Stat101 for BC*

(*Brain Consultants)

Kim, Namil

2024-08-14

Contents

복습

Brain API

Lambda (익명함수) : 간단한 함수의 적용으로 파이썬에서 자주 사용하게 되며, 특히 Pandas에서 쉽게 적용
- 사례 lambda x : x[‘name’]

JSON (Brain) : 시뮬레이터를 시작할 수 있는 형식으로 원하는 포맷을 ace_lib의 generate_alpha로 제작

DataFrame : Pandas 에서 다루게되는 데이터 라이브러리로서 df 로 줄여서 표기

Plotly : 결과값을 시각화 할 수 있는 툴 (Bokeh등을 적용할수도 있음)

ThreadPool : 동시에 여러개의 작업을 처리해줄 수 있는 방식

복습

JSON 사례

[9]:
```
#when you send multiple alphas for simulation, please make sure all alphas of a single list should have common settings
#alphas with different settings should be sent in a different list, for instance below list has all alphas with same settings

alpha_list = [ace.generate_alpha(x, region= "USA", universe = "TOP3000",) for x in expression_list]

alpha_list[0]
```

[9]:
```
{'type': 'REGULAR',
 'settings': {'nanHandling': 'OFF',
  'instrumentType': 'EQUITY',
  'delay': 1,
  'universe': 'TOP3000',
  'truncation': 0.08,
  'unitHandling': 'VERIFY',
  'testPeriod': 'P6Y',
  'pasteurization': 'OFF',
  'region': 'USA',
  'language': 'FASTEXPR',
  'decay': 0,
  'neutralization': 'INDUSTRY',
  'visualization': False},
 'regular': 'ts_skewness(vec_avg(nws35_createdtime),120)'}
```

This is an example - how alpha actually looks like when you send it to the platform.

복습

DataFrame 사례

Re-simulation

```
[20]: new_result = ace.simulate_alpha_list_multi(s, new_alpha_list)
```

```
Warning, list of alphas too short, single concurrent simulations will be used instead of multisimulations
100%|██████████| 3/3 [00:36<00:00, 12.10s/it]
```

```
[21]: result_st2 = hf.prettify_result(new_result, clickable_alpha_id=False)
      result_st2
```

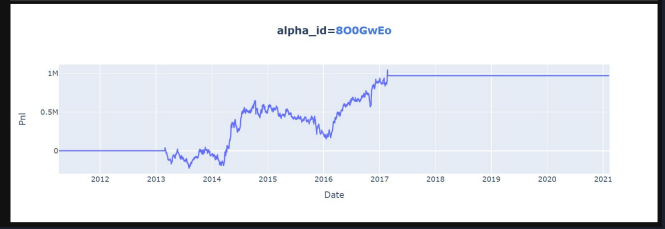
[21]:

	book_size	drawdown	fitness	long_count	margin	pnl	returns	sharpe	short_count	turnover	alpha_id	expression	concentrated_weight	high_turnover	is_ladder_sharpe	low_fitness	low_sharpe	low_sub_universe_sharpe	low_tu
0	20000000	0.0533	0.32	1037	0.000490	914176	0.0227	0.76	1036	0.0929	aN0R5Px	group_rank(ts_skewness(vec_avg(mws36_relevance...	FAIL	PASS	FAIL	FAIL	FAIL	PASS	
1	20000000	0.0442	0.19	1536	0.000415	1394786	0.0143	0.57	1533	0.0689	1o8GrmM	group_rank(ts_skewness(vec_avg(mws36_sentiment...	PASS	PASS	FAIL	FAIL	FAIL	PASS	
2	20000000	0.0594	0.13	1284	0.000273	1046677	0.0107	0.43	1274	0.0785	YNVlRWW	group_rank(ts_skewness(vec_avg(mws36_novelty_o...	FAIL	PASS	FAIL	FAIL	FAIL	PASS	

Plotly 사례

Visualise pnl of an alpha

```
[16]: alpha_pnl = ace.get_alpha_pnl(s, result_st1['alpha_id'][0])
      px.line(x = alpha_pnl.index, y = alpha_pnl.Pnl, title=f'<b>alpha_id={hf.make_clickable_alpha_id(alpha_pnl.alpha_id[0])}</b>')\
        .update_layout(xaxis_title="Date", yaxis_title="Pnl", title_x=0.5)
```



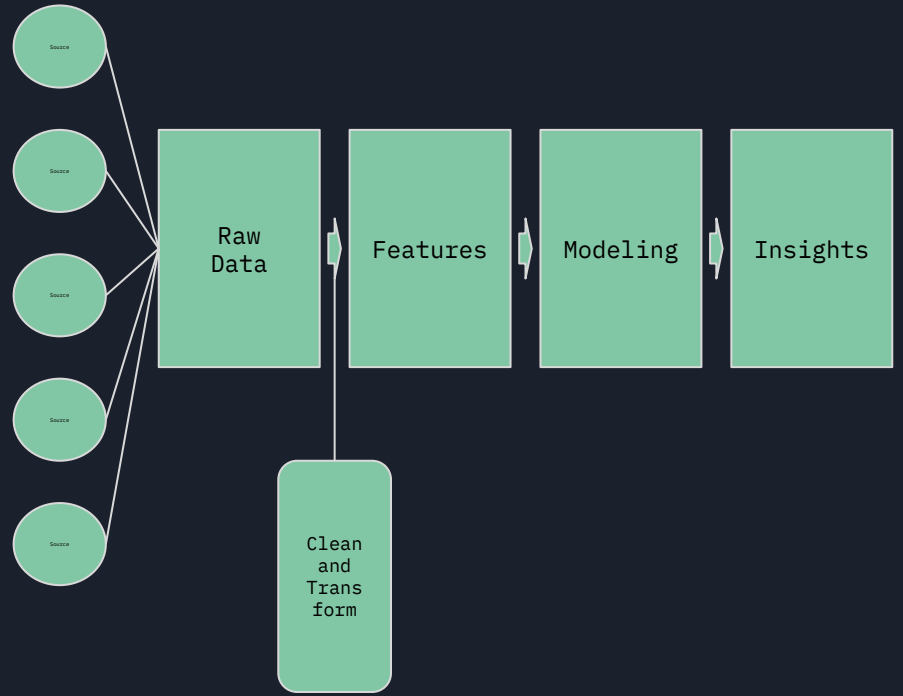
복습

시계열 분석 사례

```
In [7]:  # Display results
         print(simple_stats)
         print(log_stats)
         print(test_results)
```

	axp	cat	sbux
count	2515.000000	2515.000000	2515.000000
mean	0.014565	0.059504	0.048054
std	2.446218	2.169648	2.682622
min	-17.594900	-14.517500	-28.286200
25%	-1.111050	-1.144150	-1.247450
50%	-0.018200	0.048900	-0.051200
75%	1.092900	1.206100	1.248750
max	17.926600	14.722900	14.635400
skewness	-0.034627	0.011678	-0.082476
excess kurtosis	6.055251	4.459195	8.754924

	axp	cat	sbux
count	2515.000000	2515.000000	2515.000000
mean	-0.015434	0.035949	0.011885
std	2.452898	2.171483	2.695888
min	-19.352286	-15.685851	-33.248699
25%	-1.117268	-1.150746	-1.255296
50%	-0.018202	0.048888	-0.051213
75%	1.086971	1.198885	1.241017
max	16.489221	13.734947	13.658647
skewness	-0.336635	-0.201865	-0.597424
excess kurtosis	6.494046	4.700869	12.908121



주사위 문제
- 주사위가 공정하다는 것을 어떻게 평가할 수 있을까?

통계적 검정

- 실험설계 : 주사위를 충분히 많이 던져서 결과를 기록
- 데이터 수집 : 각 면이 나온 횟수를 기록 (1~6)
- 기대 빈도 계산 : 공정한 주사위는 각 면이 동일한 확률(⅙) 으로 나와야 함
- 통계적 검정
  - 카이제곱 검정(Chi-square test) 수집된 데이터와 기대 빈도를 비교하기 위해 카이제곱 검정 사용
  - 유의 수준 : 일반적으로 0.05 정도로 설정
- 결과 해석
  - p-value : 카이제곱 검정 결과로 p-값을 얻게 됨, p-값이 유의 수준보다 작다면 (ex: $p < 0.05$) 주사위가 공정하지 않다고 결론
  - 공정성 판단 : p-값이 유의수준보다 크다면 (ex: $p > 0.05$) 주사위가 공정하다고 결론내릴 수 있음

말 시합 문제(Horse Racing Problem)

25마리의 말/ 이들 중 가장 빠른 3마리의 말?
Condition

한번에 5마리씩만 경주
각 말의 정확한 시간 기록은 알 수 없으며, 경주에서의 순위만 알 수 있음
말들을 시합시키는 횟수를 최소화하여 가장 빠른 3마리의 말

첫번째 라운드
5마리의 말 5개 그룹 경주, 총 5번의 경주를 통해 각 그룹의 순위정하기

두번째 라운드
첫번째 라운드에서 각 그룹의 1위 말들을 모아서 다시 경주, 이 경주에서 가장 빠른

세번째 라운드
두번째 라운드에서 각 2위와 3위를 한 말들은 각각 두번째로 빠른 말과 세번째로
(각 예선시에 말이 2위 3위 할 가능성 역시 존재)

시계열 분석

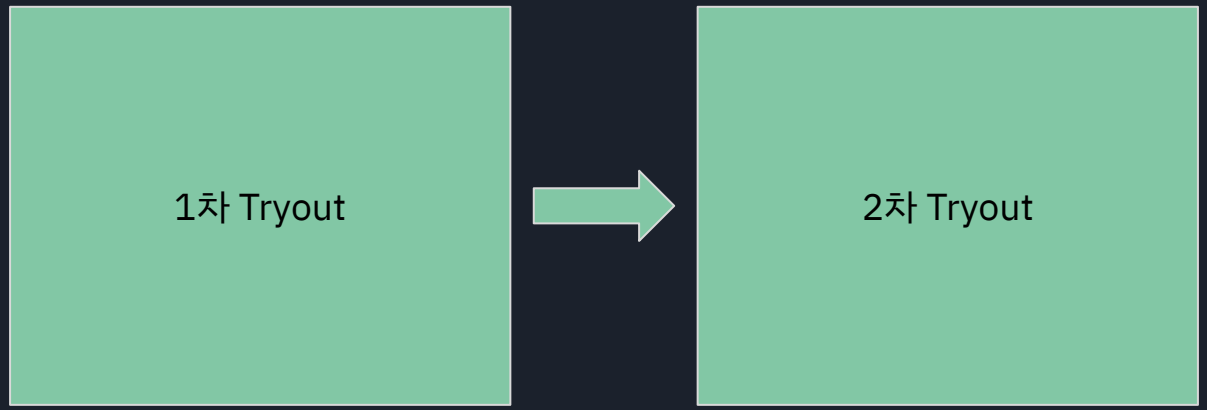
비슷하게 갈 수 있는지 확인해보기

- 데이터 수집
- 시계열 데이터의 특성 값 등이 중요

ARIMA 등...
성질을 대표하는 통계치

Value

2 Step Method

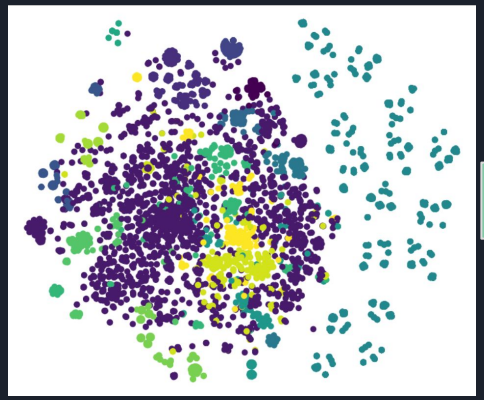


클러스터링

1st Step

일정한 값들의 특징을 뽑아내어, 특성별로 분류 및 전개하여 식 생성 (단순하며, 시뮬레이터 회당 시간이 적어야 함)

다이달로스의 알고리즘 (Desc Algo)가 그 사례 중 하나



Sharpe 0.50
Turnover 151.13%
Fitness 0.09
Returns 5.09%
Drawdown 51.56%
+
Time Series Data

2 Step Method

Fraction(x)와 Z-score(x)을 동시에 이용

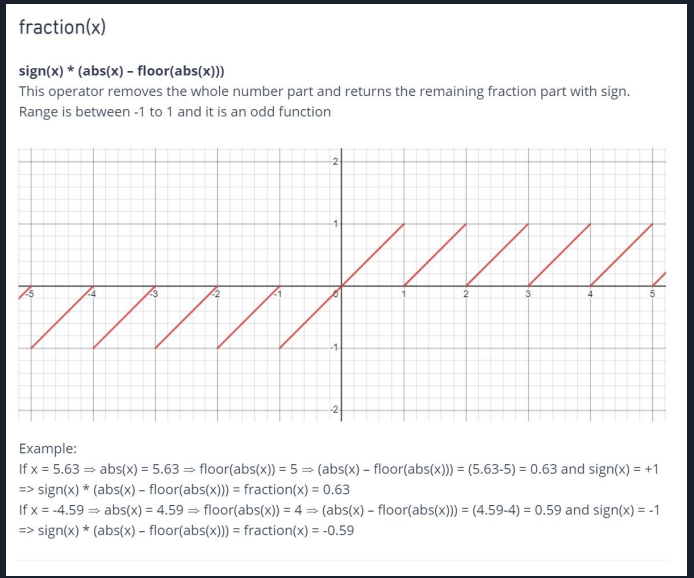
Add(filter=True), Multiply(filter=True) 을 이용

한번에 최대한 많은 데이터 동시에 검증 가능

합성값에 대해 neutralize를 하는 경우 원치 않은 결과가 발생할 수 있음

적극적으로 사용한 값에 대해 분석

휴리스틱하게 접근이 가능한 모델이지, 정확하게 잘라 말하기는 힘든 모델



2 Step Method

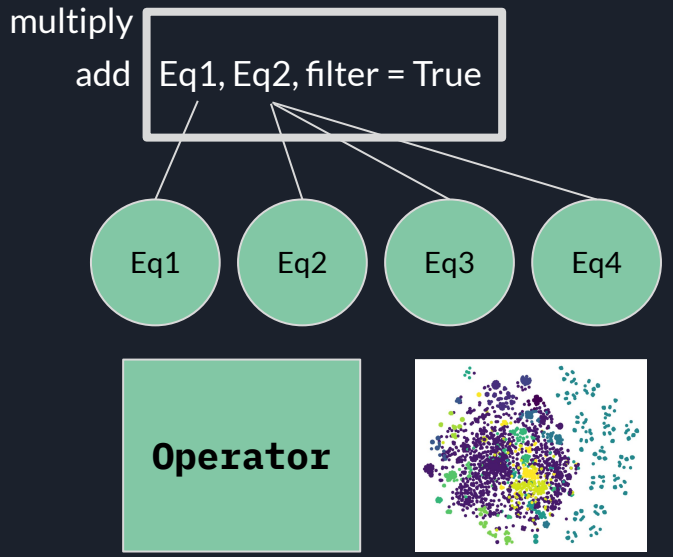
1차에서 대략적인 추세 및 경향성만 판단

long count, short count 등을 지표로 사용할 수 있음

Sharpe를 볼 수는 있으나 turnover의 경우 제일 빈번한 데이터를 기준으로 움직이므로 주의해서 봐야 함

Add, Multiply 의 경우 모델 검증 및 개선 양방향으로 적용 가능하여 상당히 유용(다이달로스)

```
add(equation1, equation2, filter = True)
```

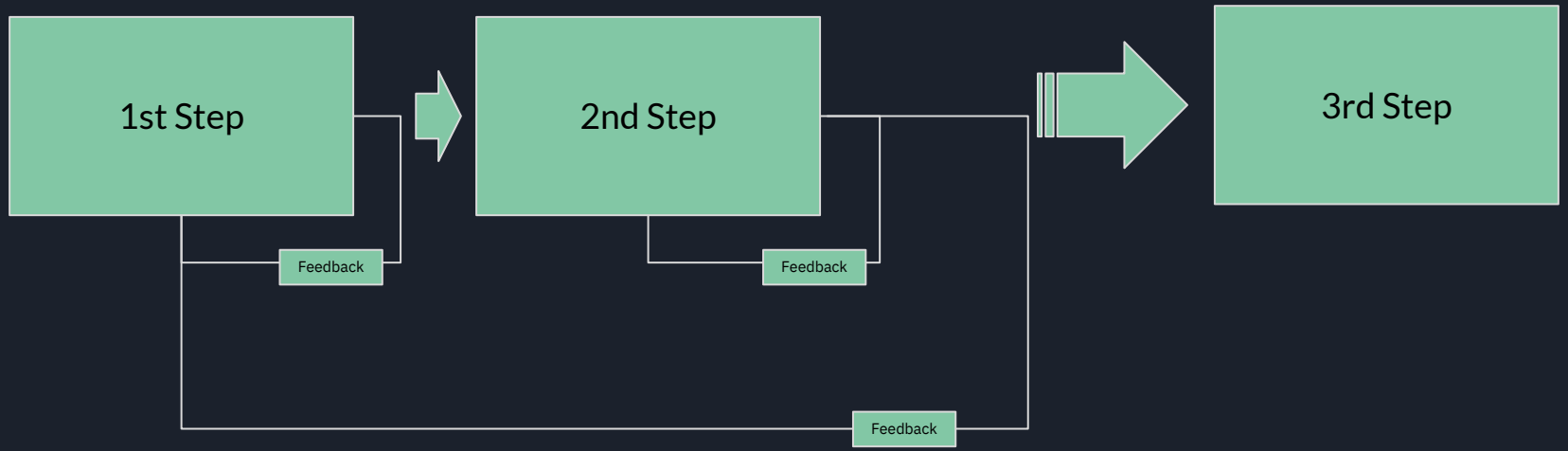


2 Step Method

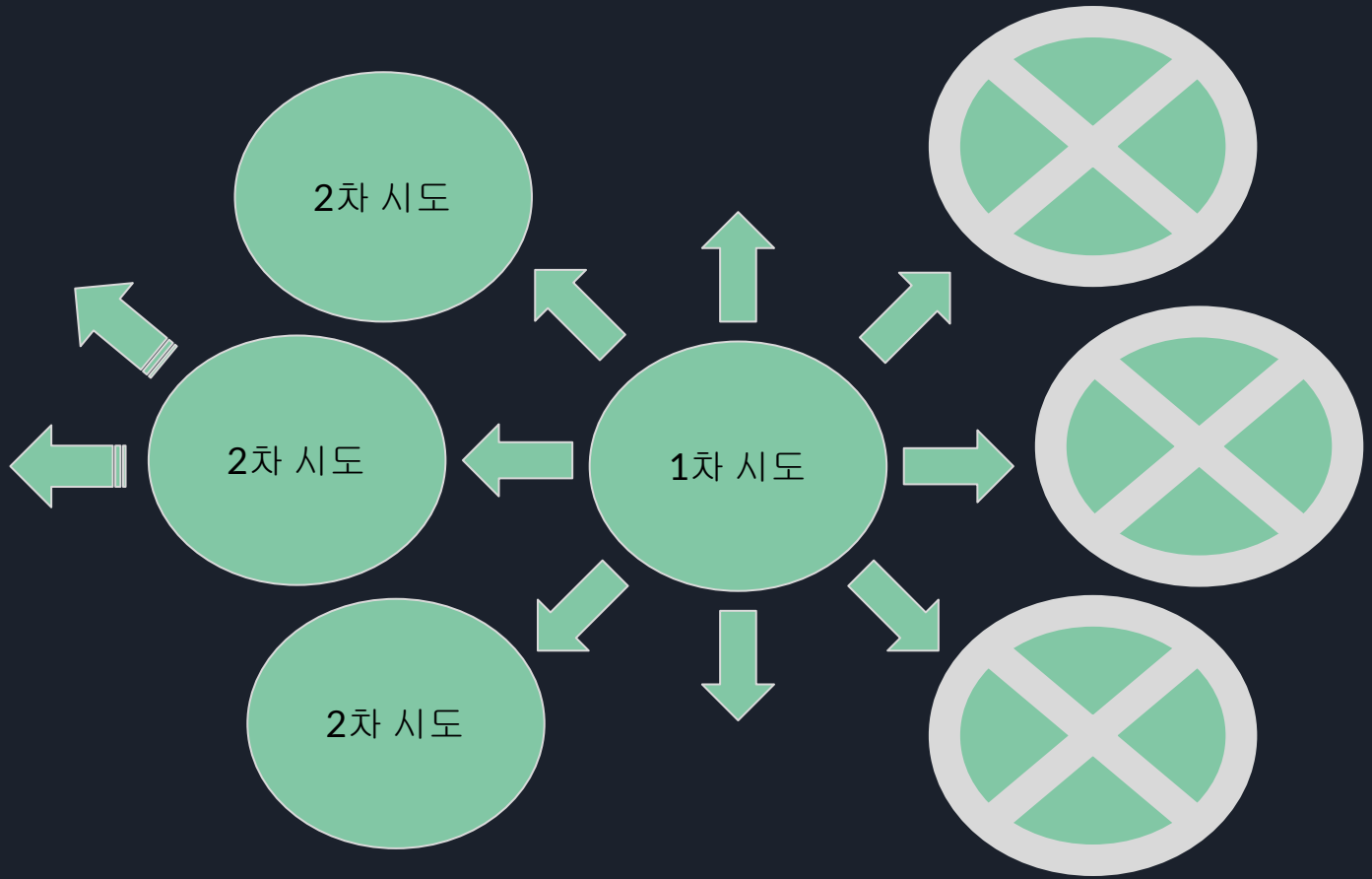
빠르게 **1st Step** 거친 이후

2nd Step의 경우 Data Driven 한 Feedback 이용하여 식 전개

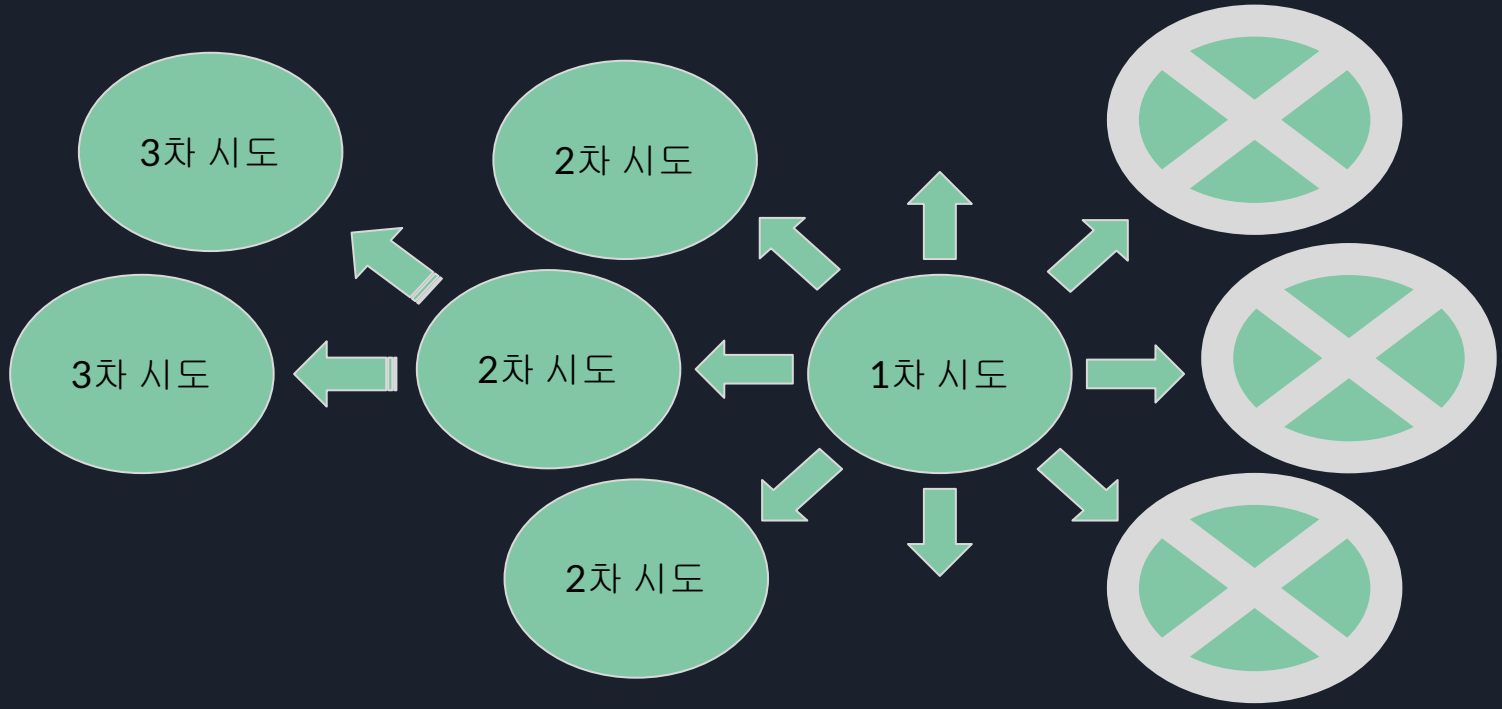
Feedback 의 연속



탐색 방향 설정, 적용



탐색 방향 설정, 적용



다양한 아이디어 도입 가능

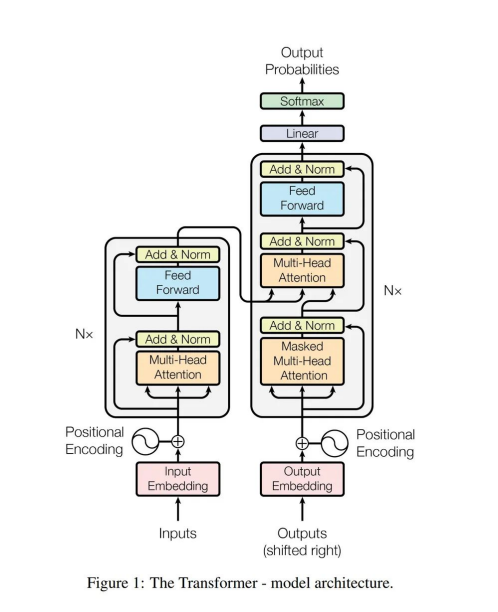


Figure 1: The Transformer - model architecture.

Transformer Model 등의 언어모델 적용이 가능함

Time Series Model에 Sequence 별로 적용 역시 가능

Are Language Models Actually Useful for Time Series Forecasting?

Mingtian Tan
University of Virginia
wtd3gz@virginia.edu

Mike A. Merrill
University of Washington
mikeam@cs.washington.edu

Vinayak Gupta
University of Washington
vinayak@cs.washington.edu

Tim Althoff
University of Washington
althoff@cs.washington

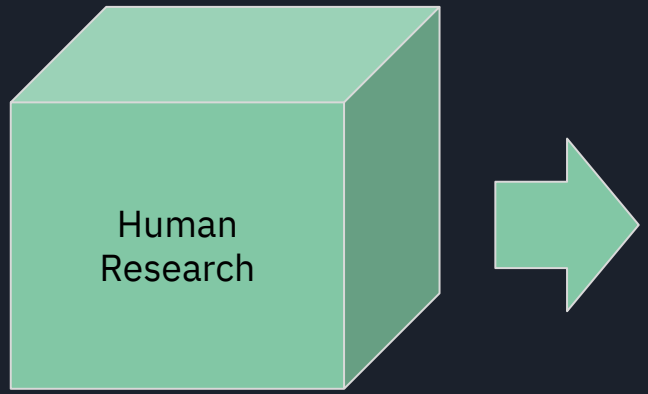
Thomas Hartvigsen
University of Virginia
hartvigsen@virginia.edu

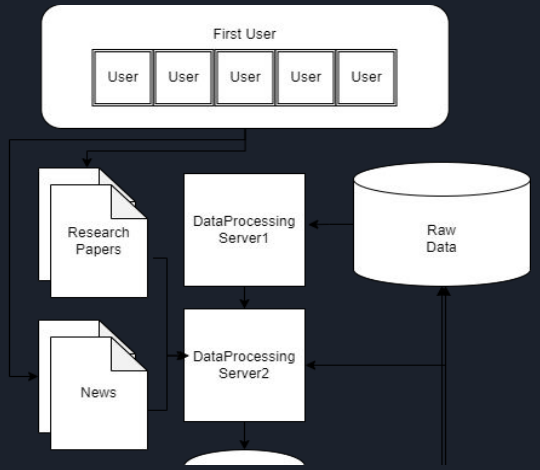
Abstract

Large language models (LLMs) are being applied to time series tasks, particularly time series forecasting. However, are language models actually useful for time series? After a series of ablation studies on three recent and popular LLM-based time series forecasting methods, we find that removing the LLM component or replacing it with a basic attention layer does not degrade the forecasting results—in most cases the results even improved. We also find that despite their significant computational cost, pretrained LLMs do no better than models trained from scratch, do not represent the sequential dependencies in time series, and do not assist in few-shot settings. Additionally, we explore time series encoders and reveal that patching and attention structures perform similarly to state-of-the-art LLM-based forecasters.[1]

지난과제 다이달로스 3,000회 시뮬레이션

받은 데이터를 활용하여, 어떻게 이 식들로부터 발전방향을 이끌어낼지 생각해보시면 좋을 것 같습니다.





과제

본인이 수집한 베이스 알파(너겟)에 대해 특정 스탯을 분석해보고, 그 결과를 사용해 어떻게 전개해나갈 수 있을지를 조금 더 고려해 볼 것