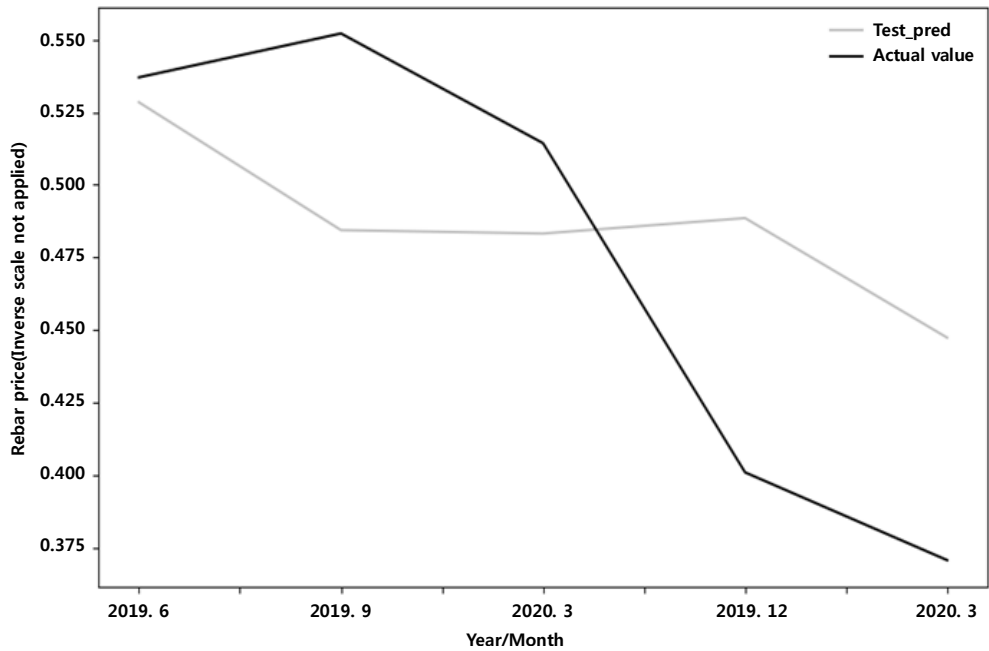
<그림 5-11> 4개월 예측결과

<표 5-8>의 결과 중에 데이터의 시간 간격을 2개월로 재구성한 데이터 셋으로 2020년 5월을 예측한 테스트2 결과는 RMSE 0.03187로 이 구간을 시각화하면 <그림 5-12>와 같다. 그리고 2020년 5월의 예측값은 약 654,000 원으로 실제값에 약 99.09%의 정확도로 예측하였다.

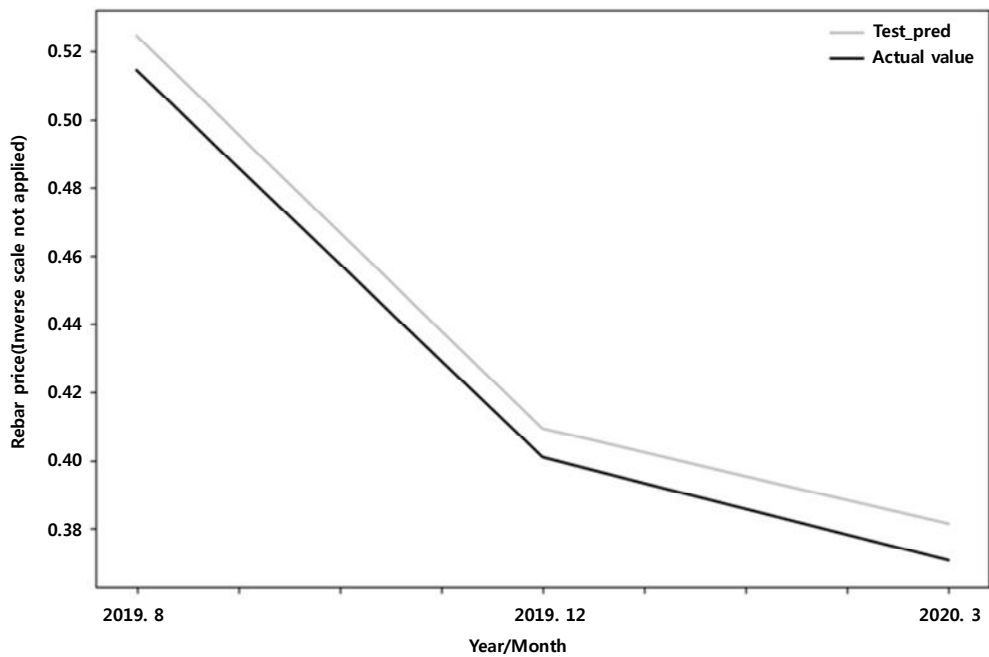


<그림 5-12> ABF 2개월 예측 테스트2 시각화

데이터의 시간 간격이 3개월인 데이터셋으로 2020년 6월을 예측한 테스트2 결과는 RMSE 0.06197로 이 구간을 시각화하면 <그림 5-13>과 같다. 2020년 6월의 예측값은 약 674,000원으로 실제값의 99.40%를 나타냈다. 데이터의 시간 간격이 4개월인 데이터셋으로 2020년 7월을 예측한 테스트2 결과는 RMSE 0.00983으로 이 구간을 시각화하면 <그림 5-14>와 같다. 2020년 7월의 예측값은 약 629,000원으로 실제값의 약 95.30%를 나타냈으며, 예측범위 4개월부터는 실제값에 대한 예측 정확도가 급격히 저하되었다. 이는 데이터의 시간 간격을 재구성하는 과정에서 늘어나는 시간 간격만큼 데이터가 배로 감소하기 때문에 학습성과 테스트1 사이에 과대적합이 발생하여 예측 정확도가 급격히 떨어진 것으로 추정된다.



<그림 5-13> ABF 3개월 예측 테스트2 시각화



<그림 5-14> ABF 4개월 예측 테스트2 시각화

5.3 소결

본 장에서는 철근 가격의 적정 구매 시기와 책정된 단가를 사전 파악하기 위해 2가지 방식을 통해 딥러닝의 예측범위 확대를 시도하였으며, 결과를 요약하면 다음과 같다.

1) 딥러닝의 예측범위를 확대하기 위해 입력과 정답데이터의 시점을 변경하는 방식으로 예측을 시도한 결과, 단기예측방식에서 예측범위를 1달에서 2달을 예측하도록 변경하면 학습과 테스트 성능, 예측 정확도가 모두 감소한다. 특히 예측범위를 3달로 확대 변경하면 단기예측성능인 약 97.05%에서 예측 정확도가 약 1.68% 감소한 95.37%를 나타냈다. 그리고 예측범위 3달의 시각화 결과에서는 예측값과 실제값이 다른 기간의 결과보다 큰 차이를 보이며 예측이 제대로 이뤄지지 않음을 알 수 있었다. 따라서 딥러닝의 예측범위 확대를 위해서는 다른 방식의 적용이 필요하다고 판단하였다.

2) 입력과 정답데이터의 시점을 변경하는 방식은 예측범위를 확장함에 따라 예측성능이 급격히 저하되는 문제가 발생한다. 따라서 예측범위를 확대하기 위해 데이터의 시간 단위를 재구성하고, 시간 단위가 조정된 데이터셋을 통해 단기예측을 진행하였다. 이 과정에서 학습과 탐색 소요시간을 줄이기 위해 데이터의 특성 일부를 제외하였기 때문에 데이터셋에 적합한 딥러닝 모델을 탐색을 우선으로 진행하였다. 결과는 RNN, LSTM, BI-LSTM의 모델 중 BI-LSTM이 99.83%의 정확도로 모델 중에서 가장 우수한 예측 정확도를 나타냈다. 그리고 예측대상인 철근 가격데이터를 제외한 5개의 데이터 특성을 조합하고, BI-LSTM의 입력데이터로 사용하여 단기예측의 정확도 향상을 위한 데이터조합 탐색을 진행하였다. 결과로는 수입물가지수, 생산자물가지수, 고철 단가로 구성된 데이터셋이 학습과 테스트 성능이 가장 우수하였고, 측정된 예측 정확도는 약 98.85%를 나타냈다.

마지막으로 수입물가지수, 생산자물가지수, 고철 단가, 철근 가격으로 구성된 데이터셋의 시간 간격을 조정하여 예측범위를 4개월까지 확대해본 결과 2개월부터 4개월까지 약 99.09%, 99.40%, 95.30%의 정확도를 나타냈으며, 4개월부터 예측 정확도가 4.1%로 감소하였다. 이는 데이터의 시간 간격조절에 의한 데이터양의 급격한 감소로 인해 과대적합이 발생하였기 때문으로 추정된다. 따라서 데이터의 시간 간격을 조절하는 방식으로 예측범위를 확장하는 방식은 1개월부터 3개월까지만 가능한 것으로 판단되며, 1개월부터 3개월까지의 평균예측정확도는 99.11%를 나타냈다.

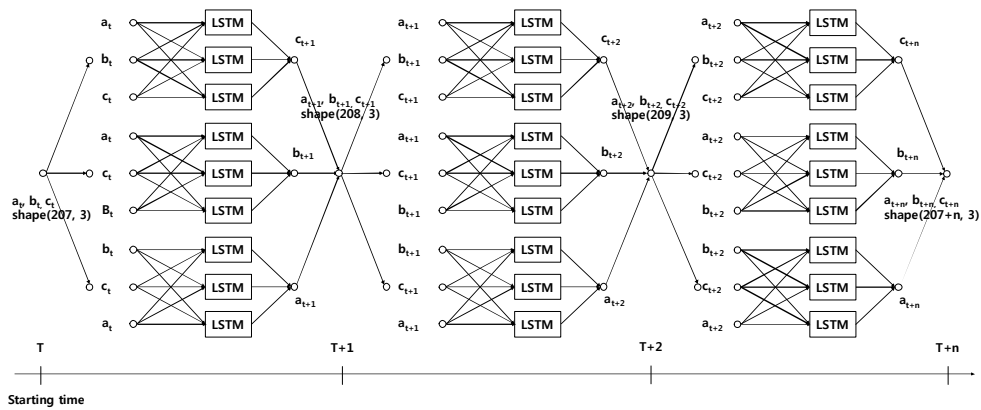
제6장 Recursive 예측방식을 활용한 철근 가격 장기예측

6.1 개요

앞장에서는 철근 구매 시기와 책정된 단가를 사전 파악하기 위해 데이터의 시간 간격을 조정하는 방식으로 예측 기간 3개월까지 예측범위를 확대할 수 있었다. 하지만 전반적인 건설 프로젝트의 기간을 고려할 때 앞선 연구에서 제시하는 예측범위는 다소 부족하다. 따라서 예측범위를 추가로 확대할 수 있는 새로운 방식의 적용이 필요하다.

6.2 Recursive 예측방법

Recursive 예측방식은 <그림 6-1>과 같이 현재 시점을 T 라 할 때 입력데이터의 특성을 각각 예측대상으로 모두 개별적 단기예측한다.



<그림 6-1> Recursive 예측방식 도식화

그리고 예측된 $T+1$ 시점 데이터를 T 까지 구성된 원 데이터에 추가한 후, $T+1$ 까지로 구성된 데이터를 모델의 입력데이터로 사용하여 $T+n$ 시점까지 예측범위를 확대하는 방식이다(Taieb, Bontempi 2010).

하지만 이 방식은 데이터셋으로 구성된 데이터 특성들만으로, (a_t , b_t ,

c_t)을 통해 c_{t+1} 을 (b_t, c_t, a_t)로 a_{t+1} 을 (a_t, c_t, b_t)로는 b_{t+1} 을 예측하기 때문에 데이터셋으로 이를 모두 잘 예측할 수 있는 모델의 선정이 필요하다. 그리고 예측대상 수가 많으면 예측의 과정도 예측대상의 수만큼 같이 증가하기 때문에 예측범위 확대에 많은 시간이 소요될 수 있다(이용성 & 김경환, 2021).

6.3 데이터 선정

본 장의 연구에서 사용할 Recursive 예측방법의 특성상 데이터셋으로 구성된 데이터 특성만으로 상호 간 예측이 가능해야 한다. 따라서 데이터 특성 간 선형상관관계 유의성이 높은 데이터를 선택하기 위해 식(3.1)인 피어슨 상관관계 분석을 활용하였다. <그림 3-10>을 살펴보면 수입물가지수인 A, 생산량 물가지수 B, 국제유가 C, D, E, 고철 가격 F, 철근 가격 J가 상호 간 강한 양의 상관관계를 나타낸다. 그리고 본 연구에서 사용할 데이터셋은 데이터 특성 수 만큼 예측의 대상 수도 같이 늘어나 학습과 탐색으로 소요시간이 증가하는 것에 대한 고려가 필요하다. 따라서 강한 양의 상관관계를 나타내는 데이터 중에서 피어슨 상관관계수가 0.9 이상을 나타내는 A, B, J만을 선택하여 데이터셋으로 구성하였다. 사용 데이터의 기간은 2003년 1월부터 2020년 3월까지이다.

6.4 예측 및 검증 방법

Recursive 예측방식을 활용한 철근 가격의 장기예측은 2가지 과정을 통해 진행한다.

첫 번째, 본 연구에서 선정한 수입물가지수, 생산자물가지수, 철근 가격으로 구성된 데이터셋을 순환신경망 모델인 RNN, LSTM, Bi-LSTM에 입력데이터로 사용하여 예측대상에 대한 단기($t+1$)예측성능이 모두 우수하게 측정되는 모델을 탐색한다.

두 번째, 첫 번째 과정으로 선택된 모델을 통해 예측대상인

수입물가지수, 생산자물가지수, 철근 가격을 모두 단기($t+1$) 예측하고, 이후 예측값($t+1$)을 원 데이터의 가장 마지막 행으로 추가한 후, 다음 시점($t+2$)을 예측한다. 또한, 이 과정을 반복하여 철근 가격예측이 가능한 유효범위($t+n$)와 이에 따른 예측성능을 측정한다. 추가로 이 방법의 유효성을 확인하고자 총 3회 반복측정한다. 예측결과는 각각의 단기 예측값을 실제값과 비교하여 정확도로 나타낸다.

데이터의 분할비율은 학습은 80% 나머지 20%는 각각 10%씩 테스트1과 테스트2로 분할된다. 그리고 각각의 단기 예측과정은 Random search에 의해 하이퍼 파라미터를 무작위로 조합하여 탐색을 수행한다. 하이퍼 파라미터는 Manual search로 한정하였고, 탐색 범위는 <표 6-1>과 같다.

<표 6-1> Recursive 예측의 하이퍼 파라미터 입력범위

하이퍼 파라미터	범위
시퀀스 길이	2 ~ 10
학습률	0.0001 ~ 0.1
은닉층 수	10 ~ 150
학습량	5000
드롭아웃 비율	0.1 ~ 1.0

출력결과는 모델에서 학습량이 5000까지 1씩 증가할 때마다 측정한다. 그리고 측정된 결과 중에서는 식(5.2)와 식(5.3)의 조건을 충족할 경우만 기록된다. 이후 기록된 결과에서는 학습 RMSE가 가장 좋은 결과를 선택한다. 테스트2 데이터는 모델의 학습과 검증과정에 이용되지 않으며, 예측성능을 측정하는 용도로만 사용한다.

6.5 예측결과 및 고찰

Recursive 예측방식을 활용한 첫 번째, 3가지 예측대상에 대한 모델 탐색결과는 <표 6-2>와 같다.

<표 6-2> 모델탐색 결과(실제 예측 시점, 2020. 4)

예측 대상	딥러닝 모델	학습 RMSE	테스트1 RMSE	테스트2 RMSE	1개월 예측	실제값	정확도 (%)
A	RNN	0.04682	0.04768	0.04760	93.22	106.29	87.70
	LSTM	0.02361	0.02407	0.02390	104.05		97.89
	Bi-LSTM	0.01855	0.01954	0.01881	105.92		99.65
B	RNN	0.05243	0.05327	0.05287	103.97	109.50	94.95
	LSTM	0.02287	0.02373	0.02309	106.40		97.17
	Bi-LSTM	0.01532	0.01622	0.01612	106.85		97.58
J	RNN	0.02542	0.02620	0.02600	630,000	610,000	96.83
	LSTM	0.02553	0.02606	0.02599	629,000		96.98
	Bi-LSTM	0.02586	0.02623	0.02602	628,000		97.13

<표 6-2>는 RNN, LSTM, Bi-LSTM에 데이터셋을 (A, B, J), (B, J, A), (A, J, B)의 형태로 입력하여 예측대상인 J, A, B를 단기예측한 결과로, 이를 살펴보면 LSTM과 Bi-LSTM으로 B, J를 예측한 결과는 1% 미만의 정확도 차이로 Bi-LSTM이 더 나은 예측 정확도를 나타냈으나, A를 예측한 결과에서는 Bi-LSTM이 LSTM보다 약 1.76% 높은 예측 정확도 차이를 나타냈다. RNN은 나머지 두 모델보다 예측 정확도가 낮은 결과를 나타냈다. 결과를 통해 이후의 과정에서는 Bi-LSTM 모델을 사용하여 진행하였다.

두 번째는 (A, B, J), (B, J, A), (A, J, B)의 순서로 구성된 데이터셋을 Bi-LSTM 모델에 각각 입력데이터로 사용하고, 이를 앞서 설명한 반복적 예측방식을 통해 2020년 4월을 시작으로 9월까지 월별 예측을 진행하였다. 그리고 결과의 신뢰성 확보를 위해 이 과정을 3회 반복 실시하여 <표 6-3>, <표 6-4>로 정리하였다. <표 6-3>은 예측대상과 예측 기간, 예측 차시별로 학습데이터, 테스트1, 테스트2에 대한 예측성능을 측정하여 기록한 결과로, MAPE를 통해 측정된 결과가 나타내는 오차를 퍼센트(%)로 확인할 수 있다. 그리고 <표 6-4>는 예측 차시 및 월별 예측값을 실제값과 비교하여 정확도로 정리하였으며, <표 6-4>의 월별 예측값을 시각화한 결과는 <그림 6-1>, <그림 6-2>, <그림 6-3>과 같다.

예측결과의 주요 내용으로 <표 6-4>를 살펴보면 총 6개월의 예측 기간 중 데이터 B가 5개월 예측값(t+5)부터 예측 정확도가 감소하기 시작한다. 또한, 이를 시작으로 6개월 예측값(t+6)부터 데이터 A, B, J의 예측 정확도도 감소하기 시작한다. 이는 6개월(t+6)을 예측하기 위해 사용한 5개월의 입력데이터가 많은 부분 오차가 누적된 예측값(t+5)으로 대체되었기 때문으로 추정된다. 최근 값을 예측하는 방법은 모델의 최종 출력에 해당하는 완전연결계층(Fully connected layer)에 저장된 가중치(W)와 바이어스(B)를 시퀀스 길이 만큼의 최근 데이터와 식(6.1)과 같이 연산하여 최종적으로 한 개의 예측값을 출력한다.

$$Recent pred_{(t+5)} = Fully\ connected \left(\begin{pmatrix} a_{(t+1)} & b_{(t+1)} & j_{(t+1)} \\ a_{(t+2)} & b_{(t+2)} & j_{(t+2)} \\ a_{(t+3)} & b_{(t+3)} & j_{(t+3)} \\ a_{(t+4)} & b_{(t+4)} & j_{(t+4)} \end{pmatrix} * W + B \right) \dots (6.1)$$

여기서, W = 가중치

B = 편향

$t+5$ = 5개월 예측값

<표 6-3> Recursive 장기예측 결과(학습 및 테스트 성능, RMSE)

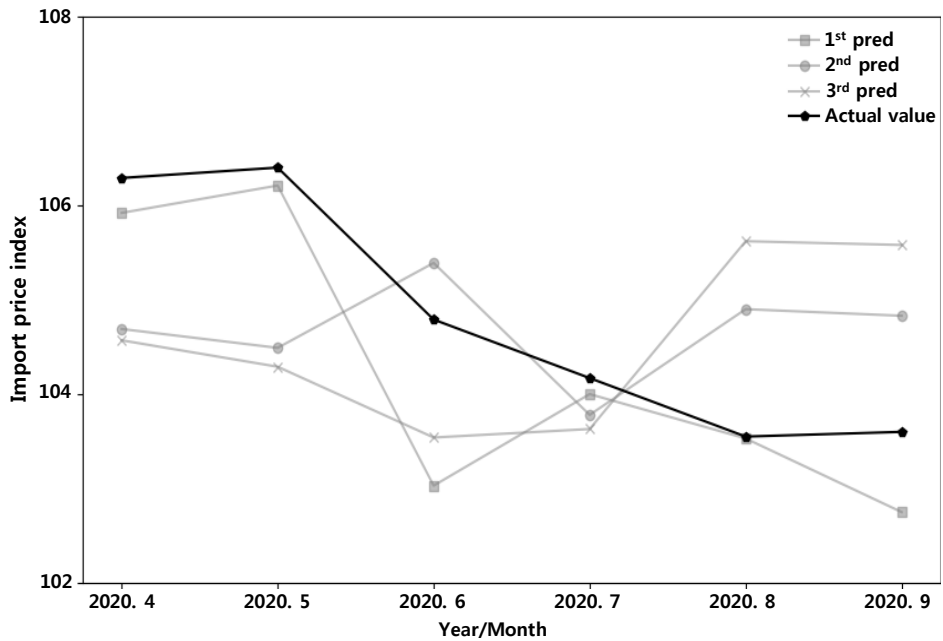
Taget	Pred month	RMSE(×1000)								
		학습			테스트1			테스트2		
		1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd
A	1	18.55	15.67	15.59	19.54	16.61	16.05	18.81	15.63	15.64
	2	20.25	17.00	16.42	21.10	17.84	16.79	20.74	16.35	16.51
	3	16.43	18.49	17.87	16.99	19.28	18.65	16.46	17.00	18.47
	4	14.81	15.94	13.63	15.37	16.77	14.29	15.22	16.43	14.22
	5	15.52	13.20	18.41	16.29	13.75	18.93	16.29	13.48	18.65
	6	15.28	15.87	16.98	15.92	16.65	17.85	15.45	16.23	16.99
B	1	15.31	17.87	15.57	16.22	18.49	15.79	16.12	13.08	15.67
	2	22.52	16.45	24.39	23.49	17.38	25.39	23.13	16.58	24.58
	3	20.18	15.7	20.51	21.05	16.47	21.48	20.93	12.77	20.97
	4	14.27	16.31	24.93	15.25	16.82	25.52	14.40	12.98	25.11
	5	21.49	17.61	13.06	22.3	18.49	14.01	21.86	17.57	13.42
	6	24.81	19.40	22.00	25.53	20.19	22.87	25.28	19.86	22.76
J	1	25.86	26.52	26.22	26.22	26.71	27.21	26.01	26.65	26.51
	2	24.56	26.88	24.97	24.84	27.60	25.94	24.73	26.59	25.66
	3	25.07	26.52	25.92	25.99	27.33	26.86	25.41	26.44	26.68
	4	26.18	25.37	24.72	27.01	26.32	25.66	26.9	25.64	25.35
	5	27.46	25.46	25.23	28.24	26.43	26.15	27.83	25.16	25.49
	6	26.76	24.58	25.80	27.74	25.23	26.79	27.02	25.15	26.37

<표 6-3> Recursive 장기예측 결과(테스트 성능, MAPE)

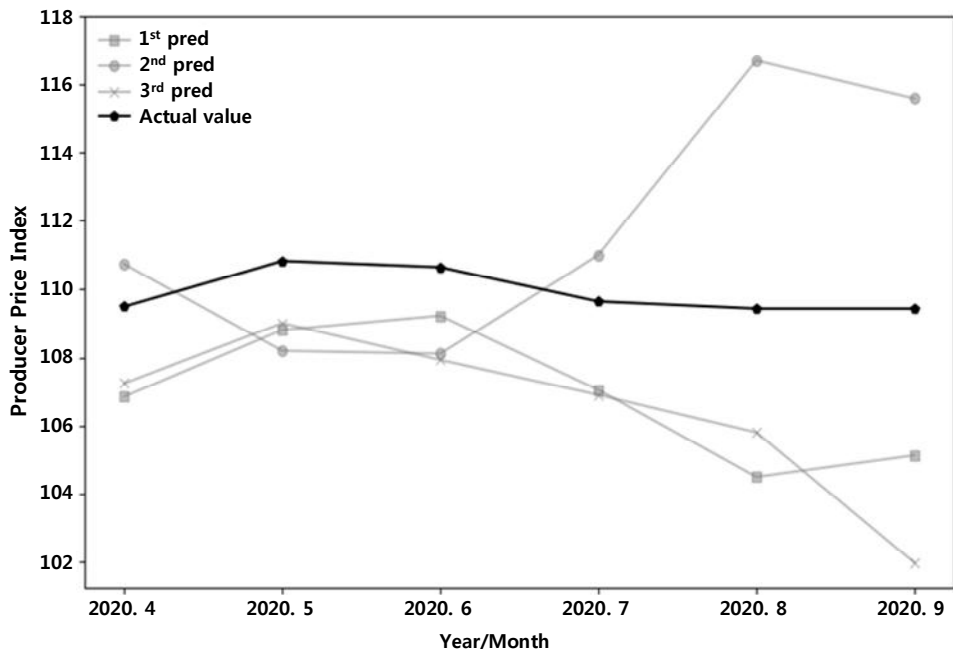
Taget	Pred month	MAPE(%)					
		테스트1			테스트2		
		1st	2nd	3rd	1st	2nd	3rd
A	1	1.73	1.57	1.46	1.63	1.41	1.33
	2	1.98	1.69	1.69	1.89	1.26	1.30
	3	1.56	1.78	1.67	1.42	1.51	1.45
	4	1.44	1.50	1.30	1.20	1.41	1.26
	5	1.40	1.24	1.67	1.40	1.12	1.37
	6	1.49	1.56	1.57	1.27	1.38	1.40
B	1	1.15	1.35	1.11	1.13	0.79	1.03
	2	1.67	1.10	1.69	1.68	1.12	1.82
	3	1.51	1.23	1.25	1.53	0.85	1.56
	4	0.91	1.24	1.75	0.88	0.98	1.93
	5	1.67	1.26	1.02	1.42	1.01	0.97
	6	1.83	1.36	1.48	1.93	1.12	1.44
J	1	2.32	2.39	2.37	2.13	2.25	2.25
	2	1.91	2.32	2.02	2.33	2.03	2.28
	3	2.06	2.26	2.27	1.98	2.28	2.16
	4	2.25	2.28	1.92	2.30	2.08	2.36
	5	2.34	2.12	2.08	2.15	2.23	2.28
	6	2.34	2.05	2.24	2.36	2.07	2.08

<표 6-4> Recursive 장기예측 결과 (예측값의 정확도, %)

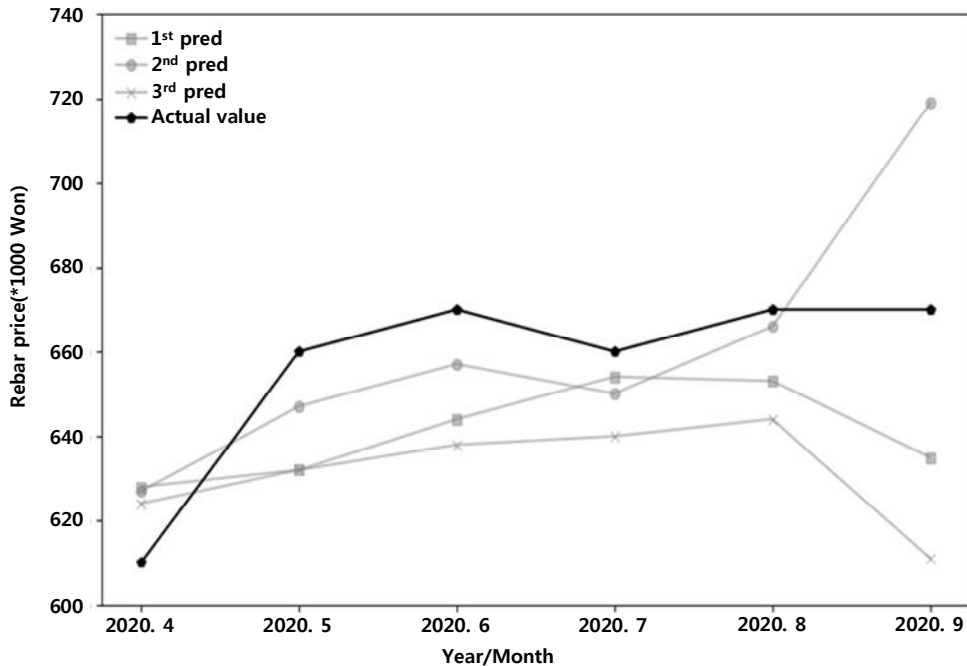
예측 대상	예측 월	실제값	예측값			정확도(%)			
			1st	2nd	3rd	1st	2nd	3rd	Ave.
A	1	106.29	105.92	104.69	104.57	99.65	98.49	98.38	98.84
	2	106.40	106.21	104.49	104.29	99.82	98.20	98.02	98.68
	3	104.79	103.03	105.39	103.54	98.32	99.43	98.81	98.85
	4	104.17	104.00	103.78	103.63	99.84	99.63	99.48	99.65
	5	103.55	103.53	104.90	105.62	99.98	98.71	98.04	98.91
	6	103.6	102.75	104.83	105.58	99.18	98.83	98.12	98.71
B	1	109.50	106.85	110.74	107.25	97.58	98.88	97.94	98.13
	2	110.83	108.81	108.20	109.01	98.18	97.63	98.36	98.06
	3	110.65	109.23	108.11	107.94	98.72	97.70	97.55	97.99
	4	109.65	107.05	111.00	106.90	97.63	98.78	97.49	97.97
	5	109.44	104.49	116.70	105.81	95.48	93.78	96.68	95.31
	6	109.44	105.12	115.59	101.97	96.05	94.68	93.17	94.63
J	1	610,000	628,000	627,000	624,000	97.13	97.29	97.76	97.39
	2	660,000	632,000	647,000	632,000	95.76	98.03	95.76	96.52
	3	670,000	644,000	657,000	638,000	96.12	98.06	95.22	96.47
	4	660,000	654,000	650,000	640,000	99.09	98.48	96.97	98.18
	5	670,000	653,000	666,000	644,000	97.46	99.40	96.12	97.66
	6	670,000	635,000	719,000	611,000	94.78	93.18	91.19	93.05



<그림 6-1> 수입물가지수 예측결과 시각화



<그림 6-2> 생산자물가지수 예측결과 시각화



<그림 6-3> 철근 가격 예측결과 시각화

이때 실제값으로만 구성된 학습, 테스트1 데이터로 일반화가 된 가중치와 바이어스는 실제값으로 구성된 테스트1 데이터 구간에 대한 오차는 작게 발생시킨다. 하지만 미래의 값들을 예측하기 위해 사용하는 데이터 구간에는 오차가 누적된 예측값들이 많은 부분 대체 되어있기 때문에 완전연결계층의 연산을 통해 출력된 예측값($t+6$)은 예측 정확도가 크게 낮아지는 것으로 추정할 수 있다. 이러한 문제점은 하이퍼 파라미터인 시퀀스 길이를 최근값을 예측하기 위해 사용한 데이터 크기보다 크게 설정하여 실제 데이터를 더 많이 포함 시키면 예측 정확도가 개선될 것으로 추정된다.

6.6 소결

본 장에서는 예측된 결과를 원 데이터에 추가하고 다음 시점을 예측하는 방식으로 예측범위를 확장해 나가는 Recursive 예측방식과 순환신경망 모델을 활용하여 예측범위 확대를 진행하였다. 결과를 요약하면 다음과 같다.

Recursive 예측방식에 사용할 모델을 결정하기 위해 RNN, LSTM, Bi-LSTM을 사용하여 예측대상인 수입물가지수, 생산자물가지수, 철근 가격에 대한 단기예측성능이 모두 우수하게 측정되는 모델탐색을 진행하였다. 탐색결과 Bi-LSTM을 사용하였을 때가 다른 두 가지 모델보다 실제값에 대한 예측성능이 모두 우수하였고, 측정된 결과는 수입물가지수는 약 99.65%, 생산자물가지수 약 97.57%, 철근 가격 약 97.13%의 정확도를 나타냈다. 그리고 탐색결과에 따라 결정된 Bi-LSTM 모델과 Recursive 예측방식을 통해 2020년 4월부터 9월까지 월별로 예측을 진행하였다. 예측결과 총 6개월의 예측 기간 중 생산자물가지수가 5개월 예측값부터 예측 정확도가 감소하기 시작한다. 또한, 이를 시작으로 6개월 예측값부터 철근 가격, 수입물가지수의 예측 정확도도 감소하기 시작한다. 이는 6개월을 예측하기 위해 사용한 5개월의 입력데이터가 많은 부분 오차가 누적된 예측값으로 대체되었기 때문으로 추정하였다. 따라서 이러한 문제점을 해결하기 위해서는 하이퍼 파라미터인 시퀀스 길이를 최근값을 예측하기 위해 사용한 데이터 크기보다 크게 설정하여 실제 데이터를 더 많이 포함 시키면 예측 정확도가 개선될 것으로 추정하였다.

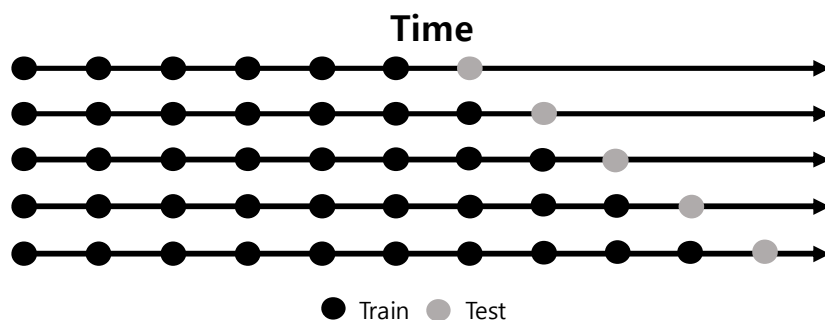
제7장 Recursive 예측방식과 시계열 교차검증을 활용한 장기예측

7.1 개요

Recursive 예측방식을 활용한 기존의 연구에서는 측정된 예측값의 신뢰구간 검증을 위해 홀드 아웃 검증방식을 사용하여 장기적 시점에 대한 예측을 3회 반복 실시하였다. 하지만 홀드 아웃 검증방식을 통해 측정된 테스트 구간 RMSE와 MAPE는 수치의 작은 차이에도 시차별로 측정된 예측값이 차이를 보이는 경우가 발생한다. 그리고 철근 가격예측에 사용하는 학습 데이터양은 다른 빅데이터 기반의 기계학습과 비교할 때 절대적으로 부족하여 과대적합의 발생으로 이와 같은 문제의 직접적 원인이 될 수 있다. 따라서 데이터의 많은 부분을 학습과 테스트로 활용할 수 있고, 기존에 사용한 홀드 아웃 방식보다 명확한 성능 추정이 가능한 검증방식의 적용이 필요하다.

7.2 시계열 교차검증 적용

일반적인 k-겹 교차검증방식과 달리 시계열 예측에 사용되는 교차검증방식은 분할을 위해 k=5를 사용하는 경우 <그림 7-1>과 같이 데이터의 시간 순서를 유지하며 분할된다.



<그림 7-1> 시계열 교차검증 (박성우 외, 2019)

이러한 과정은 모델 내부에서 scikit-learn의 ‘TimeSeriesSplit(n_splits=k)’에 의해 자동으로 진행되며, 본 연구에서 사용하는 데이터셋을 k=5를 사용하여 분할하는 경우<그림 7-2>와 같이 5개의 데이터셋으로 나뉜다⁶⁾.

Train_1 [0 1 2 ... 37 38 39]	Test_1 [40 41 42 ... 77 78 79]
Train_2 [0 1 2 ... 77 78 79]	Test_2 [80 81 82 ... 117 118 119]
Train_3 [0 1 2 ... 117 118 119]	Test_3 [120 121 122 ... 157 158 159]
Train_4 [0 1 2 ... 157 158 159]	Test_4 [160 161 162 ... 197 198 199]
Train_5 [0 1 2 ... 197 198 199]	Test_5 [200 201 202 ... 237 238 239]

<그림 7-2> 시계열 교차검증적용 예시

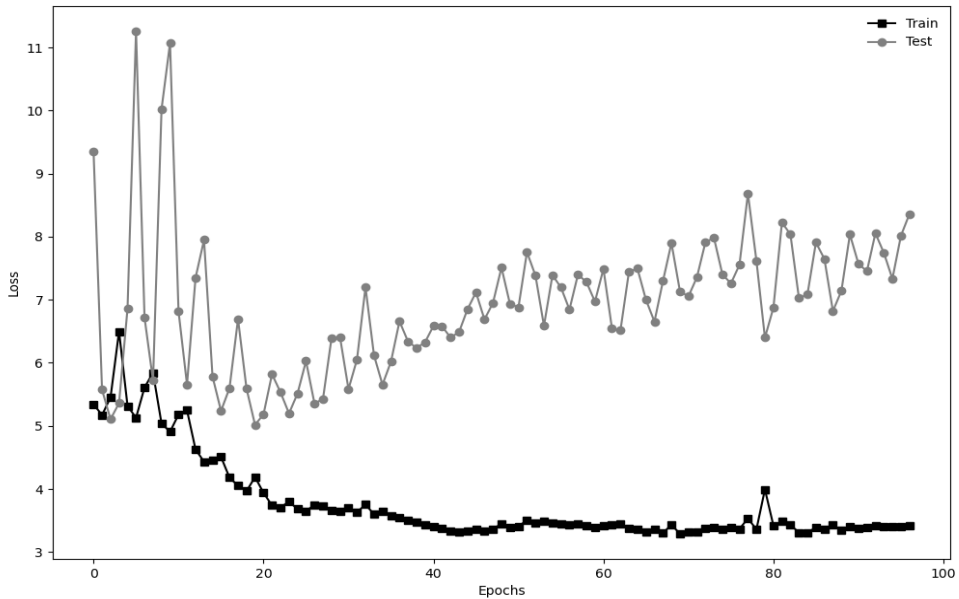
이에 따라 하나의 모델에서 한 값을 예측하기 위해서는 5개의 데이터셋에 대한 학습과 테스트 과정의 진행이 필요하다. 또한, 본 연구의 예측대상인 철근 가격, 수입물가지수, 생산자물가지수를 Recursive 방식을 통해 단기예측하게 되면 총 15번의 학습과 테스트 과정이 진행된다. 모델의 성능평가는 각각의 예측대상별로 진행된 5번의 학습과 테스트 구간을 이용하여 각각 예측성능을 측정하고, 측정된 성능을 평균 결함한다.

그리고 교차검증이 적용되면 <그림 7-2>와 같이 데이터셋의 가장 앞부분에 해당하는 ‘Train_1’과 ‘Test_1’을 사용하는 데이터셋 1번의 크기가 0부터 79까지 80행으로 작아진다. 이에 따라 학습이 제대로 이뤄지지 않을 수 있다. 따라서 학습량(Iterations)을 통해 진행하던 기존의 방식을 ‘Epoch’의 반복 학습의 형태로 변경하여 적용한다. 반복 학습은 ‘1 Epoch’이 학습데이터가 신경망을 한번 통과한 것을 의미한다.

7.3 조기 종료

본 장의 연구에서 사용하는 모델의 학습방식을 반복 학습방식으로 변경함에 따라 학습이 진행되는 과정에서 측정되는 학습과 테스트 성능의 변화를 기록하여 <그림 7-3>과 같이 나타냈다.

6) Scikit-learn.org, sklearn.model_selection.TimeSeriesSplit, accessed Sep 27, 2021, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html



<그림 7-3> 누적 학습량 증가에 따른 학습과 테스트 성능변화

이를 살펴보면 학습이 진행됨에 따라 ‘Train’인 학습성능의 값은 그래프의 전체적인 경향이 지속해서 낮아지는 것에 반해 ‘Test’인 테스트 성능은 ‘20 Epochs’ 까지는 감소하는 경향을 나타내다 증가한다. 이는 학습이 진행됨에 따라 측정되는 테스트 성능이 반복 학습의 20 이후로 예측 오차가 지속적으로 커지는 것을 의미하며, 이러한 경향은 단순히 학습량(Iteration)만으로 학습을 진행하는 <그림 4-4>보다, ‘Epoch’에 의한 반복 학습방식에서 뚜렷이 확인할 수 있다. 하지만 학습 과정을 ‘Epoch’의 형태로 변경하면 ‘1 Epoch’은 학습 데이터셋이 신경망을 한번 통과한 것을 의미하기 때문에 ‘Epoch’의 수에 따라 학습시간이 늘어난다. 그리고 시계열 교차검증의 데이터 분할 수 $k=5$ 를 사용할 경우 5배의 학습시간이 추가로 늘어난다. 그리고 여기에 <그림 2-8>과 같은 모델 복잡도에 따른 훈련과 테스트 정확도의 변화점에서 과소적합과 과대적합의 최적점을 탐색하기 위해 사용한 식(5.2), 식(5.3)과 같은 조건을 적용할 경우 해당 조건에 충족하는 결과를 탐색하기 위해 늘어나는 학습 소요시간만큼의 불필요한 모든 과정을 진행해야 한다. 따라서 본 장의 예측모델에는 학습시간의 효율적 감소를 위해 조기 종료(Early stopping) 기법을 추가하였다.

조기 종료는 측정되는 테스트 성능 값의 변화를 모니터하여 반복 학습이 진행될 때 가장 낮게 측정된 값을 저장한다. 그리고 이보다 더 낮은 값이 측정되는 경우 저장된 값을 갱신한다. 또한, 값의 변화가 없거나 연속적으로 증가하는 경우 학습을 중단시킨다⁷⁾.

7.4 예측 및 검증 방법

예측 및 검증 방법에 대한 설명에 앞서 본 장에서 제시하는 예측모델의 사용데이터는 연구의 작성 시점에 맞춰 업데이트하였으며, 데이터의 기간은 2003년 1월부터 2020년 11월까지를 기간으로 구성하였다.

Recursive 예측방식에 시계열 교차검증을 적용한 철근 가격의 장기예측은 두 가지 과정으로 예측이 진행된다.

첫 번째, 검증방식의 변경으로 철근 가격 단기예측의 성능과 예측 정확도를 확인할 필요가 있다. 이를 위한 확인방법으로는 실제값과 예측값의 비교를 통해 정확도를 확인할 수 있는 가장 최근 시점인 2020년 12월부터 2021년 4월까지의 5개월을 각각 단기예측한다. 그리고 실제값에 대한 예측 정확도를 측정한다. 이때 사용되는 데이터셋의 기간은 예를 들어, 2020년 12월을 예측하기 위해서는 2003년 1월부터 2020년 11월까지의 기간으로 구성된 데이터셋을 사용하며, 이후의 기간은 삭제하고 구성한다. 마찬가지로 2021년 4월의 예측은 2003년 1월부터 2021년 3월까지의 기간으로 구성된 데이터셋을 사용한다.

두 번째, 앞 장의 예측과정과 같이 생산자물가지수, 수입물가지수, 철근 가격을 모두 단기예측하고, 예측값을 원 데이터의 가장 마지막 행으로 추가한 후 다음 시점을 예측하는 Recursive 예측과정을 반복하여 $t+n$ 시점까지 철근 가격을 예측한다.

하이퍼 파라미터인 시퀀스 길이는 예측을 위해 사용한 데이터에 실제값을 더 많이 포함하기 위해 최소, 최대 범위 값을 더 크게 설정하였다. 학습량은 반복 학습의 형태로 변경되었기 때문에 1번의 반복 학습안에서는

7) Tensorflow.org, tf.keras.callbacks.EarlyStopping. accessed Sep 27, 2021, https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

기존의 방식과 동일한 학습량의 크기만큼 학습이 진행되고, 종료 시 2번째 반복 학습이 진행된다. 총 반복 학습량은 100회를 입력하였지만, 조기 종료에 의해 100회가 모두 진행되지 않고 종료될 수 있다. 하이퍼 파라미터의 탐색 범위는 <표 7-1>과 같다. 출력결과는 5개의 측정된 학습 및 테스트 결과에서 산출된 RMSE, MAPE를 평균하고, 평균값이 낮은 결과를 최종 선택한다.

<표 7-1> Recursive 예측의 하이퍼 파라미터 입력범위

하이퍼 파라미터	범위
반복 학습량	100
학습량	5000
학습률	0.001 ~ 0.01
시퀀스 길이	6 ~ 12
은닉층 수	10 ~ 150

7.5 예측결과 및 고찰

시계열 교차검증을 적용한 첫 번째 철근 가격 단기예측 결과는 <표 7-2>와 같다. 결과의 주요 내용으로는 측정된 결과 모두 시계열 교차검증 데이터 구간 1, 2에서 테스트의 성능이 높게 측정되었으며, 이는 학습으로 사용할 데이터양이 시계열 교차검증으로 인해 분할되어 작아지기 때문으로 판단된다. 그리고 측도의 종합 평균에 의한 학습과 테스트 성능은 각각 월별로 측정된 결과에서 학습은 최대 1.11%, 테스트는 약 2.02%의 차이를 보였으며, 대부분 유사한 성능을 나타내며 측정되었다. 또한, 5개월에 대한 예측값의 정확도는 2020년 12월을 시작으로 약 98.44%, 97.47%, 98.37%, 95.74%, 99.23%를 나타냈으며, 종합 평균은 약 97.85%로 측정되었다.

이와 같은 단기예측의 성능을 기반으로 2003년 1월부터 2020년

11월까지를 데이터셋으로 구성하여 2020년 12월부터 2021년 4월까지 생산자물가지수, 수입물가지수, 철근 가격을 Recursive 방식으로 장기예측한 두 번째 예측결과는 <표 7-3>, <표 7-4>, <표 7-5>와 같다. 그리고 측정된 예측값은 <그림 7-4>, <그림 7-5>, <그림 7-6>과 같이 시각화하였다.

<표 7-2> 철근 가격 단기예측 결과

예측 시점 (월)	데이터 구간	학습 RMSE	테스트 RMSE	학습 MAPE (%)	테스트 MAPE (%)	예측값 (원)	실제값 (원)	정확도 (%)
1 (2020. 12)	1	0.15852	0.38210	7.74	13.92	706,000	695,000	98.44
	2	0.11093	1.23267	4.59	29.57			
	3	0.07104	0.03280	5.79	2.37			
	4	0.06559	0.08513	6.14	8.83			
	5	0.05839	0.05462	4.80	4.70			
	평균	0.09289	0.35746	5.81	11.88			
2 (2021. 01)	1	0.14134	0.33160	8.52	14.54	713,000	695,000	97.48
	2	0.09897	1.01710	3.80	24.03			
	3	0.06134	0.03109	5.47	2.33			
	4	0.05506	0.06270	3.97	6.20			
	5	0.05642	0.04874	4.27	4.05			
	평균	0.08263	0.29843	5.21	10.23			

<표 7-2> 철근 가격 단기예측 결과(계속)

예측 시점 (월)	데이터 구간	학습 RMSE	테스트 RMSE	학습 MAPE (%)	테스트 MAPE (%)	예측값 (원)	실제값 (원)	정확도 (%)
3 (2021. 02)	1	0.16026	0.37152	9.98	16.60	737,000	725,000	98.37
	2	0.10300	1.08720	4.47	28.26			
	3	0.07446	0.03874	6.30	2.91			
	4	0.06213	0.07612	5.75	7.52			
	5	0.06357	0.06648	5.80	5.95			
	평균	0.09268	0.32801	6.46	12.25			
4 (2021. 03)	1	0.16031	0.35343	9.93	15.81	742,000	775,000	95.74
	2	0.10623	1.01742	4.68	26.82			
	3	0.07578	0.04359	6.55	3.45			
	4	0.06173	0.07410	5.66	7.34			
	5	0.05790	0.05122	4.90	4.24			
	평균	0.09239	0.30811	6.34	11.53			
5 (2021. 04)	1	0.17198	0.41349	10.46	18.44	769,000	775,000	99.23
	2	0.08979	0.67379	4.82	19.79			
	3	0.07256	0.03317	6.05	2.51			
	4	0.06154	0.07322	5.40	7.02			
	5	0.05821	0.05232	4.87	4.30			
	평균	0.09082	0.24920	6.32	10.41			

<표 7-3> 수입물가지수 5개월 예측결과

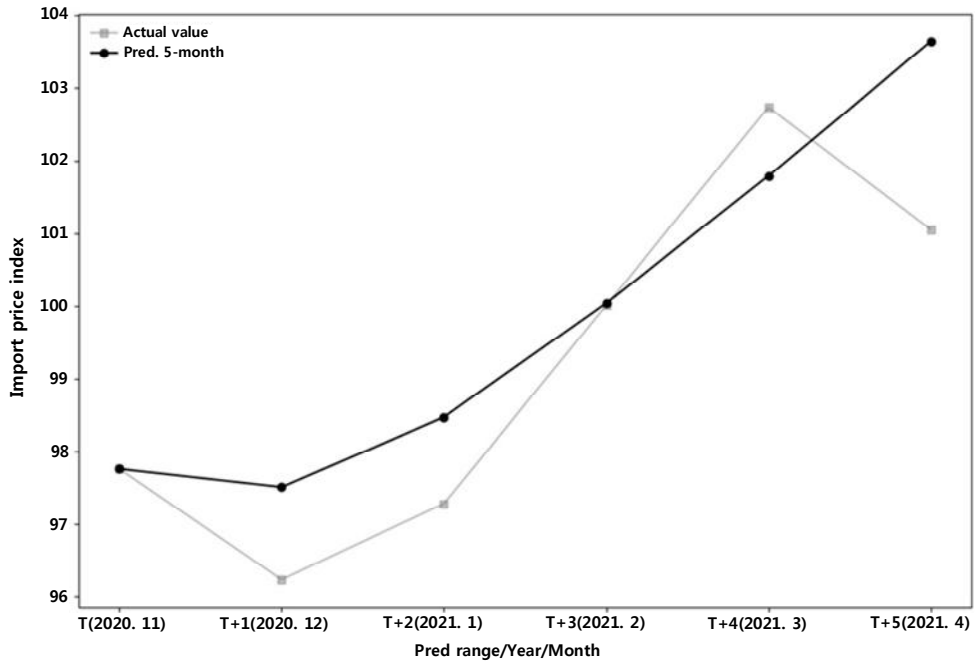
예 측 시 점 (월)	학 습 RMSE	테 스 트 RMSE	평 균 학 습 MAPE (%)	평 균 테 스 트 MAPE (%)	예 측 값	실 제 값	정 확 도 (%)
1 (2020. 12)	0.07287	0.17115	4.01	5.34	97.71	96.23	98.49
	0.07009	0.20421	3.87	6.56	97.55		98.65
	0.07159	0.18346	4.02	6.31	97.86		98.33
	0.07306	0.17135	4.35	5.55	97.55		98.65
	0.07559	0.16230	4.60	4.83	97.51		98.69
2 (2021. 01)	0.06526	0.25972	3.82	9.29	98.41	97.28	98.85
	0.06808	0.15188	4.19	5.07	98.47		98.79
	0.06878	0.16932	4.25	5.62	98.48		98.78
	0.06859	0.16676	4.25	5.58	98.44		98.82
	0.06830	0.16144	4.22	5.42	98.46		98.80
3 (2021. 02)	0.06154	0.22066	4.02	8.21	99.94	99.56	99.62
	0.06141	0.21321	3.99	8.01	99.96		99.60
	0.06236	0.20274	4.04	7.29	100.01		99.55
	0.06156	0.21072	4.02	7.82	99.97		99.59
	0.06195	0.23625	4.04	8.78	99.92		99.64
4 (2021. 03)	0.06257	0.23640	4.06	8.89	101.78	102.74	99.07
	0.06157	0.24151	4.04	8.98	101.78		99.07
	0.06149	0.23638	4.02	8.82	101.79		99.08
	0.06237	0.24318	4.01	9.03	101.80		99.99
	0.06200	0.24091	4.03	9.03	101.78		99.99
5 (2021. 04)	0.06158	0.22429	4.10	8.61	103.64	101.05	97.50
	0.06149	0.22714	4.07	8.80	103.60		97.54
	0.06225	0.23482	4.05	9.00	103.61		97.53
	0.06223	0.23027	4.02	8.74	103.64		97.50
	0.06111	0.22607	4.00	8.68	103.67		97.47

<표 7-4> 생산자물가지수 5개월 예측결과

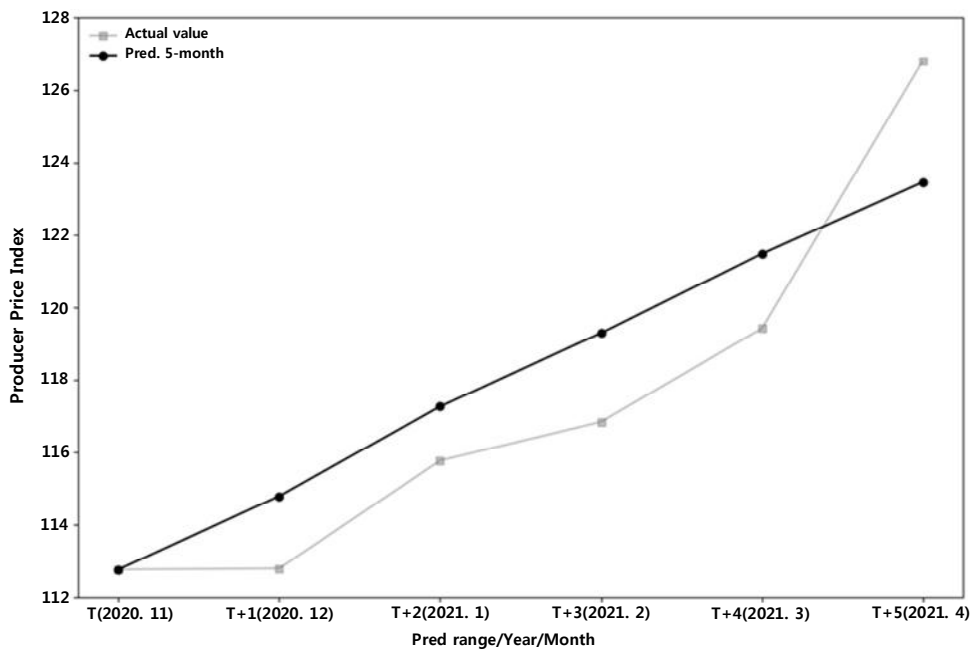
예측 시점 (월)	학습 RMSE	테스트 RMSE	평균 학습 MAPE (%)	평균 테스트 MAPE (%)	예측값	실제값	정확도 (%)
1 (2020. 12)	0.06948	0.15548	3.49	5.86	114.73	112.79	98.31
	0.07219	0.11820	3.85	4.73	114.40		98.59
	0.07142	0.11330	3.63	4.53	114.79		98.26
	0.07124	0.14736	3.66	5.53	114.72		98.32
	0.07972	0.27598	4.00	8.65	114.73		98.31
2 (2021. 01)	0.07008	0.13992	3.63	5.46	117.05	115.77	98.91
	0.08262	0.36881	4.97	13.72	116.70		99.20
	0.06966	0.12860	3.63	5.11	116.78		99.14
	0.07087	0.11531	3.50	4.64	117.03		98.92
	0.07202	0.10978	3.85	4.53	117.26		98.73
3 (2021. 02)	0.06787	0.11795	3.56	5.06	119.59	116.84	97.70
	0.06846	0.11460	3.57	4.93	119.32		97.92
	0.06879	0.11173	3.54	4.80	119.31		97.93
	0.06724	0.10433	3.47	4.61	119.29		97.95
	0.06785	0.11165	3.47	4.82	119.55		97.73
4 (2021. 03)	0.06917	0.12639	3.67	5.39	121.46	119.42	98.32
	0.06805	0.12564	3.599	5.41	121.31		98.44
	0.06924	0.09982	3.61	4.46	121.48		98.30
	0.06786	0.13036	3.57	5.62	121.27		98.47
	0.06981	0.10873	3.67	4.67	121.29		98.46
5 (2021. 04)	0.06927	0.10546	3.62	4.57	123.46	126.8	97.37
	0.08017	0.34821	5.05	14.35	123.64		97.51
	0.08072	0.34727	5.09	14.33	123.64		97.51
	0.06830	0.12104	3.62	5.29	123.64		97.51
	0.06857	0.10848	3.52	4.77	123.13		97.11

<표 7-5> 철근 가격 5개월 예측결과

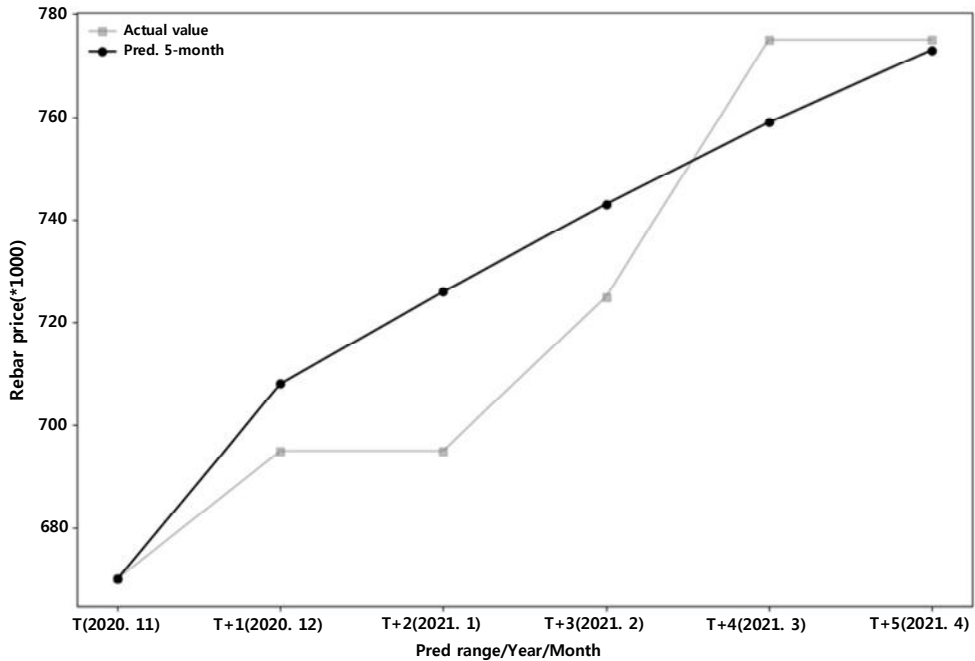
예측 시점 (월)	학습 RMSE	테스트 RMSE	평균 학습 MAPE (%)	평균 테스트 MAPE (%)	예측값 (원)	실제값 (원)	정확도 (%)
1 (2020. 12)	0.09498	0.35945	5.00	12.22	712,000	695,000	97.61
	0.09335	0.30678	5.05	10.59	708,000		98.16
	0.09425	0.32579	5.20	11.01	709,000		98.03
	0.09357	0.34254	5.11	11.51	709,000		98.03
	0.09035	0.31458	4.65	11.02	713,000		97.48
2 (2021. 01)	0.08927	0.24578	4.89	8.99	725,000	695,000	95.86
	0.08330	0.13130	4.51	6.09	726,000		95.73
	0.09053	0.24609	4.82	9.04	723,000		96.13
	0.09224	0.25552	5.15	9.28	724,000		95.99
	0.09021	0.25739	4.83	9.46	723,000		96.13
3 (2021. 02)	0.08203	0.14755	4.58	7.04	744,000	725,000	97.45
	0.08338	0.13781	4.60	6.46	743,000		97.58
	0.09259	0.24446	5.18	8.84	742,000		97.71
	0.08202	0.14805	4.73	7.62	744,000		97.45
	0.08982	0.26785	4.78	9.60	743,000		97.58
4 (2021. 03)	0.07958	0.14471	4.60	7.35	759,000	775,000	97.94
	0.08105	0.13286	4.66	7.07	759,000		97.94
	0.08036	0.14345	4.69	7.34	759,000		97.94
	0.07937	0.13796	4.53	7.28	760,000		98.06
	0.08010	0.13600	4.73	7.13	760,000		98.06
5 (2021. 04)	0.07858	0.16210	4.52	7.95	773,000	775,000	99.74
	0.08110	0.12833	4.70	6.92	773,000		99.74
	0.07963	0.14661	4.67	7.58	773,000		99.74
	0.08066	0.13777	4.84	7.63	772,000		99.61
	0.08028	0.14471	4.76	7.76	773,000		99.74



<그림 7-4> 수입물가지수 5개월 예측결과 시각화



<그림 7-5> 생산자물가지수 5개월 예측결과 시각화



<그림 7-6> 철근 가격 5개월 예측결과 시각화

결과의 주요 내용으로 생산자물가지수는 앞 장의 결과보다 약 1% 미만의 범위에서 성능이 개선되었다. 그리고 수입물가지수의 결과도 예측 기간 4개월까지 실제값의 경향을 잘 포착하는 것으로 확인된다. 하지만 수입물가지수, 생산자물가지수의 예측결과에서 5개월의 예측값부터 정확도가 감소하는 앞 장의 결과와 유사한 경향을 나타내었다. 이에 따라 철근 가격의 예측도 5개월까지만 진행하였으며, 철근 가격 5개월 장기예측의 평균 정확도는 약 97.83%를 나타내었다.

7.8 소결

본 장에서는 입력데이터가 소규모일 경우 데이터의 많은 부분을 학습으로 사용할 수 있고, 기존의 홀드 아웃 검증방식보다 명확한 모델성능 추정이 가능한 시계열 교차검증을 Recursive 예측방식에 적용하여 철근 가격을 장기예측을 진행하였다. 결과를 요약하면 다음과 같다.

검증방식의 변경으로 철근 가격 단기예측의 성능과 예측값의 정확도 확인을 위해 진행한 예측과정($t+1$)에서는 시계열 교차검증에 의해 분할된 5개 데이터셋 구간에서 측정된 학습과 테스트의 평균성능이 각각 월별로 학습은 최대 1.11%, 테스트는 약 2.02%의 차이를 보였다. 이 결과는 대부분 유사한 성능을 나타내며 측정되었고, 실제값에 대한 예측값의 정확도는 평균은 약 97.85%로 측정되었다.

측정된 단기예측의 성능을 기반으로 진행한 두 번째 장기예측결과는 2003년 1월부터 2020년 11월까지의 데이터 기간을 통해 2020년 12월부터 2021년 4월까지 5개월의 생산자물가지수를 예측하였을 때 실제값에 대한 예측값의 정확도가 약 1% 미만의 범위에서 홀드 아웃 검증방식을 사용한 결과보다 예측성능이 개선되었다. 그리고 측정된 테스트 구간 RMSE와 MAPE는 수치의 작은 차이에도 시차별로 측정된 예측값이 차이를 보이는 홀드 아웃 검증방식의 결과와 달리, 측정된 예측값이 대부분 균일한 결과를 나타냈다.

하지만 이러한 성능의 추가개선에도 생산자물가지수, 수입물가지수는 예측 시점 5개월부터 앞 절의 결과와 유사한 경향으로 오차가 커지기 시작하였다. 따라서 본 장에서 진행한 장기예측에서는 철근 가격예측의 유효범위를 5개월로 제한하였다. 철근 가격의 5개월 예측값의 정확도 평균은 약 97.83%를 나타냈다.

제8장 결론

본 연구는 건축자재 중 가격변동이 자주 발생하는 철근을 대상으로 최근 다양한 분야에서 활용되고 있는 딥러닝으로 미래의 가격변동을 예측하여 건축비용계획에 활용을 목적으로 수행하였다.

철근의 미래 가격예측을 위해 딥러닝을 활용하려면 우선으로 데이터의 확보가 필요하다. 하지만 철근 가격데이터의 경우 예측을 위해 현재 사용할 수 있는 데이터양은 월 단위로 매우 제한적이다. 그리고 딥러닝을 통해 철근 가격을 예측하기 위해서는 데이터 특성 선정과 데이터에 적합한 모델 선정, 하이퍼 파라미터 최적화 등의 복잡한 과정이 요구된다. 이에 따라 본 연구에서는 딥러닝 및 관련 기법을 활용하여 단기부터 장기의 순서로 단계적인 예측을 진행하였으며, 과정과 결과를 요약하면 다음과 같다.

1) 딥러닝을 활용하여 철근 가격을 단기 예측하기 위해 수입물가지수(철근), 생산자물가지수(철근), 두바이유, 브렌트유, 서부텍사스유, 고철 스크랩, 고철 스크랩 수입량, 철근 생산량, 건축착공현황(동수별 합계), 철근 가격(일반, SD300) 시계열 데이터를 데이터셋으로 구성한 후, 딥러닝 모델인 RNN, LSTM, Bi-LSTM에 학습시켜 예측성과 입력데이터의 범위를 벗어난 1달 이후의 가격을 예측하였다. 측정결과 Bi-LSTM이 나머지 두 모델보다 학습과 테스트 성능 모두 우수하였고, 1달 이후의 예측값은 실제값의 약 97.21%의 정확도로 예측되었다.

2) 딥러닝을 활용한 철근 가격 단기에측의 예측 정확도 향상을 위해 10가지의 데이터 특성을 조합하였고, 드롭아웃을 적용하였다. 그리고 Random search를 통해 하이퍼 파라미터의 최적 조합을 자동 탐색하였다. 이 방식의 적용으로 평균 예측 정확도는 98.32%로 앞서 10가지 데이터 특성을 모두 사용하는 방식보다 1.11% 성능이 개선되었다.

3) 철근 가격의 적정 구매 시기와 책정된 단가를 사전에 예측해보기 위해서는 딥러닝의 예측범위 확장이 필요하다. 이에 따라 데이터의 배치 재구성 시 입력과 정답데이터의 시점을 변경하는 방식으로 예측을 진행하였다. 하지만 1달 이후를 예측하는 단기예측방식에서 2달 이후를 예측하도록 변경하면 학습과 테스트 성능, 예측 정확도가 모두 감소한다. 특히 예측범위를 3달로 변경하게 되면 테스트 구간의 정확도가 극심한 차이를 보이며 예측이 제대로 이뤄지지 않음을 알 수 있었다. 따라서 딥러닝의 예측범위 확대를 위해서는 다른 방식의 적용이 필요할 것으로 판단하였다.

4) 데이터의 시간 간격을 재구성하는 방식을 통해 예측범위 확대를 진행하였다. 이 방식을 통해 예측범위를 4개월까지 확대해본 결과 2개월부터 4개월까지 약 99.09%, 99.40%, 95.30%의 정확도를 나타냈으며, 4개월부터 예측 정확도가 4.1% 감소하였다. 이는 데이터의 시간 단위가 재구성되면 늘어나는 시간 간격의 배수만큼 전체데이터양이 급격히 감소하는 것을 원인으로 판단하였다. 그리고 전반적인 건설 프로젝트의 기간과 비교할 때, 예측범위가 다소 부족하다고 판단하였다.

5) 예측범위를 추가 확대하기 위해 Recursive 장기예측방식으로 예측을 진행하였다. Recursive 예측방식은 구성된 데이터 특성을 각각 예측대상으로 변경하여 단기예측 후, 이를 원 데이터에 추가하는 방식으로 예측범위를 확장해 나간다. 따라서 예측대상이 다수일 경우 예측의 소요시간이 대상 수 만큼 증가하는 단점이 있다. 또한, 예측에 사용하는 데이터의 특성에 따라 대상의 단기예측성능이 저하될 수 있기에 피어슨 상관계수 분석을 통해 데이터 특성 상호 간의 상관관계수가 0.9 이상을 나타내는 철근 가격, 수입물가지수(철근), 생산자물가지수(철근)를 데이터셋으로 구성해 딥러닝의 입력데이터로 사용하였다.

결과로는 예측 기간 중 생산자물가지수가 5개월 예측값부터 정확도가 감소하기 시작하였고, 이후의 6개월부터 철근 가격, 수입물가지수의 예측 정확도도 감소하였다. 이는 5개월의 입력데이터가 많은 부분 오차가 누적된

예측값으로 대체 되었기 때문에 추정하였고, 이러한 문제점 해결을 위해 하이퍼 파라미터인 시퀀스 길이를 최근값을 예측하기 위해 사용한 데이터 크기보다 크게 설정하여 실제 데이터를 더 많이 포함 시키면 예측 정확도가 개선될 수 있을 것으로 추정하였다.

6) 철근 가격예측에 사용하는 학습 데이터양은 다른 빅데이터 기반의 기계학습과 비교해 절대적으로 부족하기 때문에 데이터의 많은 부분을 학습으로 사용하기 위해 Recursive 예측방식에 시계열 교차검증을 적용하였다. 결과는 생산자물가지수의 예측값이 약 1% 미만의 범위에서 성능이 개선되었고, 학습과 테스트 성능의 미미한 차이에도 예측값 별로 최대 약 4.2%의 차이를 보였던 Recursive와 홀드 아웃 검증을 사용한 예측결과와 달리 시계열 교차검증의 적용으로 측정된 예측값이 대부분 균일한 결과를 나타냈다. 하지만 이러한 성능의 추가개선에도 예측 시점 5개월부터 앞서 홀드 아웃 방식을 사용한 예측결과와 유사한 경향으로 오차가 커지기 시작하여 Recursive 방식과 시계열 교차검증을 통한 철근 가격예측의 유효한 범위를 5개월로 제한하였다. 철근 가격의 5개월 예측값의 평균 정확도는 약 97.83%를 나타냈다.

이상과 같이 본 연구에서 제안하는 딥러닝을 활용한 철근 가격예측은 과거 시계열 데이터를 기반으로 미래의 값을 예측하는 회귀 추정 방식으로 활용하였다. 그리고 예측 정확도 향상을 위해 입력데이터의 최적 데이터조합 탐색, 데이터에 적합한 딥러닝 모델 파악, 하이퍼 파라미터 최적화 등의 과정을 통해 단기예측 정확도를 개선하였다. 또한, 철근 가격 예측범위를 확대하기 위해 단기 예측값을 원 데이터에 추가하여 다음 시점을 예측하는 Recursive 방식으로 1 ~ 5개월까지 철근 가격을 장기예측할 수 있었다.

본 논문은 건설사업관리의 예상비용계획에 딥러닝의 시계열 예측을 활용하기 위한 관련 연구가 미비한 국내 현 상황에서 거시경제 영향으로 가격변동이 자주 발생하는 철근의 가격을 예측하기 위해 딥러닝의 활용방법을 제안하였다는 점에서 그 의의가 있다. 그리고 철근 이외의 건축재료를 비롯하여 시계열 데이터로 가격을 예측하고자 하는 관련

연구에서 본 논문이 제시한 방법이 예측과정과 정확도 향상 및 범위 확대를 위한 기초적 자료로 활용될 수 있을 것이다.

본 연구에서 제시하는 5개월의 철근 가격 예측범위는 공사 기간이 대체로 짧은 소규모 공사에만 적용 가능한 범위로 활용의 한계점이 있다. 이에 공사의 규모와 기간에 따라 더 많은 예측범위 확대가 필요할 것으로 판단된다. 이를 위해 본 연구에서 제시하는 시계열 데이터의 시간 단위를 조절하는 단기예측 방법과 Recursive 예측방식을 결합, 활용하는 것으로 예측범위의 추가확대가 가능할 것으로 사료된다. 그리고 새로운 딥러닝 관련 기법들이 현재에도 지속적으로 연구 및 개발되고 있다. 따라서 관련 기법들을 활용해 예측성과 범위를 확대하기 위한 연구가 추가적으로 필요하다.

참고문헌

- 강민제, 「딥러닝을 위한 경사하강법 비교」, 『한국산학기술학회논문지』, 제21권, 2호, pp. 189-194, 2020.
- 권성혁, 『LSTM 주가 예측의 후행성 문제 해결에 관한 연구』, 국민대학교 비즈니스 IT 전문 대학원, 석사학위논문, 2021.
- 김동근, 『디지털 이미지를 이용한 딥러닝 기반 흠의 물리적 특성 예측』, 서울대학교 대학원, 박사학위논문, 2020.
- 김백천, 정승환, 김민석, 김종근 & 이한수, 「기상상태 변동성을 고려한 LSTM 기반 태양광 발전량 예측」, 『한국지능시스템학회』, 제30권, 1호, pp. 7-12, 2020.
- 김준영, 이수형, 최영태 & 우상인, 「순환신경망을 활용한 콘크리트케도의 장기 침하 거동 예측」, 『한국지반공학학회논문집』, 제36권, 3호, pp. 5-14, 2020.
- 김진원, 이백래, 김주형 & 김재준, 「회귀분석을 이용한 교육시설의 공간 계획에 따른 공사비 예측 모델에 관한 연구」, 『대한건축학회 논문집 - 계획계』, 제27권, 10호, pp. 103-110, 2011.
- 마상완, 『인공신경망을 이용한 압축지수 예측』, 경북대학교 산업대학원, 석사학위논문, 2007.
- 박성우, 문지훈, 김현우 & 황인준, 「시계열 교차검증을 통한 설명 가능한 전력수요 예측 기법」, 『한국정보과학회 학술발표논문집』, pp. 311-313, 2019.
- 박우열 & 김광희, 「서포트 벡터 회귀분석을 이용한 공동주택 공사비 예측에 관한 연구」, 『대한건축학회 논문집 - 구조계』, 제23권, 4호, pp. 165-172, 2007.
- 박재형, 『초기 설계단계의 공사비 예측 및 관리에 대한 실증 분석』, 한양대학교 공학대학원, 석사학위논문, 2018.
- 배성완, & 유정석, 「머신 러닝 방법과 시계열 분석 모형을 이용한 부동산 가격지수 예측」, 『주택연구』, 제26권, 1호, pp. 107-133, 2018.
- 서양모 & 염재홍, 「기상 데이터를 활용한 LSTM 기반 미세먼지 농도

예측 방법 비교」, 『한국측량학회 학술대회자료집』, pp. 117-120, 2019.

손창우, 『인공신경망을 이용한 실트의 비배수 전단거동 예측』, 경북대학교 대학원, 석사학위논문, 2004.

신동하 & 김창복, 「하계 전력수요 예측을 위한 딥러닝 입력 패턴에 관한 연구」, 『한국정보기술학회논문지』, 제14권, 11호, pp. 127-134, 2016.

원승현, 『딥러닝 기법을 활용한 매립가스 발전소의 메탄가스 농도 및 전력생산량 예측』, 서울과학기술대학교, 박사학위논문, 2018.

윤종문, 「딥러닝 신경망을 이용한 신용카드 부도위험 예측의 효용성 분석」, 『한국금융연구원:금융연구』, 제33권, 1호, pp. 151-183, 2019.

이동훈 & 김관호, 「기후 및 계절 정보를 이용한 딥러닝 기반의 장기간 태양광 발전량 예측 기법」, 『한국전자거래학회지』, 제24권, 1호, pp. 1-16, 2019.

이발엽 & 김상욱, 「LSTM 인공신경망을 이용한 중국 주가 예측」, 『동북아경제연구』, 제32권, 2호, pp. 61-84, 2020.

이승현, 김선국, 이준복 & 한충희, 「플랜트 구매조달 및 자재관리 시스템 개발 요구사항 분석을 통한 개발 방향 제시」, 『한국건설관리학회 학술대회논문집』, pp. 204-209, 2006.

이용성 & 김경환, 「딥러닝의 반복적 예측방법을 활용한 철근 가격 장기 예측에 관한 실험적 연구」, 『한국건설관리학회 논문집』, 제22권, 3호, pp. 21-30, 2021.

이우창, 김양석, 김정민 & 이충권, 「머신러닝을 이용한 철광석 가격 예측에 대한 연구」, 『한국산업정보학회논문지』, 제25권, 2호, pp. 57-72, 2020.

이재명, 유정호, 김창덕, 이광재 & 임병수, 「건설 현장 자재수요 변동을 고려한 주문시점 산정 방법」, 『대한건축학회 논문집』, 제24권, 10호, pp. 117-125, 2008.

이종혁, 『LSTM과 양방향 순환신경망을 이용한 주가 예측모델 비교연구』, 서울과학기술대학교 일반대학원, 석사학위논문, 2019.

이철욱, 『인공신경 회로망을 이용한 암반터널 설계 전문가 시스템의 개발』, 한양대학교 대학원, 석사학위 논문, 1993.

이한승, 조재웅, 강호성 & 황정근, 「교차검증과 SVM을 이용한 도시침수 위험기준 추정 알고리즘 적용성 검토」, 『한국수자원학회논문집』, 제52권, 12호, pp. 963 - 973, 2019.

임창남, 『영상 분류를 위한 데이터 특징 커버리지 기반 심층학습 모델 검증 기법』, 아주대학교 대학원, 석사학위논문, 2021.

장필성, 『[EU] 2016 다보스포럼: 다가오는 4차 산업혁명에 대한 우리의 전략은?』, 과학기술정책, 26.2, pp. 12-15, 2016.

전병기, 김의중, 이경호, 공민석 & 신영기, 「딥러닝을 이용한 재실정보 기반 건물의 전기 수요 예측 모델」, 『설비공학논문집』, 제31권, 1호, pp. 22-31, 2019.

정동규, 「인공지능 기술과 주요 적용 산업 동향」, 『한국정보기술학회지』, 제15권, 2호, pp. 21-28, 2017.

정승현, 권오빈 & 손재호, 「딥러닝을 이용한 스마트 교육시설 공사비 분석 및 예측」, 『교육시설 논문지』, 제25권, 6호, pp. 35-44, 2018.

주일택 & 최승호, 「양방향 LSTM 순환신경망 기반 주가예측모델」, 『한국정보전자통신기술학회논문지』, 제11권, 2호, pp. 204-208, 2018.

지세현, 구영훈, 백의준, 신무곤, 윤성호, & 김명섭, 「비트코인 트랜잭션 수 예측을 위한 LSTM 학습데이터 선택기법」, 『In KNOM Conference』, 2019.

지세현, 박문서, 이현수 & 윤유상, 「건설공사 공사비 예측 및 관리기술 발전방향: 호주 사례를 중심으로」, 『한국건설관리학회논문집』, 제9권, 2호, pp. 170-181, 2008.

지세현, 백의준, 신무곤, 채병민, 문호원 & 김명섭, 「합성 곱 신경망 기반 웹 응용 트래픽 분류 모델 설계」, 『한국통신학회논문지』, 제44권, 6호, pp. 1113-1120, 2019.

최민수 & 권오현, 『건설 자재 원가 상승과 대응 방안』, 건설산업동향, pp. 2-34, 2008.

최열, 임하경 & 박봉운, 「건설자재 가격 변동에 따른 공동주택 분양가 변화 분석」, 『대한토목학회지』, 제29권, 6호, pp. 753-760, 2009.

최준영 & 이송근, 「시간별 LSTM 기법을 이용한 단기풍력발전 예측」,

- 『전기학회논문지』, 제69권, 6호, pp. 759-764, 2020.
- Tensorflow.org, 1-5-테스트와-검증, March 16, 2021, accessed <https://www.tensorflow.blog/헨즈온-머신러닝-1장-2장/1-5-테스트와-검증>.
- Tensorflow. org, 딥러닝이란 무엇인가?, March 16, 2021, accessed <https://tensorflow.blog/케라스-딥러닝/1-딥러닝이란-무엇인가/>.
- Bergstra, J., & Bengio, Y., 「Random search for hyper-parameter optimization」, 『The Journal of Machine Learning Research』, Vol.13(1), pp. 281-305, 2012.
- Cen, Z., & Wang, J., 「Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer」, 『Energy』, Vol.169, pp. 160-171, 2019.
- Chen, X., Wei, L., & Xu, J., 「House price prediction using lstm」, 『arXiv preprint arXiv:1709.08432』, 2017.
- Claesen, M., & De Moor, B., 「Hyperparameter search in machine learning」, arXiv preprint arXiv:1502.02127, 2015.
- Gao, X., Shi, M., Song, X., Zhang, C., & Zhang, H., 「Recurrent neural networks for real-time prediction of TBM operating parameters」, 『Automation in Construction』, Vol.98, pp. 225-235, 2019.
- Howard, R., & Matheson, J., 「Influence diagram retrospective」, 『Decision Analysis』, vol. 2.3, pp. 144-147, 2005.
- Hira, N. A., & Walter, J. C., 『Estimating: from Concept to Completion』, Prentice Hall Inc, 1988.
- Hochreiter, S., & Schmidhuber, J., 「Long short-term memory」, 『Neural computation』, Vol.9(8), pp. 1735-1780, 1997.
- Howard, R., & Matheson, J., 「Influence diagram retrospective」, 『Decision Analysis』, vol. 2.3, pp. 144-147, 2005.
- Jang. Y., Jeong. I. & Cho. Y., 「Business failure prediction of construction contractors using a LSTM RNN with accounting, construction market, and macroeconomic variables」, 『Journal of Management in Engineering』, Vol.36(2), 04019039, 2020.

Kohavi, R., 『A study of cross-validation and bootstrap for accuracy estimation and model selection』, Ijcai, Vol.14(2), 1995.

Lahari, M., Ravi, D.H., & Bharathi, R., 「Fuel Price Prediction Using RNN」, 『In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)』, IEEE, pp. 1510–1514, 2018.

Liao, T. W., 「Clustering of time series data—a survey」, 『Pattern recognition』, Vol.38(11), pp. 1857–1874, 2005.

Mou. L., Zhao. P. & Chen. Y., 「Short-term traffic flow prediction: A long short-term memory model enhanced by temporal information」, 『In CICTP』, pp. 2411–2422, 2019.

Olson, D. L., & Delen, D., 『Advanced data mining techniques』, Springer-Verlag, Berlin Heidelberg, 2008.

Pawar, K., Jalem, R. S., & Tiwari, V., 「Stock market price prediction using LSTM RNN」, 『In Emerging Trends in Expert Applications and Security』, Springer, Singapore, pp. 493–503, 2019.

Rumelhart, D., Hinton, G. & Williams, R., 「Learning internal representations by error propagation」, (No. ICS-8506), California Univ San Diego La Jolla Inst for Cognitive Science, (No. ICS-8506), pp. 318–362, 1985.

Samuel, A. L., 「Some studies in machine learning using the game of checkers」, 『IBM Journal of research and development』, Vol.3(3), pp. 210–229, 1959.

Schuster, M., & Paliwal, K. K., 「Bidirectional recurrent neural networks」, 『IEEE Trans Signal Process』, pp. 2673–2681, Vol.45, 1997.

Scikit-learn.org, sklearn.model_selection.TimeSeriesSplit, accessed Sep 27, 2021, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html

Srivastava. N., Hinton. G., Krizhevsky. A., Sutskever. I., & Salakhutdinov. R., 「Dropout: a simple way to prevent neural networks from overf

itting」, 『The journal of machine learning research』, Vol.15.1, pp. 1929–1958, 2014.

Taieb, S., Sorjamaa, A., & Bontempi, G., 「Multiple-output modeling for multi-step-ahead time series forecasting」, 『Neurocomputing』, Vol. 73(10–12), pp. 1950–1957, 2010.

Tensorflow.org, Overfitting and underfitting, accessed March 16, 2021, https://www.tensorflow.org/tutorials/keras/overfit_and_underfit.stand.

Tensorflow.org, tf.keras.callbacks.EarlyStopping, accessed Sep 27, 2021, https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P., 「Gradient-based learning applied to document recognition」, 『Proceedings of the IEEE』, Vol.86(11), pp. 2278–2324, 1988.

Willmott, C. J., & Matsuura, K., 「Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance」, 『Climate research』, Vol.30.1, pp. 79–82, 2005.

Yadav, S., & Shukla, S., 「Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification」, 2016 IEEE 6th International conference on advanced computing (IACC), IEEE, 2016.

Zhang. Z., Wang. Y., Chen. P., Yu. G., 「Applicetion of long short-term memory neural network for multi-step travel time forecasting on urban expressways」, 『In CICTP 2017: Transportetion Reform and Change—Equity, Inclusiveness, Sharing, and Innovetion, American Society of Civil Engineering』, pp. 444–454, 2018.

부록

[부록 1] 순환신경망 계열 모델 구성방법(Tensorflow 1.15 Ver)

#RNN

```
cell = BasicRNNCell(num_units=hidden_dim, activation=tf.tanh)
```

```
outputs, _states = dynamic_rnn(cell, X, dtype=tf.float32)
```

#activation = 활성화 함수선택

#LSTM

```
cell = BasicLSTMCell(num_units=hidden_dim, state_is_tuple=True,  
activation=tf.tanh)
```

```
outputs, _states = dynamic_rnn(cell, X, dtype=tf.float32)
```

#Bidirectional LSTM

```
cell_fw = LSTMCell(num_units=hidden_dim, forget_bias=1.0,  
state_is_tuple=True, activation=tf.tanh) #순방향
```

```
cell_bw = LSTMCell(num_units=hidden_dim, forget_bias=1.0,  
state_is_tuple=True, activation=tf.tanh) #역방향
```

```
outputs_fw, outputs_bw, _states = bidirectional_dynamic_rnn(cell_fw(),  
cell_bw(), X, dtype=tf.float32) #순방향과 역방향 결합
```

```
outputs = tf.concat([outputs_fw, outputs_bw], axis=1)
```

#Stack Bidirectional LSTM

```
cell_bw = LSTMCell(num_units=hidden_dim, forget_bias= 1.0, state_is_t  
uple=True, activation=tf.tanh) #순방향
```

```
cell_fw = LSTMCell(num_units=hidden_dim, forget_bias= 1.0, state_is_t  
uple=True, activation=tf.tanh) #역방향
```

```
cells_fw = [lstm_cell_fw() for _ in range(number_layer)] #순방향 적층
```

```
cells_bw = [lstm_cell_bw() for _ in range(number_layer)] #역방향 적층
```

#number_layer의 수만큼 신경망의 셀의 층을 늘림

```
outputs, _, _ = stack_bidirectional_dynamic_rnn(cells_fw=cells_fw, cells_  
bw=cells_bw, inputs=X, dtype=tf.float32)
```

[부록 2] 검증방식에 따른 데이터 분할방법

#홀드 아웃

```
from sklearn.model_selection import train_test_split
train_set, test_set = train_test_split(data, test_size = 분할비율, shuffle=
True 입력 시 데이터를 무작위로 섞음. 시계열 데이터는 시간순서 유지
때문에 False 사용.)
```

#시계열 교차검증

```
from sklearn.model_selection import TimeSeriesSplit
dataX = data #전체 데이터
dataY = data[:, -1] #전체 데이터 열의 마지막 칼럼(예측대상 데이터)
dataY = dataY.reshape(-1,1)
ts = TimeSeriesSplit(max_train_size = 학습데이터의 최대크기, test_size
= 학습데이터의 최대크기, n_splits = 전체 데이터셋 분할 수)
```

```
for train_index, test_index in ts.split(dataX):
    print("TRAIN:", train_index, "TEST:", test_index)
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
```

#n_splits=5를 사용할 경우 예제 출력

```
TRAIN: [0] TEST: [1]
TRAIN: [0 1] TEST: [2]
TRAIN: [0 1 2] TEST: [3]
TRAIN: [0 1 2 3] TEST: [4]
TRAIN: [0 1 2 3 4] TEST: [5]
```

[부록 3] 정규화(Minmaxscaler)

#정규화된 데이터와 예측값을 원래의 스케일로 변환하기 위해 fit을 X, Y로 분리.

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler1 = MinMaxScaler()
```

```
scaler2 = MinMaxScaler()
```

```
scaler1.fit(X_train) #학습 X 데이터의 각 열 최소 최대값을 fit
```

#X 스케일 변환 적용

```
X_train = scaler1.transform(X_train)
```

```
X_test = scaler1.transform(X_test)
```

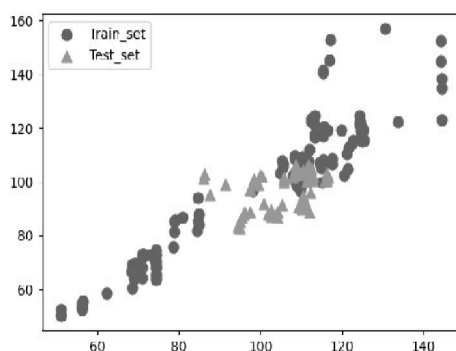
```
scaler2.fit(Y_train) #학습 Y 라벨값 최소, 최대 fit.
```

#Y 스케일 변환 적용

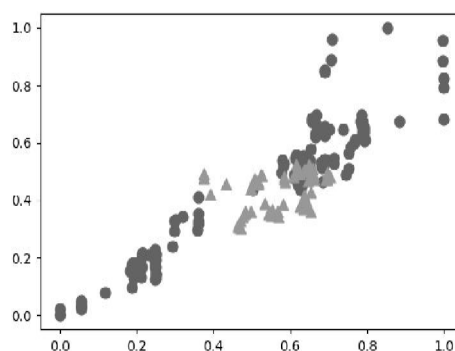
```
Y_train = scaler2.transform(Y_train)
```

```
Y_test = scaler2.transform(Y_test)
```

#학습으로 사용되는 데이터의 최소, 최대값을 기준으로 스케일 변환



(a) Data before scaler applies



(b) Data after scaler applies

#테스트 구간의 예측값을 본래 스케일로 변환.

```
testY_in_iteration_inverse = scaler2.inverse_transform(Y_test)
```

[부록 4] Random search를 이용한 하이퍼 파라미터 탐색모델 구성방법

#모델 정의

```
from sklearn.model_selection import RandomizedSearchCV
```

```
def lee_lstm(learning_rate, seq_length, hidden, dropout_rate):
```

```
    ...
```

```
    Return
```

```
#하이퍼 파라미터 탐색범위 설정
```

```
hyper_param = {'learning_rate': (0.01, 0.1),  
               'seq_length': (2, 12),  
               'hidden_dim': (10, 150),  
               'dropout_rate': (0.1, 1.0)}
```

```
#정의된 모델과 모델에 사용할 하이퍼 파라미터 범위를 Random search로 전달.
```

```
LEE_TEST = RandomizedSearchCV(lee_lstm, param_distributions=hyper_param,  
                               n_iter=탐색수, cv = 분할 검증 수, scoring = 오차평가방법)
```

딥러닝 및 관련 기법을 활용한 철근 가격

장단기 예측

최근 원자재 가격상승에 의한 철강재, 목 자재, 단열재 등과 같은 건축 자재 가격이 급등하고 있다. 특히 거시경제의 영향으로 가격이 자주 변동하는 철근의 경우 2008년의 ‘철근 파동’이 발생한 당시 기록했던 가격 최고치를 최근 경신하였다. 그리고 이로 인한 공사 지연 및 중단과 같은 과거에 발생하였던 문제가 현재 되풀이되고 있다. 특히 철근 가격의 상승은 철강사와 직접 공급계약을 맺고, 자재 조달 경로가 다양한 대형건설사에 비해 가격협상력이 떨어져 시장에서 자재를 수급하는 중소 건설사에 치명적일 수 있다. 이러한 문제의 원인으로 한국철강협회에서 제공하는 자료에 따르면, 코로나바이러스(COVID-19)에 의한 사회적 거리 두기로 인한 철근 수요감소로 철근 생산량이 급감하였고, 최근 건설경기가 예상보다 빠른 회복세로 전환되며 수요가 증가하면서 공급 부족 현상의 발생으로 철근 가격이 예상보다 큰 폭으로 폭등한 것으로 판단된다. 하지만 이러한 문제의 원인에 대한 분석은 상황이 발생한 이후에 이뤄지는 것이 대부분이며 예상치 못한 변수로 인한 돌발적 상황이 지속적으로 발생할 수 있다. 그리고 건설산업은 프로젝트의 수행 기간이 길어 여러 사회적, 경제적인 위험에 대한 노출이 타 산업과 비교하여 상대적으로 크기 때문에 이러한 위험에 직접적으로 노출되어 있다고 할 수 있다.

이에 본 연구에서는 건설사업관리의 효율적인 비용계획을 위해 최근 일부의 영역에서 인간의 판단, 속도, 정확도보다 나은 결과를 보여주는 딥러닝을 활용하고자 연구를 진행하였다. 하지만 딥러닝으로 철근 가격을 예측하기 위해 사용할 수 있는 데이터양은 다른 빅데이터 기반의 기계학습과 비교하였을 때 절대적으로 부족하다. 부족한 데이터에 추가로 철근 가격은 자연적, 물리적 특성이 아닌 사회적, 경제적 특성들이 반영되기 때문

에 예측의 어려움은 가중된다. 이에 철근 가격예측에 사용할 변수를 선정하고, 사용데이터에 적합한 딥러닝 모델탐색과 예측 정확도를 향상하기 위한 단기예측에 관한 연구가 진행되었다. 그리고 건설 프로젝트의 전반적인 기간을 고려할 때, 딥러닝을 통한 장기적 시점에 대한 예측이 필요하기 때문에 예측범위를 확대하기 위한 여러 방법적 시도를 단계적으로 진행하였다.

예측범위 확대가 가능한 방법으로는 딥러닝의 단기예측범위가 입력데이터의 시간 간격과 같이 달라지기 때문에 시계열 데이터의 시간 단위를 재구성하는 방식을 통해 예측범위를 확장할 수 있었다. 하지만 이 방식은 예측범위가 늘어나는 만큼의 배수로 입력데이터가 소실되기 때문에 예측범위확장이 제한적이며, 3개월의 예측만 가능하였다. 예측범위를 추가로 확대하기 위한 다른 방법으로는 데이터셋으로 구성된 데이터 특성 전부를 예측대상으로 설정하여 단기예측하고, 생성된 예측값을 원 데이터에 추가하여 다음 시점을 예측하는 Recursive 예측방식을 통해 예측범위를 확장하기 위한 실험을 진행하였다. 그리고 이 방식은 데이터 특성이 많을수록 예측해야 할 대상도 늘어나기 때문에, 피어슨 분석을 통해 데이터 특성 상호 간 높은 상관관계를 나타내는 생산자물가지수(철근), 수입물가지수(철근), 철근 가격을 데이터셋으로 구성하였다. 예측범위는 생산자물가지수의 5개월 예측 시점부터 오차가 증가하기 시작하였고, 이로 인해 6개월의 예측 시점부터 수입물가지수, 철근 가격의 예측값 오차도 같이 증가하기 시작하여, 철근 가격예측이 가능한 유효한 범위는 5개월로 판단하였다.

이상과 같이 본 연구는 딥러닝을 활용하여 국내의 철근 가격을 장단기 예측 방법에 대한 과정과 결과를 제시한다. 향후 본 연구의 결과가 건설사업관리의 비용계획단계에서 효율적인 사업계획 및 수행을 위한 활용과 관련 분야 후속 연구의 기초자료로 활용되기를 기대한다.