

# 텐서플로우를 이용한 주가 예측에서 가격-기반 입력 피처의 예측 성능 평가

## (Performance Evaluation of Price-based Input Features in Stock Price Prediction using Tensorflow)

송 유 정 <sup>†</sup>      이 재 원 <sup>††</sup>      이 종 우 <sup>†††</sup>  
(Yoojeong Song)      (Jae Won Lee)      (Jongwoo Lee)

**요 약** 과거부터 현재까지 주식시장에 대한 주가 변동 예측은 풀리지 않는 난제이다. 주가를 과학적으로 예측하기 위해 다양한 시도 및 연구들이 있어왔지만, 아직까지 정확한 미래를 예측하는 것은 불가능하다. 하지만, 주가 예측은 경제, 수학, 물리 그리고 전산학 등 여러 관련 분야에서 오랜 관심의 대상이 되어왔다. 본 논문에서는 최근 각광 받고 있는 딥러닝(Deep-Learning)을 이용하여 주가의 변동패턴을 학습하고 미래를 예측하고자한다. 본 연구에서는 오픈소스 딥러닝 프레임워크인 텐서플로우를 이용하여 총 3가지 학습 모델을 제시하였으며, 각 학습모델은 각기 다른 입력 피처들을 받아들여 학습을 진행한다. 입력 피처는 이전 연구에서 사용한 단순 가격 데이터를 확장해 입력 피처 개수를 증가시켜가며 실험을 하였다. 세 가지 예측 모델의 학습 성능을 측정했으며, 이를 통해 가격-기반 입력 피처에 따라 달라지는 예측 모델의 성능 변화 비교 분석하여 가격-기반 입력 피처가 주가예측에 미치는 영향을 평가하였다.

**키워드:** 주가 예측, 가격-기반 입력 피처, 텐서플로우, 기술적 분석, 인공지능, 딥러닝

**Abstract** The stock price prediction for stock markets remains an unsolved problem. Although there have been various overtures and studies to predict the price of stocks scientifically, it is impossible to predict the future precisely. However, stock price predictions have been a subject of interest in a variety of related fields such as economics, mathematics, physics, and computer science. In this paper, we will study fluctuation patterns of stock prices and predict future trends using the Deep learning. Therefore, this study presents the three deep learning models using Tensorflow, an open source framework in which each learning model accepts different input features. We expand the previous study that used simple price data. We measured the performance of three predictive models increasing the number of priced-based input features. Through this experiment, we measured the performance change of the predictive model depending on the price-based input features. Finally, we compared and analyzed the experiment result to evaluate the impact of the price-based input features in stock price prediction.

**Keywords:** stock prediction, price-based input features, tensorflow, technical analysis, artificial intelligence, deep learning

· 이 논문은 2017 한국컴퓨터종합학술대회에서 '텐서플로우를 이용한 주가 변동 예측 학습모델 설계 및 개발'의 제목으로 발표된 논문을 확장한 것임

<sup>†</sup> 학생회원 : 숙명여자대학교 IT공학과  
yjsong@sookmyung.ac.kr

<sup>††</sup> 정 회 원 : 성신여자대학교 컴퓨터공학과 교수  
jwlee@sungshin.ac.kr

<sup>†††</sup> 종신회원 : 숙명여자대학교 IT공학과 교수  
(Sookmyung Women's Univ.)  
bigrain@sookmyung.ac.kr  
(Corresponding author)

논문접수 : 2017년 8월 2일

(Received 2 August 2017)

심사완료 : 2017년 9월 18일

(Accepted 18 September 2017)

Copyright©2017 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.  
정보과학회 컴퓨팅의 실제 논문지 제23권 제11호(2017. 11)

## 1. 서론

과거부터 현재까지 주식시장에 대한 주가 변동 예측은 풀리지 않는 난제이다. 주가를 과학적으로 예측하기 위해 다양한 시도 및 연구들이 있어왔지만, 아직까지 정확한 미래를 예측하는 것은 불가능하다. 하지만, 주가 예측은 경제, 수학, 물리 그리고 전산학 등 여러 관련 분야에서 오랜 관심의 대상이 되어왔다[1]. 주가를 예측한다는 것은 주식 투자자들이 지속적으로 높은 수익을 얻을 수 있게 되는 것을 의미한다. 이 때문에 많은 사람들은 다양한 정보를 바탕으로 정확한 주식매매 의사결정을 해야 한다. 주식 종목에 대한 분석 방법으로는 기술적 분석(technical analysis)과 가치 분석(fundamental analysis)이 있다. 기술적 분석은 주식의 가격 및 거래량 등의 수치적 지표를 이용해 기술적 지표를 추출하고 이를 매매에 이용하는 분석을 하는 방법이며, 가치 분석은 회사 재무 지표를 이용해 회사의 미래를 전망하는 것이다[2]. 다양한 주가 변동 유발요인이 존재한다 해도 최대한 인간의 개입을 최소화 하여 오로지 차트의 모양, 주가가격의 추세에 따른 변동 패턴을 찾아낼 수 있다면, 정치/경제/사회적 사건등과 관계없이 주가의 변동을 예측할 수 있다[3]. 이를 위해 본 논문에서는 최근 매우 각광받고 있는 ‘딥러닝(Deep-Learning)’을 이용하여 주식의 변동 패턴을 학습하고 주가를 예측하고자 한다.

최근 패턴인식 성능의 비약적인 향상을 주도해온 데이터기반 딥러닝 기술은 음성인식, 영상 인식 등을 비롯한 다양한 패턴인식 분야의 성능 향상을 가져왔다[4]. 이처럼 패턴인식에 높은 성능을 보이고 있는 딥러닝 기술을 이용한다면, 높은 확률로 주가 변동을 예측할 수 있고, 이를 통해 투자를 진행하여 수익을 얻을 수 있을 것이다.

본 논문에서는 오픈소스 딥러닝 프레임워크인 텐서플로우(Tensorflow)[5]를 이용하여 주가 변동을 예측하는 딥러닝 학습 모델을 설계하고 구현하였다. 총 3가지 학습 모델을 제시하였으며, 각 학습모델은 각기 다른 입력피쳐(feature)를 받아들여 학습을 진행한다. 입력피쳐는 주가 분석의 기술적 분석 측면에서 접근하여 가격-기반 입력피쳐를 이용하였으며, 모든 학습 모델은 학습을 진행한 후, 학습되지 않은 데이터를 이용해 성능을 평가하고, 모의매매를 통한 적중률 측정까지 수행함으로써 예측 성능평가의 정확도를 높이고자 하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구에 대해 설명한다. 제 3장에서는 본 논문에서 제시한 3가지 딥러닝 학습 모델에 대한 설계 및 구현에 대한 내용을 설명한다. 제 4장에서는 실험 및 결과 분석 내용

에 대해 설명한다. 마지막으로 제 5장에서는 결론 및 향후 연구를 제시한다.

## 2. 관련 연구

본 장에서는 기존 기계학습 기반 주가 예측 연구에 대해 설명한다. 2.1절에서는 기존에 진행했던 단순 가격 데이터만을 이용한 딥러닝 학습모델 설계 및 개발 연구에 대해 설명한다. 2.2절에서는 기술적 분석 측면뿐만 아니라 사회, 경제적 사건 등을 고려한 딥러닝 기반 주가 예측 연구에 대해 설명한다.

### 2.1 단순 가격 데이터 딥러닝 학습 모델

이전 연구[3]에서는 삼성전자 주가 예측을 위해 5가지 간단한 주가 가격 데이터를 입력피쳐로 이용해 딥러닝 기반 학습 모델을 설계하고 구현하였으며, 학습 시 삼성전자 피쳐들만을 이용하였다. 주가의 경우 각 종목마다 가격대가 서로 많이 다르기 때문에 학습 시 고려할 사항이 많다. 따라서 기존 연구에서는 높은 성능을 내는데에 초점을 맞추기 위해 삼성전자 한 종목만의 주가 데이터를 이용해 실험을 진행하였다. 학습을 위한 입력 데이터는 일별 시가, 고가, 저가, 종가, 거래량 이렇게 5개의 단순 가격 데이터(simple price data)를 피쳐로 사용하였으며, 예측 타겟(레이블)은 5일 후 종가의 상승 또는 하락이다. 5일 후 종가가 상승 했을 시 1, 하락 시 0으로 예측 결과를 나타내며 약 2500개의 데이터로 학습 및 테스트를 진행하였다. 최종적으로 약 56%의 학습 성공률을 갖는 학습 모델이 도출되었다. 하지만 실험에 사용한 데이터의 수가 2000여개로 적고, 입력 피쳐가 너무 단순하여 데이터 확대 및 입력피쳐의 체계적인 고도화가 필요하다.

### 2.2 수치 및 텍스트 정보 이용 딥러닝 주가 예측 연구

기존의 딥러닝을 이용한 주가 예측 연구는 주로 뉴스 이벤트 분석 및 SNS 분석 등 텍스트 분석과 주가 가격 데이터 분석이 병행되는 형태로 이루어졌다.

첫 번째 관련 연구로는 Leemakdej, A.의 연구로[6] 태국 증권 시장을 대상으로 뉴스를 분석하고 준강형시장가설(Semi-strong form EMH)을 검증하기 위해 진행된 연구로, 공개적인 정보로 인해 시장의 가치가 변한다는 것을 검증하기 위해 텐서플로우를 사용하여 실험을 진행 하였다. Word2vec기법 중 Skip-gram[7]을 사용하여 뉴스의 키워드를 벡터화 시켜 학습에 사용하였으며 실험을 통해 뉴스 이벤트가 발생 후 오를 것이라고 예상 되는 종목에 대해 매수 후 5분/10분/30분/하루 전략을 이용해 매도하였다. 실험 결과 하루 전략, 5분 전략을 사용하였을 때 평균 10%의 수익을 얻을 수 있었다.

두 번째 관련 연구로는 AKITA, Ryo, et al.의 연구

[8]로 문단 벡터(paragraph vector)와 Long Short-Term Memory(LSTM) 모델[9]을 이용한 딥러닝 모델을 제안한 연구이다. 뉴스기사 안에 나타난 정보를 주가 등락과 관련된 문단 벡터 기법을 통해 분산 표상(distributed representation)[10]으로 표현하고, 표현된 벡터를 학습하기 위해 과거에 일어난 사건을 고려한 딥러닝 학습 모델인 LSTM을 이용하여 구현하였다. 최종적으로 수치 정보만을 이용한 주가 예측 방법이나 Word2vec 기법 중 BoW 방법[7]을 이용한 방법보다 텍스트 정보의 분산표상을 이용한 방법이 더 효과적이라는 결과를 도출하였다.

하지만 두 연구 모두 인간의 주관적인 의견이 반영된 뉴스기사 또는 SNS 데이터를 분석한 연구이다. 이는 실제 주식 시장에 변동이 있을 시 패턴발견에 방해 요소가 될 가능성이 있다. 따라서 본 논문에서는 인간의 개입을 최소화 하기 위해 기술적 접근 방식으로 주가 변동 예측을 하며, 주식 가격 데이터를 이용한 피쳐만을 가지고 연구를 진행한다.

### 3. 딥러닝 모델 설계 및 구현

본 장에서는 주가 예측 딥러닝 모델의 학습, 테스트, 검증에 사용된 입력피쳐 구성에 대해 설명한다. 또한 가격-기반 입력피쳐만으로도 주가 예측이 가능한지 알아보기 위해, 주가 데이터의 기간 및 매개변수는 동일하고 입력피쳐를 다르게 구성한 3개의 딥러닝 학습 모델의 네트워크 구조 및 매개변수 설정에 대해 설명한다.

#### 3.1 전체 데이터 구성

주가 예측을 위한 딥러닝 모델에 사용되는 데이터는 3가지 크게 학습데이터, 테스트데이터 그리고 검증데이터가 그것이다. 각 분류된 데이터에 대해 설명하면 다음과 같다.

학습에 사용한 데이터는 대한민국 대표 주식인 삼성전자와 이와 가격 변동 패턴이 매우 유사한 삼성전자우선주 데이터이다. 주가는 각 종목마다 가격대가 서로 많이 다르기 때문에 학습 시 고려할 사항이 많다. 따라서 본 논문에서는 삼성전자와 삼성전자우선주 두 종목만의 주가 데이터를 이용해 실험을 진행한다.

학습데이터는 총 5430개로 2005/1/3~2015/12/30 동안의 일별 주가 데이터로 구성하였다. 테스트데이터는 2016/1/4~2016/12/30 동안의 총 482개의 일별 데이터로 구성하였고, 검증데이터는 2017/1/2~2017/7/6 동안의 총 242개의 데이터로 구성하였다. 여기서, 검증데이터는 학습에 사용되지 않은 데이터로 이를 이용해 학습 결과가 미래에도 적용되는지에 대한 검증을 진행하고, 예측 성공 여부를 적중률(hit ratio)로 나타낸다. 데이터 구성에 대한 세부 내용은 표 1과 같다.

표 1 데이터 집합 구성 및 데이터의 수

Table 1 The composition of data set and the number of data set

Data set	Period	Number of Data
Training data	2005/1/3~2015/12/30	5430
Test data	2016/1/4~2016/12/30	482
Validation data	2017/1/2~2017/7/6	242

#### 3.2 가격-기반 입력 피쳐 구성

주가를 예측하기 위한 학습모델은 다양한 피쳐를 이용해 예측하도록 설정할 수 있으며, 본 논문에서는 이 입력 피쳐를 얼마나 적절하게 사용하느냐에 따라 예측 성능이 달라질 것으로 예상하였다. 이에 따라 본 논문에서는 학습피쳐에 사회적 현상 또는 사건은 반영하지 않고, 오로지 주식 가격 피쳐들만 포함하여 가격-기반 입력 피쳐에 따른 예측 성능을 평가한다.

가격-기반 입력 피쳐는 이전 연구[3]에서 사용한 단순 가격 데이터를 기반으로 종가의 이동평균과 거래량의 이동평균을 각각 5일/10일/20일/60일/120일로 구분하여 추가하였다. 이동 평균(moving average)이란 일정 기간 동안의 주가 평균을 구한 것으로 주가의 추세를 요약하기 위한 피쳐이다. 종가의 이동 평균은 MA로 표기하며 거래량(volume)의 이동평균은 VMA로 표기한다. 예로서, t 거래일에서의 종목 s의 종가는  $Close price_t^s$ 로 표현하며 종가 5일 이동평균은 아래와 같이 계산할 수 있다[11].

$$MA5_t^s = \frac{1}{5} \sum_{k=0}^4 Close price_{t-k}^s, \quad (1)$$

마찬가지로 t 거래일의 종목 s의 거래량 5일 이동평균은 아래와 같이 계산할 수 있고,  $Volume_t^s$ 은 t 거래일의 거래량이다.

$$VMA5_t^s = \frac{1}{5} \sum_{k=0}^4 Volume_{t-k}^s, \quad (2)$$

학습피쳐가 시가, 고가, 저가, 종가, 거래량으로 구성되었던 이전 연구에 더해서 이동평균을 추가한다면 이전보다 의미 있는 결과를 낼 것이라는 가정 하에 본 피쳐를 추가한 것이다. 실제로 Bollinger Band[12]라는 추세 분석 통계기법에서도 이동 평균 등의 피쳐를 사용한다. 표 2는 본 논문에서 학습 시 사용한 입력 피쳐를 나타낸다.

#### 3.3 가격-기반 딥러닝 모델의 구조

주가 예측을 위한 가격-기반 입력피쳐를 사용한 3가지 딥러닝 모델은 심층신경망(Deep Neural Network, DNN) 구조로 구현하였다. 입력을 받는 입력층, 신경망의 결과가 출력되는 출력층, 입력층과 출력층 사이에 외부로 나타나지 않는 은닉층 이렇게 3계층으로 구성하였

표 2 가격-기반 입력 피쳐 구성

Table 2 The composition of price-based input feature

Feature	Details of the feature
Simple price data	$Open\ price_t^s$
	$High\ price_t^s$
	$Low\ price_t^s$
	$Close\ price_t^s$
	$Volume_t^s$
Close price moving average (MA)	$MA5_t^s = \frac{1}{5} \sum_{k=0}^4 Close\ price_{t-k}^s$
	$MA10_t^s = \frac{1}{10} \sum_{k=0}^9 Close\ price_{t-k}^s$
	$MA20_t^s = \frac{1}{20} \sum_{k=0}^{19} Close\ price_{t-k}^s$
	$MA60_t^s = \frac{1}{60} \sum_{k=0}^{59} Close\ price_{t-k}^s$
	$MA120_t^s = \frac{1}{120} \sum_{k=0}^{119} Close\ price_{t-k}^s$
Volume moving average (VMA)	$VMA5_t^s = \frac{1}{5} \sum_{k=0}^4 Volume_{t-k}^s$
	$VMA10_t^s = \frac{1}{10} \sum_{k=0}^9 Volume_{t-k}^s$
	$VMA20_t^s = \frac{1}{20} \sum_{k=0}^{19} Volume_{t-k}^s$
	$VMA60_t^s = \frac{1}{60} \sum_{k=0}^{59} Volume_{t-k}^s$
	$VMA120_t^s = \frac{1}{120} \sum_{k=0}^{119} Volume_{t-k}^s$

는데, 두 개 이상의 은닉층이 있을 경우에 심층신경망이라고 표현한다[13]. 각 일별 주가 데이터에 대한 예측 목표표는 5일 후의 종가의 상승 또는 하락을 1또는 0으로 표현한 값을 설정하였다. 상승했을 경우 1이 되며, 하락했을 경우 0이 된다.

본 논문에서 제시하는 3가지 모델의 구조는 그림 1~3과 같다. 각각의 모델은 가격-기반 입력 피쳐만 다르게 구성하였기 때문에 입력층을 구성하는 뉴런의 수만 다르다. 3가지 모델 모두 7개의 은닉층을 가지고 있고, 하나의 층은 30개의 뉴런으로 구성되어 있다. 그림 1에서 제시한 모델은 시가, 고가, 저가, 종가, 거래량 5개의 입력피쳐만으로 학습을 한다. 그림 2에서 제시한 모델은 종가이동평균인 MA피쳐 5개를 추가하여 총 10개의 피쳐로 학습을 진행한다. 마지막으로 그림 3에서 제시한 모델은 여기에 더해 거래량 이동평균 VMA 5개 피쳐를 추가한 모델로 총 15가지의 입력피쳐로 학습을 진행한다.

#### 4. 실험 및 성능 평가

본 장에서는 설계된 딥러닝 모델을 구현하고 실험한

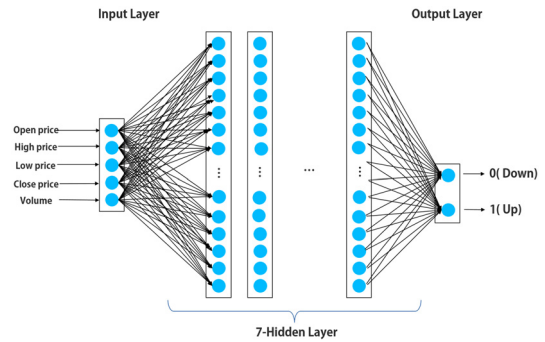


그림 1 단순 가격 피쳐 모델

Fig. 1 Simple price feature model

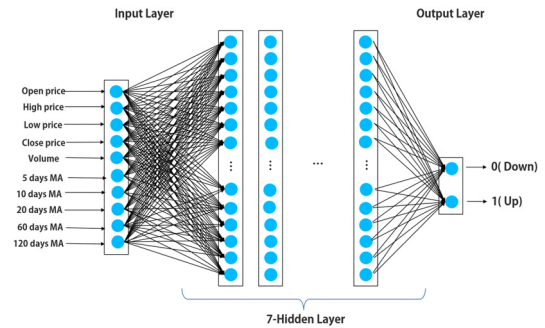


그림 2 종가이동평균 추가 모델

Fig. 2 MV added model

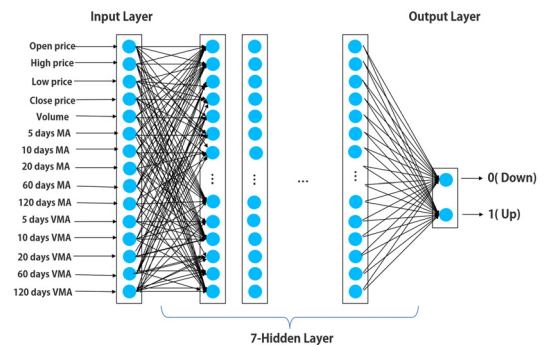


그림 3 거래량이동평균 피쳐 추가 모델

Fig. 3 VMV added model

환경 및 모델의 학습, 테스트, 검증의 결과에 대해 설명한다. 가격-기반 입력피쳐를 다르게 구성한 3가지 모델에 대해 오차 감소율과 예측 성능을 측정하였으며, 실제 학습되지 않은 최근 주가 데이터를 기반으로 검증을 수행하여 적중률을 도출하였다.

##### 4.1 실험 환경

딥러닝 프레임워크는 테아노(Theano)[14], 토치(Torch)

[15], 텐서플로우(Tensorflow)[5], 카페(Caffe)[16] 등 여러 가지가 있으며 이 중 텐서플로우는 다양한 하드웨어 환경에서 인공지능망 모델을 쉽게 생성하고 학습할 수 있는 인터페이스를 제공하여 해결하고자 하는 문제에 최적화된 모델 생성에 집중할 수 있다는 장점이 있다[3]. 따라서 본 연구에서는 텐서플로우를 이용하여 주식 데이터를 위한 딥러닝 학습모델을 구현하였다. 리눅스 우분투 16.04에 Tensorflow v.1.2를 설치한 환경을 기반으로 실험을 수행하였고, Jupyter notebook[17]과 Python[18]을 이용해 개발하였다.

#### 4.2 학습 및 테스트

세가지 학습 모델 모두 총 5430개의 학습 데이터와 482개의 테스트 데이터를 이용하여 실험을 진행하였으며, 심층신경망의 과적합(Overfitting)[19] 문제를 방지하기 위해 Dropout 기법[20]을 사용하였고, 최적화기는 AdamOptimizer를 사용하였다. 활성화 함수는 Sigmoid를 이용하였고, Learning rate는 0.01로 설정하였으며 100번의 반복학습을 수행하였다. 그림 4~6은 학습을 진행함에 따라 감소하는 오차감소율과 모델의 예측 성능을 출력해 그래프로 나타낸 그림이다. 세 가지 모델 모두 학습을 진행 할수록 점점 오차가 낮아지는 결과를 도출하였다. 100 에폭(epoch)까지만 반복한 이유는 더 이상 에러가 작아지지 않았기 때문이다. 최종적으로 단순 가격 피쳐를 이용한 모델은 52.69%의 예측 성능이 도출되었고, MA를 추가한 모델은 56.01%, 그리고 MA와 VMA를 추가한 모델은 56.84%의 예측 성능이 도출되었다.

```
Epoch: 000 , cost = 0.707000651
Epoch: 010 , cost = 0.692249612
Epoch: 020 , cost = 0.691786942
Epoch: 030 , cost = 0.691729088
Epoch: 040 , cost = 0.691651280
Epoch: 050 , cost = 0.691662805
Epoch: 060 , cost = 0.691666541
Epoch: 070 , cost = 0.691637004
Epoch: 080 , cost = 0.691586408
Epoch: 090 , cost = 0.691572687
Accuracy: 52.697098252
Optimizaiton finished.
```

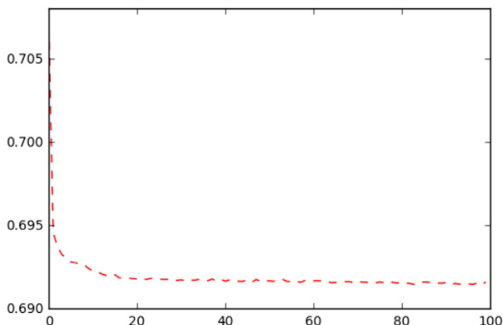


그림 4 단순 가격 피쳐 모델  
Fig. 4 Simple price feature model

```
Epoch: 000 , cost = 0.743656164
Epoch: 010 , cost = 0.693214192
Epoch: 020 , cost = 0.692655109
Epoch: 030 , cost = 0.692652222
Epoch: 040 , cost = 0.692153873
Epoch: 050 , cost = 0.691928840
Epoch: 060 , cost = 0.691788404
Epoch: 070 , cost = 0.691474867
Epoch: 080 , cost = 0.691427997
Epoch: 090 , cost = 0.691067320
Accuracy: 56.016600132
Optimizaiton finished.
```

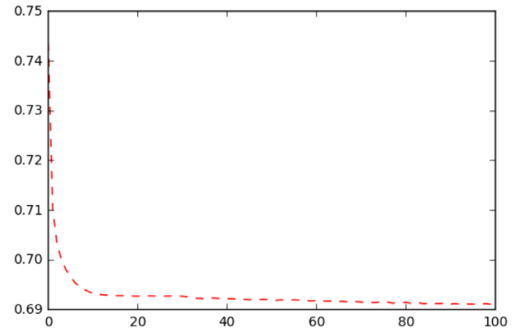


그림 5 종가이동평균 피쳐 추가 모델  
Fig. 5 MV added model

```
Epoch: 000 , cost = 0.774622457
Epoch: 010 , cost = 0.693963986
Epoch: 020 , cost = 0.690657238
Epoch: 030 , cost = 0.689867723
Epoch: 040 , cost = 0.689549271
Epoch: 050 , cost = 0.689350848
Epoch: 060 , cost = 0.689120691
Epoch: 070 , cost = 0.688795679
Epoch: 080 , cost = 0.688777970
Epoch: 090 , cost = 0.688627924
Accuracy: 56.8464756012
Optimizaiton finished.
```

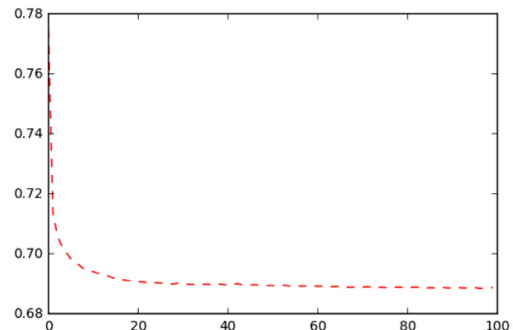


그림 6 거래량이동평균 피쳐 추가 모델  
Fig. 6 VMV added model

#### 4.3 검증

검증 시 2017년 1월부터 7월 6일까지의 242개의 최신 주가 데이터를 이용하였으며, 이 데이터는 학습되지 않은 데이터이다. 이 결과 종가의 이동평균을 추가한 10개의 입력 피쳐를 이용한 모델에서 가장 높은 55.73%의 적중률(Hit ratio)을 보였다. 5일 후 종가가 오를 경우 1, 내릴 경우를 0으로 설정하였기 때문에 출력 결과가 1과 1로 또는 0과 0으로 일치할 시 적중했다는 표현을

표 3 세 가지 모델의 비교

Table 3 The comparison of three models

	Simple price model	MV added model	VMA added model
The number of features	5	10	15
Dropout	0.5	0.5	0.5
Activation function	Sigmoid	Sigmoid	Sigmoid
Optimizer	Adam Optimizer	Adam Optimizer	Adam Optimizer
Weight init method	Xavier init	Xavier init	Xavier init
Accuracy	52.69%	56.01%	56.84%
Hit ratio	47.51%	51.01%	48.31%

쓰며 이를 이용하여 적중률을 계산하였다. 세 가지 모델 중 단순 가격 피처를 이용한 모델은 약 47.51%의 적중률이 도출되었고, 종가의 이동평균을 추가한 10개의 입력피처를 이용한 모델은 약 51.01%, 종가 이동평균과 거래량 이동평균을 추가한 15개의 입력 피처를 사용한 모델은 약 48.31%의 적중률이 도출되었다.

#### 4.4 가격-기반 입력피처 예측 성능 평가

표 4는 본 논문에서 제안한 가격-기반 입력 피처를 다르게 구성한 세 가지 딥러닝 모델에 대한 상세 매개변수 설정 및 성능, 적중률을 비교한 표이다. 입력 피처의 개수와 종류만 다르게 한 이 세 모델은 Dropout 사용여부와 확률도 동일하게 설정하였으며, 활성화 함수와 최적화기, 가중치 초기화 방법 또한 동일하게 설정했지만 성능과 적중률이 각기 다르게 도출되었다. 세 모델 모두 50% 이상의 학습단계 예측 성능을 보였으며 MA와 VMA를 모두 추가해 15개의 가격-기반 입력 피처를 이용한 모델이 가장 높은 성능인 56.84%를 보였다. MA와 VMA는 종가와 거래량의 이동평균으로 주가의 추세를 나타내는 피처로서 주가의 변동 패턴에서 추세를 추가함으로써 단순 가격 데이터만을 이용해 패턴을 예측한 모델보다 높은 성능을 도출할 수 있었다. 그러나 MA만을 추가한 10개의 피처를 사용한 모델과 MA와 VMA를 둘 다 추가한 15개의 피처를 사용한 모델도 적중률에서는 그다지 높은 성능을 보이지 못했다.

## 5. 결 론

본 논문에서는 가격-기반 입력피처가 예측성능에 미치는 영향을 비교 분석하기 위해 주식 종목의 단순 가격 데이터에 10가지 새로운 피처를 추가로 제시하였으며, 총 15가지 입력피처를 이용해 삼성전자 주가 예측을 위한 학습모델을 평가하였다. 본 연구의 결과 세 모델 모두 50%확률 이상의 예측 성능을 보였으며 최대 MA와 VMA를 모두 추가해 15개의 가격-기반 입력 피처를 이용한 모델이 가장 높은 성능인 56.84%를 보였다. MA와 VMA는 종가와 거래량의 이동평균으로 주가의 추세를 나타내는 피처로서 주가의 변동 패턴에서 추세를

를 추가함으로써 단순 가격 데이터만을 이용해 패턴을 예측한 모델보다 높은 성능을 도출할 수 있었다. 이는 주식 데이터에서 가격을 기반으로 구성된 입력피처가 달라지면 성능도 달라진다는 것을 의미하며, 보다 주식의 변동 패턴을 잘 파악할 수 있는 고도화된 입력피처를 사용한다면 주식의 변동 패턴을 더 정확한 확률로 예측할 수 있다는 것을 의미한다.

하지만 본 논문에서도 여전히 매우 간단한 구성의 입력 데이터와 간단한 구조의 네트워크를 사용했기 때문에 앞으로 더욱 고도화된 입력피처 구성 및 네트워크 구조를 적용하면 성능이 높은 딥러닝 모델로 발전할 가능성이 크다.

향후 연구로는 본 연구에서 개발 한 심층 신경망을 합성곱 신경망(Convolutional Neural Network, CNN)이나 순환신경망(Recurrent Neural Network, RNN) 등으로 확장 구현 하여 더 높은 성능의 주가 변동 예측 딥러닝 모델을 개발하고자 한다. 또한 조금 더 심화 된 기술적 접근 방식에 따른 다양한 피처들을 추가하여 실험을 진행할 예정이다. 또한 현재는 삼성전자에 특화된 학습 모델에 불과하나, 이를 더 일반화시켜 다른 종목의 주가도 예측 할 수 있을 것으로 예상된다.

## References

- [1] J. Lee, "A Stock Trading System based on Supervised Learning of Highly Volatile Stock Price Patterns," *Journal of KIISE : Computing Practices and Letters*, Vol. 19, No. 1, pp. 23-29, Jan. 2013. (in Korean)
- [2] S. Kim, "Data Mining Tool for Stock Investors' Decision Support," *The Journal of the Korea Contents Association*, Vol. 12, No. 2, pp. 472-482, Feb. 2012. (in Korean)
- [3] Y. Song and W. Lee, "A Design and Implementation of Deep Learning Model for Stock Prediction using TensorFlow," *Korea Computer Congress 2017*, pp. 799-801, 2017. (in Korean)
- [4] H. Choi, and Y. Min, "Intelligent Information System: Introduction to deep running and major issues," *Korea Information Processing Society Re-*

- view, Vol. 22. No. 1, pp. 7-21, 2015.
- [5] Tensorflow, [Online]. Available: <https://www.tensorflow.org/>
  - [6] L. Arnat, "Stock Price Prediction by Deep Learning," Dec. 2016.
  - [7] Word2vec, [Online]. Available: <https://en.wikipedia.org/wiki/Word2vec>
  - [8] A. Ryo et al., "Deep learning for stock prediction using numerical and textual information," *Computer and Information Science(ICIS), 2016 IEEE/ACIS 15th International Conference on. IEEE*, 2016.
  - [9] LSTM, [Online]. Available: [https://en.wikipedia.org/wiki/Long\\_short-term\\_memory](https://en.wikipedia.org/wiki/Long_short-term_memory)
  - [10] Le, Quoc, and Tomas Mikolov, "Distributed representations of sentences and documents," *Proc. of the 31st International Conference on Machine Learning (ICML-14)*, 2014.
  - [11] J. Lee, "A Stock Trading System based on Moving Average Patterns and Turning Point Matrix," *Journal of KIISE : Computing Practices and Letters*, Vol. 18, No. 7, pp. 528-532, Jul. 2012.
  - [12] Bollinger Band, Available: [https://en.wikipedia.org/wiki/Bollinger\\_Bands](https://en.wikipedia.org/wiki/Bollinger_Bands)
  - [13] J. Lee, "Stock price prediction model using deep learning," 2016.
  - [14] Theano, [Online]. Available: <http://deeplearning.net/software/theano/>
  - [15] Torch, [Online]. Available: <http://torch.ch/>
  - [16] Caffe, [Online]. Available: <http://caffe.berkeleyvision.org/>
  - [17] Jupyter notebook, [Online]. Available: <http://jupyter.org/>
  - [18] Python, [Online]. Available: <https://www.python.org/>
  - [19] Overfitting, [Online]. Available: <https://en.wikipedia.org/wiki/Overfitting>
  - [20] Dropout, [Online]. Available: [https://en.wikipedia.org/wiki/Dropout\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Dropout_(neural_networks))

이 종 우

정보과학회 컴퓨팅의 실제 논문지  
제 23 권 제 1 호 참조



송 유 정

2011년~2015년 숙명여자대학교 멀티미디어과 학과 학사. 2015년~현재 숙명여자대학교 IT공학과 석박사통합과정. 관심분야는 Artificial Intelligence, Embedded System Software, Mobile System Software



이 재 원

1990년 서울대학교 컴퓨터공학과(학사)  
1992년 서울대학교 컴퓨터공학과대학원(석사). 1998년 서울대학교 컴퓨터공학과 대학원(박사). 1999년~현재 성신여자대학교 IT학부 부교수. 관심분야는 Finance Engineering, Artificial Intelligence, Machine Learning, Natural Language Process

chine Learning, Natural Language Process