

A Stock Price Prediction Based on Recurrent Convolution Neural Network with Weighted Loss Function

HyunJin Kim[†] · Yeon Sung Jung^{††}

ABSTRACT

This paper proposes the stock price prediction based on the artificial intelligence, where the model with recurrent convolution neural network (RCNN) layers is adopted. In the motivation of this prediction, long short-term memory model (LSTM)-based neural network can make the output of the time series prediction. On the other hand, the convolution neural network provides the data filtering, averaging, and augmentation. By combining the advantages mentioned above, the proposed technique predicts the estimated stock price of next day. In addition, in order to emphasize the recent time series, a custom weighted loss function is adopted. Moreover, stock data related to the stock price index are adopted to consider the market trends. In the experiments, the proposed stock price prediction reduces the test error by 3.19%, which is over other techniques by about 19%.

Keywords : Artificial Intelligence, Recurrent Convolution Neural Network, Stock Price Prediction, Weighted Loss Function

가중치 손실 함수를 가지는 순환 컨볼루션 신경망 기반 주가 예측

김 현 진[†] · 정 연 승^{††}

요 약

본 논문에서는 RCNN (recurrent convolution neural network) 계층 모델을 채택한 인공 지능에 기반을 둔 주가 예측을 제안한다. LSTM (long-term memory model) 기반 신경망은 시계열 데이터의 예측에 사용된다. 다른 한편, 컨볼루션 신경망은 데이터 필터링, 평균화 및 데이터 확장을 제공한다. 제안된 주가 예측에서는 위에서 언급 한 장점들을 RCNN 모델에서 결합하여 적용함으로써 다음날의 주가 증가를 예측한다. 그리고 최근의 시계열의 데이터를 강조하기 위해 커스텀 가중치 손실 함수가 채택되었다. 또한 시장의 상황을 반영하기 위해 주가 인덱스에 관련된 데이터를 입력으로 포함하였다. 제안된 주가 예측 방식은 실제 주가를 대상으로 한 실험에서 3.19%로 테스트 오차를 줄였으며, 다른 방법 보다 약 19%의 성능 향상을 거둘 수 있었다.

키워드 : 인공지능, 순환 컨볼루션 신경망, 주가 예측, 가중치 손실 함수

1. 서 론

인공지능은 데이터의 수집 및 저장 기술의 발달과 컴퓨터 계산 능력의 향상을 통해 다양한 연구 결과와 응용을 확대하고 있다[1]. 기존의 객체 분류나 이미지 분석과 같은 비시계열 데이터를 넘어서 시계열 데이터에 대한 연구가 근래 활발하게 연구되고 있다. 일상생활에서 다양하게 접할 수 있는 많은 데이터가 시계열 데이터이므로 4차 산업혁명과 관련된 인공지능의 응용처가 더 확대될 수 있다.

시계열 데이터 기반의 인공지능 응용에 있어서 적용될 수

있는 분야가 금융 관련된 미래의 예측이다[2]. 환율, 주가, 금리 예측의 경우 과거의 가격과 거래량 등의 값의 변화 경향을 바탕으로 인공지능을 응용하여 미래의 결과를 예측할 수 있다. 주가의 경우 거래량과 주가의 등락에 따른 패턴에 기반을 둔 경험적인 방법과, 금융 공학의 관점에서 수학적 모델을 세우고 이를 통해 주가를 예측하려는 시도가 다양하게 이루어져 왔다. 주식을 매매하는 행위는 이윤을 남기기 위한 행위이므로 매수 및 매도 세력들의 매매를 통해 가격이 결정된다. 이들 매매세력의 수는 한정적이므로 그들의 매매 형태 또한 한정적이다. 여러 가지 변수가 다양하게 존재하겠지만 가격을 결정하는 패턴이 무작위 형태는 아닐 수 있다는 것을 직관적으로 알 수 있다.

주가와 같은 시계열 데이터에 대해서 인공지능을 적용하기 위한 모델에 대한 연구가 활발하게 진행되고 있다[3, 4]. 특히 순환 신경망(Recurrent Neural Network: RNN)은 내부

* 이 연구는 2017년도 단국대학교 대학연구비(융합과제)의 지원으로 연구되었음.

† 정 회 원 : 단국대학교 전자전기공학부 부교수

†† 비 회 원 : 단국대학교 경영학부 부교수

Manuscript Received : August 2, 2018

First Revision : September 6, 2018

Accepted : October 1, 2018

* Corresponding Author : Yeon Sung Jung(jys1836@naver.com)

에 루프가 있어 과거 데이터가 미래에 영향을 주는 특징을 가지고 있다. 이미 음성인식, 언어 모델링, 번역의 순차적 데이터가 필요한 분야에서 효과를 거두고 있다. 특히 장기 의존성(Long-Term Dependency)을 해결한 LSTM은 다양한 분야에 적용되고 있다[3].

이와 더불어 컨볼루션 신경망(Convolution Neural Network)은 시신경 구조를 모방한 기술로 이미지 및 사물 인식이 널리 적용되고 있다[4, 5]. 특정 범위에 한정하여 필터링 및 풀링(pooling) 통해 해당 지역적(local) 특징을 추출하고, 이를 반복함으로써 해당 영상의 크기는 줄어들게 되며 전역적(global) 특징을 추출하게 된다.

본 논문에서는 LSTM의 시계열 데이터 분석의 장점과, 컨볼루션 신경망의 필터링과 풀링을 통한 지역적 특징과 전역적 특징의 추출의 장점을 결합한 RCNN 모델을 이용해 주가 예측을 시도하였다. 이와 더불어 몇 가지 아이디어를 적용하였다. 첫 번째, 미래 주가의 예측은 과거 시계열 데이터 집합보다는 근래의 시계열 데이터 집합에 더욱 큰 영향을 받을 수밖에 없다는 직관에 판단하여 커스텀 가중치 손실 함수를 제안하고 적용하였다. 두 번째로, 종목에 대한 값들과 그 종목이 속한 주가지수의 값들을 포함하고 신경망을 학습하도록 하였다. 실험에서는 삼성전자의 2010년 6월 24일부터 2018년 6월 7일까지의 데이터를 포함하여 신경망을 학습시키고 이후 10일간의 주가를 예측한 결과 3.19%의 테스트 오차 수준으로 성능이 향상되었다.

2. 배경 지식

2.1 인공지능을 이용한 주가 예측

주식은 주주가 주식회사에 출자한 일정한 지분과 이익을 나타내는 증권을 의미한다. 이를 통해 기업의 자산 분배, 이익 분배의 권리 참여를 소유자에게 부여하게 된다. 이러한 주식은 매매 시에 가격이 결정되며 이 가격이 바로 주가이다. 주가의 예측에는 기본적 분석과 기술적인 분석이 있다. 기본적 분석은 주식회사의 내재된 가치에 따른 분석 방법이며 기술적인 분석은 해당 주가의 가격과 거래량의 흐름에 따른 분석이다. 봉차트, 평균이동선등을 이용한 패턴 분석이나 다우 이론, 엘리어트 파동이론과 같은 비수학적인 고전적 방법부터 수학적 모델을 통해 예측통계 모델을 산출하는 프로그램을 이용하는 금융공학 기반의 접근방법이 있다. 기술적인 분석 방법에서 후자의 방법의 경우 내재된 가치 데이터 기반으로 정교한 수학 모델을 개발 및 이용한다.

기존의 인공지능의 성공 사례를 볼 때, 인공지능을 이용한 주가 예측은 금융공학에 의한 복잡하고도 정밀한 수학적 모델을 세우는 것을 대신하여, 지능망의 구조를 기반으로 지능망의 파라미터를 학습을 통해 결정하는 것이라고 요약할 수 있다. 과거 주가의 데이터들은 주식 거래일에 따른 데이터베이스로 축적되어 있다. 지능망은 이 주가 데이터를 입력으로 예측한 주가와 실제 주가와 비교를 수행하고 이 차이로 인한 손실(loss)을 한 번의 배치(batch) 과정에서 계산하게 된다. 이 손실에 대해 신경망 구조의 역전파(back propagation)를

통해서 파라미터 값을 재조정하게 된다. 다음번의 배치에서 다시 손실을 구하고 역전파를 반복하며 손실을 줄이는 파라미터 값을 찾게 된다. 결론적으로 기존의 주가 데이터에 기반을 둔 최소의 손실을 가지는 파라미터 값을 찾게 되는 것이다.

인공지능을 통한 학습의 경우 지능망의 정해진 구조를 기반으로, 기존의 데이터를 이용한 학습을 통해 파라미터 값을 점차적으로 결정해 나가기 때문에 지능망의 구조를 어떻게 결정하는지가 성능에 매우 중요하다. 주가의 기술적 분석에서는 과거의 데이터의 흐름이 미래의 주가에 영향을 미친다고 생각하였다. 이에 착안하여 시계열의 데이터를 입력으로 순환 신경망을 이용한 주가 예측에 대한 연구들이 기존에 진행되어 왔다[6-8]. 또한 컨볼루션 신경망은 데이터의 확장과 필터링 및 평균화에 큰 장점을 가지고 있다. 순환 신경망과 컨볼루션 신경망에 대한 요약은 다음과 같다.

2.2 순환 신경망과 LSTM

순환 신경망은 입력을 가지고 결과를 만드는데 있어 과거의 데이터가 미래의 데이터 값에 영향을 줄 수 있는 순환 구조를 가지고 있다. 이러한 순환 구조를 풀면 Fig. 1의 리스트의 형태로 기본 구조가 정해진다. 여기서 X는 입력을 의미하며 h는 RNN의 하나의 셀 또는 내부적으로 연결된 셀들의 집합을 의미한다. d는 dense 레이어를 의미하며 완전 연결(fully-connected)된 레이어를 통해 출력 값이 도출되게 된다. RNN 셀은 입력 값을 받아 순차적으로 파라미터 값을 다음 RNN 셀로 전달하게 되며, 전달되는 값은 학습을 통해 조정되게 된다.

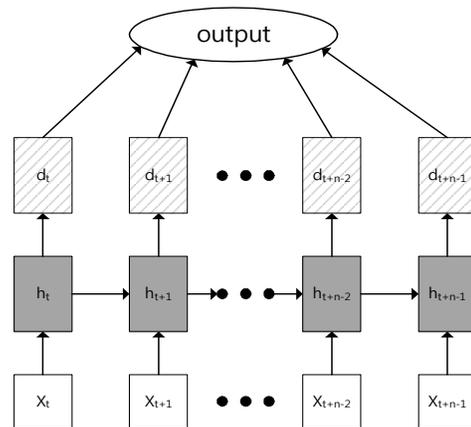


Fig. 1. Basic Structure of RNN

기본적인 RNN 셀은 과거의 값이 각 RNN 셀을 통해 전달되면서 과거의 값의 영향력이 약해지는 현상이 발생한다. 적절한 파라미터가 선택되지 않는다면 장기 의존성의 문제가 발생하게 되는데 이를 해결하는 셀 구조가 LSTM이다.

Fig. 2는 LSTM 셀의 구조를 설명하고 있다. 이는 LSTM의 입력과 출력에 대한 수식을 도식화 한 것이다. 기존의 RNN 셀과 가장 다른 특징은 C_{t-1} 의 입력이 C_t 로 시그모이드(σ)를 거치지 않고 전달될 수 있다는 점이다. 이를 통해 이전 상태의 값의 감쇄가 비교적 적게 일어나서 장기 의존성 문제를 해결할 수 있다. 이런 특징들로 인해서 LSTM 셀을 이용한 신경망

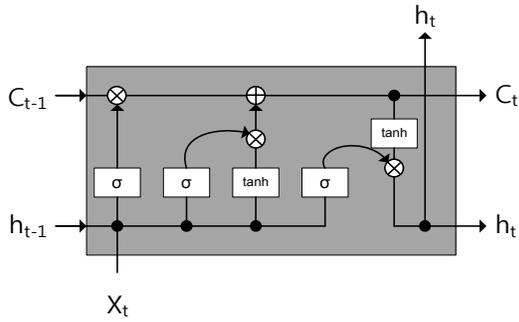


Fig. 2. Structure of LSTM Cell

기반의 연구가 많이 진행되고 있다[7-9]. 또한 [8]에서는 컨볼루션 신경망을 포함한 DNN(Deep Neural Network)과 이전 RNN 모델에 대비하여 LSTM 기반의 모델에서 더 나은 예측 결과를 도출하였다.

2.3 컨볼루션 신경망

컨볼루션 신경망은 사람의 시신경을 모방을 기반을 둔 신경망으로 특히 이미지 인식의 인공지능의 연구에 많이 활용되고 있다. 하나의 입력 데이터로부터 필터들과 컨볼루션 연산을 통해 여러 데이터를 생성한다. 이 데이터들을 풀링의 과정을 통해 더 적은 양의 데이터의 형태로 변환된다.

Fig. 3은 컨볼루션 레이어를 이용한 신경망의 한 예를 보여주고 있다. X는 입력 데이터이며 이는 5개의 필터를 가지는 conv1과 2: 1의 pooling1 레이어를 통해 정보의 양은 절반인 5개의 데이터 집합이 생성된다.

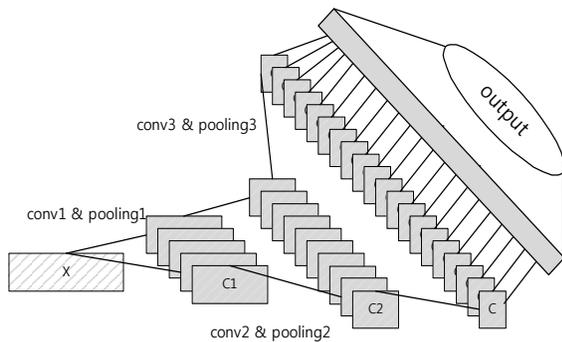


Fig. 3. Example Using Convolution Layers

그리고 conv2와 pooling2 레이어에서 정보의 양이 또 절반인 10개의 데이터 집합이 생성되며 마지막에서의 pooling3 레이어 출력은 dense 레이어를 통해 출력된다.

기존의 이미지 분류와 달리 컨볼루션 신경망은 학습의 과정을 통해서 필터 파라미터 값들의 계산을 자동화한다는 데 큰 의의가 있다. 기존의 이미지 분류 방식에서는 직관 및 수학적 방법으로 필터의 파라미터 값과 처리 과정을 결정했다. 신경망을 기반을 둔 이미지 분류에서는 학습을 통해 학습 데이터와 해당 신경망 구조에 최적인 파라미터 값을 찾는다. 당연하게도 이 파라미터 값들은 기존의 학습 데이터에 대한 손실을 최소화 하도록 정해지게 된다.

컨볼루션 신경망에서는 Fig. 3의 예와 같이 필터를 통한 해당 특징 추출과 데이터 확장(augmentation) 및 풀링을 통한 특징의 압축 등이 효과적으로 적용된다. 주로 이미지와 객체 인식에 적용되었으나, 컨볼루션 신경망의 효과를 고려할 때 다른 분야에 적용이 충분히 가능하다.

3. RCNN 기반의 제안된 신경망

3.1 순환 신경망과 컨볼루션 신경망의 결합

순환 신경망과 컨볼루션 신경망은 각각의 장점을 가지고 있다. 기존의 몇몇 응용분야에서 각각의 신경망 구조의 장점들을 결합하여 더 향상된 결과를 도출하기 위한 연구가 진행되었다[9]. 본 제안된 방법은 기존의 LSTM 셀 기반의 주가 예측보다 향상된 결과를 도출하기 위해 컨볼루션 신경망과 순환 신경망을 결합하였다. Fig. 4는 본 논문에서 주가 예측을 위해 결합된 신경망의 구조를 제시한다. 이 구조에서는 컨볼루션 레이어와 풀링 레이어는 각각 3단으로 구성되었다. 각 입력들은 매일의 주가나 거래량들이 포함된 벡터의 순차적 데이터로 구성되어 있으며, 이에 대한 컨볼루션은 벡터의 각 특징 값에 대해 1차원의 필터를 적용하게 된다. 만약 2010년 12월 1일부터의 데이터라면 2010년 12월 2일과 이후 데이터가 입력 데이터의 크기만큼 순차적으로 포함된다. 이에 따라 생성되는 데이터 또한 1차원의 순차적 데이터를 구성하게 된다. 풀링의 경우는 폭(stride)을 2로 유지하여 2개의 값을 평균하도록 한다. 이러한 과정을 3개의 컨볼루션 레이어와 풀링 레이어를 통해 반복하면서 순차적 데이터의 개수는 늘어나고 순차적 데이터내의 각 요소(element)의 평균값이 각 레이어의 출력으로 도출된다.

최종적으로의 컨볼루션 레이어를 거쳐 풀링 레이어의 출력은 LSTM 셀을 이용한 RNN 레이어에 입력되게 된다. Fig. 4에서는 풀링 레이어 각각 채널의 출력이 해당 각각의 RNN 레이어에 입력된다. RNN 만을 사용하는 경우에 비해 각 RNN 레이어에 필요한 LSTM 셀의 개수는 Fig. 4에서는 2개 밖에 되지 않는다. 크게 줄어든 RNN의 출력은 many-to-one 구조이며, 최종 LSTM 셀의 출력만을 취해 dense 레이어를 사용하여 주가를 도출하게 된다. 기존의 RNN 셀들만 이용하는 것과 달리 state의 개수는 크게 줄어들지만 RNN 레이어를 병렬로 다수를 사용한다. 이 병렬 구조에 의해서 시계열 데이터에서의 요소들의 관계가 모두 신경망에 반영될 수 있다.

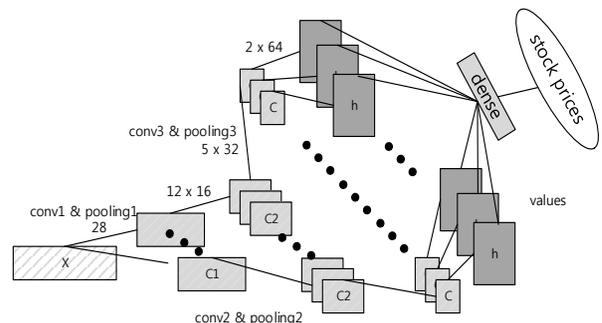


Fig. 4. Structure of RCNN in Proposed Method

본 주가 예측에서는 분류 문제가 아니라 회귀 값을 취하게 되므로 dense layer 이후에 softmax 레이어를 사용하지 않는다. 주가 분석에서 RCNN의 컨볼루션 레이어와 풀링 레이어는 예상치 못한 주가의 변화 흐름 반영 및 데이터 확장을 통한 전체적인 흐름이 반영되게 해준다. 또한 순차적 데이터 처리에 대한 RNN의 장점 또한 반영될 수 있다.

3.2 커스텀 가중치 손실 함수

순차적 데이터에 대해 근래와 가까운 데이터에 가중치를 줌으로써 성능을 높일 수 있다고 알려져 있다[10]. 본 연구에서는 정확한 주가 예측을 위해 근래와 가까운 주가의 예측에 대해 가중치 손실함수를 제시한다. 이 가중치 손실함수에서는 최종 주가 데이터의 날짜와 먼 과거일수록 가중치가 작아지고 가장 최근 데이터의 가중치는 1로 설정하였다.

$$W(x) = \min(W) + \frac{e^{\frac{x}{Y}}}{e^c} (1 - \min(W)) \quad (1)$$

Equation (1)에서는 날짜 x 에서 시작하는 경우에 사용된 가중치의 예를 보여주고 있다. 가장 먼 날짜에 관련된 데이터를 사용 시에 적용할 최소 가중치를 $\min(W)$ 라 정의하였다. 또한 상수 c 는 이 가중치 변화량을 조절하기 위한 파라미터이다. Y 는 전체 학습 데이터의 수를 의미한다. 가중치를 적용 시에 사용되는 손실함수는 Equation (2)와 같다. Equation (2)에서는 학습 시 하나의 배치(batch) 입력에 포함되는 각 입력 i 의 가중치 $W(i)$ 가 사용된다.

$$L = \sum((\text{실제 주가} - \text{예측된 주가}) \times W(i))^2 \quad (2)$$

3.3 입력 값 벡터에서의 주가 지수의 채택

한 종목의 주가를 예측하는 경우 대외적인 여러 가지 변수로 인해 종목의 주가를 정확히 예측하기 힘들다[7]. 본 연구에서는 단순히 해당 종목의 주가만을 학습에 사용하는 것이 아니라 종목이 속한 주가 지수를 적용하였다. 주가 지수는 전체적인 시장의 상황과 대외적인 여러 가지 변수의 영향을 반영한다고 할 수 있다. 해당 종목에서는 시가, 고가, 저가, 종가, 조정 후(adjusted) 종가, 거래량의 값들을 채택한다. 이와 마찬가지로 주가 지수에 대한 데이터도 위의 여섯 가지의 값들을 포함하여 전체 벡터를 구성하도록 한다.

4. 실험 결과 및 분석

4.1 실험 환경

본 논문에서는 Yahoo Finance [11]로부터 다운로드 받은 삼성전자와 KOSPI 데이터를 가지고 실험을 수행하였다. 2010년 6월 24일부터 2018년 6월 7일까지의 데이터를 포함하였다. 하루에 해당되는 하나의 벡터에는 하루의 시가, 고가, 저가, 종가, 조정 후 종가, 거래량의 6개의 값이 포함되어 있으며 삼성전자와 KOSPI 주가 데이터를 포함하는 경우에는 각각의 12개의 값을 포함하게 하였다. 이들은 각 종류의 최대 최솟값의 차로 나누어 정규화 하였다. 하나의 순차 데이터에는 28일간의 벡터

를 포함하게 하여 총 1929개의 순차 데이터를 가지고 학습을 수행하였다. 학습 결과에 대한 평가에서는 학습 데이터 이후 10일간의 주가를 역정규화를 통해 예측하고 실제 주가와 비교 테스트함으로써 분석에 필요한 정량적인 데이터를 확보하였다.

Table 1에 본 실험에서 사용된 라이브러리, 머신 및 신경망의 파라미터들에 대해서 나열하였다. Windows 10 64 bit과 Anaconda [14] 환경에서 Python 3.5으로 모든 코드는 작성되고 수행되었다. 주가 데이터는 Pandas [13]를 통해 읽어 들였으며, GPU를 지원하는 Tensorflow-gpu [12]를 사용하여 기존 및 제안된 신경망을 구현하였다. 또한 Table 1에 Fig. 4의 각각의 컨볼루션 레이어에 적용된 필터와 이에 따른 채널들의 수를 명시하였다. 풀링의 경우는 폭(stride)을 2로 유지하여 2개의 값을 평균하도록 하였으며 패딩의 경우는 valid로 설정하였다. 각각의 batch 입력의 크기는 200으로 설정하였으며 이에 대해서 500의 epoch까지 학습을 수행하였다. 각 epoch마다 테스트를 수행하여 예측한 주가를 추출하였다. 또한 손실 함수의 가중치 변화에 사용된 상수는 1로 설정하고 최소 가중치는 0.2로 설정하였다. LSTM 셀에서의 활성화(activation) 함수는 Softsign을 사용하였으며 AdamOptimizer 최적화 함수를 사용하였다.

Table 1. Experimental Environments

Library	Tensorflow-gpu 1.4.0rc1 [12], pandas [13]
CPU & RAM	Intel i7-5960x 3.00GHz & 32GB DDR4
GPU & VRAM	NVIDIA GeForce GTX 750 Ti & 2GB VRAM
RNN Hidden Dimension	20
RNN Forget Bias	1.0
RNN Stacked Layer	2
RNN Keep Probability	0.9
Learning Rate	0.005
Batch 크기	200
Epoch	500
가중치 변화(c)	1
최소 가중치	0.2

Table 2. Performance Comparisons

Model	Average Error	Max Error	Min Error	Error StDeV	Average Error Ratio
se	2,995	5,764	1,187	825	6.24
se_1c	1,643	3,981	598	541	3.42
se_1c_w	1,678	3,993	754	560	3.50
se_2c	1,827	3,803	880	482	3.81
se_2c_w	1,736	4,923	683	690	3.62
se_3c	1,929	4,353	704	746	4.02
se_3c_w	2,072	4,694	855	802	4.32
se_ks	3,081	5,837	1,585	798	6.42
se_ks_1c	1,667	3,365	416	612	3.48
se_ks_1c_w	1,608	4,136	519	570	3.35
se_ks_2c	2,013	4,008	711	604	4.20
se_ks_2c_w	1,771	3,814	632	667	3.69
se_ks_3c	1,659	3,674	642	642	3.46
se_ks_3c_w	1,530	4,779	642	676	3.19

4.2 실험 결과 및 분석

Table 2는 위의 실험 환경을 바탕으로 제안된 RCNN 구조에서 주가를 예측한 결과를 비교한 표이다. 실험 결과에서는 후반 100개의 epoch마다 테스트를 수행한 결과를 제시하였다. 여기에서 좌측의 se, ks, c, w 인덱스는 각각 삼성전자 주가 데이터, KOSPI의 주가 데이터, 컨볼루션 및 풀링 레이어, 가중치의 데이터를 택한 것을 의미한다. 본 논문에서 제안된 방법은 se_ks_3c_w로 이는 삼성전자 (se)와 KOSPI (ks) 데이터를 이용하고, 3단의 컨볼루션 레이어 (3c)를 채용하였으며, 가중치 손실함수 (w)를 택한 것을 의미하게 된다. 평균오류는 실제 주가 대비하여 예측한 주가 데이터의 오류값을 100개의 epoch에 대해 측정된 평균값이다. 최대오류와 최소오류는 모든 테스트 날짜 대상으로 예측한 값과 실제 값의 최대 및 최소 차이 값이다.

위의 결과를 보면 제안된 방법은 평균오류는 1,530원으로 대략 주가대비 3.19%의 평균오류 비율을 보였다. 최대오류는 4,779이고 최소오류는 642였다. 다른 입력 데이터 및 다른 신경망 구조들에 대비하여 1,551원~73원의 평균오류의 향상된 결과를 얻을 수 있었으며 이는 평균 19% 정도의 성능향상을

가져왔다. 또한 종목의 주가만을 사용한 경우보다, KOSPI의 지수를 같이 사용한 경우 542원의 평균오류 감소를 가져왔으며 가중치 손실 함수를 사용한 경우 129원의 평균오류의 감소를 얻을 수 있었다.

Fig. 5는 epoch에 따른 오류의 그래프이며 기존의 RNN 기반의 se와 제안된 방식을 비교하였다. 이 그림에서 보면 오류가 크게 발생하는 epoch이 존재하고 있다. 어느 정도의 epoch이 가장 좋은가에 대해서 epoch이 약 300번 이상인 경우부터 큰 값의 오류가 발생하는 경우가 줄어들게 되므로 이 추이를 보고 판단할 수 있을 것으로 분석된다. Fig. 6은 RCNN 방법을 적용함에 있어서 주가 인덱스 값을 적용하지 않은 경우와 가중치 손실함수를 적용하지 않은 경우의 epoch에 따른 오류의 그래프이다. Table 2의 값과 제안된 방법에서 오류가 적은 것을 고려할 때 주가 인덱스 데이터와 가중치 손실함수가 효과적임을 알 수 있다.

Table 3에서 학습 시간에 따른 비교를 제시하였다. 해당 환경에서의 se와 se_ks의 수행시간이 5분 31초와 6분 18초였다. se_ks_1c_w 및 se_ks_2c_w의 경우 각각 32분 4초와 33분 42초의 시간이 걸렸다. 제안된 방법을 수행하는 경우 30분

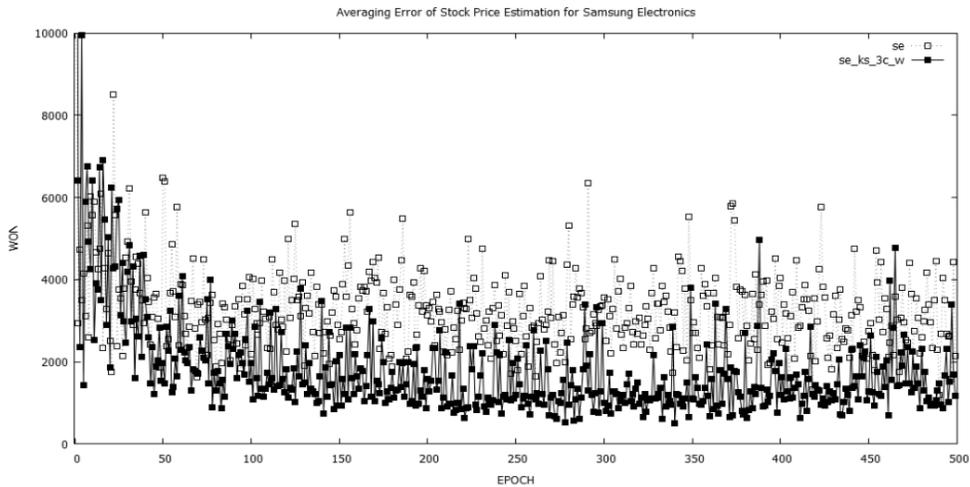


Fig. 5. Averaging Errors According to Epoch

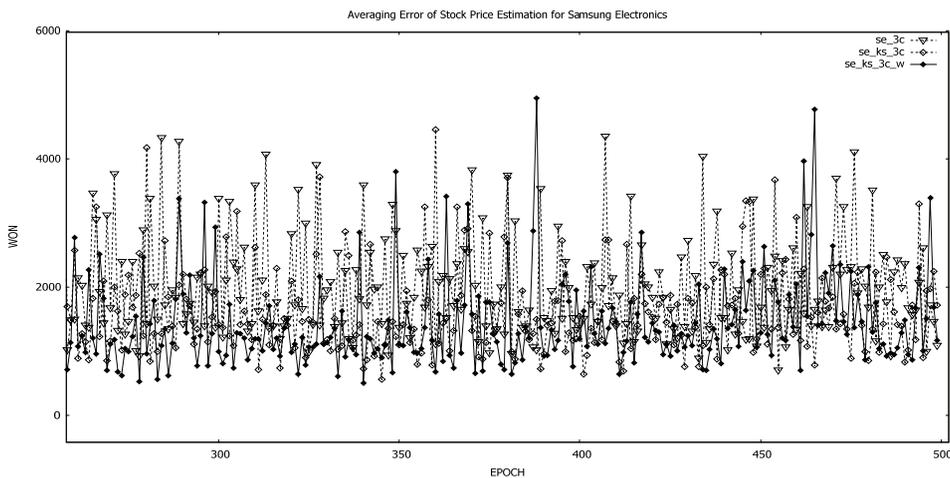


Fig. 6. Averaging Errors Compared to Other RCNN Approaches

35초의 시간이 걸리게 된다. 레이어의 단을 거치면서 데이터가 압축됨으로써 수행 시간이 감소하게 된다. RNN 구조 대비 학습 시간은 RNN 구조만을 사용한 경우보다 주가 예측의 성능은 크게 향상되나 5배~6배의 시간이 걸리게 된다. 이는 컨볼루션 레이어와 풀링 레이어를 거치면서 필요한 연산량의 증가 때문이다.

Table 3. Comparison in Terms of Training Time

Model	Elapsed Time
se	5m 31s
se_1c	6m 18s
se_1c_w	32m 4s
se_2c	33m 42s
se_2c_w	32m 20s
se_3c	29m 31s
se_3c_w	30m 4s
se_ks	6m 18s
se_ks_1c	33m 15s
se_ks_1c_w	33m 42s
se_ks_2c	32m
se_ks_2c_w	32m 4s
se_ks_3c	30m 43s
se_ks_3c_w	30m 35s

5. 결 론

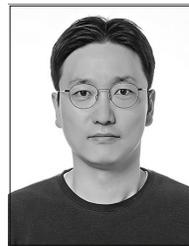
본 논문에서는 RCNN 계층 모델을 채택한 인공 지능에 기반을 둔 주가 예측을 제안 및 구현하였다. 제안된 방법은 순환 신경망과 컨볼루션의 신경망을 결합한다. 그리고 최근의 데이터의 영향을 강조하기 위해 커스텀 가중치 손실 함수가 채택하고 시장의 상황을 반영하기 위해 주가 인덱스에 관련된 데이터를 채택하였다. 본 제안으로 실험한 결과 3.19%의 예측 오차를 보였으며 다른 방법들 대비 약 19%의 성능향상이 있었다. 이를 고려할 때 제안된 방법은 주가를 예측하는 의미 있는 결과를 도출하였다고 할 수 있다.

References

[1] S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach," 3th ed., New York: Prentice Hall, 2009.
 [2] Yu Jin Oh and Yu Seop Kim, "Artificial intelligence : A two-Phase hybrid stock price forecasting model: cointegration tests and artificial neural networks," *The KIPS Transactions: Part B*, Vol.14, No.7, pp.531-540, 2007.
 [3] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning*, pp.2342-2350, 2015.
 [4] Hun Kim., Lecture of Machine Learning and Deep Learning [Internet], <https://hunkim.github.io/ml/>.
 [5] Convolution network [Internet], <https://deeplearning4j.org>.
 [6] Israt Jahan and Sayeed Sajal, Stock Price Prediction using

Recurrent Neural Network (RNN) Algorithm on Time-Series Data [internet], <https://www.micsymposium.org>.

[7] Dong-Ha Shin, Kwang-Ho Choi, and Chang-Bok Kim, "Deep learning model for prediction rate improvement of stock price Using RNN and LSTM," *Journal of KIIT*, Vol.15, No.10, pp.9-16, 2017.
 [8] Qun Zhuge, Lingyu Xu, and Gaowei Zhang, "LSTM Neural Network with Emotional Analysis for Prediction of Stock Price," *Engineering Letters*, Vol.25, No.2, 2017.
 [9] Baoguang Shi, Xiang Bai, and Cong Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vo.39, No.11, pp.2298-2304, 2017.
 [10] Jaeho Chang and Jechang Jeong, "Topic mining using time axis weighting," in *Proceedings of the Korea Data Mining Society Spring Conference*, 2012.
 [11] Yahoo finance [Internet], <https://in.finance.yahoo.com/>.
 [12] Tensorflow [Internet], <https://www.tensorflow.org>.
 [13] Pandas [Internet], <https://pandas.pydata.org/>.
 [14] Anaconda [Internet], <https://www.anaconda.com/download>.
 [15] Predict Stock Prices Using RNN [Internet], <https://lilianweng.github.io/lil-log/2017/07/08/predict-stock-prices-using-RNN-part-1.html>.



김 현 진

<http://orcid.org/0000-0001-5017-3995>
 e-mail : hyunjin2.kim@gmail.com
 1997년 연세대학교 전기공학과(학사)
 1999년 연세대학교 전기공학과(석사)
 2002년~2005년 삼성전기 ASIC LAB
 선임연구원

2010년 연세대학교 전기전자공학과(Ph.D)
 2010년~2011년 삼성전자 메모리사업부 책임연구원
 2011년~현 재 단국대학교 전자전기공학부 부교수
 관심분야 : 인공지능 하드웨어 및 시스템, 패턴 매칭 엔진 개발



정 연 승

<https://orcid.org/0000-0002-7256-2083>
 e-mail : jys1836@naver.com
 1993년 서울대학교 경영학과(학사)
 1998년 서울대학교 경영학과(석사)
 1998년~1999년 한국장기신용은행 주임
 2000년~2002년 삼성경제연구소 선임연구원

2003년~2006년 현대자동차 기획실 과장
 2007년 연세대학교 경영학과(박사)
 2007년~2010년 이노션월드와이드 부장
 2011년~현 재 단국대학교 경영학부 부교수
 관심분야 : 마케팅전략, 유통, 세일즈, 서비스전략