

RNN과 LSTM을 이용한 주가 예측을 향상을 위한 딥러닝 모델

신동하*, 최광호**, 김창복***

Deep Learning Model for Prediction Rate Improvement of Stock Price Using RNN and LSTM

Dong-Ha Shin*, Kwang-Ho Choi**, and Chang-Bok Kim***

요 약

최근 딥러닝을 이용한 주가예측은 기본적으로 보조지표를 예측요소로 사용하고 있으나, 보조지표는 분석가의 주관적인 관점이기 때문에, 예측요소에 대한 적합여부에 대한 검토가 필요하다. 본 연구는 R의 신경망 패키지인 Keras를 통해, 기존의 보조지표에 대해서 다양한 조합으로 예측요소 적합여부를 검토하고, 예측율 향상을 위해 최적의 보조지표 조합과 환율, 환율 이동평균, 전산업생산지수 등 환경 예측요소들에 대해서 연구하였다. 또한, 추출된 예측요소를 입력 패턴으로 DNN, RNN, LSTM 등의 딥러닝 모델을 제안하였다. 연구결과 대부분의 보조지표는 예측율을 저하하는 현상이 있었으며, 추가 환경 예측요소를 통해, 예측율이 향상되었다. 또한, DNN에 비해 시계열 딥러닝 네트워크인 RNN과 LSTM이 빠르고 안정적으로 학습하였으며, 종목별로 차이는 있으나 대략 15% 정도의 예측율 향상을 보였다.

Abstract

Recently, stock price prediction using deep learning has basically used assistance index as a prediction factors. However assistance index is necessary to examine whether it is suitable as prediction factors because it is subjective viewpoint of researcher. In this study, we examine the suitability as prediction factors with various combinations of existing assistance indexes through the R neural network package, and studied the optimal combinations of assistance indexes and environmental prediction factors like exchange rate, exchange rate moving average, and whole industrial production index in order to improve the prediction rate. In addition, we proposed a deep learning model like DNN, RNN, LSTM which have input-output with extracted prediction factors. As a result, most of the assistance indexes decreased the prediction rate and the prediction rate was improved through additional environmental prediction factors. Also, RNN and LSTM, which are time series deep learning networks, were learned quickly and steadily compared to DNN. Although there is a difference by items, the prediction rate improvement is about 15%.

Keywords

technical analysis, fundamental analysis, artificial neural network, deep neural network, RNN, LSTM

* 가천대학교 IT 융합공학과

- ORCID: <http://orcid.org/0000-0002-7835-6184>

** (주) 에이치티원 대표

- ORCID: <http://orcid.org/0000-0002-3956-6998>

*** 가천대학교 에너지 IT학과 교수(교신저자)

- ORCID: <http://orcid.org/0000-0002-7155-5033>

• Received: Aug. 01, 2017, Revised: Sep. 13, 2017, Accepted: Sep. 16, 2017

• Corresponding Author: Chang-Bok Kim

Department of Energy IT, Gachon University, 1342, Seongnam-daero,

Sujeong-gu, Seongnam-si, Gyeonggi-do, Korea

Tel.: +82-32-446-0695, Email: cbkim@gachon.ac.kr

I. 서 론

주식은 살아 움직이는 생명체처럼 다양한 종속변수에 의해 움직이며, 많은 잡음(Noise), 비정상성(Non-stationarity), 그리고 비선형성으로 인해 가격 변동이나 기대 수익을 예측하는 것은 매우 어려운 작업이다[1][2]. 주가는 특정시점에 알려진 모든 정보가 이미 반영되어 있으므로, 주가를 예측하는 것은 불가능하다는 주장으로 인해, 기존의 분석방법론들에 대해 회의적이었다. 그러나 최근 기계학습 분야에서 인공신경망(Artificial Neural Network)과 기술 분석의 결합에 의한 주가예측을 통해 시장 평균을 초과하는 수익의 달성이 가능하다는 연구 결과들을 제시하고 있다[3]-[6]. 딥러닝(Deep Learning)은 인간의 뇌 구조를 모방한 기계학습이며, 기존 인공신경망의 문제점을 개선하여, 영상인식, 음성인식, 자연어 처리, 미래 예측 등 다양한 영역에서 사용되고 있다. 딥러닝은 다른 형태의 기계학습과는 달리 입력 데이터의 특징 패턴을 그대로 학습하기 때문에, 학습 이전에 입력 데이터의 특징 패턴에 따라 예측 결과가 큰 차이가 있다[7].

본 연구는 기본 데이터, 보조지표 데이터와 같은 기술 분석 데이터 뿐 아니라 다양한 예측요소 데이터를 이용하여, 주가예측을 위한 딥러닝 모델에 대해서 연구하였다. 주가 데이터는 비교적 가격 안정성이 높은 KOSPI 200 종목에서 선택하였으며, 기술 분석 데이터는 기본적으로 사용되는 시가, 고가, 저가, 종가, 거래량 등의 기본 데이터와 11개의 보조지표를 사용하였다. 또한, 예측요소 데이터는 등락구분, 환율, 전산업생산지수 등을 사용하였다.

본 연구는 모든 주가 예측요소들에 대해서 통계 오픈소스인 R의 neuralnet 신경망 패키지를 이용하여, 효율적인 예측요소의 조합을 선택하였다[8]. 선택된 예측요소는 딥러닝 네트워크의 독립변수인 입력요소로 사용하였으며, 종속변수인 출력라벨은 당일의 종가로 하였다. 주가 예측을 위한 딥러닝 네트워크는 DNN(Deep Neural Network), RNN(Recurrent Neural Networks), LSTM(Long Short Term Memory) 등을 사용하였다[9][10].

본 논문은 2장에서 관련연구로서 기술 분석과 딥

러닝 그리고 기술 분석 보조지표에 대해서 서술하였다. 또한, 3장에서 주가예측모델을 제안하였으며, 4장에서 연구 결과를 나타내고 마지막으로 결론에 대해서 서술하였다.

II. 기술분석

2.1 기술 분석

주가는 주식의 수요와 공급에 의해서 결정되며, 주가 변동은 기술적요인, 기본적요인, 심리적요인 등에 대한 모든 정보가 반영되어 있다. 또한, 주가는 사소한 변동을 무시한다면, 상당 기간 같은 방향으로 움직임을 지속하려는 경향을 갖고 있다. 주가 모형은 투자의 심리가 반영된 결과이며, 투자의 심리적 속성이 바뀌지 않는 한 패턴은 미래에 반복될 가능성이 높다. 일반적으로 주가 분석을 위해 기술 분석과 기본분석을 이용한다.

기술 분석은 전통적인 주가예측 방법으로 과거 주가나 거래량을 이용하여, 주가 변화의 패턴을 분석하여 주가를 예측하는 방법이다. 기술 분석은 기본적으로 사용되는 시가, 고가, 저가, 종가, 거래량 등의 기본 데이터와 캔들 차트(Candle Chart), RSI(Relative Strength Index), CCI(Commodity Channel Index), 이동평균(Moving Average), 지수 이동평균(Exponential Moving Average), 이격도(Disparity), MACD(Moving Average Convergence and Divergence), stochastic1, SIGNAL 등 다양한 보조지표를 이용한다[11][12]. 표 1에 주가 예측에 많이 사용하는 보조지표에 대해서 나타냈다. 이동평균은 주가 및 거래량이 특정기간에 일정한 방향성을 가지고 있는 것을 수치화한 것이다. MA_t^n 은 특정일 t에서의 n일간 이동평균이며, $dose_t$ 는 당일의 종가이다. 그랜빌의 법칙은 주가의 추세 예측에 많이 사용하는 지표로서, 일반화된 형태로 예측에 적용하기 위해서, 이동평균선의 기울기(Gradient of Moving Average)와 이동평균선간 거리(Distance of Moving Average)를 표현하는 변수들의 정의가 필요하다. 지수 이동평균은 최근의 값에 좀 더 큰 가중치를 부여하는 이동 평균 방법이다. MACD 지수 이동평균 값 간의 거리를

나타낸다. 이격도 증가와 이동평균선과의 거리이며, 이동평균선의 단점을 보완하기 위해서 만들어진 보조지표이다. RSI는 현재의 주가 추세 강도를 백분율로 나타내어, 추세 전환을 예측하는 지표이다. 여기서 AU_t^n 은 특정일 t에서 n일간의 주가 상승폭의 합계의 평균을 나타내며, AD_t^n 은 특정일 t에서 n일간의 주가 하락폭의 합계의 평균을 나타낸다. CCI는 이동평균으로 부터 주가의 변동성을 측정하는 지표로 추세의 강도와 방향을 표시한다. 여기서 M_t 은 당일의 고가, 저가, 종가를 3으로 나눠서 계산한다. 또 MA_t^n 는 M_t 값을 n일의 이동평균값이며, D는 $|M_t - MA_t^n|$ 에 대한 n일의 이동평균 값이다. Stochastic1은 과일 및 침체를 나타내는 지표이다. 여기서 X는 $close_{t-m}$ 에서 $close_t$ 까지의 범위이다. Stochastic 값의 범위는 항상 0~100% 사이가 된다. 100%라면 현재 가격이 N일간 최고가이므로 매수세가 가장 강한 경우가 되며, 0%라면 현재 가격이 N일간 최저가이므로 매도세가 가장 강한 경우가 된다.

2.2 RNN과 LSTM

RNN은 하나의 입출력 패턴을 가진 DNN 병렬 체인 구조로 연결한 형태로, 과거 학습결과를 현재 학습에 사용하는 딥러닝 네트워크로서, 시계열 데이터를 처리하는데 효과적이다. 그림 1에 RNN 구조에 대해서 나타냈다.

RNN은 하나의 tanh 혹은 ReLU 활성화함수를 가진 구조로서, 체인이 길어지면 과거의 학습 결과가 사라지는 장기 의존성문제가 있다.

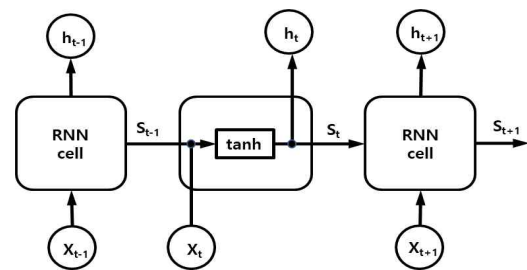


그림 1. RNN 구조
Fig. 1. RNN structure

표 1. 기술분석 및 보조지표

Table 1. Technical analysis and sub-indicators

Index	Assistance index	Formula
1	Moving Average	$MA_t^n = \frac{1}{n} \left(\sum_{i=t-n+1}^t close_i \right)$
2	Gradient of Moving Average	$g_t^n = \frac{(MA_t^n - MA_{t-1}^n)}{MA_{t-1}^n}$
3	Distance of Moving Average	$d_t^{m,n} = \frac{(MA_t^n - MA_t^m)}{MA_t^m}, (m > n)$
4	Exponential Moving Average	$EMA(m)_t = close_t \times EP + close_{t-1} \times (1 - EP)$ $EP(Exponential Percentage) = \frac{2}{m+1}$
5	Moving Average (Moving Average Convergence & Divergence)	$MACD = \sum_{i=t-9}^t (EMA(12)_i - EMA(26)_i)$
6	Disparity	$disparity = \frac{close_t}{MA_t^n} \times 100$
7	RSI(Relative Strength Index)	$RSI = \frac{AU_t^n}{AU_t^n + AD_t^n} \times 100$
8	CCI(Commodity Channel Index)	$CCI = \frac{M_t - MA_t^n}{D \times 0.015}, M_t = \frac{high_t + low_t + close_t}{3}$ $MA_t^n = \frac{1}{n} \left(\sum_{i=t-n+1}^t M_i \right) D = \frac{1}{n} \left(\sum_{i=t-n+1}^t M_i - MA_t^n \right)$
9	Stochastic1	$\%K(m) = \frac{close_t - MIN(t:t-m)}{MAX(t:t-m) - MIN(t:t-m)} \times 100$

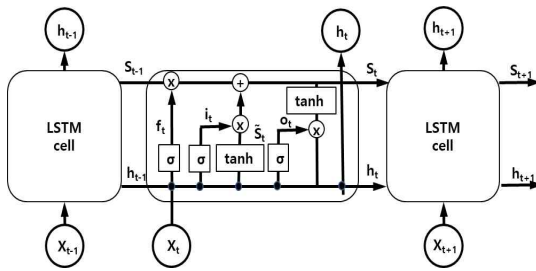


그림 2. LSTM 구조
Fig. 2. LSTM structure

LSTM은 전체 체인을 관통하는 셀 스테이트(Cell State)를 통해 과거 학습결과를 큰 변함없이 전달하는 구조로 장기 의존성 문제를 해결한다.

LSTM은 첫번째 단계로 sigmoid 함수를 이용해 삭제할 정보를 결정하고, 두 번째 단계로 또 다른 sigmoid 함수와 tanh 함수를 이용해 새로운 정보가 셀 스테이트에 저장될지 결정한다. 세 번째 단계는 셀 스테이트를 업데이트하고 마지막으로 마지막 sigmoid 함수와 셀 스테이트에서 나온 출력을 통과 시킨 마지막 tanh 함수로 어떤 출력 값을 출력할지를 결정한다[13]-[15]. 그림 2에 LSTM의 구조에 대해서 나타냈다.

III. 주가예측 제안 딥러닝

본 연구는 주가 예측요소 분석을 위해 시가, 고가, 저가, 종가, 거래량 등의 기본 데이터와 11개의 보조지표를 사용하였다. 또한, 추가 보조 예측요소로 등락구분, 환율, 전산업생산지수 등을 사용하였다. 모든 예측요소는 통계 및 데이터 마이닝을 위한 오픈소스인 R의 neuralnet 신경망 패키지를 이용하여, 가장 좋은 예측요소의 조합을 선택하였다. 주가 데이터는 2009년 2월부터 2016년 12월까지의 총 1965개의 일일 데이터에서 훈련데이터 1900개와 테스트데이터 65개를 사용하였다. 또한, 표 2에 다양한 예측요소에 대한 R의 neuralnet 결과 중 일부분을 나타냈다.

딥러닝은 입출력 데이터의 크기가 작아야 학습이 효과적이기 때문에, 보편적으로 사용되고 있는 정규화 기법을 사용하여, 모든 입출력 데이터를 [0,1]의 구간으로 스케일을 동일하게 만들었다. 또한, 실험 결과는 예측 값과 실제 값과의 RMSE (Root Mean

Square Error)이며, 결과 값이 실행할 때 마다 차이가 있어 5번 반복하여 평균으로 결과를 측정하였다. (1)은 모든 보조지표를 입력으로 한 결과로서 오히려 RMSE가 가장 높게 나타났다. (2)와 (3)은 다양한 보조지표 실험결과 중 일부분이며, (4)는 모든 보조지표 중 가장 RMSE가 낮게 측정된 결과이다. 또한, (5)와 (6)은 (4)의 결과에 환율과 전산업생산지수를 추가한 결과이다. 최종적으로 (6)의 예측요소에서 가장 우수한 결과가 산출되었다.

이와 같은 실험 결과 본 연구의 딥러닝 입출력 모델은 시가(X1), 종가(X2), 고가(X3), 저가(X4), 거래량(X5), Stochastic(X6), CCI(X7), 등락구분 5일(X8), 등락구분 20일(X9), 환율(X10), 환율 이동평균 5일(X11), 전산업생산지수(X12) 등의 예측요소를 딥러닝의 입력 패턴으로 하였으며, 출력 라벨은 매일의 종가(Y)로 하였다. 그림 3은 딥러닝 DNN 모델에 대해서 나타냈다.

표 2. 예측요소 실험결과

Table 2. Prediction elements experiment results

(1) MA(5), MA(20), MA Gradient, MA Distance, Disparity, RSI, CCI, EMA(12), EMA(26), Stochastic1, MACD, SIGNAL					
First	Second	Third	Fourth	Fifth	Avg.
0.0262	0.0257	0.0257	0.0291	0.0272	0.0268
(2) MA(5), MA(20), MA Gradient, MA Distance, Disparity, RSI, CCI					
First	Second	Third	Fourth	Fifth	Avg.
0.0231	0.024	0.025	0.0249	0.0237	0.0241
(3) EMA(12), EMA(26), Stochastic 1, MACD, SIGNAL					
First	Second	Third	Fourth	Fifth	Avg.
0.0277	0.0226	0.0255	0.0227	0.022	0.0241
(4) CCI, stochastic1, Fluctuation(5), Fluctuation(20)					
First	Second	Third	Fourth	Fifth	Avg.
0.0233	0.0229	0.0228	0.0237	0.0232	0.0232
(5) CCI, stochastic1, Fluctuation(5), Fluctuation(20), Exchange Rate, ER fluctuation Rate, ER MA(5), ER MA(20)					
First	Second	Third	Fourth	Fifth	Avg.
0.0228	0.0223	0.0235	0.022	.0229	0.0227
(6) CCI, stochastic1, Fluctuation(5), Fluctuation(20), Exchange Rate, ER MA(5), Index of All industry production					
First	Second	Third	Fourth	Fifth	Avg.
0.0221	0.0218	0.0211	0.023	0.021	0.0218

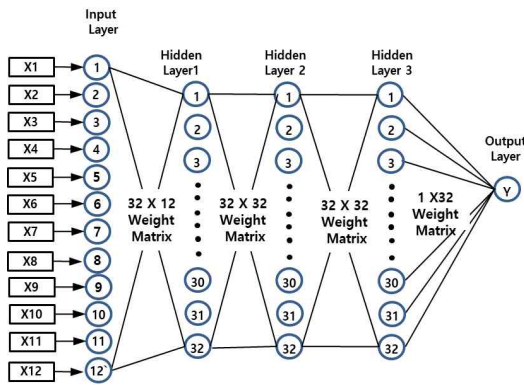


그림 3. DNN 모델
Fig. 3. DNN model

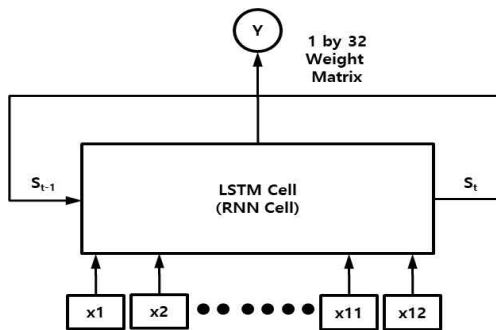


그림 4. RNN 모델
Fig. 4. RNN model

DNN 모델은 입력층 12노드, 중간층 32 노드로 하여 3층, 출력층 1 노드로 구성하였다. 또한 활성화 함수는 ReLU(Rectified Linear Unit)를 사용하고, xavier 초기화로 가중치를 초기화 하였다. 최적화 알고리즘은 오차 감소속도가 빠른 AdamOptimizer 알고리즘을 사용하였다. 본 연구는 주가 데이터가 시계열 데이터 이므로, 시계열 데이터에 적합한 딥러닝 알고리즘인 RNN과 LSTM을 통해 예측 모델을 구축하였다. RNN과 LSTM 모델은 DNN 모델과 동일한 입출력 변수를 사용하였다. 그림 4에 제안 RNN과 LSTM 모델에 대해서 나타냈다.

RNN과 LSTM 모델의 입력 노드는 12개의 주가 예측요소이다. 또한, 주식시장이 월요일부터 금요일까지 5일장이므로 타임 스탬프(time stamp)를 5개로, 5개의 셀을 구축하여, 한번에 60개의 주가 예측요소를 입력하여 1주일 씩 학습하였다. 중간층 노드는 32개 이고 출력 라벨은 다음날의 종가이다.

IV. 실험 및 결과

본 연구의 실험환경은 리눅스 우분투 운영체제와 Mac OS Sierra를 기반으로 통계 오픈소스인 R과 파이썬(Python) 2.7기반 텐서플로우(Tensorflow)를 이용하였다. 주가 데이터는 2009년 2월에서 2016년 12월 기간의 가격 안정성이 높은 KOSPI 200 종목에서 5 종목을 선택하였다. 주가 예측요소는 실험을 통해, 시가, 고가, 저가, 종가, 거래량 등 기본 데이터 5개와 CCI, stochastic1 등 보조 데이터 2개 그리고 등락구분(5), 등락구분(20), 환율, 환율 이동평균 (5), 전산업생산지수 등 환경 데이터 5개를 사용하였다.

DNN을 이용하여 실험할 때에는 다음날 예측이기 때문에 목표로 설정된 종가의 맨 앞 1일치 데이터를 삭제하였고, 그에 맞춰 예측요소의 맨 뒤 데이터를 삭제하였다. 따라서 1880개의 학습 데이터와 85개의 테스트 데이터를 사용하였다. RNN과 LSTM을 이용하여 실험할 때에는 5개의 Cell을 사용했기 때문에 목표 값인 종가보다 예측요소에서 5일 앞선 데이터가 필요했다. 따라서 1876개의 학습 데이터와 90개의 테스트 데이터를 사용하였다. 딥러닝은 작은 데이터의 입출력으로 학습하여야 효율적이기 때문에, 입력 데이터를 정규화하여, 0부터 1사이의 값을 사용하였으며, 출력 값은 추후 프로파일 등의 계산을 위해 명일의 종가를 10000 또는 100000으로 나누어 소수점 이상 한자리로 하여 실험하였다.

주가예측의 정확성은 오차율을 통해 확인할 수 있다. 본 연구는 다음과 같이 평균 제곱근오차(Root Mean Square Error)와 상대표준오차(Relative Standard Error)를 사용하였다.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (actual - predict_i)^2}{n}} \quad (1)$$

$$RSE = \frac{\sum_{i=1}^n (predict_i - actual_i)^2}{\sum_{i=1}^n (actual_i - average)^2} \quad (2)$$

본 실험은 DNN 3층과 시계열 학습을 위한 RNN과 LSTM을 이용하여 학습하였다. 또한 RNN과

14 RNN과 LSTM을 이용한 주가 예측을 향상을 위한 딥러닝 모델

LSTM의 활성화 함수를 Tanh와 ReLU를 이용하여 예측을 비교하였다. DNN, RNN, LSTM의 학습률, 가중치 초기값 등 하이퍼 파라미터 값들은 모두 동일하게 사용하였다. 표 3에 종목 5개에 대한 DNN, RNN, LSTM 실험결과에 대해서 나타냈다.

표 3. 실험 결과
Table 3. Experimental results

Item	Section	DNN (3 layer)	RNN tanh	RNN Relu	LSTM tanh	LSTM Relu
A	Avg. RSE	0.5068	0.2161	0.1682	0.1966	0.135
	Avg. RMSE	0.0587	0.0385	0.034	0.0373	0.0331
	Min. RMSE	0.0414	0.0335	0.0324	0.0347	0.0322
B	Avg. RSE	0.0977	0.0633	0.064	0.0645	0.0578
	Avg. RMSE	0.0898	0.0746	0.0683	0.0697	0.0653
	Min. RMSE	0.0776	0.0696	0.0668	0.0658	0.0646
C	Avg. RSE	0.0821	0.0615	0.0659	0.0535	0.0549
	Avg. RMSE	0.067	0.0579	0.0581	0.0544	0.0551
	Min. RMSE	0.0621	0.0559	0.0568	0.054	0.0547
D	Avg. RSE	0.182	0.1333	0.13	0.1323	0.1046
	Avg. RMSE	0.066	0.0557	0.0549	0.0558	0.0522
	Min. RMSE	0.061	0.0548	0.0526	0.0529	0.0498
E	Avg. RSE	0.0335	0.0357	0.0345	0.0342	0.0326
	Avg. RMSE	0.0938	0.0971	0.0954	0.0953	0.0928
	Min. RMSE	0.0873	0.094	0.093	0.0933	0.0909

표 4. DNN 대비 RNN의 오차율 향상
Table 4. Error rate improvement of RNN compared to DNN

Item	RNN tanh	RNN Relu	LSTM tanh	LSTM Relu	Avg.
A	34.41%	42.08%	36.46%	43.61%	39.14%
B	16.93%	23.94%	22.38%	27.28%	22.63%
C	13.58%	13.28%	18.81%	17.76%	15.86%
D	15.61%	16.82%	15.45%	20.91%	17.20%
E	-3.52%	-1.71%	-1.60%	1.07%	-1.44%
Avg.	15.40%	18.88%	18.30%	22.13%	18.68%

A 종목은 DNN의 평균 RMSE는 0.0587이며, RNN의 평균 RMSE는 0.0385과 0.034이다. 또한, LSTM의 평균 RMSE는 0.0373과 0.0331이다. B 종목은 DNN의 평균 RMSE는 0.0898이며, RNN의 평균 RMSE는 0.0746과 0.0683이다. 또한, LSTM의 평균 RMSE는 0.0697과 0.0653이다. C종목은 DNN의 평균 RMSE는 0.0621이며, RNN의 평균 RMSE는 0.0579와 0.0581이다. 또한, LSTM의 평균 RMSE는 0.054와 0.0547이다. D종목은 DNN의 평균 RMSE는 0.066이며, RNN의 평균 RMSE는 0.0557과 0.0549이다. 또한, LSTM의 평균 RMSE는 0.0558과 0.0522이다. E종목은 DNN의 평균 RMSE는 0.0938이며, RNN의 평균 RMSE는 0.0971과 0.0954이다. 또한, LSTM의 평균 RMSE는 0.0953과 0.0928이다.

그림 5에 5개 종목의 평균 RMSE에 대한 그래프를 나타냈다. 표 4에 DNN 대비 RNN, LSTM 의 예측 오차율 향상에 대해서 나타냈다.

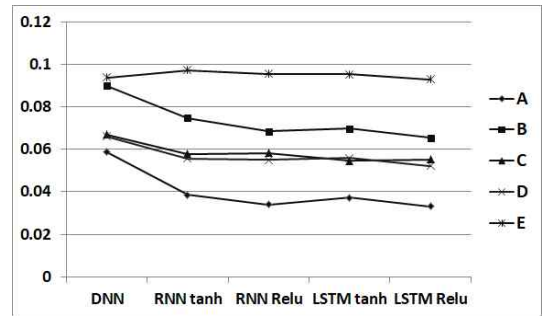


그림 5. 평균 RMSE 그래프
Fig. 5. Average RMSE graph

향상율은 다음과 같이 계산하였다.

$$ERI = \frac{RNN\text{평균 오차} - DNN\text{평균 오차}}{DNN\text{평균 오차}} \times 100 \quad (3)$$

DNN에 비해 시계열 딥러닝 네트워크인 RNN과 LSTM이 A 종목은 39.14 %, B 종목은 22.63%, C 종목은 15.86%, D 종목은 17.20%의 예측을 향상을 보였다. 그러나 E 종목은 -1.44% 예측율이 오히려 약간 감소되었다. 각 종목별로 차이는 있으나 약 15% 이상의 예측을 향상을 보였다. 그림 6에 각 딥러닝 모델에 대한 오차 그래프에 대해 나타냈다.

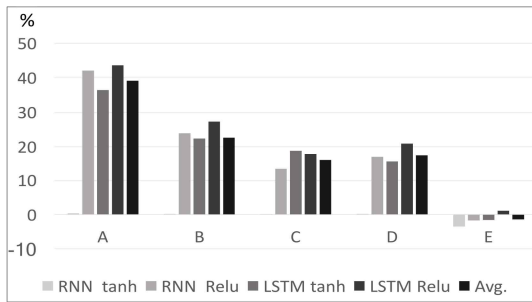


그림 6. 오차율 그래프

Fig. 6. Error rate graph

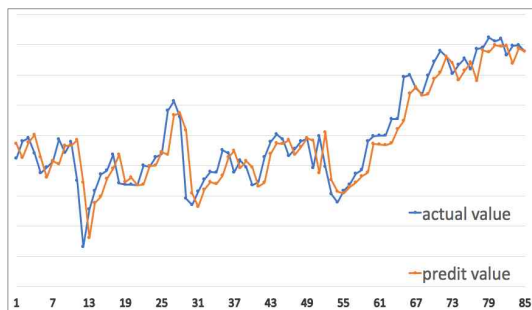


그림 7. 예측 값과 실제 값의 그래프

Fig. 7. Graph of prediction and actual values

그림 7은 A 종목에 대해서 LSTM 딥러닝 모델에 대한 텐서플로우의 실행결과 및 예측 값과 실제 값에 대한 그래프이다.

RNN과 LSTM을 실험할 때에는 총 3000번을 학습하였고, DNN을 실험할 때에는 총 20000번을 학습하였다. 또한, RNN과 LSTM의 최소 오차율과 최대 오차율이 간의 차이가 DNN의 최소 오차율과 최대 오차율의 차이 보다 작았다. 즉, RNN과 LSTM이 DNN 보다 빠르고 안정적으로 학습하였다. 원본 데이터의 주가 등락이 심할수록 RNN과 LSTM이 DNN보다 예측오차가 향상되고, 주가 등락이 적으면 RNN 및 LSTM과 DNN 간에 예측오차가 크게 차이가 없었다.

V. 결 론

본 연구는 R의 신경망 패키지를 통해, 기존의 보조지표에 대해서 예측요소로 적합 여부와 최적의 조합을 구축하고, 환율, 환율 이동평균, 전산업생산지수 등 예측률 향상을 위한 환경적 예측요소들에

대해서 연구하였다. 연구 결과 주가 예측에 사용되 는 대부분의 보조지표들이 오히려 예측율을 감소시키고, 환율, 환율 이동평균, 전 산업 생산지수 등 환경 예측요소를 사용할 때, 예측율이 상승하는 현상을 보여주었다. 따라서 주가예측에 대한 오차율을 줄이기 위해서는 보다 다양한 실험을 통해 적절한 예측요소들을 선택해야 할 것이다. 또한, 추출된 예측요소는 DNN, RNN, LSTM 등의 입력력으로 하는 딥러닝 모델을 제안하였다. 실험결과 DNN에 비해 시계열 딥러닝 네트워크인 RNN과 LSTM이 빠르고 안정적으로 학습하였으며, 종목별로 차이는 있으나 대략 15% 정도의 예측율 향상을 보였다. 또한, 주가 등락이 심할수록 RNN과 LSTM가 DNN보다 예측오차가 향상되었고, 주가 등락이 적으면 예측오차가 크게 차이가 없었다.

References

- [1] E. J. Lee, C. H. Min, and T. S. Kim, "Development of the KOSPI (Korea Composite Stock Price Index) forecast model using neural network and statistical methods", The Institute of Electronics Engineers of Korea, Computer and Information, Vol 45, No. 5, pp. 95-101, Sep. 2008.
- [2] K. Y. Kim and K. R. Lee, "A Study on the Prediction of Stock Price Using Artificial Intelligence System", Korean Journal of Business Administration, Vol. 21, No. 6, pp. 2421-2449, Dec. 2008.
- [3] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market prediction", FAU Discussion Papers in Economics, No. 11, pp. 310-342, May 2017.
- [4] J. W. Lee, "A Stock Trading System based on Supervised Learning of Highly Volatile Stock Price Patterns", Journal of Korean Institute of Information Scientists and Engineers, Vol 19, No. 1, pp. 23-29, Jan. 2013.
- [5] C. Hsu, "A hybrid procedure for stock price prediction by integrating self-organizing map and

- genetic programming", Expert Systems with Applications, Vol. 38, No. 11, pp. 14026-14036. Oct. 2011.
- [6] Y. K. Kwon, S. S. Choi, and B. R. Moon, "Stock prediction based on financial correlation", Proceedings of the 7th annual conference on Genetic and evolutionary computation. ACM, pp. 2061-2066, Jun. 2005.
- [7] S. M. An, "Deep Learning Architectures and Applications", Korea intelligent information system society, intelligent information study, Vol. 22, No. 2, pp. 127-142, Jun. 2016.
- [8] M. Y. Woo, S. Y. Park, Y. W. Han, U. C. Park, "Stock Price Estimation Experiments using R Neural Nets", Proceedings of KIIT Summer Conference 2013, pp. 498-500. May 2013
- [9] Giles, C. Lee, S. Lawrence, and A. C. Tsoi, "Noisy time series prediction using recurrent neural networks and grammatical inference", Machine learning, Vol. 44, No. 1, pp. 161-183, Jul. 2001.
- [10] K. J. Jeong and J. S. Choi, "Deep Recurrent Neural Networks," Communications of the Korean Institute of Information Scientists and Engineers, Vol. 33, No. 8, pp. 39-43, Aug. 2015.
- [11] T. J. Hsieh, H. F. Hsiao, and W. C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm", Applied soft computing, Vol. 11. No. 2, pp. 2510-2525, Mar. 2011.
- [12] K. H. Park and H. J. Shin, "Stock Price Prediction Based on Time Series Network", Korean Management Science Review, Vol. 28, No. 1, pp. 53-60, Mar. 2011.
- [13] S. Jelena, M. Nijole, and M. Algirdas, "High-low Strategy of Portfolio Composition using Evolino RNN Ensembles", Inzinerine Ekonomika-Engineering Economics 2017, Vol. 28, No. 2, pp. 162-169, Apr. 2017.

- [14] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks", Journal of Machine Learning Research 3, pp. 115-143, Mar. 2002.
- [15] <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> [Accessed: Jul. 01. 2017]

저자소개

신 동 하 (Dong-Ha Shin)



2016년 2월 : 가천대학교
에너지IT학과(공학사)
2016년 3월 ~ 현재 : 가천대학교
IT융합공학과(석사 과정)
관심분야 : 딥러닝, 빅 데이터,
IOT, 로봇제어, 로봇 액추에이터

최 광 호 (Kwang-ho, Choi)



2002년 2월 : 숭실대학교
컴퓨터공학과 졸업
2004.06 ~ 2007.04 일본 SBI증권,
일본 Monex증권 HTS개발 PM
2013년 2월 : 인천대학교
컴퓨터공학 공학석사
2007년 ~ 현재 : (주)에이치티윈

대표

관심분야 : 증권, DMA, 딥러닝, 빅 데이터, AI

김 창 복 (Chang-Bok Kim)



1986년 2월 : 단국대학교
전자공학과(공학사)
1989년 2월 : 단국대학교
전자공학과(공학석사)
2008년 2월 : 인천대학교 컴퓨터
공학과(공학박사)
1994년 ~ 현재 : 가천대학교

IT대학 에너지 IT학과 교수

관심분야 : 데이터 마이닝, 딥러닝, 강화학습, 사물인터넷