



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](#)

석사학위논문

LSTM/GRU 순환신경망을 이용한 시계열데이터 예측

Forecasting Time-series Data

Using LSTM/GRU Recurrent Neural Networks



한국방송통신대학교 대학원

정 보 과 학 과

김 호 현

2017년

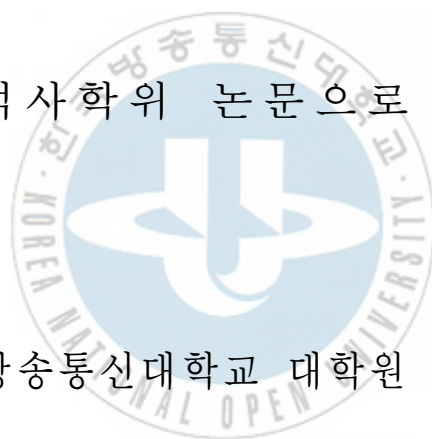
LSTM/GRU 순환신경망을 이용한 시계열데이터 예측

Forecasting Time-series Data

Using LSTM/GRU Recurrent Neural Networks

지도교수 이 관 용

이 논문을 석사학위 논문으로 제출함



한국방송통신대학교 대학원

정 보 과 학 과

김 호 현

2017년 7월

김 호 현의
석사학위 논문을 인준함

심사위원장 인

심사위원 인

심사위원 인



한국방송통신대학교 대학원

2017년 6월

논 문 요 약

LSTM/GRU 순환신경망을 이용한 시계열데이터 예측

홍 길 동

한국방송통신대학교 대학원

정보과학과

(지도교수 이관용)

주가지수, 환율 등과 같은 시계열자료에 대한 예측은 이해관계자들에게 미치는 영향이 크기 때문에 여러 연구자들의 주요 연구주제가 되어왔다. 지금까지 시계열데이터 예측에는 ARIMA와 같은 통계적인 기법들과 인공신경망을 활용한 방법들이 주로 사용되어 왔으며, 일부 연구자들은 예측 정확도를 높이기 위하여 ARIMA와 인공신경망을 결합한 혼합 모델을 제시하였다.

이 논문에서는 시계열데이터 예측문제에 LSTM/GRU 순환신경망을 사용하는 새로운 방법을 제시한다. LSTM/GRU 순환신경망과 함께 더불어 최근 딥러닝 분야의 성과인 Dropout 과다학습 방지와 Xavier 가중치 초기화 기법을 적용하여 기존의 방법론보다 향상된 예측 정확도를 갖는 시계열데이터 예측 모델을 구축하였다. 공개 데이터에 대한 실험을 통해 제안모델이 기존의 모델들과 비교하여 더 높은 예측 정확도를 보임을 입증한다.

- 목 차 -

제1장 서론	1
1.1 연구의 동기와 목적	1
1.2 연구의 범위 및 방법	2
제2장 배경 이론 및 관련 연구	3
2.1 통계기법 예측 방법	3
2.1.1. 단순 예측법	3
2.1.2. 회귀 예측법	3
2.1.3. 시계열데이터 분해법	4
2.1.4. 지수평활법	5
2.1.5. 정상성과 차분	5
2.1.6. ARIMA	6
2.2 인공신경망	8
2.2.1. 다층 퍼셉트론	8
2.2.2. Elman 신경망	10
2.2.3. NARX 신경망	10
2.3 혼합 모델	11
2.3.1. Zhang 혼합 모델	12
2.3.2. Khashei & Bijari 혼합 모델	12
2.3.3. Babu & Reddy 혼합 모델	13
제3장 제안 모델	15
3.1 순환신경망	15
3.1.1. 표준 순환신경망	16
3.2 Long Short Term Memory	17
3.2.1. Gated Recurrent Unit	18

3.3 Dropout	19
3.4 Xavier 가중치 초기화	20
3.5 제안모델	21
3.5.1. 제안모델 구조	21
3.5.2. 모델의 학습과정	21
제4장 실험 결과	23
4.1. 실험 환경 및 비교 대상 모델	23
4.2. 딥러닝 프레임워크	24
4.3. Lynx data 예측 실험	25
4.4. Sunspot data 예측 실험	28
4.5. 환율 Data 예측 실험	31
4.6. 학습시간과 실행시간 비교	33
제5장 결론	35
참고문헌	36
영문초록	38
감사의 글	40

- 표 목차 -

<표 4-1> Lynx data 실험에 사용된 모델의 형태	25
<표 4-2> Lynx data 제안모델과 기존 예측모델 간의 정확도 비교	27
<표 4-3> Sunspot data 실험에 사용된 모델의 형태	28
<표 4-4> Sunspot data 제안모델과 기존 예측모델 간의 정확도 비교	30
<표 4-5> 환율 data 실험에 사용된 모델의 형태	31
<표 4-6> 환율 data 제안모델과 기존 예측모델 간의 정확도 비교	33
<표 4-7> 모델별 학습시간과 실행시간 비교	34



- 그림 목차 -

<그림 2-1> 시계열데이터 분해 예	4
<그림 2-2> 뉴런의 작동방식	8
<그림 2-3> 전방향신경망의 구조	9
<그림 2-4> Elman 신경망의 구조	10
<그림 2-5> NARX 신경망의 구조	11
<그림 2-6> Khashei & Bijari 혼합 모델의 신경망 구조	13
<그림 2-7> Babu & Reddy 혼합 모델의 구조	14
<그림 3-1> 순환신경망의 구조	16
<그림 3-2> 표준 순환신경망의 셀 구조	16
<그림 3-3> LSTM 순환신경망의 셀 구조	17
<그림 3-4> GRU 셀의 구조	18
<그림 3-5> Dropout 구조	19
<그림 3-6> 제안 순환신경망 모델의 구조	21
<그림 4-1> Lynx data 각 모델별 실제값과 예측값 비교	26
<그림 4-2> Sunspot data 각 모델별 실제값과 예측값 비교	29
<그림 4-3> 환율 data 각 모델별 실제값과 예측값 비교	32

제1장 서론

1.1 연구의 동기와 목적

시계열데이터(time-series data)는 주가, 환율, 유가, 매출액, 출국자수 등과 같이 시간 간격을 두고 측정된 일련의 값이다. 기업이나 공공기관을 포함한 이해관계자들은 시계열데이터를 관찰하고 분석해야 하며, 데이터 값의 미래 움직임을 예측해야 하는 경우도 흔히 있으며, 이는 조직의 성과에 큰 영향을 준다. 예를 들어, 환율이나 유가를 예측하는 일은 사업계획을 수립하는데 필수적이며, 항공사에서는 출국자 수를 예측하는 일은 비행기나 인력의 운영계획을 세우기 위해 필요하다. 제조업체라면 생산계획을 수립하기 위해서는 판매액을 예측해야 한다. 펀드 매니저라면 주식이나 유가의 향방을 예측하는 일이 무엇보다 중요할 것이다. 이와 같이 시계열데이터의 예측은 우리가 현실에서 직면하게 되는 문제이므로 이전부터 여러 연구자들의 연구대상이 되어 왔으며, 문제의 중요성에 비추어 볼 때 조금이라도 더 시계열데이터 예측의 정확도를 높이려는 노력은 끊임없이 지속될 것이다.

전통적인 시계열데이터 예측방법은 선형예측방법과 비선형예측방법으로 나눌 수 있다[10]. 선형예측방법은 통계학 분야에서 광범위하게 연구되어 왔으며 AR(Auto-Regressive)모델, ARMA(Auto-Regressive Moving Average)모델, ARIMA(Auto-Regressive Integrated Moving Average)모델 등이 제안되었다. ARIMA모델은 1976년에 Box-Jenkins에 의해 제안된 방법으로 현재 데이터를 과거 데이터와의 상관관계 및 과거 예측오차와의 관계로 표현한다[2].

비선형예측방법은 기계학습에 기반을 둔 방법으로 SVM(Support Vector Machine), 인공신경망 등이 있다. 인공신경망에서는 전방향신경망, Elman 신경망, NARX 신경망 등이 시계열데이터 예측에 사용되었다. 또한 선형모델인 ARIMA와 비선형모델인 전방향신경망을 결합하여 예측 정확도를 높이려는 혼합 모델에 대한 연구가 진행되어 왔다[1,9,15].

최근 몇 년 사이에 딥러닝이 괄목한 만한 성과를 거두고 있다. 새로운 신경망 구

조와 신경망 학습 방법의 고도화, GPU의 병렬 연산능력을 활용한 빅데이터 학습을 통해 음성/영상인식, 자연어 처리, 문서번역, 자율주행 등 다방면에서 큰 진전을 이루고 있다. 본 논문에서는 LSTM(Long Short Term Memory) 순환신경망을 이용하여 시계열데이터의 예측 정확도를 높이는 방법을 시도한다.

순환신경망은 데이터의 선후관계를 학습할 수 있는 능력을 갖고 있다. 시계열데이터는 시간 순서에 따라 발생하는 데이터이므로 선후관계의 문맥을 이해하는 순환신경망은 본질적으로 시계열데이터의 학습에 강점을 갖는다고 생각할 수 있다. 순환신경망 구조와 함께, 가중치 초기화, 과다학습 방지를 위한 Dropout 등의 최근 딥러닝 이론을 시계열데이터 예측 문제에 적용함으로써 기존의 방법들보다 더 높은 예측 정확도를 얻을 수 있다.

1.2 연구의 범위 및 방법

시계열데이터 예측은 데이터 자체의 과거 패턴을 분석하여 미래 값을 예측하는 단변수 예측모델과, 다수의 설명변수를 분석하여 미래 값을 예측하는 복변수 예측모델로 나눌 수 있다. 본 연구에서는 설명변수를 어떻게 잘 선택해서 예측할 것인가가 아닌, 원 데이터만을 이용하여 예측하는 문제를 연구 대상으로 한다.

본 논문에서는 LSTM/GRU 셀을 갖는 순환신경망에 Dropout 과다학습 방지 기법 및 Xavier 가중치 초기화 기법을 적용한 제안모델을 구축하여 기존 방법들(ARIMA, 전방향신경망, 혼합모델)과 비교 평가를 실시한다. 실험은 Torch7 딥러닝 프레임워크와 R 프로그램을 이용하여 세 가지의 공개 데이터에 대해 소프트웨어 시뮬레이션을 시행한다.

제2장 배경 이론 및 관련 연구

2.1 통계기반 예측 방법

통계에 기반한 시계열데이터 예측방법은 단순예측법, 회귀예측법, 지수평활법, ARIMA 방법 등이 있다[7].

2.1.1 단순 예측법

가. 평균법

평균법은 가장 단순하지만, 이해하기 쉽고 효율적이어서 사람들이 많이 사용하는 방법이다. 식 2.1에서 보는 바와 같이 미래 예측값은 과거 데이터의 평균과 같다. 여기서 \hat{y}_{t+1} 은 예측값을 나타내고 t 는 시점을 나타낸다.

$$\hat{y}_{t+1} = (y_1 + y_2 + \dots + y_t) / t \quad (2.1)$$

나. Naive 방법

Naive 방법은 단순히 가장 최근의 관측값을 예측값으로 사용한다. 주가지수, 환율과 같이 어느 방향으로 움직일지 전혀 알 수 없다고 생각될 때 사용할 수 있다.

$$\hat{y}_{t+1} = y_t \quad (y_t = \text{마지막 관측데이터}) \quad (2.2)$$

2.1.2 회귀 예측법

회귀 예측법은 데이터의 중심을 지나며 오차의 합이 가장 작은 회귀식을 구하여 예측하는 방법이다. 설명 변수의 개수에 따라 단순회귀와 복수회귀로 나누어지며, 회귀식의 차수에 따라 선형모델과 비선형모델로 나뉜다. 단순 선형모델에서 회귀식은 식 2.3과 같이 표현되는데, 여기서 x 는 설명변수, β 는 설명변수의 계수이며 ε 는 오차를 의미한다.

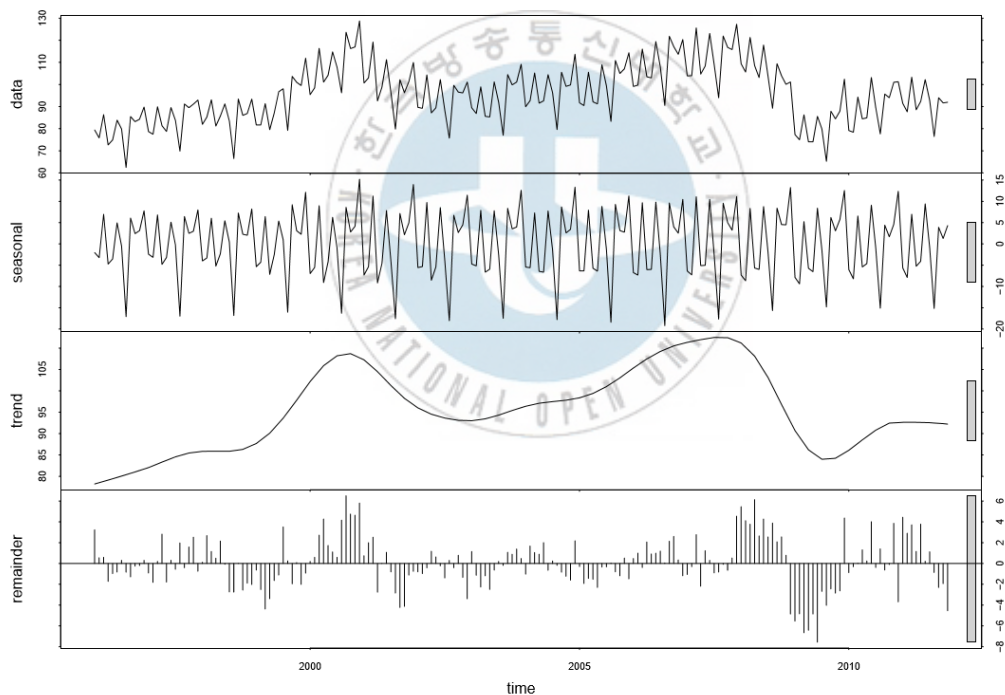
$$y = \beta_0 + \beta_1 x + \varepsilon \quad (2.3)$$

2.1.3 시계열데이터 분해법

시계열데이터는 장기간에 걸친 증감경향을 나타내는 추세요소, 일정 기간 간격으로 비슷한 패턴을 반복하는 계절요소를 내포하고 있다. <그림 2-1>과 같이 시계열데이터를 추세요소, 계절요소 및 나머지요소로 분리해 내는 것을 시계열데이터 분해라고 한다. 이렇게 데이터를 분리하는 것은 데이터에 대한 이해를 증진시키고 예측의 정확성을 높이는데 도움을 준다.

이를 식으로 표현하면 식 2.4와 같으며, 여기서 S_t 는 계절요소, T_t 는 추세요소, E_t 는 나머지 요소를 각각 나타낸다.

$$y = S_t + T_t + E_t \quad (2.4)$$



<그림 2-1> 시계열데이터 분해 예

2.1.4 지수평활법

지수평활법(exponential smoothing)은 최근 데이터들의 가중치의 합으로 예측값을 구한다. 최근 관측값에 더 많은 가중치를 부여하며 이전 데이터로 갈수록 가중치가 지수적으로 줄기 때문에 지수평활법이라고 부른다. 단순지수평활법은 데이터가 추세나 계절 패턴을 보이지 않을 때 사용 가능하다.

이 모델은 식 2.5와 같으며, 여기서 \hat{y}_{t+1} 은 예측값을 나타내며, α 는 평활계수로 0에서 1 사이의 값을 가지고, α 값이 커질수록 최근 데이터에 더 많은 가중치가 부여된다.

$$\hat{y}_{t+1} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \dots \quad (2.5)$$

Holt-Winter 모델은 추세와 계절 패턴을 가진 데이터에 적용 가능한 지수평활법의 일종이며, ARIMA와 함께 시계열데이터 예측에 자주 사용된다[7].

2.1.5 정상성과 차분

정상성(stationarity)이란 관측 시점에 따라 데이터의 속성 즉, 평균과 분산이 변하지 않는 성질을 말한다. 데이터가 계절에 따라 패턴이 바뀐다든지, 증가하거나 감소하는 경향을 보이는 경우는 정상성을 갖지 않으며, 백색잡음과 같은 데이터는 정상성을 갖는다.

ARIMA를 포함하여 통계적 시계열데이터 예측 모델이 유효하기 위해서는 데이터가 정상성을 만족해야 한다는 요구조건이 있다. 따라서 데이터가 정상성을 갖지 않는 경우, ARIMA와 같은 통계모델을 적용하기 위해서는 정상성을 갖도록 변환을 할 필요가 있는데 이 때 차분(differencing)을 사용한다. 차분 데이터 y' 는 식 2.6과 같이 연속되는 데이터 간의 차이를 구하여 얻어진다.

$$y'_t = y_t - y_{t-1} \quad (2.6)$$

만일 한 번의 차분으로 정상성을 갖는 데이터를 얻을 수 없다면, 차분을 두 번 이

상 할 수도 있다. 식 2.7은 그 예이다.

$$y''_t = y'_t - y'_{t-1} \quad (2.7)$$

2.1.6 ARIMA

ARIMA 모델은 현실 세계에서 자주 사용되는 시계열데이터 예측 방법이다. 활용도가 높은 만큼 이를 활용한 모델구축에 관한 논문도 다수 나오고 있으며, 국내에서는 2014년 ARIMA 모형을 활용하여 저가항공의 수요를 예측하는 연구가 있었다[16].

ARIMA는 AR 모델과 MA 모델, 그리고 차분을 통합한 모델로 ARIMA(p, d, q)로 나타낸다. AR 모델은 데이터를 과거 데이터와의 자기상관관계로 표현하며, p차의 AR 모델은 AR(p) 라고 쓰고, 수식으로 표현하면 식 2.8과 같다[4]. 여기서 c 는 상수이며, e_t 는 백색잡음이다. ϕ 값은 과거 데이터와의 자기상관도로서 시계열데이터의 패턴에 영향을 주며, e_t 는 진폭에 영향을 준다.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + e_t \quad (2.8)$$

MA 모델은 데이터를 과거 데이터에 대한 예측 오차를 사용하여 표현하는데, MA(q) 모델은 과거 q개의 데이터에 대한 예측오차를 가중평균한 것과 같고, 식 2.9처럼 나타낼 수 있다[4]. 여기서 c 는 상수이며, e_t 는 백색잡음이다. θ 값은 예측오차의 가중치로서 시계열데이터의 패턴에 영향을 주며, e_t 는 진폭에 영향을 준다.

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q} \quad (2.9)$$

ARIMA(p, d, q) 모델의 가운데 파라미터 d는 정상성을 얻기 위해 필요한 차분의 차수를 나타낸다. 수식 표현의 편의를 위해 “backshift 표기법”을 사용하는데 $By_t = y_{t-1}$, $B^2 y_t = y_{t-2}$ 이다. 따라서 d차의 차분은 식 2.10처럼 나타낼 수 있다.

$$(1 - B)^d y_t \quad (2.10)$$

위의 AR(p), d차 차분, MA(q)를 합하여 ARIMA(p, d, q)로 표현할 수 있다.

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 e_{t-1} + \dots + \theta_q e_{t-q} + e_t \quad (2.11)$$

우변의 AR 부분을 좌변으로 넘기고 backshift 표기법을 사용하여 식을 나타내면, 최종적으로 식 2.12를 얻을 수 있다.

$$(1 - \phi_1 B - \dots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \dots + \theta_q B^q) e_t \quad (2.12)$$

여기서 $(1 - \phi_1 B - \dots - \phi_p B^p)$ 는 AR(p)이고, $(1 - B)^d y_t$ 는 d차 차분이다. $c + (1 + \theta_1 B + \dots + \theta_q B^q) e_t$ 는 MA(q)를 나타낸다.

Box-Jenkins Method

Box-Jenkins는 다음의 단계를 거쳐 ARIMA 모형을 구축할 것을 제시하였다.

- ① **데이터 파악:** 데이터를 그래프로 그려보고 특성을 파악한다.
- ② **정상성 확인:** 데이터가 정상성을 갖는지를 확인한다. 정상성을 갖지 않는다면 정상성을 가질 때 까지 차분을 취한다.
- ③ **모형식별:** ACF/PACF 그래프를 그려보고 어떤 AR(p), MA(q) 모형이 적절한지를 판단한다.
- ④ **모수추정:** 적절하다고 생각되는 모형을 선택하여 AIC 값을 확인한다.
- ⑤ **모형진단:** 모형진단. 잔차에 대해 ACF 그래프를 그려본다. 잔차가 백색잡음의 성격을 갖는다면 유효한 모델로 볼 수 있다.
- ⑥ **순환반복:** 모형식별부터 모형진단까지의 과정을 반복하여 최적의 모델을 결정한다.

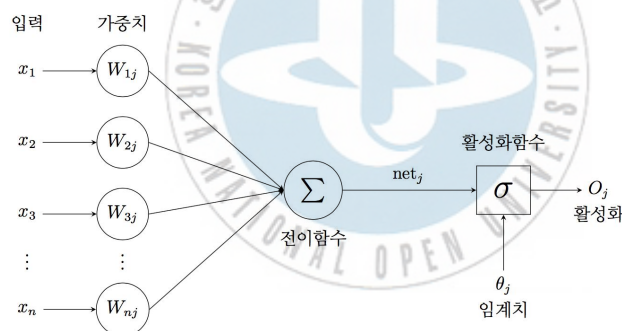
[16]은 Box-Jenkins 제안 순서에 따라 모델을 구축하는 실례를 잘 보여주고 있다.

2.2 인공신경망

인공신경망 모델은 앞에서 살펴 본 통계기반 모델과 함께 대표적인 시계열데이터 예측 모델이다. 통계기반 예측기법은 데이터가 정상성을 가지고 있어야 하고, 잔차(예측값과 실제값의 차이)는 백색잡음이어야 한다는 등의 요구조건을 가지고 있다. 이는 통계에 대한 전문적인 지식을 가지지 않은 사람이 통계기반 모델을 사용하는데 장애가 된다. 이에 반해 인공신경망은 데이터에 대한 어떠한 가정도 없이 데이터 자체에만 의존하여 학습을 함으로써 결과를 얻을 수 있다는 장점이 있다.

2.2.1 다층 퍼셉트론

지금까지 시계열 예측 문제에 이용되어 온 전통적인 인공신경망은 다층 퍼셉트론(multi layer perceptron)이다. 이 모델은 입력층, 은닉층, 출력층의 세 층으로 구성된 형태를 갖고 있으며, 모든 뉴런들이 완전히 서로 연결되어 있는 완전연결신경망(Fully Connected Neural Networks)이면서 네트워크 내에 순환이 없이 한 쪽 방향으로만 진행되는 전방향신경망(Feed Forward Neural Networks)이다.



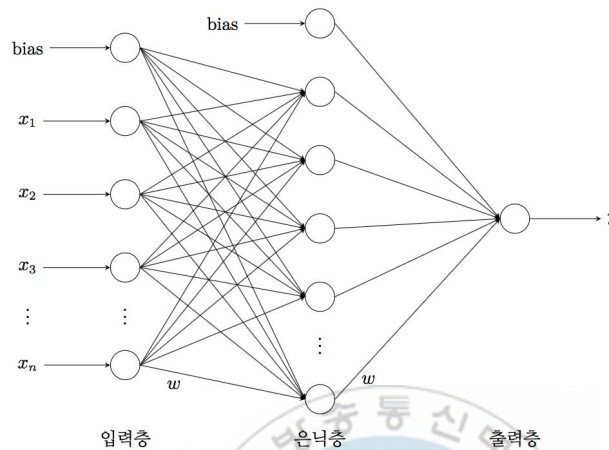
<그림 2-2> 뉴런의 작동방식

<그림 2-2>와 같이, 각 뉴런은 모든 입력 값에 가중치를 곱하고 이들을 합한 값에 시그모이드와 같은 활성화 함수를 적용하여 출력을 낸다. 활성화 함수는 뇌신경의 신호전달 방식과 같이 입력값의 합이 일정 임계치 이상이 되었을 때 신호를 출력하

고, 임계치 미만인 경우는 출력을 내지 않는 역할을 한다.

J층의 뉴런의 출력은 식 2.13과 같으며, 여기서 g 는 시그모이드 등의 활성화 함수이고, w 는 가중치, 그리고 b 는 바이어스를 나타낸다.

$$y_j = g\left(\sum_{i=1}^n w_{ij} \times x_i + b\right) \quad (2.13)$$



<그림 2-3> 전방향신경망의 구조

<그림 2-3>과 같이 모든 뉴런은 다음 층의 뉴런과 가중치(weights)를 가지고 연결되어 있어, 입력이 주어지면 입력에 가중치들이 곱해지고 활성화 함수를 통해 다음 층의 뉴런으로 전달되는 과정을 거쳐 출력층에서 최종 출력을 낸다.

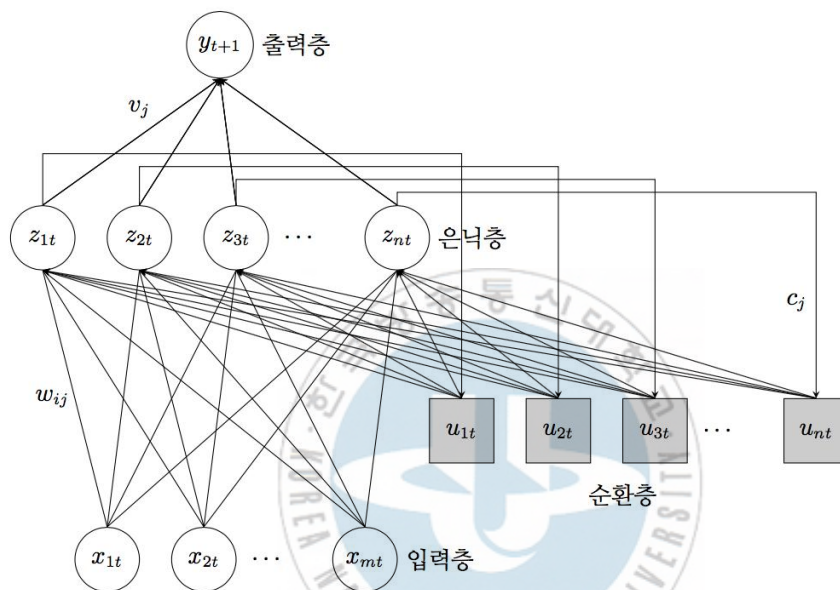
최종 출력값과 목표값의 차이를 손실함수(loss function)로 정의하고, 네트워크를 역방향으로 진행하면서 이 손실함수를 입력에 대해 편미분을 하여 손실값이 작아지도록 가중치를 조정하여 학습을 하는데, 이를 오류역전파(backpropagation) 알고리즘이라 부른다.

이주현 등[18]은 2013년에 다층 퍼셉트론 인공신경망 모형을 이용하여 가뭄을 예측하는 연구를 진행하였으며, 이재현[17]은 2014년에 신경망을 이용하여 상품수요 예측 모형을 구축하는 연구를 진행하였다.

2.2.2 Elman 신경망

Jeff Elman은 입력데이터를 처리한 결과를 다시 입력으로 순환시키는 단순한 형태의 순환신경망을 고안하였다. 입력층, 은닉층, 출력층의 세 개 층으로 이루어지기는 하나, 은닉층의 출력이 문맥정보를 기억하는 context units(U)를 거쳐 다시 은닉층의 입력으로 순환된다. (<그림 2-4>)

이 네트워크는 상태정보를 관리할 수 있으므로 단순한 다층 퍼셉트론 보다 시계열 데이터의 처리에 더 강점을 갖는다. Wang et al.은 2016년 Elman 순환신경망을 활용하여 주가지수 예측 연구[12]와 원유가격 예측 연구[13]를 진행하였다.



<그림 2-4> Elman 신경망의 구조 <출처: Wang 외(2016)[12]>

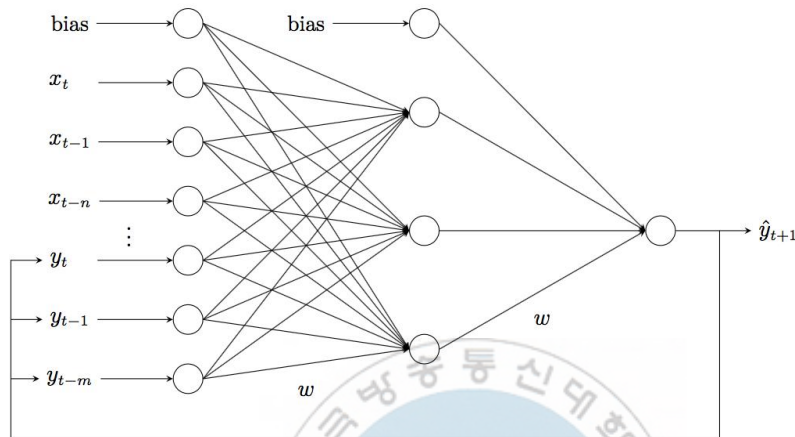
2.2.3 NARX 신경망

NARX(Nonlinear AutoRegressive eXogenous model)는 시계열데이터 모델링에서 과거의 시계열데이터와 함께 시계열 데이터 외의 외부설명변수를 입력으로 주는 것을 의미한다. 시계열데이터를 y , 외부설명변수를 u , 예측오차를 ε 이라고 할 때, NARX 모델은 식 2.14와 같이 표현된다.

$$y_t = F(y_{t-1}, y_{t-2}, y_{t-3}, \dots, u_t, u_{t-1}, u_{t-2}, u_{t-3}, \dots) + \varepsilon_t \quad (2.14)$$

NARX 신경망은 위의 함수 F를 신경망으로 구현한 것인데, 네트워크의 최종 출력 값을 외부변수로 사용하여 다시 입력으로 순환시키는 형태를 갖는다(그림 2-5).

Chaudhuri et al.[3]은 NARX 신경망을 이용하여 환율을 예측하는 연구를 진행하였으며, Jalal et al.[8]은 Elman과 NARX 신경망을 결합하여 콜센터에 접수되는 전화 건수를 예측하는 연구를 진행하였다.



<그림 2-5> NARX 신경망의 구조 <출처: www.researchgate.net>

2.3 혼합 모델

ARIMA는 선형관계를 보다 정확히 표현해 낼 수 있고, 인공신경망은 비선형관계 학습에 더 강점이 있다는 점에 착안하여 이 둘을 혼합하여 시계열데이터 예측 정확도를 높이려는 연구가 시도되었다[1,9,15]. Deep & wide 신경망이 일반화된 지금 상황에서 보면 인공신경망 만으로 어떤 선형관계도 근사할 수 있다는 점에서 이런 혼합방식의 접근법이 큰 의미가 없을 것으로 생각할 수도 있겠으나, 3개 층 이상의 신경망의 학습이 쉽지 않았던 이전의 상황을 고려하면 그러한 시도를 한 이유를 이해할 수 있다.

2.3.1 Zhang 혼합 모델

G. Peter Zhang[15]은 ARIMA와 인공신경망을 결합한 모델을 제안하였다. 그는 시계열데이터를 식 2.15와 같이 선형부분(L_t)과 비선형부분(N_t)의 합으로 보고, ARIMA를 이용하여 선형부분을 예측(\hat{L}_t)하고, 전방향신경망을 이용하여 비선형부분을 예측(\hat{N}_t)한 후, 둘을 합하여 최종 예측값(\hat{Y}_t)을 결정함으로써 ARIMA나 인공신경망의 독립적 모델보다 더 정확한 결과를 얻을 수 있다고 주장하였다

$$Y_t = L_t + N_t \quad (2.15)$$

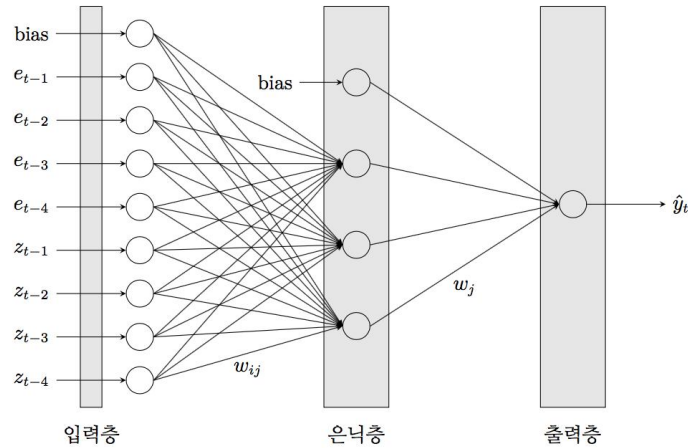
이 모델은 다음과 같은 순서를 따른다.

- ① 먼저 ARIMA 모델을 세워 원 데이터에 대한 예측값을 구한다. (\hat{L}_t)
- ② 원 데이터에서 ①의 예측값을 뺀다.
- ③ ②의 값에 대해 전방향신경망 모델을 수립하여 예측값을 구한다. (\hat{N}_t)
- ④ ①과 ③을 더하여 최종 예측값을 얻는다. ($\hat{Y}_t = \hat{L}_t + \hat{N}_t$)

2.3.2 Khashei & Bijari 혼합 모델

Khashei & Bijari[9]는 2010년에 Zhang 혼합 모델과 유사한 모델을 발표하였다. 그들도 시계열데이터가 선형과 비선형부분의 합계이며, ARIMA는 선형관계를 더 잘 표현하고, 인공신경망은 비선형관계를 더 잘 표현할 수 있다고 가정하고, ARIMA와 전방향신경망을 결합한 새로운 방식으로 Zhang 혼합 모델보다 더 좋은 결과를 얻을 수 있음을 보였다. Zhang 혼합 모델은 ARIMA로 선형부분 예측값을, 인공신경망으로 비선형부분 예측값을 각각 구한 후 둘을 합하여 최종 예측값을 출력하였으나, 이 모델은 ARIMA로 데이터를 전처리하여 인공신경망에서 예측값을 출력하는 방식이다. 먼저 ARIMA 모델을 세워 예측을 하여 예측오차를 구한 후, 이 예측오차를 원래의 시계열데이터와 함께 인공신경망에 입력으로 주어 최종 예측값은 인공신경망에서만 출력한다.

<그림 2-6>은 Khasei & Bijari 혼합 모델의 인공신경망 구조이다. $e_{t-1}, e_{t-2}, e_{t-3}, e_{t-4}$ 는 ARIMA 모델을 통해 예측하여 얻어진 예측이며, $z_{t-1}, z_{t-2}, z_{t-3}, z_{t-4}$ 는 원 데이터이다.



<그림 2-6> Khashei & Bijari 혼합 모델의 신경망 구조 <출처: Khashei 외(2010)[9]>

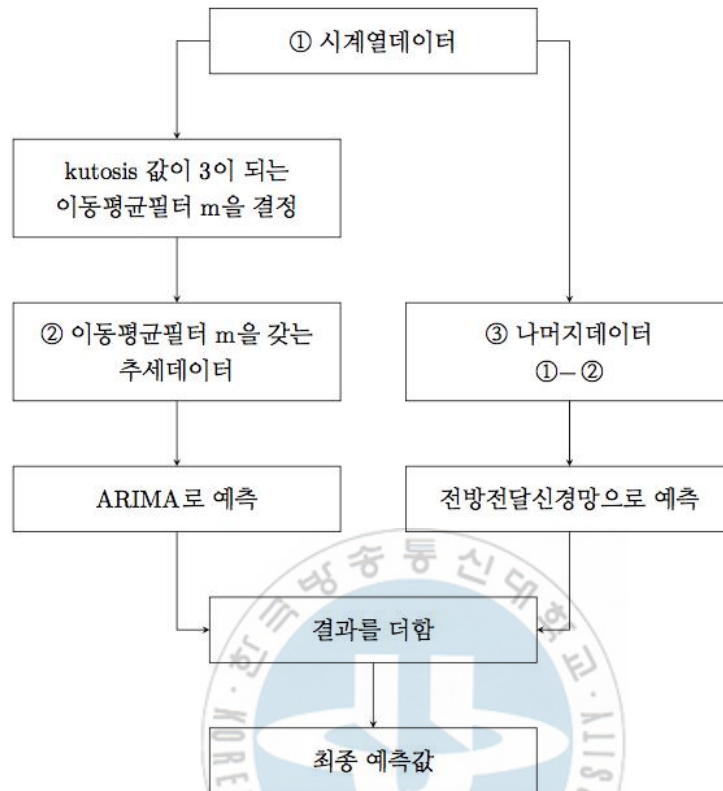
2.3.3 Babu & Reddy 혼합 모델

Babbu & Reddy[1]는 2014년 ARIMA와 전방향신경망을 혼합한 모델을 발표하였다. 그들은 변동성이 낮은 데이터는 ARIMA가 강점을, 변동성이 높은 데이터는 인공신경망이 강점을 갖는다고 보고, 데이터를 추세요소와 나머지로 분리하여 각각 ARIMA와 인공신경망을 사용하여 예측한 후, 둘을 결합하는 방법을 사용하였다. 이 모델의 예측방법은 다음과 같은 순서를 따른다. (<그림 2-7>)

- ① 원 시계열데이터를 추세요소와 나머지로 분리한다. 추세요소를 구하기 위해 사용하는 이동평균 필터값은 첨도(kurtosis)가 3이 되도록 정한다. 첨도가 3이 될 때 가우시안 분포를 따르게 되며, ARIMA 모델이 가우시안 분포를 갖는 데이터를 잘 모델링한다고 알려져 있기 때문이다[1].
- ② 추세요소에 대해 ARIMA 모델을 세워 예측값을 구한다.
- ③ 나머지에 대해 전방향신경망 모델을 세워 예측값을 구한다.

④ ②와 ③을 더하여 최종 예측값을 얻는다.

그들은 이 모델이 앞서 설명한 Zhang 혼합 모델이나 Khashei & Bijari 혼합 모델보다 더 높은 예측 정확도를 보인다고 주장하였다.



<그림 2-7> Babu & Reddy 혼합 모델의 구조 <출처: Khashei 외(2010)[9]>

제3장 제안 모델

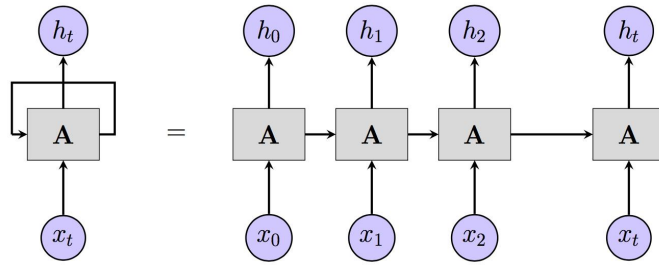
앞에서 본 전방향신경망은 데이터의 순서를 상관하지 않고 학습을 하므로 선후관계가 중요한 경우 전방향신경망의 학습은 효율적이라고 할 수 없다. 시계열데이터는 시간 간격을 두고 관측된 일련의 정보이며, 시계열데이터의 예측은 앞에서 일어난 사건이 뒤에 일어난 사건에 어떻게 영향을 미치는지를 분석하는 행위이다.

순환신경망은 과거의 문맥을 기억하여 학습을 진행하므로 데이터 선후관계가 중요한 시계열데이터에 적합한 모델이다. 앞에서 살펴본 Elman 신경망이나 NARX 신경망은 과거 데이터를 다시 입력으로 피드백 한다는 점에서 순환신경망의 일종이나, 단지 최근의 문맥만을 학습할 뿐 멀리 떨어져 있는 데이터 간의 관계는 제대로 학습할 수 없었다. 딥러닝 분야의 발전에 따라 이제는 긴 시퀀스의 데이터를 순환신경망으로 학습할 수 있게 되었다. 본 논문에서는 LSTM 순환신경망을 이용하여 시계열데이터를 예측하는 모델을 제시한다.

3.1 순환신경망

<그림 3-1>은 순환신경망의 구조이다. 순환신경망에서는 입력데이터와 이전 단계의 처리결과를 함께 입력받아 처리하는 과정을 반복한다. 식 3.1에서 x_t 는 입력데이터, h_{t-1} 은 이전 단계의 은닉층 출력, W 는 가중치, b 는 바이어스, H 는 활성화 함수이다. t 단계의 은닉층 출력 h_t 는 입력데이터에 가중치를 곱하고 이전 단계의 은닉층 출력에 가중치를 곱한 것을 더하고 여기에 바이어스를 더한 후 활성화 함수를 적용하여 값을 낸다. 따라서 과거의 데이터가 이후의 데이터에 어떤 영향을 주는지를 학습할 수 있고, 이러한 특징으로 인해 순환신경망은 데이터의 입력이 시간 순서로 주어지는 음성인식, 영상인식, 문장번역, 자연어처리 등에 사용되고 있다.

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (3.1)$$



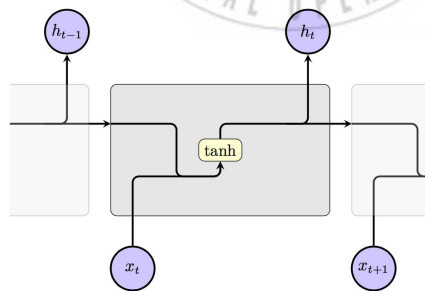
〈그림 3-1〉 순환신경망의 구조 <출처: colah.github.io>

순환신경망의 학습은 오류역전파 알고리즘을 순환신경망에 맞도록 변형한 Back-propagation Through Time 알고리즘을 사용하여 학습을 진행한다. 이는 기본적으로 오류역전파 알고리즘과 같지만, 순환층의 가중치가 서로 공유되어 학습되는 동안 가중치가 누적되어 조정된다는 차이가 있다[14].

3.1.1 표준 순환신경망

표준 순환신경망은 입력을 비선형 관계로 변환하는 활성화함수로 시그모이드, 하이퍼탄젠트 등을 사용하여 출력을 내며, 이것이 다음단계의 입력으로 전해진다(〈그림 3-2〉). 이 출력값은 $-1 \sim 1$ 범위의 작은 값을 갖기 때문에(식 3.2), 이 과정이 반복되어 곱해지면 숫자가 점점 작아져서 시퀀스가 길어지면 어느 단계부터는 학습이 더 이상 이루어지지 않게 되는데 이를 “그라디언트 소실” 문제라고 부른다.

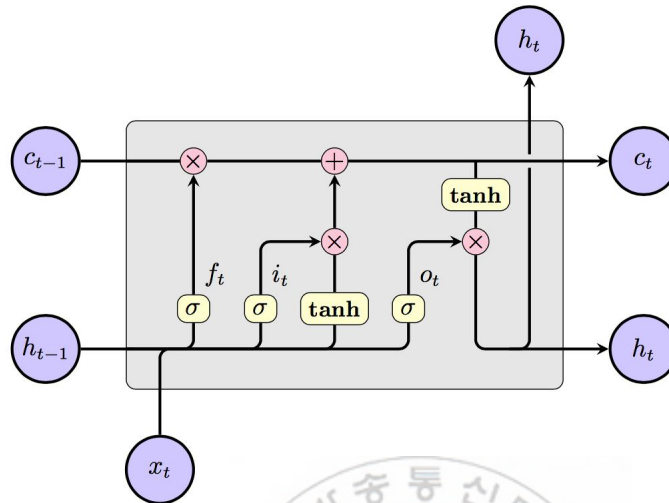
$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}, \quad \text{Tanh}(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (3.2)$$



〈그림 3-2〉 표준 순환신경망의 셀 구조 <출처: colah.github.io>

3.2 Long Short Term Memory

LSTM 셀을 사용한 순환신경망은 아무리 긴 시퀀스의 데이터라도 그라디언트 소실 문제없이 처리한다. 최근 음성인식, 자연어처리, 이미지에 자동 주석달기 등의 분야에서 사용되고 있는 순환신경망은 거의 LSTM을 이용하고 있다.



〈그림 3-3〉 LSTM 순환신경망의 셀 구조 <출처: colah.github.io>

〈그림 3-3〉는 LSTM 셀을 가진 순환신경망을 보여주고 있다. LSTM 셀은 셀 상태를 유지하면서, 입력게이트, 망각게이트, 출력게이트를 이용하여 출력값을 조절한다. 입력게이트는 입력값을 얼마나 받아들일지를 결정하고(식 3.3), 망각게이트는 이전의 셀 상태를 얼마나 잊어버릴지를 결정하며(식 3.4), 출력게이트는 얼마나 출력할지를 결정한다(식 3.6).

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3.3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (3.4)$$

$$c_t = f_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3.5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (3.6)$$

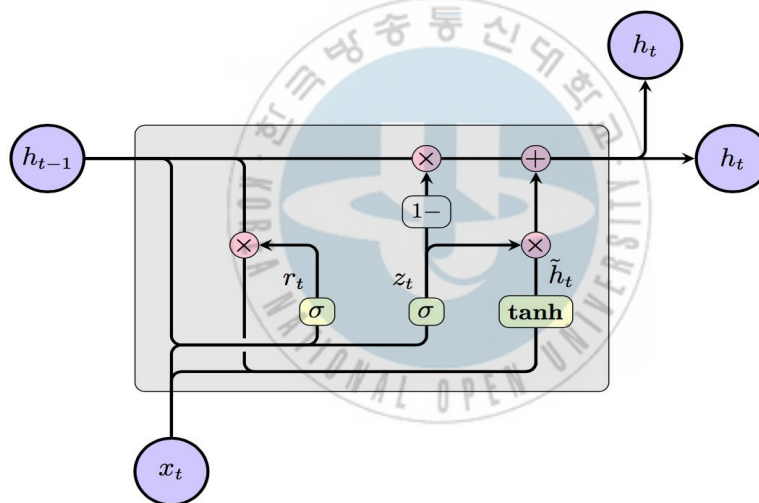
$$h_t = o_t \tanh(c_t) \quad (3.7)$$

여기서 c_t 는 셀 상태이며, h_t 는 셀의 출력이다. i_t , f_t , o_t 는 각각 입력게이트, 망각게이트, 출력게이트를 나타내며, σ 와 \tanh 는 활성화 함수로 각각 시그모이드와 하이퍼탄젠트를 나타낸다. “기존의 셀 상태에 망각게이트의 출력값을 곱하여” 셀 상태값의 일정량을 잊어버리고, “입력값과 이전 단계의 출력값을 처리한 결과에 입력게이트의 출력값을 곱하여” 입력의 일정량을 받아들임으로써 새로운 셀 상태를 만든다(식 3.5). 이 셀 상태에 출력게이트의 출력값을 곱하여 출력을 낸다(식 3.7).

이와 같이 셀 상태값을 얼마나 잊어버리고, 새로운 입력값을 얼마나 받아들일지 결정하여 더하는 구조이므로 이 과정이 반복되어도 학습이 불가능해지는 일이 발생하지 않는다.

3.2.1 Gated Recurrent Unit

GRU는 2014년 발표된, LSTM의 구조를 보다 단순하게 처리한 LSTM 변형모델의 하나이다[4].



<그림 3-4> GRU 셀의 구조 <출처: colah.github.io>

GRU는 LSTM의 망각게이트(f_t)와 입력게이트(i_t)를 갱신게이트(z_t)로 통합하고, 셀 상태(c_t)와 은닉 상태(h_t)를 h_t 하나로 통합하였다. LSTM보다 단순한 구조로 가중치

수가 작으므로 학습이 더 빠르지만, LSTM과 거의 동일한 성능을 보인다[3]. GRU 셀은 식 3.8~ 3.11과 같이 동작한다.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (3.8)$$

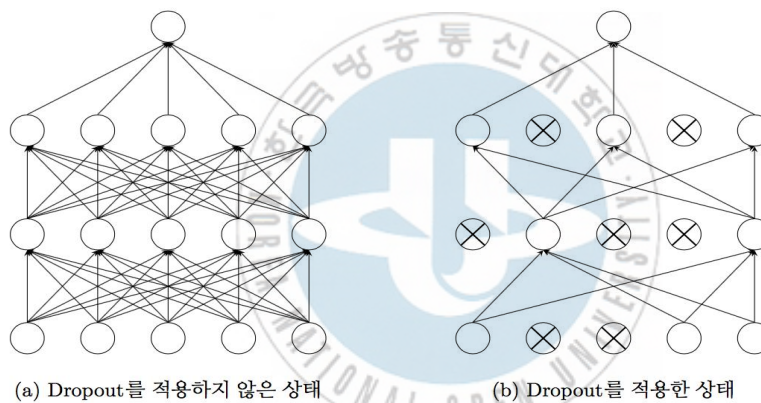
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (3.9)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \quad (3.10)$$

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \quad (3.11)$$

3.3 Dropout

많은 가중치를 갖는 큰 규모의 인공신경망은 강력한 학습능력을 갖지만, 소규모 인공신경망 보다 과다학습 문제가 더 크게 대두된다. G. E. Hinton et al.[6]은 과다학습에 효율적으로 대처하기 위해 Dropout 라는 새로운 방법을 제시하였다.



<그림 3-5> Dropout 구조 <출처: Srivastava (2014)[11]>

Dropout는 Dropout율을 지정하여, 학습 시에 <그림 3-5>의 우측 그림처럼 Dropout을 만큼의 뉴런만을 랜덤 선택하여 학습에 참여하게 하고, 나머지 뉴런은 비활성화한다. 학습이 완전히 이루어진 후 실제 사용 시는 좌측 그림처럼 모든 뉴런들을 활성화하여 동작하게 하는 것이다. 본 논문의 제안모델에서는 이 Dropout를 사용한다.

3.4 Xavier 가중치 초기화

여러 개의 은닉층을 갖는 deep networks는 2006년이 되어서야 제대로 학습을 할 수 있게 되었는데, 이 학습을 가능케 한 요소 중의 하나가 파라미터의 초기화의 중요성을 인식하기 시작했다는 점이다. Xavier Glorot와 Yoshua Bengio[5]는 2010년에 그 동안 행해진 랜덤 초기화가 왜 학습에 나쁜 영향을 주는지를 밝히며 “Xavier 가중치 초기화” 방법을 제시하였다.

Xavier 가중치 초기화는 가중치를 평균이 0이고 분산이 식 3.12와 같은 분포를 갖도록 초기화 하는 것이다. 여기서 n_{in} 은 해당 층으로 들어오는 입력 개수이고 n_{out} 는 해당 층의 출력 개수이다. 이렇게 하면 가중치가 적절한 범위 내의 값을 골고루 갖게 되고 이는 랜덤 초기화를 했을 때 보다 수렴이 더 빠르게 이루어지도록 도움을 준다.

$$Var(W) = \frac{2}{n_{in} + n_{out}} \quad (3.12)$$

본 논문의 제안모델에서는 Xavier 가중치 초기화를 사용한다.

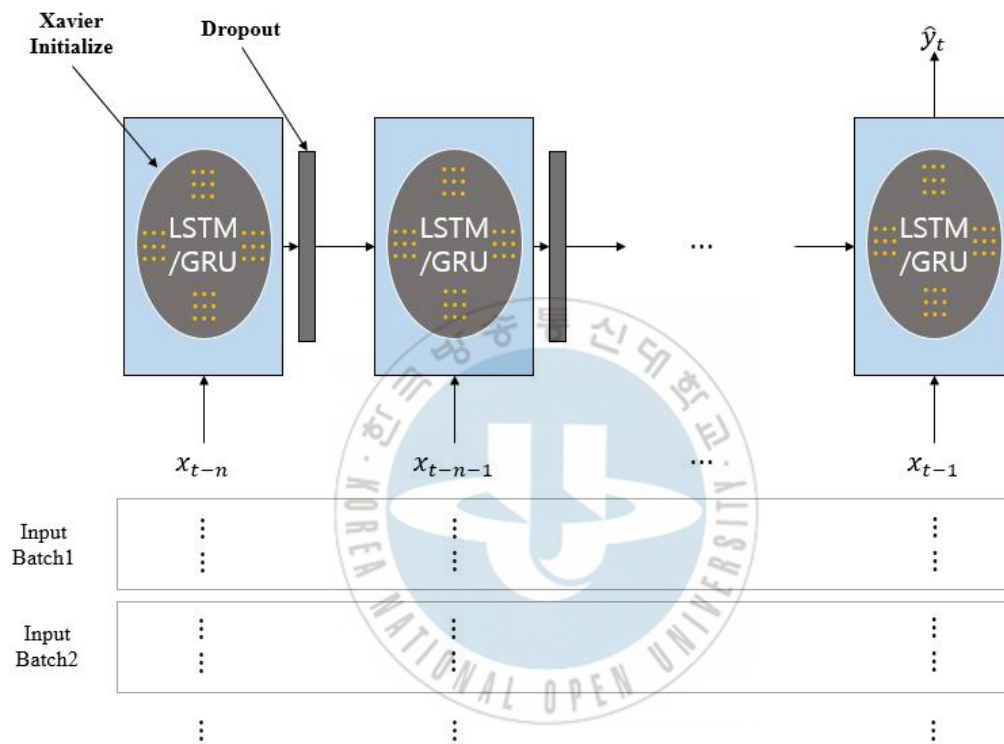


3.5 제안 모델

3.5.1 제안모델 구조

본 논문에서 제안하는 시계열데이터 예측 모델은 LSTM/GRU 셀을 갖는 순환신경망으로, Xavier 가중치 초기화와 Dropout 과다학습 방지 기법을 사용한다.

<그림 3-6>은 n time steps($x_{t-n} \dots x_{t-1}$)를 갖는 LSTM/GRU 순환신경망 모델을 예시한 것이다.



<그림 3-6> 제안 순환신경망 모델의 구조

3.5.2 모델의 학습과정

순환신경망의 학습은 일반적으로 다음과 같은 과정을 거쳐 진행한다.

① 입력데이터 준비:

원 데이터를 학습용, 확인용, 테스트용 데이터로 분할한다. 데이터 수가 많지 않다면 학습용과 테스트용으로 나누기도 한다. 그리고 원 데이터를 평균차감하여 데이터를 원점으로 이동시키고, 각 입력 데이터가 동일한 범위 내의 값을 갖도록 정규화 한다.

② 순환신경망 구조 결정:

- ㉠ 셀의 종류와 크기: 표준 셀/LSTM 셀/GRU 셀, 크기는 가중치의 개수이다.
- ㉡ 순환층의 반복 수
- ㉢ 가중치 초기화 방법: 랜덤 초기화, Xavier 초기화 등
- ㉣ 과다학습 방지 방법: L2 정규화, Dropout 등

③ 목적함수(loss function) 결정:

신경망의 학습은 목적함수의 값을 줄이는 방향으로 가중치들을 조정하는 일이다. 목적함수는 목표값과 출력값의 차이를 표현하며, 본 논문에서는 평균제곱오차 (Mean Squared Error)를 목적함수로 사용한다.

④ 전방향 학습단계:

입력데이터를 신경망에 흘려보내 출력값을 얻는다.

⑤ 역방향 학습단계:

목적함수를 편미분하여 그라디언트를 얻은 후, 네트워크를 역방향으로 거슬러 오면서 Backpropagation Through Time 알고리즘에 따라 가중치를 조정한다.

⑥ 테스트 단계:

확인데이터 혹은 테스트데이터를 이용하여 신경망의 정확도를 점검한다.

⑦ 학습종료 결정:

4~ 7 과정을 계속 반복하며 학습을 진행한다. 미리 정한 횟수만큼 반복을 완료하였거나, 학습이 더 이상 진행되지 않는 경우, 혹은 원하는 오차이내로 학습이 된 경우 학습을 중단한다.

제4장 실험 결과

4.1 실험 환경 및 비교 대상 모델

본 논문에서는 다음의 세 가지 공개 데이터에 대한 실험을 통해, 지금까지 제안된 여러 모델들과 본 논문에서 제안하는 LSTM/GRU 순환신경망 모델의 성능을 비교 평가한다.

- ① Lynx 데이터
- ② Sunspot 데이터
- ③ 환율 데이터 (USD vs KRW)

실험은 CPU Intel Core i5 1.3GHz, RAM 8GB를 갖춘 Macbook Air를 사용하여 진행하였다. 비교대상은 ARIMA 모델, 전방향신경망 모델, Khashei & Bijari 혼합 모델, Babu & Reddy 혼합 모델, 그리고 LSTM/GRU 순환신경망이다. 순환신경망 모델의 경우, 순환신경망 자체가 가지는 정확도와 함께, Khashei와 Babu 등이 주장했던 혼합 방법을 통해 정확도의 추가 향상이 가능한지를 확인하기 위해 두 가지 변형모델을 추가하여 실험을 진행하였다. 실험 비교대상 모델을 정리하면 다음과 같다.

- ① ARIMA 모델
- ② Khashei & Bijari 혼합 모델
- ③ Babu & Reddy 혼합 모델
- ④ 전방향신경망
- ⑤ 순환신경망 모델
 - ㉠ 제안모델 (A): LSTM/GRU 순환신경망만 사용한 모델
 - ㉡ 제안모델 (B): ARIMA 전처리 → LSTM/GRU 순환신경망
 - ARIMA로 얻어진 오차 데이터와 원 데이터를 함께 순환신경망에 입력하여, 순환신경망에서 최종 예측값을 출력한다. 이는 [9]의 전방향신경망을 순환신경망으로 대체한 모델이다.

㉔ 제안모델 (C): 데이터 분해 → LSTM/GRU 순환신경망

- 원 데이터를 추세부분과 나머지로 분리한다. 추세 데이터는 이동평균을 구하여 얻으며, 이동평균 필터는 첨도가 3에 가장 가까워지도록 하는 값을 선택한다. 나머지는 원 데이터에서 추세 데이터를 빼서 얻는다.
- 추세 데이터와 나머지 데이터를 순환신경망에 입력하여, 순환신경망에서 최종 예측값을 출력한다. 이는 [1]에서 ARIMA로 예측한 부분을 순환신경망으로 대체한 모델이다.

성능 측정은 MAE(Mean Absolute Error, 평균절대오차)를 사용하였다(식 4.1). 여기서 y 는 목표값, \hat{y} 는 예측값을 나타낸다.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.1)$$

4.2 딥러닝 프레임워크

현재 소스가 공개되어 많은 사람들이 이용하고 있는 딥러닝 프레임워크로는 다음과 같은 것들이 있다.

- ① Caffe: 버클리대학교
- ② Torch7: 뉴욕대학교, LuaGIT, Twitter, Facebook, DeepMind (<http://torch.ch>)
- ③ Theano: 그래프 기반
- ④ Tensorflow: 그래프 기반, Google (<https://www.tensorflow.org>)
- ⑤ CNTK: Microsoft

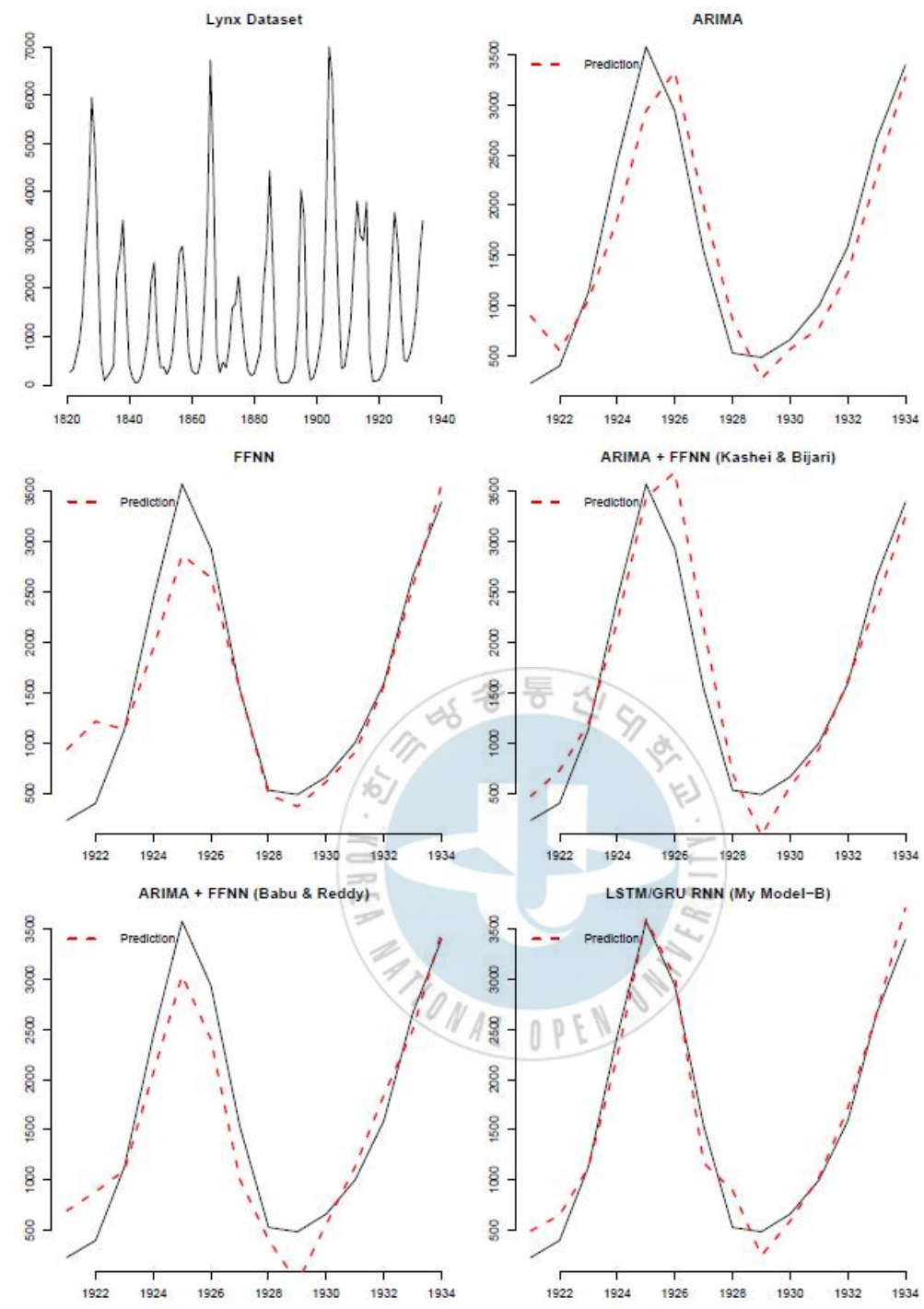
본 논문에서는 Torch7 프레임워크가 모듈형식의 구조를 채용하여 비교적 용이하게 커스터마이징을 할 수 있어 연구 목적에 가장 적합한 것으로 판단하여, Torch7을 이용하여 실험을 진행하였다.

4.3 Lynx Data 예측 실험

Lynx 데이터는 캐나다 지역에서 연간 포획된 Lynx의 수를 담고 있다. 시계열데이터 연구에 자주 등장하는 데이터로 비교대상 논문[9]에서도 사용되었다. Lynx 데이터는 1821년부터 1934년까지 114개의 데이터를 포함하고 있다(<그림 4-1> 좌상단). [9]과 동일한 방법으로 데이터를 나누어, 100개(1821-1920)를 학습에 사용하고, 14개(1921-1934)를 테스트에 사용하였다. 각 모델의 형태는 반복적 실험을 통해 예측 정확도가 가장 높은 것을 선정하였으며, 신경망의 학습율은 0.1부터 0.0001 사이의 다양한 값을 실험하여 가장 학습이 잘 이루어지는 값으로 결정하였다. 전방향신경망과 순환신경망 모두 Dropout 0.5, Xavier 가중치 초기화를 적용하여 학습하였다.

<표 4-1> Lynx data 실험에 사용된 모델의 형태

모델	형태	학습율	비고
ARIMA	ARIMA(12,0,0)	N/A	
전방향신경망	$FFNN^{12 \times 6 \times 1}$	0.001	
Khashei & Bijari 혼합 모델	ARIMA(12,0,0) → $FFNN^{8 \times 4 \times 1}$	0.001	ARIMA로 구한 최근 예측 오차 데이터 2개 + 최근 원 데이터 6개를 전방향신경망에 입력
Babu & Reddy 혼합 모델	ARIMA(3,0,0) + $FFNN^{12 \times 6 \times 1}$	0.004	원데이터를 추세(MA filter=7)와 나머지로 분리하고, 각각 ARIMA 와 전방향신경망으로 예측 후 결과를 합하여 출력
My Model (A)	순환신경망	0.01	Cell Type & Size: GRU, 256 최근 12개 데이터포인트를 순환신경망에 입력
My Model (B)	ARIMA(12,0,0) + My Model (A)	0.001	ARIMA 예측오차 12개와 원데이터 12개를 순환신경망에 입력
My Model (C)	Decomposition + My Model (A)	0.001	원데이터를 추세(MA filter=7)와 나머지로 분리하여 각각 12개씩 순환신경망에 입력



〈그림 4-1〉 Lynx data 각 모델별 실제값과 예측값 비교

〈표 4-2〉 Lynx data 제안모델과 기존 예측모델 간의 정확도 비교

모델	MAE	향상율	비고
ARIMA	326.1777	-	ARIMA(12,0,0)
전방향신경망	258.9010	20.63%	$FFNN^{12 \times 6 \times 1}$
Khashei & Bijari 혼합 모델	251.2469	22.97%	ARIMA(12,0,0) \rightarrow $FFNN^{8 \times 4 \times 1}$
Babu & Reddy 혼합 모델	307.3331	5.78%	ARIMA(3,0,0) + $FFNN^{12 \times 6 \times 1}$
My Model (A)	228.3116	30.00%	순환신경망만 사용
My Model (B)	167.5677	48.63%	ARIMA(12,0,0) \rightarrow 순환신경망
My Model (C)	215.1818	34.03%	데이터 분리 \rightarrow 순환신경망

ARIMA 방법은 기존 연구에서 가장 적합한 것으로 판명된 ARIMA(12,0,0) 모델을 사용하였다. 전방향신경망 모델은 입력층 뉴런 12개, 은닉층 뉴런 6개를 갖는 경우 최적으로 판명되었다. Khashei & Bijari 혼합 모델은 먼저 ARIMA(12,0,0) 모델을 수립하여 예측값을 구하고 예측값과 실측값과의 오차를 구한다. 그 후 이 오차 데이터의 최근 2개와 원 데이터의 최근 6개를 입력으로 주었다. Babu & Reddy 혼합 모델은 데이터를 추세데이터(이동평균 필터: 7)와 나머지로 분해한 후, 추세 데이터에는 ARIMA(3,0,0)을 사용하여 예측하고, 나머지데이터는 $FFNN^{12 \times 6 \times 1}$ 을 사용하여 예측한 후 둘을 합하여 최종 예측값을 구하였다.

제안모델은 셀 크기가 256개인 GRU 순환신경망으로 하고, 최근 데이터 12개를 사용하여 학습하도록 하였다. 실험에 사용한 모델의 자세한 형태는 〈표 4-1〉에 정리하였으며, 실험결과는 〈표 4-2〉에서 보는 바와 같다.

〈그림 4-1〉는 각 모델별 예측결과를 보이고 있는데, 검은실선이 원 데이터이며 붉은점선이 예측데이터이다. 제안모델 (A)의 예측 정확도는 ARIMA 대비 30% 향상되었다. 제안모델 (B)는 ARIMA 대비 49%의 정확도 향상을 보였는데, 이는 ARIMA로 데이터를 전처리하여 추가적인 정보를 순환신경망에 준 것이 학습에 도움이 되었기 때문으로 볼 수 있다.

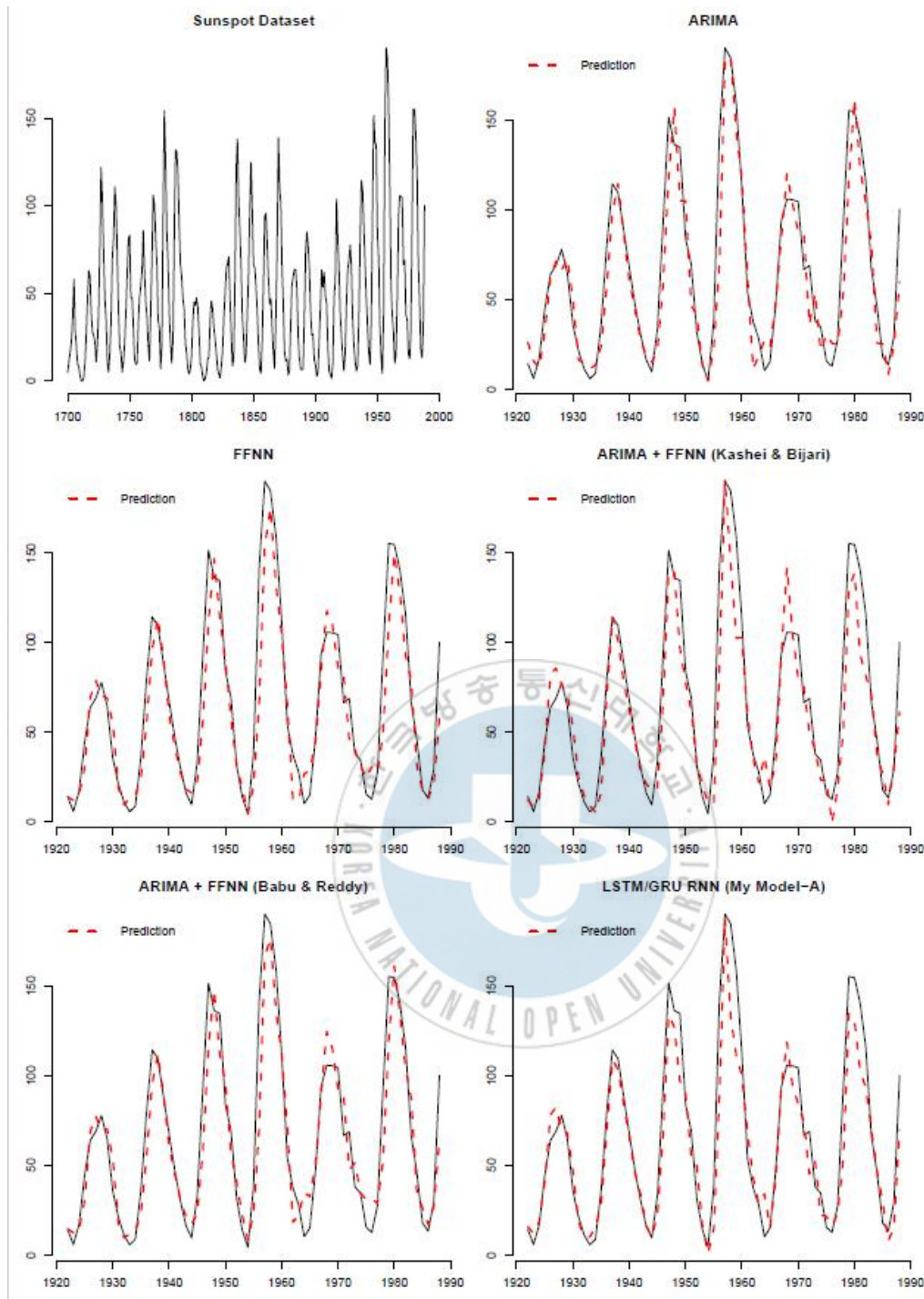
4.4 Sunspot data 예측 실험

Sunspot data는 태양의 흑점활동에 관한 관찰 데이터로, 1900년부터 1987년까지 총 288개의 연간 흑점 발생 수치를 담고 있다. 이 데이터는 Babu & Reddy의 논문[1]과 Khashei & Bijari의 논문[9]에서도 사용되었다.

공정한 비교를 위해 [9]과 동일한 방법으로 데이터를 나누어, 총 288개의 데이터 포인트 중 222개(1700년-1921년)는 데이터 학습에 사용하고, 66개(1922년-1987년)는 테스트에 사용하였다. 전방향신경망과 순환신경망 모두 Dropout 0.5, Xavier 가중치 초기화를 적용하여 학습하였다.

〈표 4-3〉 Sunspot data 실험에 사용된 모델의 형태

모델	형태	학습율	비고
ARIMA	ARIMA(9,0,0)	N/A	
전방향신경망	$FFNN^{9 \times 9 \times 1}$	0.001	
Khashei & Bijari 혼합 모델	ARIMA(9,0,0) → $FFNN^{18 \times 64 \times 1}$	0.01	ARIMA로 구한 최근 예측 오차 데이터 9개 + 최근 원데이터 9개를 전방향신경망에 입력
Babu & Reddy 혼합 모델	ARIMA(4,0,1) + $FFNN^{9 \times 9 \times 1}$	0.001	원데이터를 추세(MA filter=38) 와 나머지로 분리하고, 각각 ARIMA 와 전방향신경망으로 예측 후 결과를 합하여 출력
My Model (A)	순환신경망	0.01	Cell Type & Size: GRU, 256 최근 7개 데이터포인트를 순환신경망에 입력
My Model (B)	ARIMA(9,0,0) + My Model (A)	0.01	ARIMA 예측오차 7개와 원데이터 7개를 순환신경망에 입력
My Model (C)	Decomposition + My Model (A)	0.008	원데이터를 추세(MA filter=38) 와 나머지로 분리하여 각각 9개씩 순환신경망에 입력



〈그림 4-2〉 Sunspot data 각 모델별 실제값과 예측값 비교

〈표 4-4〉 Sunspot data 제안모델과 기존 예측모델 간의 정확도 비교

모델	MAE	향상율	비고
ARIMA	13.5082	-	ARIMA(9,0,0)
전방향신경망	13.3576	1.11%	$FFNN^{9 \times 9 \times 1}$
Khashei & Bijari 혼합 모델	13.2640	1.81%	ARIMA(9,0,0) \rightarrow $FFNN^{18 \times 64 \times 1}$
Babu & Reddy 혼합 모델	13.1163	2.90%	ARIMA(4,0,1) + $FFNN^{9 \times 9 \times 1}$
My Model (A)	12.4618	7.75%	순환신경망만 사용
My Model (B)	12.7572	5.56%	ARIMA(9,0,0) \rightarrow 순환신경망
My Model (C)	12.7145	5.88%	데이터 분리 \rightarrow 순환신경망

ARIMA 모델은 관련 논문에서 사용되었던 ARIMA(9,0,0) 모델을 사용하였다. 전방향신경망 모델은 여러 번의 실험을 거쳐 입력층 뉴런 9개, 은닉층 뉴런 9개를 갖는 $FFNN^{9 \times 9 \times 1}$ 모델을 선정하였다. Kashei & Bijari 혼합 모델은 $FFNN^{18 \times 64 \times 1}$ 을 사용하였는데, 먼저 ARIMA(9,0,0) 모델을 작성하여 예측하여 예측오차를 구한 후, 이 오차 데이터 9개와 원 데이터의 최근 9개 데이터를 입력으로 주었다. Babu & Reddy 혼합 모델은 데이터를 추세데이터(이동평균 필터: 38)와 나머지로 분해한 후, 추세 데이터에는 ARIMA(4,0,1)을 사용하여 예측하고, 나머지데이터는 $FFNN^{9 \times 9 \times 1}$ 을 사용하여 예측한 후 둘을 합하여 최종 예측값을 구하였다.

제안모델은 셀 크기가 256개인 GRU 순환신경망으로 하고, 최근 데이터 7개를 사용하여 학습하도록 하였다. 실험에 사용한 모델의 자세한 형태는 〈표 4-3〉에 정리하였으며, 실험결과는 〈표 4-4〉에서 보는 바와 같다. 제안모델 (A)는 ARIMA 대비 8%의 정확도 향상을 보였으며, 이는 〈그림 4-2〉를 통해 확인할 수 있다.

Sunspot 데이터의 경우에는 “순환신경망만 사용한 모델”이 “순환신경망 혼합 모델”보다 더 좋은 정확도를 보였다. 따라서 데이터 전처리를 하여 추가적인 정보를 입력하여도 학습이 도움이 되지 않는 경우도 있음을 알 수 있다.

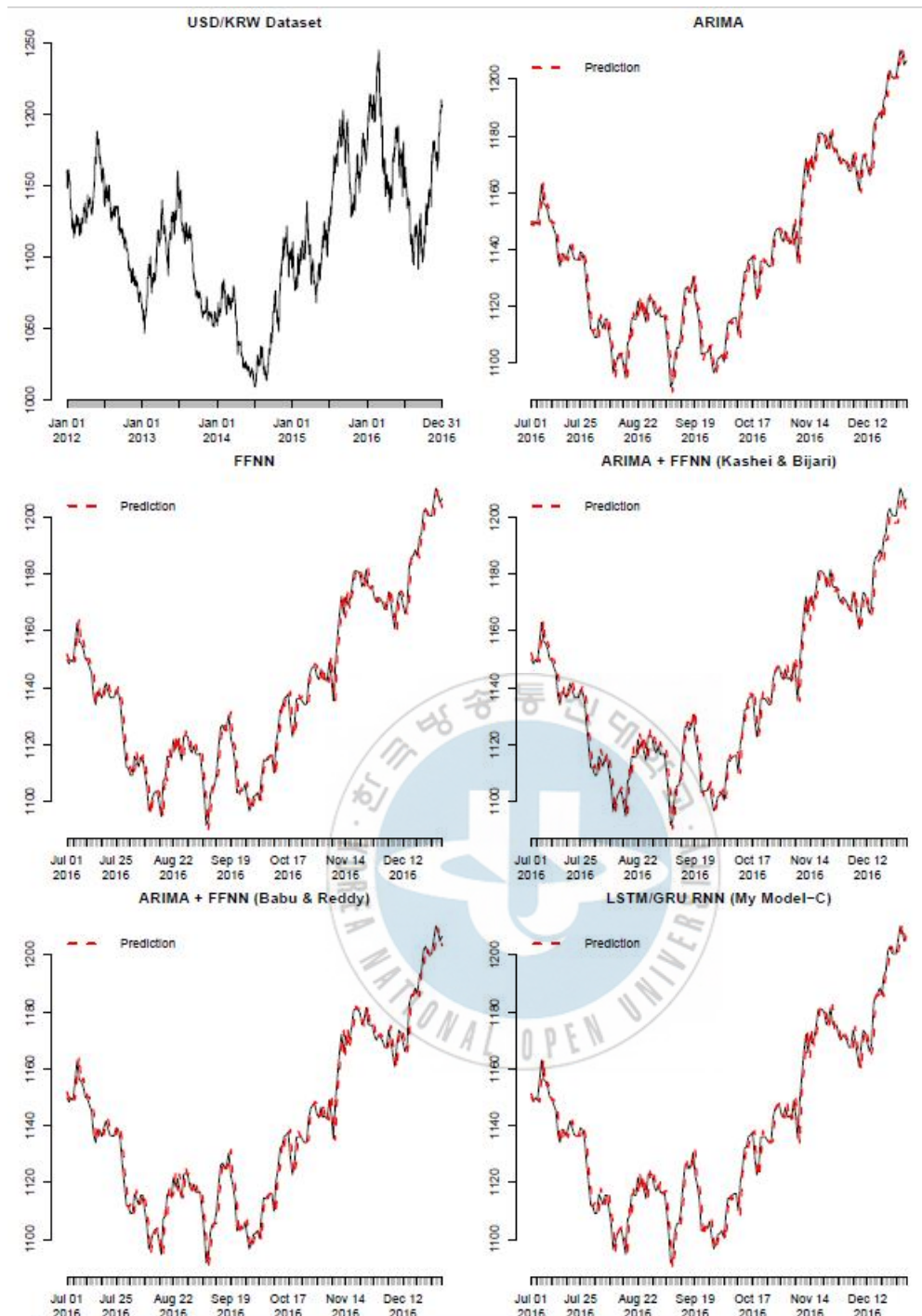
4.5 환율 Data 예측 실험

2011.1.1일 부터 2016.12.31일 까지 5년간의 USD vs KRW 환율데이터를 사용하여 실험을 하였다. 총 데이터 포인트 1,827개 중 1,643개(2012.1.1-2016.6.30)는 데이터 학습에 사용하고, 184개(2016.7.1-2016.12.31)는 테스트에 사용하였다.

Kashei & Bijari 혼합 모델은 AR(0,1,2)로 전처리하여 예측오차를 구하고 최근 예측 오차 6개와 최근 원데이터 6개를 전방향신경망 $FFNN^{12 \times 64 \times 1}$ 에 입력하여 예측값을 구하였다. Babu & Reddy 혼합 모델은 데이터를 추세 데이터(이동평균 필터: 60)와 나머지로 분해하여 추세 데이터는 ARIMA(3,2,0)을 사용하여 예측하고, 나머지는 전방향신경망 $FFNN^{6 \times 4 \times 1}$ 을 사용하여 예측하여 둘을 더하여 예측값을 구하였다. 모든 신경망은 Dropout 0.5, Xavier 가중치 초기화를 적용하여 학습하였다.

<표 4-5> 환율 data 실험에 사용된 모델의 형태

모델	형태	학습율	비고
ARIMA	ARIMA(0,1,2)	N/A	
전방향신경망	$FFNN^{6 \times 4 \times 1}$	0.005	
Khashei & Bijari 혼합 모델	ARIMA(0,1,2) → $FFNN^{12 \times 64 \times 1}$	0.01	ARIMA로 구한 최근 예측 오차 데이터 6개 + 최근 원데이터 6개를 신경망에 입력
Babu & Reddy 혼합 모델	ARIMA(3,2,0) + $FFNN^{6 \times 4 \times 1}$	0.005	원데이터를 추세(MA filter=60)와 나머지로 분리하고, 각각 ARIMA 와 전방향신경망으로 예측 후 결과를 합하여 출력
My Model (A)	순환신경망	0.01	Cell Type & Size: GRU, 256 최근 3개 데이터포인트를 순환신경망에 입력
My Model (B)	ARIMA(0,1,2) + My Model (A)	0.001	ARIMA 예측오차 3개와 원데이터 3개를 순환신경망에 입력
My Model (C)	Decomposition + My Model (A)	0.001	원데이터를 추세(MA filter=60)와 나머지로 분리하여 각각 3개씩 순환신경망에 입력



<그림 4-3> 환율 data 각 모델별 실제값과 예측값 비교

〈표 4-6〉 환율 data 제안모델과 기존 예측모델 간의 정확도 비교

모델	MAE	향상율	비고
ARIMA	3.4564	-	ARIMA(0,1,2)
전방향신경망	3.5074	-1.48%	$FFNN^{6 \times 4 \times 1}$
Khashei & Bijari 혼합 모델	3.6432	-5.40%	ARIMA(0,1,2) \rightarrow $FFNN^{12 \times 64 \times 1}$
Babu & Reddy 혼합 모델	3.5159	-3.83%	ARIMA(3,2,0) + $FFNN^{6 \times 4 \times 1}$
My Model (A)	3.5134	-1.65%	순환신경망만 사용
My Model (B)	3.7204	-7.64%	ARIMA(0,1,2) \rightarrow 순환신경망
My Model (C)	3.4540	0.07%	데이터 분리 \rightarrow 순환신경망

제안모델은 셀 크기가 128개인 GRU 순환신경망으로 하고, 최근 데이터 3개를 사용하여 학습하였다.

실험결과는 〈표 4-6〉에서 보는 바와 같이 ARIMA, 전방향신경망, 제안모델 (A)와 제안모델 (C)가 거의 동일한 예측 정확도를 보였으며, Khashei & Bijari 혼합 모델과 Babu & Reddy 혼합 모델은 오히려 더 나쁜 결과를 보였다.

앞의 두 실험과 다르게 제안모델이 더 우수한 결과를 보이지 못한 이유는, 환율 데이터가 과거 데이터와의 연관성이 없이 랜덤하게 움직이기 때문에 선후 문맥 정보가 학습에 별 도움이 되지 않기 때문이다. 그러나 이러한 경우에도 순환신경망으로 최소한 ARIMA나 전방향신경망 만큼의 예측 정확도를 얻을 수 있었다.

4.6 학습시간과 실행시간 비교

본 논문의 주된 목적은 시계열데이터에 대한 예측 정확도를 향상시키는데 있으며, 지금까지 예측 정확성에 대한 비교 실험 결과를 기술하였다. 그러나 알고리즘의 성능을 평가할 때는 정확성 뿐 아니라 메모리, 실행시간의 측면을 함께 고려할 필요가 있다.

〈표 4-7〉 모델별 학습시간과 실행시간 비교

모델		학습시간 (초)	실행시간 (초)	가중치 (개수)	학습반복 (횟수)
■ Lynx 데이터: 학습데이터 100개					
ARIMA	ARIMA(12,0,0)	0.2519	1.4×10^{-4}	-	1
전방향신경망	$FFNN^{12 \times 6 \times 1}$	8.5692	1.8×10^{-4}	85	941
My Model (A)	셀: GRU 256개 순환층 반복: 12회	37.8173	1.5×10^{-2}	1,025	653
■ Sunspot 데이터: 학습데이터 222개					
ARIMA	ARIMA(9,0,0)	0.1787	3.0×10^{-5}	-	1
전방향신경망	$FFNN^{9 \times 9 \times 1}$	14.1746	9.1×10^{-5}	100	653
My Model (A)	셀: GRU 256개 순환층 반복: 7회	362.3245	2.0×10^{-3}	1,025	551
■ 환율 데이터: 학습데이터 1,643개					
ARIMA	ARIMA(0,1,2)	0.0231	7.1×10^{-6}	-	1
전방향신경망	$FFNN^{6 \times 4 \times 1}$	2805.7492	9.6×10^{-5}	33	20,000
My Model (A)	셀: GRU 256개 순환층 반복: 3회	17962.4720	3.7×10^{-4}	1,025	20,000

학습 소요시간을 보면 ARIMA는 1초 미만이나, 전방향신경망은 8~2806초, 순환신경망(A)의 경우는 37~17962초가 소요되었다. 학습시간에서는 ARIMA가 인공신경망보다 훨씬 유리하다. 인공신경망의 경우는 반복학습 메커니즘을 사용하여 학습하기 때문에 시간이 많이 소요되며, 뉴런의 수와 학습 반복 횟수에 따라 학습시간이 크게 달라진다.

실행시간에서도 ARIMA가 가장 유리하다. 하지만 제일 느린 모델의(Lynx 데이터, 순환신경망) 실행시간이 0.015초를 보여 학습시간 만큼 차이가 크지는 않다.

인공신경망의 메모리 사용량을 비교하기 위해 가중치 개수를 조사하였다. 이 실험에서는 순환신경망이 전방향신경망보다 가중치를 많이 가지므로 메모리도 많이 소비한다.

제5장 결론

주가, 환율, 매출액, 출국자수 등의 시계열데이터의 정확한 예측은 사업계획 수립이나 조직의 성과에 큰 영향을 주기 때문에 예측 정확도를 향상시키려는 노력이 계속되어 왔다. 지수평활법이나 ARIMA와 같은 통계적인 접근법과 인공신경망을 이용한 방법이 있었고 ARIMA와 인공신경망의 장점을 결합한 혼합 접근방법이 시도되었다. 인공신경망의 경우 지금까지 완전연결 전방향신경망이 시계열데이터 예측에 주로 사용되어 왔다.

순환신경망은 외부 입력데이터와 이전 단계의 상태정보를 함께 입력받아 처리하여 출력하는 과정을 반복하므로, 데이터의 선후관계의 문맥을 인식하는 학습이 가능하다. 그로 인해 시간 흐름에 따라 데이터가 주어지는 시계열데이터를 전방향신경망보다 잘 처리할 수 있다. 과거에는 순환반복이 많아지면 그라디언트 소실 문제가 발생하여 순환신경망 학습이 쉽지 않았으나, LSTM/GRU 셀을 이용하면 시퀀스가 긴 시계열데이터도 학습이 가능하다. 인공신경망의 학습은 가중치를 일정 범위 내에서 골고루 분포하도록 초기화 할 경우 좋은 결과를 보이며[5], Dropout 기법을 통해 과다 학습을 방지할 수 있다[6].

이 논문에서는 Xavier 가중치 초기화 기법과 Dropout 과다학습 방지 기법을 적용한 LSTM/GRU 순환신경망 시계열데이터 예측 모델을 제안하였으며, 공개 데이터에 대한 실험을 통해 기존의 방법들(ARIMA, 전방향신경망, 이 둘의 혼합 모델)보다 더 높은 예측 정확도를 얻을 수 있음을 확인하였다. 순환신경망과 ARIMA의 혼합을 통해 더 나은 결과를 얻을 수 있는 경우가 있었으나, 데이터에 따라 상반된 결과를 보이고 있어, 혼합 모델의 유용성에 대해서는 추가 연구가 필요하다.

순환신경망은 여러 개의 층으로 쌓을 수도 있고, 역방향으로 정보가 흐르도록 망을 구성할 수도 있으며, 컨볼루션신경망과 같이 서로 다른 종류의 신경망을 결합하여 새로운 형태의 신경망을 구축하는 것도 가능하다. 향후 다양한 신경망 결합 모델에 대한 연구를 통해 한층 높은 예측력을 갖는 시계열데이터 예측모델을 만들 수 있을 것으로 생각한다.

참고문헌

- [1] C Narendra Babu and B Eswara Reddy, “A moving-average filter based hybrid arima-ann model for forecasting time series data”, Applied Soft Computing, Vol.23, 27-38, 2014.
- [2] George Edward P Box and Gwilym M Jenkins, “Time series analysis: forecasting and control, 1976”, ISBN: 0-8162-1104-3, 1970.
- [3] Tamal Datta Chaudhuri and Indranil Ghosh, “Artificial neural network and time series modeling based approach to forecasting the exchange rate in a multivariate framework”, arXiv preprint arXiv:1607.02093, 2016.
- [4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation”, arXiv preprint arXiv:1406.1078, 2014.
- [5] Xavier Glorot and Yoshua Bengio, “Understanding the difficulty of training deep feedforward neural networks.”, in Aistats, volume 9, 249-256, 2010.
- [6] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, “Improving neural networks by preventing coadaptation of feature detectors”, arXiv preprint arXiv:1207.0580, 2012.
- [7] Rob J Hyndman and George Athanasopoulos, Forecasting: principles and practice, OTexts, 2014.
- [8] Mona Ebadi Jalal, Monireh Hosseini, and Stefan Karlsson, “Forecasting incoming call volumes in call centers with recurrent neural networks”, Journal of Business Research, 4811-4814, 2016.
- [9] Mehdi Khashei and Mehdi Bijari, “An artificial neural network (p, d, q) model

- for timeseries forecasting” , Expert Systems with applications, Vol.37, No.1, 479-489, 2010.
- [10] Fei Li, Jin Liu, and Lei Kong, “Predict time series with multiple artificial neural networks” , International Journal of Hybrid Information Technology, 9, 7, 313-324, 2016.
- [11] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting.” , Journal of Machine Learning Research, 15, 1, 1929-1958, 2014.
- [12] Jie Wang and Jun Wang, “Forecasting energy market indices with recurrent neural networks: Case study of crude oil price fluctuations” , Energy, Vol.102, 365-374, 2016.
- [13] Jie Wang, Jun Wang, Wen Fang, and Hongli Niu, “Financial time series prediction using elman recurrent random neural networks” , Computational Intelligence and Neuroscience, Vol.2016, Article ID 4742515, 14 pages, 2016.
- [14] Paul J Werbos, “Backpropagation through time: what it does and how to do it” , Proceedings of the IEEE, 78, 10, 1550-1560, 1990.
- [15] G Peter Zhang, “Time series forecasting using a hybrid arima and neural network model” , Neurocomputing, Vol.50, 159-175, 2003.
- [16] 김영주, “계절 arima 모델을 활용한 저가항공 수요예측 - 국내선 제주권,내륙권 예측 비교 분석” , 관광연구논총, 제26권, 제1호, 3-25, 2014.
- [17] 이재현, “웨이블렛 신경회로망을 이용한 상품 수요 예측 모형에 관한 연구” , JKIECS, 131-136, 2014.
- [18] 장호현 이장춘 이주현, 김종석, “다층 퍼셉트론 인공신경망 모델을 이용한 가물예측” , 한국수자원학회논문집, 제46권, 제12호, 1249-1263, 2013.

A B S T R A C T

Forecasting Time-series Data Using LSTM/GRU Recurrent Neural Networks

by

Kim, Ho Hyun

Department of Computer Science
Graduate School
Korea National Open University

Supervised by Professor : Lee, Kwan Yong

Forecasting for time series data such as stock price, exchange rate, sales amount has been very important subject for many researchers. Traditionally, statistic based ARIMA model and simple feed forward neural networks model have been used widely in real world. Some researchers suggested hybrid model of the two to improve forecasting precision.

Recurrent neural networks has a special nature to learn how things change as time flows. But it has not been used often in time series forecasting domain because training recurrent neural networks was not easy thing due to some unsolved problems including vanishing gradient.

In this paper, I suggest new approach using LSTM/GRU recurrent

neural networks in the time series forecasting problems. By incorporating recurrent neural networks, LSTM/GRU cell, and recent deep learning theories such as Dropout and Xavier parameter initialization, it is possible to establish the time series forecasting model that has higher prediction precision than current models. This paper proves it by experiments on open datasets.



감사의 글

저는 약 5년 전 어느 날 아내를 통해 우연히 방송통신대학교를 접하게 된 후, 평소 IT에 대한 호기심과 퇴직 이후의 삶에 대한 고민의 결과로 뒤늦은 공부를 다시 시작하게 되었습니다. 2013년 3월 방송통신대학교 컴퓨터과학과 3학년에 편입하여 2년간의 학부과정을 마치고, 2015년 봄 방송통신대학교 대학원 정보과학과에 입학하여 2년 반의 석사과정을 마치고 이제 졸업을 앞두고 있습니다.

경영학도였던 제게는 비교적 생소한 분야여서 장성한 세 자녀를 둔 가장으로서 부끄럽지 않으려고 오히려 젊은 날 학창시절 보다 더 열심히 하려고 했는데, 이제 목표했던 논문을 끝내고 졸업할 수 있게 되어 감개가 무량합니다.

이 논문은 이관용 교수님의 정성어린 지도가 없었다면 완성할 수 없었을 것입니다. 처음부터 끝까지 이끌어 주신 이관용 교수님과 바쁘신 일정에도 불구하고 심사과정에서 세세한 부분까지 열심히 지도해 주신 이병래 교수님, 정광식 교수님께 감사 드립니다. 또한 좋은 강의로 후학들을 성심껏 지도해 주시는 정보과학과 교수님들께도 감사의 말씀을 전합니다.

학업의 모든 과정에 함께 해 주시고 결실을 맺을 수 있도록 인도해 주신 하나님께 특별히 감사 드리며 이 모든 영광을 주님께 돌립니다. 그리고 항상 저를 위해 기도해 주시는 인천국제침례교회의 믿음의 형제 자매님들께 감사 드립니다.

언제나 곁에서 기도와 격려로 힘이 되어 준 사랑하는 아내와 하나님의 자녀로 잘 자라 준 든직한 아들 규혁이와 규석이, 그리고 예쁘기 그지없는 딸 규원이에게 이 논문을 바칩니다.

2017년 7월 김 호 현

학위논문 공개 동의서

한국방송통신대학교 대학원은 귀하의 학위논문을 데이터베이스화하여 내용의 일부 또는 전부를 어떤 형태로든 국내·외 이용자에게 공개(열람 또는 배포)하는 것에 대하여 동의를 얻고자 하오니 동의여부를 기재하여 주시기 바랍니다.

학 위	이학석사	졸업년도	2017년
전 공	정보과학		
학위논문 제 목	한글	LSTM/GRU 순환신경망을 이용한 시계열데이터 예측	
	영문	Forecasting Time Series Data Using LSTM/GRU Recurrent Neural Networks	
동 의 여 부	1. 찬성(○) 2. 조건부 찬성() 3. 반대()		
조건 또는 사유	※ “조건부 찬성” 또는 “반대” 하는 경우 그 사유 기재		

2017년 7월 5일

논문저자명 : 김 호 현 (서명 또는 날인)

한국방송통신대학교총장 귀하