Stock price prediction using Generative Adversarial Networks

Chen Chen, HungChun Lin

Outline

- Introduction
- Problem Statement
- Data Source
- Data Structure
- Model theory
- Experimental and Result
- Evaluation
- Conclusion

Introduction

Stock price prediction is an interesting and challenging topic, and is a kind of time series forecasting. Many classic algorithms are used in time series forecasting, such as long short-term memory (LSTM) and ARIMA.

- Project goal
  - Compare the basic LSTM and GRU model with GAN, and improve the model to get more accurate prediction.
- Contribution
  - Input different features
  - Compare the different models' performance
  - Improve GAN by adjusting the loss function

Problem Statement

In this paper, we want to use **GAN** to predict the stock price and to see whether the adversarial system can help improve the time series prediction.

- We will utilize the **GANs** to see if it can performs better than traditional **LSTM** and **GRU** model, unlike the traditional GANs, the **GAN** model will be implemented with a **RNN** as a generator and a **CNN** as a discriminator.

- We will utilize the **WGAN-GP** to see if it can improve the result of basic GAN.

Data Source

- Target(predicted stock price): Apple.Inc closing price
- Feature:
    1. The stock price and stock index are from Yahoo Finance
    2. The dollar index is from Fred
    3. News sentiment data are scrapped from SeekingAlpha
    4. Calculated statistical data

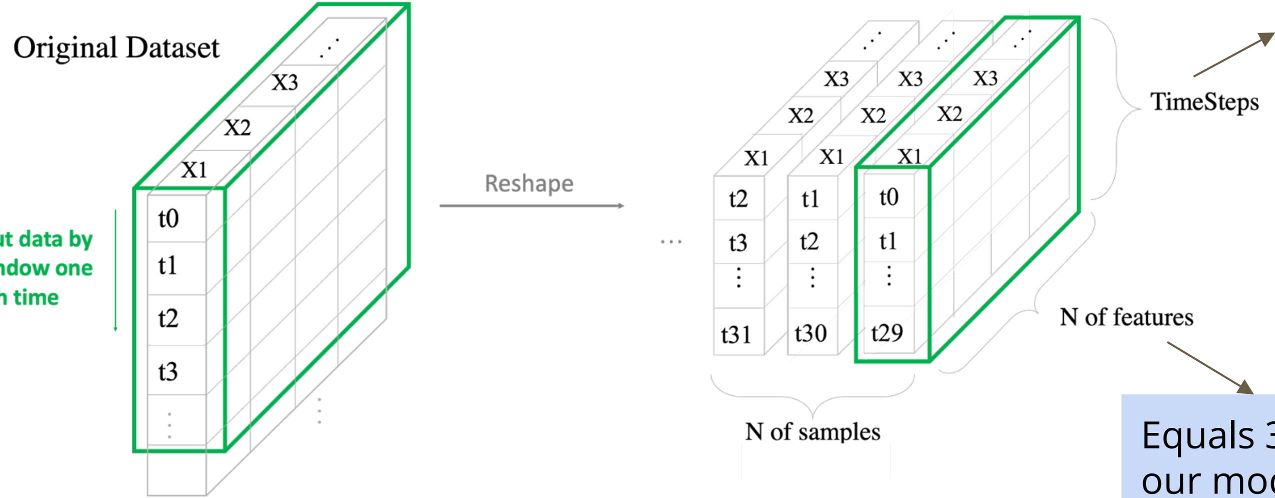
Data Source(cont.)

- Days: 2497
- Features: 36
- Train/Test split: 7 : 3

Feature Name			
Open	Nikki225	Amazon	EMA
High	BSE SENSEX	Goolgle	logmomentum
Low	RUSSELL2000	Microsoft	absolute of 3 comp
Close	HENGSENG	MA7	angle of 3 comp
Volume	SSE	MA21	absolute of 6 comp
NASDAQ	CrudeOil	20SD	angle of 6 comp
NYSE	Gold	MACD	absolute of 9 comp
S&P500	VIX	upper	angle of 9 comp
FTSE100	USD index	Lower	News

Data Structure(input data of generator)

Create the input data by moving the window one step down each time

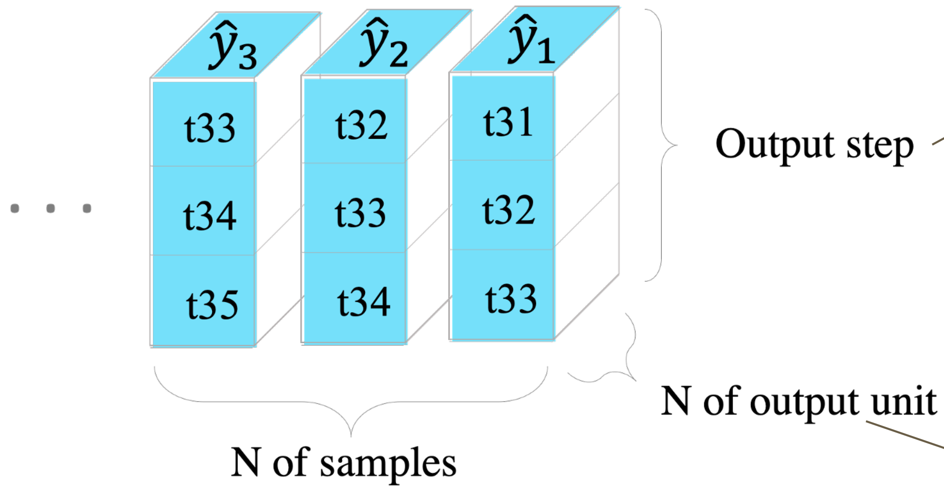


How many historical days data we want to use to make the prediction

Equals 36 in our model

- The original dataset is 2 dimensional, we need to reshape the data to 3 dimensions according to the timesteps.

Data Structure(output data of generator)

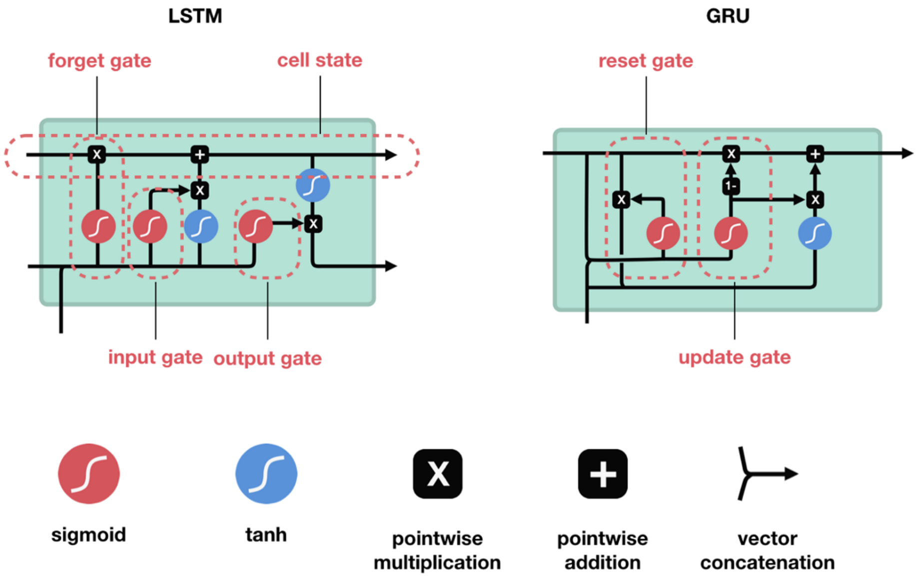


How many days stock price we will predict

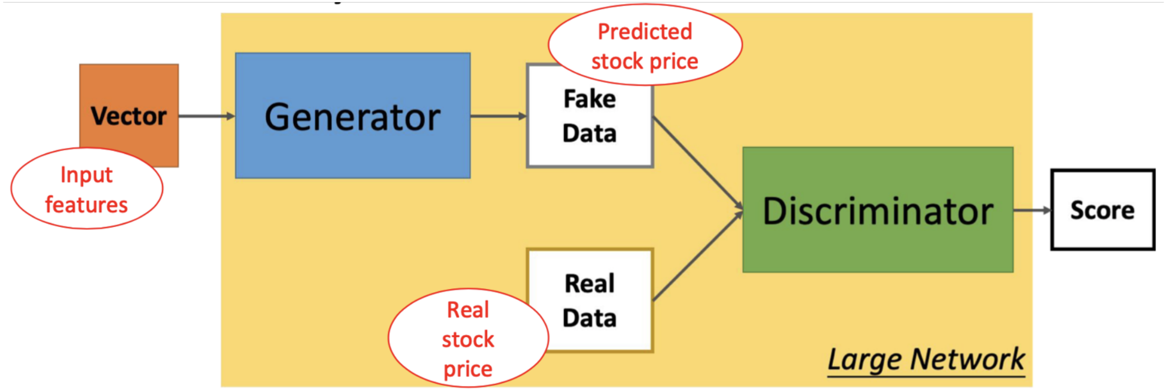
Equals 1 in our model. we only predict closing price

Model Theory (RNN)

- Structure:
  - LSTM(1997): Input, output and forget gate
  - GRU(2014): Reset and update gate
- Samepoints:
  - Prevent the vanishing gradient problem in traditaional RNN
  - Perform well on dealing with sequence of data
- Differentpoints:
  - LSTM has the cell state to store the memory, but GRU only has the hidden state
  - GRU train quick and has lsee parameters



Model Theory(Original GAN)



x : Input for generator
y : Real price from original data
$G(x^i)$: Generated price (fake price)

- GAN basically made up of two competing neural network models
- The **Generator generates** fake data and tries to fool the Discriminator
- The **Discriminator** tries to **distinguish** between the real data and fakedata

Loss function of Descriminator:

$$-\frac{1}{m}\sum_{i=1}^{m} log\, D(y^i)\ -\ \frac{1}{m}\sum_{i=1}^{m}(1 - log\, D(G(x^i)))$$

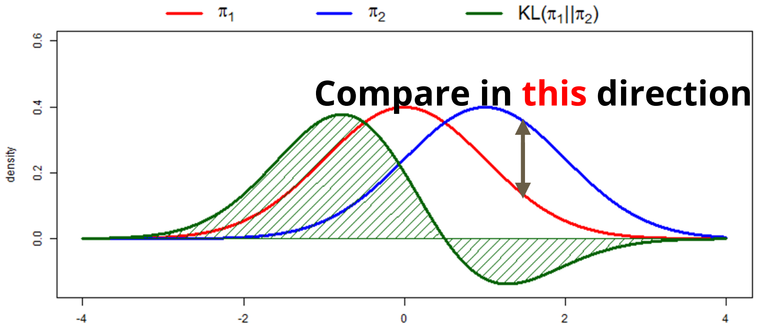
Loss function of Generator:

$$-\frac{1}{m}\sum_{i=1}^{m}(\,log\, D(G(x^i)))$$

Model Theory(Original GAN vs WGAN-GP)

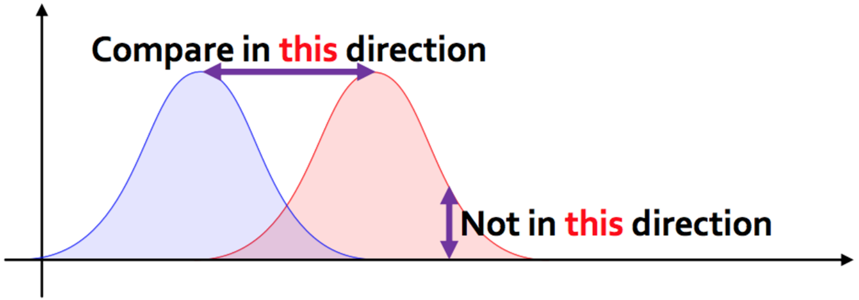
- Loss function: minimize the difference between real distribution and generated distribution

- Origianl GAN: JS-KL divergence



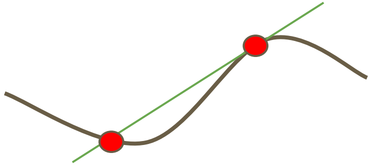
If two distribution have no overlaps, loss is constant.

- WGAN-GP: Wasserstein distance



Gradient descent is valid when two distribution have no overlap.

Model Theory(WGAN-GP)



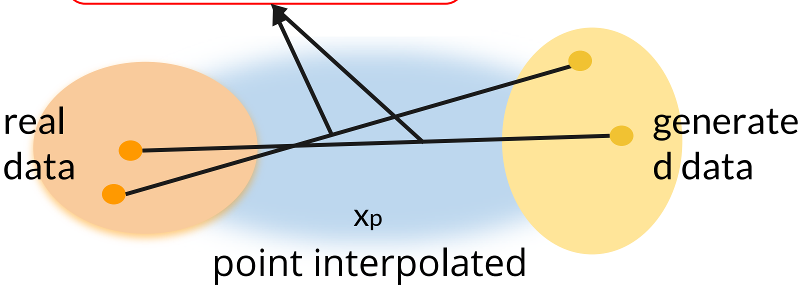
- 1-Lipschitz function: $|f(x_1) - f(x_2)| \leq |x_1 - x_2|$
- WGAN-GP uses gradient penalty to enforce the Lipschitz constraint. A differentiable function f is 1-Lipschitz if and only if it has gradients with norm at most 1 everywhere $\|\nabla f\|_2 \leq 1$
- WGAN-GP loss function:

Discriminator:

$$\frac{1}{m}\sum_{i=1}^{m}\left[D(y^i) - D\left(G(x^i)\right) + \boxed{\lambda(\|\nabla D(x_p)\|_2 - 1)^2}\right]$$

Generator:

$$-\frac{1}{m}\sum_{i=1}^{m} D(G(x^i))$$

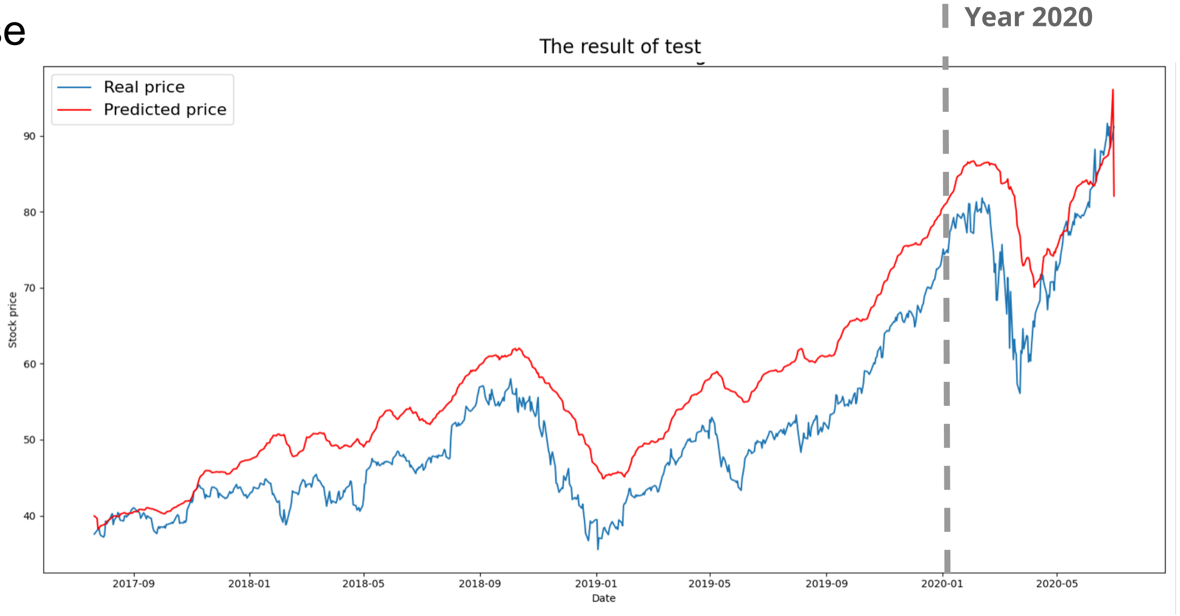


Experimental and Result (LSTM)

- Structure:
  Bidirectional(LSTM) + Dense

- Hypermarameter:
  Batch_size: 64
  Epoch: 50
  Learning _rate: 0.001

- RMSE(include 2020): **6.60**
- RMSE(exclude 2020): **9.42**

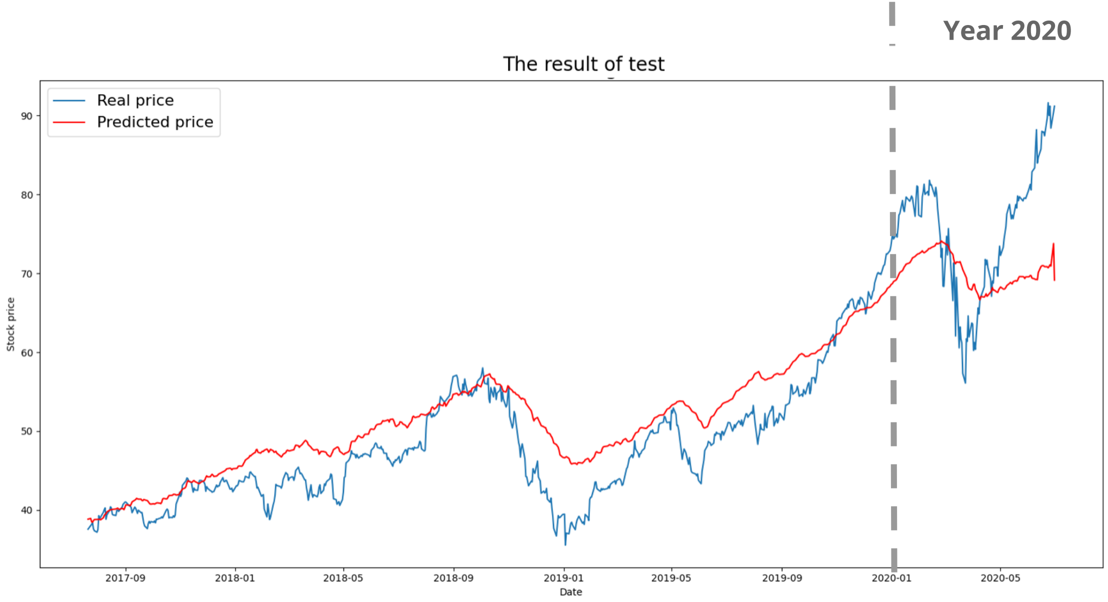


Experimental and Result (GRU)

- Structure:
  GRU + GRU

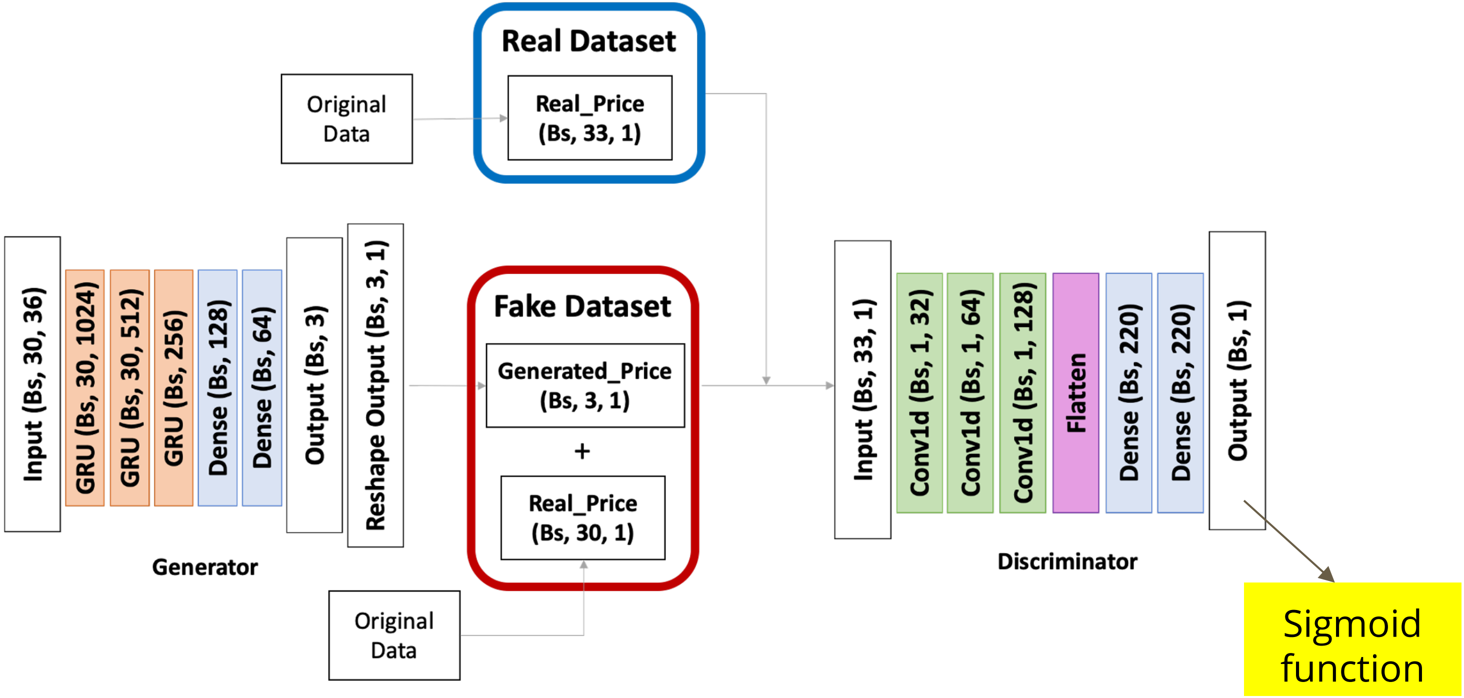
- Hypermarameter:
  Batch_size: 128
  Epoch: 50
  Learning _rate: 0.0001

- RMSE(include 2020): **5.33**
- RMSE(exclude 2020): **4.08**



Experimental and Result (Basic GAN)

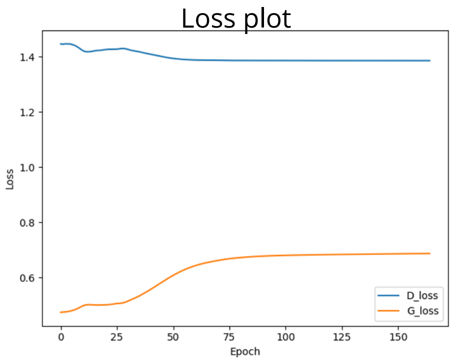
- Structure:

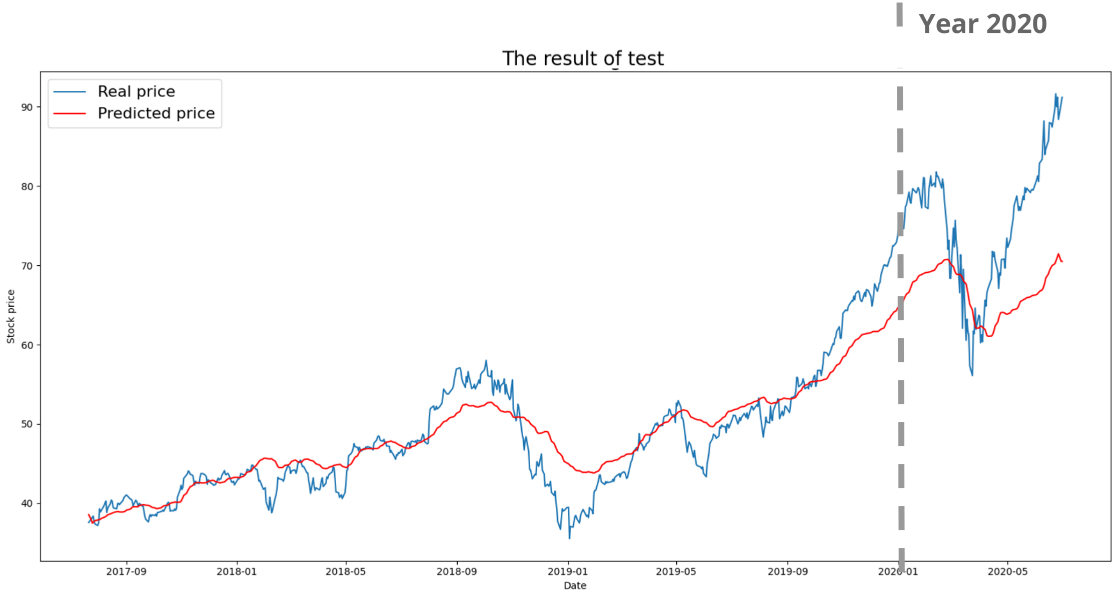


Experimental and Result (Basic GAN)

- Hypermarameter:
  Batch_size: 128
  Epoch: 165
  Learning _rate: 0.00016

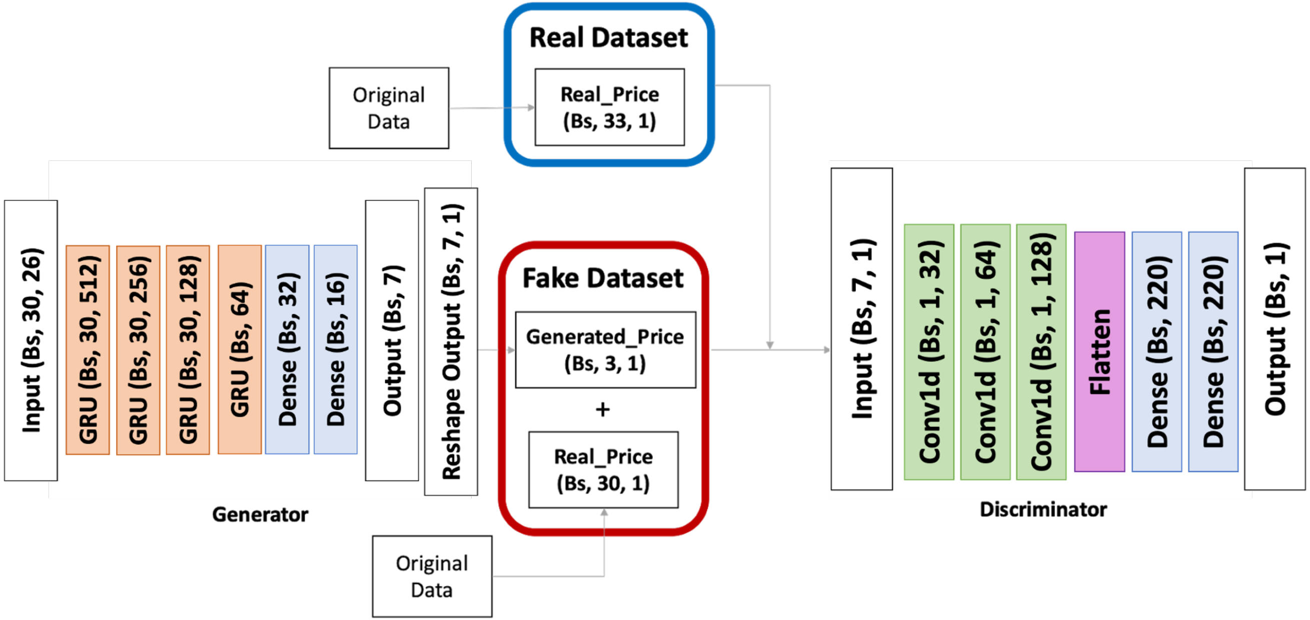
- RMSE(include 2020): **5.36**
- RMSE(exclude 2020): **3.09**





Experimental and Result (WGAN-GP)

- Structure:

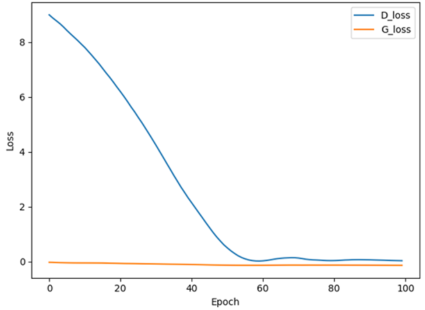


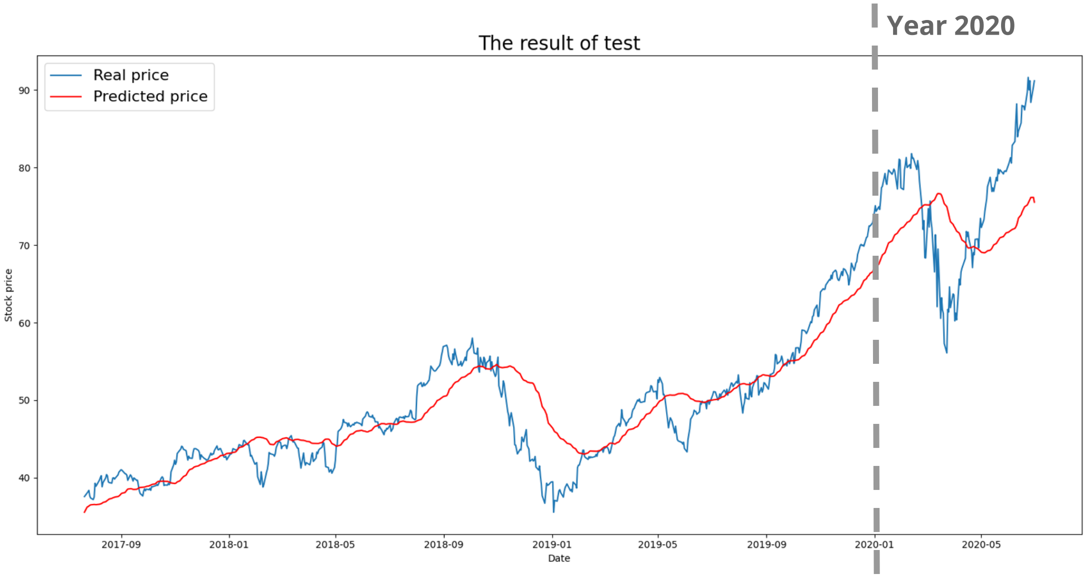
output is a scaler score, without sigmoid compared with original GAN

Experimental and Result (WGAN-GP)

- Hypermarameter:
  Batch_size:128
  Epoch:100
  Learning _rate:0.0001
  Train D once, train G 3 times
- RMSE(include 2020): **4.77**
- RMSE(exclude 2020): **3.88**

Loss plot





Evaluation

	LSTM	GRU	Basic GAN	WGAN-GP
RMSE of Training dataset	1.52	1.00	1.64	1.74
RMSE of Testing dataset (include 2020)	6.60	5.33	5.36	4.77
RMSE of Testing dataset (exclude 2020)	9.45	4.08	3.09	3.88

- Training dataset, GRU performs the best
- Testing data, when we include COVID-19 period data, WGAN-GP performs the best
- Testing data, when we exclude that period, Basic GAN performs the best
- Overall, GANs models perform better than the baseline traditional models according to our result.

Conlusion

we proposed a GAN which sets GRU as a generator and CNN as a discriminator. According to the experimental result, we have some conclusions.

- Compared the GAN model with the traditional models, the GAN model can help to improve the GRU model and LSTM model, both basic GAN and WGAN-GP perform better than traditional models.
- When there is an unexpected event like COVID-19, WGAN-GP performs better than basic GAN, but in normal periods, basic GAN performs better.
- GAN model including RNN is unstable, it is very difficult for these models to tune hyperparameters, without good parameters you may have bad results.

Future work

Future research should be devoted to the development of hyperparameter tuning. In the GAN model, if each of the parameters, in each layer and for the whole model, can be tuned more accurately, we believe the result would have significantly improved.

Reinforcement learning for hyperparameter optimization： Rainbow based on Q-learnin and Proximal Policy Optimization (PPO).

Thank you !

Any questions?

Reference