Stock Prices Generative Adversarial Network

MATH 6397 - Pattern Recognition

Anton Myshak (PSID 2005887)

Introduction

- Problems:
  - How to decrease the timeframe of the asset price chart in conditions of lack of data?
  - How to improve risk management in trading?
  - How to predict price trends for assets better?

- Solution:
  - Create a generative model, which determine the true distribution of arbitrary asset price chart
  - Generate price movement trajectories and calculate the probability of making a profit with the yield of interest to us

Data

- We can use real price charts of arbitrary asset
  - Problem: lack of data to fit the models, extra data is not free

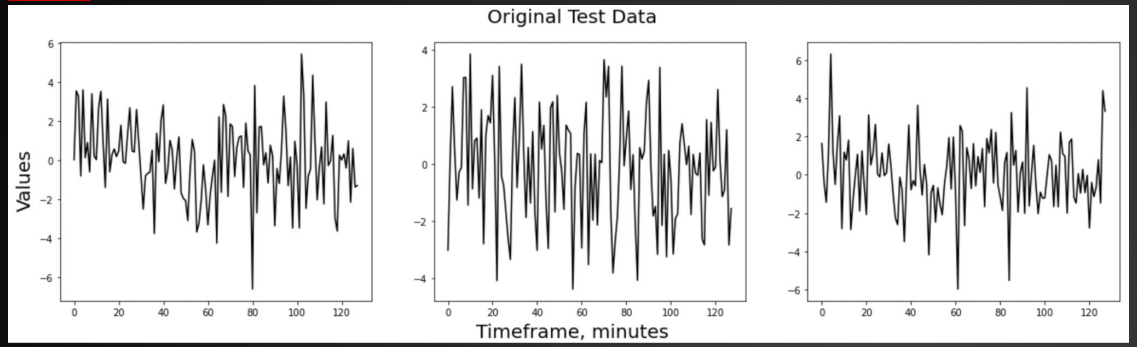
- We can use Ornstein–Uhlenbeck process to generate the data
  - Problem: Although strong, this is still an assumption about the similarity with real data

$$dX_t = -\gamma X_t dt + \sigma W_t \text{ where } X_t = x(t),\ W_t \text{ - Wiener process (Brownian motion)}$$

$$X_t - X_s = -\gamma X_t \Delta t + \sigma(W_t - W_s) \text{ where } W_t - W_s \sim N(0, t-s)$$

$$\Delta X = -\gamma X_t \Delta t + \sigma\sqrt{\Delta t} N(0,1) \text{ where } N(0,1) \text{ - normal distribution}$$

Data



- Stationary Time Series
  -

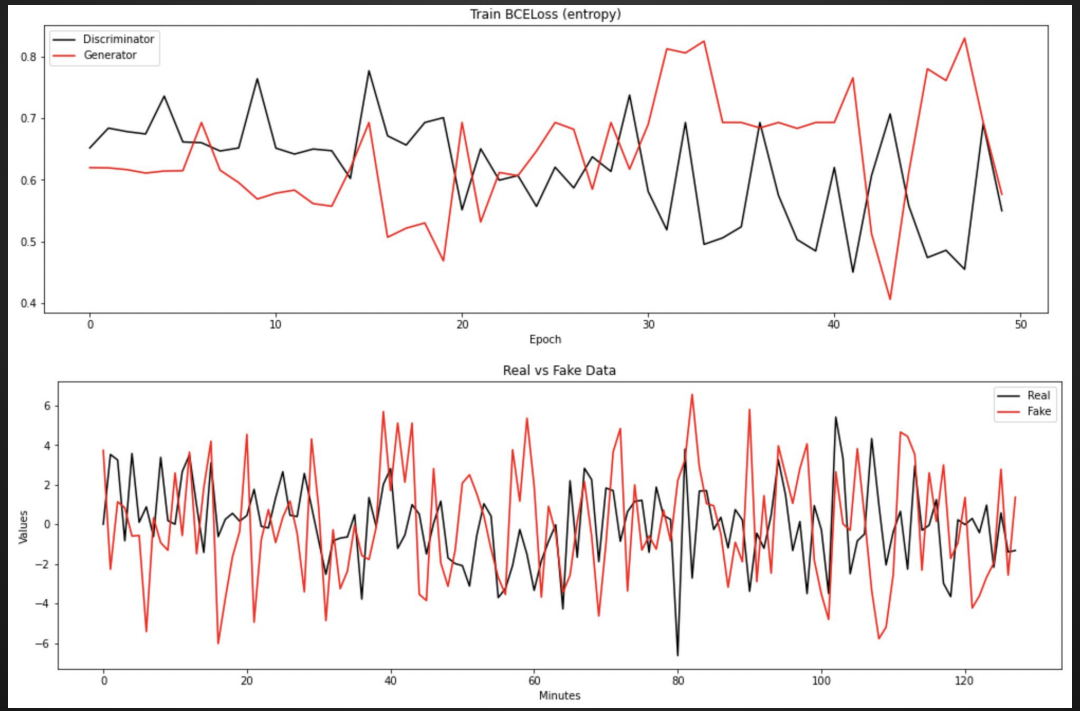
- Financial Time Series
  -

GAN

- Brief idea
- First setup: architecture, parameters
- How to evaluate the results? Introduce the metrics

GAN Results

Results -> conclusion (explain why we need to modify the approach)

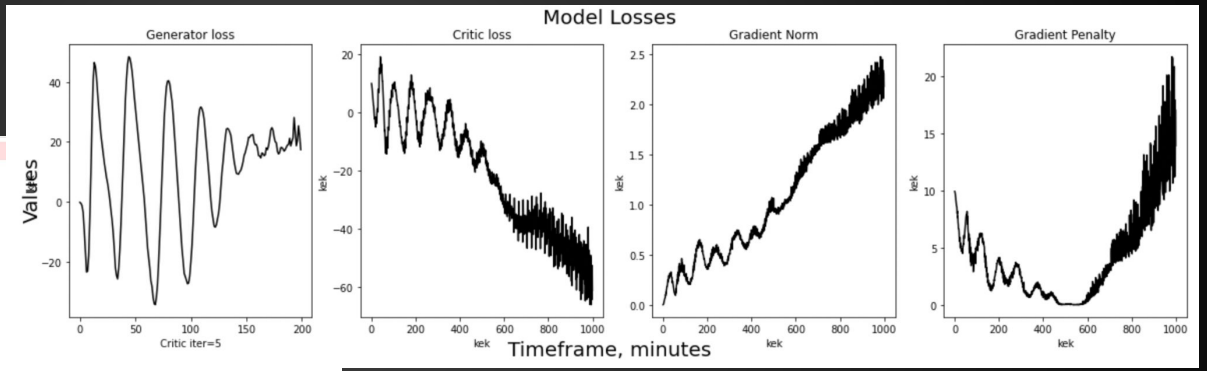


WGAN-GP

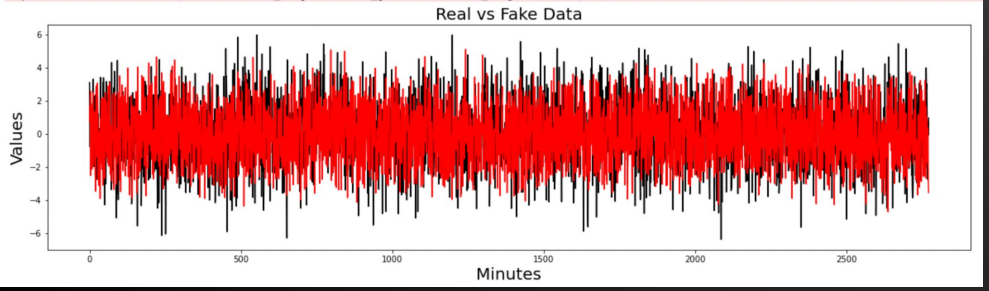
- Brief idea
- First setup: architecture, parameters

WGAN-GP Results

- Results Comparison -> Conclusion



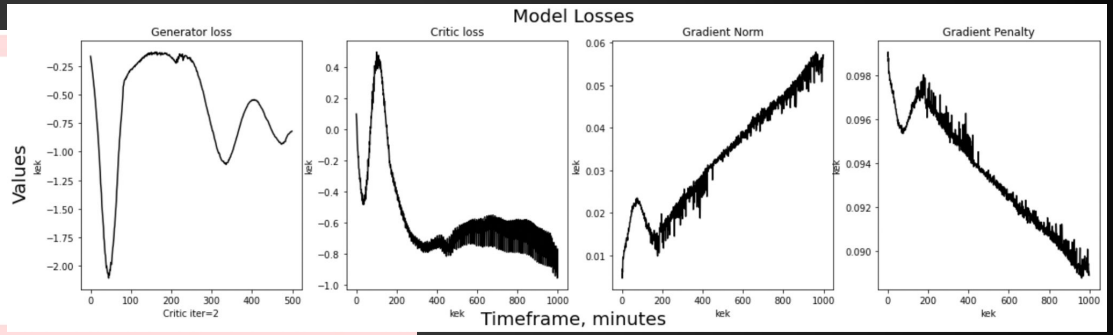

```
100%|██████████| 50/50 [01:18<00:00,  1.58s/it]
1 moment of original data =  0.0
1 moment of generated data =  0.0
Relative error = nan %

2 moment of original data =  3.995978107338854
2 moment of generated data =  3.2164972
Relative error = 19.51 %

3 moment of original data =  -0.239201104050113 86
3 moment of generated data =  0.5413596
Relative error = -326.32 %

4 moment of original data =  46.19699792999312
4 moment of generated data =  24.96563
Relative error = 45.96 %

/Users/Saizt/Documents/USA/UH/Courses/Summer ML (special problem)/my_funcs.py:46: RuntimeWarning: invalid value encountered in double_scalars
  print('Relative error =', round(abs(res_original - res_generated) / res_original * 100, 2), '%')
```



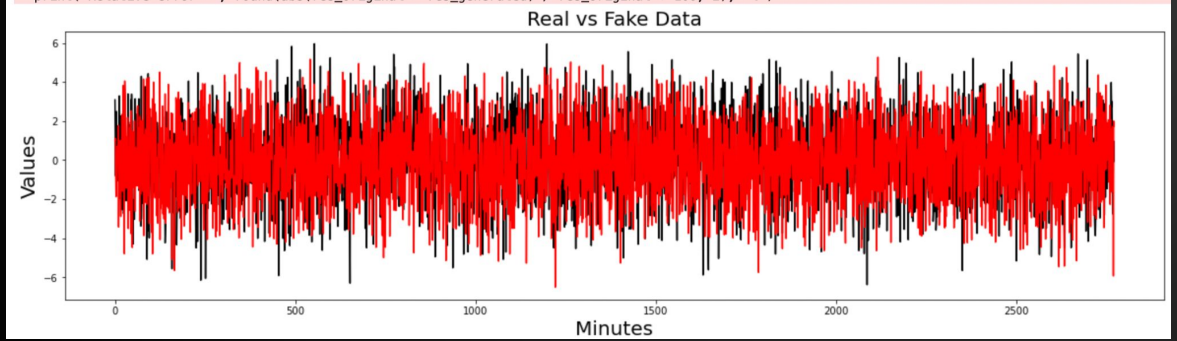
Tuned WGAN-GP Results

- Results Comparison -> Conclusion

```
100%|██████████| 100/100 [02:49<00:00,  1.69s/it]
1 moment of original data =  0.0
1 moment of generated data =  0.0
Relative error = nan %

2 moment of original data =  3.995978107338854
2 moment of generated data =  4.2004843
Relative error = 5.12 %

3 moment of original data =  -0.23920110405011386
3 moment of generated data =  -0.219377
Relative error = -8.29 %

4 moment of original data =  46.19699792999312
4 moment of generated data =  44.192837
Relative error = 4.34 %
```




```
/Users/Saizt/Documents/USA/UH/Courses/Summer ML (special problem)/my_funcs.py:46: RuntimeWarning: invalid value encountered in double_scalars
  print('Relative error =', round(abs(res_original - res_generated) / res_original * 100, 2), '%')
```

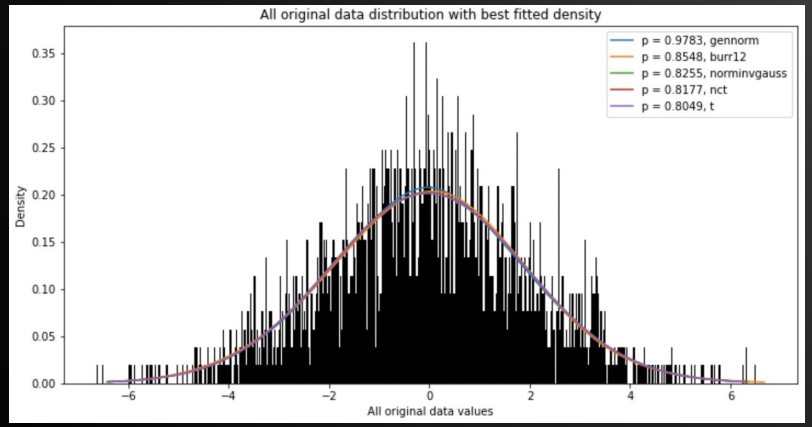


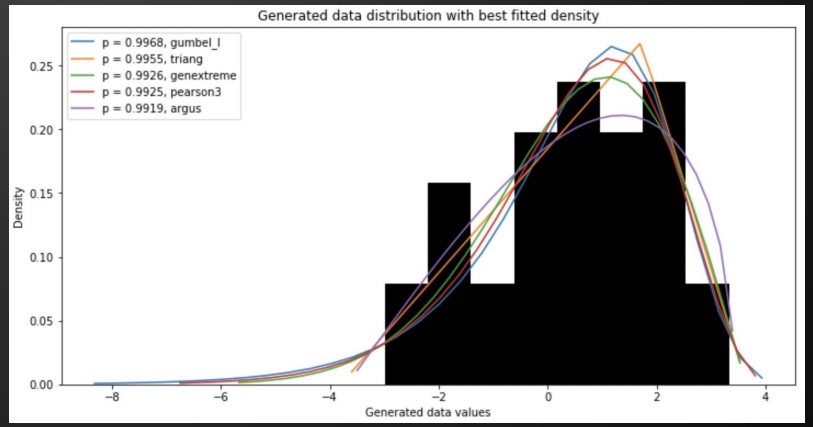
Hypothesis Testing

We know the real distribution of data, so let's check by hypothesis testing will we get something pretty close to what we know?

Let's also test what distribution have our generated data by tuned WGAN-GP

Is it close enough?





Further Work

- How to make generator start with a fixed price value?
- How to improve the model?
  - Implement Conditional WGAN-GP with better tuning
  - Are metrics sufficient enough?
- Check results for non-stationary Time Series ($0>\gamma>>-0.1$, $0<\sigma<2$)
- Check results for real asset's price charts
- Pack the model into algorithm of making decisions
- Test on paper money
- Test on real money

TO BE CONTINUED