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 models, extra data is not free

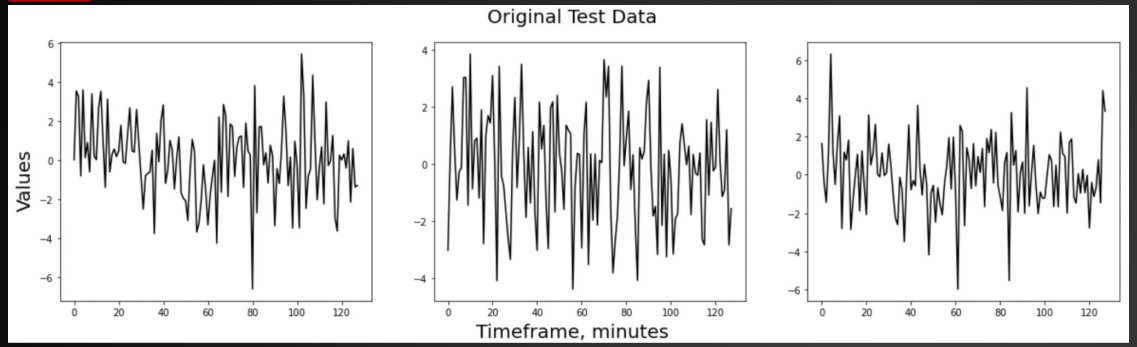
- We can use Ornstein–Uhlenbeck Process to generate the data
  - Problem: it is still an approximation, not real

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

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

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

Data

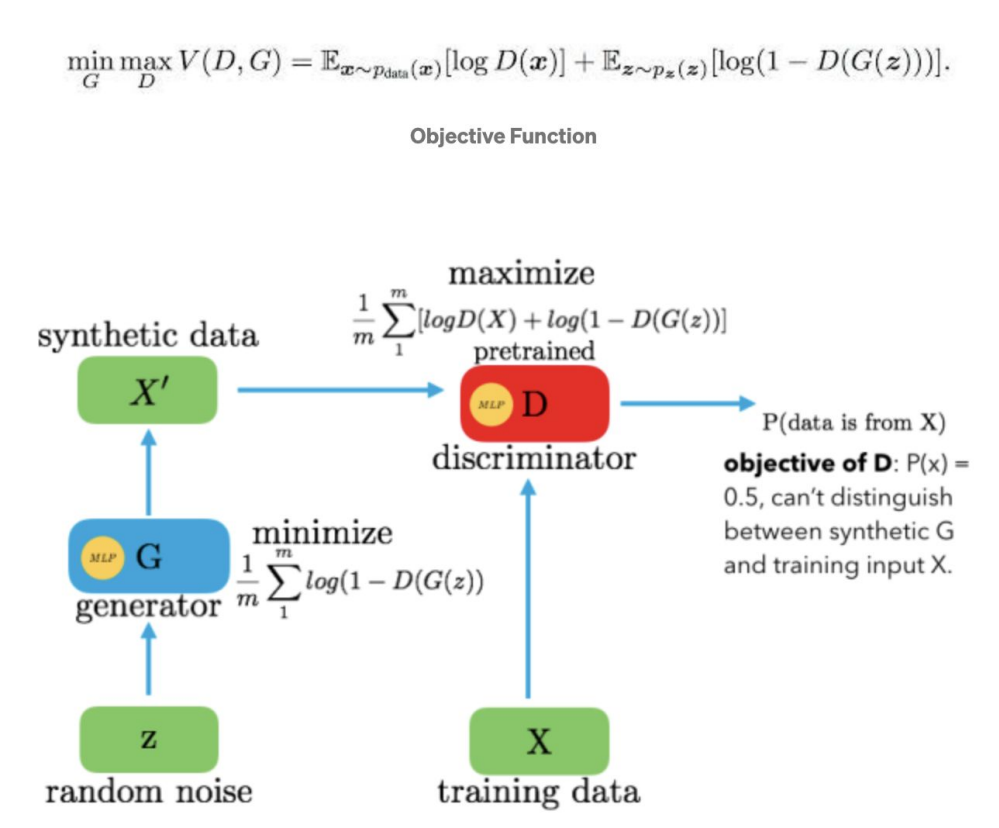


- Stationary Time Series
  - $\gamma=1, \sigma=2$

- Financial Time Series
  - $-0.1<<\gamma<0, 0<\sigma<2$
  - $\gamma=-0.00005, \sigma=2$

GAN

How to construct generator?



GAN Setup

- Architecture

Critic

Linear(128, 64)
LeakyReLU(0.01)
Linear(64, 32)
LeakyReLU(0.01)
Linear(32, 16)
LeakyReLU(0.01)
Linear(16, 8)
LeakyReLU(0.01)
Linear(8, 1)
Sigmoid()

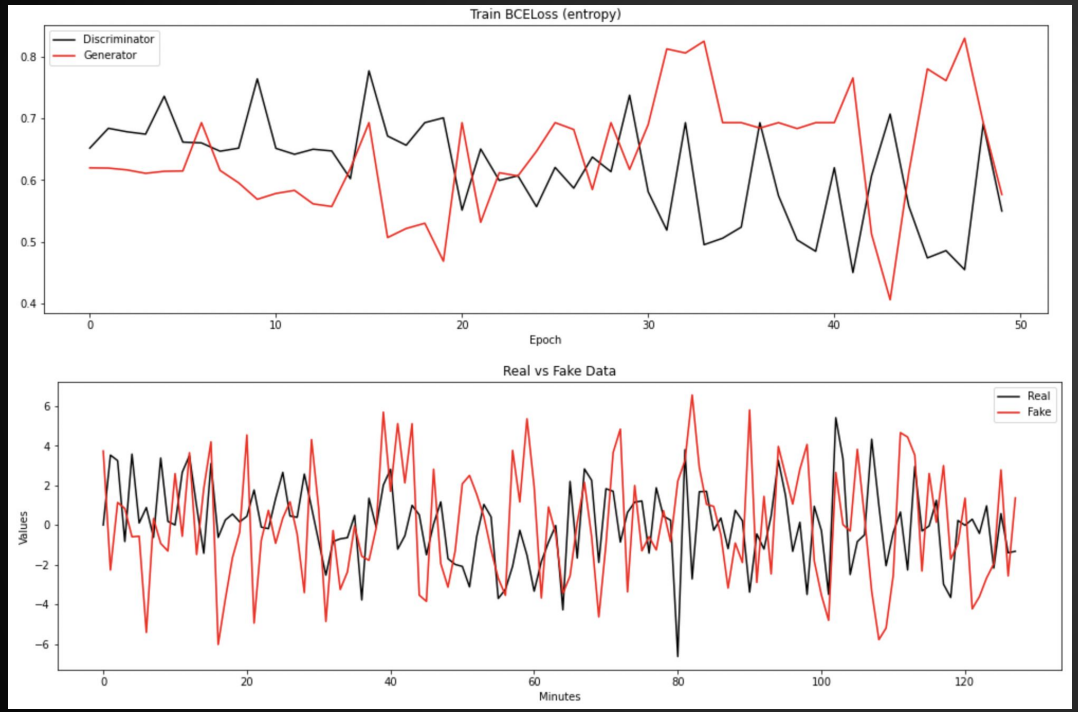
Generator

Linear(128, 256)
Dropout(0.3)
Linear(256, 512)
Linear(512, 256)
Linear(256, 128)

For both NN:

- ADAM optimizer (Stochastic Gradient Descent)
- Binary Cross Entropy Loss

GAN Results



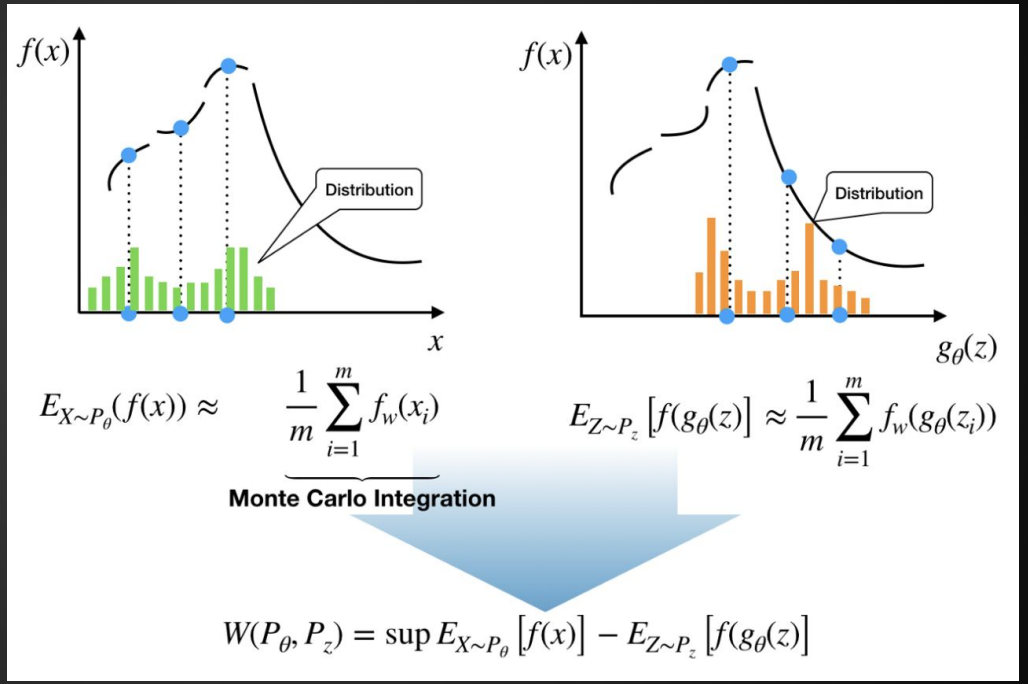
HOW TO EVALUATE?

Our generator should aim to repeat the distribution, so we need to improve our training approach

WGAN-GP

EVALUATION METRIC:

Statistical Moments Comparison



WGAN-GP Setup

- Architecture

For both NN:

- RMSProp optimizer
  (root mean square propagation)
- Wasserstein Distance Loss
- Gradient Penalty

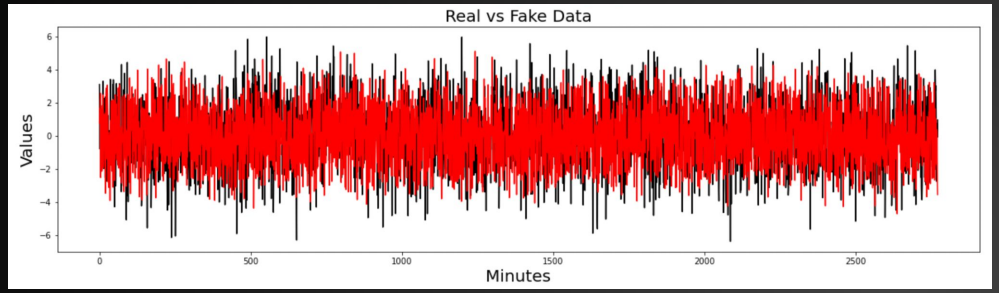
Critic

SpectralNorm(Conv1d(1, 32, 3))
LeakyReLU(0.2)
MaxPool1d(2)
SpectralNorm(Conv1d(32, 32, 3))
LeakyReLU(0.2)
MaxPool1d(2)
SpectralNorm(Conv1d(32, 32, 3))
LeakyReLU(0.2)
Flatten()
Linear(22176, 50)
LeakyReLU(0.2)
Linear(50,15)
LeakyReLU(0.2)
Linear(15, 1)

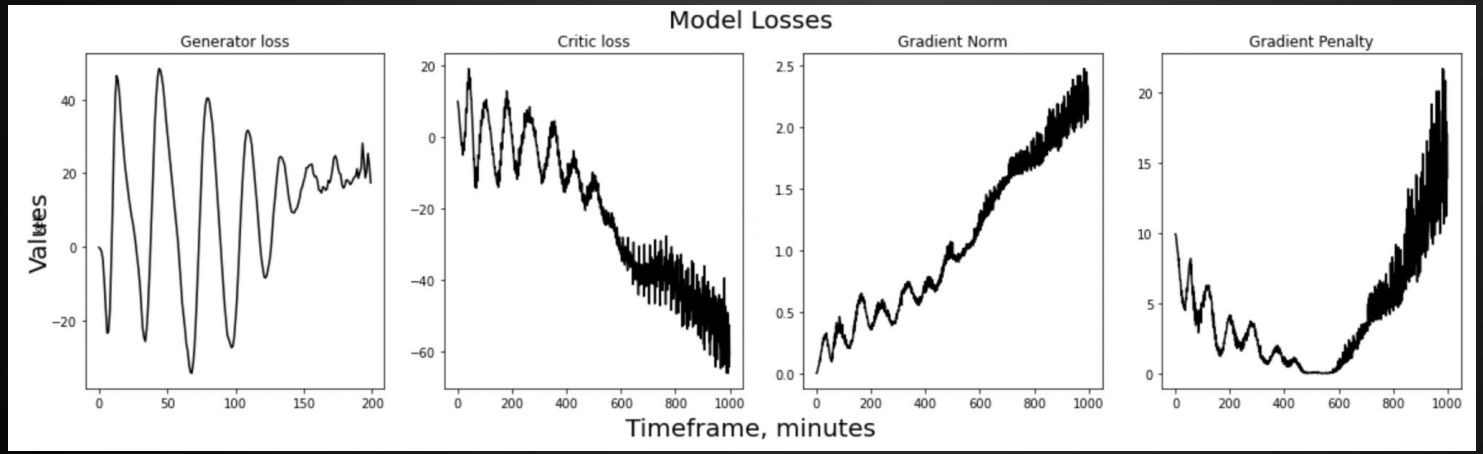
Generator

Linear(50, 2772)
LeakyReLU(0.2)
SpectralNorm(Conv1d(1, 32, 3))
LeakyReLU(0.2)
Upsample(5544)
SpectralNorm(Conv1d(32, 32, 3))
LeakyReLU(0.2)
Upsample(11088)
SpectralNorm(Conv1d(32, 32, 3))
LeakyReLU(0.2)
Upsample(22176)
SpectralNorm(Conv1d(32, 1, 3))
LeakyReLU(0.2)
Linear(22176, 2772)

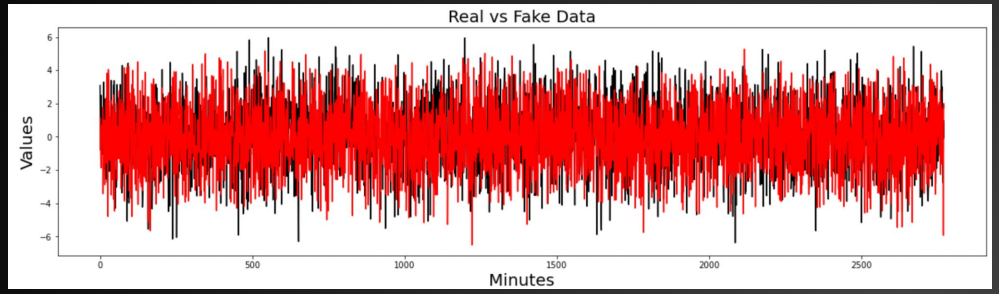
WGAN-GP Results



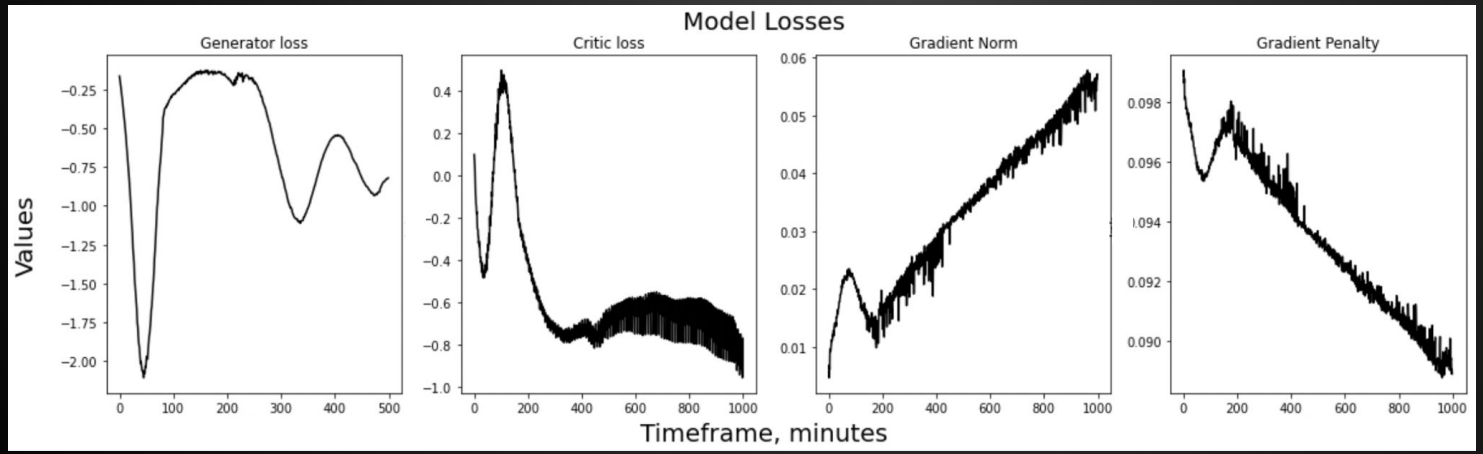

```
1 moment of original data =  0.0
1 moment of generated data =  0.0
Relative error = nan %

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

3 moment of original data =  -0.23920110405011386
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 %
```



Tuned WGAN-GP Results



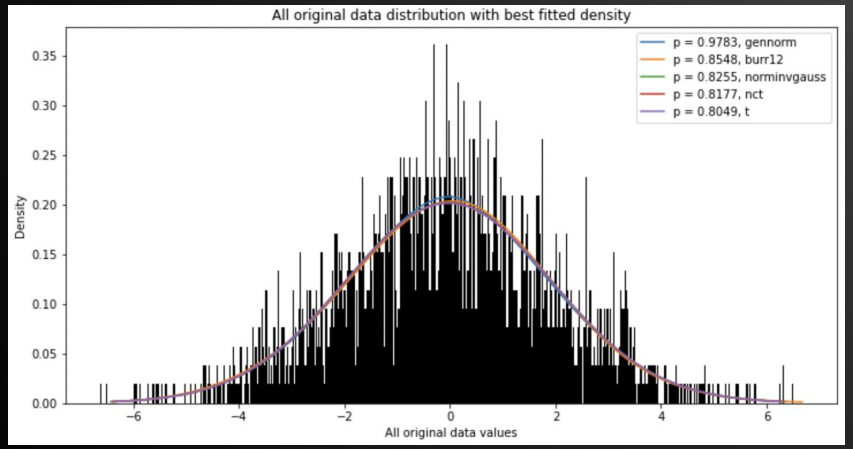

```
1 moment of original data =  0.0
1 moment of generated data =  0.0
Relative error = nan %

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 %
```



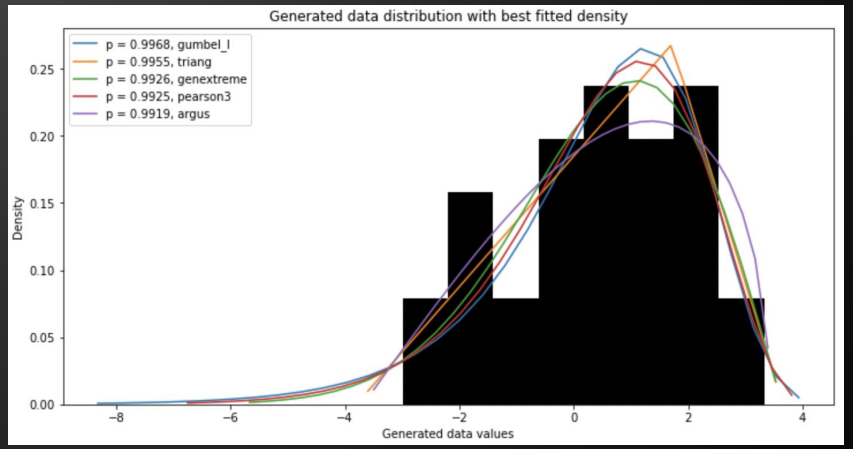
Statistical Hypothesis Testing

- **Kolmogorov-Smirnov Test**

Best fit for train data



Best fit for generated data by Tuned WGAN-GP



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 ($-0.1 < < \gamma < 0$, $0 < \sigma < 2$)
- 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