



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](#)

석 사 학 위 논 문

시계열모형의 예측에서 LSTM 성능에 대한 연구

계 명 대 학 교 대 학 원

통 계 학 과

박 혜 원

박
혜
원

지 도 교 수 하 정 철

2
0
1
9
년

2
월

2 0 1 9 년 2 월

시계열모형의 예측에서 LSTM 성능에 대한 연구

지도교수 하 정 철

이 논문을 석사학위 논문으로 제출함

2019년 2월

계명대학교 대학원

통계학과

박 혜 원

박혜원의 석사학위 논문을 인준함

주 심 박 철 용

부 심 하 정 철

부 심 조 정 효

계 명 대 학 교 대 학 원

2019년 2월

목 차

1. 서론	1
1.1 연구의 필요성 및 목적	1
1.2 기존연구 고찰	2
2. 이론적 배경	4
2.1 Box-Jenkins 방법론	4
2.2 기계학습(machine learning)	5
2.3 인공신경망(artificial neural network, ANN)	6
2.4 순환신경망(recurrent neural network, RNN)	7
2.5 장단기메모리(long short term memory, LSTM)	7
3. 연구방법	9
3.1 연구방법의 개요	9
3.2 ARMA모형을 이용한 모수적 예측 방법	10
3.3 LSTM	12
4. 모수적 방법과 LSTM의 예측 성능 비교	13
4.1 AR(1)모형과 MA(1)모형	13
4.2 AR(2)모형	17
4.3 MA(2)모형	19
4.4 ARMA(1,1)모형	21
5. 결론	23
참고문헌	25
영문초록	27
국문초록	29

표 목 차

<표 3-1> 모형별 예측 식	11
<표 4-1> AR(1)모형에서 모수적 방법과 LSTM 예측 성능 비교	14
<표 4-2> MA(1)모형에서 모수적 방법과 LSTM 예측 성능 비교	15
<표 4-3> AR(2)모형에서 모수적 방법과 LSTM 예측 성능 비교	18
<표 4-4> MA(2)모형에서 모수적 방법과 LSTM 예측 성능 비교	20
<표 4-5> ARMA(1,1)모형에서 모수적 방법과 LSTM 예측 성능 비교	22

그림 목 차

[그림 2-1] Box-Jenkins 방법론	4
[그림 2-2] ANN 구조	6
[그림 2-3] RNN 구조	7
[그림 2-4] LSTM 구조	8
[그림 4-1] AR(1)모형에서 모수적 방법 예측 시계열 그래프	16
[그림 4-2] AR(1)모형에서 LSTM(SGD) 예측 시계열 그래프	16

1. 서론

1.1 연구의 필요성 및 목적

시계열 예측은 시간의 흐름에 따라 관측된 과거의 데이터를 이용하여 미래를 예측하는 방법으로, 자기회귀(auto-regressive, AR)모형, 이동평균(moving average, MA)모형, 그리고 두 모형이 결합된 자기회귀이동평균(ARMA)모형이 널리 사용되어 왔다. 하지만 최근에는 수많은 데이터와 정보가 쌓이고 고성능 컴퓨터의 등장으로 딥 러닝(deep learning)을 이용한 지도 학습 알고리즘의 성능이 향상되어 이를 활용한 방법이 많이 개발되었다. 지도 학습 중에서도 순환신경망(recurrent neuron network, RNN)이 시계열 예측에 적합하다고 알려져 있으며 RNN의 단점을 보완한 장단기메모리(long short term memory, LSTM)가 더 좋은 성능을 보이는 것으로 알려져 있다(Sak, Senior, & Beaufays, 2014).

시계열 예측은 예측 시점에 따라 1 시차 이후 예측(one-step ahead prediction), 여러 시차 이후 예측(multi-step ahead prediction)으로 나뉘는데, 일반적으로 예측되는 시점이 멀어지면 마지막 관측 값에 대한 의존도가 감소하여 예측력이 떨어지는 결과를 보여준다.

본 연구에서는 모수적 방법과 LSTM 방법에 대해서 소개하고, 모의실험을 통해 1 시차 이후 예측에서 두 방법의 성능을 비교하고자 한다. 모수적 방법의 적용에는 모형 식별이 중요한데, 모형 식별 오류가 발생하였을 경우와 LSTM의 예측 성능을 비교하는 모의실험도 동시에 시행한다.

1.2 기존연구 고찰

Siarni-Namini와 Namin(2018)은 자기회귀누적이동평균(auto-regressive integrated moving average, ARIMA)모형을 가정하고 분석하는 방법과 LSTM으로 분석하는 방법을 비교하였다. ARIMA모형은 재 추정을 통한 1 시차 이후 예측(one-step out-of-sample forecast with re-estimation) 방법을 이용하였으며, LSTM은 1개의 에포크(epoch)와 4개의 뉴런(neurons)를 이용하였다. 이 두 가지 방법을 재무 데이터에 적용한 결과 LSTM과 같은 심층 학습 기반 알고리즘이 ARIMA모형보다 성능이 뛰어났지만 에포크의 수를 변경하여도 결과는 크게 개선되지 않는다는 것을 확인할 수 있었다.

Han(2018)은 시계열 예측을 위해 ARIMA와 LSTM 두 가지 방법을 사용하여 예측의 정확도를 비롯한 효율성에 대해 연구하였다. 미국과 호주의 맥주 데이터와 M3 competition의 시계열 데이터에 적용한 결과 강한 계절성 구조를 가진 ARIMA모형이 LSTM보다 예측이 잘 되는 것으로 나타났다. 하지만 강한 계절성이 없다면 ARIMA, LSTM 두 가지 방법 모두 예측이 제대로 수행되지 않는 경향이 있다는 것을 확인할 수 있었다. 또한, 이 연구에서는 복잡하고 훈련하기가 어려운 방법(LSTM)이라 하더라도 간단한 예측 방법(ARIMA)보다 반드시 정확한 예측을 제공하는 것이 아니라는 것을 주장하고 있다.

배성완과 유정석(2018)은 부동산 가격 지수 시계열 데이터에 시계열모형과 머신 러닝 방법을 적용하여 예측력을 비교하였다. 그 결과, 시장 상황이 일정한 추세를 보이면서 움직이는 경우에는 시계열모형과 머신 러닝 방법 모두 의미 있는 예측력을 보여주었으며, 시장이 비선형 형태로 급변하는 경우 시계열모형은 선형을 가정하는 한계점으로 인해 시장 예측이 어려운 반면 비선형 모델링이 가능한 머신 러닝 방법은 의미 있는 예측이 가능한 것을 보여주었다. 하지만 분석 자료, 변수 설정에 따라 분석 결과가 달라질 수 있기 때문에, 특정 방법이 우수하다고 단정하기에는 무리가 있다고 하였다. 특히 머신 러닝 방법인 SVM(support vector machine),

DNN(deep neural network), LSTM은 결과 값이 산출되는 이유를 설명할 수 없는데 이러한 점은 인과관계를 중요시하는 분야에서는 적용에 어려움이 있다고 판단하였다. 또한, 머신 러닝 방법은 모형을 최적화하기 위한 명확한 기준이 없다는 점, 적용 변수에 따라 결과가 달라질 수 있다는 점에서 추가적인 연구가 필요하다고 하였다.

Fu, Zhang과 Li(2016)은 실시간 교통 흐름 예측에서 LSTM과 게이트 순환 유닛(gated recurrent unit, GRU)을 ARIMA모형과 성능 비교하였는데 그 기준으로 MSE(mean squared error)와 MAE(mean absolute error)를 사용하였다. 그 결과, LSTM과 GRU신경망 방법은 ARIMA모형보다 우수한 성능을 가진다는 것을 보였지만 이 세 가지 방법의 성능은 크게 다르지 않았다.

Hansson(2017)은 미국, 브라질 및 스웨덴 증권 거래소의 예측 가능성을 조사하기 위해 S&P 500 지수, Bovespa 50 지수, OMX 30 지수를 사용하여 ARMA-GJRGARCH(Glosten-Jagannathan-Runkle GARCH)와 LSTM, Deep LSTM, Softmax LSTM, Softmax deep LSTM의 성능을 비교하였다. 그 결과, S&P 500 지수에서는 LSTM, Bovespa 50 지수에서는 ARMA-GJRGARCH, OMX 30 지수에서는 Softmax deep LSTM이 가장 잘 예측하는 것으로 나타났다. 신경망 모델이 작동되는 이유를 정확히 설명할 수 없지만 예측 성능이 거의 동등하거나 더 나은 결과를 보여주기 때문에 기존의 시계열모형과 경쟁할 수 있다고 판단하였다.

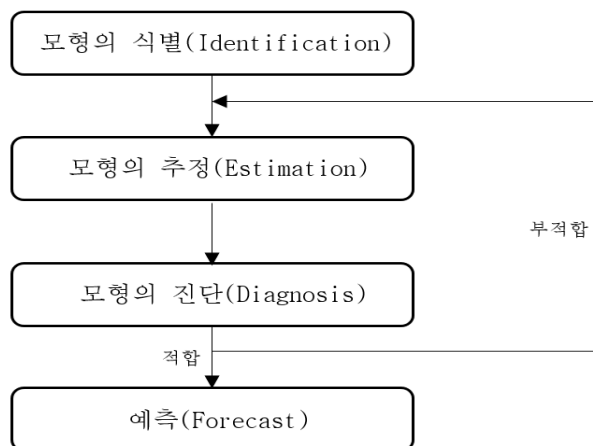
본 논문은 다음과 같이 구성되어 있다. 앞에서 제시된 제 1장 서론에서는 논문의 연구의 필요성 및 목적, 기존연구에 대해서 기술하였다. 제 2장에서는 본 연구의 이론적 배경이 되는 Box-Jenkins 방법론, 기계학습, 인공신경망, RNN, LSTM에 대하여 살펴보고, 제 3장에서는 연구방법의 개요와 모수적 시계열모형의 예측 방법, LSTM 예측 방법에 대해 설명하고자 한다. 제 4장에서는 모수적 방법과 LSTM의 예측 성능을 모의실험을 통해 비교한다. 마지막으로 제 5장에서는 본 연구의 결과를 요약하고 정리한다.

2. 연구의 이론적 배경

2.1 Box-Jenkins 방법론

Box와 Jenkins(1970)의 자기회귀이동평균은 Yule(1926)의 자기회귀모형과 Slutsky(1937)의 이동평균모형이 결합된 모형이다. 시간 순서를 갖는 확률변수들의 집합인 $\{Z_t, t=0,1,2,\dots,n\}$ 를 확률 과정(stochastic process)이라고 할 때 Box-Jenkins 방법론은 정상성을 만족한다는 가정으로부터 시작한다. 정상성이란 시계열의 확률적인 성질들이 시간의 흐름에 따라 불변이라는 것을 의미한다(조신섭, 손영숙, 2002).

Box-Jenkins의 3단계 모형 적합 과정은 모형의 식별, 추정, 진단으로 구성되어 있으며, 모형의 적합성을 진단하는 단계에서 적절한 모형이라 판단되지 않으면 식별 단계로 돌아가야 한다. 이 단계들을 거쳐 적합하다고 판단된 모형을 이용하여 예측(forecast) 한다.



[그림 2-1] Box-Jenkins 방법론

자기회귀모형은 시계열의 t 시점의 값 Z_t 를 그 이전 시점의 값 $\{Z_{t-1}, Z_{t-2}, \dots\}$ 로 회귀시킨 모형이며, p 시차 자기회귀모형($AR(p)$)은 다음과 같이 표현된다(이우리, 2013).

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t, \quad a_t \sim WN(0, \sigma^2) .$$

여기서 $WN(0, \sigma^2)$ 은 서로 독립이며 평균이 0, 분산이 σ^2 인 백색잡음을 나타낸다. 이 모형에서 현재 시점은 과거 시점들에 영향을 받는다는 의미를 가진다.

q 시차 이동평균모형($MA(q)$)은 다음과 같이 표현된다.

$$Z_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q}, \quad a_t \sim WN(0, \sigma^2) .$$

이 모형에서 현재 시점의 값은 현재 오차부터 과거 q 시점 전까지의 오차들에 영향을 받는다는 의미를 가지며, 이동평균 모형은 항상 정상성을 만족한다.

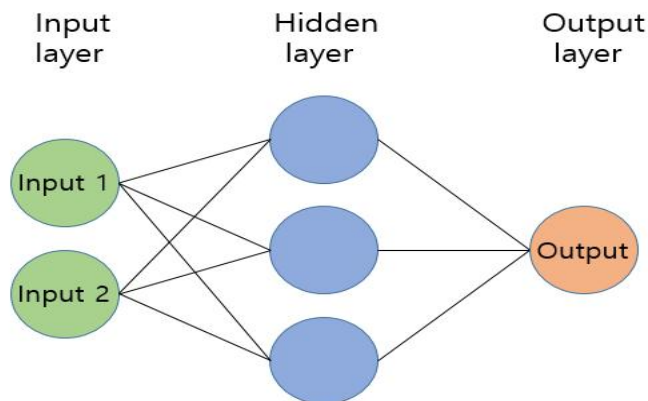
2.2 기계학습(machine learning)

기계학습은 크게 지도학습, 비지도학습, 강화학습 3가지로 나뉜다. 지도학습은 정답을 알려주어 학습시키는 것으로 분류와 회귀로 나뉘며, 분류는 사진이나 숫자를 인식하는 전형적인 지도학습 방법이고, 회귀는 학습 데이터의 특성을 이용하여 값을 예측하는 방법이다. 지도학습의 종류로 SVM, CNN(convolutional neural network), RNN, LSTM 등이 있다. 비지도학습은 정답을 알려주지 않아도 스스로 데이터를 학습하고 데이터에서 유사한 그룹끼리 분류하여 관계를 찾는 것이다. 예를 들어 다리가 4개인 동물로 분류한 후 그 중에서 목이 길거나 짧은 동물로 분류하고 또 다른 기준으로

분류하여 나타낸다. 하지만 데이터를 어떤 기준으로 분류한 것인지 결과 해석의 어려운 점이 있다. 비지도학습의 종류로 K-means, embedding 등이 있다. 강화학습은 현재의 상황을 인식하여 선택 가능한 행동들 중 특정 목적을 달성할 수 있도록 학습하는 방법이다. 그 예로 알파고(AlphaGo)와 같은 컴퓨터 프로그램, 로봇 등이 있다(박별터, 김한근, 김동훈, 2018).

2.3 인공신경망(artificial neural network, ANN)

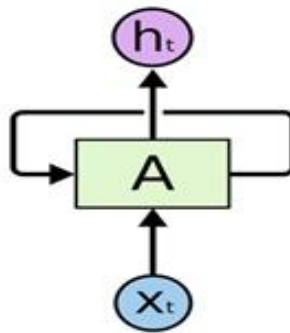
ANN은 McCulloch와 Pitts(1943)에 의해 제안된 모델이다. 입력 층(input layer), 은닉 층(hidden layer), 출력 층(output layer) 구조로 이루어져 있으며 각 층에는 입력 노드, 은닉 노드, 출력 노드를 포함하고 있다. 인공신경망 접근 방식의 목표는 인간의 두뇌와 같이 감각 뉴런을 통해 받은 자극을 연합 뉴런에서 정보를 판단하고, 운동 뉴런을 통해 반응을 하는 방식의 뉴런 네트워크와 유사하게 만들어 문제를 해결하는 것이다.



[그림 2-2] ANN 구조

2.4 순환신경망(recurrent neural network, RNN)

순환신경망은 인공지능의 한 종류로, [그림 2-3]과 같이 시간 t 에서 입력 층 X_t 와 출력 층 h_t , 은닉 층 A 를 갖고 있다. 과거의 입력 데이터에 대한 정보를 기억하여 현재의 입력 데이터에 영향을 미치는 순환적 구조의 특징을 가지고 있기 때문에 순차적인 시계열 데이터 처리에 유용하다. 그러나 은닉 층이 깊어지고 복잡해지면 역전파(back-propagation)를 통해 가중치를 개선해 나갈 때 기울기 소멸(vanishing gradient) 문제로 인해 학습이 제대로 되지 않는 상황이 발생하게 된다(Sundermeyer, Schlüter, & Ney, 2012). 이것을 장기 의존성(long term dependency) 문제를 가진다고 한다.



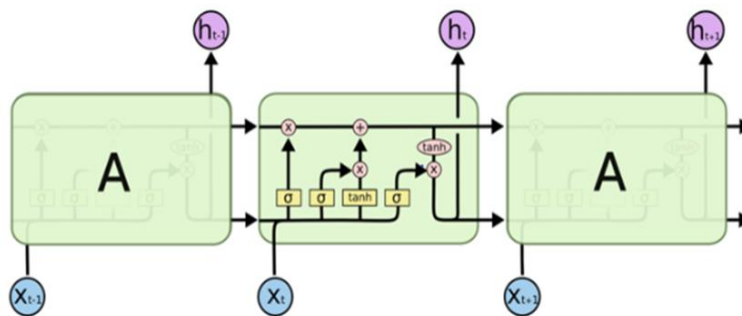
[그림 2-3] RNN 구조(Olah, 2015)

2.5 장단기메모리(long short term memory, LSTM)

순환신경망의 한 종류인 LSTM은 Hochreiter와 Schmidhuber(1997)에 의해 소개되었으며, 최근에 장기 의존성 문제를 가지는 다양한 분야에서 잘 작동하는 것으로 알려져 있어 많이 사용되는 구조이다. LSTM은 일시적으로 기억해야 할 것(short term memory)과 오랫동안 기억해야 할 것(long term

memory)을 가지고 있어 순환신경망을 학습할 때 발생할 수 있는 기울기 소멸 문제를 해결해 줄 수 있기 때문에 대규모의 깊은 신경망에서 더 효율적으로 작업할 수 있게 만들어 준다.

LSTM에는 순환신경망 구조의 잊기 게이트(forget-gate), 입력 게이트(input-gate), 출력 게이트(output-gate)가 존재한다. 잊기 게이트는 새로운 입력 값을 시그모이드(sigmoid) 함수를 통해 0과 1사이의 값으로 변환시켜 정보의 중요성을 판단하는 곳이다. 입력 게이트는 잊기 게이트를 통해 들어온 정보를 통해 과거의 정보를 업데이트하는 역할을 한다. 출력 게이트는 입력 게이트를 통해 들어온 정보를 이용하여 어떤 값을 출력할지 결정하는 역할을 한다.



[그림2-4] LSTM 구조(Olah, 2015)

본 연구에서는 미래 예측 성능 확인을 위해 지도 학습 중에서도 시계열 데이터를 학습하고 예측하는데 적합하다고 알려진 LSTM을 이용한다.

3. 연구방법

3.1 연구방법의 개요

본 연구에서는 ARMA모형을 따르는 난수를 생성하여 모수적 방법과 LSTM의 예측 성능을 비교한다. 모수적 방법은 모형 식별이 정확히 되었을 경우와 모형 식별 오류가 발생한 경우 두 가지를 고려한다. 시계열을 예측할 때는 일반적으로 마지막 관측 값에서부터 시차를 늘려가며 예측해 나가지만, 본 연구에서는 새로운 관측 값에 따라 1 시차 이후 예측 추정 값을 업데이트하는 방법을 사용하여 시차에 따라 예측력이 떨어지는 단점을 보완하였다.

LSTM은 25개의 에포크와 4개의 메모리 셀(memory cell)이 있는 은닉 층, 확률적 경사하강(stochastic gradient descent, SGD) 기법과 시그모이드 활성화 함수(sigmoid activation function)를 이용하였다. 학습 데이터(train data)를 통해 모형을 적합하고 테스트 데이터(test data)에 적용하여 예측 값을 구하였다. 또한, SGD 기법 대신 ADAM(adaptive moment estimation) 기법을 사용할 경우 예측력에 차이가 발생하는지 살펴보았다.

하나의 난수 데이터를 이용하여 예측 성능을 비교하는 것은 일반화된 결론을 내리기 힘들기 때문에 자기회귀모형, 이동평균모형의 정상성과 가역성 조건을 만족하는 모수 범위에서 모의실험을 50번 반복하여 시행하였다. 또한, 모수적 방법과 LSTM(SGD), 모수적 방법과 LSTM(ADAM)에 대해 각각이 표본 t-검정(two sample t-test)을 시행하여 제공근평균제곱오차(root mean square error, RMSE)의 평균값 간에 차이가 있는지 검증하였다. 또한, 모형 식별 오류가 발생한 경우와 LSTM(ADAM)에 대해서도 분석하였다.

분석에 사용한 프로그램은 Python 3.6.2이며, LSTM을 위한 모듈은 Keras를 사용하였다. 난수 생성된 시계열은 표본크기가 1000이며 그 중 2/3인 667개를 학습 데이터로, 나머지 333개를 테스트 데이터로 배정하였다.

본 연구에서는 다음과 같이 모의실험을 진행하였다.

[단계1] 시계열모형을 따르는 난수 생성

[단계2] 학습 데이터를 이용하여 모수모형을 통해 테스트 데이터의 예측 값 계산하고 RMSE 계산

[단계3] 학습 데이터를 이용하여 LSTM을 통해 테스트 데이터의 예측 값 계산하고 RMSE 계산

[단계4] 단계 1 - 3을 50번 반복하여 RMSE의 평균과 표준오차 계산

[단계5] 단계 1 - 4를 정상성, 가역성을 만족하는 모수 조건에서 모의실험 시행

3.2 ARMA모형을 이용한 모수적 예측 방법

평균이 0인 AR(1)모형은 $Z_t = \phi_1 Z_{t-1} + a_t$, $a_t \sim WN(0, \sigma^2)$ 을 따른다. t 시점에서 l 시차 이후 예측 값을 $\hat{Z}_t(l)$ 이라 할 때, $\hat{Z}_t(1) = \hat{\phi}_1 Z_t$ 와 같이 1 시차 이후 예측 값을 이용한다. 새로운 관측 값이 발생할 때마다 1 시차 이후 예측 값을 업데이트하여 시간이 지나도 예측을 잘 수행할 수 있도록 $\hat{Z}_{t+1}(1) = \hat{\phi}_1 Z_{t+1}$, $\hat{Z}_{t+2}(1) = \hat{\phi}_1 Z_{t+2}$ 와 같이 예측 값(\hat{Z}_t)을 구한다.

예측 값과 관측 값의 RMSE를 구하여 예측 성능을 계산한다. AR(2), MA(1), MA(2), ARMA(1,1)모형에서도 같은 방법을 이용하며 구체적인 모형별 예측 식은 <표 3-1>과 같다.

<표 3-1> 모형별 예측 식

모형	예측 식
AR(1)	$\hat{Z}_{t-1}(1) = \hat{\phi}_1 Z_{t-1}$
	$\hat{Z}_t(1) = \hat{\phi}_1 Z_t$
	$\hat{Z}_{t+1}(1) = \hat{\phi}_1 Z_{t+1}$
	\vdots
AR(2)	$\hat{Z}_{t-1}(1) = \hat{\phi}_1 Z_{t-1} + \hat{\phi}_2 Z_{t-2}$
	$\hat{Z}_t(1) = \hat{\phi}_1 Z_t + \hat{\phi}_2 Z_{t-1}$
	$\hat{Z}_{t+1}(1) = \hat{\phi}_1 Z_{t+1} + \hat{\phi}_2 Z_t$
	\vdots
MA(1)	$\hat{Z}_{t-1}(1) = -\hat{\theta}_1 \hat{a}_{t-1}$
	$\hat{Z}_t(1) = -\hat{\theta}_1 \hat{a}_t$
	$\hat{Z}_{t+1}(1) = -\hat{\theta}_1 \hat{a}_{t+1}$
	\vdots
MA(2)	$\hat{Z}_{t-1}(1) = -\hat{\theta}_1 \hat{a}_{t-1} - \hat{\theta}_2 \hat{a}_{t-2}$
	$\hat{Z}_t(1) = -\hat{\theta}_1 \hat{a}_t - \hat{\theta}_2 \hat{a}_{t-1}$
	$\hat{Z}_{t+1}(1) = -\hat{\theta}_1 \hat{a}_{t+1} - \hat{\theta}_2 \hat{a}_t$
	\vdots
ARMA(1,1)	$\hat{Z}_{t-1}(1) = \hat{\phi}_1 Z_{t-1} - \hat{\theta}_1 \hat{a}_{t-1}$
	$\hat{Z}_t(1) = \hat{\phi}_1 Z_t - \hat{\theta}_1 \hat{a}_t$
	$\hat{Z}_{t+1}(1) = \hat{\phi}_1 Z_{t+1} - \hat{\theta}_1 \hat{a}_{t+1}$
	\vdots

3.3 LSTM

본 연구에서는 최적화 기법으로 SGD 기법, ADAM 기법을 사용한다. 경사 하강(gradient descent, GD) 기법은 예측 값과 실제 값의 차이를 최소화하는 최적의 가중치(weight)를 구하기 위해서 출력 층에서부터 미분을 통해 기울기(gradient)를 구하여 가중치 값을 업데이트하는 방법이며, SGD 기법은 각 뉴런에서 가중치를 구할 때 일부 데이터에 대해서만 가중치를 계산하는 기법이다(Bottou, 2010). 하지만, SGD 기법은 현재의 기울기만으로 다음 값을 추적하기 때문에 방향성이 존재하지 않는다.

이러한 단점을 보완하기 위해 사용하는 ADAM 기법은 과거의 정보에 가중치를 적게 부여하는 AdaGrad(adaptive gradient)와 재조정된 기울기의 모멘텀(momentum)을 사용하여 매개 변수를 업데이트하는 RMSProp(root mean square propagation)을 결합한 형태로서 뉴런에서 매 단계 사이즈와 방향을 적절하게 적용하는 매우 효과적인 기법이다(Kingma & Ba, 2014).

본 연구에 사용된 시그모이드 함수는 대표적인 활성화 함수로 다음과 같이 주어진다.

$$\sigma(x) = \frac{1}{1 + e^{-x}} .$$

여기서 입력 값과 가중치를 곱한 값들의 합인 x 가 마이너스 무한대($-\infty$)로 갈 때 $\sigma(x)$ 는 0으로 변환하며, 플러스 무한대($+\infty$)로 갈 때 $\sigma(x)$ 는 1로 변환하여 출력한다.

4. 모수적 방법과 LSTM의 예측 성능 비교

4.1 AR(1)모형과 MA(1)모형

AR(1)모형과 MA(1)모형에서 모수적 방법과 LSTM(SGD), LSTM(ADAM)의 예측 성능을 비교한 모의실험 결과는 각각 <표 4-1>, <표 4-2>에 제시되어있다. 이 표에서 *는 모수적 방법과 비교해서 유의수준 0.05에서 유의한 차이가 있는 것을 나타내며, **는 모수적 방법과 비교해서 유의수준 0.01에서 유의한 차이가 있는 것을 나타낸다.

먼저, AR(1)모형에서 LSTM(SGD)은 $|\phi_1|$ 가 비교적 작은 $\phi_1 = -0.4 \sim 0.3$ 범위에서는 모수적 방법과 통계적으로 유의한 차이가 없었지만, $|\phi_1|$ 가 커질수록 모수적 방법이 유의하게 성능이 좋은 것으로 나타났다. LSTM(ADAM)은 $\phi_1 = 0.9$ 에서만 모수적 방법이 통계적으로 유의하게 좋은 것으로 나타났다.

MA(1)모형에서 LSTM(SGD)은 $|\theta_1|$ 가 비교적 작은 $\theta_1 = -0.3 \sim 0.2$ 범위에서는 모수적 방법과 통계적으로 유의한 차이가 없었지만, $|\theta_1|$ 가 커질수록 모수적 방법이 유의하게 성능이 좋은 것으로 나타났다. LSTM(ADAM)은 $\theta_1 = -0.4 \sim 0.4$ 범위에서는 모수적 방법과 유의한 차이가 없었지만, 이 범위 밖에서는 모수적 방법이 유의하게 좋은 것으로 나타났다.

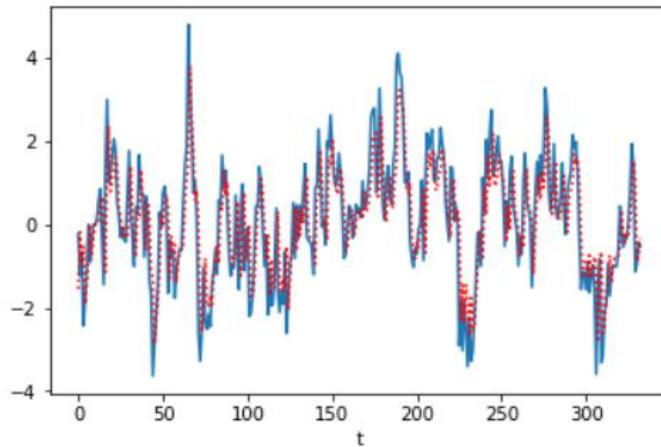
즉, AR(1), MA(1)모형에서 모두 모수적 방법이 LSTM(SGD), LSTM(ADAM)보다 다소간의 예측 성능이 우수한 것으로 나타났고, LSTM(ADAM)의 예측 성능은 상대적으로 모수적 방법과 차이가 적은 것으로 나타났다.

<표 4-1> AR(1)모형에서 모수적 방법과 LSTM 예측 성능 비교

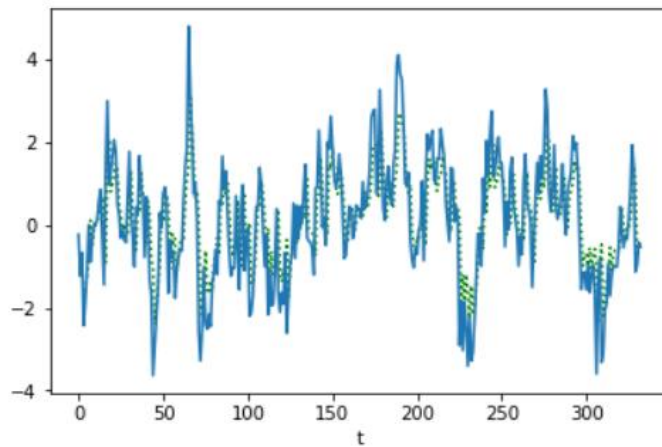
ϕ_1	모수적 방법	SGD	ADAM
0.9	0.996(0.044)	1.064(0.128)**	1.007(0.046)*
0.8	0.996(0.044)	1.055(0.100)**	1.003(0.045)
0.7	0.996(0.044)	1.045(0.078)**	1.001(0.045)
0.6	0.996(0.044)	1.036(0.061)**	1.000(0.045)
0.5	0.996(0.044)	1.027(0.052)**	1.000(0.045)
0.4	0.996(0.044)	1.019(0.048)*	1.000(0.045)
0.3	0.996(0.044)	1.012(0.048)	0.999(0.045)
0.2	0.996(0.044)	1.007(0.044)	0.999(0.045)
0.1	0.996(0.044)	1.003(0.044)	0.999(0.045)
-0.1	0.996(0.044)	1.002(0.046)	0.999(0.045)
-0.2	0.996(0.044)	1.004(0.047)	0.999(0.046)
-0.3	0.996(0.044)	1.009(0.049)	1.000(0.046)
-0.4	0.996(0.044)	1.014(0.053)	1.000(0.046)
-0.5	0.996(0.044)	1.022(0.059)*	1.001(0.046)
-0.6	0.996(0.044)	1.032(0.072)**	1.002(0.046)
-0.7	0.996(0.044)	1.048(0.095)**	1.003(0.046)
-0.8	0.996(0.044)	1.074(0.149)**	1.006(0.045)
-0.9	0.996(0.044)	1.132(0.313)**	1.011(0.046)

<표 4-2> MA(1)모형에서 모수적 방법과 LSTM 예측 성능 비교

θ_1	모수적 방법	SGD	ADAM
0.9	0.996(0.044)	1.198(0.062)**	1.169(0.057)**
0.8	0.996(0.044)	1.147(0.059)**	1.119(0.054)**
0.7	0.996(0.044)	1.104(0.055)**	1.079(0.051)**
0.6	0.996(0.044)	1.070(0.052)**	1.047(0.049)**
0.5	0.996(0.044)	1.045(0.049)**	1.025(0.047)**
0.4	0.996(0.044)	1.027(0.047)**	1.011(0.046)
0.3	0.996(0.044)	1.015(0.045)*	1.003(0.045)
0.2	0.996(0.044)	1.007(0.044)	1.000(0.045)
0.1	0.996(0.044)	1.003(0.044)	0.999(0.045)
-0.1	0.996(0.044)	1.002(0.046)	0.999(0.045)
-0.2	0.996(0.044)	1.005(0.047)	1.000(0.046)
-0.3	0.996(0.044)	1.011(0.049)	1.003(0.046)
-0.4	0.996(0.044)	1.022(0.052)**	1.011(0.047)
-0.5	0.996(0.044)	1.039(0.055)**	1.025(0.049)**
-0.6	0.996(0.044)	1.063(0.059)**	1.047(0.051)**
-0.7	0.996(0.044)	1.097(0.064)**	1.078(0.054)**
-0.8	0.996(0.044)	1.139(0.068)**	1.118(0.056)**
-0.9	0.996(0.044)	1.190(0.072)**	1.167(0.059)**



[그림 4-1] AR(1)모형에서 모수적 방법 예측 시계열 그래프



[그림 4-2] AR(1)모형에서 LSTM(SGD) 예측 시계열 그래프

[그림 4-1]은 AR(1)모형에서 $\phi_1 = 0.8$ 일 때 생성된 하나의 시계열에 대해 테스트 데이터의 시계열 그래프(실선)와 모수적 방법의 예측 시계열 그래프(점선)를 보여주고 있고, [그림 4-2]는 같은 시계열에 대해 LSTM(SGD) 예측 시계열 그래프(점선)를 보여주고 있다.

두 예측 시계열 그래프를 보아 190, 230시점에서 예측 성능에 약간의 차이가 있어 보인다.

4.2 AR(2)모형

AR(2)모형에서 모수적 방법과 LSTM(SGD), LSTM(ADAM)과의 예측 성능 비교하고 LSTM(ADAM)과 ‘AR(1)으로 모형 식별 오류’와의 예측 성능을 비교한 모의실험 결과는 <표 4-3>에 제시되어있다. 이 표에서 *는 유의수준 0.05, **는 유의수준 0.01에서 유의한 차이가 있는 것을 나타낸다.

AR(2)모형에서 LSTM(SGD)는 $|\phi_1|$ 가 비교적 작은 $\phi_1 = -0.2 \sim 0.1$ 범위에서는 모수적 방법과 통계적으로 유의한 차이가 없었지만, 이 범위 밖에서는 모수적 방법이 통계적으로 유의하게 성능이 좋은 것으로 나타났다. LSTM(ADAM)은 $\phi_1 = -0.8$ 에서만 모수적 방법이 통계적으로 유의하게 좋은 것으로 나타났다.

또한, AR(2)모형을 AR(1)모형으로 식별 오류가 발생한 경우에는 $\phi_1 = -0.2 \sim 0.1$ 범위에서는 LSTM(ADAM)과 통계적으로 유의한 차이가 없었지만, 이 범위 밖에서는 통계적으로 유의한 차이가 나타남으로써 LSTM(ADAM)의 성능이 좋은 것으로 나타났다.

즉, AR(2)모형에서는 모형 식별이 정확한 경우에는 모수적 방법이 LSTM(ADAM)과 성능이 비슷하고, LSTM(SGD)보다 성능이 우수하였으며, 모형 식별 오류가 발생한 경우에는 모수적 방법보다 LSTM(ADAM)의 성능이 우수한 것을 알 수 있다.

<표 4-3> AR(2)모형에서 모수적 방법과 LSTM 예측 성능 비교

ϕ_1	ϕ_2	모수적 방법	SGD	ADAM	AR(1) 식별 오류
0.8	-0.1	0.997(0.044)	1.092(0.055)**	1.012(0.047)	1.333(0.068)**
0.7	-0.1	0.997(0.044)	1.079(0.053)**	1.010(0.046)	1.280(0.064)**
0.6	-0.1	0.996(0.044)	1.067(0.051)**	1.008(0.045)	1.227(0.061)**
0.5	-0.1	0.996(0.044)	1.054(0.049)**	1.007(0.045)	1.176(0.057)**
0.4	-0.1	0.996(0.044)	1.043(0.048)**	1.007(0.447)	1.128(0.054)**
0.3	-0.1	0.996(0.044)	1.031(0.047)**	1.007(0.045)	1.084(0.051)**
0.2	-0.1	0.996(0.044)	1.022(0.046)**	1.006(0.045)	1.046(0.048)**
0.1	-0.1	0.996(0.044)	1.014(0.046)	1.005(0.045)	1.017(0.046)
-0.1	-0.1	0.996(0.044)	1.006(0.046)	1.003(0.046)	1.000(0.045)
-0.2	-0.1	0.996(0.043)	1.008(0.047)	1.004(0.046)	1.022(0.049)
-0.3	-0.1	0.996(0.043)	1.016(0.049)*	1.005(0.046)	1.072(0.056)**
-0.4	-0.1	0.996(0.043)	1.029(0.052)**	1.007(0.047)	1.157(0.068)**
-0.5	-0.1	0.996(0.043)	1.052(0.056)**	1.008(0.047)	1.289(0.085)**
-0.6	-0.1	0.996(0.043)	1.086(0.066)**	1.010(0.048)	1.482(0.110)**
-0.7	-0.1	0.997(0.043)	1.142(0.084)**	1.013(0.049)	1.760(0.146)**
-0.8	-0.1	0.997(0.043)	1.233(0.124)**	1.017(0.051)*	2.178(0.204)**

4.3 MA(2)모형

MA(2)모형에서 모수적 방법과 LSTM(SGD), LSTM(ADAM)과의 예측 성능 비교하고 LSTM(ADAM)과 ‘MA(1)으로 모형 식별 오류’와의 예측 성능을 비교한 모의실험 결과는 <표 4-4>에 제시되어있다. 이 표에서 *는 유의수준 0.05, **는 유의수준 0.01에서 유의한 차이가 있는 것을 나타낸다.

MA(2)모형에서 LSTM(SGD)은 $|\theta_1|$ 가 비교적 작은 $\theta_1 = 0.1 \sim 0.2$ 범위에서는 모수적 방법과 통계적으로 유의한 차이가 없었지만, 이 범위 밖에서는 모수적 방법이 통계적으로 유의하게 성능이 좋은 것으로 나타났다. LSTM(ADAM)은 $\theta_1 = -0.7$ 에서만 모수적 방법이 통계적으로 유의하게 좋은 것으로 나타났다.

반면, MA(2)모형을 MA(1)모형으로 식별 오류가 발생한 경우에는 모든 범위에서 LSTM(ADAM)과 통계적으로 유의한 차이가 없었다.

즉, MA(2)모형에서는 모형 식별이 정확한 경우와 모형 식별 오류가 발생한 경우의 모수적 방법이 LSTM(ADAM)과 성능이 비슷하고, LSTM(SGD)보다는 성능이 우수한 것을 알 수 있다.

<표 4-4> MA(2)모형에서 모수적 방법과 LSTM 예측 성능 비교

θ_1	θ_2	모수적 방법	SGD	ADAM	MA(1) 식별 오류
0.7	0.2	0.997(0.043)	1.123(0.056)**	1.011(0.047)	1.017(0.045)
0.6	0.2	0.997(0.043)	1.084(0.052)**	1.009(0.046)	1.017(0.045)
0.5	0.2	0.997(0.043)	1.053(0.049)**	1.010(0.045)	1.017(0.045)
0.4	0.2	0.997(0.043)	1.032(0.047)**	1.011(0.044)	1.016(0.045)
0.3	0.2	0.997(0.043)	1.019(0.045)*	1.009(0.044)	1.016(0.045)
0.2	0.2	0.996(0.044)	1.013(0.045)	1.006(0.045)	1.016(0.045)
0.1	0.2	0.996(0.044)	1.014(0.046)	1.005(0.045)	1.015(0.046)
-0.1	0.2	0.996(0.044)	1.028(0.049)**	1.005(0.046)	1.015(0.046)
-0.2	0.2	0.996(0.044)	1.037(0.051)**	1.007(0.047)	1.015(0.047)
-0.3	0.2	0.996(0.044)	1.046(0.053)**	1.009(0.048)	1.014(0.047)
-0.4	0.2	0.996(0.044)	1.053(0.054)**	1.010(0.048)	1.014(0.047)
-0.5	0.2	0.996(0.044)	1.060(0.058)**	1.010(0.048)	1.014(0.048)
-0.6	0.2	0.996(0.044)	1.067(0.063)**	1.011(0.048)	1.013(0.048)
-0.7	0.2	0.997(0.044)	1.078(0.070)**	1.015(0.049)*	1.013(0.048)

4.4 ARMA(1,1)모형

ARMA(1,1)모형에서 모수적 방법과 LSTM(SGD), LSTM(ADAM)과의 예측 성능 비교하고 LSTM(ADAM)과 ‘MA(1)으로 모형 식별 오류’와의 예측 성능을 비교한 모의실험 결과는 <표 4-5>에 제시되어있다. 이 표에서 *는 유의수준 0.05, **는 유의수준 0.01에서 유의한 차이가 있는 것을 나타낸다.

ARMA(1,1)모형에서 LSTM(SGD)은 $|\phi_1|$ 가 비교적 작은 $\phi_1 = -0.4 \sim 0.2$ 범위에서는 모수적 방법과 통계적으로 유의한 차이가 없었지만, 이 범위 밖에서는 모수적 방법이 통계적으로 유의하게 성능이 좋은 것으로 나타났다. LSTM(ADAM)은 $\phi_1 = -0.6 \sim -0.8$ 범위에서만 모수적 방법이 통계적으로 유의하게 좋은 것으로 나타났다.

또한, ARMA(1,1)모형을 MA(1)모형으로 식별 오류가 발생한 경우에는 $\phi_1 = -0.4 \sim 0.4$ 범위에서는 LSTM(ADAM)과 통계적으로 유의한 차이가 없었지만, 이 범위 밖에서는 통계적으로 유의한 차이가 나타남으로써 LSTM(ADAM)의 성능이 좋은 것으로 나타났다.

즉, ARMA(1,1)모형에서 모형 식별이 정확한 경우에는 모수적 방법이 LSTM(ADAM)과 성능이 비슷하고, LSTM(SGD)보다는 성능이 우수하였으며, 모형 식별 오류가 발생한 경우에는 모수적 방법보다 LSTM(ADAM)의 성능이 우수한 것을 알 수 있다.

<표 4-5> ARMA(1,1)모형에서 모수적 방법과 LSTM 예측 성능 비교

ϕ_1	θ_1	모수적 방법	SGD	ADAM	MA(1) 식별 오류
0.8	0.1	1.000(0.044)	1.063(0.110)**	1.008(0.045)	1.258(0.084)**
0.7	0.1	0.999(0.044)	1.052(0.086)**	1.005(0.045)	1.133(0.062)**
0.6	0.1	0.998(0.044)	1.043(0.069)**	1.004(0.045)	1.069(0.052)**
0.5	0.1	0.997(0.044)	1.035(0.057)**	1.002(0.045)	1.033(0.048)**
0.4	0.1	0.997(0.044)	1.027(0.050)**	1.001(0.045)	1.013(0.045)
0.3	0.1	0.997(0.044)	1.019(0.047)**	1.001(0.045)	1.003(0.044)
0.2	0.1	0.997(0.044)	1.012(0.045)	1.000(0.045)	0.998(0.044)
0.1	0.1	0.997(0.044)	1.007(0.044)	0.999(0.045)	0.996(0.044)
-0.2	0.1	0.997(0.043)	1.002(0.046)	0.999(0.045)	0.996(0.044)
-0.3	0.1	0.996(0.043)	1.005(0.047)	1.000(0.045)	0.998(0.044)
-0.4	0.1	0.996(0.043)	1.010(0.049)	1.001(0.045)	1.003(0.045)
-0.5	0.1	0.996(0.044)	1.017(0.054)*	1.002(0.045)	1.015(0.047)**
-0.6	0.1	0.997(0.044)	1.027(0.063)**	1.003(0.045)*	1.040(0.051)**
-0.7	0.1	0.997(0.044)	1.042(0.082)**	1.005(0.045)*	1.090(0.059)**
-0.8	0.1	0.998(0.045)	1.067(0.128)**	1.008(0.045)*	1.193(0.079)**

5. 결론

시계열 예측은 경제, 사회, 금융 등 많은 분야에서 중요한 역할을 하고 있기 때문에 정확한 예측을 위한 분석방법의 연구가 필요하다.

본 연구에서는 ARMA모형을 따르는 난수를 생성하여 모수적 방법과 LSTM의 예측 성능을 비교하였다. 모수적 방법은 모형 식별이 정확히 되었을 경우와 모형 식별 오류가 발생한 경우 두 가지를 고려하였으며, 새로운 관측 값에 따라 1 시차 이후 예측 값을 업데이트하는 방법을 사용하여 시차에 따라 예측력이 떨어지는 단점을 보완하였다. LSTM은 SGD 기법과 ADAM 기법을 사용하여 분석하였다. RMSE를 통해 모수적 방법과 LSTM의 예측 성능을 평가하였으며, 이 표본 t-검정을 시행하여 평균 간 차이에 대해 유의성을 검증하였다.

연구 결과, 모형 식별이 정확한 경우에는 모수적 방법이 LSTM(ADAM)과 성능이 비슷하고, LSTM(SGD)보다는 성능이 우수하였다. 또한, 모형 식별 오류가 발생한 경우에는 LSTM(ADAM)의 성능이 모수적 방법보다 많은 경우에서 우수한 것을 알 수 있었다.

하지만 계산속도 면에서는 LSTM을 이용한 예측보다 모수적 방법이 훨씬 빠르고 효율적이었으며, LSTM은 최적화 기법 등에 따라 예측 성능이 달라진다는 점에서 주의해야 할 필요가 있다. 또한, 모형 식별 오류가 발생한 경우에는 LSTM(ADAM)의 예측 성능이 더 나았기 때문에 모수적 방법은 모형 식별에 주의해야 할 필요가 있다.

추후 연구과제로 고려할 수 있는 것은 다음과 같이 두 가지로 정리할 수 있을 것이다.

첫째, 대부분의 경제, 사회 및 금융 데이터 등은 비정상 시계열 데이터이기 때문에 본 연구에서 다룬 정상적 시계열모형에서 더 나아가 추가적으로 ARCH, GARCH 등 비정상시계열 모형에서 모수적 방법과 LSTM의 성능 비교 연구가 필요할 것이다.

둘째, LSTM은 빅데이터 분석에 맞춰진 구조로서 표본의 크기가 1000인

경우는 작은 값이기 때문에 예측 성능이 좋지 않게 나타났을 수도 있다.
따라서 표본의 크기를 더 높인 후 분석하거나 실제 데이터를 이용하여 예측 성능을 비교할 필요성이 있다.

참고문헌

- 박별터, 김한근, 김동훈. (2018). 인공지능, 그리고 자율운항의 상용화. *로봇과 인간*, 15(2), 45-54.
- 배성완, 유정석. (2018). 머신 러닝 방법과 시계열 분석 모형을 이용한 부동산 가격지수 예측. *주택연구*, 26, 107-133.
- 이우리. (2013). *시계열 분석과 예측*. 서울: 탐진.
- 조신섭, 손영숙. (2002). *SAS/ETS를 이용한 시계열분석*. 서울: 율곡출판사
- Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. *Proceedings of the 19th International Conference on Computational Statistics*, 177-186.
- Box, G. E. P., & Jenkins, G. M. (1970). *Time-series analysis: forecasting and control*. San Francisco: Holden-day.
- Fu, R., Zhang, Z., & Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. *Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 324-328.
- Han, J. (2018). *Comparing models for time series analysis*. Wharton Research Scholars. 162. University of Wharton, Pennsylvania.
- Hansson, M. (2017). *On stock return prediction with LSTM networks*. University of Lund, Lund.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint arXiv:1412.6980*.

- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115-133.
- Olah, C. (2015). Understanding LSTM networks-colah's blog. *Colah.Github.Io*, Retrieved November 27, 2018, from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Fifteenth Annual Conference of the International Speech Communication Association*, 338-342.
- Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: ARIMA vs. LSTM. *ArXiv Preprint arXiv:1803.06386*.
- Slutzky, E. (1937). The summation of random causes as the source of cyclic processes. *Econometrica*, 5(2), 105-146.
- Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. *Thirteenth Annual Conference of the International Speech Communication Association*, 194-197.
- Yule, G. U. (1926). Why do we sometimes get nonsense-correlations between time-series?--a study in sampling and the nature of time-series. *Journal of the Royal Statistical Society*, 89(1), 1-63.

A Study on LSTM Performance in the Prediction of Time Series Model

Park, Hye Won

Department of Statistics

Graduate School

Keimyung University

(Supervised by Professor Ha, Jeongcheol)

(Abstract)

Time series forecasting is to predict present or future observation using historical data observed over time. For example, ARMA model which combines Auto-Regressive (AR) model and Moving Average (MA) model has been widely used. However, in recent years, a lot of data and information has accumulated, and the performance of the learning algorithms such as deep learning has improved due to the development of high performance computers. In this study, we compare the performance of parametric model prediction methods with that of LSTM (long short term memory) based on RMSE (root mean square error) for one-step ahead prediction. In parametric model prediction method, we also consider the case where model is mis-specified. Our simulation results indicate that the parametric method shows almost the same

performance as LSTM with ADAM (adaptive moment estimation) method and that it is better than LSTM with SGD (stochastic gradient descent) algorithm. And it is also shown that LSTM with ADAM algorithm shows better performance than mis-specified parametric prediction in many cases.

시계열모형의 예측에서 LSTM의 성능에 대한 연구

박 혜 원

계명대학교 대학원

통계학과

(지도교수 하 정 철)

(초록)

시계열 예측은 시간의 흐름에 따라 관측된 과거의 데이터를 이용하여 미래를 예측하는 방법으로, 자기회귀(auto-regressive, AR)모형, 이동평균(moving average, MA)모형, 그리고 두 모형이 결합된 자기회귀이동평균(ARMA)모형이 널리 사용되어 왔다. 하지만 최근에는 수많은 데이터와 정보가 쌓이고, 고성능 컴퓨터의 등장으로 딥 러닝(deep learning)을 이용한 지도 학습 알고리즘의 성능이 향상되어 이를 활용한 방법이 많이 개발되었다. 이 연구에서는 모의실험을 통해 모수적 방법과 LSTM(long short term memory)의 성능을 RMSE(root mean square error)에 기반하여 비교하였다. 모수적 방법은 정확한 모형 식별을 하였을 때와 모형 식별 오류가 발생한 모수적 방법을 동시에 고려하였으며, 예측에는 1 시차 이후 예측 값을 이용하였다. LSTM은 최적화 기법으로 확률적 경사하강 기법(stochastic gradient descent optimizer, SGD)과 ADAM(adaptive moment estimation)

기법을 사용하였다. 모의실험 결과, 모형 식별이 정확한 경우에는 모수적 방법이 ADAM 기법을 이용한 LSTM과 성능이 비슷하고, SGD 기법을 이용한 LSTM보다는 성능이 우수하였다. 또한, 모형 식별 오류가 발생한 경우에는 ADAM 기법을 이용한 LSTM의 성능이 모수적 방법보다 많은 경우에서 우수한 성능을 보였다.