



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](#)

S&P 500 지수를 이용한 순환신경망의 특성에 대한 연구



2019년 12월

서강대학교 대학원
경영학과
유한정

S&P 500 지수를 이용한 순환신경망의 특성에 대한 연구

지도교수 이 군 희

이 논문을 경영학석사 학위논문으로 제출함

2019년 12월

서강대학교 대학원

경 영 학 과

유 한 정



논문인준서

유한정의 경영학석사 학위논문을 인준함

2019년 12월

주심 김 명 석



부심 이 군 희



부심 이 윤 동



<목차>

국문 초록	v
I. 서론	1
1. 연구의 배경 및 목적	1
1.1. 연구의 배경	1
1.2. 연구의 목적	2
2. 논문의 구성	3
II. 문헌 고찰	4
1. 주가 예측에 대한 고찰	4
1.1. 주가 예측의 이론적 배경	4
1.2. 주가 예측에 대한 선행연구	6
2. 순환신경망에 대한 고찰	8
2.1. 신경망의 이론적 배경	8
2.1.1. 인공신경망(ANN: Artificial Neural Network)	8
2.1.2. 심층신경망(DNN: Deep Neural Network)	13
2.1.3. 순환신경망(RNN: Recurrent Neural Network)	13
2.1.4. 장단기 메모리(LSTM: Long Short-Term Memory)	17
2.1.5. 게이트드 순환 유닛 GRU(Gated Recurrent Unit)	21
2.2. 순환신경망에 대한 선행연구	25
3. 정규화에 대한 고찰	26
3.1. 전처리의 이론적 배경	26
3.2. 정규화의 이론적 배경	29
3.3. 정규화에 대한 선행연구	30
4. 선행연구의 한계점 및 본 연구의 차별점	32



Ⅲ. 실증 분석	34
1. 자료의 구성	34
1.1. 데이터 기간 및 수집	34
1.2. 자료의 생성 및 분할.....	34
2. 연구모형	35
Ⅳ. 실험 및 성능 평가	39
Ⅴ. 결론	43
1. 연구의 결과	43
2. 연구의 시사점 및 한계점	44
참고 문헌	45
영문 초록	54



<표 목차>

[표 1] 데이터 정규화 방법	29
[표 2] 연구 모형의 설정	36
[표 3] 적층 수, 정규화 방법에 따른 MAPE 및 MASE	40



〈그림 목차〉

[그림 1] K 개의 설명변수, n 개의 은닉노드, 1 개의 출력값을 가진 ANN ..	10
[그림 2] RNN 구조 1.....	14
[그림 3] RNN 구조 2.....	14
[그림 4] RNN 의 내부 구조.....	14
[그림 5] 순환신경망의 BPTT	16
[그림 6] LSTM 구조	17
[그림 7] LSTM 내부 구조	18
[그림 8] GRU 구조	22
[그림 9] GRU 내부 구조	23
[그림 10] Han & Kamber(2001), Famili et al(1997) 의 전처리 분류.....	28
[그림 11] 적층 수가 한 개인 LSTM 연구 모형	37
[그림 12] 적층 수가 두 개인 LSTM 연구 모형	37
[그림 13] 적층 수가 한 개인 GRU 연구 모형	38
[그림 14] 적층 수가 두 개인 LSTM 연구 모형	38



국문 초록

S&P 500 지수를 이용한 순환신경망의 특성에 대한 연구

유한정

본 연구에서는 입력값 정규화 방법에 따른 금융 시계열 데이터 예측 성과를 비교해보고자 하였으며, 추가로 적층 수에 따른 예측 성과를 비교하였다. 순환 신경망 중 장단기 메모리 모형과 게이트드 순환 유닛을 사용하여 S&P 500 지수를 예측하였다. 실험 결과, 최소-최대, Z-스코어 방법으로 정규화를 하였을 때 가장 좋은 예측 성과를 보이는 것으로 나타났다. 한편 1층보다 2층을 적층했을 때 대체적으로 예측 성능이 향상되었다. 그러나 예외가 존재하여 절대적으로 적층 수를 늘리는 것이 예측 성능을 향상시킨다고 말하기 어려웠다. 다섯 가지 정규화 방법, 두 가지의 적층 방법, 총 10가지 경우 중에서 가장 낮은 예측 오차를 가지는 경우는 최소-최대 정규화와 두 층을 적층한 경우였다.

주제어 : 주가 예측, 순환신경망, 장단기 메모리 모형, 게이트드 순환 유닛, 정규화, 적층



I. 서론

1. 연구의 배경 및 목적

1.1. 연구의 배경

효과적인 투자를 위해 주가를 예측하는 것은 경제, 수학, 전산 등 여러 분야에서 오랜 관심의 대상이었다. 과거부터 주가를 예측하기 위해 많은 연구가 진행되었지만 여전히 해결되지 않은 난제로 남아있다. 주가는 다양한 설명 변수에 의해 움직이며, 많은 잡음, 비정상성, 그리고 비선형성으로 인해 가격 변동이나 기대 수익을 예측하는 것은 매우 어려운 작업이기 때문이다(Choudhry & Greg, 2008). 또한 주가는 특정시점에 알려진 모든 정보가 이미 반영되어 있으므로 주가를 예측하는 것은 불가능하다는 주장을 기반으로 기존의 분석방법론에 대해 회의적인 시각도 존재한다.

그러나 최근 딥러닝에 의한 주가예측을 통해 시장 평균을 초과하는 수익의 달성이 가능하다는 연구 결과들이 제시되었다. (Tomas & Crhistopher, 2017) 특히 딥러닝 기법 중 하나인 순환신경망은 데이터의 시계열 맥락 정보를 포착할 수 있다는 장점 때문에 분석 방법으로서의 활용이 증가하고 있다. 순환신경망을 이용한 다양한 주가예측 연구가 수행되었으며(Lin et al., 2009; Nelson et al., 2017; Roondiwala et al., 2017; Zhuge et al., 2017), 주가 예측에 대한 순환신경망의 적합성을 증명하였다.



본 논문은 딥러닝 방법 중 순환신경망 방법을 이용하여 다음 날 주가를 예측하여 이에 대한 유용성을 파악하고, 주식 투자를 위한 보조 지표로 사용할 수 있는지에 대한 분석을 진행하고자 한다.

1.2. 연구의 목적

본 연구의 목적은 다음과 같다. 먼저, 데이터 정규화 방법에 따른 순환신경망의 예측 성능을 비교하고자 한다. 딥러닝(심층신경망)에서 데이터의 정규화는 중요한 요소이다(Sola & Sevilla, 1997). 데이터 정규화 방법에 따른 모형의 성능을 비교하는 선행 연구는 다수 있었다. 반면, 금융 시계열 데이터를 다루는 문제에 순환신경망을 적용한 연구에서 입력 데이터의 정규화 방법에 대한 선행연구는 많이 이루어지고 있지 않았다. 따라서 S&P 500 인덱스를 예측하는 문제에 순환신경망 모형을 적용하고, 데이터 정규화 방법에 따른 예측 성과를 비교하고자 한다. 순환신경망으로는 순환 신경망의 대표 모형인 장단기 메모리 모형과 게이트드 순환 유닛을 사용한다. 데이터 정규화 방법으로는 Z-스코어, 정규 점수, 로그 변환, 최대-최소, 로버스트 정규화 방법을 사용하고자 한다. 다음으로, 순환신경망 셀의 적층 수에 따른 예측 성과를 비교하고자 한다. Li et al. (2018)의 연구에 따르면 순환신경망 모형에서 4층 이상의 과도한 적층은 오히려 성능을 저하시킨다는 결과가 있었다. 그러나 적절한 수 이내에서의 적층은 성능을 증진시킨다고 알려져있다. 본 연구를 통해 순환신경망의 적층 수에 따른 주가 예측 성능의 향상 여부를 확인해보고자 한다.



이러한 연구를 통하여, 순환신경망을 활용한 주가 예측에서 가장 우수한 예측 성과를 내는 정규화 방법 및 적층 수를 밝혀, 딥러닝을 활용한 주가 예측 연구에 공헌하고자 한다.

2. 논문의 구성

본 논문은 총 5장으로 구성되어 있으며 각 장의 내용은 다음과 같다.

제 1장은 연구의 배경과 목적, 논문의 구성을 설명하였다. 제 2장은 이론적 배경 및 선행연구에 대한 내용이다. 다양한 주가 예측 방법에 대해 통계적 방법론, 인공지능 방법론에 중점을 두고 방법론과 선행연구를 정리하였다. 인공지능 방법론 중 순환 신경망에 대한 선행연구는 따로 분리하여 정리하였다. 다음으로 데이터 전처리, 그 중 정규화에 초점을 두고 방법에 대한 내용과 선행연구를 정리하였다. 마지막으로 선행 연구의 한계점과 본 연구의 차별점에 대해 언급하였다. 제 3장은 실증 분석에 대한 내용이다. 데이터의 출처 및 기간, 자료의 생성, 자료의 분할에 대한 내용을 포함하는 자료의 구성과 연구 모형에 대한 내용을 담고있다. 제 4장은 실험 및 성능 평가를 기술하였다. 장단기 메모리 모형과 게이티드 순환 유닛을 통한 예측 오차를 제시하였다. 제 5장은 결론을 정리하였다. 연구 결과를 요약하였으며, 본 연구의 시사점 및 한계점을 기술하였다.



II. 문헌 고찰

본 장에서는 주가를 예측하는 다양한 기법을 방법론별로 살펴보고, 관련 문헌을 고찰하였다. 특히 본 연구에서 진행하는 주요 방법론인 순환신경망에 대한 내용은 따로 분리하여 고찰하였다. 또한 신경망에서 중요하게 취급되는 데이터 정규화의 방법과 관련 문헌을 고찰하고자 한다. 마지막으로, 선행 연구들에 대한 고찰을 바탕으로 선행 연구의 한계점에 대해 서술하고자 한다.

1. 주가 예측에 대한 고찰

1.1. 주가 예측의 이론적 배경

주가 예측 방법은 분석하는 방법론을 기준으로 기본적 분석과 기술적 분석으로 구분할 수 있으며(김선웅 & 안현철, 2010), 기술적 분석은 다시 분석방법론을 기준으로 통계적 분석 방법과 인공지능 분석 방법으로 구분할 수 있다.

기본적 분석은 시장에서 형성되는 주식의 가격이 그 주식을 발행한 기업의 가치에 의하여 결정된다는 이론을 배경으로 주식 가격을 예측하게 된다(구승환 & 장성용, 2012). 주로 금리, 통화량, 환율 등의 거시 경제 변수와 회사의 재정 상태를 나타내는 재무제표 비율을 기반으로 한다(Chang and Liu, 2008).



기술적 분석은 ‘주가는 수요와 공급의 원리에 따라 결정되며 수급의 결과는 그래프로 나타난다’는 이론을 배경으로, 그래프 분석을 통해 미래 주가를 예측하는 방법이다(이우식, 2017). 기술적 분석은 과거의 금융 시계열 데이터를 기반으로 한다(Chang and Liu, 2008). 구체적으로 과거 주식의 시가, 고가, 저가, 종가, 거래량 등의 주식 가격을 이용해 산출된 기술적 지표를 사용한다(홍태호 외, 2017). 기술적 분석은 주식의 수요와 공급에 대한 변화가 주식시장의 움직임과 관련되는 변수를 통해 관찰되며, 이러한 변수를 분석함으로써 궁극적으로 미래 주식 가격의 변화를 파악할 수 있다는 믿음에 기초하고 있다(김진화 외, 2011).

과거에는 기술적 지표를 활용하여 주식 가격을 예측하는 방법으로 통계적 분석 방법이 활용되었으며(김진백 & 김유일, 2000), 주로 이동평균 분석법이나 몬테카를로 시뮬레이션 등의 방법으로 주가를 예측하였다.

하지만 시간이 흐름에 따라 통계적 분석 방법으로 시작한 주가 예측은 인공지능을 이용한 방법으로 발전하였으며, 인공지능 분석 방법은 주가 예측 기술 개발에 공헌하였다(이석준 & 오경주, 2011). 인공 지능 분석 방법은 비선형성과 동적인 성질이 두드러지는 문제에서 통계적 방법에 비해 높은 예측 및 분류 정확도를 가지는 경향이 있다. 인공신경망, 지지벡터기계, 의사결정나무 등의 기계학습 기법에서 시작된 인공지능 분석 방법은 최근 컴퓨팅 기술이 발달함에 따라 심층신경망, 순환신경망 등의 딥러닝 기법으로 발전하였다. 딥러닝 기법은 단순하지만 강력한 비선형 모듈로 네트워크를



구성하여 입력 데이터를 추상적인 정보로 변환하는 과정을 반복함으로써 매우 복잡한 문제도 해결할 수 있게 되었다(Lecun et al., 2015).

1.2. 주가 예측에 대한 선행연구

본 연구에서는 주가 예측에 대한 선행연구 중 통계적 분석 방법, 인공지능 분석 방법 위주로 살펴보았다.

우선 통계적 분석 방법에 대한 선행연구는 다음과 같다. 박인찬, 권오진, 김태운 (2009)는 코스피 200 지수의 방향성을 예측하기 위해 모형 내 편차와 모형 외 편차 데이터에 자기회귀 누적 이동평균 모형과 시계열 요소분해 모형을 적용하였다. 연구 결과, 방향성 예측에 요소분해 모형이 ARIMA 모형에 비해 우월하며, 모형에 관계없이 모형 내 편차 데이터를 사용하는 것이 더 유용하다는 결론을 도출하였다. 유진 (2002)은 주식 시장에 가격 제한이 존재할 때, 실제 시장에서의 주가에 대한 VaR(Value at Risk)을 산출하는 방법을 몬테카를로 시뮬레이션 기법을 활용하여 제안하였다. 연구 결과, 가격 제한을 감안하지 않고 전통적인 방식으로 산출했을 때의 VaR는 정확하지 않을 수 있다는 사실을 밝혔다. 또한 VaR의 산출에 몬테카를로 시뮬레이션 기법의 유용성을 확인했다. 이윤선 (2003)은 단기 이동평균선을 활용하여 매매하는 교차매매전략, 단일 이동평균의 등락을 이용한 추세매매전략, 주가와 이동평균선의 교차매매전략에 대한 검증을 시도하였으며, 주가 이동평균 분석의 유용성을 확인하였다. 신동백 (2007)은 환율의 변화를 예측하기 위해 환율 시계열 데이터의 통계수치에 몬테카를로



시뮬레이션 기법을 적용하였다. 연구 결과, 전문예측기관의 예측값과 비교하여 몬테카를로 시뮬레이션 기법의 유용성을 확인하였다. 김현지 & 장우진 (2010)은 한국, 미국, 일본의 주식시장에 대한 이동평균 분석법의 효과성에 대한 검증을 수행하였으며, 5일 이동 평균과 20일 이동평균으로 조합된 투자 전략이 모든 국가에서 효과적이라는 점을 확인하였다.

인공지능 분석 방법의 경우 과거에는 주가 예측을 위해 지지백터기계를 주로 사용하였으나, 최근 딥러닝과 강화학습을 이용한 주가 예측 연구에서 괄목할만한 성과를 보이며 다양한 기법들의 활용 가능성을 볼 수 있었다. 인공지능 분석 방법을 활용한 주가 예측 선행연구는 다음과 같다. 이형웅 (2008)은 코스피 지수 등락 예측을 위해 구축된 로지스틱 회귀모형, 인공신경망, 지지백터기계 모형을 결합한 모형을 사용하였다. 연구 결과, 인공지능 분석 방법이 주식시장 예측에 유용함을 확인하였다. Ince & Trafalis (2004)는 미국의 나스닥 종목들의 가격 변화를 예측하기 위해 지지백터기계 알고리즘과 주성분 분석을 활용하여 기술적 지표를 분석하였다. 연구 결과, 지지백터기계가 주가 예측에 유용함을 확인하였다. Huang et al. (2005)의 연구에서는 지지백터기계를 사용하여 일본의 닛케이 225 지수의 주간 가격 변동을 연구하였다. 그 결과, 선형판별분석, 이차판별분석의 분류 성능보다 지지백터기계 및 지지백터기계와 다른 분류기를 합친 복합 분류기의 분류 성능이 보다 우수함을 확인하였다. 허준영 & 양진용 (2015)은 주가 예측을 위해 재무 정보에 지지백터기계 모형을 적용하였다. 연구 결과, 주가 예측에 대한 지지백터기계의 유용성을 확인하였다. Lai et al. (2009)은 대만 거래소에



상장된 종목들의 가격 등락을 예측하기 위해 인공지능 분석 방법 중 하나인 퍼지 결정 트리와 유전 알고리즘을 활용하여 기술적 지표를 분석하였고, 그 결과 의사결정나무가 주식시장 예측에 적합함을 확인하였다. 안성원 & 조성배 (2010)는 코스피 종목의 가격 등락을 예측하기 위해 텍스트 마이닝을 활용하여 해당 종목과 관련된 뉴스 기사를 분석했고, 텍스트 마이닝 기법이 주가 예측에 유용함을 확인하였다. Kim & Han (2000)은 코스피에 상장된 종목들의 일별 가격 등락을 예측하기 위해 유전 알고리즘 기반의 인공신경망을 활용하여 기술적 지표를 분석했고, 인공신경망이 주식시장 예측에 유용함을 확인하였다. Huang et al. (2008)은 인공신경망과 군집 분석, 지지백터기계 등의 알고리즘으로 대만 거래소와 코스피에 상장된 종목들의 가격 등락을 예측하였으며, 그 결과 인공지능 분석 방법이 주식시장 예측에 유용함을 확인하였다. 이모세 & 안현철 (2018)은 주가의 방향을 예측하기 위해 합성곱신경망 분류기를 사용하였다. 연구 결과, 합성곱신경망 분류기가 통계적으로 유의한 수준의 예측 정확도를 나타내어, 주가 방향성 예측에서 합성곱신경망의 유용성을 확인하였다.

2. 순환신경망에 대한 고찰

2.1. 신경망에 대한 이론적 배경

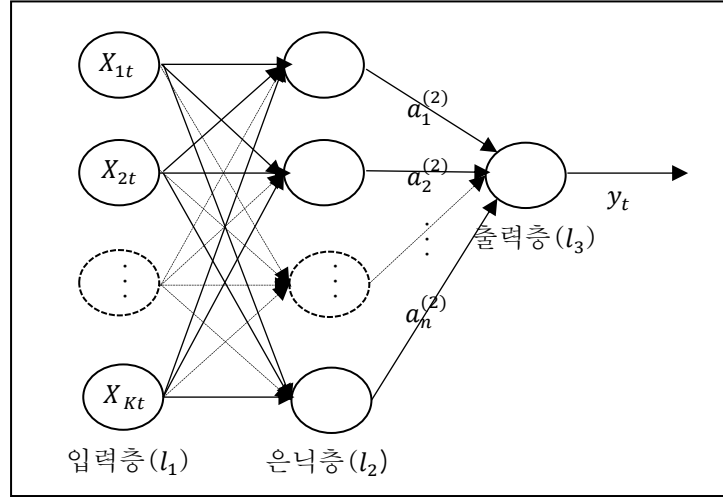
2.1.1. 인공신경망(ANN: Artificial Neural Network)



딥러닝은 인공신경망을 기초로 하고 있다. 이 인공신경망은 사람의 신경망 원리와 구조에서 영감을 얻은 기계학습 알고리즘이다. 특정 뉴런에 신호가 전달되면 시냅스를 통해 해당 뉴런에서 다른 뉴런으로 신호를 전달하고 이 과정을 통해 뇌는 새로운 것을 학습한다. 뉴런은 일정 자극 이상이 되면 다음 뉴런에게 신경을 전달하며, 가까이 있는 뉴런에 대해 흥분성으로, 멀리 있는 뉴런에 대해 억제성으로 동작한다. 다시 말해 거리에 따라 신경을 달리 받아들인다는 뜻이다. 인공신경망에서는 뉴런을 노드, 시냅스는 가중치로 대체하여 네트워크를 구성한다. 노드는 입력값을 가중치와 곱하여 시그마에 입력하고, 활성화 함수 f 를 통해 다음 노드로 전달해주는 역할을 한다. 뇌에서 특정 임계치를 넘을 경우에만 신경이 전달되는 시스템을 활성화 함수로 구현하였으며, 거리에 대한 가중을 주는 시스템을 노드 간 연결선에 가중을 주는 방식으로 구현하였다.

t 가 시간, K 가 설명변수의 수라고 할 때, X 라는 설명 변수로 Y 라는 목적 변수를 예측하기 위한 데이터셋을 $\{X_t = (X_{1t}, \dots, X_{Kt})', y_t, t = 1, \dots, T\}$ 로 설정하고, K 개의 설명 변수, n 개의 은닉 노드를 가진 인공신경망 구조를 도식화 하면 아래의 [그림 1]과 같다.





[그림 1] K개의 설명변수, n개의 은닉노드, 1개의 출력값을 가진 ANN

식으로 표현하면 식(1), ..., 식(6)과 같다.

$$a_1^{(2)} = f(W_{11}^{(1)} X_{1t} + W_{21}^{(1)} X_{2t} + \dots + W_{K1}^{(1)} X_{Kt} + b_1^{(1)}) \quad (1)$$

$$a_2^{(2)} = f(W_{12}^{(1)} X_{1t} + W_{22}^{(1)} X_{2t} + \dots + W_{K2}^{(1)} X_{Kt} + b_2^{(1)}) \quad (2)$$

$$a_n^{(2)} = f(W_{1n}^{(1)} X_{1t} + W_{2n}^{(1)} X_{2t} + \dots + W_{Kn}^{(1)} X_{Kt} + b_n^{(1)}) \quad (3)$$

$$y_t = a_1^{(3)} = f(W_{11}^{(2)} a_1^{(2)} + W_{21}^{(2)} a_2^{(2)} + W_{n1}^{(2)} a_n^{(2)} + b_1^{(2)}) \quad (4)$$

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)} \quad (5)$$

$$a^{(l+1)} = f(z^{(l+1)}) \quad (6)$$

$W_{ij}^{(l)}$ 는 레이어 l의 유닛 i와 레이어 l+1의 유닛 j를 연결시켜주는 파라미터(또는 가중치), $b_j^{(l)}$ 는 레이어 l+1의 유닛 j의 바이어스, $a_i^{(l)}$ 는 레이어 l의 유닛 i의 활성화 함수 출력값을 의미한다. f 는 활성화 함수로, 시그모이드와 하이퍼볼릭 탄젠트 등 곡선 형태의 활성화 함수 혹은 렐루(ReLU), 리키 렐루(Leaky ReLU) 등 직선 형태의 활성화 함수를

사용한다. 언급한 네 가지 활성화 함수는 아래의 식(7), 식(8), 식(9), 식(10)과 같다.

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (7)$$

$$\tanh(x) = \frac{1-e^{-x}}{1+e^{-x}} = \frac{2}{1+e^{-2x}} - 1 \quad (8)$$

$$\text{ReLU}(x) = \max(0, x) \quad (9)$$

$$\text{Leaky ReLU}(x) = \max(0.1x, x) \quad (10)$$

[그림 1]과 식(1) - 식(6)에서 알 수 있듯이 인공신경망은 두 가지 단계로 진행된다. 먼저, 입력 노드에서 수신된 설명 변수의 선형 결합값이 비선형 함수(활성화함수)를 통과한 값($a_1^{(2)}, a_2^{(2)}, a_3^{(2)}$)을 은닉 노드로 전송한다. 다음으로, 이전 단계의 결과값을 선형 결합한 후, 비선형 함수를 통과시킨 값($a_1^{(3)}$)을 출력 노드로 전송한다. 이 값이 최종 예측 값이 된다.

인공신경망은 일련의 배치를 학습시켜서 반복적으로 가중치(W)와 바이어스(b)를 업데이트 시킨다. 최적의 가중치와 바이어스를 찾기 위해 오차를 최소화하는 방향으로 가중치와 바이어스를 증가시키거나 감소시킨다. 오차 함수(E)로는 평균 절대 오차(MAE: mean absolute error), 평균 제곱 오차(MSE: mean squared error) 등 다양한 오차 함수를 사용할 수 있다. 가중치와 바이어스의 갱신 진행 방향은 식(11), 식(12)와 같다.

$$W^{(l)} = W^{(l)} - \alpha \frac{\partial E}{\partial W^{(l)}} \quad (11)$$

$$b^{(l)} = b^{(l)} - \alpha \frac{\partial E}{\partial b^{(l)}} \quad (12)$$

식(11)과 식(12)의 α 는 학습률을 의미한다.



한편 $\{B_1, B_2, \dots, B_m\}$ 은 데이터 $(\{X_t = (X_{1t}, \dots, X_{Kt})', y_t, t = 1, \dots, T\})$ 를 m 개의 배치를 가지도록 랜덤 분할한 것을 의미한다. 랜덤 분할한 결과는 $\{(X_t, y_t), t \in B_j, j = 1, \dots, m\}$ 와 같이 표기할 수 있다. 예를 들어, $T = 200$ 이고, 배치 사이즈 20을 선택한다면 $m = 10$ 배치를 가지게 된다. 이 때, $\{t = 1, \dots, T\}$ 를 $\{B_1, B_2, \dots, B_m\}$ 로 분할한 것으로 진행되는 학습은 하나의 에포크를 형성한다. 학습 절차 동안 다양한 랜덤 분할을 가지고 여러 번 반복하는데, 이 반복 횟수를 에포크라고 한다.

그런데 이러한 인공신경망은 여러 문제점을 가지고 있었다. 먼저 학습 과정에서 파라미터(가중치, 바이어스)의 최적값을 찾기 어려운 문제가 있었다. 다음으로 학습 과정에서 학습 데이터에 과적합되어 새로운 데이터를 받았을 때, 예측 성과가 떨어지는 문제가 있었다. 국소 최적점을 최적점으로 인식하여 학습을 멈추는 문제가 발생하기도 하였다. 마지막으로 은닉층이 증가할수록 연산량이 급격히 증가하여 최적화 과정에 부담이 되며, 학습 시간이 느리다는 단점이 있었다.

그러나 과적합, 국소 최적점에서 학습을 멈추는 문제가 드롭-아웃과 같은 정칙화 방법, 가중치 초기화 방법 등의 발전으로 개선되었고, 시간 소요 문제가 그래픽 처리 장치의 등장으로 개선되었다.



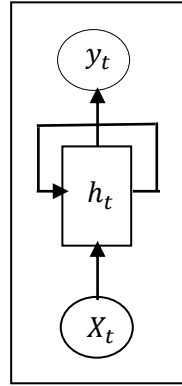
2.1.2. 심층신경망(DNN: Deep Neural Network)

인공신경망 기법의 문제들이 해결되면서, 알고리즘 내의 은닉층을 늘릴 수 있었는데, 이러한 배경에서 등장하는 방법이 심층신경망이다. 심층신경망은 은닉층을 두 개 이상 지닌 학습 방법을 일컫는다. 심층신경망은 인공신경망에 비해 설명 변수와 목적 변수 사이의 복잡한 비선형 관계를 더욱 잘 포착하였다. 항상 그런 것은 아니지만 일반적으로 네트워크가 깊을수록 더 나은 성능을 보이는 것으로 알려져 있다. 대표적인 심층신경망 응용 알고리즘으로 합성곱 신경망, 순환신경망, 장단기 메모리 모형, 게이트 순환 유닛 등이 있다. 그러나 심층신경망은 시간적 맥락 정보를 포착하는데 한계가 있어 시계열 데이터에 대해서 취약하다는 단점을 가진다.

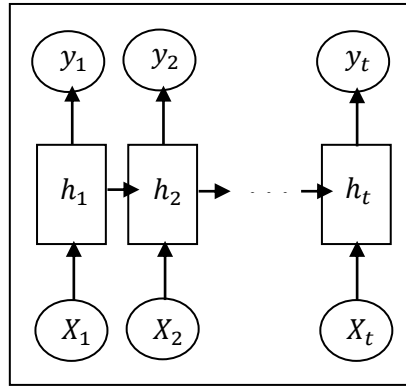
2.1.3. 순환신경망(RNN: Recurrent Neural Network)

순환신경망은 Elman(1990)에 의해 제안되었으며, 기존의 신경망과 달리 이전 시점의 출력을 현재 시점의 입력으로 공급하는 순환구조를 통해(Zheng et al., 2019), 시계열 데이터에 존재하는 시간적 맥락 정보를 포착하도록 설계되었다(Liu & Sullivan, 2019). 순환신경망의 기본적인 구조는 아래의 [그림 2], [그림 3]과 같다. [그림 2]을 펼치면 [그림 3]과 같아진다. [그림 2]와 [그림 3]에서 이전 시점의 출력을 현재 시점의 입력으로 공급하기 위해 은닉층에서 순환하는 구조를 확인할 수 있다.



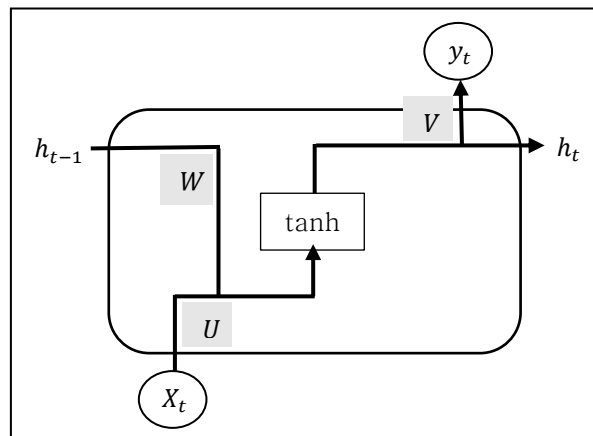


[그림 2] RNN 구조1



[그림 3] RNN 구조2

[그림 2]와 [그림 3]에서 X_t 는 t 시점의 입력값을 의미하고, h_t 는 t 시점의 은닉층으로부터 계산된 은닉값을 의미하며, y_t 는 t 시점의 출력값을 의미한다. 순환신경망은 각 시점의 입력값(X_t)으로부터 산출된 은닉값(h_t)을 다음 시점의 은닉값(h_{t+1})의 산출에 활용하여 시계열 데이터에 존재하는 시간적 맥락 정보를 포착한다. 아래의 [그림4]는 순환신경망의 핵심이라 할 수 있는 은닉층의 구조를 도식화하고 있다.



[그림 4] RNN의 내부 구조

순환신경망은 t 시점의 입력값(X_t)과 $t-1$ 시점의 은닉값(h_{t-1})을 활용하여 현재 시점의 은닉값(h_t) 및 출력값(y_t)을 산출하게 된다. 현재 시점의 은닉값은 이전 시점의 은닉값과 현재 시점의 입력값에 하이퍼볼릭 탄젠트 함수를 적용하여 산출되고, 현재 시점의 출력값은 산출된 현재 시점의 은닉값을 활용하여 산출되게 된다. 아래의 식(13), 식(14)는 t 시점의 은닉값에 대한 수식과 출력값에 대한 수식이다.

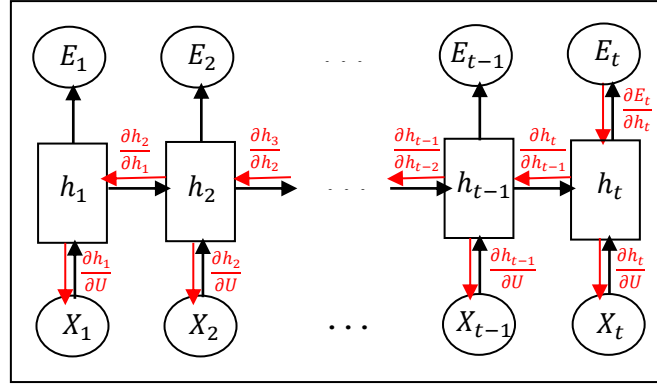
$$h_t = \tanh(W \times h_{t-1} + U \times X_t + b_h) \quad (13)$$

$$y_t = V \times h_t + b_y \quad (14)$$

이 때, W 는 h_t 와 선형결합하게 되는 가중치이며, U 는 X_t 와 선형결합하게 되는 가중치이다. V 는 h_t 와 선형결합하게 되는 가중치이다.

일반적으로 순환신경망은 BPTT(Back-Propagation Through Time) 기법을 활용하여 학습한다(Williams & Peng, 1990). BPTT의 원리는 기본적으로 오류역전파 알고리즘과 같지만, 은닉층의 가중치가 서로 공유되어 학습되는 동안 누적되어 조정된다는 차이가 있으며(Werbos, 1990), 자세한 내용은 아래의 [그림 5], 식(15)와 같다.





[그림 5] 순환신경망의 BPTT

위 [그림 5]는 BPTT의 과정을 도식화한 것이다. 손실함수 E 는 실제값과 예측값 사이의 차이 $\hat{y}_t - y_t$ 를 의미하고, U 는 식(13)의 W 와 같다. 파라미터 갱신은 이 절의 1.1에서 언급한 것처럼 E 값을 최소화시키는 방식으로 식(11)의 W 를 U 로 대체한 $U = U - \alpha \frac{\partial E}{\partial U}$ 과 같이 진행된다. $\frac{\partial E}{\partial U}$ 는 [그림 5]에서 볼 수 있듯이 네트워크의 역방향으로 계산되며 수식은 아래의 식(15)와 같다.

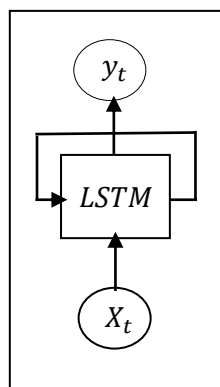
$$\begin{aligned} \frac{\partial E_t}{\partial U} &= \frac{\partial E_t}{\partial h_t} \times \frac{\partial h_t}{\partial U} + \frac{\partial E_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_{t-1}} \times \frac{\partial h_{t-1}}{\partial U} + \frac{\partial E_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_{t-1}} \times \frac{\partial h_{t-1}}{\partial h_{t-2}} \times \dots \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_2}{\partial U} + \frac{\partial E_t}{\partial h_t} \times \\ &\frac{\partial h_t}{\partial h_{t-1}} \times \frac{\partial h_{t-1}}{\partial h_{t-2}} \times \dots \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_2}{\partial h_1} \times \frac{\partial h_1}{\partial U} \end{aligned} \quad (15)$$

그런데 식(15)는 시퀀스가 길어질수록 $\frac{\partial h_t}{\partial h_{t-1}} \times \dots \times \frac{\partial h_2}{\partial h_1}$ 가 길어지는 구조를 가지고 있기 때문에 시퀀스가 긴 경우, $\frac{\partial h_t}{\partial h_{t-1}}, \dots, \frac{\partial h_2}{\partial h_1}$ 의 값에 따라 $\frac{\partial E_t}{\partial U}$ 가 폭발하거나 사라지는 문제가 발생할 수 있다. 곱셈이 거듭되면서 $\frac{\partial h_t}{\partial h_{t-1}}, \dots, \frac{\partial h_2}{\partial h_1}$ 가 1보다 큰 경우, 값이 폭발하고, $\frac{\partial h_t}{\partial h_{t-1}}, \dots, \frac{\partial h_2}{\partial h_1}$ 가 1보다 작은 경우 값이 사라지기 때문이

다. 결국 $U - \alpha \frac{\partial E}{\partial U}$ 가 영똥한 값으로 수렴하게 된다. 이러한 이유로 순환신경망은 일반적으로 짧은 시퀀스 모델링에서, 즉 t 의 개수가 적을 때 잘 수행된다. 하지만 시퀀스가 길어지면 장기 의존성 문제가 발생할 수 있다(Bengio et al., 1994; Güler et al., 2005).

2.1.4. 장단기 메모리 (LSTM: Long Short-Term Memory)

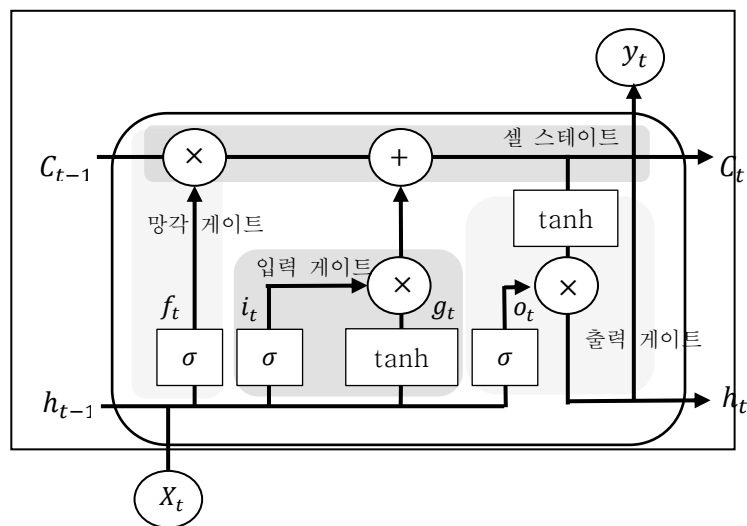
장단기 메모리 모형은 Hochreiter & Schmidhuber (1997)에 의해 제안된 모형으로 순환신경망의 한 종류이다. 등장 배경은 다음과 같다. 순환신경망은 시계열 예측에 있어서 우수한 성능을 보인다는 것으로 증명되었지만, 시계열의 길이를 늘렸을 때 장기 의존성 문제가 발생하기도 하였다(Bengio et al., 1994; Tian et al., 2018). 장단기 메모리 모형은 이러한 장기 의존성 문제를 개선하기 위해 고안된 알고리즘이다(Hochreiter & Schmidhuber, 1997). 장단기 메모리 모형 구조를 도식화하면 [그림 6] 같다.



[그림 6] LSTM 구조



장단기 메모리 모형의 구조는 순환신경망과 유사하지만 은닉층에서 차이가 있다. 장단기 메모리 모형의 은닉층 내에 메모리 셀(memory cell)이라는 기본 구조를 가지고 있으며(Liu & Sullivan, 2019), 메모리셀의 핵심은 셀 스테이트이다. 셀 스테이트가 하나의 컨베이어 벨트와 같이 전체 체인을 관통하면서 정보를 큰 변함없이 다음 단계로 전달하여 시간적 맥락 정보를 기억한다. 장단기 메모리 모형 메모리셀의 내부 구조는 아래의 [그림 7]과 같다.



[그림 7] LSTM 내부 구조

장단기 메모리 모형은 [그림 7]의 각 게이트(망각 게이트, 입력 게이트, 출력 게이트)를 이용하여 어느 정보를 얼마만큼 메모리셀 상태(C_t)에 포함시킬지 결정하게 되며(Graves & Schmidhuber, 2005), 게이트에 대한 자세한 내용은 다음과 같다.

첫 번째로 망각 게이트는 과거 정보를 잊기 위해 만들어진 게이트(Gers et al., 1999)이다. 망각 게이트는 이전 시점의 메모리셀 상태(C_{t-1})를 현재 시점의 메모리셀 상태(C_t)로 얼마나 전달할지 결정하는 역할을 한다. 망각 게이트는 f_t 값에 의해 제어되며, f_t 를 출력값으로 내보내게 된다. f_t 값은 이전 시점의 은닉값(h_{t-1})과 현재 시점의 입력값(X_t)에 대해 시그모이드 함수를 취해 얻게 된다. 시그모이드 함수의 출력 범위가 0에서 1 사이이므로 f_t 의 값이 0이라면 이전 시점의 정보는 완전히 잊어버리게 되고, 값이 1이라면 이전 시점의 정보를 온전히 기억하게 된다. f_t 에 대한 수식은 아래의 식(16)와 같다.

$$f_t = \sigma(W_f \times h_{t-1} + U_f \times X_t + b_f) \quad (16)$$

식(16)에서 W_f 는 X_{t-1} 과 선형결합하게 되는 가중치이며, U_f 는 X_t 와 선형결합하게 되는 가중치이다.

두 번째로 입력 게이트는 새로운 정보를 기억하기 위해 만들어진 게이트이며 자세한 내용은 다음과 같다. 입력 게이트는 i_t 값에 의해 제어되며, i_t 과 g_t 의 곱을 출력값으로 내보내게 된다. i_t 값은 f_t 값과 마찬가지로 이전 시점의 은닉값과 현재 시점의 입력값에 대해 시그모이드 함수를 취해 얻게 되며, g_t 를 현재 시점의 셀 상태로 얼마나 전달할지 결정하는 역할을 한다. g_t 값은 이전 시점의 은닉값과 현재 시점의 입력값에 대해 하이퍼볼릭 탄젠트 함수를 취해 얻게 된다. i_t , g_t 에 대한 수식은 차례대로 아래의 식(17), 식(18)과 같다.



$$i_t = \sigma(W_i \times h_{t-1} + U_i \times X_t + b_i) \quad (17)$$

$$g_t = \tanh(W_g \times h_{t-1} + U_g \times X_t + b_g) \quad (18)$$

식(17)에서 W_i 는 h_{t-1} 과 선형결합하게 되는 가중치이며, U_i 는 X_t 와 선형결합하게 되는 가중치이다. 또한 식(18)에서 W_g 는 h_{t-1} 과 선형결합하게 되는 가중치이며, U_g 는 X_t 와 선형결합하게 되는 가중치이다.

마지막으로 출력 게이트는 현재 시점의 은닉값과 출력값(y_t)을 출력하기 위해 현재 시점의 메모리셀 상태의 어느 부분을 채택할지 선택하기 위해 만들어진 게이트이며 자세한 내용은 다음과 같다. 출력 게이트는 o_t 값에 의해 제어되며, o_t 와 $\tanh(C_t)$ 의 곱을 출력값으로 내보내게 된다. o_t 값은 f_t , i_t 와 마찬가지로 이전 시점의 은닉값과 현재 시점의 입력값에 대해 시그모이드 함수를 취해 얻게되며, $\tanh(C_t)$ 를 최종 출력값에 얼마나 반영할지 결정하는 역할을 한다. o_t , 망각 게이트와 입력 게이트를 거친 현재 시점의 메모리셀 상태, 현재 시점의 은닉값과 출력값에 대한 수식은 차례대로 아래의 식(19), 식(20), 식(21)과 같다.

$$o_t = \sigma(W_o \times h_{t-1} + U_o \times X_t + b_o) \quad (19)$$

$$C_t = f_t \times C_{t-1} + i_t \times g_t \quad (20)$$

$$h_t = o_t \times \tanh(C_t) \quad (21)$$

식(19)에서 W_o 는 h_{t-1} 과 선형결합하게 되는 가중치이며, U_o 는 X_t 와



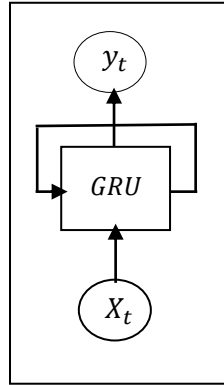
선형결합하게 되는 가중치이다.

모든 정보를 덮어쓰는 구조를 가진 기존의 순환신경망과 달리, 장단기 메모리 모형은 각 게이트들을 이용하여 선택적으로 정보를 채택하고 유지할 수 있다. 따라서 시계열의 길이가 길어져도 장단기 메모리 모형이 학습 단계에서 검출한 중요한 정보는 장기 의존성 문제가 발생하지 않고 잘 유지될 수 있다(Chung et al., 2014).

2.1.5. 게이트드 순환 유닛(GRU: Gated Recurrent Unit)

게이트드 순환 유닛은 Cho et al. (2014)가 제안한 모형으로 순환신경망의 한 종류이다. 장단기 메모리 모형은 메모리셀 상태 구조를 이용하여 장기 의존성 문제를 개선하였지만 모델의 복잡성으로 인해 연산량이 크다는 문제점이 있다(Shin & Sung, 2016). 게이트드 순환 유닛은 이러한 연산량의 문제를 개선하기 위해 장단기 메모리 모형의 구조를 단순화시킨 장단기 메모리 모형의 변형 모델 중 하나이다(Cho et al., 2014) 게이트드 순환 유닛의 구조는 아래의 [그림 8]과 같다.



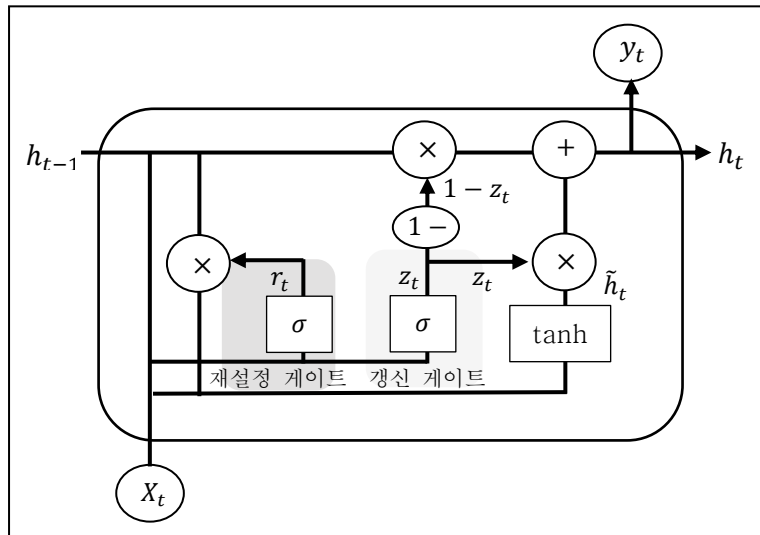


[그림 8] GRU 구조

게이티드 순환 유닛은 장단기 메모리 모형과 구조적으로 유사하지만, 두 가지 측면에서 차이점을 가진다. 먼저 장단기 메모리 모형은 총 세 개의 게이트(망각 게이트, 입력 게이트, 출력 게이트), 게이티드 순환 모형은 총 두 개의 게이트(갱신 게이트와 채설정 게이트)로 구성되어 있다. 장단기 메모리 모형의 망각 게이트와 입력 게이트를 결합하여 1개의 갱신 게이트를 만들었다고 보면 된다. 다음으로 장단기 메모리 모형 상에 존재하는 메모리 셀 상태(C_t)와 은닉값(h_t)을 게이티드 순환 유닛에서는 하나의 은닉값으로 대체하여 사용한다는 차이점이 있다(송경환, 2018). 종합하면 게이티드 순환 유닛은 이전의 정보를 개별적인 내부 기억 장소(셀 스테이트)에 담아두지 않으며, 한 은닉 상태에서 다른 은닉 상태로의 갱신에 필요한 게이트 수가 더 적다. 이러한 이유로 서로 다른 시간의 정보를 적응적으로 포착한다는 장단기 메모리 모형의 장점은 그대로 유지하면서, 성능 또한 장단기 메모리 모형에 뒤지지 않고, 학습 속도는 장단기 메모리 모형보다 빠르다(Chung et al,



2014). 게이트드 순환 유닛 내부 구조에 대한 자세한 내용은 아래 그림과 같다.



[그림 9] GRU 내부 구조

게이트드 순환 유닛은 갱신 게이트와 재설정 게이트를 이용해서 이전 시각의 정보를 현재 상태에 반영하며, 현재 시점의 은닉값을 산출하기 위해 중간 은닉값(\tilde{h}_t)을 사용한다.

우선 갱신 게이트는 장단기 메모리 모형의 망각 게이트와 입력 게이트를 합친 역할을 수행하며, 이전 시점과 현재 정보의 최신화 비율을 결정한다. 갱신 게이트는 z_t 값에 의해 제어되며, z_t 는 현시점 정보의 양을 결정하고, $1 - z_t$ 는 직전 시점의 은닉층 정보에 곱해준다. z_t 와 $1 - z_t$ 순서대로 장단기 메모리 모형의 입력, 망각 게이트와 유사하다. 궁극적으로 은닉값을 산출하기 위해 이전 시각의 은닉값(h_{t-1})과 현재 시점의 중간은닉값(\tilde{h}_t)을 어떻게

조합할지를 결정한다. z_t 값은 이전 시점의 은닉값과 현재 시점의 입력값(X_t)에 대해 시그모이드 함수를 취한 값을 출력하며, z_t 에 대한 수식은 아래의 식(22)와 같다.

$$z_t = \sigma(W_z \times h_{t-1} + U_z \times X_t + b_z) \quad (22)$$

식(22)에서 W_z 는 h_{t-1} 과 선형결합하게 되는 가중치이며, U_z 는 X_t 와 선형결합하게 되는 가중치이다.

재설정 게이트는 r_t 값에 의해 제어되며, 과거의 정보를 적당히 리셋시키는 것이 목적이다. 현재 시점의 중간은닉값을 산출할 때 이전 시점의 은닉상태가 현재 은닉상태에 얼마나 반영되는지를 제어한다. r_t 는 z_t 와 마찬가지로 이전 시점의 은닉값과 현재 시점의 입력값에 대해 시그모이드 함수를 취한 값을 출력으로 내보내며, r_t 에 대한 수식은 아래의 식(23)과 같다.

$$r_t = \sigma(W_r \times h_{t-1} + U_r \times X_t + b_r) \quad (23)$$

식(23)에서 W_r 는 h_{t-1} 과 선형결합하게 되는 가중치이며, U_r 는 X_t 와 선형결합하게 되는 가중치이다.

최종적으로 재설정 게이트와 갱신 게이트를 거쳐 중간 은닉값과 은닉값이 산출된다. 중간 은닉값은 재설정 게이트를 거친 이전 시점의 은닉값과 현재 시점의 입력값에 하이퍼볼릭 탄젠트 함수를 취한 값이고, 은닉값은 중간은닉값과 이전 시점의 은닉값이 갱신 게이트를 거친 값이다. 아래의 식(24), 식(25)는 차례대로 중간은닉값, 은닉값에 대한 수식이다.



$$\tilde{h}_t = \tanh(r_t \times W_h \times h_{t-1} + U_h \times X_t + b_h) \quad (24)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (25)$$

식(24)에서 W_h 는 h_{t-1} 과 선형결합하게 되는 가중치이며, U_h 는 X_t 와 선형결합하게 되는 가중치이다.

2.2. 순환신경망에 대한 선행연구

순환신경망과 관련된 국내외 선행연구 조사 결과 대부분 주식시장 예측에 활용된 것으로 나타났다. 자세한 내용은 아래와 같다.

먼저 국내에서 수행된 순환신경망 모형에 관한 선행연구를 살펴보면 다음과 같다. 주일택 & 최승호 (2018)는 종가를 예측하는 문제에서 구글의 일별 주가 및 거래량 데이터에 장단기 메모리 모형을 적용하였다. 연구 결과, 주가 예측에 장단기 메모리 모형이 적합함을 확인하였다. 김성수 & 홍광진 (2017)은 KOSPI 200 선물 지수를 예측하기 위해 순환신경망을 활용하였다. 연구 결과, 제안한 모델이 주가 예측에 유용함을 입증하였고, 기존의 기술적 분석과 결합하여 로보어드바이저 서비스를 통한 수익성을 창출할 수 있다고 주장하였다. 성노운 & 남기환 (2017)은 금값을 예측하기 위해 뉴스 기사 데이터와 거시경제 변수를 예측 변수로 두고, 장단기 메모리 모형과 텍스트 마이닝 기법을 활용하였다. 연구 결과, 경제 변수와 텍스트 데이터가 함께 분석될 때 예측 성능이 향상되는 것을 확인하였으며, 금값 예측에서 장단기 메모리 모형의 유용성을 확인하였다.



해외에서 수행된 순환신경망 관련 선행연구를 살펴보면 다음과 같다. Lin et al. (2009)은 S&P 500 종목을 대상으로 종가를 예측하는 문제에서 일별 가격 및 거래량 데이터에 순환신경망 모형 중 하나인 ESN (Echo State Network)과 주성분 분석을 적용하였다. 연구 결과, 주가 예측에 순환신경망이 주가 예측에 유용함을 확인하였다. Roondiwala et al. (2017)은 니프티 50 지수를 대상으로 한 가격 및 거래량 예측 문제에서, 장단기 메모리 모형을 적용하였다. 연구 결과, 주가 예측에 순환신경망이 유용함을 확인하였다. Zhuge et al. (2017)은 상해종합주가지수를 예측하기 위해 데이터마이닝 기법을 활용하여 주식관련 온라인 게시물, 가격 및 거래량 등을 분석한 후, 장단기 메모리 모형을 적용하였다. 연구 결과, 주가 예측에 있어서 장단기 메모리 모형이 유용함을 확인하였다.

3. 정규화에 대한 고찰

3.1. 전처리의 이론적 배경

정규화는 다양한 데이터 전처리 방법 중 하나이다. 기계학습 알고리즘은 상당히 복잡한 데이터베이스에서 지식을 추출하기 때문에 일반적으로 데이터의 품질에 따라 기계학습 알고리즘의 성공 여부 및 성능의 정도가 결정될 수 있다(Sola and Sevilla, 1997). 관련이 없는 외부 정보가 포함된 데이터를 사용하면 정확도가 떨어지고 신뢰할 수 없는 결과값을 얻을 수



있다(Kotsiantis et al., 2006). 따라서 데이터 전처리는 기계학습 프로세스에서 중요한 단계로 취급된다.

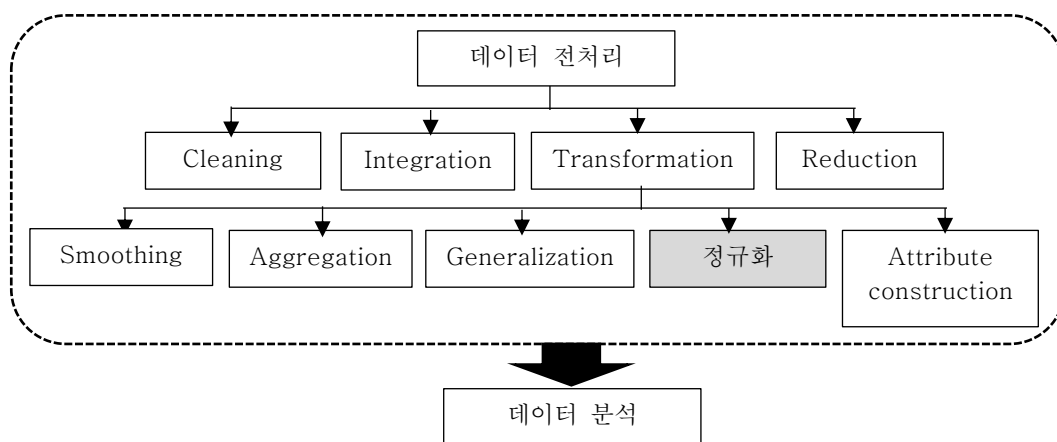
데이터 전처리는 실제 데이터에 대해 모델링 이전에 수행된 모든 조치를 일컫는다. 데이터 전처리는 데이터 정리, 통합, 변환, 축약을 포함한다(Han and Kamber, 2001; Famili et al., 1997). 데이터 정리는 노이즈와 데이터의 부적합성을 제거하기 위해 사용한다. 데이터 통합은 다양한 정보원으로부터 얻은 데이터들을 웨어하우스와 같은 데이터 저장소로 모아 통합하는 것을 의미한다. 주로 데이터 정리와 데이터 통합은 데이터 웨어하우스에 보관하기 위한 데이터를 준비하기 위해 사용한다. 데이터 축약은 중복되는 특징 군집을 통합하고, 제거하기 위해 사용한다(Han & Kamber, 2001). 데이터 변환은 원데이터를 가공하여 데이터마이닝에 적절한 형태로 변환하는 것을 의미한다. 예를 들어, 정규화와 같은 기법을 적용하여 목적에 맞는 데이터로 변환한다.

데이터 변환은 다시 다섯 가지 카테고리로 세분화할 수 있다. 먼저, 첫번째 카테고리는 스무딩(smoothing)으로 데이터의 노이즈를 제거하기 위해 사용되며, 비닝(bining), 선형회귀, 군집화를 포함한다. 두번째 카테고리는 집계(aggregation)로 데이터를 요약하고, 큐브를 생성하는 방법이다. 세번째 카테고리는 데이터의 일반화로 저수준 혹은 원시적 데이터를 개념 기반 계층 구조를 활용하여 고수준 데이터로 대체한다. 네번째 카테고리는 정규화로 데이터를 특정 범위 안으로 한정하기 위해 사용한다. 예를 들어 -1 이상 1 이하 또는 0 이상 1 이하 같은 작은 범위로 정규화한다. 마지막으로, 다섯번째



카테고리는 속성 구성(attributes construction)으로 기존의 속성에 새로운 속성을 생성하고, 더하기 위해 사용한다.

데이터 전처리의 분류 및 데이터 변환의 세부 카테고리는 아래의 [그림 10]과 같이 도식화할 수 있다.



[그림 10] Han & Kamber (2001), Famili et al (1997)의 전처리 분류

데이터 변환 중 정규화는 신경망 알고리즘의 네트워크 수렴 속도를 개선하고 모든 변수의 값 범위를 큰 차이 없이 공통 스케일을 갖도록 변환하여 국소 최적점에서 학습을 멈추는 것을 막는 효과가 있다(Choi et al, 2019).

본 논문은 데이터의 정규화 방법에 따른 최적화 문제를 연구하고자 하였다. 따라서 이 절에서 이어질 내용인 데이터 전처리 관련 문헌 고찰은 정규화에 중점을 두고 진행하고자 한다.



3.2. 정규화의 이론적 배경

본 논문에 언급한 선행 연구에서 사용한 정규화 방법을 포함하여 본 연구에 사용할 정규화 방법을 정리한 내용은 아래의 [표 1]과 같다.

[표 1] 데이터 정규화 방법

방법	내용
로그 변환	$\log(x)$
Z-스코어	$(\log(x) - \log(x)_{\text{mean}}) / \log(x)_{\text{std}}$
최대-최소	$(\log(x) - \log(x)_{\text{min}}) / (\log(x)_{\text{max}} - \log(x)_{\text{min}})$
로버스트 정규화	$(\log(x) - \log(x)_{2/4}) / (\log(x)_{3/4} - \log(x)_{1/4})$
정규 점수	$\Phi^{-1}(\text{rank}(\log(x)) / (N + 1/4))$

$\log(x)_{\text{mean}}$ 과 $\log(x)_{\text{std}}$ 는 각각 $\log(x)$ 의 평균과 표준편차를 의미한다. $\log(x)_{\text{min}}$ 과 $\log(x)_{\text{max}}$ 는 각각 $\log(x)$ 의 최솟값과 최댓값을 의미한다. $\log(x)_{1/4}$, $\log(x)_{2/4}$, $\log(x)_{3/4}$ 는 각각 $\log(x)$ 의 1/4분위수, $\log(x)$ 의 2/4분위수, $\log(x)$ 의 3/4분위수를 의미한다. Φ^{-1} 는 표준정규분포 누적분포함수의 역함수, N은 표본의 수를 의미한다.

첫째, 로그 변환은 데이터에 자연로그를 취하는 방법이다. 둘째, Z-스코어 방법은 평균을 차감하여 0으로 평균을 이동시킨 후, 표준편차로 나누어줌으로써 데이터의 표준편차를 1로 변환하는 방법이다. 셋째로, 최소-최대 방법은 일정 범위 안에 머무르도록 스케일링을 적용하는 방법이다. 대표적으로 [0,1] 범위를 많이 사용한다. 넷째로, 로버스트 정규화 방법은 Z-스코어와 비슷하나 평균과 표준편차 대신에 사분위값을 이용하는 방법이다. 사분위값을 이용하기 때문에 이상치의 영향을 덜 받는다. 다섯째로, 정규 점수는 데이터 내의 관측치 순위에서 파생된 데이터 값이다. 데이터와



동일한 크기의 표준 정규 확률 변수의 샘플에서 동일한 순위의 순서 통계량과 같은 값 혹은 근사값을 할당한다. 이 경우 정규화는 표준정규분포에서 원래 데이터 값의 집합이 발생했을 때 예상되는 것과 일치하는 값 집합을 만든다. 정규 점수로 정규화하는 여러가지 방법 중 Van der Waerden 방법을 사용하였으며 정규 점수의 세부적인 정규화 절차는 다음과 같다. 먼저 데이터에 순위를 매긴다. 다음으로 이전 단계의 결과인 순위 데이터를 데이터의 수에 $1/4$ 을 더한 값으로 나눈다. 마지막으로 표준정규분포 누적분포함수에 대한 역을 이전 단계의 결과인 확률 값에서 계산하여 반환한다. 최종적으로 표준정규 분포의 형태로 변환된다(Conover, 1999).

3.3. 정규화에 대한 선행 연구

기계학습에서 데이터 정규화의 효과를 밝히기 위해 다양한 연구가 진행되었으며, 연구 내용을 종합한 결과, 절대적으로 성능의 향상 효과를 가져오는 정규화 방법은 없었다. 적용 모형, 데이터의 성격에 따라 더 나은 성능을 보이는 정규화 방법이 상이하였다. 자세한 내용은 다음과 같다.

Al-Shalabi & Shaaban (2006)은 HSV 데이터에 ID3 알고리즘을 적용하는 문제에서 세 가지 정규화 방법(Z-스코어, 최소-최대, 십진법)을 연구하였다. 잎 노드, 정확도, 나무 성장 시간을 기준으로 성능을 평가한 결과, 실험 최소-최대 정규화가 가장 좋은 성능을 보였다. Akdemir and Yu (2009)는 금융 시장에서 주식의 다음 변동을 예측하기 위해 인공신경망을 적용하는 문제에서 유클리드 기반의 정규화 방법을 제안하였다. 최소-최대 정규화를



벤치마크 방법으로 사용하였는데 연구 결과, 제안한 정규화 방법이 최소-최대 정규화 방법보다 나은 결과를 보였다. Cha et al. (2006)는 인공신경망을 적용한 파동에 의한 해저 액화 문제에서 최소-최대 정규화로 데이터를 전처리했을 때 인공신경망의 결과가 개선됨을 입증하였다. Jain et al. (2005)는 멀티 모드 생체 인식 시스템에서 정규화 기법 효과를 조사하였다. 연구 결과, 정규화를 하였을 때 그렇지 않을 때에 비하여 더 나은 성능을 보였다. 또한 최소-최대 및 Z-스코어 정규화는 특이치에 민감하고, 탄젠트 정규화 방법은 강건하고 효율적이라고 주장하였다. Jolai & Ganbari (2010)는 HNN(Hopfield neural network)를 사용한 여행 판매원 문제에서 Z-스코어와 로그 정규화 방법을 사용했을 때, 모형의 정확성이 향상됨을 입증하였다. Kim (1999)은 역전파 알고리즘 사용 시 데이터의 정규화 방법에 따른 성능 차이를 연구하였다. 연구에서 두 가지 최소-최대 정규화 방법을 제안하였는데 첫번째는 전체 입력 패턴에 대해 정규화하는 방법이고, 두번째는 각각의 개별 입력 패턴에 대해 정규화하는 방법이었다. 연구 결과, 역전파 알고리즘 사용 시 데이터의 정규화 방법에 따라 결과가 민감하게 반응한다는 결론을 도출하였다. Kim et al. (2005)는 역전파를 사용하여 운전자의 최적 경로를 찾는 문제에서 세 가지 최소-최대 정규화 방법을 제안하고, 비교하였다. 연구 결과, 제안된 방법을 사용했을 때, 모형의 성능이 향상되었으며, 데이터에 대한 정규화는 역전파 알고리즘을 학습시키기 전에 반드시 고려되어야 한다는 결론을 도출하였다. Mazzatorta et al. (2002)는 역전파를 적용한 독성 예측에서 6가지 정규화 방법을 비교하였다. 연구 결과,



탄젠트 하이퍼볼릭 로그 스케일링 수정 방법이 이상적인 정규화 방법으로 밝혀졌다. Shanker et al. (1996)은 순방향 네트워크 모형에서 가장 일반적으로 알려진 두 가지의 정규화 방법(최대-최소 및 Z-스코어)을 비교하였다. 연구 결과, Z-스코어 정규화 방법을 사용한 분류의 평균 제곱 오차가 가장 낮고, 정분류율이 가장 높았다. Visalakshi & Thangavel (2009)는 k-평균 군집에서 정규화의 영향을 연구하기 위해 6개의 벤치마크 데이터와 세 가지 정규화 방법(최소-최대, Z-스코어, 십진법)을 사용하였다. 연구 결과, 정규화를 하였을 때 군집 내, 군집 외 분산 측면에서 더 나은 질의 군집이 생성되었다. 해당 데이터에 제안된 정규화 절차를 선택하는 것이 성능을 향상시킨다고 주장하였다. Wang and Zhang (2009)는 퍼지 클러스터링에서 네 가지 정규화 방법(Z-스코어, 최소-최대, 최대값, 평균값)을 적용하고 비교하였다. 연구 결과, 평균과 최대치에 가중을 둔 정규화 방법의 결과가 다른 방법보다 우월하였다.

4. 선행연구의 한계점 및 본 연구의 차별점

다양한 국내외 선행연구들을 고찰한 결과, 주가 예측 관련 연구들은 S&P 500, KOSPI 200, NIFTY 50, 상해종합주가지수, 구글 등 다양한 지수 및 종목들에 대해 통계학적 방법 및 인공지능 분석 방법 등을 활용하여 예측하였으며 다양한 방법론들이 주식시장 예측에 적합함을 밝혔다. 특히 인공지능 분석 방법 중 하나인 순환신경망을 이용한 주가 예측 연구 또한



다양하게 이루어졌으며, 주가 예측에 대한 순환신경망의 적합성을 증명하였다. 정규화 관련 연구들은 분류, 예측, 군집 등을 목적으로 다양하게 진행되었으며 신경망, 군집 모형 등 다양한 모형을 활용하여 진행된 바 있다. 그런데 해당 연구들을 살펴보면, 순환신경망을 연구 모형으로 사용하는 연구에서 시계열 데이터 정규화를 집중적으로 다룬 연구는 많이 진행되지 않았다. 또한 순환신경망 관련 연구의 경우 대부분이 로그 차분 데이터 혹은 수익률 데이터라는 일률적으로 정규화된 데이터만 사용하여 학습을 수행하였다. 따라서 본 연구에서는 다양한 정규화 방법을 사용하여 순환신경망의 예측 성과를 비교해보고자 한다. 추가로, 장단기 메모리 모형의 적응 수에 따른 예측 성과를 비교해보고자 한다.



Ⅲ. 실증 분석

1. 자료의 구성

1.1. 데이터 기간 및 수집

본 연구에서는 주가 예측 모형 학습에 사용될 데이터를 수집하기 위해 구글 파이낸스에서 제공하는 Open API 서비스를 이용하였다. 데이터 수집 대상은 S&P 500 지수의 일별 종가이며, 2010월 1월 4일부터 2018년 12월 28일까지로 설정하였다. 즉, 총 2263일의 S&P 500 지수 일별 종가를 학습 데이터로 사용한다.

1.2. 자료의 생성 및 분할

입력 변수와 출력 변수를 구성하기 위해 무빙 윈도우 기법(Jang et al., 1993; Lee et al., 2017)을 활용하였다. 입력 변수는 $[1, 2, \dots, t]$ 기간 동안의 p_t 로 이루어진 $[p_1, p_2, \dots, p_t]$ 로 구성되며, p_t 는 t 시점의 일별 종가를 의미한다. 입력 변수가 위와 같을 때, 출력 변수는 p_{t+1} 로 설정한다. 본 연구의 t , 즉 시퀀스는 40으로 설정하고, 윈도우를 옮겨가며 입력 변수와 출력 변수가 한 쌍인 데이터를 모아, 총 2223 쌍의 데이터로 구성된 자료를 생성한다.

모형 학습 및 검증 단계에서는 자료의 분할, 모형 학습, 모형 검증 과정이 수행되며 자세한 내용은 다음과 같다. 자료의 분할 과정에서는 모형의 학습 및 검증을 위해 자료가 9:1의 비율로 학습용 자료와 테스트용 자료로



분할되게 된다. 그 다음, 학습용 자료를 7:3의 비율로 학습용 데이터와 검증용 자료로 재분할한다. 모형 학습 과정에서는 앞서 분할된 학습용 자료를 활용하여 주식시장 예측 모형의 학습이 수행되며, 검증용 자료를 활용하여 모형을 고정하고, 테스트용 자료를 활용하여 고정된 모형을 적용했을 때의 예측 성과를 평가한다.

2. 연구 모형

순환신경망 모형 중 가장 대표적인 장단기 메모리 모형과 게이트드 순환 유닛이 본 연구의 모형으로 활용된다. 모형 학습 시 파라미터 학습 방법은 RMS 경사하강법, 학습율은 0.0001, 에포크는 1000회, 배치 크기는 64로 설정하였다. 학습 과정에서 검증용 자료의 손실 함수를 최소화하는 방향으로 모델의 파라미터를 고정시키는데, 이 때, 손실 함수로는 평균 절대 오차를 사용하였다. 파라미터 초기화 방법으로는 He et al. (2015)의 방법, 활성화 함수는 leaky ReLu를 사용하였다.

모형의 설성을 요약하면 아래의 [표 2]와 같다.



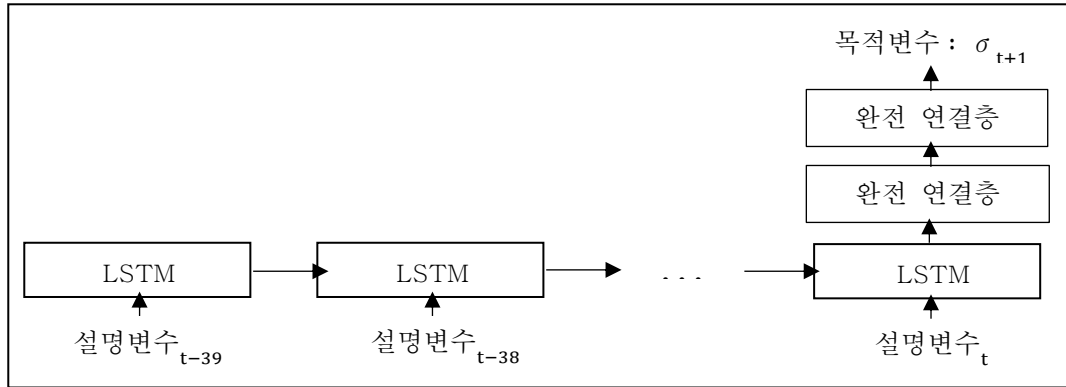
[표 2] 연구 모형의 설정

설정 변수	내용
학습률	0.0001
에포크	1000
배치 사이즈	64
초기화 방법	He et al. (2015)
활성화 함수	leaky ReLu
경사하강법	RMS

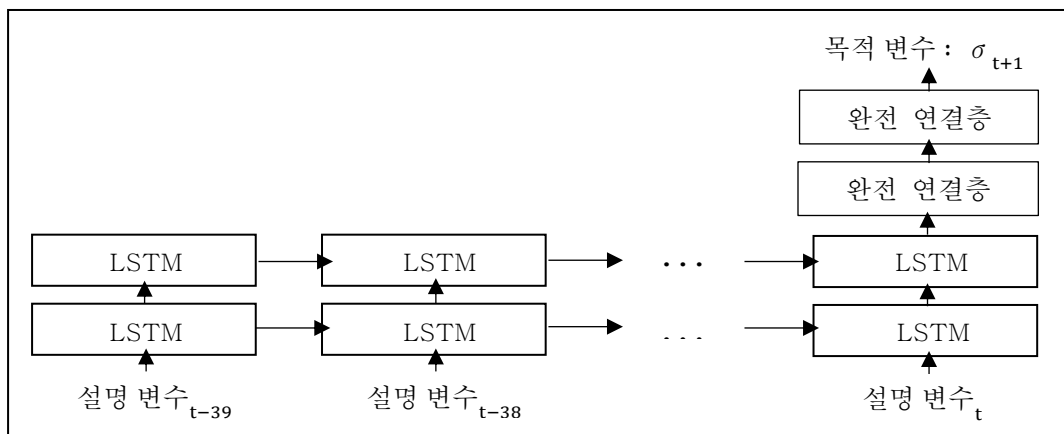
1.2.에 근거하여 연구 모형은 40개의 1차원 입력 변수로 구성된 입력 계층, 순환신경망 셀로 구성된 40개의 은닉 계층, 한 개의 1차원 출력 변수로 구성된 출력 계층으로 이루어진다. 출력 계층에는 노드 수가 각각 200, 100개인 완전연결층을 추가하였다. 학습용 자료는 셔플링 후 사용하였다. 학습용 자료와 검증용 자료를 활용하여 모형 학습을 완료한 후 테스트 자료를 활용하여 모형의 예측 성과를 측정하였다. 마지막으로 위에 언급한 바와 같이 모형을 설정하고, 같은 모형 설정에 대해 장단기 메모리 모형과 게이트드 순환 유닛 셀의 적층 수만 한 개와 두 개로 달리하여 모형의 성과를 비교한다.

아래의 [그림 11]과 [그림 12]는 차례대로 각각 적층 수가 한 개, 두 개인 장단기 메모리 모형을 도식화하였다.



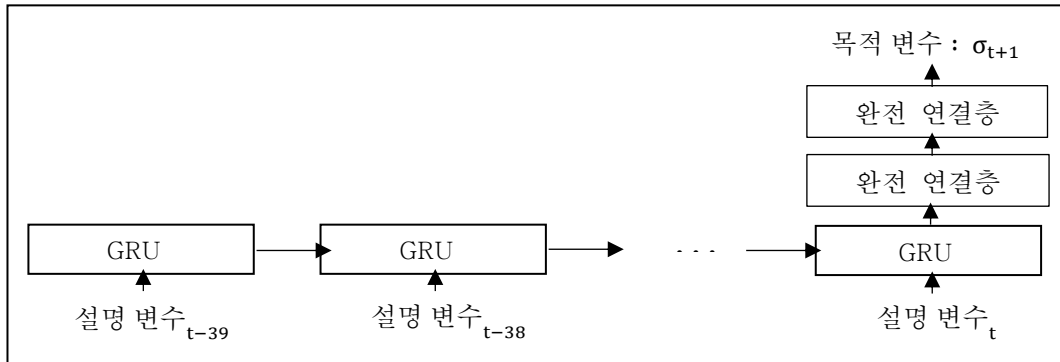


[그림 11] 적층 수가 한 개인 LSTM 연구 모형

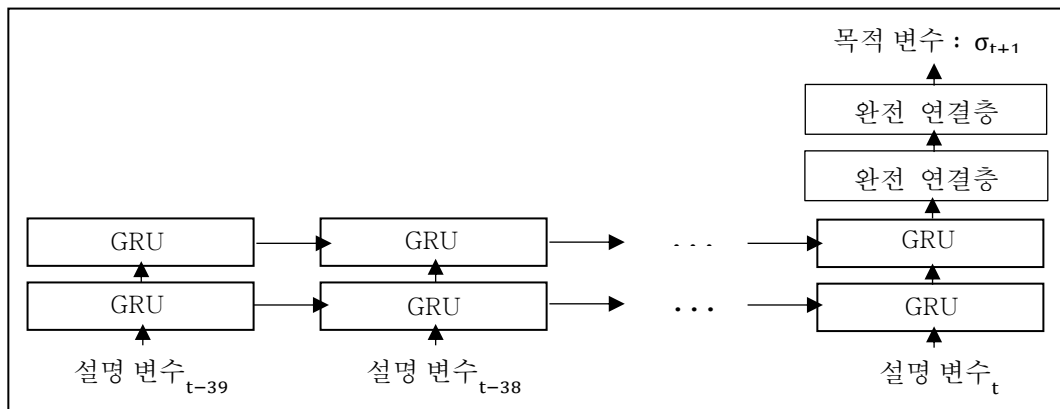


[그림 12] 적층 수가 두 개인 LSTM 연구 모형

아래의 [그림 13]과 [그림 14]는 차례대로 적층 수가 한 개, 두 개인 게이트드 순환 유닛 모형을 도식화하였다.



[그림 13] 적층 수가 한 개인 GRU 연구 모형



[그림 14] 적층 수가 두 개인 GRU 연구 모형

IV. 실험 및 성능 평가

성능 평가를 위한 오차 함수로는 테스트셋의 평균절대백분율오차(MAPE: Mean Absolute Percentage Error)와 평균절대스케일오차(MASE: Mean Absolute Scaled Error)를 사용한다. 평균절대백분율오차는 크기 의존적 에러의 단점을 커버하기 위한 오차 함수이다. 실제값 대비 오차의 절대값들의 평균에 100을 곱한 값으로 식(26)과 같다.

$$MAPE = \frac{1}{T} \sum_{t=1}^T \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100 \quad (26)$$

식(26)에서 y_t 는 실제값, \hat{y}_t 는 예측값, T 는 시간 지표로 $t \in 0, \dots, T$ 를 의미한다.

식(26)에서 알 수 있듯이 y_t 값이 1보다 작을 경우, 평균절대백분율오차 값이 무한대에 가까운 값을 가질 수 있다는 단점을 지닌다. 이를 보완하기 위해 평균절대스케일오차를 사용한다.

평균절대스케일오차는 평소의 변동 폭 대비 오차를 측정하는 기준이라고 할 수 있다. 본 논문과 같이 변동성이 큰 지표와 변동성이 낮은 지표의 오차를 같이 예측, 비교할 때 유용하게 사용할 수 있다. 예측 값과 실제 값의 차이를 평소에 움직이는 평균 변동폭으로 나눈 값으로 식(27)과 같다.

$$MASE = \frac{1}{T} \sum_{t=1}^T \left(\frac{|y_t - \hat{y}_t|}{\frac{1}{T-1} \sum_{t=2}^T |y_t - y_{t-1}|} \right) \times 100 \quad (26)$$



식 (27)에서 y_t 는 실제값, \hat{y}_t 는 예측값, T 는 시간 지표로 $t \in 0, \dots, T$ 를 의미한다.

각 모형별 평균절대백분율오차와 평균절대스케일오차는 아래의 <표>와 같다. 1000 에폭시 동안 최소 평균절대백분율오차와 평균절대스케일오차를 기재하였다.

[표 3] 적층 수, 정규화 방법에 따른 MAPE 및 MASE

정규화 방법		Stack : 1		Stack : 2	
		LSTM	GRU	LSTM	GRU
log	MAPE	0.174132	0.345493	0.055703	0.143189
	MASE	2.138822	4.240635	0.683536	1.756089
Z-Score	MAPE	0.836227	0.803111	0.928392	0.714253
	MASE	0.602031	0.579095	0.668364	0.515848
최소-최대	MAPE	0.430436	0.357890	0.316312	0.399811
	MASE	0.678954	0.564613	0.500112	0.630869
robust 정규화	MAPE	1.007021	0.985117	0.995191	1.193296
	MASE	0.623365	0.609933	0.616234	0.739915
normal score	MAPE	23.737314	26.580204	22.379934	12.074524
	MASE	3.615839	3.657193	3.540399	3.426639

[표 3]의 내용을 종합하여, 적층 수에 따른 결과, 정규화 방법에 따른 결과, 두 경우를 조합한 결과의 순으로 정리하였다.

먼저, 적층 수에 따른 모형을 비교한 결과는 다음과 같다. 장단기메모리 모형의 경우, Z-스코어 정규화를 사용했을 때를 제외하고, 모두 두 층을 적층한 모형이 한 층만 적층한 모형에 비해 평균절대백분율오차 및



평균절대스케일오차가 더 작았다. 게이트드순환유닛의 경우, 최소-최대 정규화와 로버스트 정규화를 제외하고, 두 층을 적층한 모형이 한 층만 적층한 모형에 비해 평균절대백분율오차 및 평균절대스케일오차가 더 작았다. 장단기메모리 모형과 게이트드순환유닛을 종합하면, 적층 수를 1에서 2로 늘림에 따라 정규화 방법에 관계없이 전반적으로 오차가 감소하는 모습을 보였다. 그러나 Z-스코어, 최소-최대, 로버스트 정규화에서 적층 수가 늘어났음에도 불구하고, 오차가 증가하는 예외적인 모습을 보였다.

다음으로, 입력값 정규화 방법에 따른 모형을 비교한 결과는 다음과 같다. 장단기메모리 모형의 경우, 적층 수에 관계없이 평균절대백분율오차와 평균절대스케일오차가 모두 1 미만을 기록한 정규화 방법은 Z-스코어, 최소-최대화 방법이다. Z-스코어 방법의 경우, 1층을 적층했을 때, 평균절대백분율오차와 평균절대스케일오차가 0.836227, 0.602031을 기록하였으며, 2층을 적층했을 때, 0.928392, 0.668364를 기록하였다. 최소-최대 정규화 방법의 경우, 1층을 적층했을 때, 평균절대백분율오차와 평균절대스케일오차가 각각 0.430436, 0.678954를 기록하였으며, 2층을 적층했을 때, 0.316312, 0.500112를 기록하였다. 게이트드순환유닛의 경우, 적층 수에 관계없이 평균절대백분율오차와 평균절대스케일오차가 모두 1 미만을 기록한 정규화 방법은 Z-스코어, 최소-최대화 방법이다. Z-스코어 방법의 경우, 1층을 적층했을 때, 평균절대백분율오차와 평균절대스케일오차가 각각 0.803111, 0.579095을 기록하였으며, 2층을 적층했을 때, 0.714253, 0.515848를 기록하였다. 최소-최대 정규화 방법의 경우, 1층을 적층했을 때,



평균절대백분율오차와 평균절대스케일오차가 각각 0.357890, 0.564613를 기록하였으며, 2층을 적층했을 때, 0.399811, 0.630869를 기록하였다. 결론적으로 장단기메모리 모형과 게이트드순환유닛 모두 Z-스코어, 최소-최대화 정규화 방법의 결과가 가장 좋았다.

마지막으로, 평균절대백분율오차와 평균절대스케일오차의 평균을 기준으로 보았을 때, 정규화 방법과 적층 수의 조합 중 가장 우수한 결과를 보이는 경우 최소-최대와 2층을 적층한 모형이었다.



V. 결론

1. 연구의 결론

본 연구는 정규화 방법, 적층 수에 따른 순환신경망의 성능을 비교하였다. 연구의 결론에서는 정규화 방법, 적층 수에 대한 결과를 차례대로 서술하였다.

딥러닝(심층신경망)에서 데이터의 정규화는 중요한 요소이다(Sola & Sevilla, 1997). 하지만 데이터 정규화 방법에 따른 모형의 성능을 비교하는 선행 연구는 다수 있었던 반면, 금융 시계열 데이터에 순환신경망을 적용한 연구에서 정규화 방법에 대한 선행연구는 많이 이루어지고 있지 않았다. 따라서 본 연구에서는 금융 시계열 데이터인 S&P 500 일별 종가 데이터를 로그 변환, Z-스코어, 최소-최대, 정규 점수, 로버스트 방법으로 정규화하고, 방법별로 성능을 비교하고자 하였다. 연구 결과, Z-스코어, 최소-최대화 정규화 방법의 결과가 가장 좋았다. 정규화 방법이 모형의 성능을 향상시킨다는 사실을 금융시계열 데이터를 사용한 순환신경망 모형에서도 확인할 수 있었다.

Li et al. (2018)의 연구에 따르면 순환신경망 모형에서 4층 이상의 과도한 적층은 오히려 성능을 저하시킨다는 결과가 있었다. 그러나 적절한 수 안에서 층을 늘리는 것은 일반적으로 성능을 향상시키는 것으로 알려져있다. 연구 결과, 1층을 적층했을 때보다 2층을 적층했을 때 성능이 향상되었으며, 이는 선행연구의 결과와 동일한 결과이다. 그러나 정규화 방법에 따라 예외적인



경우도 있었다.

2. 연구의 시사점 및 한계점

군집 분석, 인공신경망 등 다양한 데이터 분석을 진행할 때, 어떤 정규화 방법이 좋은지에 대한 다수의 연구가 있었다. 그러나 금융 시계열 데이터에 순환신경망을 적용할 때, 정규화 방법에 따른 모형 성능을 비교한 연구를 찾기 어려웠다. 금융 시계열 데이터를 사용하여 정규화 방법에 따른 순환신경망의 성능을 연구하고, 어떤 정규화 방법을 사용했을 때 가장 나은 결과를 얻을 수 있는지 밝혔다는 점에서 의의가 있다. 또한 선행 연구에서 알려진 바와 같이 적절한 층의 수 이내에서 순환신경망을 적층하면 예측 성능을 향상시킨다는 것을 증명하였다.

한편, 정규화 방법 중 평균절대비율오차와 평균절대스케일오차 모두가 가장 작은 값을 가지는 정규화 방법은 존재하지 않았다. 보다 정확한 오차 측정 함수의 부재가 원인일 수도 있다고 판단하였으며, 이는 본 연구의 한계점이다. 또한 최대값이나 평균값에 가중을 두는 정규화 방식 등 보다 다양하고 발전된 정규화 방법을 추가하여 비교하는 방식으로 연구를 발전시킬 여지가 있다고 사료된다.



참고 문헌

[해외논문]

Akdemir, B. and Yu, L.P. Elliot Waves Predicting for Stock Marketing Using Euclidean Based Normalization Method Merged with Artificial Neural Network. (2009). 2009 Fourth International Conference on Computer Science and Convergence Information Technology, 562–567.

Al-Shalabi, L. and Shaaban, Z. Normalization as a Preprocessing Engine for Data Mining and the Approach of Preference Matrix. Proceeding of the International Conference on Dependability of Computer Systems, (2006).

Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), 157–166.

Cha, D.H., Blumenstein, M., Zhang, H., and Jeng, D.S. (2006). Improvement of an Artificial Neural Network Model Using Min-max Preprocessing for the Prediction of Wave-induced Seabed Liquefaction. 2006 International Joint Conference on Neural Networks, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 16–21.

Chang, P. C., & Liu, C. H. (2008). A TSK type fuzzy rule based system for stock price prediction. Expert Systems with applications, 34(1), 135–144.

Choudhry, R., Greg, K. (2008). A hybrid machine learning system for stock market



forecasting. World Academy of Science, Engineering and Technology 39.3, 315–318.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Conover, W. J. (1980). Practical nonparametric statistics, John Wiley.

Elman, J. L. (1990). Finding structure in time. Cognitive science, 14(2), 179–211.

Famili, A., Shen, W.M., Weber, R., and Simoudis, E. (1997). Data Preprocessing and Intelligent Data Analysis, Intelligent Data Analysis 1, 3–23.

Fisher, T., Krauss, C. (2017). Deep learning with long short-term memory networks for financial market predictions, European Journal of Operational Research, 270(2), 654–669

Gers, F. A., Schmidhuber, J., Cummins, F. (1999). Learning to forget: Continual prediction with LSTM.

Graves, A., Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18(5–6), 602–610.

Güler, N. F., Übeyli, E. D., Güler, I. (2005). Recurrent neural networks employing Lyapunov exponents for EEG signals classification. Expert systems with applications, 29(3), 506–514.

Han, J. and Kamber, M. (2001). Data mining : Concepts and Techniques. Morgan



Kaufmann, USA.

Han, K. H., Kim, J. H. (2000). Genetic quantum algorithm and its application to combinatorial optimization problem. In Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512) (Vol. 2, pp. 1354–1360). IEEE.

Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

Huang, C. J., Yang, D. X., Chuang, Y. T. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications*, 34(4), 2870–2878.

Ince, H., Trafalis, T. B. (2008). Short term forecasting with support vector machines and application to stock price prediction. *International Journal of General Systems*, 37(6), 677–687.

Jain, A., Nandakumar, K., and Ross, A. (2005). Score Normalization in Multimodal Biometric Systems. *Pattern Recognition* 38, 2270–2285.

Jang, G. S., Lai, F., Jiang, B. W., Parng, T. M., Chien, L. H. (1993). Intelligent stock trading system with price trend prediction and reversal recognition using dual-module neural networks. *Applied Intelligence*, 3(3), 225–248.

Junyoung, C., Caglar, G., Kunghyun, C., Yoshua, B. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, arXiv preprint arXiv, 1412.3555v1.



Kim, D.Y. (1999). Normalization Methods for Input and Output Vectors in Backpropagation Neural Networks. *International Journal of Computer Mathematics* 71, 161–171.

Kim, K.W., Kim, D.Y., and Jung, H.Y. (2005). Normalization Methods on Backpropagation for the Estimation of Drivers' Route Choice. *KSCE Journal of Civil Engineering* 9, 5, 403–406.

Kotsiantis, S.B., Kanellopoulos, D., and Pintelas, P.E. Data Preprocessing for Supervised Learning. (2006). *International Journal of Computer Science* 1, 2, 1306–4428.

K. Park & H. Shin. (2011). Stock Price Prediction Based on Time Series Network. *Korean Management Science Review* 28.1, 53–60.

Lai, R. K., Fan, C. Y., Huang, W. H., Chang, P. C. (2009). Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications*, 36(2), 3761–3773.

LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436.

Lin, X., Yang, Z., Song, Y. (2009). Short-term stock price prediction based on echo state networks. *Expert systems with applications*, 36(3), 7313–7317.

Li S, Li W, Cook, C, Shu, C, and Gao, Y. (2018). Independently recurrent nueral network(IndRNN): building a longer and deeper RNN, arXiv, preprint, arXiv: 1803.04831.



Liu, Z., Sullivan, C. J. (2019). Prediction of weather induced background radiation fluctuation with recurrent neural networks. *Radiation Physics and Chemistry*, 155, 275–280.

Mazzatorta, P., Benfenati, E., Neagu, D., and Gini, G. The Importance of Scaling in Data Mining for Toxicity Prediction. *Journal of Chemical Information and Computer Sciences* 42, (2002), 1250–1255.

Nelson, D. M., Pereira, A. C., de Oliveira, R. A. (2017, May). Stock market's price movement prediction with LSTM neural networks. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 1419–1426). IEEE.

Roondiwala, M., Patel, H., Varma, S. (2017). Predicting stock prices using LSTM. *International Journal of Science and Research (IJSR)*, 6(4), 1754–1756.

Shanker, M., Hu, M.Y., and Hung, M.S. (1996). Effect of Data Standardization on Neural Network Training. *Omega, International Journal of Science* 24, 4, 385–397.

Shin, S., Sung, W. (2016, May). Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks. In 2016 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 2274–2277). IEEE.

Shin, S., Sung, W. (2016, May). Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks. In 2016 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 2274–2277).



IEEE.

Wei, H., Yoshiteru, N., Shou-Yang, W. (2005). Forecasting stock market movement direction with support vector machine, *Computers & Operations Research*. 32(10), 2513–2522.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10), 1550–1560.

Williams, R. J., Peng, J. (1990). An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural computation*, 2(4), 490–501.

Zheng, C., Wang, S., Liu, Y., Liu, C. (2019). A novel RNN based load modelling method with measurement data in active distribution system. *Electric Power Systems Research*, 166, 112–124.

Zhuge, Q., Xu, L., Zhang, G. (2017). LSTM Neural Network with Emotional Analysis for Prediction of Stock Price. *Engineering Letters*, 25(2).



[국내 논문]

구승환, 장성용. (2012). 기본적 분석방법을 통한 주식 투자 전략에 관한 시뮬레이션 연구. 경영과학, 29(2), 53-64.

김성수, 홍광진. (2017). 순환 신경망 기술을 이용한 코스피 200 지수에 대한 예측 모델 개발 및 성능 분석 연구. 한국산업정보학회논문지, 22(6), 23-29.

김선웅, 안현철. (2010). Support Vector Machines 와 유전자 알고리즘을 이용한 지능형 트레이딩 시스템 개발. 지능정보연구, 16(1), 71-92.

김진백, 김유일. (2000). 인공 신경망의 학습에 있어 가중치 변화방법과 은닉층의 노드수가 예측정확성에 미치는 영향. 정보시스템연구, 9(1), 27-44.

김진화, 홍광현, 민진영. (2011). 지식 누적을 이용한 실시간 주식시장 예측. 지능정보연구, 17(4), 109-130.

김현지, 장우진. (2010). 지수 이동 평균선을 이용한 거래 전략. 한국경영과학회 학술대회논문집, 1124-1130.

박인찬, 권오진, 김태윤. (2009). 시계열 모형을 이용한 주가지수 방향성 예측, 한국데이터정보과학회지, 20(6), 991-998.

성노운, 남기환. (2017). 온라인 뉴스 및 거시경제 변수를 활용한 주가예측. Entrue Journal of Information Technology, 16(2), 41-54.

신동하, 최광호, 김창복. (2017). RNN과 LSTM을 이용한 주가 예측을 향상을 위한 딥러닝 모델. 한국정보기술학회논문지, 15(10), 9-16.



- 신동백. (2007). 몬테카를로 시뮬레이션방법을 이용한 환율예측분석. 산업경제연구, 20(5), 2075-2093.
- 안성원, 조성배. (2010). 뉴스 텍스트 마이닝과 시계열 분석을 이용한 주가예측. 한국정보과학회 학술발표논문집, 37(1C), 364-369.
- 유진. (2002). 가격제한 존재 시 VaR 추정 방법론: 몬테카를로 시뮬레이션. 선물연구, 10(1), 81-111.
- 이모세, 안현철. (2018). 효과적인 입력변수 패턴 학습을 위한 시계열 그래프 기반 합성곱 신경망 모형. 지능정보연구, 24(1), 167-181
- 이석준, 오경주. (2011). 선물시장의 시스템트레이딩에서 동적시간 와핑 알고리즘을 이용한 최적매매빈도의 탐색 및 거래전략의 개발. 한국데이터정보과학회지, 22(2), 255-267.
- 이우식. (2017). 딥러닝분석과 기술적 분석 지표를 이용한 한국 코스피 주가지수 방향성 예측. 한국데이터정보과학회지, 28(2), 287-295.
- 이윤선. (2003). 한국증권시장에서 기술적 분석의 경제적 효과-주가 이동평균선을 중심으로. 금융공학연구, 2(2), 1-21.
- 이형용. (2008). 한국 주가지수 등락 예측을 위한 유전자 알고리즘 기반 인공지능 예측기법 결합모형. Entrue Journal of Information Technology, 7(2), 33-43.
- 주일택, 최승호. (2018). 양방향 LSTM 순환신경망 기반 주가예측모델. 한국정보전자통신기술학회 논문지, 11(2), 204-208.
- 허준영, 양진용. (2015). SVM 기반의 재무 정보를 이용한 주가 예측. 정보과학회



컴퓨팅의 실제 논문지, 21(3), 167-172.

홍태호, 김은미, 차은정. (2017). 뉴스 감성분석과 SVM 을 이용한 다우존스 지수와 S&P500 지수 예측. 인터넷전자상거래 연구, 17(1), 23-36.



영문 초록

A Study on the Characteristics of Recurrent Neural Networks Using the S&P 500 Index

Han-Jeong You

The purpose of this study is to compare the prediction performance of financial time series data according to the input value normalization method and stacking method. LSTM & GRU in the recurrent neural network were used to predict the S&P 500 index. Experimental results show that the best prediction results are obtained when normalized by the minimum-maximization and Z-score methods, and the prediction performance was generally improved, but not always when two layer were stacked rather than one layer. The lowest prediction error case among the total 10 cases (five normalization methods multiplied by two stacking methods) was the minimum-maximum normalization and the two layers stacking case.

Keyword : Stock Prediction, Recurrent Neural Network, Long Short-Term Memory, Gated Recurrent Unit, Normalization, Layer Stacking



