Big Data Term Project Report

생성 모델을 이용한 주식 변동 탐지 모델



2021. 12. 13.

정치국제학과 16 최종찬

응용통계학과 17 김권호

|  |
| --- |
| Big Data Term Project Report |

|  |
| --- |
| **□ 목차**  **Ⅰ. 서론**  1. 문제 정의(Problem Statement)  2. EDA + Data Preprocessing  **Ⅱ. 본론**  1. Analyzing previous approaches (Baseline Model)  1) LSTM  2) 학습과정  2. Modeling  1) Time Series GAN  2) 학습과정  3. 모델 성능 비교  1) 그래프 개형 비교  2) 성능 지표 비교    **Ⅲ. 결론**   1. 의의 2. 한계점 및 보완점 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ⅰ. 서론**   1. 문제 정의   코로나19 확산 이후로 전 세계 주식 시장이 급락하고, 이후 반등한 직후 ‘동학개미 운동’으로 불리는 개인 투자자들의 주식 열풍이 거세졌다. 특히, 비대면 문화의 확산으로 IT 분야의 급성장이 눈에 띄었는데, 이는 구글을 비롯한 세계적인 IT 기업의 주식에도 영향을 미쳤다. 이러한 흐름은 대중들의 관심을 지속적으로 끌었고, 시장을 보다 정확하게 예측하는 것이 더욱 중요해졌다.  이에 우리는 특정 도메인 분야에 일반적으로 적용될 수 있는 주식 예측 모델 개발을 목표로 하였다. 이는 해당 분야에 대한 지식 없이도 주식 시장에 대한 예측을 간편하게 해 일반인들의 접근성 향상 역시 도모할 수 있다.    그러나 기존 딥러닝 기반 시계열 분석에서 많이 사용되는 RNN 기반의 예측 모델(LSTM 등)의 예측 결과는 그 실효성이 의심되는 경우가 존재한다.  https://codingapple.com/wp-content/uploads/2020/11/1_A-ubY-due4lcTEOgGSCvoQ.jpeg  위의 그래프는 임의의 시계열 데이터에 대한 단일 LSTM모델의 예측 결과와 실제 값을 표현한 것이다.  언뜻 보면 실제 수치와 예측 값이 매우 잘 일치하여 해당 모델의 성능이 매우 좋은 것으로 보인다. 하지만 결과를 자세히 들여다보면 예측 그래프의 개형이 실제 추세보다 오른쪽으로 치우친, 추종 현상이 발생했음을 확인할 수 있다.(Right shift 문제) 즉, 예측 결과가 실제 값의 추세를 뒤늦게 따라가고 있는 모습을 보인다. 이는 모델이 다음 시점의 주식을 예측할 때, 이전 시점의 값과 동일한 수치를 결과로 내놓는다는 것이다. 결국 실제 적용 시 전혀 실효성이 없는 예측이 된다. 따라서 Right shift 문제의 해결과 실효성 있는 예측 결과를 얻는 것으로 문제를 정의 했다.    우선 Right shift 문제에 영향을 덜 받는 GAN 모델을 제안하려고 한다. 증감 패턴 탐지의 task의 경우 이상치 탐지의 task와 비슷한 부분이 많다고 생각했다. 따라서 해당 task에서 주로 사용되는 생성 모델을 적용해보기로 했다. Right shift문제가 해결 되었는가를 평가하기 위해, 실제 값과 모델의 예측 값을 그래프로 시각화하여 Baseline의 그래프와 비교하며 평가할 예정이다.  그리고 예측의 결과가 실효성이 있는가를 평가하기 위해서 평가 지표를 수치 예측에 사용되는 거리 기반의 metric(RMSE, MSE, MAE)이 아닌 다른 metric을 이용하기로 했다. 이전 시점을 기준으로 수치의 증감을 얼마나 잘 탐지하는 가를 평가하는 metric을 통해 모델 평가를 진행했다. (패턴 탐지) 즉, 데이터의 True 값을 이전 시점을 기준으로 증가했는지 감소했는지 따져 labeling하여 모델의 결과가 이를 얼마나 잘 맞췄는지를 평가한다. 이를 통해 Right shift 문제에 구애 받지 않고 모델의 예측 결과를 평가할 수 있으며, 예측 자체의 실효성도 높게 가져갈 수 있게 하였다.   1. EDA + Data preprocessing 2. EDA   사용한 데이터 셋은 Kaggle의 FAANG 데이터 셋을 이용했다. 총 5개의 IT 기업에 대한 일별 주식 데이터로 구성되어 있으며, 기록된 기간은 다음과 같다.   |  |  |  | | --- | --- | --- | | 기업 | 기록 시작 시점 | 기록 종료 시점 | | Amazon | 1997-05-15 | 2020-08-14 | | Apple | 1980-12-15 | 2020-09-01 | | Facebook | 2012-05-18 | 2020-08-18 | | Google | 2004-08-19 | 2020-09-04 | | Netflix | 2002-05-23 | 2020-08-03 |   각 데이터 파일은 7개의 동일한 column을 보유하고 있다. 이 중,수정 종가란 통상적으로 ‘주가’라고 이야기 되어지는 것과 같은 개념이다. 주가에 증자, 액면 분할 등 이벤트가 발생했을 때, 이전 시점들의 주가도 비교가 가능하도록 이에 맞게 수정을 하는데, 이것이 수정 종가이다. 따라서 예측의 실효성을 위해서 Adj Close 값을 예측 대상으로 선정했다. 이에 따라 기존 Close(종가) column은 제거하였다.   |  |  | | --- | --- | | Column 명 | 설명 | | Date | 일시 | | Open | 일 시작가 | | High | 일 최고가 | | Low | 일 최저가 | | Close | 종가 | | Adj Close | 수정 종가 | | Volume | 거래량 |   <데이터 예시>   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Date | Open | High | Low | Close | Adj Close | Volume | | 2012.01.01 | - | - | - | - | - | - | | 2012.01.02 | - | - | - | - | - | - |   FAANG 데이터 셋은 EDA 결과 전반적으로 동일하게 증가 패턴을 보이고 있으며, 최근 시점에서 급격한 상승을 보이고 있다. 이는 IT 도메인의 특성 상 급변하는 기술 흐름이 시장에 반영된 결과로 보인다. 따라서 우리는 FAANG 데이터 셋이 동일 도메인의 특성을 잘 보여주고 있고, 그 도메인의 특성 상 급변하기에 패턴 탐지에 난이도가 존재해 일반화 성능을 평가하기 적절하다고 생각해 해당 데이터를 모델링에 사용했다.     1. Data preprocessing   모델링에 앞서 목적에 맞는 평가 데이터 셋 구성이 우선되어야 한다. 시계열 데이터의 특성(순서 존재)을 고려했을 때, 일반적으로 적용되는 기법(cross validation 등 구성에 무작위 성이 존재하는 기법)을 그대로 적용하기에는 어려움이 있다. 따라서 이런 특성을 반영함과 동시에 증감 패턴 탐지 목적에 맞는 테스트 데이터를 구성할 필요가 있었다.  5개 기업은 데이터 수집 기간 동안 전반적으로 증가하는 경향을 보여줬다. 하지만 그 기간을 좀 더 나누어 연도별로 보았을 경우, 증가 패턴 만을 보이지는 않았다. 예를 들어, 아래 그래프는 2000년도 한 해동안 Amazon 기업의 수정 종가의 변화를 그린 것이다.    일반화된 성능을 위해서는 주가가 어떤 종류의 추세를 보이더라도 증감을 잘 예측할 수 있어야 한다. 그래서 이를 고려해 전체 기업 별 데이터를 연도별로 나누어 증가 추세와 감소 추세로 구분하여 증감 추세를 모두 포함한 평가 데이터 셋을 구성했다.  연도 별로 증감 추세를 구분하기 위해 기준을 정해야 했는데, 이는 해당 연도의 증가 패턴과 감소 패턴이 감지된 개수를 세서 정했다. 현재 시점으로부터 3개월 이후의 주가가 상승하였으면, 증가 패턴이 하나 감지되었다고 판단하였고, 반대로 현 시점으로부터 3개월 이후의 주가가 감소하였으면, 감소 패턴의 개수를 하나 늘렸다.    이렇게 3개월 단위로 window를 sliding하며 9월까지 감지된 증가 패턴과 감소 패턴의 개수를 세어 유독 증감 패턴의 개수가 한쪽으로 치우친 년도들의 개형을 그래프로 확인한 후, 증가 패턴과 감소 패턴을 각각 4개년씩 총 8개년을 평가 데이터 셋으로 선택했다. 선택한 연도와 해당 연도의 주가 추세는 다음과 같다.   |  |  |  | | --- | --- | --- | | 패턴 | 연도 | 기업 | | 증가 | 2015 | Amazon | | 2017 | Apple | | 2017 | Facebook | | 2017 | Netflix | | 감소 | 2014 | Amazon | | 1985 | Apple | | 2008 | Google | | 2012 | Netflix |   <증가 패턴>            <감소 패턴>          마지막으로 모델의 효율적인 학습을 위해 정규화를 진행하였다. (Min Max Scaler)  머신러닝 - 8. Feature Scaling &amp; Feature Selection  **Ⅱ. 본론**   1. Analyzing previous approaches (Baseline Model) 2. LSTM   2017년 한국 정보 전자 통신기술학회에 발표된 ‘양방향 LSTM 순환신경망 기반 주가 예측 모델’ 논문  에서 제시된 모델을 Baseline model로 삼았다. 우리의 데이터 구성과 동일한 일별 시가, 고가, 저가, 종가, 거래량을 column으로 사용하였으며, 주가 예측 모델에 주로 사용되는 LSTM 구조를 사용하였기 때문에 Baseline model로 선택했다. 해당 논문을 기반으로 구성한 모델 구조는 다음과 같다.    단일 LSTM 층과 FC layer 한 층으로 이루어져있다. 모델을 학습하기 전, 데이터 셋을 LSTM의 모델 구조에 맞게 다시 전처리 해야 할 필요가 있었다.    따라서 window\_size에 맞춰서 rolling segmentation을 진행하였다. 예측 시점으로부터 window\_size만큼까지 이전 시점의 데이터를 모델에 넣어 바로 다음 시점의 값을 예측 하는 방식으로 학습을 진행했다. 하지만 평가를 위해 특정 연도를 학습에서 제외하였기 때문에, 남은 학습데이터에 rolling segmentation을 바로 적용할 시, 연속성에 문제가 발생한다. 예를 들어, 2015년도 Amazon 데이터를 제외하고 해당 기법을 적용하여 데이터를 재구성하면, 2014년도와 2016년도 데이터가 이어져버린다. 따라서 이를 고려해 평가 데이터 셋으로 선택한 년도를 기준으로 앞 뒤로 각각 rolling segmentation을 적용해 다시 데이터 셋을 합쳐주는 방식으로 학습 데이터 셋을 구성했다.   1. 학습과정   하이퍼 파라미터는 다음과 같다.   |  |  | | --- | --- | | hyperparameter |  | | LSTM\_hidden\_size | 100 | | LSTM\_num\_layers | 3 | | LSTM\_window\_size | 19 | | FC\_layer | 200 | | Learning\_rate | 0.01 | | Batch\_size | 64 | | Optimizer | Adam | | Bidirectional | True |   학습은 60 epoch 동안 진행했으며, Learning curve는 다음과 같다.     1. Modeling 2. Time series GAN   2018년 발표된 Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets 논문을 참고하여 구현하였다. 해당 논문에서는 Generator 부분을 LSTM을 이용하여 다음 시점의 데이터를 예측하였다.(T+1 시점) Discriminator에서는 1시점부터 T시점까지의 주가 데이터와 지표 데이터(시가, 고가, 저가 등)와 생성된 예측 값을 input으로 받아 데이터의 진위여부를 구분하게 했다. Discriminator에서는 1d convolution layer를 이용했다.  해당 논문에서 제시된 Loss는 다음과 같다.  기본적으로 일반적인 생성 모델의 학습 방식과 동일하게 학습된다. Loss를 보면, Generator는 Discriminator를 속이기 위해 생성된 데이터를 실제 데이터와 가깝도록 학습이 진행된다. 반대로 Discriminator는 가짜 데이터와 진짜 데이터를 잘 구분할 수 있도록 학습된다.  여기서 추가적으로 제시되는 Loss는 다음과 같다.  는 Generator가 직접적으로 다음 시점의 예측 값에 가까워지도록 학습하는 데 관여한다. 기존 MSE 기반의 loss와 큰 차이는 없는 것이다. 반면, direction prediction loss(dpl loss)는 증감 패턴 탐지 task에 직접적인 영향을 미치는 loss이다. Generator가 생성하는 데이터가 실제 데이터의 증감 방향을 따라가도록 직접적인 constraint를 가한다.  기존 논문에서 제시된 모델 구조에 우리는 Right shift 문제의 해결을 위한 추가적인 기법을 더했다. Right shift 문제는 모델이 다음 시점의 예측 값을 현재 시점과 같을 것이라고 판단하여 전반적인 예측 개형이 실제 값에 비해 오른쪽으로 치우친 형태를 띄게 되는 것을 말한다. 즉, 예측 목표로 되는 시점 바로 이전 시점의 정보만이 반영이 되는 것이다. 따라서 이전 시점의 정보를 모델의 예측 값에 더 반 영 할 수 있다면 해결 될 수 있는 문제라고 보았다. 그러나 LSTM은 구조 상 이전 시점의 정보가 전달은 되지만, 그 과정에서 정보 손실이 발생할 수밖에 없다. 그래서 skip-connection 기법을 적용함으로써 이러한 문제를 해결하고자 했다.  skip-connection 기법은 본래 이미지 데이터를 convolution layer에 적용할 때, 모델 깊이가 깊어질수록 발생할 수 있는 gradient vanishing 문제를 해결하기 위해 layer 간의 연속적인 연결 뿐만 아니라 일정 거리 이상의 layer까지도 정보가 전달 될 수 있도록 하는 기법이다. 이러한 기법은 LSTM을 비롯한 RNN 계열 모델에서도 동일하게 적용할 수 있는데, 여기서는 sequence 간의 연속적인 연결에 더해 멀리 떨어진 시점까지도 정보를 전달할 수 있도록 돕는다. Skip-connection을 적용한 최종 Generator와 1d convolution이 적용된 Discriminator의 구조는 다음과 같다.  <Generator 구조>    <Discriminator 구조>     1. 학습과정   하이퍼 파라미터는 다음과 같다.   |  |  | | --- | --- | | Generator hyperparameter |  | | LSTM\_hidden\_size | 100 | | LSTM\_num\_layers | 3 | | LSTM\_window\_size | 19 | | FC\_layer | 200 | | Learning\_rate | 0.01 | | Lambda\_adv | 0.01 | | Lambda\_p | 20 | | Lambda\_dpl | 0.8 | | Batch\_size | 64 | | Optimizer | Adam | | Betas | (0.9,0.999) |  |  |  | | --- | --- | | Discriminator hyperparameter |  | | Kernel\_size | 3 | | stride | 1 | | Conv1d\_1\_in\_channels | 5 | | Conv1d\_1\_out\_channels | 32 | | Conv1d\_2\_out\_channels | 64 | | Conv1d\_3\_out\_channels | 128 | | FC\_layer\_1\_in\_features | 1792 | | FC\_layer\_1\_out\_features | 192 | | FC\_layer\_1\_out\_features | 1 | | Batch\_size | 64 | | Optimizer | Adam |   학습은 60 epoch 동안 진행했으며, Learning curve는 다음과 같다.     1. 모델 성능 비교 2. 그래프 개형 비교   우선 Right shift 문제가 해결되었는지 평가하기 위해 평가 데이터 셋의 예측 결과에 대한 시각화를 진행했다. 다음 그래프들은 증가 패턴과 감소 패턴 데이터 셋에 대한 LSTM과 GAN 모델의 예측 값 시각화 결과이다.  <LSTM-증가패턴>    <GAN-증가패턴>    <LSTM-감소패턴>    <GAN-감소패턴>    LSTM 모델의 그래프를 보면 증가 패턴과 감소 패턴 모두에서 Right shift 현상이 발생하고 있음을 알 수 있다. 모든 구간에 대해서 예측 값에 대한 그래프가 실제 값에 대한 그래프를 뒤늦게 따라가고 있다.  반면, GAN 모델의 그래프는 Right shift 문제가 상당히 개선되었음을 확인할 수 있었다. 특정 구간에서 Right shift문제가 발생하는 경우도 아직 남아있었지만, 그 시차 차이 역시 LSTM에 비해서 줄어들었다.   1. 성능 지표 비교   우리는 증감 패턴 탐지에 맞는 분류 문제(증가 또는 감소 분류)에 적용되는 지표를 사용했다. 바꾼 평가 지표를 바로 loss로 사용해보기 위해서 GAN에서 나온 예측 값을 기반으로 증감을 분류하는 후속 Xgboost 모델도 실험해보았다. 실험 결과 평가된 성능 지표는 다음과 같다.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 평가 지표 | 패턴 | LSTM | GAN(non-skip connection) | GAN(skip connection) | GAN(skip connection)+Xgboost | | Accuracy | 증가 | 0.462 | 0.464 | 0.703 | 0.571 | | 감소 | 0.507 | 0.511 | 0.712 | 0.615 | | Precision | 증가 | 0.499 | 0.505 | 0.705 | 0.641 | | 감소 | 0.499 | 0.473 | 0.716 | 0.716 | | Recall | 증가 | 0.501 | 0.503 | 0.734 | 0.594 | | 감소 | 0.467 | 0.47 | 0.678 | 0.565 | | F1\_score | 증가 | 0.5 | 0.504 | 0.719 | 0.617 | | 감소 | 0.483 | 0.472 | 0.696 | 0.632 |     단일 LSTM 모델과 GAN 모델의 성능 차이는 크지 않았다. 단일 GAN 모델 역시 LSTM 층을 기반으로 하고 있기 때문에, sequence가 진행되면서 소실되는 정보량이 크기 때문으로 생각한다. 반면에 skip-connection 기법을 적용했을 때, 급격한 성능 향상을 보였다. Skip-connection 기법이 거리가 멀리 떨어진 시점의 정보를 효과적으로 전달 했음을 확인할 수 있었다. 후속 모델로 Xgboost를 적용한 예측 결과는 별로 좋지 못했다. Xgboost를 통과하면서 시계열 정보가 소실되기 때문으로 보인다.  **Ⅲ. 결론**   1. 의의   모델 학습 및 평가 결과 skip-connection을 적용한 GAN 모델의 성능이 가장 좋게 나타났다. 그래프의 개형 역시 LSTM과 비교했을 때, Right shift 문제가 크게 개선된 모습을 확인했다. Dpl loss를 통해 다음 시점의 증감 패턴을 따라갈 수 있도록 추가적인 constraint를 적용했고, skip-connection을 통해 lstm 구조의 단점이 성공적으로 보완됐기 때문이라고 생각한다.   1. 한계점 및 보완점   성능 지표와 그래프 개형에서 모두 유의미한 성과를 보였지만, Right shift문제를 완전히 해결하지 못한 것은 후에 보완해야할 점이다. 또한, IT 도메인과 연관성이 높은 다른 도메인의 데이터나 미국 주식 시장의 전반적인 데이터 등 추가 데이터를 고려하지 못했다.  주식의 절대량이 아닌 편차 값을 이용했다면, 주식의 trend(추세)가 제거되기 때문에 right shift 문제도 어느정도 해결할 수 있을 뿐더러, 증감 패턴 탐지라는 task 자체에도 보다 적절한 target 값이기 때문에 더 좋은 결과를 낼 수 있었을 것이라고 생각한다. |
| **Github:** [skynunu/StockPrediction\_Project\_bigdata: 중앙대학교 소프트웨어공학과 빅데이터 과목 StockPrediction\_Project (github.com)](https://github.com/skynunu/StockPrediction_Project_bigdata) |