안녕하세요, 오토 인코더를 활용한 Anomaly Detection 및 외국인 투자자의 거래량과 주가 간의 상관관계 분석이라는 주제로 이번 프로젝트를 진행하게 된 고지환, 윤혜진, 한지우라고 합니다.

저희는 오토 인토더, LSTM, 결론 및 한계점 순으로 발표를 진행하고자 합니다.

(PPT에 쓰여있는 내용 그대로 말한다)

오토 인코더는 입력 데이터를 가공하여 목표값을 출력하는 방식이 아니기 때문에 딥러닝 중에서도 Unsupervised Learning의 일종입니다.

오토 인코더에는 입력 데이터에 노이즈를 추가하는 Denoising 오토 인코더, 합성곱 연산을 활용하는 Convolutional 오토 인코더 등이 있습니다. 저희는 기본적인 오토 인코더를 활용했습니다.

이런 오토 인코더는 주로 흑백 사진의 컬러 복원, 디지털 풍화가 일어난 이미지 복원, 이미지 압축, 비정상 거래 탐지 등의 목적으로 사용될 수 있습니다.

이해를 돕기 위해 간단한 이미지를 준비했습니다. 왼쪽의 파란색 부분은 인코더, 오른쪽의 주황색 부분은 디코더입니다.

보시면 오토 인코더는 좌우가 대칭이 되는 형태로 이루어져 있습니다. 위의 이미지는 오토 인코더 중에서도 여러 개의 은닉된 계층을 가지는 Stacked Auto Encoder입니다. Input 데이터가 Encoder로 들어가면 데이터는 여러 과정을 걸쳐서 낮은 차원으로 압축이 되게 됩니다. 이때 이 압축된 데이터는 다시 Decoder로 들어가서 복원이 되는 방식입니다. 이 과정에서 일부 데이터가 소실이 되거나 노이즈가 생길 수 있는데, 에포크를 반복하면서 이런 오차값을 최대한 줄여나가는 방식으로 학습합니다.

처음에 저희 조의 진행이 더디었던 여러 이유 중에 하나는 바로 데이터였습니다. 처음에는 기존에 사용했던 Pykrx 라이브러리가 아닌 대신증권이 제공하는 API를 활용해 프로젝트를 진행하려고 했었습니다. Pykrx의 경우 네이버 증권 사이트의 자료를 크롤링해오는 방식이기 때문에 아무래도 증권사 측이 제공하는 데이터가 질이나 양의 측면에 있어서 더 낫지 않을까라고 생각했기 때문입니다. 그러나 여러가지 문제점이 발생했습니다. 우선 데이터를 받아오는 속도가 매우 느렸습니다. Pykrx를 이용하면 20년이 넘는 데이터를 받아오는 데에 1분이 채 걸리지 않았으나, 대신증권의 API는 이에 비교하면 매우 느렸고 증권계좌 등급이 낮아서 한번에 많은 데이터의 열람이 어렵다는 메시지가 계속 등장했습니다. 이 메시지에 확인 버튼을 누르지 않으면 더 이상 데이터를 받아오지 않아 번거로운 일들이 있었습니다. 또 이를 해결하기 위해 time의 sleep 함수를 사용하였으나 그 빈도수만 줄어들었을 뿐 메시지가 계속 등장했고, 데이터를 받아오는 시간은 더더욱 길어져서 문제가 있었습니다. 다음으로 대신 API의 특정 함수들은 일정 기간 이전의 데이터를 받아오는 것을 제한했습니다. 예를 들어 어떤 함수는 한 번에 17일의 데이터만 받아오는 것이 가능했고, 또 어떤 함수는 현재 날짜로부터 일정 기간 이전의 데이터를 불러오는 것이 시스템 상에서 막혀 있었습니다. 보다 다양한 변수를 사용하고 싶었기 때문에 이는 저희에게 큰 걸림돌이 되었습니다. 마지막으로 사용 방법이 복잡하고 에러도 잦았습니다. 처음 세팅을 할 때 시간이 오래 걸렸고, 매번 컴퓨터를 껐다가 킨 후 다시 코드를 돌릴 때마다 복잡한 실행 과정을 거쳤습니다. 그리고 조금 오래 사용을 하고 있으면 어느 순간 서버 연결이 끊기는데, 이 경우 무조건 컴퓨터를 다시 부팅해야 다시 API를 사용할 수 있었습니다. 따라서 저희는 비교적 사용이 간편하고 다양한 데이터를 제공하며 비교적 제약이 없는 Pykrx 라이브러리를 사용해 프로젝트를 진행하게 되었습니다.

저희는 우리나라의 메모리 반도체 회사인 SK하이닉스의 주가 및 데이터를 바탕으로 이번 실험을 진행했습니다. SK하이닉스는 거래량이 많고, 상장 일자도 1996년이기 때문에 데이터 또한 충분해서 저희의 목적에 잘 부합한다고 판단했습니다. 또 반도체 D램에 대한 관심도가 높아 외국인이나 기관 투자가 많이 이루어지는 주식이기 때문에 변동성을 확인하기 좋다고 생각해 SK하이닉스를 선정했습니다. 진행 과정에서 저희는 실험을 크게 2개로 나눠서 오토 인코더에 데이터를 넣었습니다. 먼저 처음에는 최대한 다양한 변수를 넣어 오토 인코더를 실행했습니다. 어떤 변수들은 특정 기간 이전의 데이터가 없었기 때문에, 비교적 짧은 2016년 7월 1일부터 2021년 7월 14일까지의 데이터를 사용했습니다. Input 데이터는 화면에 나와있는 다음과 같습니다. 시가, 고가, 저가, 종가 등의 주가 데이터는 물론 개인과 기관, 외국인의 거래량을 포함했습니다. 또 공매도에 대한 정보도 추가해 정보량을 늘렸습니다.

다음으로는 앞의 실험과는 다르게 정보량을 줄이고, 대신 기간을 늘리는 방법을 선택했습니다. 1999년 1월 6일부터 2021년 7월 19일까지의 데이터를 대상으로 했으며, 이번에는 시가, 고가, 저가 데이터를 빼고 종가 데이터만 사용했습니다. 또 외국인 거래량과 대조를 위해 전체, 개인, 기관의 거래량도 포함시켰습니다. 두번째 실험에서는 LSTM과 결합을 시도했는데, 2016년 1월 28일부터 2021년 7월 19일까지의 종가를 LSTM을 이용해 예측했고, 기존의 대조군의 데이터프레임에서 이 기간 동안의 종가만 예측값으로 바꿔서 다시 한번 오토 인코더를 실행해 실험을 진행했습니다.

오토 인코더에 데이터를 넣기 전에 몇 가지 주의 사항들이 있었는데요, 먼저 각 변수 별로의 최소값과 최대값을 구한 뒤 minmaxscaling을 사용해서 input 데이터를 전부 0에서 1 사이의 숫자로 변환했습니다.

또 오토 인코더의 결과물은 결국 Input 데이터의 재구성이기 때문에 우리가 알고자 하는 오차값을 구하기 위해서 오토 인코더의 결과값과 Input 데이터의 차의 절댓값을 이후 분석에 활용했습니다. 이 오차값이 크면 클수록 오토 인코더의 결과물과 Input 데이터 사이의 괴리가 크다는 의미이기 때문에 데이터가 변칙적이라는 의미로 해석할 수 있습니다.

다음은 첫번째 실험의 결과입니다. 산점도를 보시면 파란 점은 외국인 거래량, 주황 점은 개인 거래량, 초록 점은 기관 거래량의 오차값입니다. 이렇게 보면 잘 안 보이죠? 다음 페이지로 넘겨보면

이렇게 선형회귀의 결과가 보이는데요, 검정 선이 외국인 거래량, 파란 선이 개인 거래량, 붉은 선이 기관 거래량입니다. 보시면 개인 거래량의 오차값이랑 기관 거래량의 오차값의 선형회귀 결과는 유사하지만 외국인 거래량의 오차값은 대략 0.1정도 더 큰 것을 확인할 수 있습니다. 더 쉬운 이해를 위해 다음 그래프를 준비했습니다.

기관 거래량을 제외한 그래프입니다. 이렇게 보면 조금 더 보기 편하죠? 육안으로도 파란 점들이 붉은 점들보다 상대적으로 위에 위치하고 있는 모습을 확인할 수 있습니다. 여담으로, 외국인 거래량의 오차값을 선형회귀한 후 그 선형회귀 결과와 괴리가 큰 날짜들을 선별, 이 날로부터 10일, 20일, 30일 이전의 주가와 비교했을 때의 수익률을 확인했습니다. 또 외국인, 기관, 개인의 DTW 결과도 체크했습니다. DTW 결과의 경우 거의 항상 외국인의 수치가 가장 낮았지만, 수익률의 경우 유의미한 결과가 나오지는 않았습니다. 따라서 향후 연구를 이어서 진행할 경우 수익률 그 자체보다는 그래프의 패턴을 분석하는 등의 다른 방안을 활용할 수 있을 것으로 보입니다.

다음의 그래프 중 파란 그래프는 1번째 실험 기간동안의 종가를 MinMaxScaling한 후 그래프로 나타낸 데이터입니다. 그리고 빨간 그래프는 오토 인코더의 결과값 중 종가의 오차값을, 마찬가지로 MinMaxScaling한 후 나타낸 것입니다. 얼핏 보면 두 그래프의 형태가 유사하게 생겼지만 DTW를 이용해서 Weight를 계산해 본 결과 223이라는, 생각보다 큰 결과값이 나왔습니다.

반면에 이 그래프는 파란 선은 앞의 슬라이드와 동일하게 실제 종가의 그래프이지만, 빨간 선은 1번째 실험 기간동안의 오차값을 날짜별로 합산한 데이터를 그래프로 나타낸 것입니다. 이 두 그래프 사이의 DTW는 159로 앞의 결과보다 상대적으로 작은 수치를 보였습니다.

이 그래프는 2번째 실험의 그래프입니다. 2번째 실험에서 주목할 점은, SK하이닉스의 감자가 2003년에 진행이 되어 그 부근의 데이터가 비정상적으로 치솟는 모습을 보였다는 점입니다. 또 그 이후의 데이터는 대부분 수평선 모양의 일정한 모습을 유지했습니다. 따라서 저희는 예측값을 넣은 2016년 데이터 위주로 실험 결과를 확인했습니다. 다음 중 파란 점이 나타내는 것은 원래 종가를 넣은 오토 인코더 종가 오차값이고, 주황 점은 예측값의 오토 인코더 오차값입니다. 초반부에는 기존 데이터와 비슷한 형태를 띄고 있으나 막판에 오차값이 치솟는 모양을 확인할 수 있습니다.

선형회귀 결과는 다음과 같습니다.