Assignment#2
코스피 지수 최저점 예측

국민대학교
소프트웨어학부
20191631
윤민상

1. 주제 선정 동기 및 문제 설명

제가 선택한 주제는 Transformer를 사용하여 코스피 지수의 최저점을 예측하는 것입니다. 예전부터 비트코인과 같은 가상화폐, 주식에 관심이 있었고 실제로 많은 투자를 진행했습니다. 최근까지도 투자를 이어오고 있는 상황인데, 코스피 시장에 있는 종목들 중 하나에서 큰 손해를 보고 있습니다. 매일 주가를 확인하며 이 종목을 계속 보유하고 있는 것이 맞을지, 아니면 지금이라도 매도하는 것이 맞을지 많은 고민을 합니다. Transformer를 통해 코스피 지수의 최저점을 예측해본다면 관심 분야와 겹쳐 과제를 잘 진행할 수 있을 것 같았고, 신뢰할 수 있는 결과값이 나온다면 제 고민을 해결할 수 있을 것이라는 생각이 들어 주제로 선정했습니다. 실제 과제 제출날 직전의 코스피 지수를 확인할 수 있는 6/2일까지의 코스피 지수 데이터로 학습을 완료한 모델을 통해 다음 날 코스피 지수의 최저점을 예측해보았을 때, 현재의 값보다 높다면 종목을 계속 보유하고 있는 것이 좋다는 판단을 내릴 수 있을 것입니다. 반대로 현재의 값보다 다음 날의 지수가 낮게 나왔을 때는 종목을 지금이라도 매도하는것이 맞다는 판단을 내릴 수 있을 것입니다.

2. Code의 작동 원리 및 전체 구조

제가 제출한 Code의 전체 구조는 data의 전처리를 위한 코드인 preprocessing.py, 모델의 학습을 위한 코드인 train.py, 학습에 사용할 data directory가 있습니다. data directory의 내부에는 데이터의 원본값이 담긴 raw.csv 파일과 preprocessing.py를 통해 데이터 전처리를 진행해준 minmax.csv 파일이 있습니다. csv data 파일들에는 그 날의 시장이 열렸을 때와 닫혔을 때의 코스피 지수, 그 날 코스피 지수의 최대값과 최소값, 그 날의 거래량, 변화율 값이 담겨 있습니다. 그리고 사양 문제로 학습이 돌아가지 않을 것

을 대비하여 미리 전처리해준 data 파일과 함께 Colab 환경에서 학습을 진행하기 위한 ipynb 파일이 있습니다.
 우선 preprocessing.py 파일에서는 min-max 방식을 사용하여 데이터 전처리를 진행하여 줍니다. min-max 방식은 feature 값의 범위를 0~1로 조정하는 방법으로, 아래 첨부 사진과 같은 수식으로 구현됩니다.

$$x_{scaled} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

data directory에 있는 raw.csv 파일은 코스피 지수가 처음 시작된 1981년 05월 01일부터 현재까지의 데이터가 들어있습니다. raw.csv를 min-max 방식을 통해 전처리된 데이터는 0과 1사이의 값만으로 구성되어 minmax.csv 파일로 저장됩니다. 이 파일은 이후 train.py 에서 학습을 진행할때 실제로 사용할 데이터입니다.
 train.py 에서는 minmax.csv 데이터와 Transformer 방식을 적용하여 실제 학습을 진행합니다. 30일동안의 데이터를 입력으로 사용하여 31일째의 코스피 지수 저점의 예상값을 출력으로 알려주는 구조입니다. 모델에 학습이 진행될때 매 epoch마다 loss값이 출력되어 학습이 잘 되어가는지 확인할 수 있도록 하였고, 최종적으로는 오늘의 최저 코스피 지수와 내일의 예측 최저 코스피 지수를 출력해주도록 하였습니다. min-max 방식으로 데이터를 전처리해주었기 때문에 코스피 지수가 0~1 사이의 값으로 출력됩니다. 이를 좀 더 편하게 확인하기 위해 오늘과 내일의 최저 코스피 지수를 float 형으로 동시에 출력하도록 하였고, 사용자는 이 두개의 값을 비교하여 자신의 투자를 유지할지 매도할지에 대해 판단할 때 도움을 받을 수 있습니다.
 전체적인 코드의 구조와 사용하는 data는 Assignment1에 사용하였던 것과 똑같지만, train.py 에서 LSTM 방식이 아

닌 Transformer 방식을 사용하여 학습을 진행하였다는 점이 다릅니다.

3. 결과 분석

처음엔 Colab 환경에서 epoch을 LSTM 방식을 사용하였을 때와 같은 1500으로 설정한 후 코드를 실행한 결과, loss의 값이 너무 높게 출력되었습니다. 아직 학습이 완벽히 진행되기엔 epoch이 부족하다고 생각되어, overfitting이 발생하지 않을 정도의 적당한 epoch 값을 찾기 위해 계속 실험을 진행하였습니다. 최종적으로 epoch을 5000으로 설정하였을 때가 가장 좋은 결과값을 보여주었습니다. overfitting이 발생하지 않고, loss 값과 다음날의 코스피 지수 최저값 또한 괜찮게 예상되었습니다. epoch 값이 5000일 때, 처음에는 약 0.2796로 시작했던 loss가 학습을 진행하며 마지막엔 아래 사진과 같이 약 0.0006까지 떨어진 걸 확인하였습니다.

loss 자체는 LSTM을 사용하였을 때의 최종값보다 약간 더 높지만, 첫 epoch의 loss와 최종 epoch에서의 loss를 비교하였을 때의 loss의 감소폭은 Transformer 방식에서가 더 크다는 점을 알 수 있습니다. 그리고 0.0005라는 loss 값의 차이는 사실 무의미한 차이라고 생각하였습니다. 개인적인 판단이지만 최종 loss 값은 비록 LSTM이 더 낮아도 결과값 까지 비교하였을 때 Transformer가 더 정확한 예측값을 보여주었기 때문에 코스피 지수 예측에서는 Transformer 모델이 더

좋은 성능을 보여주었다고 생각합니다. 아래 사진과 같이 오늘의 최저 코스피 지수와 다음날의 예상 최저 코스피 지수를 비교하였을때 Transformer 방식이 더 좋은 결과를 도출해냈다는 것을 알 수 있습니다. Transformer 방식으로 학습을 진행하였을때 내일의 예측 최저값은 0.9923 으로 출력되었습니다.

```
오늘의 최저값 :  tensor([0.9897])  / 내일의 예측 최저값 :  tensor([[0.9923]], device='cuda:0')
```

Assign1과 마찬가지로 비례식을 사용하여 다음날의 예측 최저값을 역변환해 보았습니다.

0.9897 : 0.9923 = 3210.31 : 예측 최저값

내항의 곱과 외항의 곱의 값이 같다는 원리를 사용하였을 때 예측 최저값은 약 3218.74가 나옵니다. 어제의 최저값보다 8.43 높은 값으로, 현재 가지고 있는 종목을 계속 보유하고 있는 것이 이득이라는 판단을 도출해낼 수 있습니다. LSTM 방식때와는 다르게 실제로 가능한 범위의 코스피 지수 변화가 이루어졌고, 실제 그 다음날의 코스피 데이터를 확인해보았을 때 3222.98이었음을 확인할 수 있습니다. LSTM을 통해 얻은 예측 최저값은 전날보다 감소하여 주식을 매도하는 것이 좋다 판단되었는데, Transformer를 통하여 얻은 예측 최저값은 전날보다 증가하여 실제 데이터와 마찬가지로 주식을 보유하고 있는게 더 좋다는 결론이 나왔습니다. LSTM과 다르게 Transformer 모델은 예측값이 실제값과 거의 유사하게 나왔고, 코스피 지수의 최저값이 증가할 것인지 감소할 것인지 또한 맞게 예측하였습니다. 따라서 믿을만한 결과를 도출해내는데에 성공하였다고 판단했습니다.

비록 코스피 지수를 결정하는 데에 외부적인 요인 또한 있다는 것을 감안해도, 예측값을 얻어낸 시기 근처에는 외부적인 요인이 따로 없었기 때문에 Transformer 모델은 꽤 정확한 예측값을 얻었다 생각합니다.

Transformer의 경우 LSTM 보다 시계열 데이터의 긴 의존성을 파악하기 좋은 모델입니다. 따라서 더 멀리 떨어진 시간

간격의 관계도 잘 학습할 수 있습니다. 이번 과제에서 다음날의 코스피 지수를 예측할 때에는 전 30일의 데이터를 사용하여 크게 멀리 떨어진 시간 간격은 아니지만, 결과값을 LSTM과 비교하여 보았을 때 확실히 Transformer 모델이 더욱 잘 학습이 진행되어 좋은 결과값을 도출하였다고 생각합니다. LSTM 같은 경우엔 다음날 코스피 지수 최저값이 증가할지 감소할지도 맞추지 못했지만, Transformer 방식은 확실히 맞추는 동시에 예상값이 실제값과 거의 유사하였기 때문에 그렇게 판단하였습니다.

또한 Transformer 모델은 입력 길이에 큰 영향을 받지 않기 때문에 LSTM에 비해 긴 데이터에도 좋은 성능을 보여줄 수 있습니다. 제가 적용한 데이터는 코스피 지수이기 때문에 1980년대부터의 데이터지만, 다른 주제에 적용하여 더 크고 긴 데이터를 사용하였더라면 Transformer의 장점이 더 부각될 수 있었을 것이라 생각합니다.

LSTM 모델과 비교하였을때 Transformer 모델에서는 꽤 정확한 예측 결과값을 얻을 수 있었고, 앞으로도 다른 주제를 선정해 Transformer 모델을 통한 문제 해결을 진행해보고 싶다는 생각이 들었습니다. 이번 Assign 1, 2를 진행하며 많은 것을 배울 수 있었고, 스스로 성장할 수 있었던 과제였다고 생각합니다.