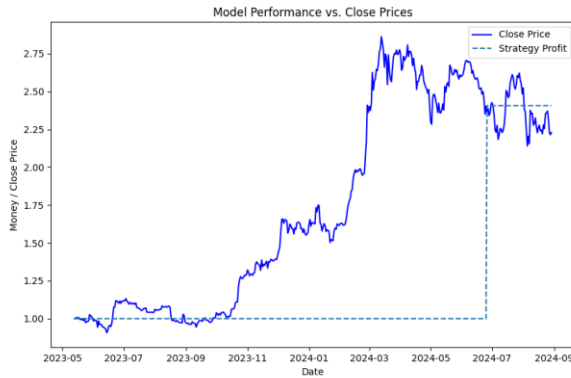


Advanced CNN Report

Ybigta 25TH

Suhyun Choi

1) ViT



Model: model_save/ViT.pt

{'USD': 0, 'KRW-BTC': 0.00030692610219361984, 'v_USD': 24059.323298753472, 'v_KRW-BTC': 0.00030692610219361984, 'buy_count': 2, 'hold_count': 0, 'sell_count': 1}

Profit percentage: 0.35148308246883675

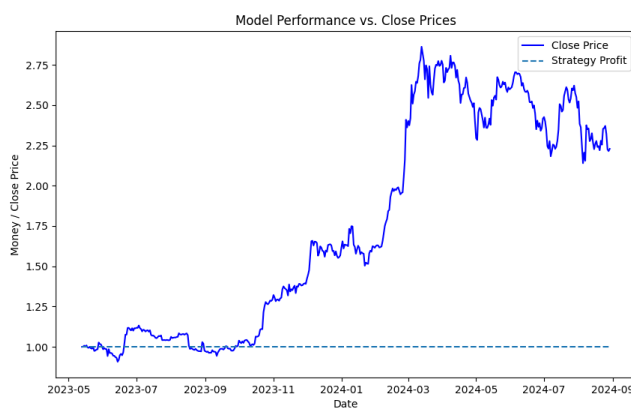
Cumulative return: 1.4061729471700644

Volatility: 1.4047888520530405

Sharpe ratio: 1.0009852691491685

2) CNN

아래 사진은 단순히 Conv layer 2개, activation layer 2개, pooling 2개, fc 2개로 구현했던 것



Model: model_save/CNN.pt

{'USD': 0, 'KRW-BTC': 0.0002751437770011832, 'v_USD': 9999, 'v_KRW-BTC': 0.0002751437770011832, 'buy_count': 1, 'hold_count': 0, 'sell_count': 0}

Profit percentage: -2.499999999997247e-05

Cumulative return: 0.0

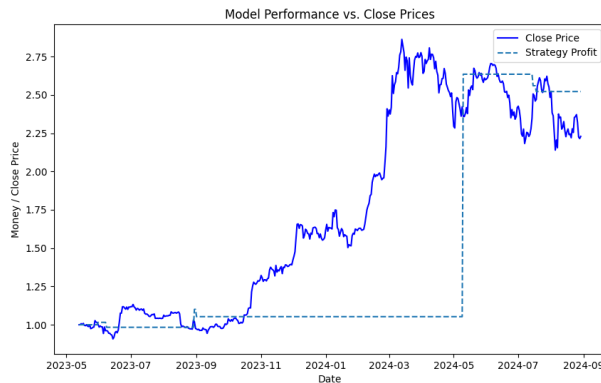
Volatility: 0.0

c:\Users\gydbs\Desktop\24-2\YBIGTA\정규세션\ViT_for_finance\eval.py:155: RuntimeWarning:

```
invalid value encountered in scalar divide
  sharpe = cumulative_return / vol
Sharpe ratio: nan
```

거래 자체를 하지 못하는 모습을 보이니,

Output channels의 수를 늘리고, filter의 크기를 늘렸더니 성능이 소폭 상승했다
또한, dropout layer를 추가하여서 값이 튀는 것을 방지했고, 이에 따라 성능이 많이 상승했다.



```
Model: model_save/CNN.pt
{'USD': 0, 'KRW-BTC': 0.00028065130786334255, 'v_USD': 25216.52001152133, 'v_KRW-BTC': 0.00028065130786334255, 'buy_count': 8, 'hold_count': 0, 'sell_count': 7}
Profit percentage: 0.38041300028803327
Cumulative return: 1.52190419157129
Volatility: 1.5074119617625823
Sharpe ratio: 1.0096139808999276
```

그리고 train.py에서 하이퍼파라미터를 조절해보았는데 크게 유의미한 차이는 드러나지 않았다. Batch_size를 키우면 성능이 소폭 상승하는 것으로 보이는데, 오히려 학습 시간이 증가해 효율성이 떨어졌다.

+)추가적으로 하이퍼 파라미터를

Batch_size : 512

Epoch: 20

Learning_Rate: 0.0001로 했는데 오히려 성능이 엄청나게 저하했다.

하이퍼파라미터의 영향도 분명 있는 듯하다.

종합적 결론

ViT와 CNN의 수익률 차이가 거의 나지 않는다. 하지만, 거래량에서 차이를 보여서 ViT가 복잡하게 설계가 되지 않았다고 볼 수도 있을 것 같다. 하지만, CNN의 경우, 수익률이 적어도 200%는 나올 것이라고 생각했는데, 예상과 달라서 추후에 구조를 정비해봐야할 것 같다. 그리고 비트코인을 hold하는 것이 오히려 딥러닝 방식보다 수익이 좋을 수도 있을 것으로 보이는데, 만일 시간이 더 충분했다면 여러가지 시도를 해볼 수 있었을 것 같다. 근데, 정말 딥러닝 설계를 잘한다면 인간이 하는 것보다 훨씬 나올 것이다.