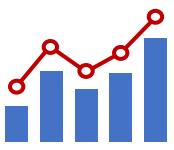


AI 펀드 매니저

201533794 유재형
201739417 박윤재
201739414 박기완

목차

- 
- 01 주제 선정 배경
 - 02 벤치마킹 및 차별성
 - 03 주제 및 요약
 - 04 프로그램 소개
 - 05 시나리오
 - 06 결론



주제 선정 배경

트럼프 담화 끝나자 실망매물… 8년 5개월만에 사이드카 발동
[코로나 팬데믹 코스피 3.9% 폭락]

파이낸셜뉴스 입력 2020.03.12 17:56 수정 2020.03.12 20:20



사이드카 : 프로그램 매매 호가 효력 일시 정지 제도.

선물 가격이 기준가 대비 ±5% 이상인 상황이 1분간 지속하는 경우 **발동된다**.



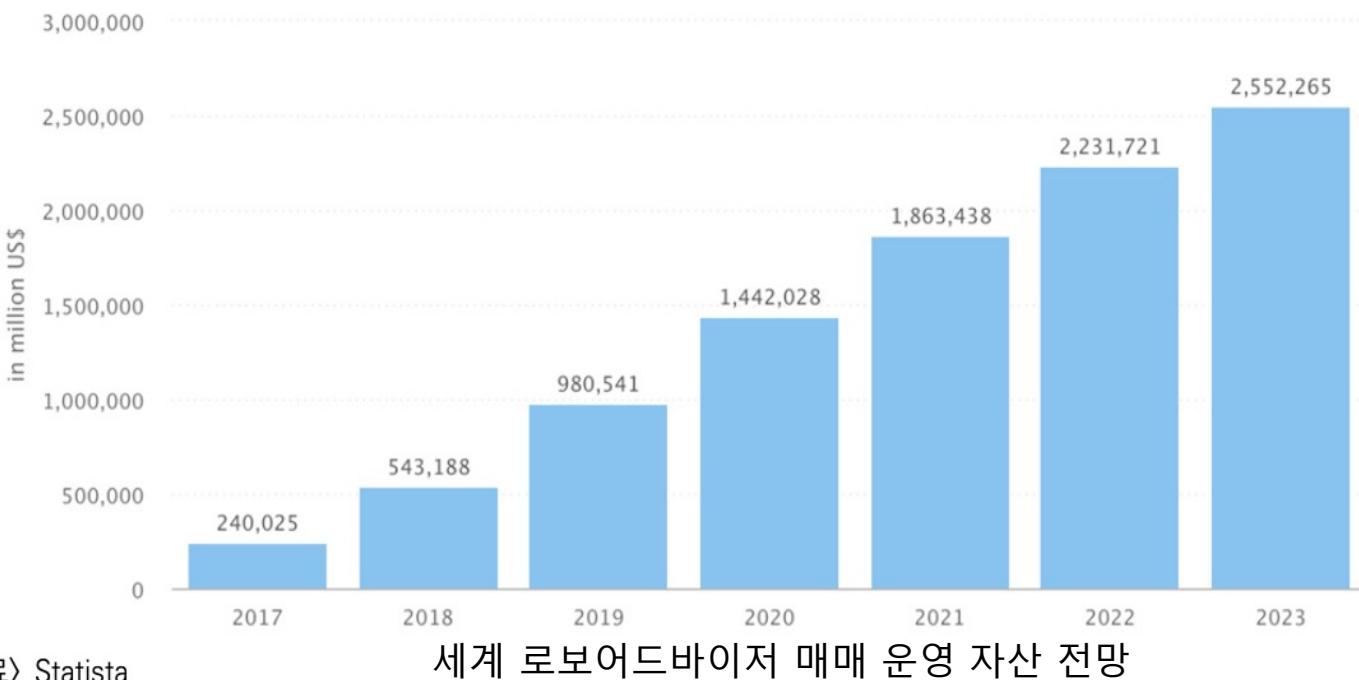
프로그램 매매 영향력 ↑

세계보건기구(WHO)의 코로나19 팬데믹(세계적 대유행) 선언에 금융시장이 출렁했다. 12일 코스피지수는 전 거래 일 대비 3.87%, 코스닥지수는 5.39% 각각 급락했고 원·달러 환율은 13.5원 올랐다. 이날 오후 서울 여의도 KB국민 은행 딜링룸에서 직원이 업무를 보고 있다. 사진=박범준 기자



주제 선정 배경

- 미국은 주식 거래의 70% 이상이 AI 투자를 활용하고 투자은행 ‘골드만삭스’ 에서는 2017년에 600명의 주식 매매 트레이더를 AI 투자를 위한 컴퓨터 엔지니어 2명만 남기고 해고.
- 모든 매매를 컴퓨터 시스템으로 진행하는 미국 펀드회사 ‘르네상스 테크놀로지’의 연평균 수익률 30%
- 주식 거래대회에서 ‘AXE’라는 인공지능의 우승 사례, 19년 기준 누적수익률 217.35% 기록
- 홍콩의 자산관리업체 Aidyia Limited의 해지펀드 등 주식 시장에서 딥러닝 주식 거래는 조금씩 상용화가 이루어지고 있으며 큰 수익률 보다는 지속적인 이익을 가져오고 있다.



* 로보어드바이저 : 알고리즘을 통해서 프로그램이 투자결정 및 자산배분을 하는 프로그램



벤치마킹 및 차별성

벤치마킹:
로보어드바이저





벤치마킹 및 차별성

로보어드바이저 종류

인공지능 방식

머신 러닝이나 딥 러닝을 통해 방대한 데이터를 사전에 학습하고 모델을 생성하여 해당 모델을 실시간 데이터에 적용

알고리즘 방식

발생 할 수 있는 상황들을 전문가가 직접 정의하고 제어 규칙화 하여 통제화된 방식의 판단을 수행

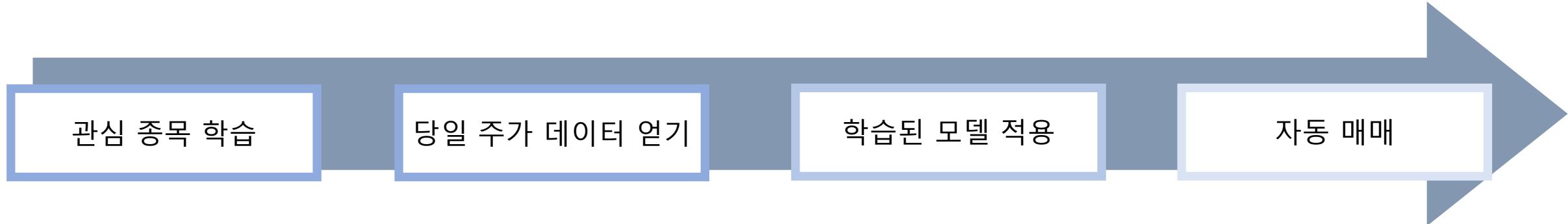
차별성

- 인공지능 방식을 사용하는 로보어드바이저 적음
- 2019년 말 기준 로보어드바이저가 직접 매매하지 않음. 프로젝트에서는 최종 매매까지 구현
- 로보어드바이저는 주로 상장 지수펀드(ETF) 국한되어있다. 개별 종목에 대한 자동매매 구현



주제 및 요약

주제 : 강화 학습을 통한 실시간 주식 매매 프로그램



학습된 모델을 적용하여 자동매매를 하는 프로그램

키움 openAPI

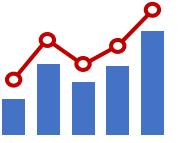
주식의 가격정보를 얻어오고, 매수와
매도 주문을 넣는 데 사용하는 기본
적인 함수를 제공



딥러닝/강화학습

강화학습과 신경망을 이용하여
과거 데이터를 학습.
학습된 모델을 이용하여 매매 결정

* 매매 수수료를 고려하여 매일 2시에 매매를 결정



개발 환경

개발 환경

파이썬, 텐서플로우, 케라스, PyQt, 키움 openAPI 이용

RLTrader 오픈소스 사용 : 강화학습 투자 시뮬레이션 프로그램

실시간 매매 기능 추가
적절한 신경망, 강화 학습 기법을 사용
주가데이터 이외에 다양한 학습 데이터



프로그램 소개

학습 데이터

	A	B	C	D	E	F	G	H	I
1	date	open	high	low	close	volume	per	pbr	roe
2	20160104	147500	148000	143500	144000	445332	7.67	0.43	6.06
3	20160105	143000	145000	142000	143500	530496	7.67	0.43	6.06
4	20160106	144000	145000	139000	140000	769406	7.67	0.43	6.06
5	20160107	138500	140000	137500	137500	586187	7.67	0.43	6.06
6	20160108	134500	138000	133500	136500	888701	7.67	0.43	6.06
7	20160111	134500	140500	134000	140000	542749	7.67	0.43	6.06
8	20160112	140500	141500	139000	139500	386506	7.67	0.43	6.06
9	20160113	140500	141500	139500	140500	338336	7.67	0.43	6.06
10	20160114	138500	143500	136500	141000	664829	7.67	0.43	6.06

971	20191216	120500	121000	119000	119000	1015046	7.56	0.43	6.08
972	20191217	119500	123000	119000	122000	1904603	7.56	0.43	6.08
973	20191218	122000	123500	120500	123000	1135407	7.56	0.43	6.08
974	20191219	122500	123500	122000	123500	905299	7.56	0.43	6.08
975	20191220	124000	124000	122500	124000	1001899	7.56	0.43	6.08
976	20191223	124000	125000	122000	122500	1462906	7.56	0.43	6.08
977	20191224	123000	123000	121500	122000	600726	7.56	0.43	6.08
978	20191226	122500	123000	121500	122500	552469	7.56	0.43	6.08
979	20191227	121000	121000	120000	120500	422565	7.56	0.43	6.08
980	20191230	120500	121500	120000	120500	467794	7.56	0.43	6.08

과거 데이터 수집

전처리



- open, close / last close
- high, low / close
- volume / last volume
- close/ MA 5,10,20,60,120 close
- volume/ MA 5,10,20,60,120 volume

+

- KOSPI close/ MA 5 ,20,60,120 close
- 환율 close/ MA 5 ,20,60,120 close
- 3년 국채가 close/ MA 5 ,20,60,120 close

- 에이전트 상태
- 주식 보유 비율
- 포트폴리오 가치 비율

종목의 시가, 종가, 고가, 저가, 거래량을 비율로 전처리하고
환율, kospi지수, 국채가격의 비율 데이터
에이전트 상태 추가

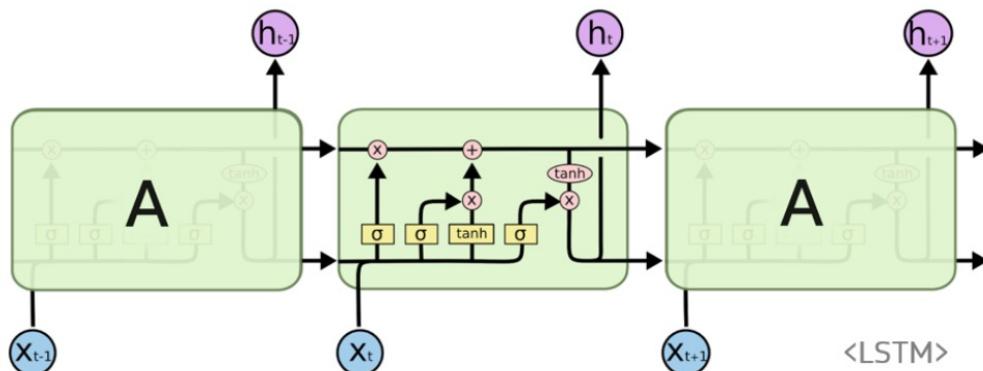
비율 데이터로한 이유는 데이터들의 범위를 일정하게 정규화 하기위해



프로그램 소개

인공 신경망

LSTM 신경망



- * 시계열의 특성을 고려해 일주일(5일) 정도의 데이터를 한번의 예측에 사용 LSTM의 스텝수 5로 두어
- * LSTM 레이어는 네 겹, Dropout 과 배치 정규화
- * Dropout 0.4

	APPL	IBM	PG	ES	IF					
	ACC	AR(3)	ACC	AR(3)	ACC	AR(3)	ACC	AR(3)	ACC	AR(3)
SVM	0.52	44.5%	0.50	-1.2%	0.49	18.6%	0.51	13.4%	0.50	9.4%
DNN	0.48	41.2%	0.52	1.4%	0.48	17.5%	0.53	16.1%	0.53	10.3%
CNN	0.52	45.2%	0.54	3.4%	0.46	14.5%	0.56	16.8%	0.51	10.7%
LSTM	0.57	50.5%	0.60	6.7%	0.58	21.6%	0.59	19.6%	0.55	15.94%

관련 논문 “Deep Robust Reinforcement Learning for Practical Algorithmic Trading(2019, August)”



프로그램 소개

강화학습 기법

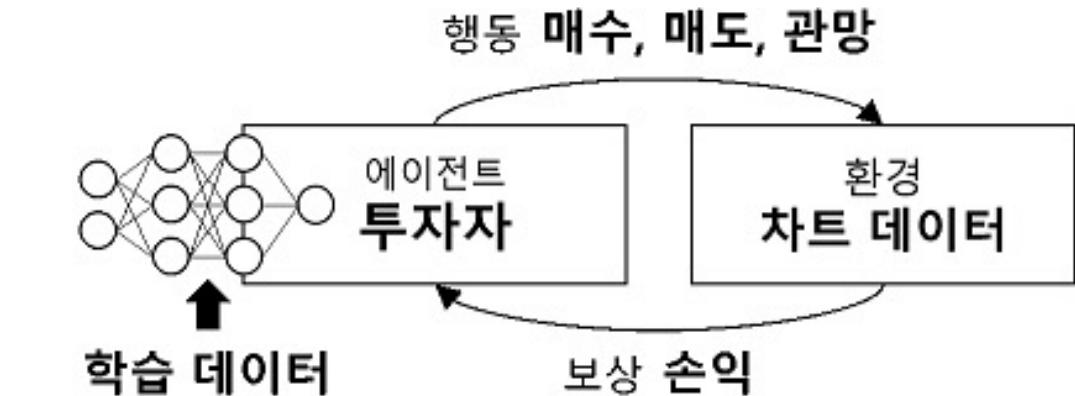
Actor - Critic

Epsilon(탐험률)

: 무작위로 행동을 결정하는 비율

에포크에서 경험을 얻기 위해 탐험해야함.

경험이 쌓이기 시작하면 신경망을 통해 행동 결정



⇒ 1로 시작해서 마지막 에포크에서는 0이 되게 함

Delayed_reward_threshold(지연보상임계값)

: 강화학습시 학습 수행 여부를 결정하는 임계값

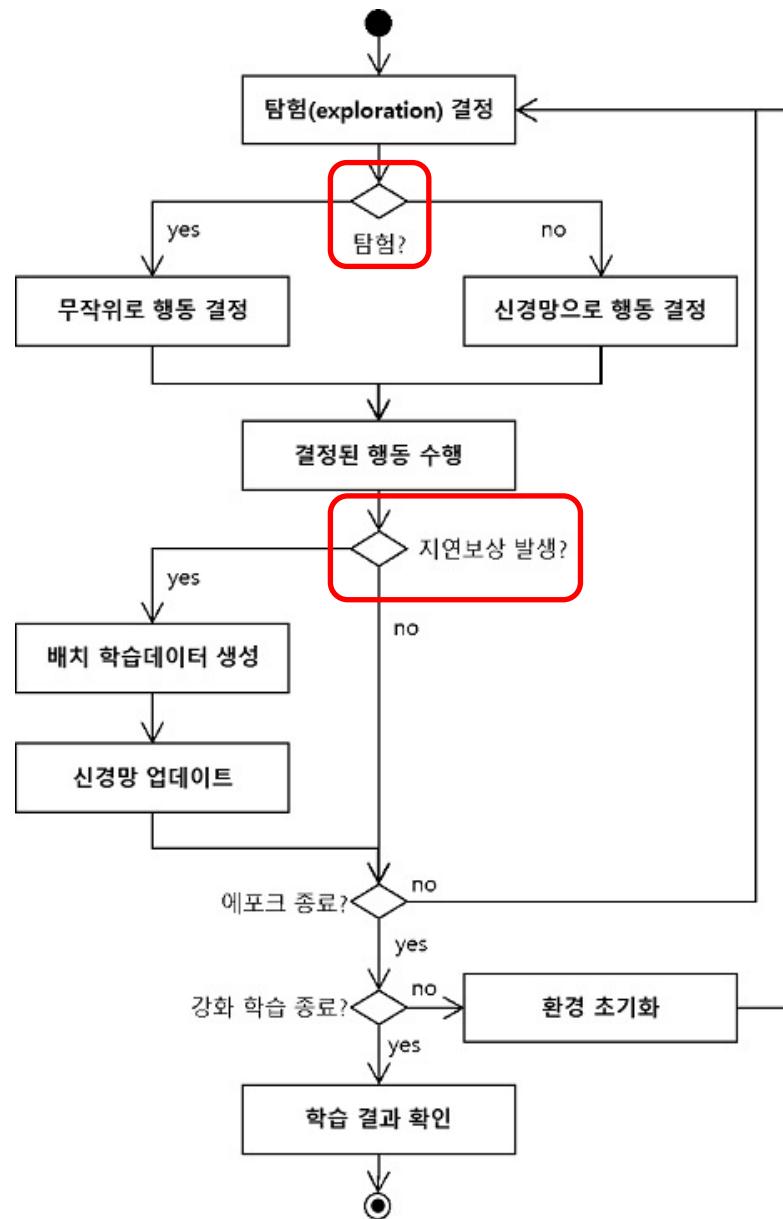
수익률 5% 이상 => 투자행동을 긍정적으로 학습
손실률 5% 이상 => 투자행동을 부정적으로 학습

⇒ 5%로 설정



프로그램 소개

학습 순서도



Epsilon : 1 -> 0

Delayed_reward_threshold: 5%



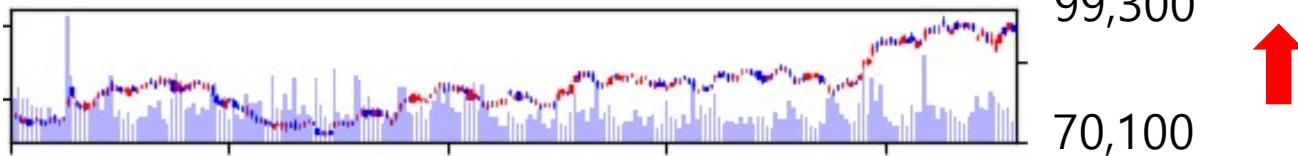
프로그램 소개

수익비교

SK하이닉스 (000660)

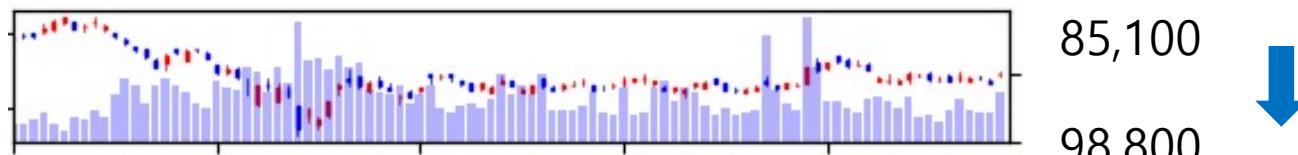
학습 기간 : 2017년 1월 1일 ~ 2019년 2월 27일

1) 2019년 3월 1일 ~ 2020년 2월 9일



주가: +41.65% 투자 수익률: +34.37%

2) 2020년 2월 10일 ~ 2020년 6월 30일



주가: -13.87% 투자 수익률: -0.31%

3) 2020년 2월 10일 ~ 2020년 10월 23일



주가: -15.08% 투자 수익률: -2.02%



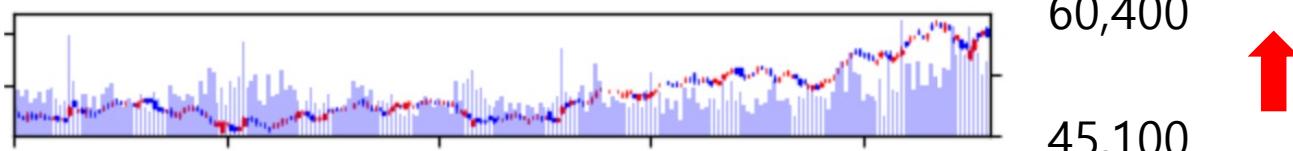
프로그램 소개

수익비교

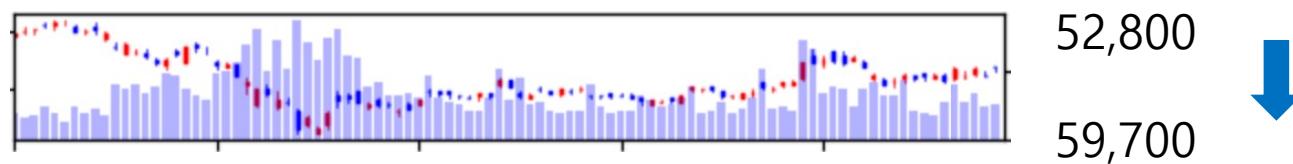
삼성전자 (005930)

학습 기간 : 2017년 1월 1일 ~ 2019년 2월 27일

1) 2019년 3월 1일 ~ 2020년 2월 9일



2) 2020년 2월 10일 ~ 2020년 6월 30일



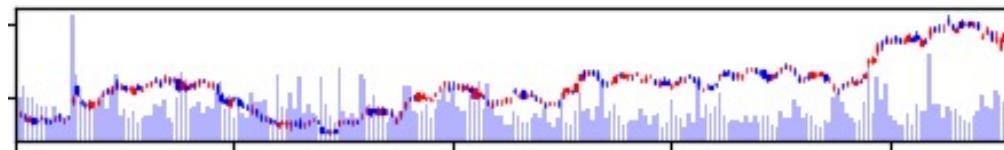
3) 2020년 2월 10일 ~ 2020년 10월 31일



SK하이닉스 (000660)

학습 기간 : 2017년 1월 1일 ~ 2019년 2월 27일

1) 2019년 3월 1일 ~ 2020년 2월 9일

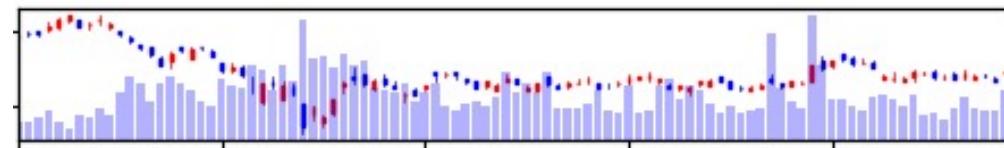


99,300
70,100



주가: +41.65% 투자 수익률: +34.37%

2) 2020년 2월 10일 ~ 2020년 6월 30일

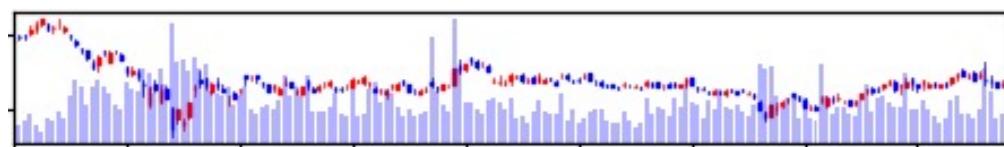


85,100
98,800



주가: -13.87% 투자 수익률: -0.31%

3) 2020년 2월 10일 ~ 2020년 10월 23일

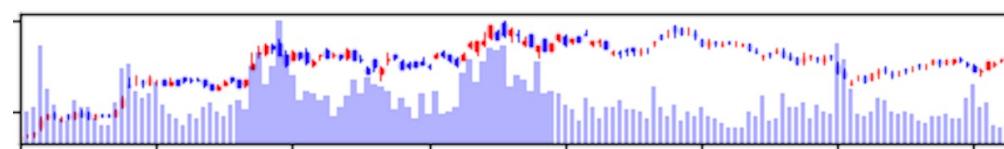


83,900
98,800



주가: -15.08% 투자 수익률: -2.02%

4) 2020년 11월 11일 ~ 2021년 06월 16일



129,500
85,900

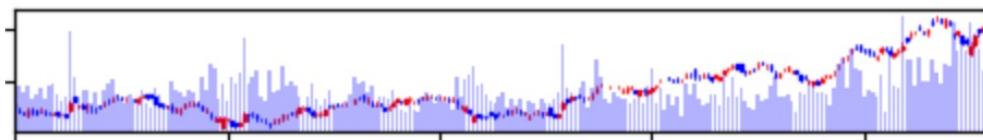


주가: +50.75% 투자 수익률: +0.19%

삼성전자 (005930)

학습 기간 : 2017년 1월 1일 ~ 2019년 2월 27일

1) 2019년 3월 1일 ~ 2020년 2월 9일

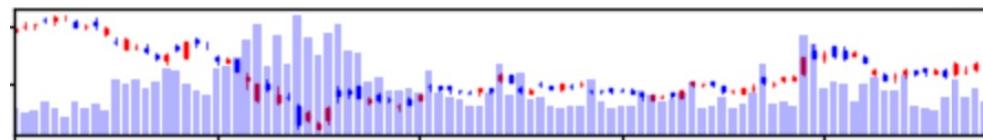


60,400
45,100



주가: +33.92% 투자 수익률: +5.40%

2) 2020년 2월 10일 ~ 2020년 6월 30일



52,800
59,700



주가: -11.56% 투자 수익률: +7.69%

3) 2020년 2월 10일 ~ 2020년 10월 31일

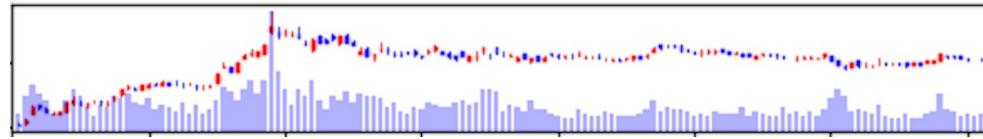


60,200
59,700



주가: +0.84% 투자 수익률: +18.37%

4) 2020년 11월 11일 ~ 2021년 06월 16일



81,800
60,400



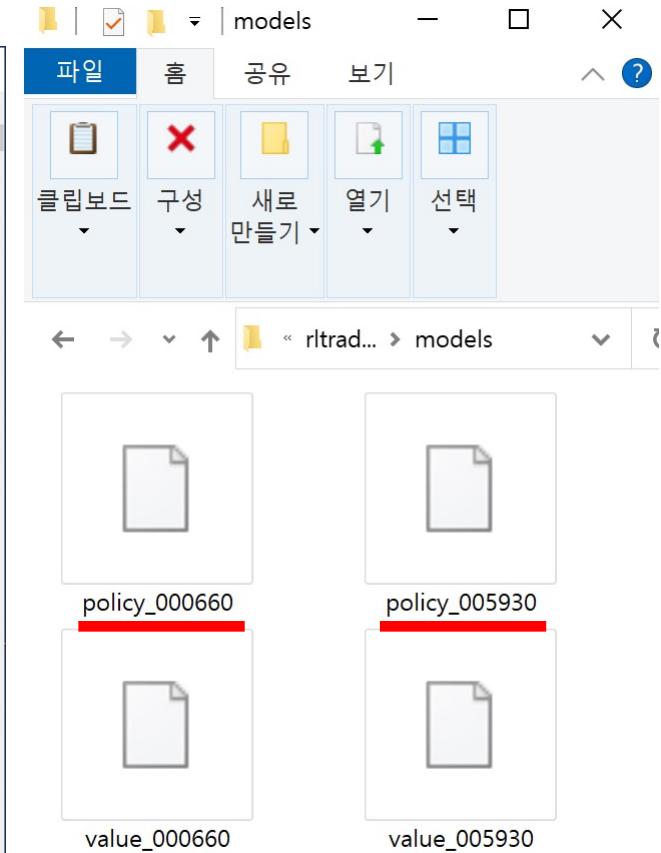
주가: +35.43% 투자 수익률: +1.67%



시나리오

자동매매 전 투자 종목 학습 후 모델 생성

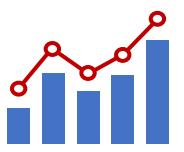
```
■ Anaconda Prompt (anaconda3) - python main.py --stock_code 000660 --num_steps 5 --learning --num_epochs 300 --lr 0.001 --start_epsilon 1 --discount_factor 0.9 --start_date 2017... - □ ×  
Instructions for updating:  
Call initializer instance with the dtype argument instead of passing it to the constructor  
From C:\Users\wogud\anaconda3\lib\site-packages\tensorflow_core\python\resource_variable_ops.py:1630: calling BaseResourceVariable.__init__()  
(from tensorflow.python.resource_variable_ops) with constraint is deprecated and will be removed in a future version.  
Instructions for updating:  
If using Keras pass *_constraint arguments to layers.  
From C:\Users\wogud\anaconda3\lib\site-packages\tensorflow_core\python\ops\nn_impl.py:183: where (from tensorflow.python.ops.array_ops)  
is deprecated and will be removed in a future version.  
Instructions for updating:  
Use tf.where in 2.0, which has the same broadcast rule as np.where  
[000660] RL:a2c Net:Istm LR:0.001 DF:0.9 TU:[1,13] DRT:0.05  
[000660] [Epoch 001/300] Epsilon:1.0000 #Expl.:515/515 #Buy:159 #Sell:129 #Hold:227 #Stocks:183 PV:14,057,920 LC:53 Loss:46.259467 ET:15.6452  
[000660] [Epoch 002/300] Epsilon:0.9967 #Expl.:515/515 #Buy:243 #Sell:206 #Hold:66 #Stocks:160 PV:12,857,858 LC:39 Loss:43.080757 ET:6.5595  
[000660] [Epoch 003/300] Epsilon:0.9933 #Expl.:509/515 #Buy:217 #Sell:187 #Hold:111 #Stocks:169 PV:14,226,501 LC:47 Loss:51.552758 ET:6.6725  
[000660] [Epoch 004/300] Epsilon:0.9900 #Expl.:509/515 #Buy:254 #Sell:235 #Hold:26 #Stocks:132 PV:10,329,782 LC:16 Loss:20.214860 ET:5.5342  
[000660] [Epoch 005/300] Epsilon:0.9866 #Expl.:510/515 #Buy:63 #Sell:34 #Hold:418 #Stocks:194 PV:14,372,334 LC:59 Loss:44.985847 ET:7.0172  
[000660] [Epoch 006/300] Epsilon:0.9833 #Expl.:503/515 #Buy:234 #Sell:213 #Hold:68 #Stocks:157 PV:13,488,771 LC:45 Loss:47.061688 ET:6.6652  
[000660] [Epoch 007/300] Epsilon:0.9799 #Expl.:505/515 #Buy:146 #Sell:99 #Hold:270 #Stocks:199 PV:14,735,206 LC:53 Loss:50.067221 ET:6.8507  
[000660] [Epoch 008/300] Epsilon:0.9766 #Expl.:507/515 #Buy:138 #Sell:89 #Hold:288 #Stocks:198 PV:14,648,266 LC:53 Loss:47.239028 ET:6.7769  
[000660] [Epoch 009/300] Epsilon:0.9732 #Expl.:497/515 #Buy:257 #Sell:230 #Hold:28 #Stocks:116 PV:13,424,626 LC:40 Loss:34.398566 ET:6.3490  
[000660] [Epoch 010/300] Epsilon:0.9699 #Expl.:496/515 #Buy:244 #Sell:186 #Hold:85 #Stocks:176 PV:13,197,206 LC:51 Loss:39.464851 ET:6.8896  
[000660] [Epoch 011/300] Epsilon:0.9666 #Expl.:495/515 #Buy:96 #Sell:44 #Hold:375 #Stocks:195 PV:14,409,090 LC:57 Loss:44.869985 ET:7.0900  
[000660] [Epoch 012/300] Epsilon:0.9632 #Expl.:497/515 #Buy:110 #Sell:60 #Hold:345 #Stocks:196 PV:14,510,768 LC:56 Loss:47.467536 ET:7.1289
```



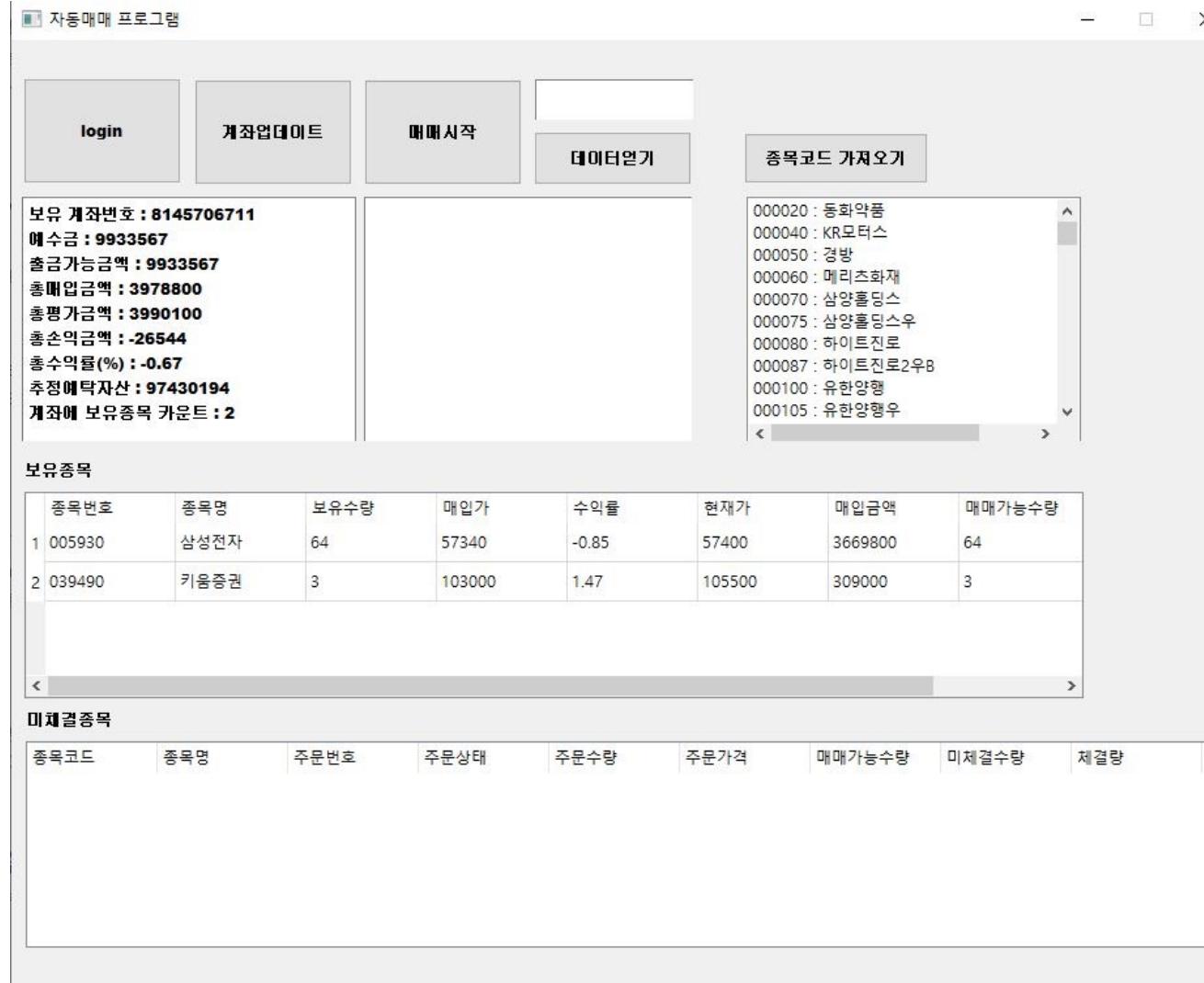
에포크 수만큼 반복 학습하며 신경망 업데이트

4개 항목

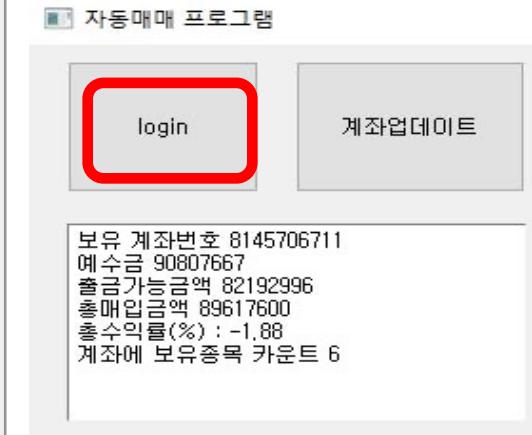
모델 저장



시나리오



1. 로그인



로그인 후 키움 연동
현재 자신의 계좌 조회



증목코드 가져오기를 통해
증목별 코드를 조회



시나리오

2. 종목 데이터 얻어오기 : 당일 주식의 매매를 판단하기 위해 데이터 얻기



원하는 종목 코드를 입력 후
데이터 얻기를 클릭



Api를 통해 종목코드와 같은 주가데이터
.csv 파일이 생성된다.



A	B	C	D	E	F	G	H	I	J
1	date	time	open	high	low	close	volume	kospi_close	exchange_bond_close
2	20201109	1400	86500	87300	85600	86400	1758227	25380	11855
3	20201106	1400	86500	87700	86000	86400	2312773	25000	11960
4	20201105	1400	84100	85700	83300	85600	2654868	24845	12055
5	20201104	1400	82100	84000	81200	83800	2692140	24355	12100
6	20201103	1400	79600	81100	79600	80800	1578923	24050	12090

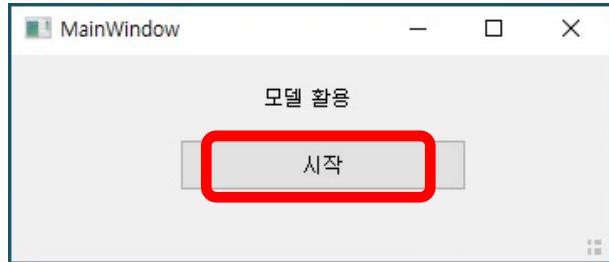
* LSTM의 길이를 5로 두었기 때문에
당일의 가격을 판단하려면 5일치 데이터 필요

날짜별 학습데이터



시나리오

3. 데이터 전처리



A	B	C	D	E	F	G	H	I	J
1	date	time	open	high	low	close	volume	kospi_close	exchange_bond_close
2	20201109	1400	86500	87300	85600	86400	1758227	25380	11855 111465
3	20201106	1400	86500	87700	86000	86400	2312773	25000	11960 111560
4	20201105	1400	84100	85700	83300	85600	2654868	24845	12055 111570
5	20201104	1400	82100	84000	81200	83800	2692140	24355	12100 111465
6	20201103	1400	79600	81100	79600	80800	1578923	24050	12090 111425



비율데이터로 전처리

A	B	C	D	E	F	G	H	I	J	K	
1	-0.00562	-0.00318	-0.01056	-0.01916	-0.03148	-0.01709	0.005093	-0.02376	0.006838	0.524813	0.009945 0
2	-0.00611	-0.00411	-0.01149	-0.02067	-0.03307	0.008489	0.015253	-0.00339	0.001698	-0.27938	0.00717 0
3	-0.00572	-0.00592	-0.01221	-0.02214	-0.03471	-0.00508	0.003356	-0.01845	0.010169	-0.08677	0.011197 -
4	-0.00671	-0.00991	-0.01517	-0.02581	-0.0386	0.0151	0.011666	-0.00833	0.006711	0.55325	0.013513 -
5	-0.00812	-0.0136	-0.01877	-0.03025	-0.04325	0	0.003322	-0.00498	0.003333	-0.25804	0.011085 -

모델 활용

전처리된 데이터를 모델의 입력값에 넣어
매매량을 자동으로 판단 후
000000_date_data.csv에 저장



날짜, 매매량, 거래가, 종목명

A	B	C	D	
1	date	vol	close	code
2	20201109	-49	86400	660



시나리오

4. 매매 시작

매매시작	000660
데이터얻기	



저장된 거래량을 이용하여
키움 API에 매매 요청



A	B	C	D
date	vol	close	code
1	20201109	-49	86400
2			660

000660_date_data

예시

*

A	B
date	vol
20201101	8

*

A	B
date	vol
20201101	-4

*

A	B
date	vol
20201101	0

매매시작	크래온데이터얻기
오늘의 날짜 : 20201101 매매량 : 8 <u>매수한다</u> 총목 : 005930, 매수량 : 8	

매매시작	크래온데이터얻기
오늘의 날짜 : 20201101 매매량 : -4 <u>매도한다</u> 총목 : 005930, 매도량 : -4	

매매시작	크래온데이터얻기
오늘의 날짜 : 20201101 매매량 : 0 <u>매매하지 않는다.</u>	

- Vol에 따라
- 양수 -> 매수
 - 음수 -> 매도
 - 0 -> 관망



시나리오

5. 보유 종목 확인

보유종목							
종목번호	종목명	보유수량	매입가	수익률	현재가	매입금액	매매가능수량
1 005930	삼성전자	156	80300	0.98	81800	8946900	156

6. 미체결 종목 표시

미체결종목				
종목코드	종목명	주문번호	주문상태	주문수

미체결 종목이 있을 시 상태를 표시해준다.

156

80300

81800

0.98

8946900

156



결론

개발 이유 :

1. 주식시장에 프로그램 매매의 영향력 상승
2. 인공지능이 어느 정도의 수익률을 가져올지에 대한 호기심

제공하는 기능 :

1. 종목 차트 데이터 조회
2. 학습된 모델을 이용한 자동매매
3. 계좌 조회

기대 효과 :

1. 주식시장에 투자하고 싶지만 방법을 모르는 초보자
2. 업무시간과 장시간이 겹쳐 집중하지 못하는 직장인
3. 큰 수익은 아니지만 안정적인 수익을 원하는 사람



Thank You