Assignment #1

RNN 모델과 LSTM 모델을 활용한 Apple 주가 예측

소속 : 소프트웨어융합대학 소프트웨어학부

학번 : 20191610

이름 : 성정규

문제 설명

너 우크라이나 러시아가 전쟁이 일어날거라 생각해?

저는 인공지능 모델이기 때문에 개인적인 의견이나 예측을 할 수 없습니다. 전세계의 정치적, 사회적, 경제적 상황은 매우 복잡하고 예측하기 어렵습니다. 전쟁이 발생할 가능성이 있는지 여부는 현재의 정치적 상황과 국제 관계에 따라 달라지며, 이는 전문가들이 심도있게 분석해야 하는 주제입니다. 정확한 정보와 신뢰할 수 있는 소식 매체를 통해 최신 동향을 파악하시는 것이 좋습니다.

문제에 대한 접근은 인공지능 모델이 전쟁과 같은 외부 상황으로 인한 주가 하락을 예측할 수 있을까? 라는 발상에서 시작하였습니다. 2021년 말, 군 전역 이후 모았던 군 적금이 만기된 저는 모든 돈을 주식에 투자하였습니다. 허나 2022년 2월, 우크라이나와 러시아의 전쟁이 시작되며 저는 투자 수익률 -70%를 기록하며 모았던 돈의 절반 이상을 잃어버리고 말았습니다. 인공지능 모델이 외부 상황으로 인한 주가 변동에 대해 어느정도 예측할 수 있다면 이렇게 모아뒀던 돈을 다 잃는 불상사가 없지 않을까 해서 이 주제를 선정하였습니다.

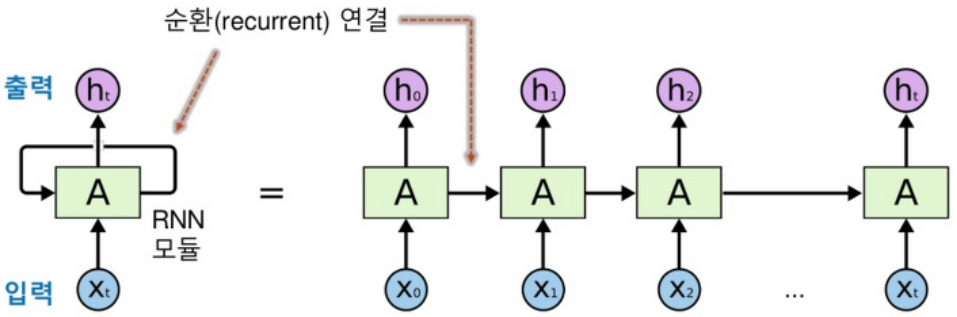
저는 이 주제를 RNN(Recurrent Neural Network)과 LSTM(Long Short-Term Memory)을 사용하여 해결하려 합니다. 2000년 1월 1일부터 2022년 2월 24일까지의 애플 주식 데이터를 수집하여 모델을 학습시키고, 특정 시점부터 모델이 예측하는 주가와 실제 데이터와 비교하여 올바르게 예측했는지 확인할 생각입니다. 2022년 2월 24일까지의 데이터를 사용하는 이유는 이 날 러시아가 우크라이나를 침공하며, 애플의 주식이 급락하였기 때문입니다. 모델이 특정 시점에서부터 예측을 시작하여, 2월 24일에 주가가 급락하는 것까지 예측해낸다면 외부 상황에 의한 주가 변동을 예측할 수 있다는 것이고 이는 굉장히 긍정적인 결과입니다. 결과적으로 RNN 모델과 LSTM 모델로 전부 예측해보고, 어떤 모델이 성능이 더 좋은지, 주가 변동을 예측할 수 있는지 확인할 예정입니다.

해결 과정

1. 데이터 수집 및 전처리: 2000년 1월 1일부터 2022년 2월 24일까지의 애플 주가 데이터를 수집합니다. 수집한 데이터를 정제하고 주가와 외부 상황 데이터를 적절한 형식으로 변환합니다.
2. 모델 구성: RNN과 LSTM으로 순환 신경망 모델을 구성합니다. 이 모델은 입력으로 주가 데이터를 활용하며, 출력으로 다음 시간 단계의 주가를 예측합니다. 모델의 구조, 은닉 상태 크기, LSTM 레이어 수 등을 설정합니다.
3. 모델 학습: 구성한 모델을 학습시킵니다. 주어진 주가 데이터를 사용하여 모델을 훈련하고, 손실 함수와 최적화 알고리즘을 활용하여 가중치를 업데이트합니다. 훈련 데이터와 검증 데이터를 사용하여 모델의 성능을 평가합니다.
4. 주가 예측: 학습된 모델을 활용하여 테스트 데이터에 대한 주가를 예측합니다. 이때 외부 상황 데이터를 함께 입력으로 사용하여 예측을 수행합니다.
5. 성능 평가: 예측 결과를 실제 주가와 비교하여 모델의 성능을 평가합니다. 평가 지표로 평균 절대 오차(MAE) 를 사용합니다. 이를 통해 모델의 예측 능력을 평가하고 향상시키는 방법을 고려합니다.
6. 결과 시각화: 모델의 예측 결과와 실제 데이터를 그래프를 그려서 비교합니다.

작동 원리

RNN(Recurrent Neural Network)

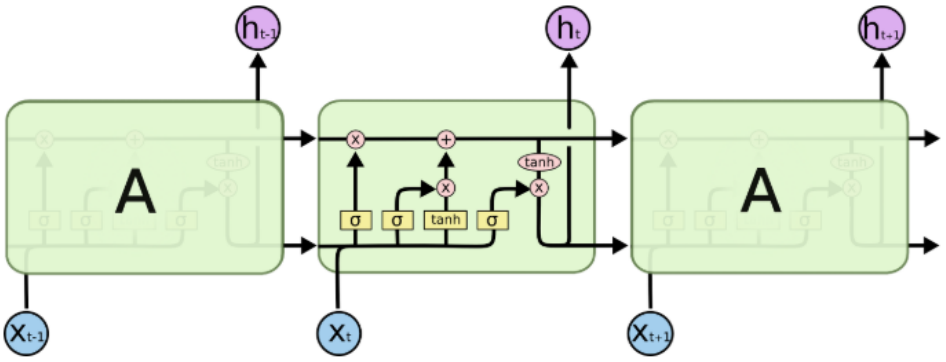


RNN(Recurrent Neural Network)은 순차적인 데이터, 특히 시계열 데이터를 처리하는 데 특화된 신경망 구조입니다. RNN은 이전 단계의 출력을 현재 단계의 입력과 함께

처리하여 순환적인 구조를 가지고 있습니다. 이를 통해 RNN은 이전 단계의 정보를 현재 단계로 전달하며, 시퀀스 데이터의 패턴을 파악할 수 있습니다.

RNN 모델은 입력을 순차적으로 처리하면서 시계열 데이터의 패턴을 학습하고 예측합니다. 은닉 상태를 통해 이전 단계의 정보를 유지하면서 새로운 입력에 대한 응답을 생성합니다. 이를 통해 RNN은 시계열 데이터의 긴 의존성을 처리할 수 있고, 주기성, 추세, 패턴 등과 같은 시계열의 특성을 학습하여 예측에 활용할 수 있습니다.

LSTM(Long Short-Term Memory)



LSTM(Long Short-Term Memory)은 RNN(Recurrent Neural Network)의 한 종류로, 시계열 데이터나 긴 의존성을 가지는 데이터를 처리하는 데 특화된 신경망 구조입니다. LSTM은 RNN과 달리 장기 의존성을 더 잘 학습할 수 있는 기능을 갖추고 있습니다. 이는 LSTM이 기억 셀(memory cell)이라는 구조를 도입하여 이전 정보를 오랫동안 기억할 수 있게 되었기 때문입니다. LSTM 모델은 입력 시퀀스를 순차적으로 처리하면서 각 단계에서 입력 게이트, 삭제 게이트, 출력 게이트의 값을 업데이트하고, 이를 이용하여 현재 기억 셀 값을 계산합니다. 이를 통해 LSTM은 장기 의존성을 학습하고 이전 정보를 오랫동안 유지하여 시계열 데이터의 패턴을 파악하고 예측할 수 있습니다. LSTM은 기존의 RNN보다 긴 의존성을 처리할 수 있으며, 자연어 처리, 음성 인식, 주가 예측 등의 다양한 작업에서 뛰어난 성능을 보여줍니다.

Code의 작동 원리

1. 필요한 라이브러리 및 패키지를 가져옵니다. 이 코드에서는 numpy, pandas, matplotlib, datetime, yfinance, torch 등의 패키지를 사용합니다.
2. 사용할 파이토치의 버전과 디바이스(CPU 또는 CUDA)를 확인하고 설정합니다.

3. 학습에 필요한 하이퍼파라미터를 설정합니다. 이는 학습 에포크 수, 학습률, 입력 크기, 은닉 상태 크기, LSTM 레이어 수, 출력 클래스 수 등을 의미합니다.
4. 주식 데이터를 가져옵니다. 이 코드에서는 yfinance를 사용하여 애플(AAPL) 주식 데이터를 가져옵니다. 데이터는 시작일부터 오늘까지의 종가, 고가, 저가, 시가 등을 포함하고 있습니다.
5. 입력 데이터와 출력 데이터를 준비합니다. 입력 데이터는 'Volume' 열을 제외한 나머지 열로 구성되며, 출력 데이터는 'Close' 열로 구성됩니다. 이후, 데이터를 정규화하기 위해 StandardScaler와 MinMaxScaler를 사용하여 데이터를 변환합니다.
6. 학습 데이터와 테스트 데이터를 분할합니다. 학습 데이터는 처음 4500개의 데이터로 구성되고, 테스트 데이터는 나머지 데이터로 구성됩니다.
7. 데이터를 텐서로 변환하고 모델에 적합한 형태로 재구성합니다. 이 코드에서는 입력 데이터를 3D 텐서로 변환하여 모델에 입력으로 사용합니다.
8. RNN 모델을 정의합니다. 이 모델은 입력 크기, 은닉 상태 크기, LSTM 레이어 수, 출력 클래스 수에 따라 구성됩니다.
   LSTM 모델로 돌리는 코드는 LSTM 모델을 정의합니다.
9. 손실 함수와 최적화 알고리즘을 설정합니다. 이 코드에서는 평균 제곱 오차(MSE)를 손실 함수로 사용하고, Adam 최적화 알고리즘을 선택합니다.
10. 주어진 에포크 수만큼 모델을 학습시킵니다. 학습 데이터를 모델에 입력하여 예측 값을 계산하고, 손실을 계산한 후 역전파와 최적화를 수행합니다. 학습이 진행됨에 따라 손실 값이 출력됩니다.
11. 실제 데이터와 예측 데이터를 비교하고 시각화합니다.

결과 분석

```
Epoch: 100/30000, Loss: 0.30538487434387207
Epoch: 200/30000, Loss: 0.30325847864151
Epoch: 300/30000, Loss: 0.30113765597343445
```

```
Epoch: 29800/30000, Loss: 0.00257907179184258
Epoch: 29900/30000, Loss: 0.0025646535214036703
Epoch: 30000/30000, Loss: 0.0025520636700093746
```

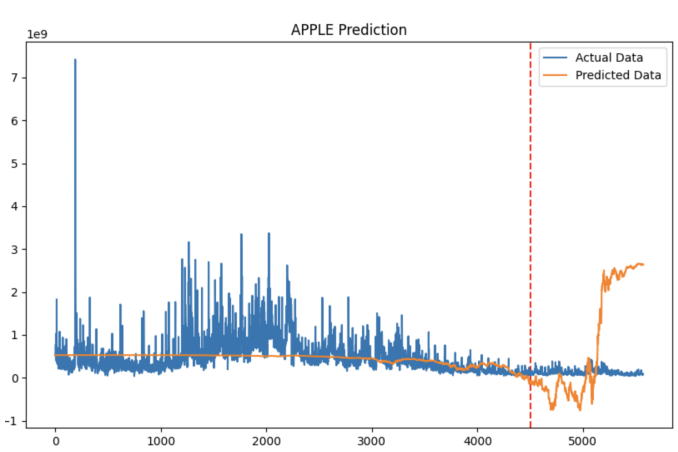
RNN 모델의 300번째 Epoch까지의 Loss와 30000번째 Epoch까지의 Loss입니다.

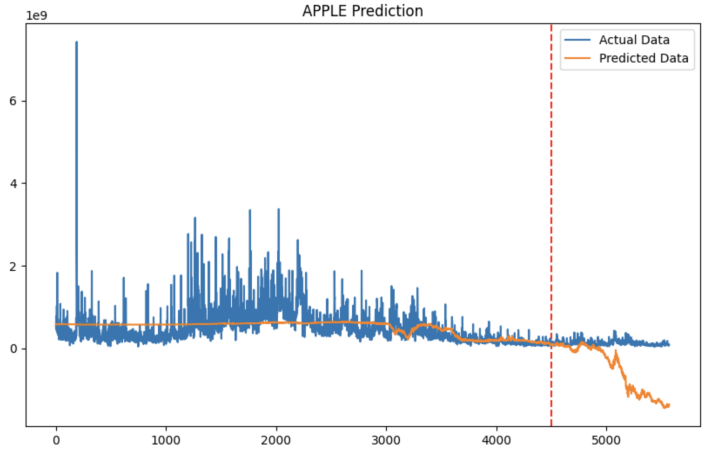
```
Epoch: 100/30000, Loss: 0.016156906262040138
Epoch: 200/30000, Loss: 0.015763625502586365
Epoch: 300/30000, Loss: 0.015391457825899124
```

```
Epoch: 29800/30000, Loss: 0.0023332154378294945
Epoch: 29900/30000, Loss: 0.0023325050715357065
Epoch: 30000/30000, Loss: 0.0023317947052419186
```

LSTM 모델의 300번째 Epoch까지의 Loss와 30000번째 Epoch까지의 Loss입니다.

LSTM 모델이 RNN 모델보다 적은 Loss인 것을 확인할 수 있습니다. 예측값과 실제값의 손실이 LSTM 모델이 더 적으므로 성능이 더 좋다고 판단됩니다.





좌측이 RNN 모델의 실제 데이터와 모델이 예측한 데이터를 비교하여 시각화한 자료이고, 우측이 LSTM 모델의 실제 데이터와 모델이 예측한 데이터를 비교하여 시각화한 자료입니다. 빨간 점선부터가 모델이 예측한 값인데, 그래프의 차이가 많이 나는걸로 보아 외부 상황에 의한 주가 변동을 성공적으로 예측해내지 못한 것으로 보입니다.

한계 및 문제점, 발전 방안

가장 큰 한계는 이 문제의 목표였던 외부 상황에 의한 주가 변동에 대한 예측을 하지 못했다는 것입니다. 그리고 데이터 처리에 문제가 있었는지 정규화한 데이터에서 실제 데이터로 역변환을 제대로 해내지 못하였습니다. 역변환을 위해 다양한 방법을 시도해보았지만 어려웠습니다. 이러한 한계 및 문제점을 해결하기 위해서는 주식 데이터 뿐만 아니라 관련 뉴스, 경제 지표 등의 다른 데이터를 추가하여 모델의 성능을 개선하는 방안이 있다고 생각합니다. 또한 RNN과 LSTM 외에 다른 딥러닝 아키텍쳐를 이용하여 시도해보는 방법이 있다고 생각합니다. 예를 들면 GRU나 Transformer가 있습니다.