



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](#)

碩士學位論文

Attention Mechanism 기반
Bi-Directional LSTM을 이용한
주가 예측 모델

**Attention Mechanism Based Using
Bi-Directional LSTM Stock Price
Forecasting Model**

한밭대학교 情報通信專門 大學院

컴퓨터工學科

金 玗 材

2019년 8월

Attention Mechanism 기반
Bi-Directional LSTM을 이용한
주가 예측 모델

**Attention Mechanism Based Using Bi-Directional
LSTM Stock Price Forecasting Model**

指導教授 김 윤 중

이 論文을 工學碩士學位
請求論文으로 제출함

2019년 5월

한밭大學校 情報通信專門 大學院

컴퓨터工學科

金 琬 材

목 차

표 목 차	i
그 림 목 차	ii
국 문 요 약	vi
I. 서 론	1
1. 연구의 배경 및 목적	1
2. 주가 분석 방법	2
II. 이 론	4
1. 주가 예측 관련 국내 선행 연구	4
2. 순환 신경망(Recurrent Neural Network)	6
3. LSTM(Long-Short term Memory) 네트워크	10
(1) cell state	12
(2) forget gate	13
(3) input gate	14
(4) output gate	15
4. Bi-directional Recurrent Neural Network	16
5. Attention Mechanism 기반 시퀀스-투-시퀀스	18
6. Bollinger Band	21
III. Attention Mechanism 기반 Bi-LSTM을 이용한 주가 예측 모델	23
1. Attention Bi-LSTM-Only Model	25
2. Attention Bi-LSTM-Concat Model	26
3. Attention Bi-LSTM-Dot Model	27
IV. 실험 및 성능 평가	28
1. 실험 방법	28

2. 실험 데이터	28
3. 실험 결과	32
(1) LG U+ 종목에 적합한 학습 파라미터	33
1) 적절한 Sequence Length / Hidden Layer 분석	33
2) 적절한 Attention Size 분석	37
3) U+ 종목의 Att-Bi-LSTM-Only 모델 실험 결과	40
4) U+ 종목의 Att-Bi-LSTM-Concat 모델 실험 결과	41
5) U+ 종목의 Att-Bi-LSTM-Dot 모델 실험 결과	42
6) U+ 종목의 Bi-LSTM 대비 제안 모델별 RMSE 비교	44
(2) KT 종목에 적합한 학습 파라미터	45
1) 적절한 Sequence Length/Hidden Layer 분석	45
2) 적절한 Attention Size 분석	46
3) KT 종목의 Att-Bi-LSTM-Only 모델 실험 결과	48
4) KT 종목의 Att-Bi-LSTM-Concat 모델의 실험 결과	49
5) KT 종목의 Att-Bi-LSTM-Dot 모델 실험 결과	50
(3) SK텔레콤 종목에 적합한 학습 파라미터	52
1) 적절한 Sequence Length / Hidden Layer 분석	52
2) 적절한 Attention Size 분석	53
3) SK텔레콤 종목의 Att-Bi-LSTM-Only 모델 실험 결과	55
4) SK텔레콤 종목의 Att-Bi-LSTM-Concat 모델 실험 결과	56
5) SK텔레콤 종목의 Att-Bi-LSTM-Dot 모델 실험 결과	57
(4) 실험 결과 분석	59
 IV. 결 론	 61
 참 고 문 헌	 63
 ABSTRACT	 65

표 목 차

표 2-1 볼린저 밴드 지표 계산식	22
표 4-1 입력 데이터 구성	29
표 4-2 입력 데이터 항목 계산식(1)	30
표 4-3 입력 데이터 항목 계산식(2)	31
표 4-4 종목별 학습 파라미터	31
표 4-5 U+ Bi-LSTM을 이용한 시퀀스 길이와 hidden layer 크기에 따른 RMSE 비교	36
표 4-6 U+의 입력 23, seq 28, hidden 16, attention_size변화에 따른 RMSE비교	37
표 4-7 U+ Bi-LSTM과 Attention-Bi-LSTM 모델별 평균 RMSE 비교	39
표 4-8 KT Bi-LSTM을 이용한 시퀀스 길이와 hidden layer 크기에 따른 RMSE 비교	45
표 4-9 KT의 입력 23, seq 28, hidden 10, attention_size변화에 따른 RMSE비교	46
표 4-10 KT Bi-LSTM과 Attention-Bi-LSTM 모델별 평균 RMSE 비교	47
표 4-11 SK텔레콤 Bi-LSTM을 이용한 시퀀스 길이와 hidden layer 크기에 따른 RMSE 비교	52
표 4-12 SK텔레콤의 입력 23, seq 28, hidden 10, attention_size변화에 따른 RMSE 비교	53
표 4-13 SK텔레콤 Bi-LSTM과 Attention-Bi-LSTM 모델별 평균 RMSE 비교 ..	54
표 4-14 종목별 Bi-LSTM 대비 Attention-Bi-LSTM 모델 평균 RMSE	59
표 4-15 종목별 Bi-LSTM 대비 Attention-Bi-LSTM 모델 평균 감소량	59
표 4-16 종목별 Bi-LSTM 대비 Attention-Bi-LSTM 모델 평균 오차 향상률	60

그 립 목 차

그림 2-1 인공 신경망과 순환 경로를 포함하는 순환신경망	6
그림 2-2 RNN 계층의 순환 구조 펼쳐진 모습(출처:[13])	7
그림 2-3 RNN 계층의 계산 그래프	7
그림 2-4 Recurrent Neural Networks 설계 유형(출처:[14])	8
그림 2-5 기본적인 RNN 계층과 LSTM 계층 비교(출처:[13])	11
그림 2-6 기본적인 RNN 계층과 LSTM 계층 내부(출처:[13])	11
그림 2-7 cell state	12
그림 2-8 무엇을 잊을 것인가(forget gate)	13
그림 2-9 무엇을 저장할 것인가(input gate)	14
그림 2-10 무엇을 출력할 것인가(output gate)	15
그림 2-11 LSTM의 반복 모듈 내부구조	16
그림 2-12 양방향 순환신경망(Bidirectional Recurrent Neural Network)	16
그림 2-13 Bi-RNN 전개도	17
그림 2-14 기본적인 시퀀스투시퀀스 모델	19
그림 2-15 Attention을 기반으로 하는 시퀀스투시퀀스 모델	21
그림 2-16 LG유플러스의 현재가와 볼린저 밴드 그래프	22
그림 3-1 Attention Layer 구조(TensorFlow 제공)	23
그림 3-2 Attention Bi-LSTM 모델 기본 설계	24
그림 3-3 Attention context vector로만 학습한 모델	25
그림 3-4 Attention context vector와 hidden layer의 마지막 시퀀스를 연결하여 학습한 모델	26
그림 3-5 Attention context vector와 hidden layer의 마지막 시퀀스의 요소간 곱을 하여 학습한 모델	27
그림 4-1 U+(032640.KS) 종목의 다운로드 데이터	29
그림 4-2 U+(032640.KS) 종목의 볼린저 밴드 지표 추가한 항목	29
그림 4-3 U+(032640.KS) 종목의 각 항목과의 비율 계산 추가한 항목	29
그림 4-4 이동통신 3사의 5년간 주가와 거래량	32

그림 4-5 U+ seq5, hidden5 rmse비교	33
그림 4-6 U+ seq5, hidden10 rmse비교	33
그림 4-7 U+ seq5, hidden16 rmse비교	33
그림 4-8 U+ seq5, hidden32 rmse비교	33
그림 4-9 U+ seq5, hidden64 rmse비교	33
그림 4-10 U+ seq10, hidden5 rmse비교	33
그림 4-11 U+ seq10, hidden10 rmse비교	33
그림 4-12 U+ seq10, hidden16 rmse비교	33
그림 4-13 U+ seq10, hidden32 rmse비교	34
그림 4-14 U+ seq10, hidden64 rmse비교	34
그림 4-15 U+ seq20, hidden5 rmse비교	34
그림 4-16 U+ seq20, hidden10 rmse비교	34
그림 4-17 U+ seq20, hidden16 rmse비교	34
그림 4-18 U+ seq20, hidden32 rmse비교	34
그림 4-19 U+ seq20, hidden64 rmse비교	34
그림 4-20 U+ seq28, hidden5 rmse비교	34
그림 4-21 U+ seq28, hidden10 rmse비교	35
그림 4-22 U+ seq28, hidden16 rmse비교	35
그림 4-23 U+ seq28, hidden32 rmse비교	35
그림 4-24 U+ seq28, hidden64 rmse비교	35
그림 4-25 U+ Att_Bi-LSTM seq 28, hidden 16, attention 5	37
그림 4-26 U+ Att_Bi-LSTM seq 28, hidden 16, attention 10	37
그림 4-27 U+ Att_Bi-LSTM-only seq 28, hidden 16, attention 16	38
그림 4-28 U+ Att_Bi-LSTM-only seq 28, hidden 16, attention 20	38
그림 4-29 U+ Att_Bi-LSTM seq 28, hidden 16, attention 32	38
그림 4-30 U+ Att_Bi-LSTM seq 28, hidden 16, attention 40	38
그림 4-31 U+ Att_Bi-LSTM seq 28, hidden 16, attention 64	38

그림 4-32 U+ Att_Bi-LSTM seq 28, hidden 16, attention 128	38
그림 4-33 U+ Att-Bi-LSTM-Only 모델 1회 RMSE, 실제값과 예측값	40
그림 4-34 U+ Att-Bi-LSTM-Only 모델 2회 RMSE, 실제값과 예측값	40
그림 4-35 U+ Att-Bi-LSTM-Only 모델 3회 RMSE, 실제값과 예측값	40
그림 4-36 U+ Att-Bi-LSTM-Only 모델 4회 RMSE, 실제값과 예측값	40
그림 4-37 U+ Att-Bi-LSTM-Only 모델 5회 RMSE, 실제값과 예측값	41
그림 4-38 U+ Att-Bi-LSTM-Concat 모델 1회 RMSE, 실제값과 예측값	41
그림 4-39 U+ Att-Bi-LSTM-Concat 모델 2회 RMSE, 실제값과 예측값	41
그림 4-40 U+ Att-Bi-LSTM-Concat 모델 3회 RMSE, 실제값과 예측값	41
그림 4-41 U+ Att-Bi-LSTM-Concat 모델 4회 RMSE, 실제값과 예측값	42
그림 4-42 U+ Att-Bi-LSTM-Concat 모델 5회 RMSE, 실제값과 예측값	42
그림 4-43 U+ Att-Bi-LSTM-Dot 모델 1회 RMSE, 실제값과 예측값	42
그림 4-44 U+ Att-Bi-LSTM-Dot 모델 2회 RMSE, 실제값과 예측값	42
그림 4-45 U+ Att-Bi-LSTM-Dot 모델 3회 RMSE, 실제값과 예측값	43
그림 4-46 U+ Att-Bi-LSTM-Dot 모델 4회 RMSE, 실제값과 예측값	43
그림 4-47 U+ Att-Bi-LSTM-Dot 모델 5회 RMSE, 실제값과 예측값	43
그림 4-48 U+ 종목의 비교 모델 대비 제안 모델별 RMSE 비교그래프	44
그림 4-49 KT Att-Bi-LSTM-Only 1회 RMSE, 실제값와 예측값	48
그림 4-50 KT Att-Bi-LSTM-Only 2회 RMSE, 실제값와 예측값	48
그림 4-51 KT Att-Bi-LSTM-Only 3회 RMSE, 실제값와 예측값	48
그림 4-52 KT Att-Bi-LSTM-Only 4회 RMSE, 실제값와 예측값	49
그림 4-53 KT Att-Bi-LSTM-Only 5회 RMSE, 실제값와 예측값	49
그림 4-54 KT Att-Bi-LSTM-Concat 1회 RMSE, 실제값와 예측값	49
그림 4-55 KT Att-Bi-LSTM-Concat 2회 RMSE, 실제값와 예측값	49
그림 4-56 KT Att-Bi-LSTM-Concat 3회 RMSE, 실제값와 예측값	50
그림 4-57 KT Att-Bi-LSTM-Concat 4회 RMSE, 실제값와 예측값	50
그림 4-58 KT Att-Bi-LSTM-Concat 5회 RMSE, 실제값와 예측값	50

그림 4-59	KT Att-Bi-LSTM-Dot 1회 RMSE, 실제값과 예측값	50
그림 4-60	KT Att-Bi-LSTM-Dot 2회 RMSE, 실제값과 예측값	51
그림 4-61	KT Att-Bi-LSTM-Dot 3회 RMSE, 실제값과 예측값	51
그림 4-62	KT Att-Bi-LSTM-Dot 4회 RMSE, 실제값과 예측값	51
그림 4-63	KT Att-Bi-LSTM-Dot 5회 RMSE, 실제값과 예측값	51
그림 4-64	SK Att-Bi-LSTM-Only 1회 RMSE, 실제값과 예측값	55
그림 4-65	SK Att-Bi-LSTM-Only 2회 RMSE, 실제값과 예측값	55
그림 4-66	SK Att-Bi-LSTM-Only 3회 RMSE, 실제값과 예측값	55
그림 4-67	SK Att-Bi-LSTM-Only 4회 RMSE, 실제값과 예측값	56
그림 4-68	SK Att-Bi-LSTM-Only 5회 RMSE, 실제값과 예측값	56
그림 4-69	SK Att-Bi-LSTM-Concat 1회 RMSE, 실제값과 예측값	56
그림 4-70	SK Att-Bi-LSTM-Concat 2회 RMSE, 실제값과 예측값	56
그림 4-71	SK Att-Bi-LSTM-Concat 3회 RMSE, 실제값과 예측값	57
그림 4-72	SK Att-Bi-LSTM-Concat 4회 RMSE, 실제값과 예측값	57
그림 4-73	SK Att-Bi-LSTM-Concat 5회 RMSE, 실제값과 예측값	57
그림 4-74	SK Att-Bi-LSTM-Dot 1회 RMSE, 실제값과 예측값	57
그림 4-75	SK Att-Bi-LSTM-Dot 2회 RMSE, 실제값과 예측값	58
그림 4-76	SK Att-Bi-LSTM-Dot 3회 RMSE, 실제값과 예측값	58
그림 4-77	SK Att-Bi-LSTM-Dot 4회 RMSE, 실제값과 예측값	58
그림 4-78	SK Att-Bi-LSTM-Dot 5회 RMSE, 실제값과 예측값	58
그림 4-79	종목별 Bi-LSTM대비 Attention 기반 Bi-LSTM 모델별 RMSE 비교	59

국 문 요 약

Attention Mechanism 기반 Bi-Directional LSTM을 이용한 주가 예측 모델

논문 제출자 김윤재
지도교수 김윤중

2016년 알파고가 이세돌과의 바둑대결에서 이기고 난 후 많은 사람들은 머신러닝의 위력을 더욱 인정하기 시작했고, 금융 투자 분야에서도 사람이 직접 하던 투자를 머신러닝기술을 적용하기 시작하였다. 그리고 경제학, 통계학, 컴퓨터 공학 등 관련 분야에서도 최신 지도 학습 딥러닝 기술에 의한 주가 예측이나 강화학습에 의한 주가 예측 연구들이 꾸준히 진행되고 있다.

하지만 미래의 주가 예측은 기업의 매출변동, 영업이익, 순이익 변동, 환율 변동, 원재료가격의 변화, 정부의 정책, 신기술 개발, 특허 등 다양한 주가 변동을 유발시키는 요인들로 인해 주가가 예상하지 못했던 방향으로 변동되어 버리는 경우가 있어서 여전히 어려움을 겪고 있다.

주가 데이터는 모두 숫자로 이루어져 있고, 숫자를 잘 계산하기 위한 수학적 지식들도 잘 갖추어져 있기 때문에 딥러닝(Deep Learning) 기술은 주가 예측에 적용하기 매우 용이하다. 다른 형태의 기계 학습과 달리 입력 데이터의 특징 패턴을 그대로 학습하기 때문에, 학습 이전에 입력 데이터의

특징 패턴에 따라 예측 결과가 큰 차이를 보이기 때문에 주가예측에 도움이 되는 다양한 지표들을 이용하여 시도하고 있다. 금융 투자를 할 때에도 주가 예측에 도움이 되는 데이터들을 수집한 후 딥러닝 기술로 예측 모델을 설계하고 그 예측 결과에 따라 투자를 진행하는 것이기 때문에 무엇보다도 정확도가 가장 우선시 되어야 한다.

따라서 본 연구에서는 주가예측의 정확도를 향상시키기 위해 무선 이동통신 3사의 2014년부터 2018년까지 5년간 주식 가격 데이터를 yahoo finance에서 수집하고, 주가의 기본 데이터인 시가, 고가, 저가, 종가, 수정종가, 거래량에 추가하여 주가예측에 도움을 줄 수 있도록 종가의 5일, 20일 이동평균, 거래량의 5일 이동평균, 20일 이동평균, 이동평균을 이용하여 볼린저 밴드 지표를 추가하고, 당일 종가 대비 당일 각 항목의 비율, 당일 종가 대비 전일 종가 비율, 당일 거래량 대비 전일 거래량 비율, 종가 이동평균 대비 당일 종가 비율, 거래량 이동평균 대비 당일 거래량 비율 등 17개 항목을 추가한 23개의 항목으로 정규화시켜서 TensorFlow를 이용하여 Attention Mechanism을 기반 양방향 LSTM 순환신경망 모델을 제안한다.

제안한 Attention Mechanism 기반 양방향 LSTM 순환신경망 모델의 세 가지 방법을 제안했는데 이 모델의 특징은 다음과 같다. 과거의 주식 가격의 변동 패턴을 학습시켜 미래의 주식 가격 예측에 적합한 순환신경망의 내부에 LSTM 메모리 셀로 장기 의존성 문제를 해결하고, 데이터 흐름의 역방향에 hidden layer이 추가된 구조를 가지고 있으며 주식 가격 입력 데이터에서 출력 결과에 가장 영향력을 주는 데이터에 집중하여 학습하는 모델이다.

첫 번째 실험은 Attention layer의 context vector만으로 그대로 사용하는 Att-Bi-LSTM-Only 모델이고, 두 번째 실험은 Attention의 context vector와 hidden layer의 마지막 시퀀스 벡터를 연결하여 더 많은 정보를 포함시켜 Att-Bi-LSTM-Concat 모델이고, 세 번째 실험 Attention의 context vector와 hidden layer의 마지막 시퀀스 벡터를 요소별 벡터 곱을 하여 데이터를 혼합하여 Att-Bi-LSTM-Dot 모델로 실험하였다.

그리고 정확도를 더욱 향상시키기 위해 각 종목코드에 맞게 학습 파라미터를 변화시키며 평균 제곱근 오차를 구해 실험을 하였다. 실험 결과 오차 향상률의 크기는 종목마다 주가 변동 패턴이 다르므로 종목코드마다 5회씩 반복 실행하여 최소값, 최대값, 평균을 구했다. 그 결과 기존의 Bi-LSTM 모델에 대비 Attention Bi-LSTM 모델들이 세 가지 모델 평균이 LG U+종목에서

는 25.72%, KT종목에서는 10.23%, SK텔레콤 종목에서는 6.12%을 보여 주가 예측의 성능 면에서 전체 평균 14.02%의 오차가 감소되어 정확도가 향상되었다.

키워드 : 볼린저 밴드, RNN, LSTM, Attention Mechanism, context vector

I. 서 론

1. 연구의 배경 및 목적

2016년 알파고가 이세돌과의 바둑대결에서 이기고 난 후 많은 사람들은 머신러닝의 위력을 더욱 인정하기 시작했고, 금융 투자 분야에서도 사람이 직접 하던 투자를 머신러닝기술을 적용하기 시작하였다. 그리고 경제학, 통계학, 컴퓨터 공학 등 관련 분야에서도 최신 지도 학습 딥러닝 기술에 의한 주가 예측이나 강화학습에 의한 주가 예측 연구들이 꾸준히 진행되고 있다.

하지만 미래의 주가 예측은 기업의 매출변동, 영업이익, 순이익 변동, 환율 변동, 원재료가격의 변화, 정부의 정책, 신기술 개발, 특허 등 다양한 주가 변동을 유발시키는 요인들로 인해 주가가 예상하지 못했던 방향으로 변동되어 버리는 경우가 있어서 여전히 어려움을 겪고 있다.

주가 데이터는 모두 숫자로 이루어져 있고, 숫자를 잘 계산하기 위한 수리적인 지식들도 잘 갖추어져 있기 때문에 딥 러닝(Deep Learning) 기술은 주가 예측에 적용하기 매우 용이하다. 다른 형태의 기계 학습과 달리 입력 데이터의 특징 패턴을 그대로 학습하기 때문에, 학습 이전에 입력 데이터의 특징 패턴에 따라 예측 결과가 큰 차이를 보이기 때문에 주가예측에 도움이 되는 다양한 지표들을 이용하여 시도하고 있다. 금융 투자를 할 때에도 주가 예측에 도움이 되는 데이터들을 수집한 후 딥러닝 기술로 예측 모델을 설계하고 그 예측 결과에 따라 투자를 진행하는 것이기 때문에 무엇보다도 정확도가 가장 우선시 되어야 한다.

기계학습의 지도학습에는 분류(classification) 모델과 회귀(regression) 모델이 대표적인데, 주가가 앞으로 올라갈지 내려갈지를 예측하고자 한다면 분류 모델로 정의하는 것이 더 예측력이 있고, 회귀(regression)는 연속된 데이터를 입력데이터로 받아 패턴을 분석한 후, 연속되는 데이터 형태로 예측 결과를 출력해주기 때문에 주가 자체를 예측하는 것은 회귀 모델로 정의하는 것이 더 예측력이 있다. 딥러닝 모델에 주가 분석의 보조지표를 적용한다면 과거 가격 움직임으로 현재와 미래의 가격 움직임을 더 정확히 예측할 수 있게 한다.

본 연구에서는 오픈소스 딥러닝 프레임워크인 TensorFlow를 이용하여 시계열 데이터인 과거의 주식 가격의 변동 패턴을 학습시켜 미래의 주식 가격

을 예측할 수 있는 딥러닝 모델로 적합한 순환신경망의 내부에 LSTM 메모리 셀로 장기 의존성 문제를 해결하고, 데이터 흐름의 역방향에 hidden layer이 추가된 구조의 신경망으로 주식 가격 입력 데이터에서 출력결과에 가장 영향을 미치는 부분을 집중 학습시킬 수 있는 Attention Mechanism 기반 Bi-LSTM 모델로 주가예측의 정확도가 향상되었다.

2. 주가 분석 방법

주가를 분석하는 방법에는 '기본적 분석(Fundamental Analysis)'과 '기술적 분석(Technical Analysis)'과 정서적 분석(Sentiment analysis)으로 나누어 볼 수 있다.

기본적 분석은 기업의 내재적 가치를 분석하여 미래의 주가를 예측하는 방법이다. 해당 기업의 매출액, 영업이익, 당기순이익, 영업이익률, 순이익률, 부채비율, 당좌비율, 주당배당금, 시가배당률 등의 내재가치가 주가보다 낮다고 판단되면 그 주식을 사고, 내재가치보다 높다고 판단되면 팔아서 수익을 낼 수 있다. 기본적 분석은 기술적 분석을 위해서도 중요하다. 재무제표, 주가와 손익 등의 수치 자료 등을 분석하는 것은 양적분석이고, 산업동향, 노사문제, CEO의 능력 등을 분석하는 것은 질적 분석이라고 하는데 이런 기본적 분석의 내재가치 평가는 주관적 판단이므로 검증이 어렵다.

또 기술적 분석은 과거의 주가와 거래량 등 주식시장에 나타난 과거의 데이터를 기초로 미래의 주가를 예측하고 매매시점을 포착하는 방법으로 흔히 '차트분석'이라고 한다. 개별주거나 시장흐름은 기본적으로 주식에 대한 수요와 공급에 의해 결정되는데 과거의 패턴을 분석하고 앞으로의 주가 추이를 예측하는 것이다. 가격의 움직임은 반복되고, 일정한 추세로 움직이기 때문에 주가 분석에 아주 유용하지만 기술적 분석이 현재 가격 수준과 미래 가격 수준을 예측할 수 있는 유용한 수단이고 차트 분석만으로는 장기투자를 하는 것이 위험할 수 있고, 거래량이 적으면 적을수록 오차가 발생하기 쉬우며, 작전세력이나 뉴스, 기사 등에 의해 주가가 예측하지 못했던 방향으로 변동되어 버릴 수 있다는 한계를 가지고 있어서, 차트분석도 결국은 주관적 분석이라는 논란이 있다. 기술적 분석은 차트에서 이동평균선의 배열, 볼린저 밴드[1], 골든크로스과 데드크로스 등의 추세선 지표들과 외국인/기관의 거래량 등을 파악한 후 이를 바탕으로 해당 차트의 지지와 저항, 돌파와 이탈 등을 판단하게 된다. 개인 투자자들은 기

본적 분석보다 이 기술적 분석을 더 많이 사용한다. 기술적 분석이 단기간에 큰 수익을 줄 수 있기 때문이다.

정서적 분석은 투자자들의 시장에 대한 욕심이나 두려움을 분석하는 것이다. 대상에 대한 긍정적 또는 부정적 표현을 분석하여 의사 결정에 활용되고 있고, 투자자들의 두려움이나 욕심을 수치화하여 텍스트 마이닝, 뉴스 텍스트 마이닝, 어휘 트렌드 분석, 소셜 정서 분석 등을 통해 주관적인 정보를 추출하여 분석하는 것이다.

본 연구는 다음과 같이 구성된다.

1장에서는 연구 배경과 목적, 주가 분석 방법을 기술하고, 2장에서는 주가 예측 모델 국내 선행 연구와 주가 예측 모델을 구축하기 위한 이론에 대해 설명하고, 3장에서는 본 연구에서 제안한 Attention Mechanism 기반 양방향 LSTM 순환 신경망을 설명하고 4장에서는 실험의 결과를 분석하며 마지막으로 5장에서는 결론을 논한다.

II. 이 론

1. 주가 예측 관련 국내 선행 연구

Song. et al(2017)[2]은 삼성전자 주가 예측을 위해 5가지 주가 가격 학습 모델을 설계하고 구현하였으며, 학습시 삼성전자에 관련된 입력 변수들만을 이용하였다. 주가의 경우 각 종목마다 가격대가 서로 많이 다르기 때문에 학습시 고려할 사항이 많으므로 삼성전자 한 종목만의 주가 데이터를 이용해 실험을 하였다. 학습을 위한 입력 데이터는 일별 시가, 고가, 저가, 종가, 거래량 5개의 단순 가격 데이터를 입력피처로 사용하였으며 예측레이블은 종가가 상승시 1, 하락시 0으로 예측 결과를 나타내며 약 2500개의 데이터로 학습과 테스트를 진행하였다. 최종적으로 56%의 학습 성공률을 갖는 학습 모델이 도출되었다. 하지만 실험데이터수가 2000여개로 적고, 입력 변수가 너무 단순하였다. 그래서 같은 해 Song. Lee et. al(2017)에서 텐서플로우를 이용해 단순 가격 피처데이터를 확장해 이동평균[15]을 추가하여 입력 변수 개수를 증가시켜가며 실험을 하여 세 가지 예측모델 DNN 구조로 단순 가격모델, 종가의 이동평균을 추가한 모델과 거래량의 이동평균을 추가하여 성능변화를 비교 분석하여 가격-기반 입력 변수가 주가예측에 미치는 영향을 평가하였다. 시가, 고가, 저가, 종가, 거래량을 입력변수로 사용한 DNN 모델에서는 기존 모델들보다 47.51%를 향상시킬 수 있었고, 종가의 5일, 10일, 20일, 60일, 120일 이동평균을 추가한 DNN 모델에서는 51.01%를 향상시킬 수 있었고, 종가 5일, 10일, 20일, 60일, 120일 이동평균과 거래량 5일, 10일, 20일, 60일, 120일 이동평균을 추가한 DNN 모델에서는 48.31%를 향상시킬 수 있었다.

Sin. et al(2017)[3]은 R의 신경망 패키지를 통해, 기존의 보조지표에 대해 다양한 조합으로 예측요소 적합여부를 검토하고, 예측률 향상을 위해 최적의 보조지표 조합과 환율, 환율 이동평균, 전산업 생산지수 등 환경 예측 요소들에 대해 연구하였다. 또한 추출된 예측요소를 입출력 패턴으로 DNN에 비해 RNN, LSTM 모델이 더 빠르고 안정적으로 학습하였으며, 종목별로 차이는 있으나 대략 15% 정도의 예측률 향상을 보였다.

Lee. et. al(2016)[4]은 인공지능 기술은 금융권에서도 투자, 신용 평가, 사기 방지 등에서도 사용되고 있다. 그래서 인공지능 알고리즘 Multi Layer

Perceptron, CNN, RNN을 사용하여 금융데이터를 분석하고 주가예측 모델을 구축하는 연구를 하였다. 10일 동안의 주가의 기술적 지표를 바탕으로 15일 후의 주가 당락을 예측한다. 시작가, 당일 최고가, 당일 최한가, 종가, 수정 종가, 거래량을 Bollinger Band, DMI, 이동평균, Stochastic, MACD로 변환하여 학습데이터로 사용하여 종목이 주가지수에 받는 영향을 감안해 KOSPI20, NIKKI, S&P 500, NASDAQ의 지수를 기술적 지표로 변환하여 학습에 사용하였다. 신경망은 종목마다 다른 알고리즘을 적용하여 정확도를 측정한 결과 MLP는 51.11%, CNN은 51.19%, RNN은 52.09%를 기록하였다. 그리고 종목마다 다른 모델을 생성하는 것이 유리함을 알게 되었다.

Lee, et, al(2018)[5], 약 8년간의 KOSPI200 데이터 2009년 1월부터 2017년 2월까지의 2,026건의 데이터를 가지고 이미지와 영상의 인식 및 분류에 효과적인 딥러닝의 일종인 합성곱 신경망을 이용하여 주식시장에서 기술적 분석가들에게 검증된 중요 기술지표들의 5일 간의 변동을 그래프로 만들어 이미지를 입력 데이터 세트로 사용했다. 합성곱 모델의 주식시장에서의 주가지수 등락 예측 적용 가능성을 확인하고자 KOSPI200 데이터를 사용하여 또한 로지스틱 회귀분석, 인공신경망, SVM과 같은 전통적인 머신러닝 알고리즘과 비교하여 제안한 모델이 더 우수한 성능을 보였고 그 차이가 통계적으로 로지스틱 회귀분석 99%, 인공신경망과 95%, SVM과 95% 신뢰성을 보였다.

Joo. et. al(2018)[6]은 순환신경망의 시계열 데이터의 직전 패턴 기반으로만 학습하는 경향을 보이는 한계를 해결하기 위하여, 데이터의 흐름의 역방향에 hidden layer이 추가되는 양방향 LSTM 순환신경망을 이용하여 주가예측 모델을 구현하였다. 실험에서는 제시된 주가예측 모델에 텐서플로우를 이용하여 주가와 거래량을 입력 값으로 학습을 하였다. 주가예측의 성능을 평가하기 위해서, 실제 주가와 예측된 주가 간의 평균 제곱근 오차를 구하였다. 실험결과로는 단방향 LSTM 순환신경망보다, 양방향 LSTM 순환신경망을 이용한 주가예측 모델이 더 작은 오차가 발생하여 주가예측 정확성이 $0.68 * 10^{-3}$ 향상되었다.

2. 순환 신경망(Recurrent Neural Network, RNN)[4][13]

“Recurrent” 는 라틴어에서 온 말로, “몇 번이나 반복해서 일어나는 일” 을 말하고 우리말로 “주기적으로 일어난다”, “순환한다” 등으로 번역할 수 있다. 최근 들어 널리 사용되는 딥 러닝 기술 중 순환신경망(Recurrent Neural Network)은 인공신경망에 시계열 개념이 추가된 것으로 입력되는 모든 순차적인 요소에 대해 동일한 계산을 하기 때문에 순환적(Recurrent)으로 진행되고 hidden layer에 이전 정보를 기억시킬 수 있다. 순환 신경망에서는 입력되는 주가 데이터와 이전 단계 결과를 함께 입력으로 받아들여 처리하는 절차를 반복하고, 이전 데이터가 다음 데이터에 어떤 영향을 주는지 학습하기 때문에 시간 순서대로 주어지는 주가 예측에 적합한 모델이다. 순환 신경망은 기계번역(Machine Translation, MT)과 같은 복잡한 문제에도 사용되어 번역 대상인 단어를 순차적으로 입력받아 원하는 언어로 번역되면서 음성인식과 이미지인식, 자연어처리 등에 많이 응용되고 있다.

순환신경망 모델이 순환되기 위해서는 “닫힌 경로”가 필요하다. “닫힌 경로”라는 것은 “순환하는 경로”가 존재해야 데이터가 같은 장소를 반복해서 왕래할 수 있는 것이고 데이터가 순환하면서 정보가 끊임없이 과거의 정보를 기억하는 동시 최신 데이터로 갱신될 수 있다.

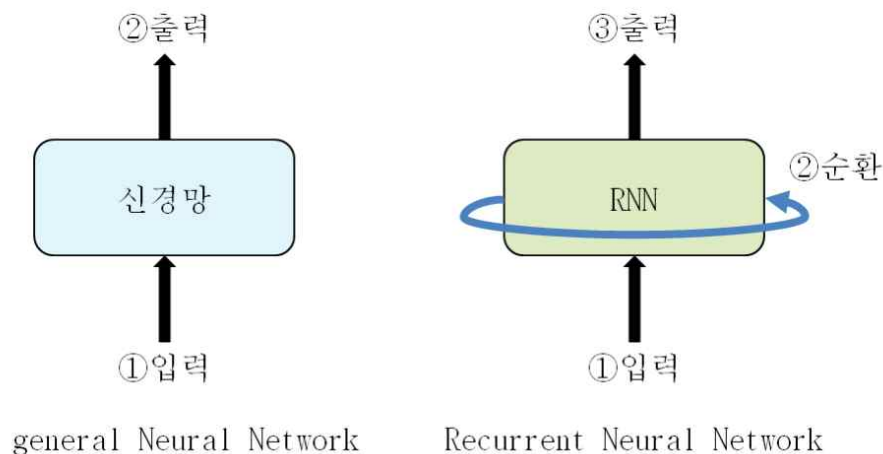


그림 2-1 인공 신경망과 순환 경로를 포함하는 순환신경망

<그림 2-1>처럼 순환 신경망은 순환하는 경로를 포함한다. 이 순환 경로를 따라서 데이터를 layer안으로 순환시킨다. 실제로 하나의 RNN 계층을 펼쳐보면 <그림 2-2>와 같다.

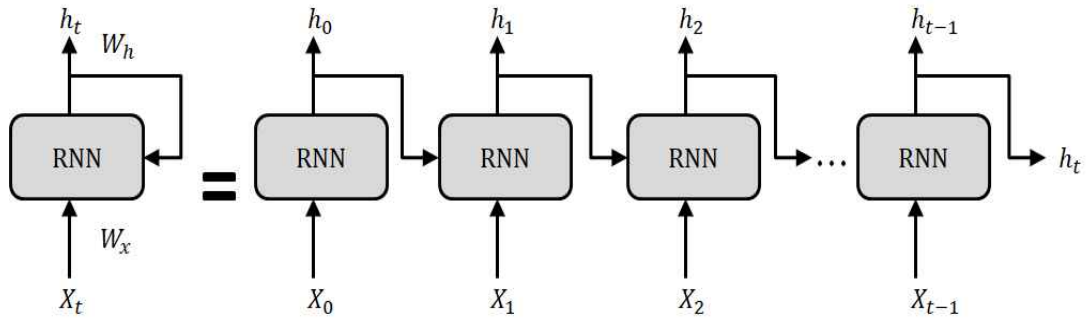
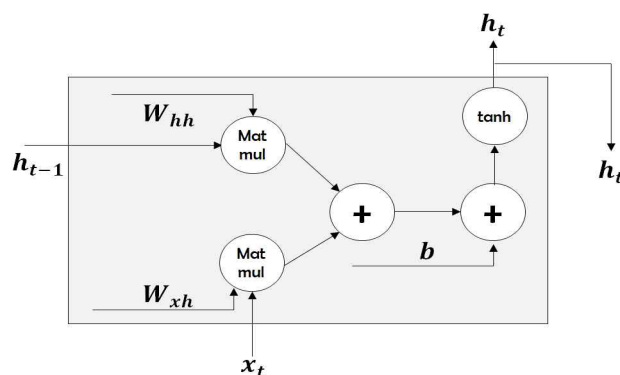


그림 2-2 RNN 계층의 순환 구조 펼쳐진 모습(출처:[13])

RNN의 순환 구조를 펼친 모습은 마치 한 방향으로 흐르는 feedforward 신경망처럼 보이며, 신경망의 사용 사이에 상태를 순환계층이라는 특별한 유형의 신경망을 활용하기 위해 feedforward 신경망과는 달리 내부에 루프가 있어 네트워크의 정보가 지속되도록 도와주는 역할을 한다.

각 시점의 RNN 계층은 현재 시간의 입력과 이전 시간의 은닉 상태 값으로부터의 은닉 상태(hidden state)를 입력으로 받는다. 은 입력이고, t 는 시점을 뜻하는데, 이는 시계열 데이터($x_1, x_2, x_3, \dots, x_t$)가 RNN 계층에 입력됨을 의미한다. 그리고 그 입력에 대해서 hidden layer($h_1, h_2, h_3, \dots, h_t$)이 출력된다. RNN 계층은 이전 계층으로부터 h_{t-1} 을 출력을 받아들이고, 현재 입력 x_t 를 받아 가중치(weight)를 부여하여 그 다음 결과 값 h_t 를 출력하여 계산한다. 이 때 사용하는 비선형 함수로는 보통 tanh와 sigmoid함수를 사용한다. 이 비선형 함수는 출력 값의 범위를 제한하고 전 구간에서 미분이 가능한 함수이기 때문에 역전과 계산에 잘 적용되며 첫 은닉 상태를 계산하기 위해서 보통 0으로 초기화 된다.



(출처 : [13])

그림 2-3 RNN 계층의 계산 그래프

$$= \tanh(h_{t-1} \cdot W_h + x_t \cdot W_x + b) \quad \text{수식 1}$$

RNN 계층의 계산 그래프는 <그림 2-3>과 같이 표현할 수 있는데 가중치 (*weight*)가 2개 있다. 입력 x 를 출력 h 로 변환하기 위한 가중치 W_x 가 있고, 또 하나는 1개의 RNN 출력을 다음 시각의 출력으로 변환하기 위한 가중치 W_h 와 편향 b 이 있다. 참고로 h_{t-1} 과 x_t 는 행 vector이다.

각 시각에 입력되는 x_t 가 vector라고 가정할 때, 1일의 주가 데이터를 다룬다면, 각 가격의 분산 표현(vector)가 x_t 가 되며, 이 vector가 순서대로 RNN 계층에 입력되는 것이다. RNN 계층의 feedforward 신경망은 <그림 2-3>과 같이 행렬의 곱인 ‘matmul’과 ‘+’, ‘tanh’ 3개의 연산으로 구성되어 있고 편향 b 의 덧셈에서는 브로드캐스트 연산이 일어난다. RNN은 데이터의 길이에 상관없이 같은 연산을 각 데이터에 반영하고, 이전 데이터의 연산 결과가 현재 데이터에 반영되어 연속적인 입력 데이터를 같은 크기의 vector로 표현할 수 있다.

<수식 1>의 결과에서 각 시각 t 의 출력은 h_t 가 된다. 이 h_t 는 다른 계층을 향해 위쪽으로 출력되는 동시에 다음 시각의 RNN 계층(자기 자신)을 향해 오른쪽으로 출력된다. 따라서 현재 출력 h_t 는 이전 출력 h_{t-1} 에 기초해 계산된다. h 는 hidden 계층(은닉 레이어 또는 은닉 상태 vector) 또는 “상태를 가지는 레이어 또는 기억력을 가지는 계층” 이라고 할 수 있다. 그리고 해당 weight(W)는 모든 타임스텝에 동일하게 적용이 된다. 그래서 RNN은 시계열 데이터(time-series data)와 같이 시간의 흐름에 따라 변화하는 데이터를 학습하기 위한 딥러닝 모델인 것이다.

RNN의 결과를 더욱 개선하기 위한 노력은 계속되어 왔고, 입력값과 출력값을 어떻게 설정하느냐에 따라 여러 가지 상황에서 이를 적용되고 있다.

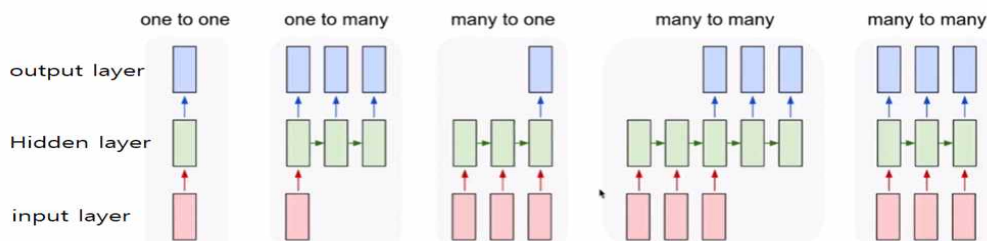


그림 2-4 Recurrent Neural Networks 설계 유형(출처:[14])

RNN은 Input 계층, Hidden 계층, Output 계층로 구성되는데, Input 계층에는 고정된 사이즈의 Input Vector가 들어가고 Hidden 계층을 거쳐 Output 계층에서 고정된 사이즈의 Vector가 나온다. 이런 전통적인 RNN은 one-to-one 유형이라고 볼 수 있는데, 매 입력마다 출력반응을 각각 생성된다.

그리고 사진 한 장을 입력으로 받아들이 사진의 설명 문장을 출력 값으로 구하는 Image Captioning을 하는 경우는 One To Many 유형(단일 입력 다수 출력)으로 설계할 수 있고, 소비자의 상품평이나 Twitter의 review 글, 문장을 읽고 뜻을 파악하는 단어로 표현하는 경우에는 긍정적인지 부정적인지 하나의 클래스 Sentiment로 판별하는 Sentiment Classification(감성 분석) 같은 경우에는 Many To One(다수 입력 단일 출력) 유형으로 설계할 수 있고, 또 영어로 된 문장을 한국어로 된 문장으로 자동 번역하는 기계번역(Machine Translation)과 Video Classification On Frame Level로 블랙박스 영상을 모든 프레임 하나하나를 분류하여 예측이 현재시점에 국한되지 않고 현재의 프레임 이전의 모든 프레임을 기억해서 모든 각각의 Time Step의 프레임이 현재 프레임과 지나간 모든 프레임들까지 함수로 입력 받아 사고 발생여부를 판별하는 경우도 Many To Many(다수 입력 다수 출력)유형으로 설계할 수 있다.

본 연구의 주가 예측 모델은 다양한 과거의 가격들을 입력으로 받아 들여 다음 날의 주가를 예측하므로 Many To One 유형으로 설계하였다. 과거의 주식시장의 데이터를 기반으로 하여 주식시장에서의 주가 변화 학습 방법으로 미래의 주가 예측하는 경우에는 패턴을 학습하기 위해 hidden layer의 가중치와 출력층의 가중치를 수정하기 위해 역전파 신경망을 사용하게 된다. 그런데 역전파 신경망에서는 정답 레이블이 주어진 시점으로부터 과거 방향으로 기울기를 전달하게 된다. RNN 계층이 과거 방향으로 의미 있는 기울기를 전달함으로써 시간 방향의 의존관계를 학습한다. 그때 기울기는 원래대로라면 학습해야 할 의미가 있는 정보가 들어있고, 그것을 과거로 전달함으로써 장기 의존 관계를 학습하게 된다. 활성화 함수들의 미분 값들은 0으로 수렴하게 되는데, 기울기가 0에 가까운 값으로 곱해지면 이전 계층의 기울기들도 0으로 수렴하게 만든다. 행렬에 작은 값들이 들어 있는데 행렬 곱이 계속 이루어지면 기울기는 지수 함수로 감소하고, 시간 단계가 몇 단계 진행되면 기울기가 아예 사라져 버리는 것이다. 그러다 보니 입력에서 여러 단

계가 멀어지게 되면 기울기가 제대로 전달되지 못하고, 또 먼 과거의 상태는 현재 상태의 학습에 영향을 주지 못하여 장기적인 의존성을 제대로 배우지 못한다.

RNN 계층이 많으면 많을수록 Gradient 소실 문제가 발생하는데 만약에 기울기가 중간에 소실되면 가중치 매개변수는 전혀 갱신될 수 없게 된다. 그래서 RNN은 구조가 단순하기 때문에 구현하기 쉽지만 시계열의 데이터에서 시간적으로 멀리 떨어진 장기 의존 관계를 잘 학습할 수 없어서 성능 면에서 효율이 떨어진다.

특히 첫 번째 입력과 계산된 Backward Gradient는 순간마다 감소하거나 증가하는 경향이 있고, 특정 순간 이후에 0으로 수렴하거나 무한대로 폭발하는 경향이 있다는 단점을 가지고 있다. 이 문제를 해결하기 위해서 장단기 메모리(LSTM : Long-Short term Memory) 구조의 순환 신경망을 사용할 수 있다.

3. LSTM(Long-Short term Memory) 네트워크

LSTM은 1997년 호이체르이터(Hochreiter)와 슈미트후버(Schmidhuber)가 처음 생각해낸 특수한 순환 신경망이 진화한 형태이다. Long short-term memory(LSTM) 네트워크가 소개된 지 20여년이 되었지만, 이러한 유형의 신경망들이 최근 딥러닝 맥락에서 계속 재발견되고 있어 소멸되는 기울기 문제가 없어지기 때문에 뛰어난 결과와 성능을 제공한다. LSTM 네트워크는 순차 데이터를 모델링하고 시간 순서에 따라 예측하거나 분류하는데 이상적이라 자연어 처리(NLP) 분야에도 많이 사용되고 있고, 많은 전통적인 접근 방식들 보다 성능이 뛰어나 이를 대체하여 사용되고 있다.

반복을 많이 해야 하는 RNN의 단점을 보완하기 위해 개발된 LSTM은 기본적인 RNN의 입력된 데이터와 참고해야 할 데이터의 위치 차이가 커질 때 연결하기 힘들었던 부분을 해결하는데 도움을 준다. LSTM의 핵심인 cell state라고 불리는 이 메모리 셀은 Weight를 계속 기억할 것인지 말 것인지를 결정한다.

LSTM에서는 장기 의존성이 있는 데이터를 학습시키기 위해 각 단계에 두 개의 출력을 두게 된다. 하나는 모델의 실제 출력이 되고, 다른 하나는 해당 단계의 내부에 작은 메모리를 가진 LSTM Cell로 교체해서 사용한다. 이 LSTM Cell은 RNN과 같이 연결되어 셀 내부 상태가 모두 다음 단계에 공급되

어 여러 단계를 거친다고 해도 경사가 소실되거나 발산할 가능성이 낮아진다. LSTM 네트워크는 오래된 정보를 계속 유지할 수 있어서 큰 장점을 가진다. LSTM은 과거의 data를 계속해서 update하므로, RNN보다 지속적이다 라는 장점을 가지고 있고, 각각의 메모리 제어가 가능하여 결과 값을 제어할 수 있다는 장점도 있지만, 메모리가 났어 썩어질 가능성이 있고 연산 속도가 느리다는 단점도 있다.[10][11][12][13]

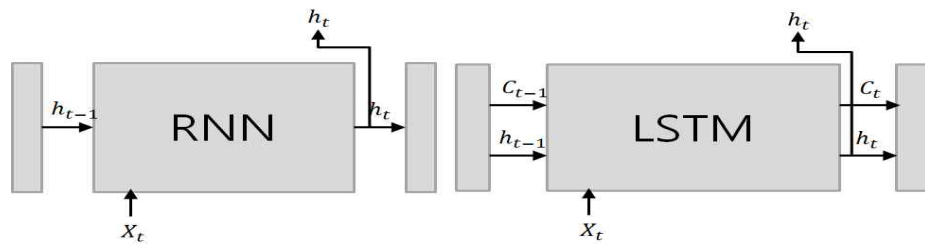


그림 2-5 기본적인 RNN 계층과 LSTM 계층 비교(출처:[13])

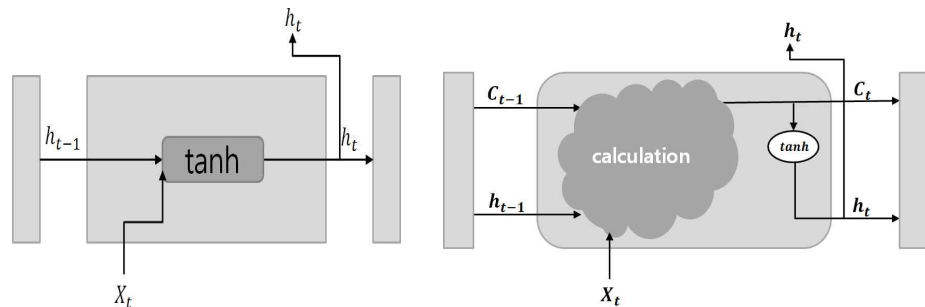


그림 2-6 기본적인 RNN 계층과 LSTM 계층 내부(출처:[13])

<그림 2-5>×<그림 2-6>과 같이 기본적인 RNN은 이전 단계의 hidden layer의 내용과 현재 단계의 입력을 받아들여 tanh함수를 통과하여 다음 hidden layer으로 전달되는 반복적인 모듈 체인 형태를 갖는 single 계층이지만, LSTM은 hidden 계층 역할을 하는 것이 h_{t-1} 와 c_{t-1} 두 개인데, c_{t-1} 와 세 번의 gate들의 결과를 적절히 섞어서 하나의 c_t 를 만들어 낸다. 현재의 기억 셀 c_t 는 3개의 입력(c_{t-1} , h_{t-1} , x_t)으로부터 ‘어떤 계산’을 통해 구하는 특별한 방법으로 상호작용하는 gate layer라는 복잡한 구조를 가지고 있다. 여기서 핵심은 갱신된 c_t 를 사용해 은닉 상태 h_t 를 계산한다는 것이다. 또한 이때 계산은 $h_t = \tanh(c_t)$ 로 이는 c_t 의 각 요소에 tanh 함수를 적용하게 된다. c_t 의 각 원소에 tanh 함수를 적용할 때 기억 셀 c_t 와 은닉 상태 h_t 의 원소 수는 같게 해서 계산을 한다.

(1) cell state

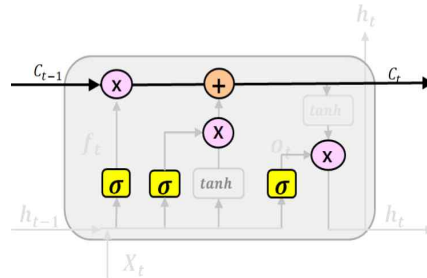


그림 2-7 cell state

기억 셀 는 세 개의 gate들을 이용하여 셀 상태를 제어하고 정보의 반영 여부를 결정한다. cell state는 다이어그램의 상단을 가로 지르는 수평선으로 전체 체인을 통과한다. 이 구조를 통해 정보는 큰 변함없이 계속 다음 단계로 전달이 될 수 있다. 간단히 표현하면 hidden 계층에 기존 RNN처럼 단순히 hidden unit하고 input을 이용해서 activation function을 거쳐 바로 output으로 나가거나 순환되어 다음의 hidden unit으로 작용되는 게 아니라, 기억할 것이냐 잊을 것이냐를 학습할 수 있도록 $cell(c_t)$ 이라는 개념이 추가되었다. 각 시간적 흐름을 조절을 해서 꾸준히 그 메모리의 내용을 관리하여 다음에 넘겨줄지 잊어버릴지 컨트롤하는 네트워크를 각 유닛에 가지고 있는 것이다.

LSTM은 cell state와 cell 상태를 보호하고 제어하는 세 개의 gate, 네 개의 파라미터를 갖는다. LSTM은 이 cell state에 신중하게 정제된 구조를 가진 gate라는 구조로 신중하게 규제되는 cell 상태에 정보를 제거하거나 추가 할 수 있는 기능이 있다. 그리고 gate들은 선택적으로 정보들이 흘러들어갈 수 있도록 만드는 장치를 의미한다.

gate들은 (sigmoid) 신경망 계층과 element 단위 \otimes (곱셈)연산으로 이루어져 있고, cell state는 곱셈과 덧셈을 거쳐 LSTM cell을 통과한다. 이 c 에는 시각 t 에서의 LSTM의 기억이 저장되어 있는데, 과거로부터 t 시각까지 필요한 모든 정보가 저장되도록 학습을 수행한다. 그리고 필요한 정보를 모두 간직한 이 기억을 바탕으로, 외부 계층에 그리고 다음 시각의 LSTM에 은닉 상태 h_t 를 출력한다.

c_t : cell은 LSTM을 기억한다. LSTM은 적절한 c_t, c_{t-1} 의 조합에 의해 만들어진 장단기 메모리다. 이것이 output으로 출력된다.

3개의 gates는 i_t, o_t 인데 모두 0에서 1사이의 값이다.

f_t : 이전의 것을 얼마나 잊어버릴지 결정하고, i_t : 현재 값을 얼마나 반영할지(저장할지) 결정하고, o_t : 무엇을 출력할지를 결정한다.

2개의 outputs은 h_t, c_t 이고, x 파라미터는 i, W_f, W_o, W_h 네 개가 있다.

결론은 반복 모듈인 LSTM는 상호작용을 하는 3개의 gate layer가 존재하는데 무엇을 기억하고 잊을 것인가(forget gate), 무엇을 저장할 것인가(input gate), 무엇을 출력할 것인가(output gate)라는 세 개의 gate를 가지고 있다는 것이다.

(2) forget gate

LSTM의 첫 번째 단계에서는 <그림 2-8>과 같이 세 개의 gate들이 각 정보를 어느 정도 반영할 것인지 결정한다. 메모리 cell은 시간 경과에 따라 학습한 중요한 정보를 유지하고 있으며, 신경망은 많은 시간 단계에 걸쳐 메모리 cell에서 유용한 정보를 효과적으로 유지하도록 설계됐다. 시간 단계마다 한 LSTM unit은 세 단계의 새로운 정보로 메모리 cell을 수정한다.

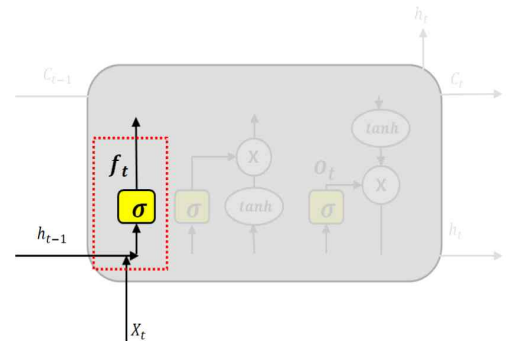


그림 28 무엇을 잊을 것인가(forget gate)

forget gate는 어떤 정보를 반영할지 안할지 결정하는 gate인데 σ (sigmoid) 층을 통해 이루어진다. forget gate는 <그림 2-8>과 같이 h_{t-1} 와 x_t 를 입력으로 받아들이며 σ (sigmoid) 활성화 함수를 통해 $0 < \sigma < 1$ 의 값을 출력한다. 각 구성요소가 영향을 주게 될지 말지를 결정해주는 역할을 한다. 0이라는 값을 가지게 되면 해당 구성요소가 미래의 결과에 아무런 영향을 주지 않게 되고, 1이라는 값을 가지게 되면 해당 구성 요소가 확실히 미래의 예측결과에 영향을 주도록 데이터가 흘러가게 된다. <수식 2>를 실행하면 forget gate의 출력 f_t 가 구해진다.

$$f_t = \sigma(x_t W_x^{(f)} + h_{t-1} W_h^{(f)} + b^{(f)}) \quad \text{수식 2}$$

그리고 이 f_t 와 이전 기억 셀인 c_{t-1} 와의 원소별 곱, 즉 $c_t = f_t \odot c_{t-1}$ 를 계산하여 c_t 를 구한다. 이전 cell state에서(c_{t-1})의 기억 중에서 불필요한 내용을 잊게 해주는 gate로 h_{t-1} 어떤 정보를 삭제할지 기억할지 정보를 결정하

는 이전 cell의 output인 h_{t-1} 과 현재의 입력인 x_t 를 sigmoid 활성화 계층(σ)에 적용해 각 hidden unit별로 0과 1사이의 값을 얻는다. 이 값을 현재 상태와 element끼리 곱한다. h_{t-1} 과 x_t 를 보고 cell 상태 c_{t-1} 의 각 번호에 대해 0과 1 사이의 숫자를 출력한다. 1은 "이 항목을 보관하라"를 나타내며 0은 "이 항목을 제거하라"를 의미한다.

(3) input gate

두 번째 단계는 셀 상태에 어떤 값을 저장할 건지를 결정하는 단계이다. input gate에는 두 개 계층이 있는데, 하나는 sigmoid 계층으로 어떤 값을 업데이트 할 것인지 결정하는 것이고 input gate 계층(좌측), 또 다른 하나는 C_t 라고 부르는 sigmoid 함수와 tanh함수를 이용해 cell state에 영향을 주기 위해서 더해질 새로운 예비 값들의 vector를 저장할지 결정하는 계층이다. \tilde{C}_t 의 각 요소가 새로 추가되는 정보로써의 가치가 얼마나 큰지를 판단한다. 이전의 주식 가격 흐름 패턴을 버리고 새로운 흐름 패턴을 저장하고자 하는 것이다. <수식 3>은 sigmoid 층, <수식 4>은 tanh층을 표현한 수식이다.

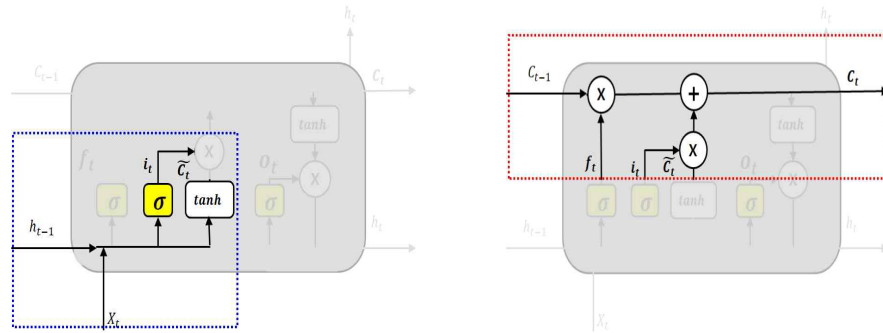
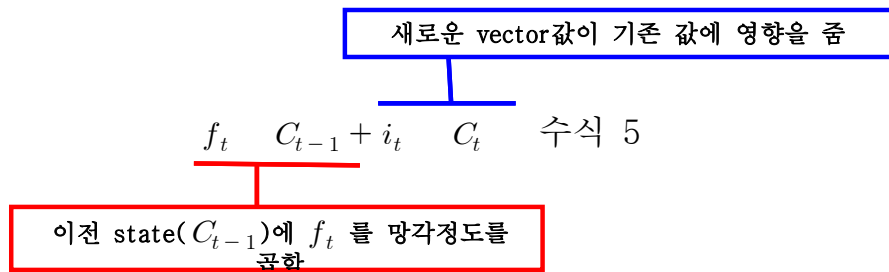


그림 2-9 무엇을 저장할 것인가(input gate)

$$i_t = \sigma(x_t W_x^{(i)} + h_{t-1} W_h^{(i)} + b^{(i)}) \quad \text{수식 3}$$

$$\tilde{C}_t = \tanh(x_t W_x^{(c)} + h_{t-1} W_h^{(c)} + b^{(\tilde{c})}) \quad \text{수식 4}$$

망각 게이트와 입력게이트에서 출력된 값들을 cell state C_{t-1} 를 C_t 로 업데이트 해준다. f_t C_{t-1} 계산후 i_t \tilde{C}_t 을 계산해서 더해준다. 최종 C_t 는 다음과 같다.



(4) output gate

output gate에서는 출력 값인 h_t 를 결정한다. h_t 는 cell state C_t 를 여과 시켜준 값이다. 첫 번째로 출력할 cell state를 결정하는 sigmoid 계층을 통해 o_t 를 계산한다. 그런 다음 전 단계에서 업데이트 된 cell state C_{t-1} 를 tanh를 통해서 -1~1사이의 값으로 출력된 값과 sigmoid를 통해 나온 o_t 를 곱하여 원하는 값만 결과 값에 반영된다. o_t 가 0이면 반영될 수 없다. <수식 6>은 sigmoid 결과이고, <수식 7>은 tanh 결과이다.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad \text{수식 6}$$

$$h_t = o_t \cdot \tanh(C_t) \quad \text{수식 7}$$

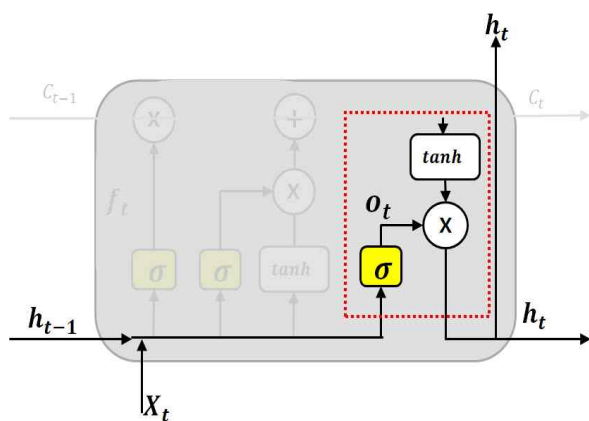


그림 2-10 무엇을 출력할 것인가(output gate)

마지막 sigmoid 함수와 cell state에서 나온 출력을 통과시킨 내용과 마지막 tanh 함수로 어떤 출력 값을 출력할지를 결정한다. 어떤 출력 값을 출력할 것인지 결정하는 sigmoid 계층인 “output gate”이다. sigmoid로 0~1사이의 값이 나왔다가 현재 cell state값이 tanh 활성 계층에 통과해서 -1과 1사이의 값으로 출력된다. cell input(h_{t-1}, x_t)을

sigmoid 활성 계층()에 통과시킨 결과와 element 곱을 한다. 이 과정이 우리가 어떤 결과를 내보내줄 것인지 필터링하는 역할을 수행한다. 이 결과인 h_t 가 다음 시퀀스의 입력으로 넘겨지며, 또한 네트워크의 다음 계층으로 전달된다. 전체적으로 LSTM의 반복 모듈의 내부 구조를 살펴 보면 <그림 2-11>과 같다.

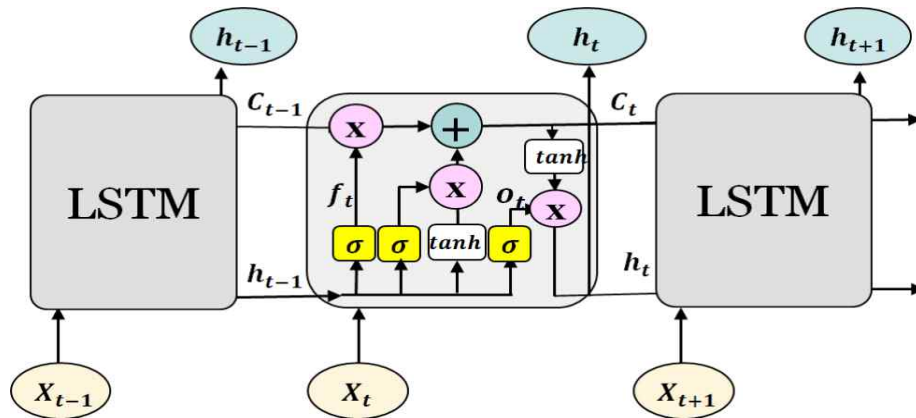


그림 2-11 LSTM의 반복 모듈 내부구조

4. Bi-directional Recurrent Neural Network

순환 신경망은 일반 신경망 학습과 달리 시계열 개념이 추가되어 은닉 계층에 과거의 데이터의 정보를 기억할 수 있다. 하지만 입력 순서를 시간 순서대로 처리하기 때문에 결과는 주로 직전 패턴만을 기반으로 하는 경향을 보이는 한계가 있다. 이러한 한계를 해결하기 위해서는, 양방향 순환 신경망을 사용하는 것이 효율적이다. 양방향 순환신경망은 순방향과 역방향의 두 개의 분리된 순환신경망을 통해 학습이 진행된다. 이러한 방법으로 학습되는 시계열 데이터의 역방향에 hidden layer를 추가시킨다면 성능이 더 향상된다.[6]

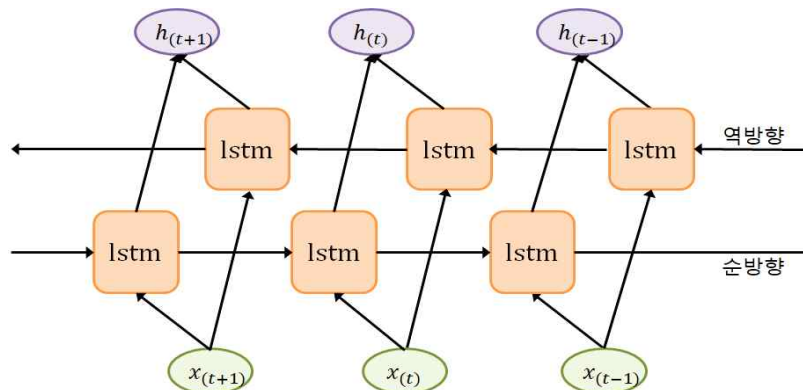


그림 2-12 양방향 순환신경망(Bidirectional Recurrent Neural Network)

순환신경망은 입력과 출력 계층에 순방향의 hidden layer만 존재하지만, <그림 2-12>에서의 양방향 순환신경망은 입력과 출력 계층에 순방향 hidden layer와 추가로 역방향 은닉 계층이 연결된 구조를 보여준다.

Bi-directional RNN(Schuster, 1997)은 ‘과거로부터’와 ‘미래로부터’라는 단방향 시간 축에 대해 hidden 계층의 상태를 전달해준다. 두 종류의 hidden 계층으로 하나의 hidden 계층은 과거의 상태를 반영하고 나머지 하나가 미래의 상태를 반영하고 있다. 즉 양방향으로 학습을 진행하고 이를 각각의 hidden 계층에서 나온 출력 값을 연산하여 최종 출력 값을 활용한다.[13]

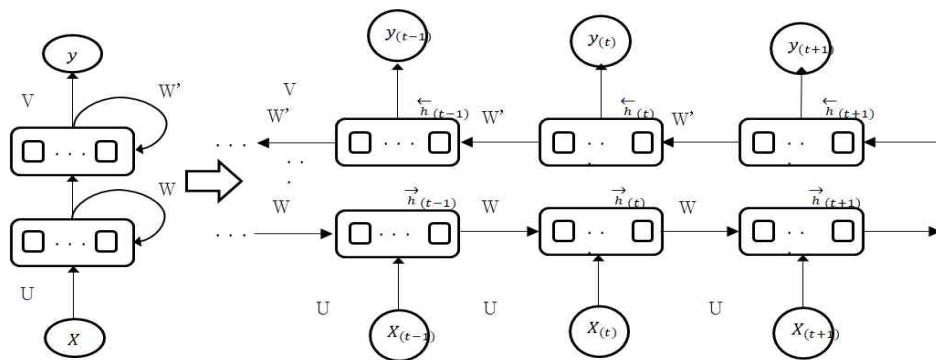


그림 2-13 Bi-RNN 전개도

입력에서 출력까지를 시간 축에 따라 $t-1$, t , $t+1$ 이라는 3단계로 전개해보면 과거의 상태만 생각하는 경우는 한 단계 이전 과거의 hidden 계층 값이 각각 다음 단계로 전달된다. 이 hidden 계층에 있는 각 뉴런을 LSTM으로 설계한 것이다.

그렇다면 BiRNN은 과거와 함께 미래의 hidden 계층의 상태도 재귀적으로 feedback할 수 있게 만든 모델이므로 <그림 2-13>과 같이 나타낼 수 있다.

시간 t 에서의 출력 값이 이전 time step 외에, 이후의 미래 요소에 의존할 수 있다는 생각에 근거한다. 이를 구현하려면 두 RNN의 출력을 혼합하면 된다. 첫 번째 프로세스를 단방향으로 실행하고, 두 번째 프로세스는 반대 방향으로 실행한다. 네트워크는 정규 RNN의 뉴런을 양의 시간 방향(순방향 상태)과 음의 시간 방향(역방향 상태)의 두 방향으로 분리한다. 이 구조에 의해 출력층은 과거 및 미래 상태로부터 정보를 얻을 수 있다.

시간 $t-1$, t , $t+1$ 에 따라 전개하는 것으로 두 종류의 hidden 계층 각각을 보면 입력 계층에서 값을 받아 출력 계층으로 값을 전달한다는 것은 일반적인 신경망과도 같다. 그러나 각각의 hidden 계층은 동일한 hidden 계층의 과거 또는 미래로부터 값을 받는다는 것이다. 과거용 hidden 계층과 미래용 hidden 계층 사이에는 아무런 관련이 없고 완전히 별개의 것이라고 생각해

도 무관하다. 시간의 흐름을 보면 forward는 과거에서 미래로 순방향으로 전달하고, backward는 미래에서 과거라는 후 방향으로 값을 전달하는 것이다.

예를 들어 이미 가지고 있는 시계열 데이터가 어느 클래스로 분류되는지를 알고 싶은 경우에는 과거에서 미래(현재)까지 모든 것을 알고 있는 상태에서 모델을 만들어간다. 이경에는 과거→미래라는 단방향 보다는 미래→과거라는 방향과 함께 두 방향으로 진행되는 시계열의 의존관계를 고려하는 편이 훨씬 좋은 정확도를 기대할 수 있다. Bi RNN은 ‘과거로부터 ‘와 ’ 미래로부터 ‘라는 단방향 시간 축에 대해 hidden 계층의 상태를 전파해가는 것이다. 이 두 방향을 구현하려면 일반적인 hidden 계층의 구조를 약간 바꿔야 한다. hidden 계층의 구조가 두 종류로 구성되어 하나는 과거의 상태를 반영하고 하나는 미래의 상태를 반영하고 있는 상태를 만들어 주면 된다.

시간 t에서의 순방향 층과 후 방향 층의 값은 각각 별개로 생각할 수 있으므로 각각을 $h(t)$ 와 $\overleftarrow{h}(t)$ 와 같이 화살표의 방향으로 표시하기도 한다. 그러면 각각의 순방향은 <수식 8>와 같이 나타낼 수 있기 때문에 일반적인 신경망과 다르지 않다.

$$\vec{h}(t)=f(\vec{U}x_{(t)} * \vec{W}h_{(t-1)}+\vec{b}) \quad \text{수식 8}$$

$$\overleftarrow{h}(t)=f(\overleftarrow{U}x_{(t)} * \overleftarrow{W}h_{(t-1)}+\overleftarrow{b}) \quad \text{수식 9}$$

따라서 hidden 계층(전체)의 값 $h(t)$ 는

$$h(t)=\begin{pmatrix} \vec{h}(t) \\ \overleftarrow{h}(t) \end{pmatrix} \quad \text{수식 10}$$

위와 같이 단순히 $\vec{h}(t)$ 와 $\overleftarrow{h}(t)$ 를 나열해서 표현할 수 있고, 출력도 그대로 $y(t)=h(t)+b_y$ 와 같이 나타낼 수 있다.

5. Attention Mechanism 기반 시퀀스-투-시퀀스

시퀀스-투-시퀀스는 입력된 시퀀스로부터 다른 도메인의 시퀀스를 출력하는 다양한 분야에서 사용되는 모델이다. 예를 들어 Chatbot이나 Machine Translation이 그러한 대표적인 예인데, 입력 시퀀스와 출력 시퀀스를 각각 질문과 대답으로 구성하면 Chatbot으로 만들 수 있고, 입력 시퀀스와 출력

시퀀스를 각각 입력 문장과 번역 문장으로 만들면 번역기로 사용할 수 있다. 그 외에도 Text Summarization(내용 요약), STT(Speech to Text) 등에 쓰일 수 있다.

seq2seq는 번역기에서 대표적으로 사용되는 모델이다. seq2seq는 크게 두 개로 구성된 아키텍처로 구성되는데, 바로 인코더와 디코더이다. 인코더는 입력 문장의 모든 단어들을 순차적으로 입력받은 뒤에 마지막에 이 모든 단어 정보들을 압축해서 하나의 벡터로 만드는데, 이를 context vector라고 한다. 입력 문장의 정보가 하나의 context vector로 모두 압축되면 인코더는 context vector를 디코더로 전송한다. 디코더는 context vector를 받아서 번역된 단어를 한 개씩 순차적으로 출력한다. 영어 문장을 프랑스어 문장 쌍을 이루는 데이터에 대해 유용한 학습 방법으로 처음 소개 되었다. 이 학습 방법은 순환 신경망의 한 종류이며, 순환신경망이 여러 개 연결된 인코더와 디코더의 구조를 사용한다.[16]

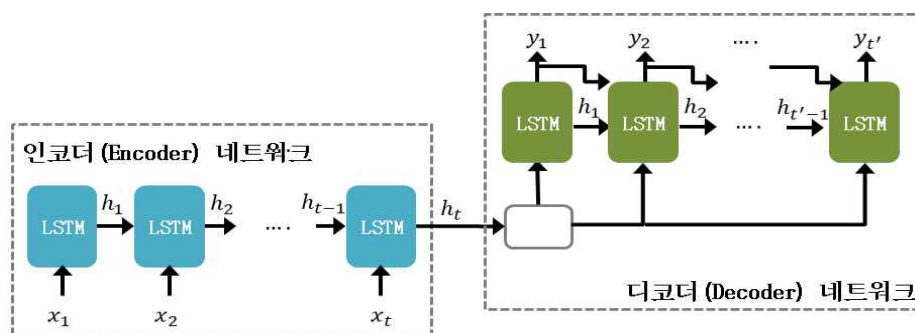


그림 2-14 기본적인 시퀀스투시퀀스 모델

기본적인 시퀀스투시퀀스는 영어를 프랑스어로 번역하는 신경망 기계번역 모델을 구출할 목적으로 2014년 뉴욕대학교 조경현 교수가 제안한 것으로, <그림 2-14>와 같이 인코더 네트워크와 디코더 네트워크로 불리는 두 개의 RNN이다. 왼쪽은 인코더 네트워크로 출력층이 제거된 RNN이다. 모든 RNN은 입력 와 은닉상태 h_{t-1} 를 연결해서 출력 h_t 를 만들어 낸다. 오른쪽 디코더 네트워크는 인코더의 출력 h_t 와 이전 단계의 출력 y_{t-1} 를 입력으로 받는 RNN네트워크이다. RNN 셀은 성능문제로 주로 LSTM/GRU 셀들로 구성된 구조를 사용한다. 인코더는 입력문장을 단어 토큰화를 통해 단어단위로 쪼개고 단어 토큰 각각을 LSTM셀의 각 시점의 입력이 된다. 인코더 LSTM 셀은 모든 단어를 입력받은 뒤에 인코더 LSTM셀의 마지막 은닉상태를 디코더 RNN셀로

넘겨주는 데이터를 context vector라고 한다. context vector는 디코더 LSTM 셀의 첫 번째 은닉상태로 사용된다. 디코더의 첫 번째 LSTM 셀은 첫 번째 은닉 상태의 값과, 현재 t 에서의 입력 값으로 부터, 다음에 등장할 단어를 예측한다. 그리고 이 예측된 단어는 다음 시점인 $t+1$ RNN에서의 입력 값이 되고, 이 $t+1$ 에서의 RNN 또한 이 입력값과 t 에서의 은닉 상태로부터 $t+1$ 에서의 출력 벡터. 즉, 또 다시 다음에 등장할 단어를 예측하게 되는 것이다.

그런데 입력 데이터 길이가 길어지면 길어질수록 Sequence-to-Sequence 학습은 인코더가 생성한 context vector가 입력 데이터의 정보를 제대로 담기 어려워지는 문제가 발생한다. 이를 해결하기 위해 디코딩 과정에서 context vector를 동적으로 생성하는 Attention 기반 시퀀스 투 시퀀스 모델이 연구되었다.

기본적인 Sequence-to-Sequence는 디코더가 context vector로만 모든 출력을 생성했다면, Attention 기반 Sequence-to-Sequence는 디코더가 각 시퀀스마다 서로 다른 context vector를 활용한다. 이때 문맥 vector들은 인코더의 각 시퀀스에서 생성된 모든 은닉 상태의 가중치 합으로 나타난다.[9][10]

문장 길이가 길고 층이 깊으면 인코더가 길고 층이 깊으면, 인코더가 압축해야 할 정보가 너무 많아서 정보 손실이 일어나고, 디코더는 인코더가 압축한 정보를 초반 예측에만 사용하는 경향을 보인다. 이 때문에 인코더-디코더 사이에 병목현상이 일어난다. 이에 디코더 예측 시 가장 의미 있는 인코더 입력에 주목하게 만드는 attention mechanism이 제안되었다. 아울러 앞에서 뒤, 뒤에서 앞, 모두 고려하는 양방향 네트워크 또한 성능 개선에 도움이 된다.

Attention Mechanism은 네트워크가 출력 시퀀스의 각 항목에 대한 입력 시퀀스에서 주의를 기울여야 할 곳을 알 수 있게 해주는 제한 메커니즘을 극복하는 메커니즘이다. 따라서 텍스트 번역, 음성 인식 등과 같은 영역에서 반복적인 신경 네트워크를 이용한 Attention Mechanism이 많이 응용되고 있다. LSTM 인코더에 양방향 네트워크, LSTM 디코더에 어텐션 메커니즘이 적용된 Attention Bi-LSTM은 특히 기계번역에서 좋은 성능을 보여 주목을 받았다. 하지만 본 연구에서는 여기서 설명한 Attention 기반 시퀀스투시퀀스 구조를 사용하지 않고 텐서플로우에서 제공하는 Attention Module을 사용하였다.

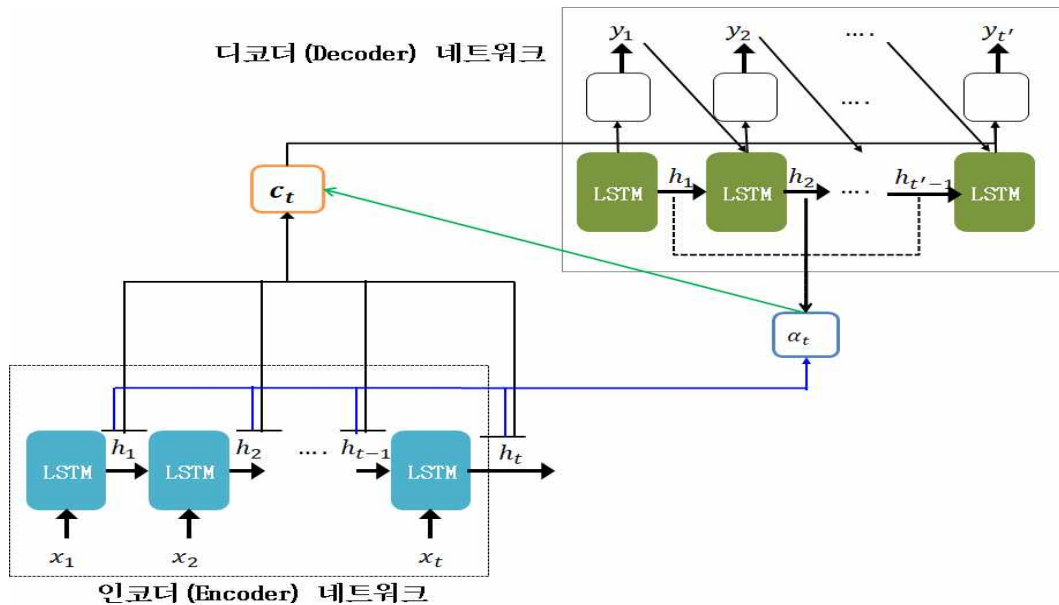


그림 2-15 Attention을 기반으로 하는 시퀀스투시퀀스 모델

6. Bollinger Band

1980년대 존볼린저(John Bollinger)가 가격이 급변하는 시점에서 Envelop 지표가 적절한 매매 시점을 알려주지 못한다는 단점을 보완하기 위해 고안해낸 변동미를 탄력적으로 변화시켜 만든 대표적인 변동성 지표이다.[1]

볼린저 밴드는 통계적 기법을 이용하여 주가의 변동 가능 범위를 나타내는 지표 중 하나로 주가가 수렴과 확장을 반복하기 때문에 볼린저 밴드 역시 수렴과 확장을 반복하게 설계된다. 볼린저 밴드는 변동성 분석과 추세 분석을 동시 가능하기 때문에 상한선과 하한선 사이에서 주가가 오르내린다는 이론을 토대로 주가가 밴드 상한선에 도달하면 과열되었으니 하락할 수 있다고 판단하여 매도 신호를 나타내고, 밴드 하한선에 도달하면 침체되어 있으니 앞으로 상승할 수 있다고 판단하여 매수 신호라는 의미를 갖게 된다.

볼린저 밴드 지표는 3개의 선으로 구성되는데, 상한선과 하한선 사이의 밴드 폭이 중요하다. 이 두 선 사이의 밴드 폭이 좁아지며 수렴할 경우 향후 상승이나 하락이 올 수 있음을 주의해야 한다.

표 2-1 볼린저 밴드 지표 계산식

밴드 지표	계산 공식
중심선	종가의 20일 이동평균선
상한선	중심선 + (표준편차 * 2)
하한선	중심선 - (표준편차 * 2)
볼린저 밴드 폭	(상한선 - 하한선) / 중심선
밴드내의 주가의 위치	(종가 - 하한선) / (상한선 - 하한선)

본 연구에서는 볼린저 밴드 지표의 중심선, 상한선, 하한선, 밴드폭, 밴드내의 종가의 위치를 입력 피처로 추가하여 사용하였다.

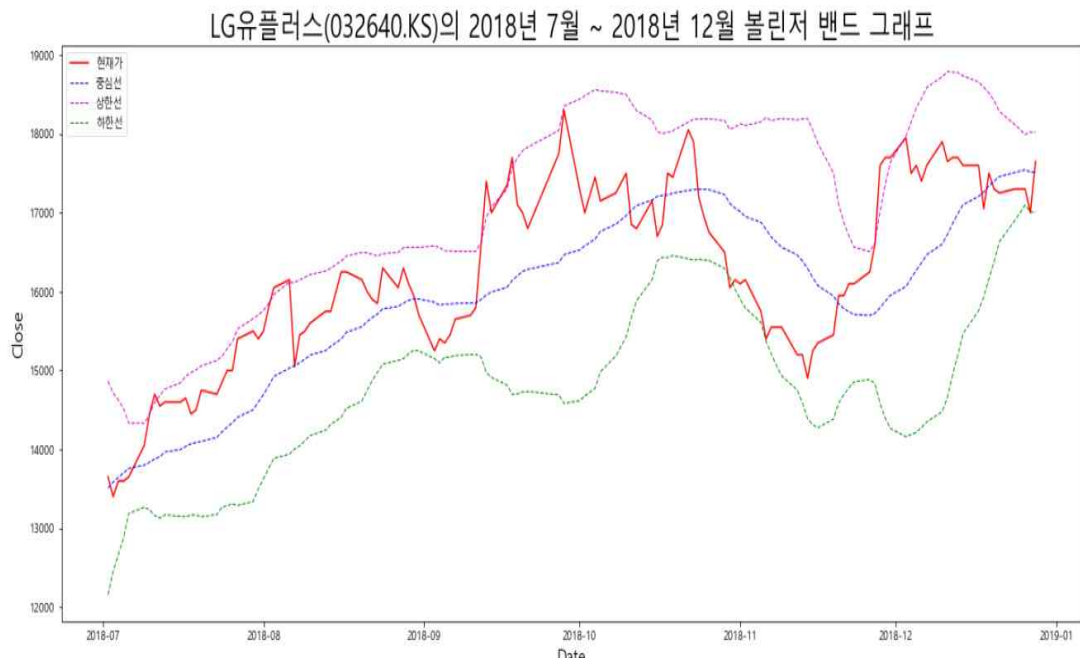


그림 2-16 LG유플러스의 현재가와 볼린저 밴드 그래프

III. Attention Mechanism 기반 Bi-LSTM을 이용한 주가 예측 모델

본 장에서 주가 예측 모델로 Attention Mechanism을 기반으로 하는 Bi-LSTM 모델을 제안하다. 본 연구에서 제안한 Attention 기반 Bi-LSTM 모델은 TensorFlow에서 제공하는 attention 계층을 사용하였다. 각 시퀀스의 가중치 정보를 가진 context vector를 사용해 학습하는 모델로 각각의 입력 시퀀스에서 중요도가 높은 입력된 주식 가격에 많은 가중치를 주는 모델이다.

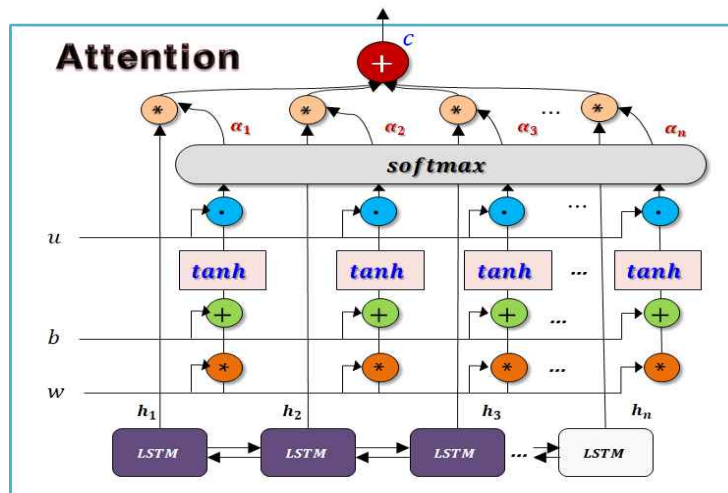


그림 3-1 Attention Layer 구조(TensorFlow 제공)

<그림 3-1>에서와 같이 TensorFlow에서 제공되는 Attention Layer는 Attention은 간단한 vector이며, softmax 기능을 사용하는 밀집된 layer의 출력이다.

Attention 기반 LSTM 신경망은 각 시퀀스의 hidden layer 출력의 중요도인 attention weight를 각 hidden layer 출력에 가중치 합으로 계산하여 attention context vector를 구한다. Attention의 context vector를 구하기 위한 가중치는 학습을 통해 얻으며 수식은 다음과 같다. [26]

$$[\text{Attention vector}] \quad U \quad V_a \tanh(W_a h_t + b_a) \quad \text{수식 11}$$

$$[\text{Attention weights}] \quad \alpha_t = \frac{\exp(VU_t)}{\sum_{k=1}^T \exp(VU_k)} \quad \text{수식 12}$$

$$[\text{Context vector}] \quad c = \sum_{t=1}^{T_x} \alpha_t h_t \quad \text{수식 13}$$

V_a 는 attention을 구하기 위한 가중치 vector이고, h_t 는 수식에서 구한 각 시퀀스의 hidden layer의 출력을 나타낸 vector(attention을 호출할 때 넘겨받은 양방향 hidden layer)이고, W_a 는 attention을 구하기 위해 학습되는 가중치 행렬이고, b_a 는 bias이다.

α_t 는 attention 각 h_t 에 대해 가중치 값이 된다. c 는 attention vector를 구할 때 각 h_t 에 대해 가중치 값이 된다. α_t 는 각 h_t 에 가중치로 곱하여 그 합으로 attention vector를 구하고 주식 가격 항목 내에서 결과에 영향을 크게 미치는 h_t 를 강조하게 된다.

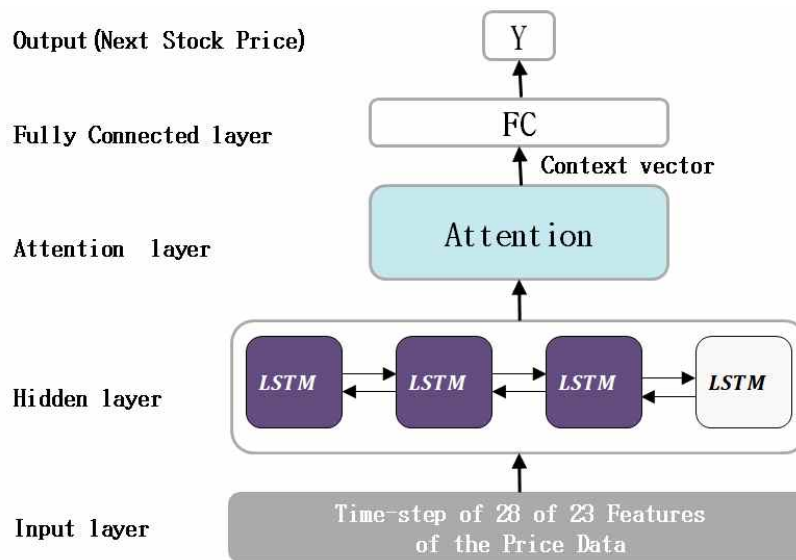


그림 3-2 Attention Bi-LSTM 모델 기본 설계

<그림 3-2>과 같이 제안한 모델의 기본 설계는 주식 가격 데이터를 입력 데이터로 받아들이어 양방향 LSTM을 적용한 후 hidden layer의 출력을 다시 Attention layer의 입력으로 사용하여 결과에 가장 영향력이 큰 hidden

layer을 집중 학습시키기 위해 Attention Layer를 통과한 후 완전연결신경망을 통해 다음날의 종가를 예측하는 모델이다.

1. Attention Bi-LSTM-Only Model

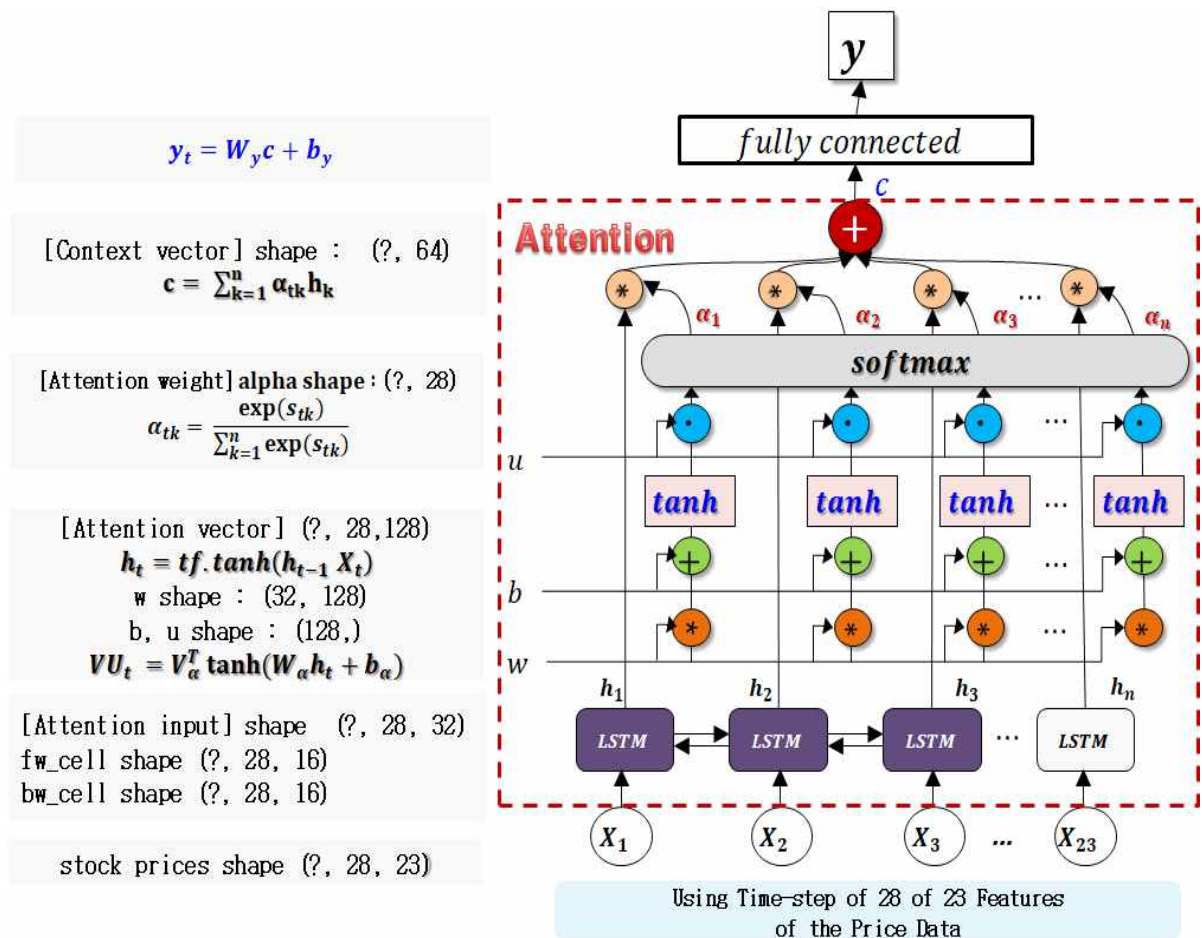


그림 3-3 Attention context vector로만 학습한 모델

첫 번째 실험은 <그림 3-3>에서와 같이 각 시퀀스의 가중치 정보를 가진 attention 모듈에서 얻은 결과인 context vector로 사용해 학습하는 모델로 각각의 입력 시퀀스에서 중요도가 높은 입력변수에 많은 가중치를 가진 기본 모델이다.

주식 가격 데이터를 (?, 28, 23) 형태로 받아들여 양방향 LSTM cell을 적용한 후 hidden layer의 출력을 다시 Attention layer의 입력으로 받아 들인 후 가중치들을 곱한 뒤 tanh함수로 -1 ~ 1사이의 수로 변환하여

attention vector를 구한다.

그리고 u 를 곱해 softmax계산으로 확률을 구해 Attention weight을 구한다. Attention weight과 입력으로 받아 들였던 양방향 LSTM cell을 적용한 hidden layer의 출력을 곱하여 합을 구한 context vector와 hidden layer의 마지막 시퀀스 출력을 벡터 곱을 한 후 완전연결망을 통과하여 다음 날의 종가를 출력 값으로 예측하는 모델이다.

$$= {}_y c + b_y \quad \text{수식 14}$$

2. Attention Bi-LSTM-Concat Model

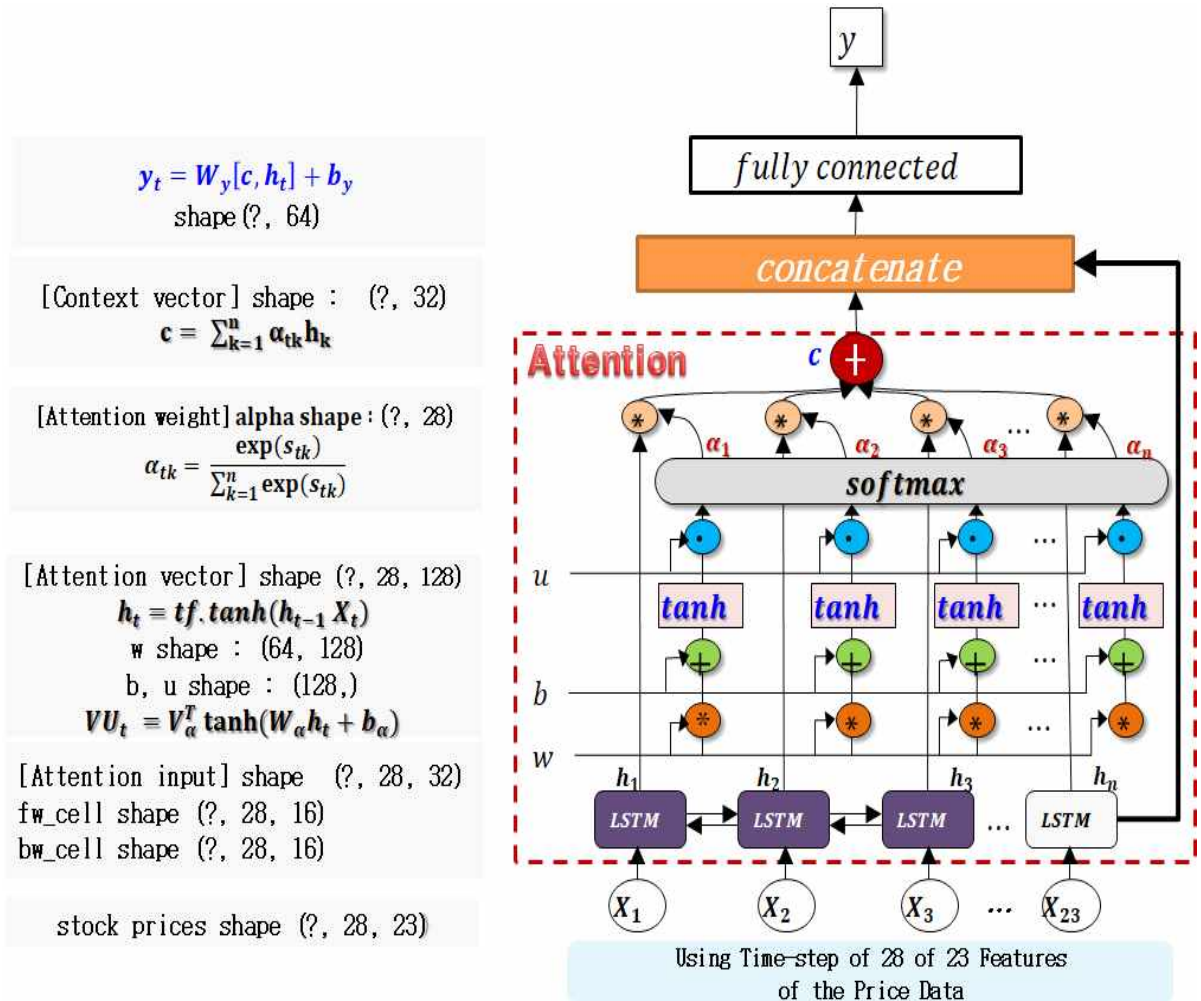


그림 3-4 Attention context vector와 hidden layer의 마지막 시퀀스를 연결하여 학습한 모델

두 번째 실험은 <그림 3-4>에서와 같이 첫 번째 실험과 모두 동일하게 진행하여 context vector를 구한 후 양방향 LSTM hidden layer의 마지막 시퀀스 출력 vector를 더 많은 정보를 담기 위해 연결하여 학습시키는 모델이다.

$$= y[c, h_t] + b_y \quad \text{수식 15}$$

3. Attention Bi-LSTM-Dot Model

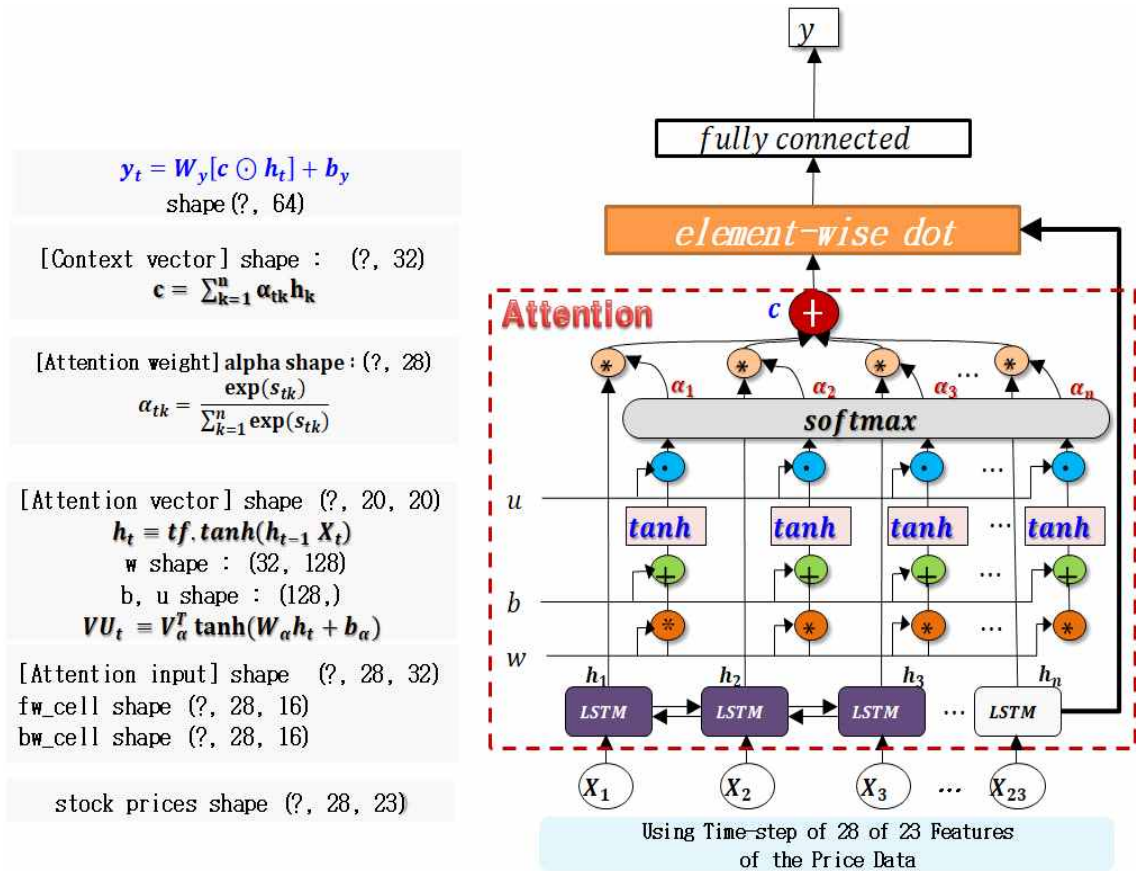


그림 35 Attention context vector와 hidden layer의 마지막 시퀀스의 요소간 곱을 하여 학습한 모델

$$y_t = W_y[c \ h_t] + b_y \quad \text{수식 16}$$

세 번째 실험은 <그림 3-5>에서와 같이 첫 번째 실험과 모두 동일하게 진행된 후 양방향 LSTM hidden layer의 마지막 시퀀스 출력 vector와 attention output의 차원이 동일하므로 두 vector를 요소 간 벡터 곱을 하여 정보를 섞은 후 학습하는 모델이다.

IV. 실험 및 성능 평가

본 연구에서 정확한 주식 가격 예측을 위한 방법으로 제안한 모델 Attention Bi-LSTM Recurrent Neural Network를 이용하여 세 가지 방법으로 실험을 진행하였다.

1. 실험 방법

본 연구에서 주식 가격 예측을 위해 제안한 신경망 모델은 Attention Bi-LSTM Recurrent Neural Network이다. 네트워크 전체적인 구조는 일자별로 주식의 시가, 고가, 저가, 종가, 거래량, 수정종가, 볼린저 밴드 지표인 종가의 5일, 20일 이동평균을 이용한 중심선, 상한선, 하한선, 밴드 폭, 밴드내의 종가의 위치, 금일 가격 항목과 종가, 거래량의 비율 등 입력을 23개로 하여 LSTM 셀로 구성되었다. Attention Bi-LSTM 순환신경망으로 hidden layer 1개와 Attention layer 1개와 완전연결망 1개와 주식 가격 예측을 위한 출력 layer로 구성한다.

첫 번째 실험은 Attention 은닉층을 이용하여 context vector를 그대로 학습하는 모델로 실험하고, 두 번째 실험은 Attention layer를 이용하여 context vector를 hidden layer의 LSTM 마지막 시퀀스와 연결하여 더 많은 정보를 담아 학습하는 모델로 실험하고, 세 번째 실험은 Attention layer를 이용하여 context vector를 hidden layer의 LSTM 마지막 시퀀스와 요소 간 곱을 하여 정보를 혼합하여 학습하는 모델로 실험하였다.[7]

2. 실험 데이터

본 연구에서는 시스템 환경은 윈도우 10 환경에서 Intel(R) Core(TM) i7-7700HQ @ 2.80GHZ, Ram 16GB, NVIDIA Geforce GTX 1060 환경에서 Anaconda3(Python 3.7), 텐서플로우 v.1.14.0를 사용하였다.

주가 예측 실험에 사용된 데이터는 야후 파이낸스(Yahoo Finance)에서 KOSPI 종목 중 무선 이동통신사 LG유플러스(032640.KS), KT(030200.KS), SK텔레콤(017670.KS) 종목으로 정하였고 세 종목의 2014/01/02 ~ 2018/12/28까지의 5년간의 일간 주식 가격 데이터를 기본으로 Open(시가), High(고가), Low(저가), Close(종가), Volume(거래량), Adj Close(수정종가, 주식의 분

할, 배당, 배분 등을 고려해 조정한 증가)가 제공된다. 그리고 시가, 고가, 저가, 수정증가, 종가의 5일 이동평균, 종가의 20일 이동평균, 20일 이동평균의 상한밴드, 하한밴드, 그리고 중간선인 종가가布林저 밴드[1] 범위를 크게 벗어나지 않고 제자리로 돌아오는 원리를 이용해서 %b(밴드내의 종가의 위치), 20일 이동평균 밴드 폭, 당일증가대비 당일시가비율, 당일증가대비 당일고가비율, 당일증가대비 당일저가비율, 당일증가대비 전일증가비율, 당일증가대비 당일거래량비율, 20일평균증가 대비 당일증가비율, 20일 거래량이동평균 대비 당일거래량비율, 거래량, 5일 거래량이동평균, 20일 거래량 이동평균 등 보조 데이터로 17개 추가하여 형성된 23개의 입력 피쳐로 추가된 형태로 2019년 1월 2일의 종가를 예측하면서 정확도를 확인하기 위해 평균 제곱근 오차(RMSE)를 통해 오차의 최소값, 최대값, 평균을 구하는 실험을 진행하였다.

표 4-1 입력 데이터 구성

Total	train data	test data
LG유플러스, KT, SK텔레콤 종목당 5년	70%	30%
1230개	847개	363개

	Date	High	Low	Open	Close	Volume	Adj Close
0	2014-01-02	11000.0	10600.0	10800.0	10900.0	1434392.0	9686.131836
1	2014-01-03	11000.0	10650.0	10900.0	10700.0	1564299.0	9508.403320
2	2014-01-06	10800.0	10550.0	10700.0	10700.0	1383894.0	9508.403320
3	2014-01-07	10700.0	10500.0	10550.0	10500.0	1321450.0	9330.676758
4	2014-01-08	10650.0	10350.0	10600.0	10500.0	1306000.0	9330.676758

그림 4-1 U+(032640.KS) 종목의 다운로드 데이터

	MA5	MA20	close_ma20_upbb	close_ma20_dobb	MA20_pb	MA20_bwidth	open_lastclose_ratio	high_close_ratio	low_close_ratio
1244	17340.0	17457.5	18284.208757	16630.791243	0.374502	0.094711	0.000000	0.017391	-0.005797
1245	17280.0	17510.0	18118.362773	16901.637227	0.327406	0.069487	0.005797	0.017341	-0.005780
1246	17330.0	17545.0	17992.095892	17097.904108	0.226010	0.050966	-0.005780	0.005780	-0.011561
1247	17230.0	17515.0	18022.937004	17007.062996	-0.006953	0.058000	-0.020231	0.023529	-0.008824
1248	17300.0	17512.5	18017.084248	17007.915752	0.636251	0.057626	0.000000	0.008499	-0.036827

그림 4-2 U+(032640.KS) 종목의布林저 밴드 지표 추가한 항목

close_lastclose_ratio	volume_lastvolume_ratio	close_ma5_ratio	close_ma20_ratio	volume_ma5_ratio	volume_ma20_ratio	volume_ma5	volume_ma20
-0.002890	0.138448	-0.005190	-0.011886	-0.125285	-0.469280	1563555.8	2576999.55
0.002899	-0.366315	0.001157	-0.011993	-0.454339	-0.659324	1588290.6	2543970.25
0.000000	0.212712	-0.001731	-0.013964	-0.187304	-0.577766	1293250.4	2489186.85
-0.017341	0.117793	-0.013349	-0.029403	0.037551	-0.490813	1132304.2	2307253.55
0.038235	0.259814	0.020231	0.007852	0.245791	-0.352155	1188047.2	2284587.90

그림 4-3 U+(032640.KS) 종목의 각 항목과의 비율 계산 추가한 항목

전체 주식 가격 데이터는 각 종목당 1230개 데이터에서 70%인 847개를 학습용 데이터로, 나머지 30%인 363개를 테스트용 데이터로 이용하여 주식 가격 데이터 전처리는 가격과 거래량이 크기 차이가 많이 나고, 추가된 항목들 중 실수값인 비율들을 분리해서 주식 가격 데이터를 0과 1사이의 값으로 max-min scaling(최대-최소) 정규화한 후 다시 연결하여 전체 데이터를 훈련용 데이터셋, 테스트용 데이터셋으로 나누어 구성하고 오차를 줄여서 예측 정확도를 높일 수 있도록 종목별 적합한 학습 파라미터를 찾아내었다.

$$= \frac{d_i - \min(d)}{\max(d) - \min(d)} \quad \text{주식 17}$$

본 연구에서 실험데이터로 입력한 항목들은 다음과 같다.

표 4-2 입력 데이터 항목 계산식(1)

입력 변수	Detail of the feature		
Simple data (6개 항목)	$en\ price^k$	Moving Average (4개 항목)	$A^k = \frac{1}{5} \sum_{i=0}^4 Close\ price_{t-i}^k$
	$High\ price_t^k$		
	$Low\ price_t^k$		$MA20_t^k = \frac{1}{20} \sum_{i=0}^{19} Close\ price_{t-i}^k$
	$Adj\ Close\ price_t^k$		$VMA5_t^i = \frac{1}{5} \sum_{i=0}^4 Volume_{t-i}^k$
	$Volume\ price_t^k$		
	$Close\ price_t^k$		$VMA20_t^i = \frac{1}{20} \sum_{i=0}^{19} Volume_{t-i}^k$
	Bollinger Band 지표 (4개 항목)		$SA20_t^k = stdd(MA20_t^k)$ $Upbb_t^k = MA20_t^k + (SA20_t^k * 2)$
$Dobb_t^k = MA20_t^k - (SA20_t^k * 2)$			
$BW_t^k = (Upbb_t^k - Dobb_t^k) / MA20_t^k$			
$PB_t^k = (MA20_t^k - Dobb_t^k) / (Upbb_t^k - Dobb_t^k)$			

표 4-3 입력 데이터 항목 계산식(2)

Ratio data (9개 항목)	$enLastCloseRatio_t^k = (Open\ price_t^k - Close\ price_{t-1}^k) / Close\ price_{t-1}^k$ 전일 증가 대비 당일 시가 비율
	$HighLastCloseRatio_t^k = (High\ price_t^k - Close\ price_t^k) / Close\ price_t^k$ 당일 증가 대비 당일 고가 비율
	$LowLastCloseRatio_t^k = (Low\ price_t^k - Close\ price_t^k) / Close\ price_t^k$ 당일 증가 대비 당일 저가 비율
	$CloseLastCloseRatio_t^k = (Close\ price_t^k - Close\ price_{t-1}^k) / Close\ price_{t-1}^k$ 당일 증가 대비 전일 증가 비율
	$VolumeLastVolumeRatio_t^k = (Volume_t^k - Volume_{t-1}^k) / Volume_{t-1}^k$ 당일 거래량 대비 전일 거래량 비율
	$Close\ priceMA5Ratio_t^k = (Close\ price_t^k - MA5_t^k) / MA5_t^k$ 5일 증가 이동평균 대비 당일 증가 비율
	$Close\ priceMA20Ratio_t^k = (Close\ price_t^k - MA20_t^k) / MA20_t^k$ 20일 증가 이동평균 대비 당일 증가 비율
	$VolumeVMA5Ratio_t^k = (Volume_t^k - VMA5_t^k) / VMA5_t^k$ 5일 거래량 이동평균 대비 당일 거래량 비율
	$VolumeVMA20Ratio_t^k = (Volume_t^k - VMA20_t^k) / VMA20_t^k$ 20일 거래량 이동평균 대비 당일 거래량 비율

네트워크 전체적인 구조는 입력 데이터는 주식의 일별 시가, 고가, 저가, 증가, 거래량, 수정증가와 예측 보조 지표 17개의 항목을 추가하여 23개의 항목으로 구성하였으며 Attention Bi-RNN LSTM 모델 세 가지 방법으로 sequence_length, 은닉 layer Size, attention_size는 종목마다 주가 변동 패턴이 모두 다르므로 별도로 측정하여 실험하였다.

표 4-4 종목별 학습 파라미터

	LG U+ (032640.KS)	KT (030200.KS)	SK 텔레콤 (017670.KS)
seq_length	28	28	28
hidden layer	16	10	10
Attention_size	128	40	128
learning_rate	0.01	0.01	0.01
학습 횟수	1,000회	1,000회	1,000회

LSTM의 활성화 함수로는 Tanh(Hyperbolic Tangent)함수를, 최적화 알고리

즘으로는 오차 감소도가 빠른 AdamOptimizer 알고리즘을, 학습률은 0.01, 학습 횟수는 1,000회로 실험하였다.

종목마다 적합한 LSTM셀로 구성된 hidden layer과 Attention layer로 주가 예측을 위한 1개 출력 계층이 구성되어 있고 과거와 현재의 주가, 이동 평균 데이터를 이용하여 학습하고 미래의 주가를 예측하고, 실제의 주식 가격과 예측 가격 간의 정확도를 측정하기 위해서는 평균 제곱근 오차(Root Mean Square Error)로 오차를 측정하였다.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (targets - predict)^2} \quad \text{수식 18}$$

3. 실험 결과

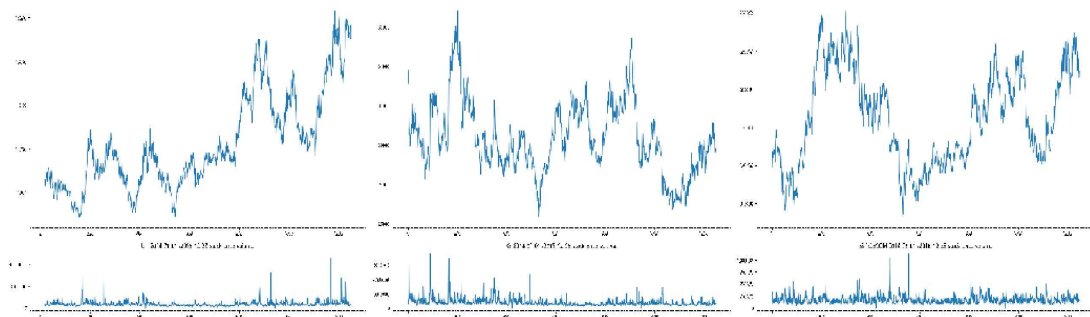


그림 4-4 이동통신 3사의 5년간 주가와 거래량

<그림 4-4>처럼 이동통신 3사의 2014년~2018년까지 5년간 주가와 거래량의 흐름을 보면 모두 상승하는 것처럼 보이는데 LG유플러스는 2018년에 거래량도 늘면서 안정적으로 계속 상승해가는 패턴을 보인다.

양방향 순환신경망과 Attention mechanism을 기반으로 한 양방향 순환신경망 모델로 입력 시퀀스 데이터의 시퀀스 길이를 5일, 10일, 20일, 28일 단위로, hidden layer는 5, 10, 16, 32, 64로, LSTM을 적용한 Attention mechanism을 기반으로 한 양방향 순환신경망 모델의 attention size는 5, 10, 16, 20, 32, 40, 64, 128로 변화시켜가며 학습에서 발생하는 평균 제곱근 오차를 통해 가장 적합한 파라미터를 찾기 위한 실험을 진행하였다.

(1) LG U+ 종목에 적합한 학습 파라미터

1) 적절한 Sequence Length / Hidden Layer 분석

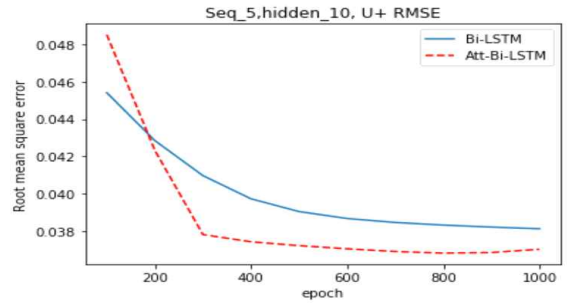
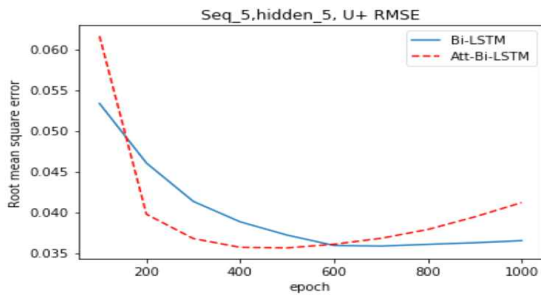


그림 4-5 U+ seq5, hidden5 rmse비교 그림 4-6 U+ seq5, hidden10 rmse비교

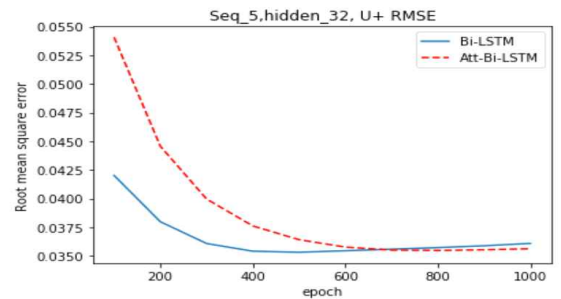
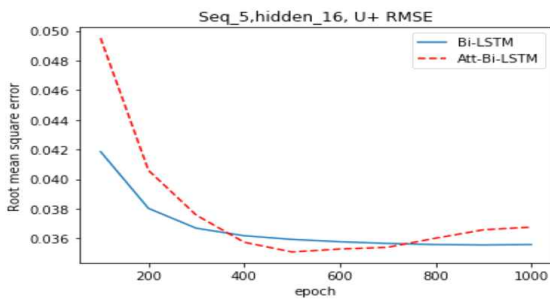


그림 4-7 U+ seq5, hidden16 rmse비교 그림 4-8 U+ seq5, hidden32 rmse비교

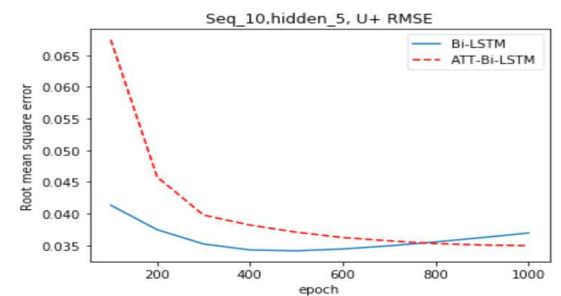
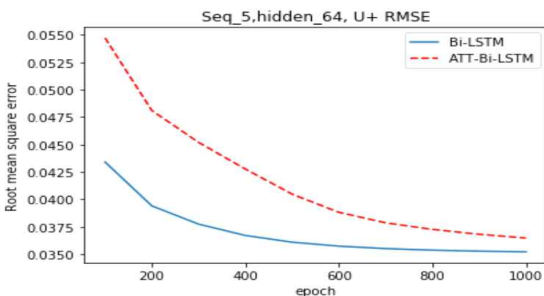


그림 4-9 U+ seq5, hidden64 rmse비교 그림 4-10 U+ seq10, hidden5 rmse비교

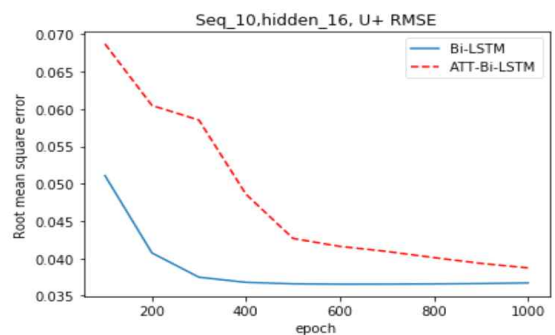
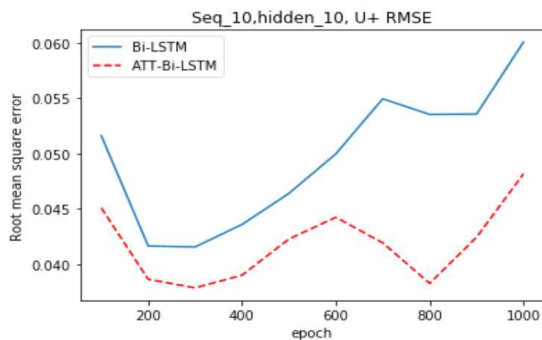


그림 4-11 U+ seq10, hidden10 rmse비교 그림 4-12 U+ seq10, hidden16 rmse비교

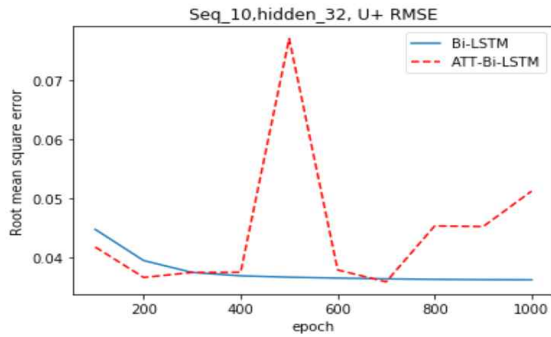


그림 4-13 U+ seq10, hidden32 rmse비교

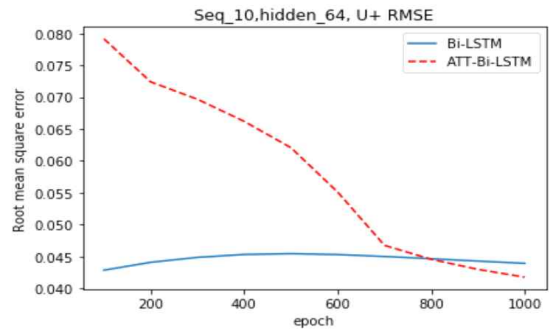


그림 4-14 U+ seq10, hidden64 rmse비교

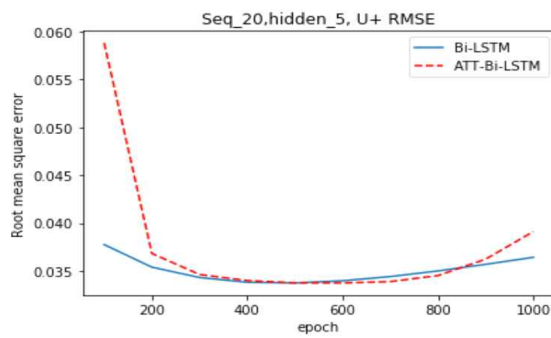


그림 4-15 U+ seq20, hidden5 rmse비교

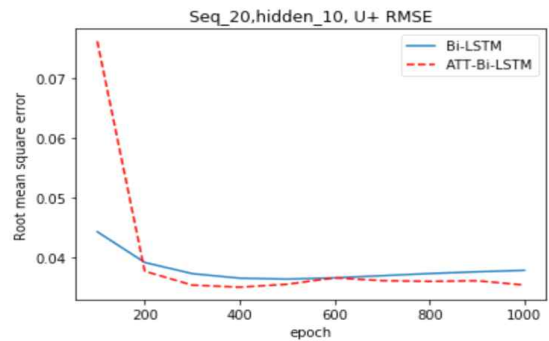


그림 4-16 U+ seq20, hidden10 rmse비교

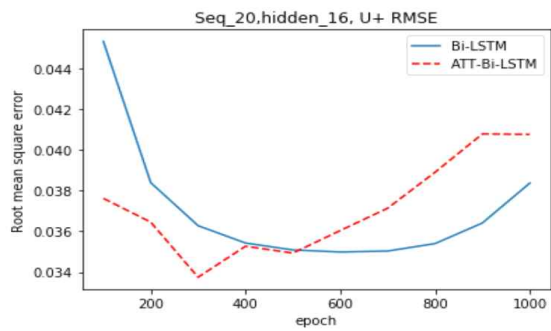


그림 4-17 U+ seq20, hidden16 rmse비교

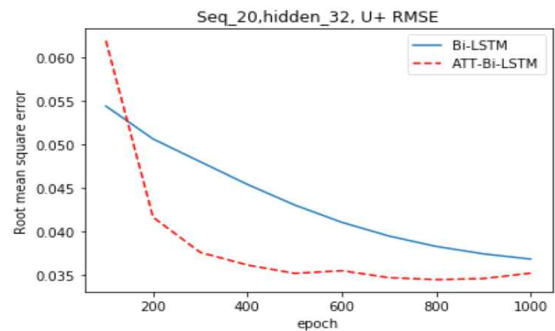


그림 4-18 U+ seq20, hidden32 rmse비교

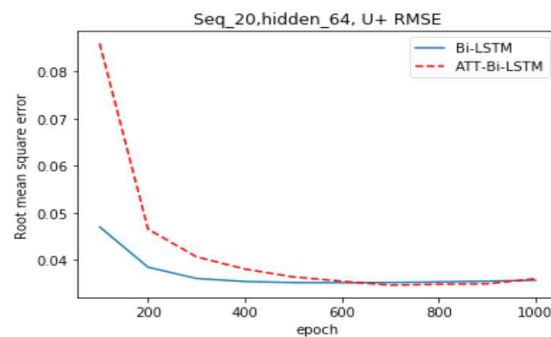


그림 4-19 U+ seq20, hidden64 rmse비교

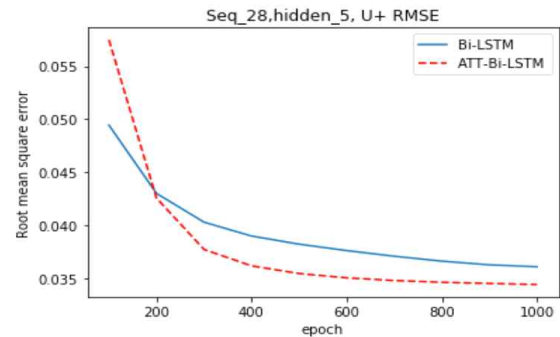


그림 4-20 U+ seq28, hidden5 rmse비교

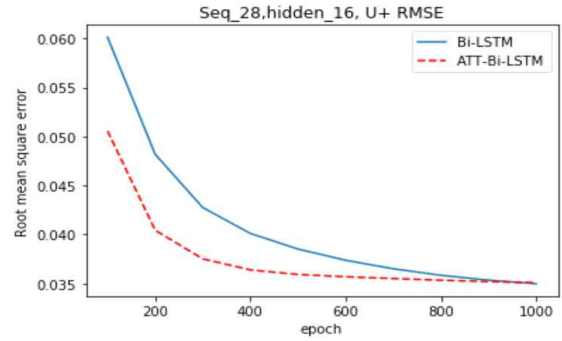
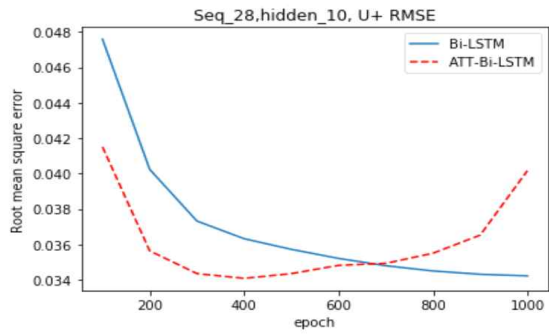


그림 4-21 U+ seq28, hidden10 rmse비교 그림 4-22 U+ seq28, hidden16 rmse비교

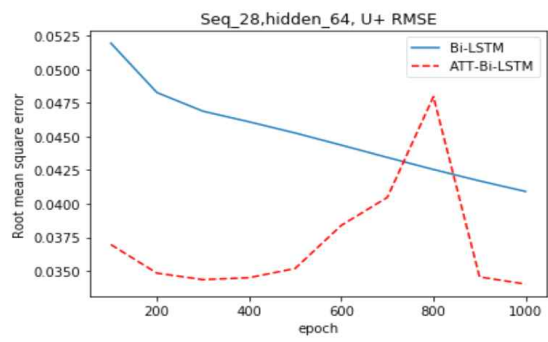
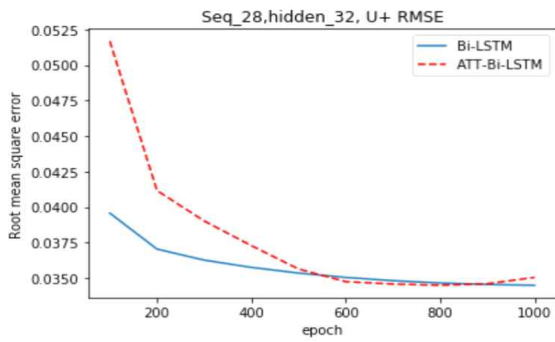


그림 4-23 U+ seq28, hidden32 rmse비교 그림 4-24 U+ seq28, hidden64 rmse비교

표 4-5 U+ Bi-LSTM을 이용한 시퀀스 길이와 hidden layer 크기에 따른 RMSE 비교

input	seq_length	hidden_dim	Training			test		
			Min	Max	Avg	Min	Max	Avg
23	5	5	0.025981	0.034699	0.029230	0.040559	0.056785	0.046835
23	5	10	0.023627	0.034809	0.027137	0.037183	0.055113	0.042260
23	5	16	0.024544	0.034837	0.027530	0.038116	0.055994	0.043372
23	5	32	0.025691	0.041212	0.030241	0.042310	0.062693	0.048275
23	5	64	0.026732	0.057738	0.034682	0.040625	0.114633	0.057690
최소			0.023627	0.034699	0.027137	0.037183	0.055113	0.042260
평균			0.025315	0.040659	0.029764	0.039759	0.069044	0.047686
23	10	5	0.024013	0.036021	0.026826	0.051996	0.074550	0.060008
23	10	10	0.023087	0.034375	0.025191	0.045011	0.066148	0.050550
23	10	16	0.024798	0.042924	0.028433	0.035796	0.068786	0.041343
23	10	32	0.024549	0.057832	0.031441	0.038501	0.087059	0.049944
23	10	64	0.024866	0.048923	0.030383	0.036578	0.077557	0.045512
최소			0.023087	0.034375	0.025191	0.035796	0.066148	0.041343
평균			0.024263	0.044015	0.028455	0.041576	0.074820	0.049471
23	20	5	0.023817	0.032469	0.026208	0.036096	0.055151	0.042774
23	20	10	0.024817	0.039192	0.028432	0.039323	0.051408	0.041574
23	20	16	0.025721	0.029717	0.026966	0.041075	0.048331	0.043300
23	20	32	0.023808	0.036327	0.026113	0.036160	0.060681	0.040467
23	20	64	0.031000	0.075832	0.044173	0.038503	0.112458	0.055750
최소			0.023808	0.029717	0.026113	0.036096	0.048331	0.039774
평균			0.025833	0.042707	0.030378	0.038231	0.065606	0.044173
23	28	5	0.023342	0.031883	0.025551	0.041529	0.062633	0.045143
23	28	10	0.024844	0.040332	0.028148	0.044526	0.060606	0.047372
23	28	16	0.023976	0.030528	0.026127	0.040538	0.048601	0.042220
23	28	32	0.024558	0.041975	0.028336	0.041605	0.063738	0.046988
23	28	64	0.025526	0.042572	0.030139	0.040026	0.066424	0.050969
최소			0.023342	0.030528	0.025551	0.040026	0.048601	0.042220
평균			0.024449	0.037458	0.027660	0.041645	0.060400	0.046538

<표 4-5>, <그림 4-5>~<그림 4-24>를 보면 U+의 주식 가격 예측 모델의 시퀀스 길이가 28이면서 hidden layer 크기가 16일 때 가장 오차가 작은 상태임으로 판단되어 Bi-LSTM과 Attention Bi-LSTM 모델에 적용하였다.

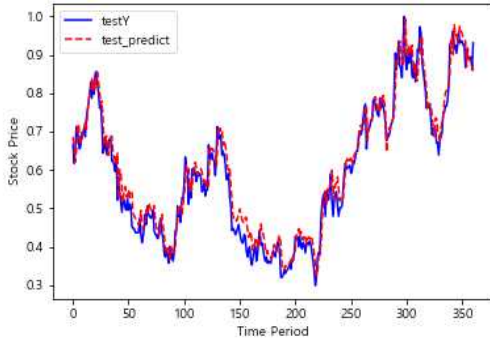
1) 적절한 Attention Size 분석

하지만 Attention Mechanism을 기반으로 한 양방향 순환신경망 모델은 attention_size라는 파라미터를 하나 더 측정해야 한다. Attention mechanism을 기반으로 한 양방향 순환신경망 모델의 시퀀스 길이가 28, hidden layer크기가 16 일 때, attention_size를 구하기 위해 5, 10, 16, 20, 32, 40, 64, 128로 변화시키며 평균 제곱근 오차(RMSE)를 측정하였다.

표 4-6 U+의 입력 23, seq 28, hidden 16, attention_size변화에 따른 RMSE비교

seq	hidden	attention_size	training_RMSE			test_RMSE		
			Min	MAx	Avg	Min	MAx	Avg
28	16	5	0.024954	0.053166	0.034045	0.036241	0.084595	0.051267
28	16	10	0.0235	0.029943	0.02466	0.035026	0.042457	0.037317
28	16	16	0.024629	0.05768	0.032126	0.035878	0.095553	0.050398
28	16	20	0.02375	0.052899	0.028747	0.035242	0.089187	0.04237
28	16	32	0.02394	0.061171	0.029562	0.036403	0.124967	0.049598
28	16	40	0.023811	0.062856	0.029064	0.035286	0.112177	0.045014
28	16	64	0.026386	0.058450	0.037272	0.047442	0.130182	0.075622
28	16	128	0.025432	0.040096	0.027678	0.033539	0.045252	0.035366

U+ Attention_Bi-LSTM_seq_28_hidden_16_attention_5 Targets & Predict



U+ Attention_Bi-LSTM_seq_28_hidden_16_attention_10 Targets & Predict

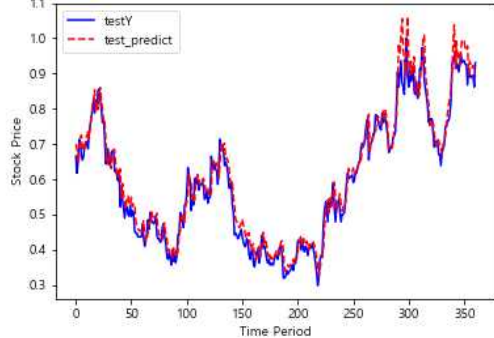


그림 4-25 U+ Att_Bi-LSTM seq 28, hidden 16, attention 5

그림 4-26 U+ Att_Bi-LSTM seq 28, hidden 16, attention 10

U+ Attention_Bi-LSTM_seq_28_hidden_16_attention_16 Targets & Predict

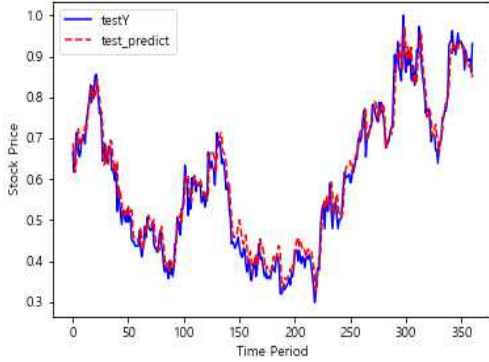


그림 4-27 U+ Att_Bi-LSTMonly seq 28, hidden 16, attention 16

U+ Attention_Bi-LSTM_seq_28_hidden_16_attention_20 Targets & Predict

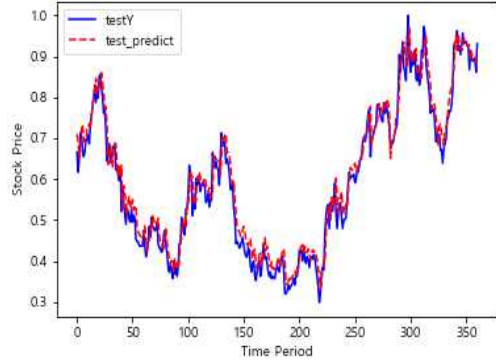


그림 4-28 U+ Att_Bi-LSTMonly seq 28, hidden 16, attention 20

U+ Attention_Bi-LSTM_seq_28_hidden_16_attention_32 Targets & Predict

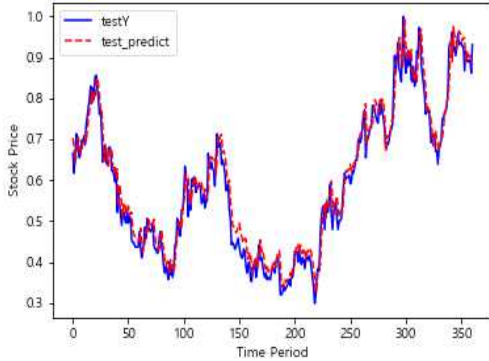


그림 4-29 U+ Att_Bi-LSTM seq 28, hidden 16, attention 32

U+ Attention_Bi-LSTM_seq_28_hidden_16_attention_40 Targets & Predict

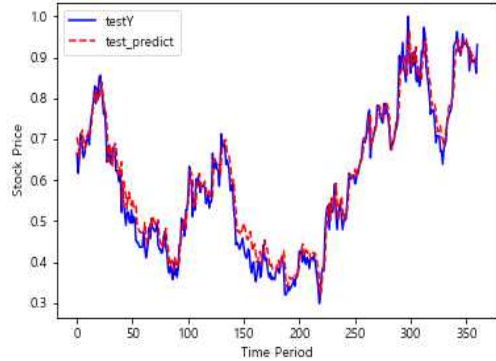


그림 4-30 U+ Att_Bi-LSTM seq 28, hidden 16, attention 40

U+ Attention_Bi-LSTM_seq_28_hidden_16_attention_64 Targets & Predict

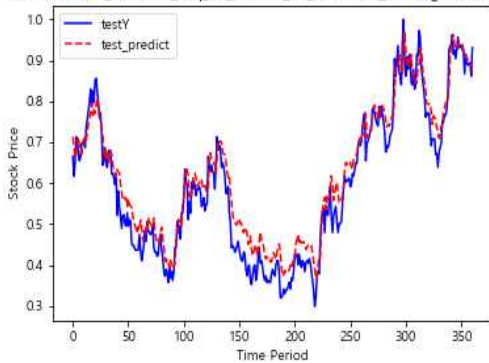


그림 4-31 U+ Att_Bi-LSTM seq 28, hidden 16, attention 64

U+ Attention_Bi-LSTM_seq_28_hidden_16_attention_128 Targets & Predict

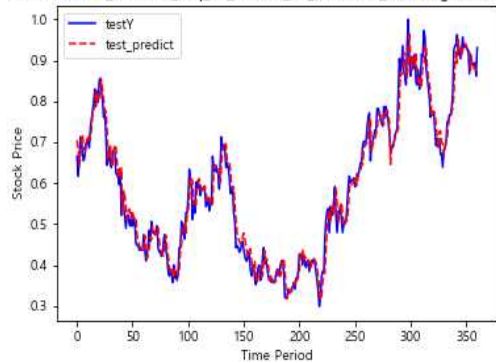


그림 4-32 U+ Att_Bi-LSTM seq 28, hidden 16, attention 128

<표 4-6>와 <그림 4-25>~<그림 4-32>에 따라 U+ 종목은 시퀀스 길이를 28, hidden layer 크기 16, Attention size는 128로 결정하였다. 그리고 5회 반복 수행하여 Attention-Bi-LSTM-Only 모델과 Attention-Bi-LSTM-Concat 모델과

Attention-Bi-LSTM-Dot 모델의 RMSE의 최소값, 최대값, 평균을 측정하였다.

표 4-7 U+ Bi-LSTM과 Attention-Bi-LSTM 모델별 평균 RMSE 비교

LG U+		training_RMSE			test_RMSE		
		Min	Max	Avg	Min	Max	Avg
1 회차	Bi-LSTM	0.022905	0.028041	0.023850	0.044884	0.066585	0.051523
	Att-Bi-LSTM-Only	0.026978	0.072979	0.035241	0.040171	0.094388	0.051011
	Att-Bi-LSTM-Concat	0.024386	0.032767	0.026552	0.045931	0.054002	0.047828
	Att-Bi-LSTM-Dot	0.024230	0.028799	0.025354	0.032954	0.036386	0.033705
2 회차	Bi-LSTM	0.022758	0.030856	0.024433	0.046362	0.059880	0.052653
	Att-Bi-LSTM-Only	0.026447	0.039178	0.029119	0.037080	0.063898	0.042985
	Att-Bi-LSTM-Concat	0.024163	0.032143	0.025933	0.035341	0.050748	0.038028
	Att-Bi-LSTM-Dot	0.024879	0.031137	0.026022	0.033194	0.040330	0.035210
3 회차	Bi-LSTM	0.022896	0.027758	0.024032	0.051577	0.070376	0.056783
	Att-Bi-LSTM-Only	0.026191	0.070898	0.032453	0.036083	0.115496	0.045659
	Att-Bi-LSTM-Concat	0.025287	0.045631	0.030096	0.039324	0.075964	0.050453
	Att-Bi-LSTM-Dot	0.025110	0.030051	0.025919	0.033452	0.040376	0.034475
4 회차	Bi-LSTM	0.023392	0.031206	0.025472	0.064404	0.096332	0.080380
	Att-Bi-LSTM-Only	0.026499	0.076791	0.036426	0.038835	0.115742	0.057188
	Att-Bi-LSTM-Concat	0.023887	0.041759	0.027631	0.035026	0.072571	0.042195
	Att-Bi-LSTM-Dot	0.024771	0.029887	0.025709	0.034113	0.042981	0.036486
5 회차	Bi-LSTM	0.023091	0.029067	0.024033	0.033278	0.046766	0.037692
	Att-Bi-LSTM-Only	0.025432	0.040096	0.027678	0.033539	0.045252	0.035367
	Att-Bi-LSTM-Concat	0.023582	0.027367	0.024665	0.034817	0.040149	0.036507
	Att-Bi-LSTM-Dot	0.024585	0.027404	0.025362	0.034089	0.035937	0.034695
평 균	Bi-LSTM	0.023008	0.029386	0.024364	0.048101	0.067988	0.055806
	Att-Bi-LSTM-Only	0.026310	0.059988	0.032183	0.037142	0.086955	0.046442
	Att-Bi-LSTM-Concat	0.024261	0.035933	0.026975	0.038088	0.058687	0.043002
	Att-Bi-LSTM-Dot	0.024715	0.029456	0.025673	0.033560	0.039202	0.034914

3) U+ 종목의 Att-Bi-LSTM-Only 모델 실험 결과

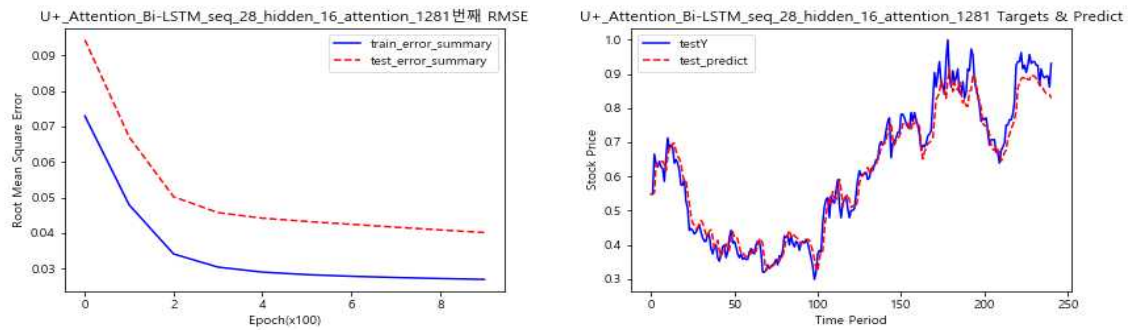


그림 4-33 U+ Att-Bi-LSTM-Only 모델 1회 RMSE, 실제값과 예측값

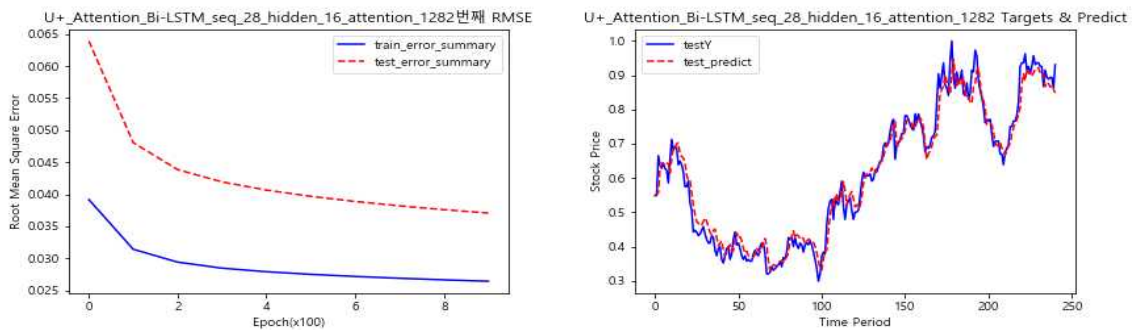


그림 4-34 U+ Att-Bi-LSTM-Only 모델 2회 RMSE, 실제값과 예측값

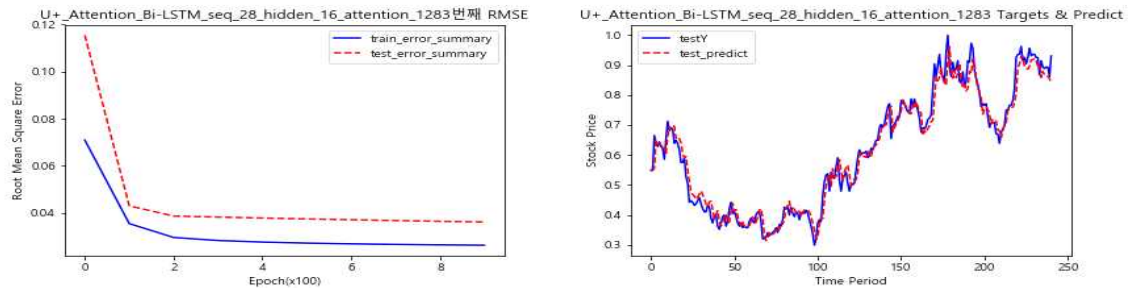


그림 4-35 U+ Att-Bi-LSTM-Only 모델 3회 RMSE, 실제값과 예측값

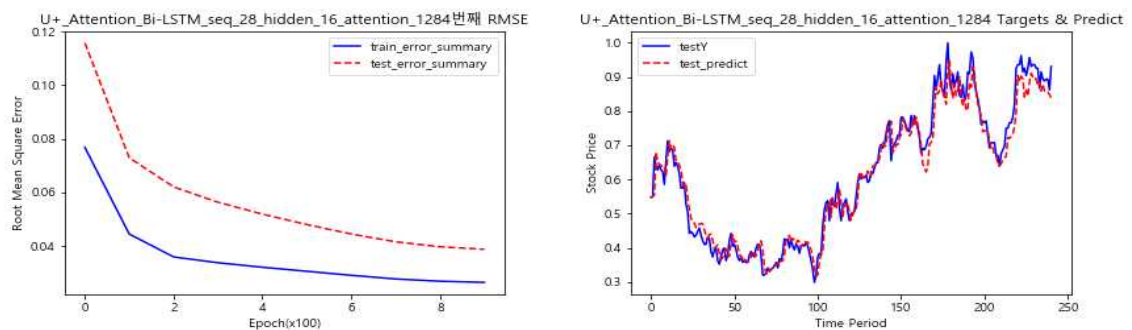


그림 4-36 U+ Att-Bi-LSTM-Only 모델 4회 RMSE, 실제값과 예측값

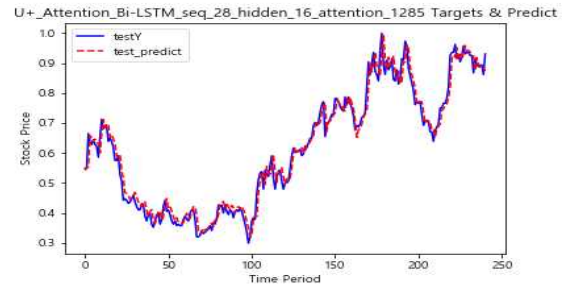
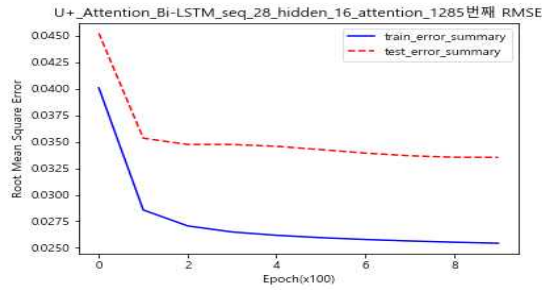


그림 4-37 U+ Att-Bi-LSTM-Only 모델 5회 RMSE, 실제값과 예측값

4) U+ 종목의 Att-Bi-LSTM-Concat 모델 실험 결과

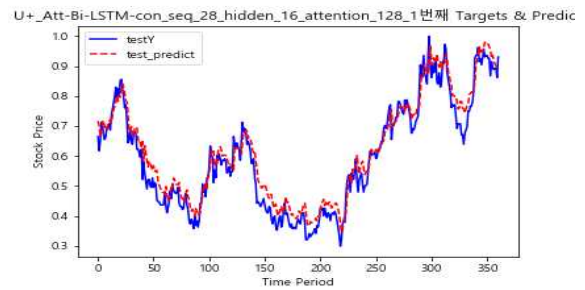
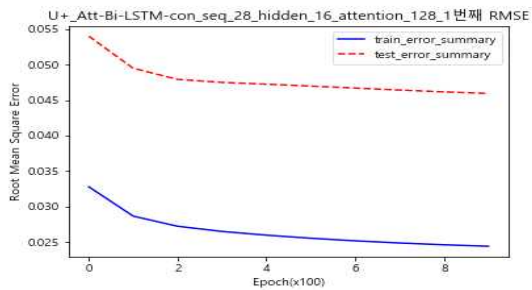


그림 4-38 U+ Att-Bi-LSTM-Concat 모델 1회 RMSE, 실제값과 예측값

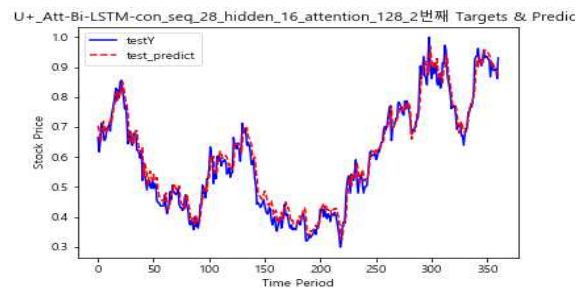
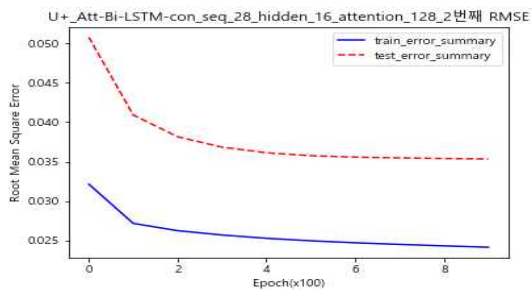


그림 4-39 U+ Att-Bi-LSTM-Concat 모델 2회 RMSE, 실제값과 예측값

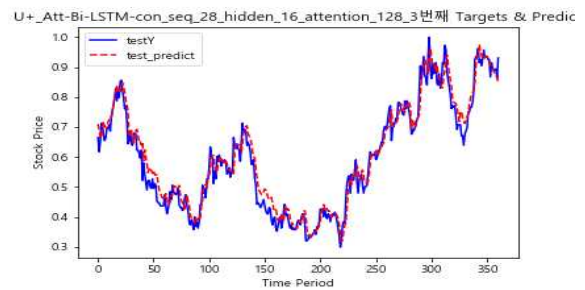
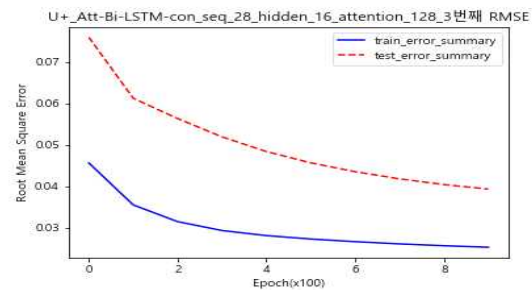


그림 4-40 U+ Att-Bi-LSTM-Concat 모델 3회 RMSE, 실제값과 예측값

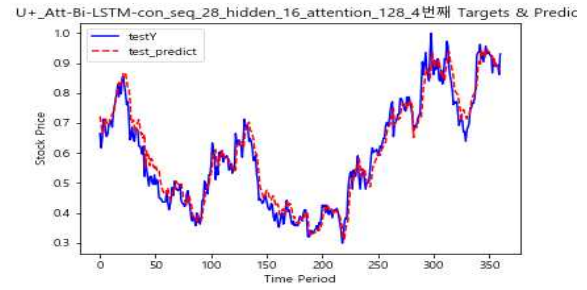
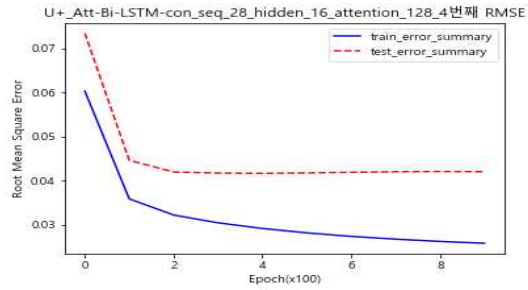


그림 4-41 U+ Att-Bi-LSTM-Concat 모델 4회 RMSE, 실제값과 예측값

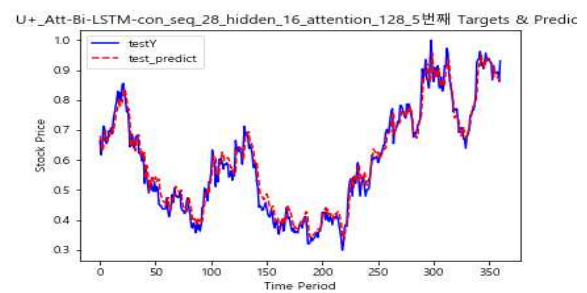
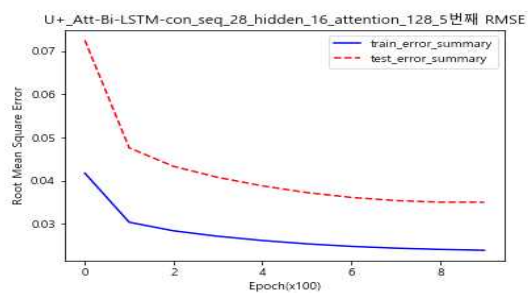


그림 4-42 U+ Att-Bi-LSTM-Concat 모델 5회 RMSE, 실제값과 예측값

5) U+ 종목의 Att-Bi-LSTM-Dot 모델 실험 결과

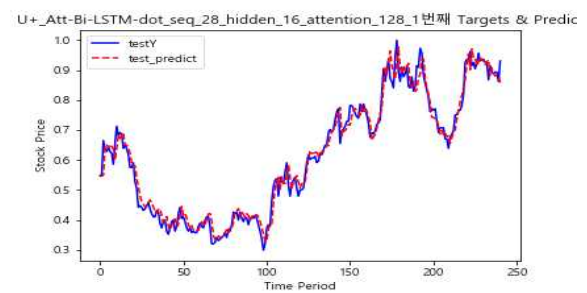
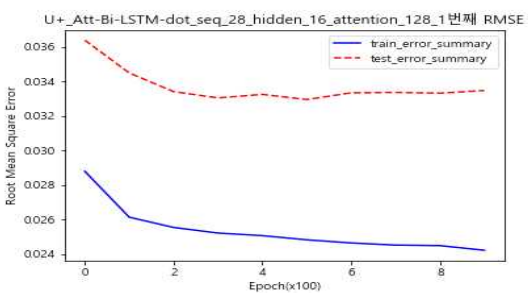


그림 4-43 U+ Att-Bi-LSTM-Dot 모델 1회 RMSE, 실제값과 예측값

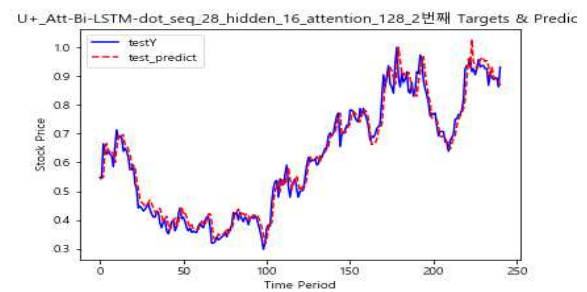
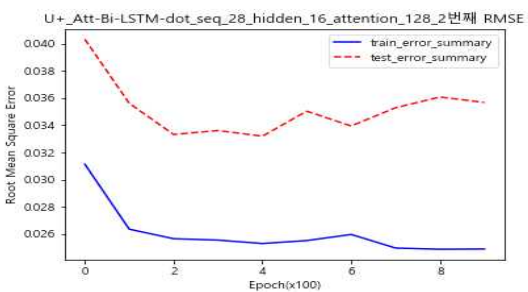


그림 4-44 U+ Att-Bi-LSTM-Dot 모델 2회 RMSE, 실제값과 예측값

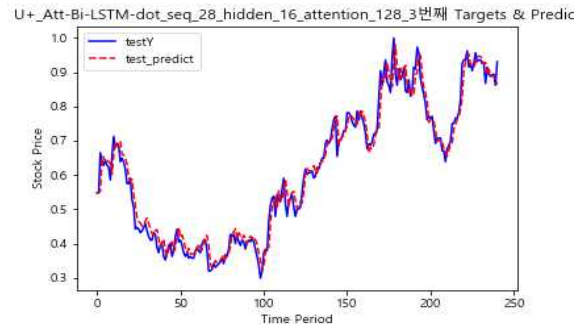
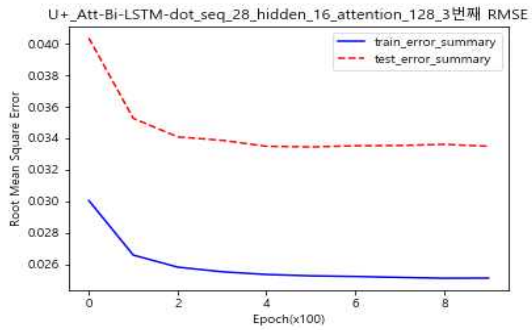


그림 4-45 U+ Att-Bi-LSTM-Dot 모델 3회 RMSE, 실제값과 예측값

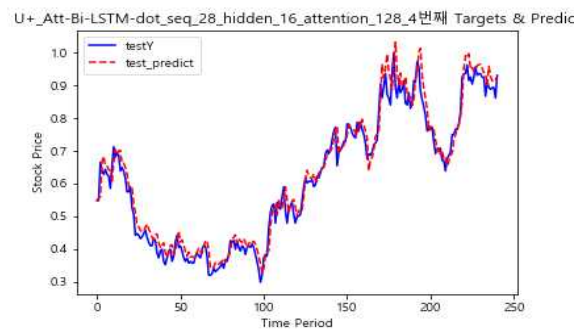
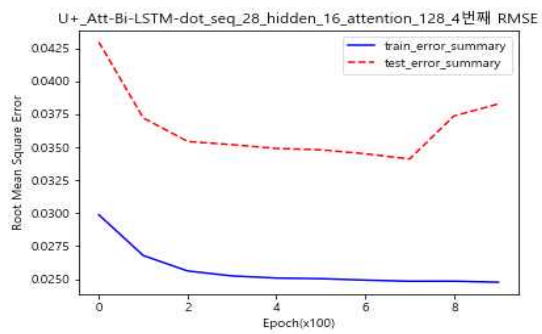


그림 4-46 U+ Att-Bi-LSTM-Dot 모델 4회 RMSE, 실제값과 예측값

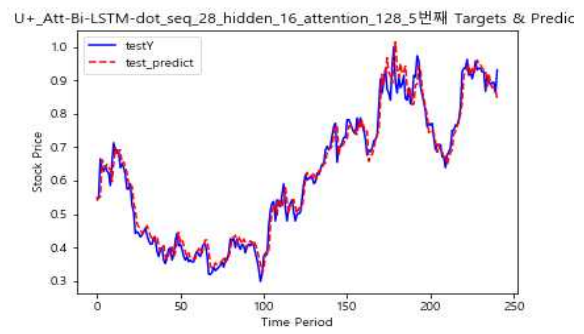
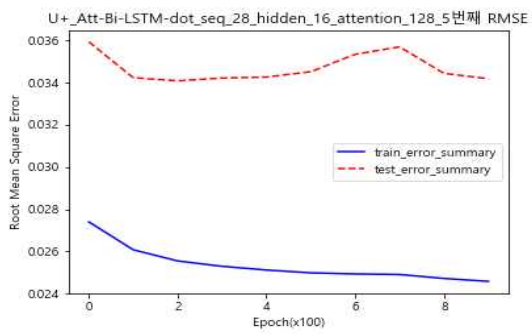


그림 4-47 U+ Att-Bi-LSTM-Dot 모델 5회 RMSE, 실제값과 예측값

6) U+ 종목의 Bi-LSTM 대비 제안 모델별 RMSE 비교

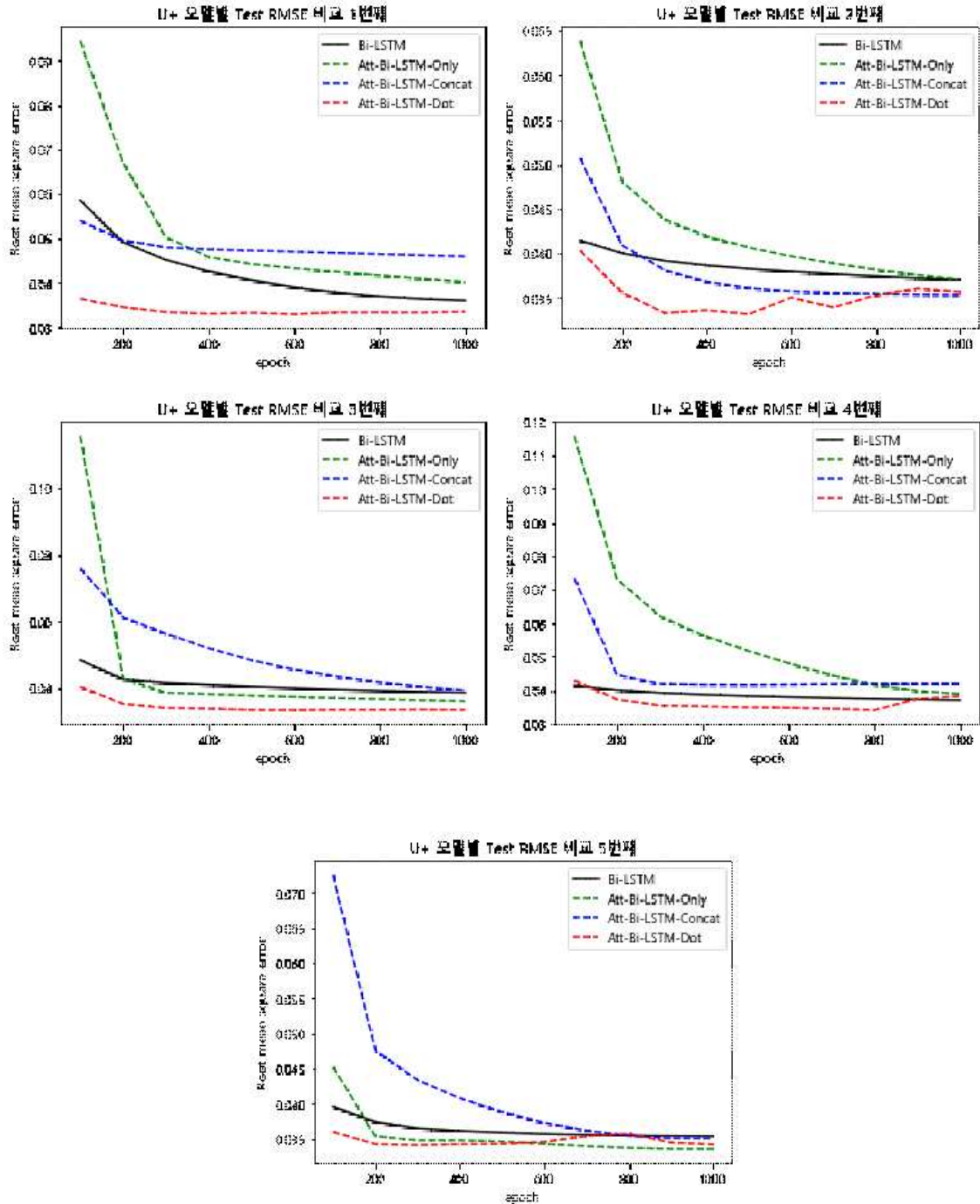


그림 4-48 U+ 종목의 비교 모델 대비 제안 모델별 RMSE 비교그래프

LG 유플러스 종목에서는 제안한 Attention 기반 Bi-LSTM 모델 3가지 방법 중 Att-Bi-LSTM-Dot 모델이 대체로 가장 오차가 작은 것을 알 수 있다..

(2) KT 종목에 적합한 학습 파라미터

2) 적절한 Sequence Length/Hidden Layer 분석

표 4-8 KT Bi-LSTM을 이용한 시퀀스 길이와 hidden layer 크기에 따른 RMSE 비교

input_ data	seq_ length	hidden_ dim	training rmse			test rmse		
			Min	Max	Avg	Min	Max	Avg
23	5	5	0.034883	0.058441	0.040927	0.037996	0.046704	0.042549
23	5	10	0.035915	0.065500	0.043973	0.041821	0.069459	0.057752
23	5	16	0.032536	0.055035	0.037258	0.032470	0.067610	0.044635
23	5	64	0.038608	0.068708	0.046923	0.035840	0.056134	0.044766
23	5	32	0.033991	0.056904	0.040996	0.033388	0.050527	0.039991
최소			0.032536	0.055035	0.037258	0.032470	0.046704	0.039991
평균			0.035187	0.060917	0.042015	0.036303	0.058087	0.045938
23	10	5	0.034012	0.048979	0.037405	0.031456	0.048151	0.034844
23	10	10	0.033044	0.053444	0.037967	0.032920	0.057888	0.043362
23	10	16	0.031972	0.053776	0.035850	0.033296	0.061530	0.040260
23	10	32	0.032416	0.051543	0.037053	0.032109	0.051576	0.039686
23	10	64	0.032826	0.055498	0.037971	0.035142	0.060930	0.043575
최소			0.031972	0.048979	0.035850	0.031456	0.048151	0.034844
평균			0.032854	0.052648	0.037249	0.032985	0.056015	0.040345
23	20	5	0.033886	0.080348	0.042643	0.032217	0.063521	0.037967
23	20	10	0.033760	0.059406	0.039686	0.039452	0.063648	0.050973
23	20	16	0.037624	0.071440	0.046426	0.064019	0.116194	0.089139
23	20	32	0.034686	0.068639	0.041904	0.032835	0.059901	0.040568
23	20	64	0.032957	0.055154	0.038785	0.032631	0.061085	0.043182
최소			0.032957	0.055154	0.038785	0.032217	0.059901	0.037967
평균			0.034583	0.066997	0.041889	0.040230	0.072870	0.052366
23	28	5	0.035481	0.053503	0.039863	0.039160	0.058363	0.045421
23	28	10	0.032574	0.044050	0.035175	0.031490	0.039382	0.033114
23	28	16	0.034711	0.080114	0.044616	0.035875	0.064777	0.042397
23	28	32	0.037062	0.070851	0.046412	0.045839	0.083587	0.058766
23	28	64	0.037403	0.064631	0.044928	0.055678	0.106495	0.074334
최소			0.032574	0.044050	0.035175	0.031490	0.039382	0.033114
평균			0.035446	0.062630	0.042199	0.041609	0.070521	0.050806

2) 적절한 Attention Size 분석

<표 4-8>에 따라 KT는 시퀀스 길이를 28, hidden layer 크기는 10으로 지정하여 Attention size 변화를 측정하였다.

표 4-9 KT의 입력 23, seq 28, hidden 10, attention_size변화에 따른 RMSE비교

seq	hidden	attention_size	training_RMSE			test_RMSE		
			Min	MAx	Avg	Min	MAx	Avg
28	10	5	0.031985	0.091184	0.04164	0.0351	0.078736	0.046428
28	10	10	0.032927	0.048936	0.035861	0.032111	0.054638	0.035772
28	10	16	0.034499	0.091049	0.043516	0.034432	0.085812	0.046942
28	10	20	0.036198	0.108538	0.050082	0.060456	0.124954	0.077062
28	10	32	0.035839	0.099713	0.050564	0.038665	0.102025	0.059986
28	10	40	0.032351	0.042714	0.034913	0.0312	0.038498	0.034164
28	10	64	0.034818	0.096631	0.046628	0.038136	0.112204	0.054668
28	10	128	0.034684	0.047248	0.039137	0.045098	0.080294	0.064322

<표 4-9>에 따라 KT종목은 시퀀스 길이를 28, hidden layer 크기는 10으로 지정하여 Attention size는 40로 결정하였다. 그리고 5회 반복 수행하여 평균을 구하여 Bi-LSTM 모델과 Attention-Bi-LSTM 모델의 RMSE를 비교하였다.

표 4-10 KT Bi-LSTM과 Attention-Bi-LSTM 모델별 평균 RMSE 비교

KT		training_RMSE			test_RMSE		
		Min	Max	Avg	Min	Max	Avg
1회차	Bi-LSTM	0.033490	0.048250	0.036460	0.034800	0.049970	0.041470
	Att-Bi-LSTM-Only	0.030570	0.037900	0.032930	0.034050	0.051970	0.037830
	Att-Bi-LSTM-Concat	0.032150	0.038680	0.034120	0.032530	0.040100	0.034770
	Att-Bi-LSTM-Dot	0.032290	0.043920	0.034290	0.032700	0.057820	0.039970
2회차	Bi-LSTM	0.032560	0.041950	0.034880	0.033420	0.053050	0.040670
	Att-Bi-LSTM-Only	0.031540	0.043900	0.034070	0.031700	0.046340	0.036930
	Att-Bi-LSTM-Concat	0.033940	0.061490	0.038670	0.046570	0.111630	0.066790
	Att-Bi-LSTM-Dot	0.032150	0.038090	0.033880	0.032180	0.046110	0.037460
3회차	Bi-LSTM	0.035130	0.047590	0.039520	0.046800	0.092620	0.068850
	Att-Bi-LSTM-Only	0.034980	0.099260	0.046260	0.039090	0.123690	0.064680
	Att-Bi-LSTM-Concat	0.031260	0.038110	0.033270	0.032080	0.043850	0.035940
	Att-Bi-LSTM-Dot	0.032500	0.041130	0.034690	0.032910	0.053570	0.039190
4회차	Bi-LSTM	0.033540	0.042890	0.036030	0.039560	0.054050	0.048740
	Att-Bi-LSTM-Only	0.033300	0.047990	0.036610	0.037240	0.057740	0.046060
	Att-Bi-LSTM-Concat	0.034070	0.057170	0.039050	0.037370	0.062230	0.047830
	Att-Bi-LSTM-Dot	0.031030	0.041040	0.033470	0.033750	0.056590	0.039000
5회차	Bi-LSTM	0.033500	0.048200	0.036620	0.033540	0.050770	0.039810
	Att-Bi-LSTM-Only	0.032570	0.040060	0.034300	0.034160	0.046850	0.037760
	Att-Bi-LSTM-Concat	0.033950	0.050440	0.038380	0.035230	0.060640	0.044160
	Att-Bi-LSTM-Dot	0.032480	0.042670	0.034510	0.032110	0.051610	0.036770
평균	Bi-LSTM	0.033644	0.045776	0.036702	0.037624	0.060092	0.047908
	Att-Bi-LSTM-Only	0.032592	0.053822	0.036834	0.035248	0.065318	0.044652
	Att-Bi-LSTM-Concat	0.033074	0.049178	0.036698	0.036756	0.063690	0.045898
	Att-Bi-LSTM-Dot	0.032090	0.041370	0.034168	0.032730	0.053140	0.038478

3) KT 종목의 Att-Bi-LSTM-Only 모델 실험 결과

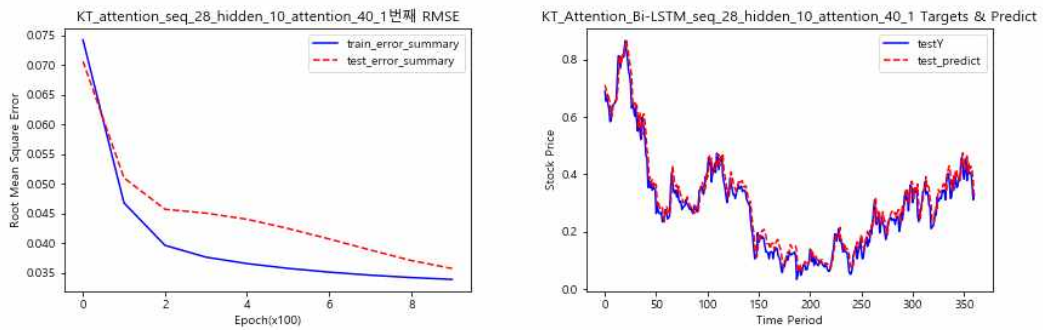


그림 4-49 KT Att-Bi-LSTM-Only 1회 RMSE, 실제값과 예측값

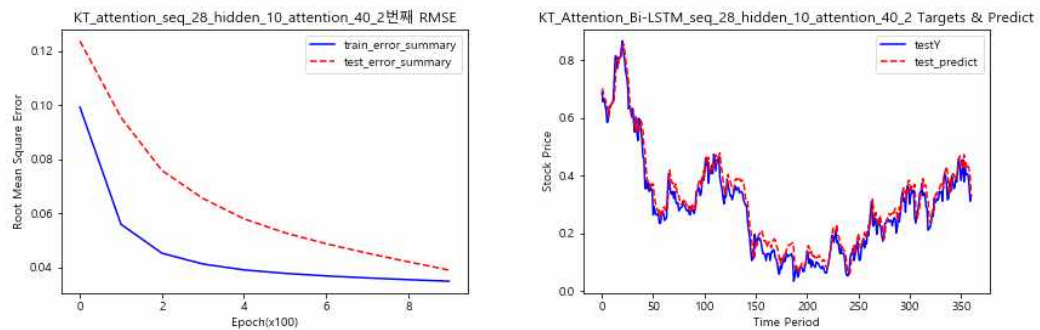


그림 4-50 KT Att-Bi-LSTM-Only 2회 RMSE, 실제값과 예측값

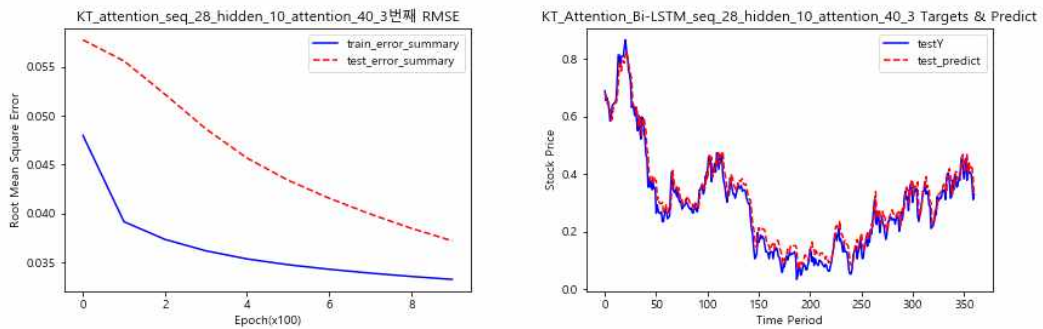


그림 4-51 KT Att-Bi-LSTM-Only 3회 RMSE, 실제값과 예측값

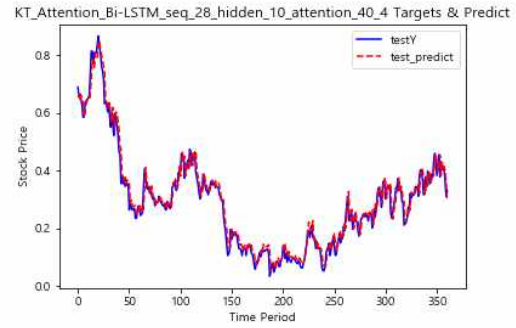
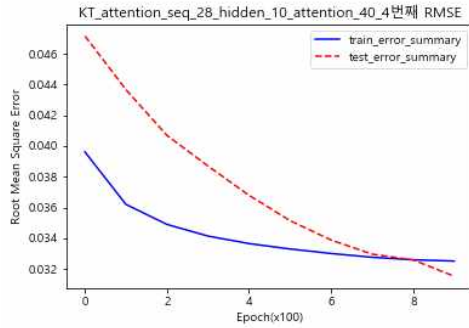


그림 4-52 KT Att-Bi-LSTM-Only 4회 RMSE, 실제값과 예측값

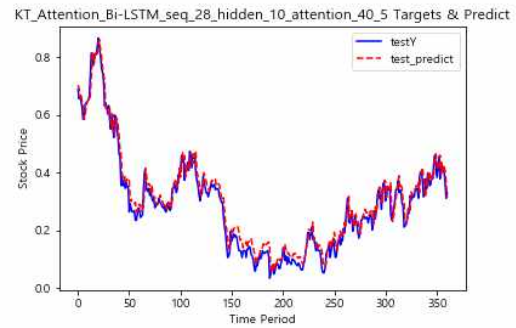
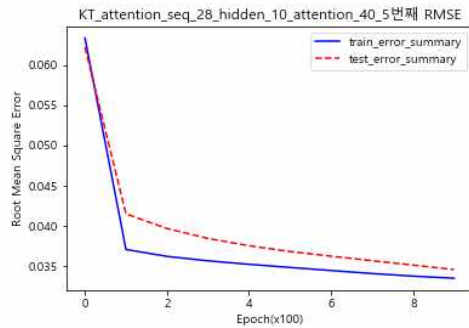


그림 4-53 KT Att-Bi-LSTM-Only 5회 RMSE, 실제값과 예측값

4) KT 종목의 Att-Bi-LSTM-Concat 모델의 실험 결과

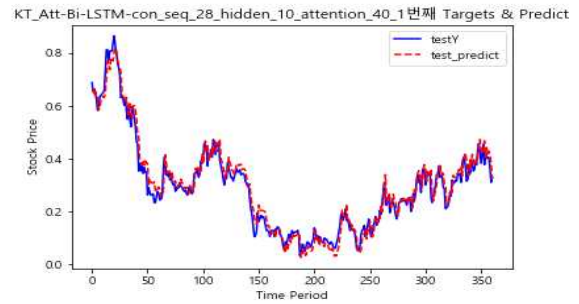
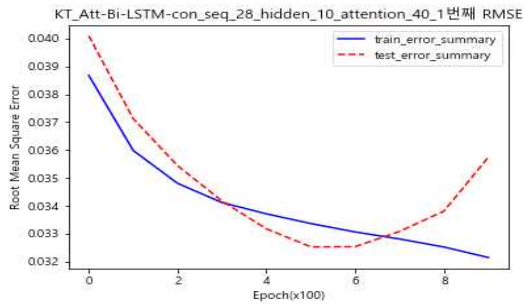


그림 4-54 KT Att-Bi-LSTM-Concat 1회 RMSE, 실제값과 예측값

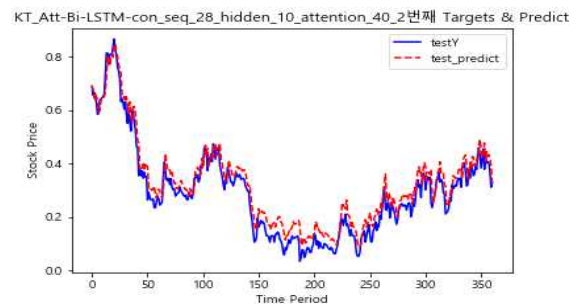
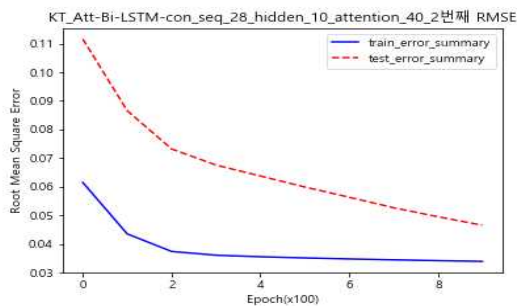


그림 4-55 KT Att-Bi-LSTM-Concat 2회 RMSE, 실제값과 예측값

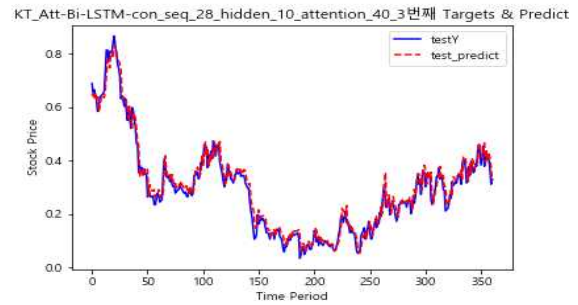
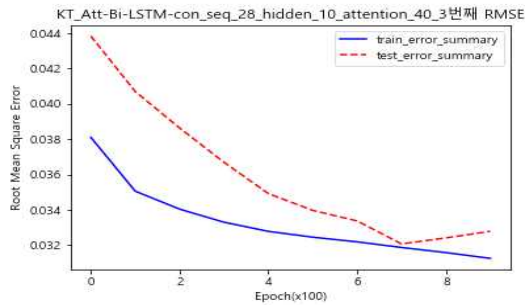


그림 4-56 KT Att-Bi-LSTM-Concat 3회 RMSE, 실제값과 예측값

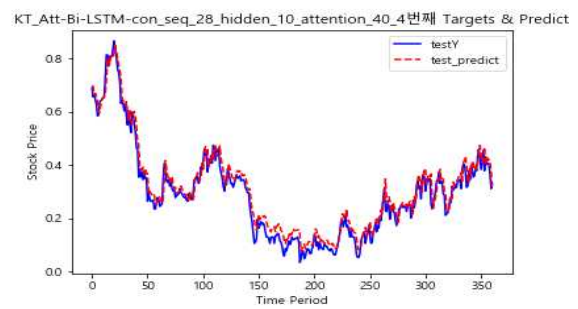
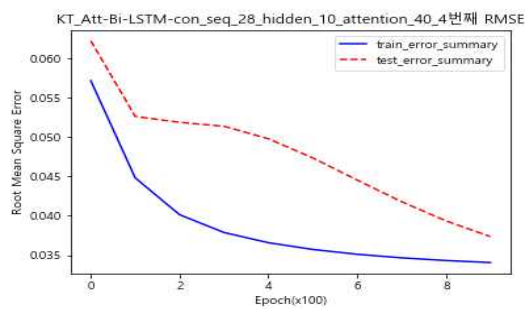


그림 4-57 KT Att-Bi-LSTM-Concat 4회 RMSE, 실제값과 예측값

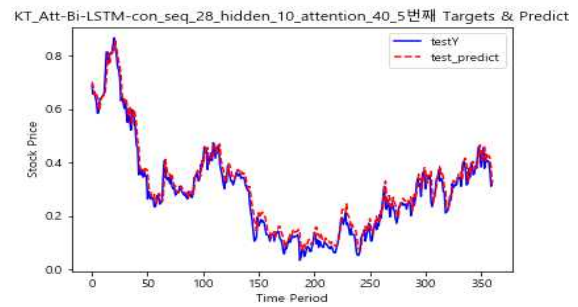
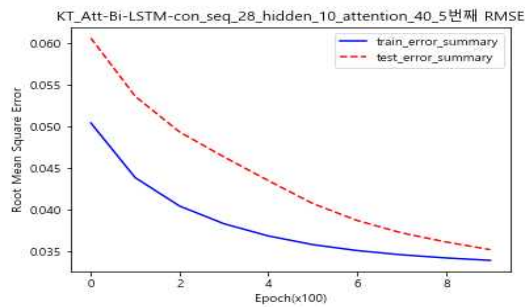


그림 4-58 KT Att-Bi-LSTM-Concat 5회 RMSE, 실제값과 예측값

5) KT 종목의 Att-Bi-LSTM-Dot 모델 실험 결과

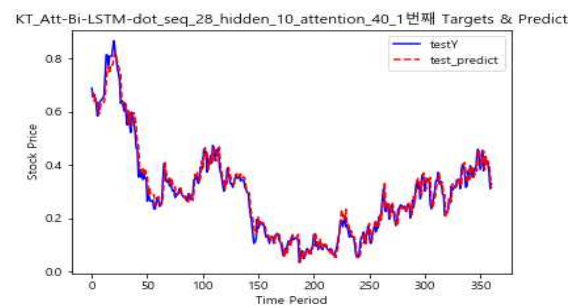
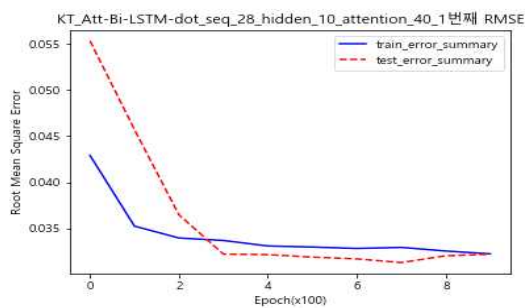


그림 4-59 KT Att-Bi-LSTM-Dot 1회 RMSE, 실제값과 예측값

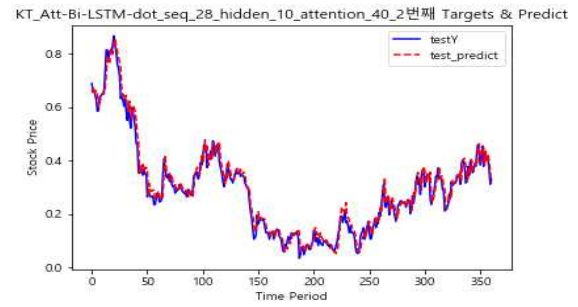
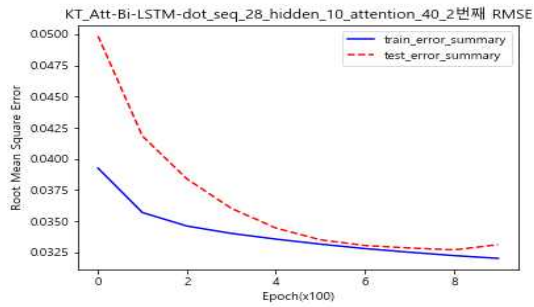


그림 4-60 KT Att-Bi-LSTM-Dot 2회 RMSE, 실제값과 예측값

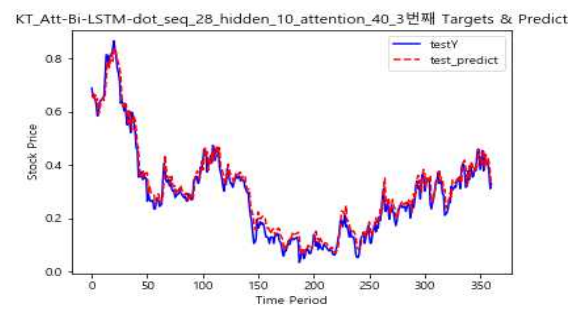
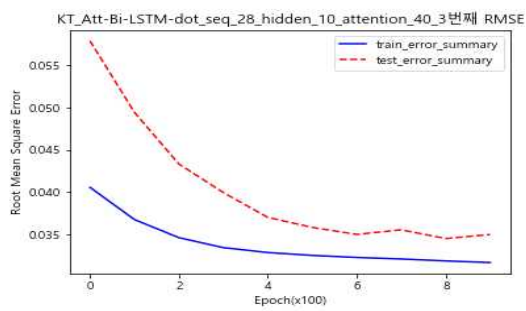


그림 4-61 KT Att-Bi-LSTM-Dot 3회 RMSE, 실제값과 예측값

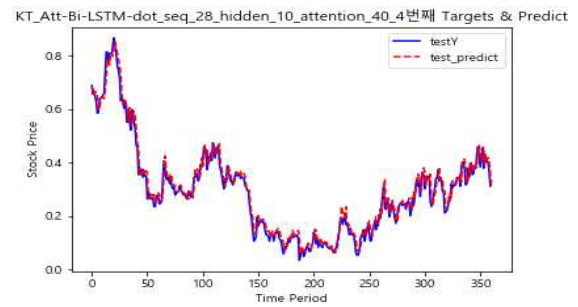
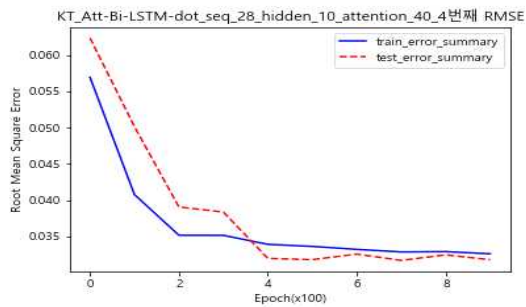


그림 4-62 KT Att-Bi-LSTM-Dot 4회 RMSE, 실제값과 예측값

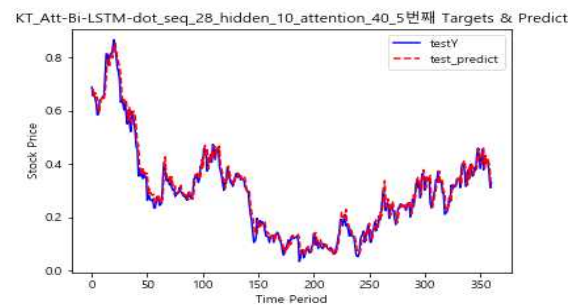
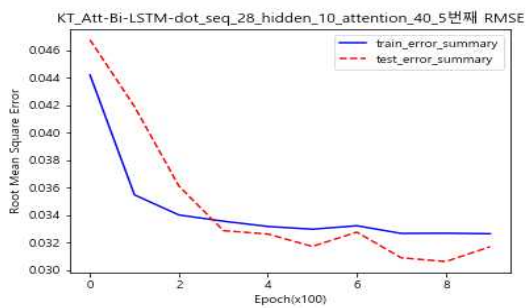


그림 4-63 KT Att-Bi-LSTM-Dot 5회 RMSE, 실제값과 예측값

(3) SK텔레콤 종목에 적합한 학습 파라미터

1) 적절한 Sequence Length / Hidden Layer 분석

표 4-11 SK텔레콤 Bi-LSTM을 이용한 시퀀스 길이와 hidden layer 크기에 따른 RMSE 비교

input_ data	seq_ length	hidden_ dim	training rmse			test rmse		
			Min	Max	Avg	Min	Max	Avg
23	5	5	0.036102	0.059329	0.040960	0.038283	0.068487	0.045282
23	5	10	0.035884	0.058782	0.043581	0.037514	0.061775	0.046050
23	5	16	0.037271	0.060775	0.043275	0.037510	0.053485	0.042232
23	5	32	0.033916	0.055888	0.040373	0.036078	0.061548	0.042961
23	5	64	0.041019	0.063495	0.049767	0.040240	0.062960	0.048432
최소			0.033916	0.055888	0.040373	0.036078	0.053485	0.042232
평균			0.036839	0.059654	0.043591	0.037925	0.061651	0.044991
23	10	5	0.035629	0.058378	0.041669	0.036997	0.064749	0.043047
23	10	10	0.031129	0.049752	0.034992	0.040030	0.051075	0.042715
23	10	16	0.039571	0.059780	0.046095	0.040692	0.066098	0.048302
23	10	32	0.038225	0.080250	0.049523	0.038184	0.102639	0.056290
23	10	64	0.036900	0.072729	0.046507	0.038848	0.074864	0.048659
최소			0.031129	0.049752	0.034992	0.036997	0.051075	0.042715
평균			0.036291	0.064178	0.043757	0.038950	0.071885	0.047803
23	20	5	0.039101	0.075675	0.047184	0.046122	0.079138	0.052718
23	20	10	0.033687	0.058649	0.038541	0.037542	0.067344	0.041900
23	20	16	0.036391	0.076017	0.047153	0.040701	0.114350	0.065405
23	20	32	0.037785	0.088837	0.049153	0.038501	0.089752	0.049501
23	20	64	0.035567	0.066549	0.042857	0.037820	0.072899	0.045724
최소			0.033687	0.058649	0.038541	0.037542	0.067344	0.041900
평균			0.036506	0.073145	0.044978	0.040137	0.084697	0.051049
23	28	5	0.034095	0.060097	0.0396	0.038186	0.061356	0.042208
23	28	10	0.032692	0.049566	0.036039	0.035815	0.050731	0.038708
23	28	16	0.031647	0.049576	0.036228	0.037674	0.059375	0.042493
23	28	32	0.034014	0.060024	0.040194	0.042532	0.062915	0.04738
23	28	64	0.043672	0.17646	0.075214	0.044638	0.185971	0.076713
최소			0.031647	0.049566	0.036039	0.035815	0.050731	0.038708
평균			0.035224	0.079145	0.045455	0.039769	0.084070	0.049501

<표 4-11>에 따라 SK 텔레콤은 시퀀스 길이를 28, hidden layer 크기는 10으로 지정하여 Attention size를 측정하였다.

표 4-12 SK텔레콤의 입력 23, seq 28, hidden 10, attention_size변화에 따른 RMSE 비교

seq	hidden	attention_size	training_RMSE			test_RMSE		
			Min	MAx	Avg	Min	MAx	Avg
28	10	5	0.035609	0.083896	0.04327	0.039179	0.087039	0.047245
28	10	10	0.034280	0.052205	0.038684	0.03569	0.067292	0.041911
28	10	16	0.035560	0.078547	0.042983	0.036516	0.081632	0.045100
28	10	20	0.034128	0.079606	0.040492	0.036752	0.079803	0.043245
28	10	32	0.035285	0.063883	0.040214	0.037047	0.07696	0.043827
28	10	40	0.033162	0.040702	0.035082	0.036899	0.047672	0.039001
28	10	64	0.033953	0.071109	0.0394	0.037797	0.072987	0.041983
28	10	128	0.036600	0.052528	0.040654	0.03746	0.045906	0.038928

3) 적절한 Attention Size 분석

<표 4-12>에 따라 SK 텔레콤은 시퀀스 길이를 28, hidden layer 크기는 10으로 지정하여 Attention size는 128로 결정하였다. 그리고 5회 반복 수행하여 평균을 구하여 Bi-LSTM 모델과 Attention-Bi-LSTM 모델의 RMSE를 비교하였다.

표 4-13 SK텔레콤 Bi-LSTM과 Attention-Bi-LSTM 모델별 평균 RMSE 비교

SK 텔레콤		training_RMSE			test_RMSE		
		Min	Max	Avg	Min	Max	Avg
1 회 차	Bi-LSTM	0.036701	0.051631	0.040623	0.040405	0.063340	0.047598
	Att-Bi-LSTM-Only	0.035293	0.057620	0.039499	0.035685	0.074783	0.043409
	Att-Bi-LSTM-Concat	0.035557	0.057632	0.040065	0.036728	0.057300	0.040182
	Att-Bi-LSTM-Dot	0.033365	0.042305	0.035586	0.036983	0.039237	0.037740
2 회 차	Bi-LSTM	0.034125	0.042360	0.036096	0.035841	0.046767	0.037675
	Att-Bi-LSTM-Only	0.034895	0.048577	0.037885	0.035085	0.043048	0.036932
	Att-Bi-LSTM-Concat	0.033571	0.072563	0.039753	0.038028	0.071966	0.042572
	Att-Bi-LSTM-Dot	0.033843	0.040573	0.035409	0.035790	0.050542	0.038029
3 회 차	Bi-LSTM	0.033771	0.044224	0.036180	0.036712	0.051123	0.038848
	Att-Bi-LSTM-Only	0.033852	0.045366	0.036185	0.035592	0.044031	0.037220
	Att-Bi-LSTM-Concat	0.034962	0.062577	0.040544	0.037038	0.067691	0.042664
	Att-Bi-LSTM-Dot	0.033627	0.041415	0.035970	0.036131	0.046225	0.038527
4 회 차	Bi-LSTM	0.035509	0.049888	0.039039	0.037494	0.070924	0.047012
	Att-Bi-LSTM-Only	0.035730	0.085464	0.042235	0.034982	0.088330	0.041909
	Att-Bi-LSTM-Concat	0.033637	0.051920	0.036990	0.036220	0.063001	0.040205
	Att-Bi-LSTM-Dot	0.034225	0.040396	0.035926	0.035916	0.046036	0.037748
5 회 차	Bi-LSTM	0.034280	0.045639	0.037293	0.036524	0.055912	0.040264
	Att-Bi-LSTM-Only	0.033445	0.042518	0.035683	0.037571	0.050192	0.039813
	Att-Bi-LSTM-Concat	0.034927	0.050290	0.038419	0.036162	0.047381	0.038245
	Att-Bi-LSTM-Dot	0.033409	0.047102	0.036077	0.036316	0.059832	0.040190
평 균	Bi-LSTM	0.034877	0.046748	0.037846	0.037395	0.057613	0.042279
	Att-Bi-LSTM-Only	0.034643	0.055909	0.038297	0.035783	0.060077	0.039857
	Att-Bi-LSTM-Concat	0.034531	0.058996	0.039154	0.036835	0.061468	0.040774
	Att-Bi-LSTM-Dot	0.033694	0.042358	0.035794	0.036227	0.048374	0.038447

4) SK텔레콤 종목의 Att-Bi-LSTM-Only 모델 실험 결과

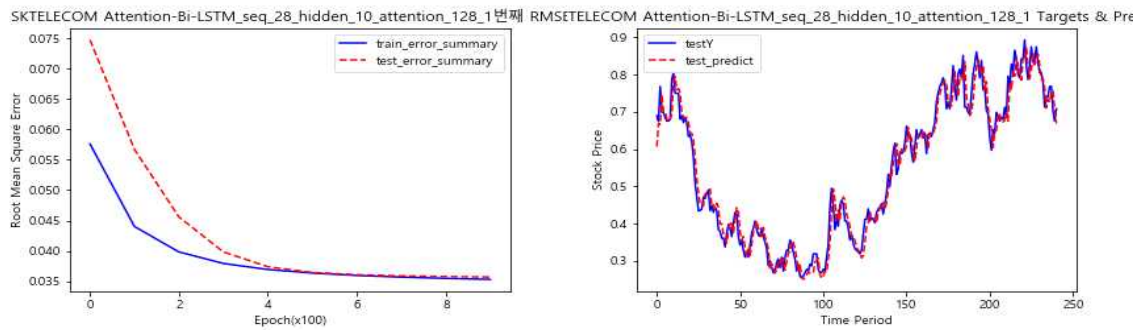


그림 4-64 SK Att-Bi-LSTM-Only 1회 RMSE, 실제값과 예측값

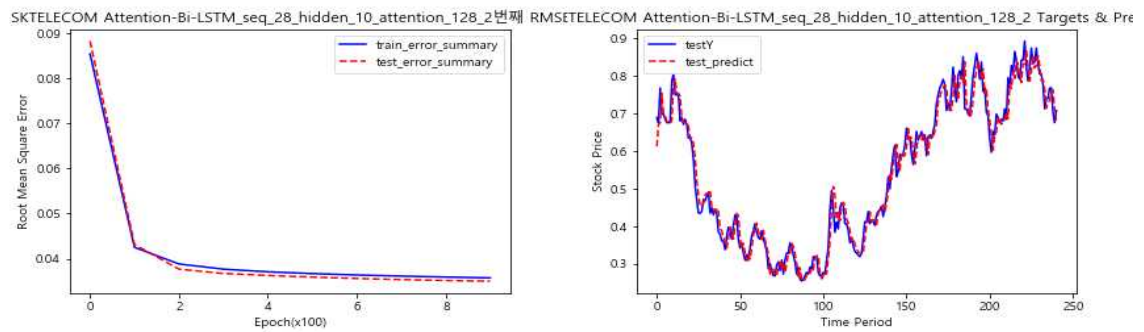


그림 4-65 SK Att-Bi-LSTM-Only 2회 RMSE, 실제값과 예측값

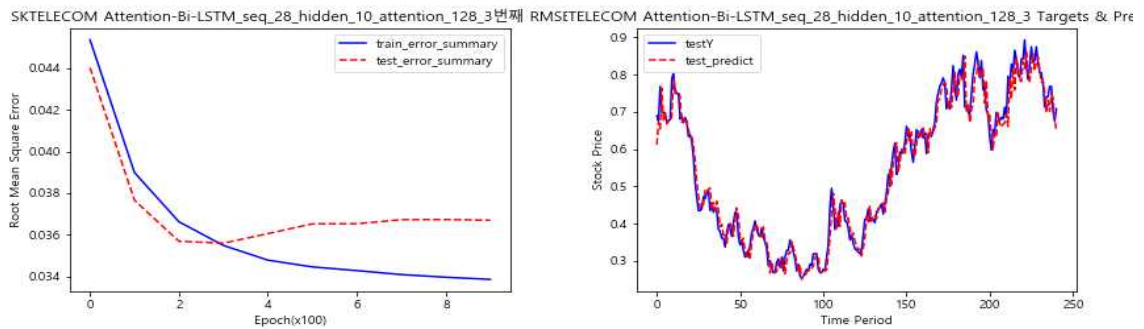


그림 4-66 SK Att-Bi-LSTM-Only 3회 RMSE, 실제값과 예측값

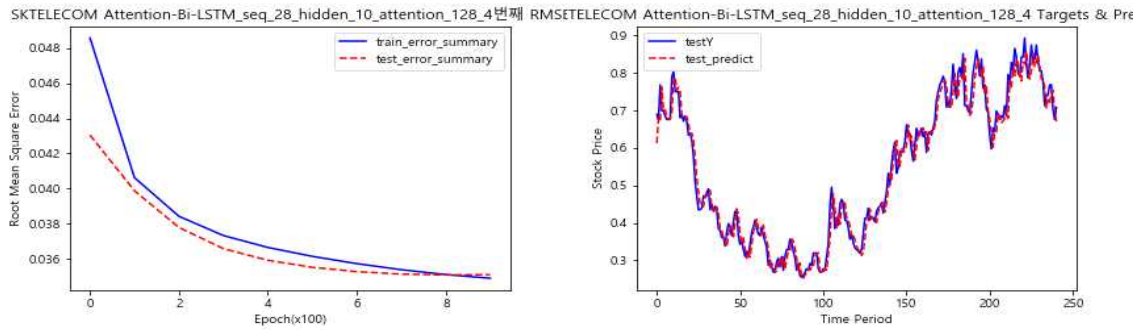


그림 4-67 SK Att-Bi-LSTM-Only 4회 RMSE, 실제값과 예측값

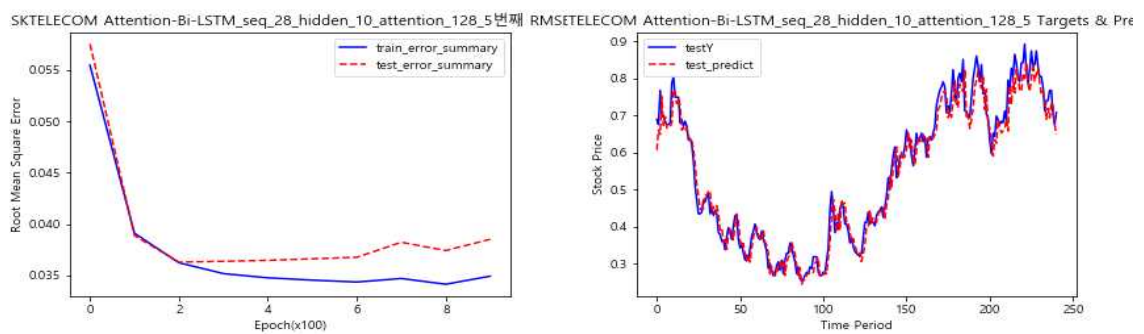


그림 4-68 SK Att-Bi-LSTM-Only 5회 RMSE, 실제값과 예측값

5) SK텔레콤 종목의 Att-Bi-LSTM-Concat 모델 실험 결과

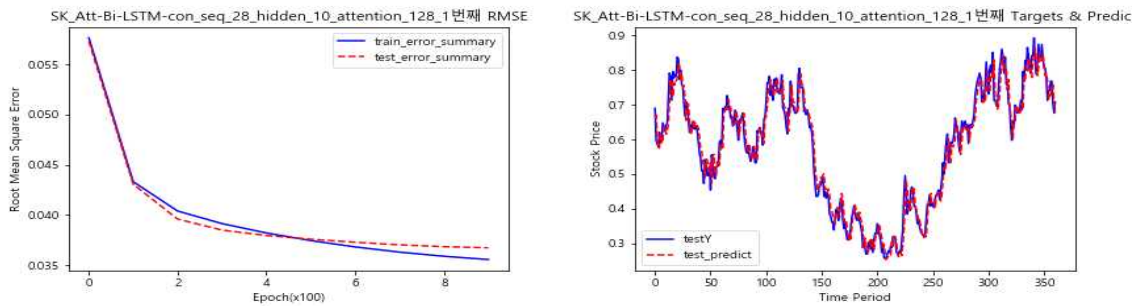


그림 4-69 SK Att-Bi-LSTM-Concat 1회 RMSE, 실제값과 예측값

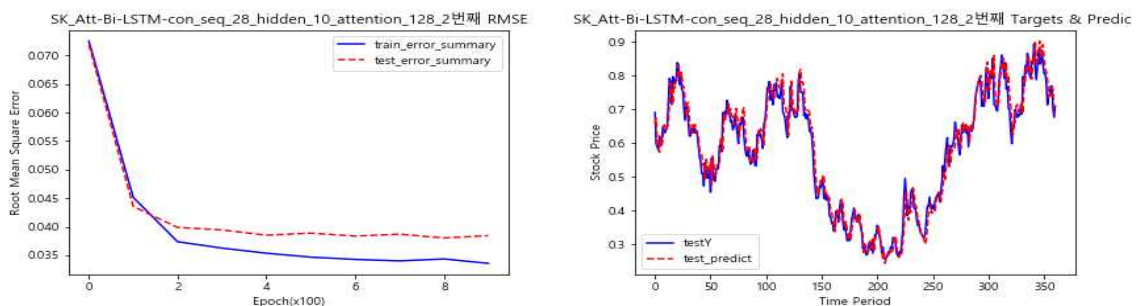


그림 4-70 SK Att-Bi-LSTM-Concat 2회 RMSE, 실제값과 예측값

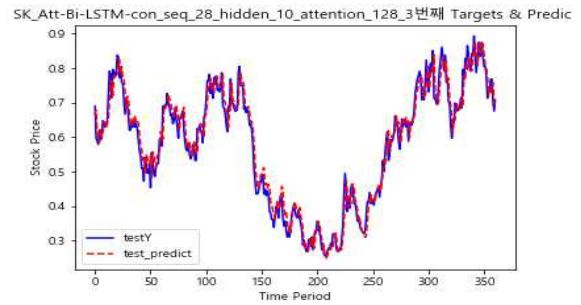
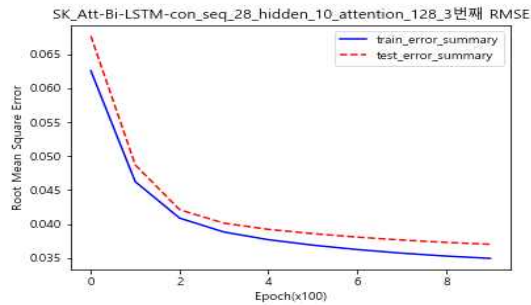


그림 4-71 SK Att-Bi-LSTM-Concat 3회 RMSE, 실제값과 예측값

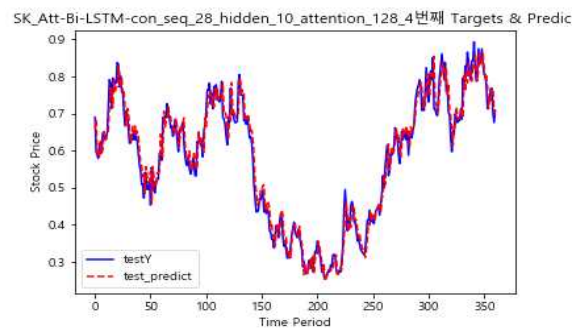
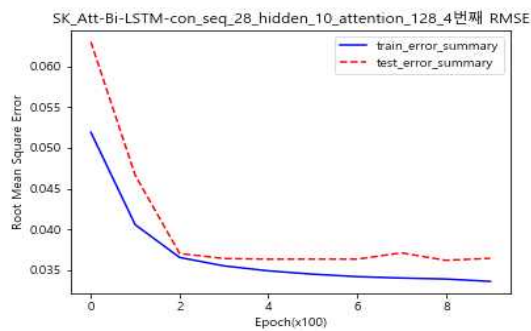


그림 4-72 SK Att-Bi-LSTM-Concat 4회 RMSE, 실제값과 예측값

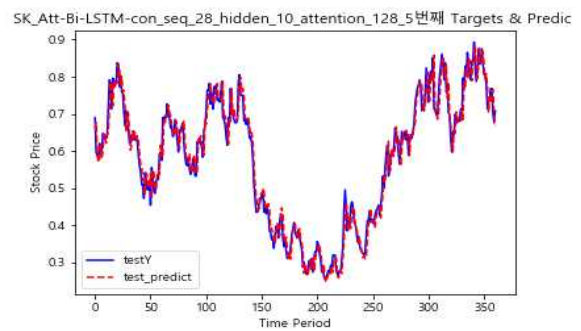
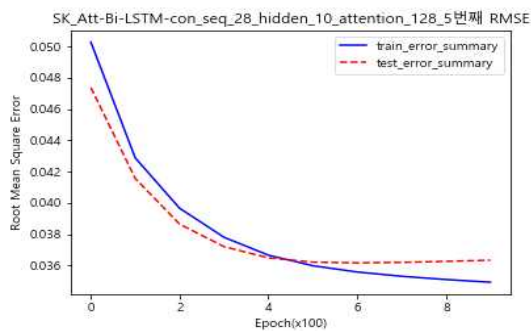


그림 4-73 SK Att-Bi-LSTM-Concat 5회 RMSE, 실제값과 예측값

6) SK텔레콤 종목의 Att-Bi-LSTM-Dot 모델 실험 결과

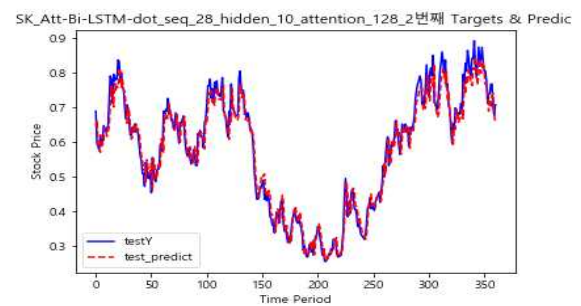
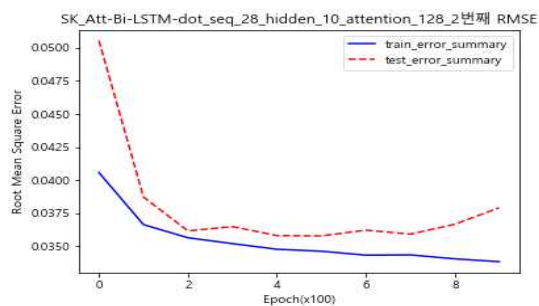


그림 4-74 SK Att-Bi-LSTM-Dot 1회 RMSE, 실제값과 예측값

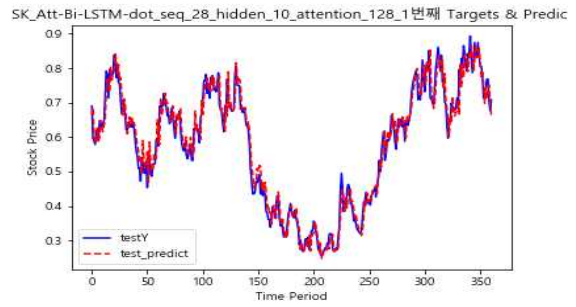
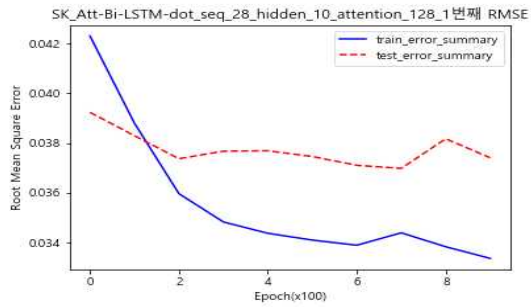


그림 4-75 SK Att-Bi-LSTM-Dot 2회 RMSE, 실제값과 예측값

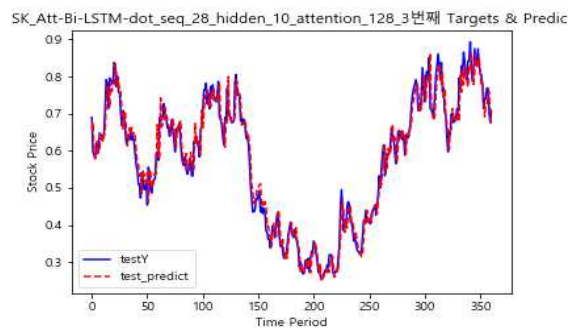
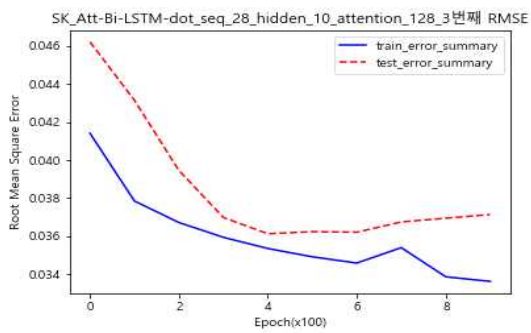


그림 4-76 SK Att-Bi-LSTM-Dot 3회 RMSE, 실제값과 예측값

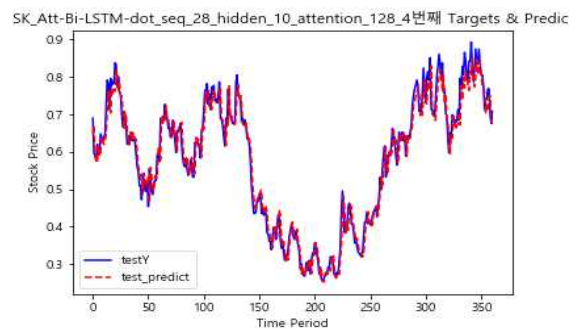
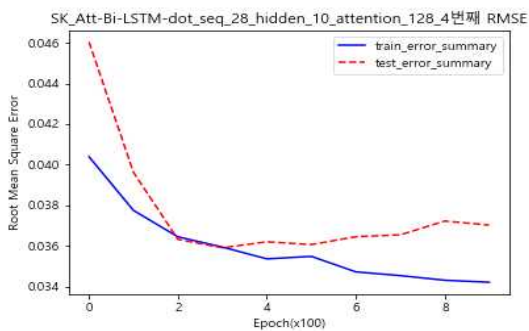


그림 4-77 SK Att-Bi-LSTM-Dot 4회 RMSE, 실제값과 예측값

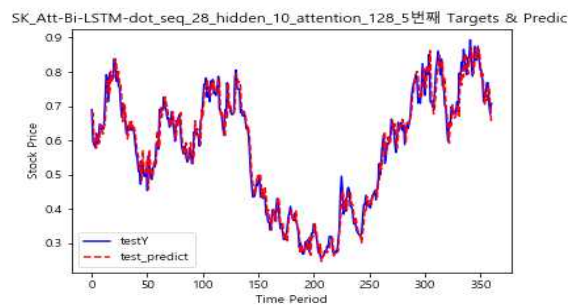
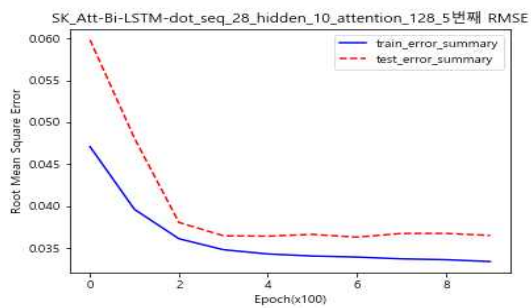


그림 4-78 SK Att-Bi-LSTM-Dot 5회 RMSE, 실제값과 예측값

(4) 실험 결과 분석

표 4-14 종목별 Bi-LSTM 대비 Attention-Bi-LSTM 모델 평균 RMSE

모델 \ 종목	LG U+	KT	SK 텔레콤
Bi-LSTM	0.055806	0.047908	0.042279
Att-Bi-LSTM-Only	0.046442	0.044652	0.039857
Att-Bi-LSTM-Concat	0.043002	0.045898	0.040774
Att-Bi-LSTM-Dot	0.034914	0.038478	0.038447

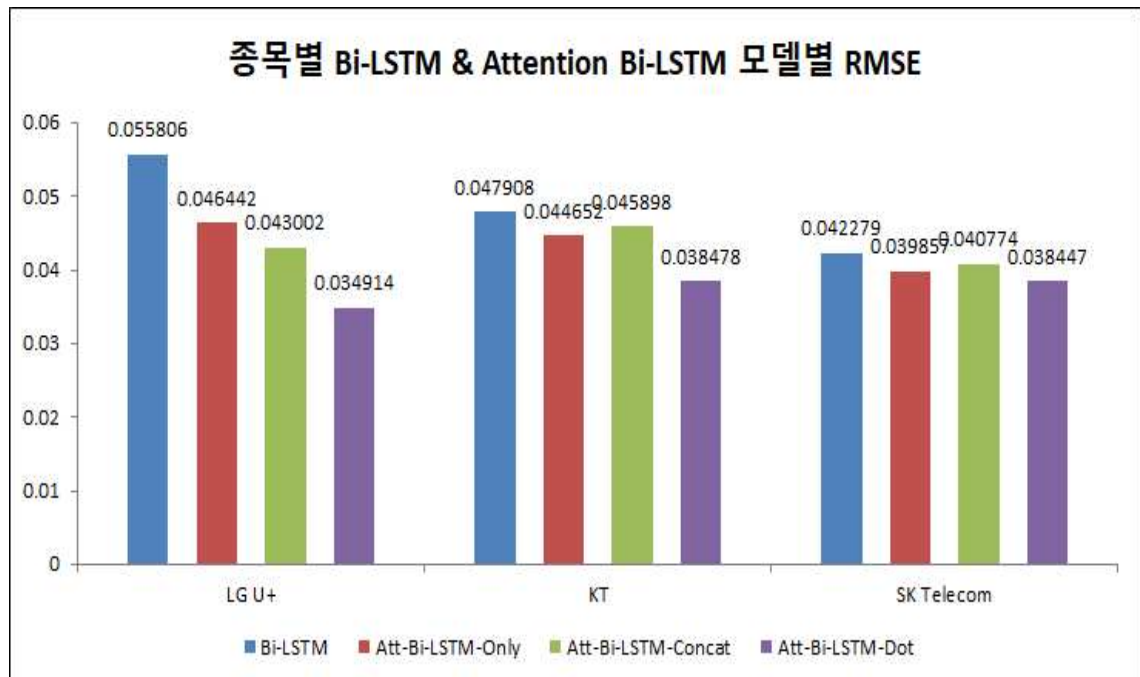


그림 4-79 종목별 Bi-LSTM 대비 Attention 기반 Bi-LSTM 모델별 RMSE 비교

표 4-15 종목별 Bi-LSTM 대비 Attention-Bi-LSTM 모델 평균 감소량

모델 \ 종목	LG U+ 감소량	KT 감소량	SK 텔레콤 감소량
Att-Bi-LSTM-Only	0.009364	0.003256	0.002423
Att-Bi-LSTM-Concat	0.012804	0.002010	0.001506
Att-Bi-LSTM-Dot	0.020892	0.009430	0.003832

$$RI = \frac{BiLSTM \text{ 평균제곱근오차} - \text{Attention BiLSTM 평균제곱근오차}}{BiLSTM \text{ 평균제곱근오차}} * 100$$

수식 19

오차 향상률(ERI)은 위 <수식 19>과 같이 계산하였다.

표 4-16 종목별 Bi-LSTM 대비 Attention-Bi-LSTM 모델 평균 오차 향상률

모델 \ 종목	LG U+	KT	SK 텔레콤	평균
Att-Bi-LSTM-Only	16.78%	6.80%	5.73%	9.77%
Att-Bi-LSTM-Concat	22.94%	4.20%	3.56%	10.23%
Att-Bi-LSTM-Dot	37.44%	19.68%	9.06%	22.06%
Att-Bi-LSTM 전체평균	25.72%	10.23%	6.12%	14.02%

LG U+종목에서는 Bi-LSTM 대비 평균적으로 오차가 Att-Bi-LSTM-Only 모델이 0.009364, Att-Bi-LSTM-Concat 모델이 0.012804, Att-Bi-LSTM-Dot 모델이 0.020892만큼 감소되어 오차율이 16.78%, 22.94%, 37.44%로 가장 많이 향상되었고, KT 종목에서는 Bi-LSTM 대비 평균적으로 오차가 Att-Bi-LSTM-Only 모델이 0.003256, Att-Bi-LSTM-Concat 모델이 0.002010, Att-Bi-LSTM-Dot 모델이 0.009430 만큼 감소되어 오차율이 6.80%, 4.20%, 19.68%로 향상되었고, SK텔레콤 종목에서는 Bi-LSTM 대비 평균적으로 오차가 Att-Bi-LSTM-Only 모델이 0.002423, Att-Bi-LSTM-Concat 모델이 0.001506, Att-Bi-LSTM-Dot 모델이 0.003832 만큼 감소되어 오차율이 5.73%, 3.56%, 9.06%로 가장 작게 향상되었다.

평균 Att-Bi-LSTM-Dot 모델이 22.06%로 가장 많은 오차 향상률을 보였고, 두 번째로는 Att-Bi-LSTM-Concat 모델이 10.23%를 보였고, 세 번째로는 Att-Bi-LSTM-Only 모델이 9.77%의 오차 향상률을 보였다. 오차 향상률의 크기는 종목마다 주가 패턴이 다르므로 차이를 보였지만 기존의 Bi-LSTM 모델 대비 Attention Bi-LSTM 모델이 평균적으로 LG U+종목에서는 세 가지 모델 평균 25.72%, KT종목에서는 10.23%, SK텔레콤 종목에서는 6.12%, 주가 예측의 성능 면에서 전체 평균 14.02% 오차 향상률을 보였다.

V. 결 론

2016년 알파고가 이세돌과의 바둑대전에서 이기자 많은 사람들은 머신러닝의 위력을 인정하기 시작했고, 금융 투자 분야에서는 사람이 직접 하던 투자를 머신러닝을 이용하여 금융투자에 적용하기 시작했고 경제학, 통계학, 컴퓨터 공학 등 관련 학계에서도 연구가 꾸준히 진행되고 있다.

주가 데이터는 모두 숫자로 이루어져 있고, 숫자를 잘 계산하기 위한 필요한 수학적 지식들도 잘 갖추어져 있기 때문에 머신러닝(Machine Learning)의 일부인 딥 러닝(Deep Learning)기술은 주식 가격 예측에 적용하기 매우 용이하다. 하지만 다양한 주가 변동 요인들로 인하여 머신러닝기술이 미래의 주가를 정확히 예측하는 데 아직까지 어려움을 겪고 있다.

주가 예측에 도움이 되는 데이터들을 수집해서 딥러닝 기술로 예측 모델을 설계한 후 예측된 결과에 따라 금융투자를 진행해야 하므로 높은 정확도를 우선으로 해야 한다. 따라서 본 연구에서는 무선 이동통신 3사의 2014년부터 2018년까지 5년간 주식 가격 데이터를 야후 파이낸스에서 수집했고, 주가의 기본 데이터인 시가, 고가, 저가, 종가, 수정종가, 거래량과 주가에 예측에 도움을 줄 수 있는 5일 이동평균, 20일 이동평균, 볼린저 밴드 지표, 각 당일 항목들과 종가와 거래량의 비율 등 17개 항목을 추가하여 23개의 항목으로 정규화를 시킨 다음 텐서플로우를 이용하여 Attention Mechanism을 기반으로 양방향 LSTM 순환신경망 모델을 설계하여 세 가지 방법으로 예측하였다.

첫 번째로는 Attention layer의 context vector를 사용함에 있어 context vector만으로 그대로 사용하는 Att-Bi-LSTM-Only 모델로 예측하였고, 두 번째로는 Attention의 context vector와 hidden layer의 마지막 시퀀스 벡터를 연결하여 더 많은 정보를 포함시켜 Att-Bi-LSTM-Concat 모델로 예측하였으며, 세 번째로는 Attention의 context vector와 hidden layer의 마지막 시퀀스 벡터를 요소별 벡터 곱을 하여 데이터를 혼합하여 Att-Bi-LSTM-Dot 모델로 예측하였으며, 정확도를 더욱 높이기 위해 각 종목에 맞게 학습 파라미터를 찾아가며 평균 제곱근 오차로 구하는 실험을 하였다. 결론적으로 기존의 양방향 LSTM 순환신경망보다 Attention Mechanism을 기반으로 한 양방향

LSTM 순환신경망 모델이 평균 제공된 오차를 더욱 감소시켰는데, LG유플러스 종목에서는 평균 25.72%, KT종목에서는 10.32%, SK텔레콤 종목에서는 6.12%을 향상시켰고, 전체 평균으로는 오차를 14.02% 향상시켰다.

본 연구에서는 이동평균을 이용한 볼린저 밴드 지표와 종가와 거래량과 금일 주가데이터 항목간의 비율로만 주가 예측 정확도 향상을 시켰지만, 향후 더 영향력 있는 보조 지표를 연구하고 지도학습과 심층강화학습을 융합시켜 주가 변동 요인의 영향을 최대한 작게 받을 수 있는 모델을 연구하고자 한다.

참 고 문 헌

1. 위키백과, Bollinger Band, https://ko.wikipedia.org/wiki/볼린저_밴드
2. 송유정, 이재원, 이종우, “텐서플로우를 이용한 주가예측에서 가격-기반 입력 피처의 예측 성능 평가”, 한국컴퓨터종합학술대회, KIISE, Vol. 23, No. 11, pp. 625-631, 2017. 11,
3. 신동하, 최광호, 김창복, “RNN과 LSTM을 이용한 주가 예측을 향상을 위한 딥러닝 모델”, 한국정보기술학회논문지, 15(10), 2017.10, 9-16 (8 pages)
4. 이지훈, “딥러닝을 위한 주가 예측 모델”, 숭실대학교, 석사학위논문, 2016
5. 이모세, “효과적인 입력변수 학습을 위한 그래프 이미지 기반 Convolutional Neural Network 모델”, 국민대학교 비즈니스IT전문대학원, 석사학위논문, 2018
6. 주일택, “양방향 순환신경망을 이용한 주식 가격 예측 모형 연구”, 동신대학교 대학원, 박사학위논문, 2018
7. 조륜, “Attention 기반 LSTM 신경망을 이용한 한국어 화행 분석”, 연세대학교, 석사학위논문, 2017.
8. <https://distill.pub/2016/augmented-rnns/>
9. Sutskever, I., Vinyals, O., & Le, Q.V., “Sequence to sequence learning with neural networks.”, In Advances in neural information processing systems (pp. 3104-3112), 2014
10. Bahdanau, D., Cho, K., & Bengio, Y., “Neural machine translation by jointly learning to align and translate. In Proceedings of the International Conference on Learning Representations”, 2015
11. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
12. <https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/>
13. “Deep Learning from Scratch 2”, 사이트 고키 지음, 개앞맵시 옮김. 한빛미디어 P210~381

14. Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 10. 8 Feb 2016
15. 위키백과, 이동평균, https://en.wikipedia.org/wiki/Moving_average
16. <https://wikidocs.net/24996>
17. <https://medium.com/syncedreview/a-brief-overview-of-attention-mechanism-13c578ba9129>
18. <https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>

ABSTRACT

Attention Mechanism Based Using Bi-Directive LSTM Stock Price forecasting Model

Kim Yoon Jae

**Dept. of Department of Computer Science
Graduate School of Information & Communications
Hanbat National University
Advisor : Kim Yoon Joong**

After AlphaGo's victory in the 2016 Go match against Lee Sedol, many people began to recognize the power of machine learning. Even in the financial investment field, machine learning was applied to financial investment programs, and related academic fields such as economics, statistics, and computer engineering are also steadily conducting research using stock price forecasts or stock price predictions based on enhanced learning using the latest deep learning technology.

However, future stock price forecasts are still imprecise due to the fact that stock prices fluctuate in unexpected directions, as a result of changes in corporate sales, operating profit, net income, foreign exchange rate, changes in raw material prices, government policies, development of new technologies, and patents.

Deep Learning technology is very easy to apply to stock price forecasts because stock data are all numeric and the mathematical knowledge required are also available. Unlike other forms of machine learning, Deep Learning Technology learns the feature patterns of input data, so before learning, it tries to understand the characteristics of predictive results according to the feature patterns of input data, and use various sub-indicators to help predict

stock prices. For financial investments, accuracy should be given the highest priority, because the investment proceeds according to the results of the forecast model, which is designed to predict stock prices using deep learning technology and historic stock price data.

Therefore, in this paper, to improve the accuracy of stock price prediction, stock price data for five years from 2014 to 2018, which include open price, high price, low price, close price, modified close price and volume were collected from Yahoo Finance through the three mobile telecommunication companies. With the help of this basic stock price data, we calculate 17 additional items namely, Bollinger band using moving average, each item's close price ratio and volume ratio, making a total of 23 items. Then we train our model using the TensorFlow deep-learning framework.

In this paper, the Attention Mechanism-based bi-directional recurrent neural network model was experimented in three methods. It is a model in which long-term dependency problems are solved with LSTM memory cells inside a recurrent neural network. It has a structure in which hidden layers are added in the reverse direction of data flow, and it focuses on data that has the most influence on output results. This model can forecast future stock prices by learning patterns of stock price changes.

First we experimented with the Att-Bi-LSTM-Only model, which uses only the context vector of the Attention layer for prediction. Secondly, we experimented the Att-Bi-LSTM-Concat model which concatenates the final sequence vector of the hidden layer and the Attention layer's context vector to obtain more information. And thirdly, the Attention layer's context vector and the last sequence vector of the hidden layer were mixed by element-wise vector multiplication in the Att-Bi-LSTM-Dot model.

In order to increase the accuracy, we perform experiments to find the best learning parameters for each stock code and compared the root mean square errors. The experimental results show that the error

improvement rate varies slightly for each stock code because of the different patterns of stock price changes, however, the three Attention-Based Bi-LSTM models compared to the existing Bi-LSTM models achieved a combined average error improvement rate of 25.72% for LG U+, 10.23% for KT, 6.12% for SK Telecom. In terms of the performance of the stock price prediction, the overall average improved by 14.02%.

keyword : Bollinger band, RNN, LSTM, Attention Mechanism, context vector