**6501 Capstone Final Report – Group 1**

**Stock price prediction using GAN**

Team member: Chen Chen, HungChun Lin

**Week 1 – 09/14**

This week, we did some researches to write down our proposal, and found the proper dataset.

1. Dataset:

A picture containing table

Description automatically generated

1. Literary Research

* Github: Using the latest advancements in AI to predict stock market movements:

This Github is also using the GAN to predict stock price, and it included whole process of the project that we will build up our own model based on this Github.

Reference: <https://github.com/borisbanushev/stockpredictionai#overview>

* Paper: Stock Market Prediction Based on Generative Adversarial Network:

This paper is about predicting the stock price through GAN, MLP and LSTM methods, and it only used 7 features to do the prediction.

* Stanford Course Report: Generative Adversarial Network for Stock Market price Prediction:

This is a report from Stanford course, this team used three methods to predict the stock price, which are ARIMA, LSTM and GAN, for them, they set ARIMA model as their base line.

* Paper: Computational Intelligence in Data-Driven Modelling and Its Engineering Applications:

This paper includes ARIMA, LSTM and GAN methods, they have tried GAN-F/ GAN-D/ GAN-FD/ LSTM-FD model.

1. Benchmark

In our project, we will use LSTM and GAN to predict the stock price, and the benchmark of our project is **Basic LSTM**.

Base on the basic model, we will try different activation function on each model, and for the GAN, we will try to use different loss function to improve the result.

1. The goal of this project

The goal of this project is to predict stock price more accurate.

1. The contribution of this project

In this project, we will improve the performance of predicting the stock price by using GAN and adjusting the loss function.

**Week2 – 09/22**

In the week, we set up our model structure, and will show the detail of our structure below.

1. The input data -> (n-sample, feature, input-steps):

Text

Description automatically generated

1. Output data -> (n-samples, output-steps):

Text

Description automatically generated

1. LSTM (stacked LSTM):

Text, letter

Description automatically generated

1. GAN model:

Diagram

Description automatically generated

Suggestion from Professor Jafari:

* The detail of the structure is still not clear.
* We need to choose a baseline for the benchmark.
* Create some neat files in Github.

**Week3 – 09/28**

In this project, the dataset includes 2518 rows and 26 columns (for now).

Text

Description automatically generated

The data preprocessing in for this project:

1. Filling the N/A value with the average of previous and next value
2. Normalized the data

A picture containing graphical user interface

Description automatically generated

1. Split the data into train and test

Structure:

*(In this project, we will input 30 days to predict 7 days)*

* Diagram

  Description automatically generatedLSTM

Graphical user interface, text

Description automatically generated

reshape

* CNN

A picture containing table

Description automatically generatedDiagram

Description automatically generated

* GAN

Diagram

Description automatically generated

Future work:

* Feature Engineering:
  + Add one more feature: Add “News” feature through NLP
  + Do the feature selection: XGBoost
* Improvement of GAN model

**Week4 – 10/05**

This week, we did some feature engineering to extract the features we would like to use in our prediction.

1. Technical Indicator

After did some technical calculation, we got some new features, **“MA7, MA21, MACD, 20SD, upper\_band, lower\_band, EMA, logmomentum”**, and after calculation these features, we will drop the first 21 rows which don’t contain the feature MA21.

Table

Description automatically generated

1. Fourier Transform

We calculated these features to generalize several long- and short-term trends. Using these transforms we will eliminate a lot of noise (random walks) and create approximations of the real stock movement.

Table

Description automatically generatedChart, line chart

Description automatically generated

1. News (Sentiment analysis)

For collecting the news, we extracted the news from the website: Seeking Alpha, and used Finbert pretrained model (particular for finance data)

A picture containing text

Description automatically generated

Then did the Sentiment analysis and giving score to the news (positive, neutral, negative).

Text

Description automatically generated

Results:

* **Basic LSTM**

(LR = 0.001, BATCH\_SIZE = 64, N\_EPOCH = 50) -> **RMSE = 6.55**

Table

Description automatically generated

Chart, line chart

Description automatically generated Chart, line chart, histogram

Description automatically generated

* **Basic GAN**

(We are still tuning the parameters)

Generator: LSTM

Chart, histogram

Description automatically generatedText

Description automatically generated

Discriminator: CNN

A picture containing graphical user interface

Description automatically generated

The result of features selection:

A picture containing building, large, city, field

Description automatically generated Chart

Description automatically generated

Future work:

* Improving GAN structure
* Working on plot predicted result and the RMSE
* Working on WGAN

Suggestions from Professor Jafari:

* About the feature selection, we should do the check of autocorrelation.

**Week6 – 10/19 (Mid-presentation)**

From the last week, about the suggestion of doing the autocorrelation check.

We did the statistical check:

A picture containing chart

Description automatically generated

This is the plot of Autocorrelation for target value, and we found out that previous stock price has influence when making predict, so we decided to use previous close price as a feature in the model.

In this week, we tried the LSTM with bidirectional layer, and the result looks better than basic LSTM layer.

* **RMSE = 2.75**

Graphical user interface, text

Description automatically generated

Chart

Description automatically generated Chart, line chart

Description automatically generated

We have got some result of GAN:

A picture containing graphical user interface

Description automatically generatedChart, line chart

Description automatically generated

The result of GAN for now looks not good as LSTM, from the loss plot, we can say the generator and discriminator both seem not learning, and from the training result plot, we can see there are two spikes at the beginning and the end.

We should find out where are the spikes from and keep improving the model.

In this week, we also tried to do the hyper parameters tuning, we did the tuning through Bayesian optimization.

A close up of a sign

Description automatically generated

This time, we tuned the learning rate, epoch and batch size, in the future, we will try to tune the strides and kernel size as well.

Here is the result of our benchmark:

Table

Description automatically generated

We do encounter some challenges in our project, the first one is that, the loss in GAN training process looks not good, because as normal, we though that the G\_loss should be larger than D\_loss, and it may indicate that our Discriminator is too weak; Second, the hyperparameter tuning through Bayesian optimization seems not improving our result; the last one is that the RMSE of GAN is still larger than basic LSTM.

Future work:

* Keep working on WGAN and WGAN-GP
* Adjust CNN model structure to improve the Discriminator
* Work on the Bayesian Optimization

Suggestion from Professor Jafari:

* The loss of G is not necessarily greater than loss of D, since we are not using the normal GAN, we put the RNN into our model, it might be a different situation as usual.
* We should find out the reason of the spikes, if remove the spikes, the RMSE of GAN model may be good.

**Week7 – 10/26**

This week, we tried to solve some problems we have encountered.

There are about the spikes appear at the beginning and the end, our GAN model is unstable and the generator and discriminator seem not learning.

To solve these problems, we tried to

1. Reduce output steps first, we would like to check if the problem comes from the too many output steps.
2. Set MLP as our discriminator
3. Add batch normalization, dropout, initializer in CNN
4. Adjust the generator

From the first try, change the output steps, we thought the strikes might come from too many output steps, so we tried to change the output steps from 7 to 3

* **RMSE = 5.56**

Table

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

The spikes phenomenon didn’t be solved, and also the RMSE didn’t reduce, so the problem may not come from the output steps.

Second, we tried to put the MLP as discriminator according to a paper we found, but the result was even worse.

* **RMSE = 27.62**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

But from this paper, we found out that it looks normal that D loss is larger than G loss in this case, but D loss is better to decrease through the process.

Chart

Description automatically generated

\*Loss plots from the paper

Reference: <https://reader.elsevier.com/reader/sd/pii/S1877050919302789?token=B294C2B4282B0B6792946527DDCE625AEE6A7B868F7C96A5BE9C7BD65239880760F8E36BA0BD3EDE90A274911BA2D83D>

Then we tried to set Batchnorm(), Dropout() and Initializer() into our discriminator, and it turns out not working.

Last thing we did in this week, which did improve our result a lot is that we adding one more layer in our generator, and also increased the neuron numbers.

Previous, we through that all the problems come from the discriminator is too weak, but while we tried to adjust the generator, the result did get improved.

(\*All the model we ran this time, the output step was 3.)

* On the **Basic GAN**:

Previous generator:

Table

Description automatically generated

New generator:

Table

Description automatically generated

From this model, we can see the spikes disappeared.

Chart, line chart

Description automatically generatedChart

Description automatically generated

* **And the RMSE = 2.57**
* On the **WGAN-GP**

Chart, line chart

Description automatically generatedChart, line chart, histogram

Description automatically generated

* **RMSE = 1.80**
* **Basic LSTM**

Chart, line chart, histogram

Description automatically generatedChart

Description automatically generated

* **RMSE = 3.51**

Conclusion for this week:

* The strikes disappear
* The result is much more stable than previous version, the RMSE range is about 1.5 to 6
* more complicated LSTM in GAN help the model perform better.
* WGAN model can resolve the problem of D is not learning, and the result is better than basic LSTM and basic GAN

Future work:

* Continue to improve the GAN model
* Find a way to make the model much more stable
* Try to predict different output days

Suggestion from Professor Jafari:

* We should try to use GRU but not LSTM, since LSTM is more complicated than GRU, so it might be unstable.
* We should try to use basic GRU, and then build up a larger model.
* Using one directional, not use bidirectional.
* Try to predict only the last one test dataset.

**Week8 – 11/02**

We tried to set the GRU in our generator, and the result did improve.

(\*For the models we ran this time, input = 30 days, output = 3 days.)

* **Basic GAN**

The structure:

A picture containing graphical user interface

Description automatically generated A picture containing graphical user interface

Description automatically generated

The result of GRU improve a lot from LSTM

The result of training dataset:

Chart

Description automatically generated Chart

Description automatically generated

* **RMSE = 2.07**

The result of test dataset:

Chart

Description automatically generated

* **RMSE = 5.54**

The result of predicting last three days:

Table

Description automatically generatedChart, line chart

Description automatically generated

* **RMSE = 17.88**
* **Basic GAN – Train G 2 times**

The result of training dataset:

Chart, line chart

Description automatically generated Chart

Description automatically generated

* RMSE = 1.64

The result of test dataset:

Chart

Description automatically generated

* **RMSE = 5.88**

The result of predicting last three days:

Table

Description automatically generatedChart, line chart

Description automatically generated

* RMSE = 20.17

For Basic GAN, the training times of G seems has no obvious effect on the result.

* **WGAN-GP**

For WGAN-GP, we trained the generator 3 times and discriminator 1 time.

The result of training dataset:

Chart, line chart

Description automatically generated Chart, line chart, histogram

Description automatically generated

* RMSE = 1.89

The result of test dataset:

Chart, histogram

Description automatically generated

* RMSE = 4.77

The result of predicting last three days:

Table

Description automatically generatedChart, line chart

Description automatically generated

* RMSE = 12.75
* **Baseline model – LSTM**

The result of training dataset:

Chart, line chart

Description automatically generated Chart

Description automatically generated

* RMSE = 1.52

The result of test dataset:

Chart, line chart, histogram

Description automatically generated

* RMSE = 6.60

The result of predicting last three days:

Table

Description automatically generatedChart, line chart

Description automatically generated

* RMSE = 11.95

Model comparison:

Table

Description automatically generated

There is a special phenomenon of the last three days prediction we found out from our result:

The prediction of the last three days through **GAN** is very inaccurate, this might cause from unpredictable sharp drop and growth which due to the COVID-19.

Chart, histogram

Description automatically generated

On the other hand, the prediction of the last three days through **Basic LSTM** is more accurate, but as it can be seen from the plot, the prediction of LSTM is always much higher than the real price, which makes the prediction of the recent period more accurate.

Chart, line chart, histogram

Description automatically generated

The problem for this week:

* GAN model did not perform better than Baseline model and model still need to be improved

Question for the professor:

* WGAN-GP best model (train RMSE 1.89, test RMSE 4.77) when train RMSE smaller than 1.89 test RMSE will increase, overfitting?

Suggestion from Professor Jafari:

* For our question, it’s hard to say if it is overfitting or not.
* In our test dataset, we can remove the data of 2020, and see if the GAN model performs better than the base line model on predicting the last three days.
* How does the LSTM know about our dataset has time series between each sample?
* We can try to build up a model for input = 3 days and output = 3 days.

Week9 – 11/09

This week, we focused on understanding how the model knows our dataset does have a time-series relation with each sample.

And for doing the time-series prediction with LSTM, there is a parameter called stateful, we can build up a stateful model or a stateless model.

The concept is that, **stateful** connections (state) between sequences(samples) would be retained. It can enable the model to learn the timing characteristics between samples. Which sample is in the first place and which sample is in the second place will have an impact on the model; **Stateless** during training, the state is reset after each sequence, state isn't retained between the sequences(samples), Which can be said to be independent of each sample, with no contextual relationship between them.

For splitting the original data to stateful or stateless data.

*Stateless*:  *Stateful:*

Diagram

Description automatically generated A picture containing diagram

Description automatically generated

The concept of stateful data:

Calendar

Description automatically generated

For building up the stateful model, we need to adjust the dimension of the input of the generator, and also need to adjust the input of discriminator.

Generator:

Text

Description automatically generated

Discriminator:

Text

Description automatically generated

Future work:

* Working on the code with stateful model
* Pretrain the model for Generator and Discriminator

Suggestions from Professor Jafari:

* From our structure, it looks like we are doing the input = 3 steps, we can try to do input = 3.