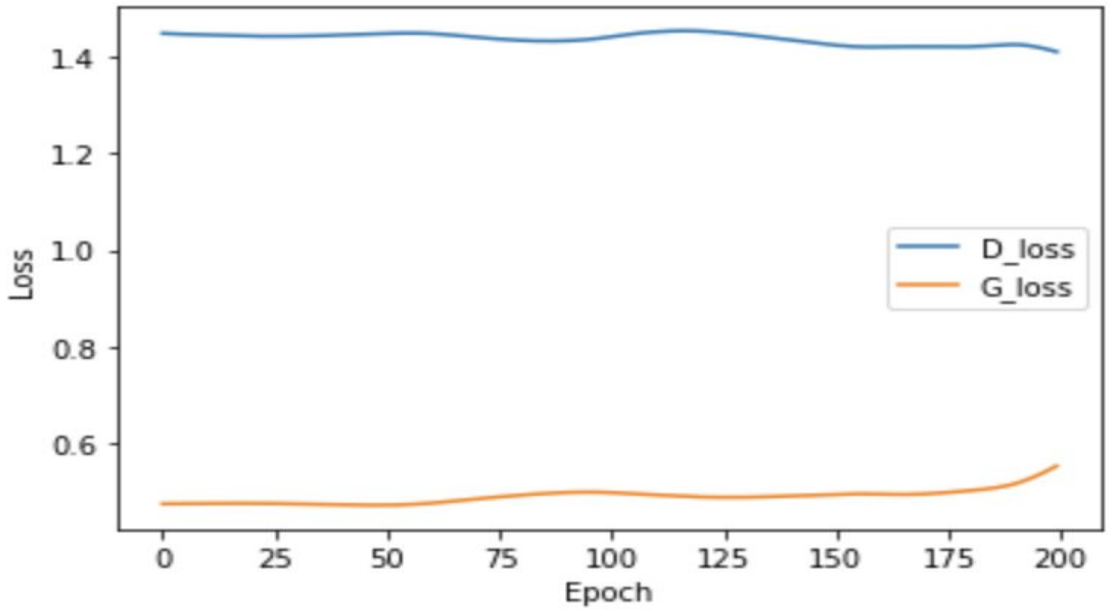
Week 7
meeting note

Problem

1. Strikes appear at the beginning and the end
2. The GAN model is unstable
   - The RMSE range can be 2 to 60
3. The losses of G and D seems not learning



To solve the problem

- reduce the predict output steps
- change discriminator to mlp
- try batch normalization, dropout, initializer in CNN
- adjust generator in GAN model

Change the output step (7 -> 3)

- We thought the strikes might come from too many output steps

  -> tried to change the output steps from 7 to 3

Model: "Generator-LSTM"

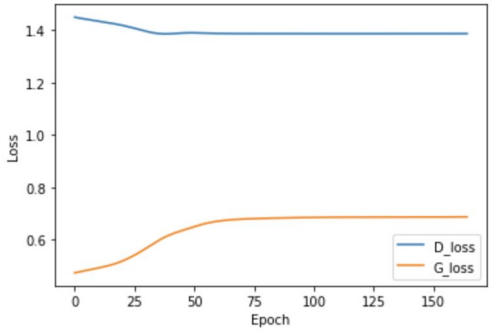
Layer (type)	Output Shape	Param #
bidirectional_5 (Bidirection	(None, 128)	51712
dense_20 (Dense)	(None, 3)	387

*Output: 3 days

Model: "Discriminator-CNN"

Layer (type)	Output Shape	Param #
conv1d_12 (Conv1D)	(None, 17, 32)	128
conv1d_13 (Conv1D)	(None, 9, 64)	6208
conv1d_14 (Conv1D)	(None, 5, 128)	41088
flatten_5 (Flatten)	(None, 640)	0
dense_21 (Dense)	(None, 128)	81920
leaky_re_lu_20 (LeakyReLU)	(None, 128)	0
dense_22 (Dense)	(None, 64)	8192
dense_23 (Dense)	(None, 1)	65

strikes phenomenon did not improve



RMSE: 5.56

RMSE did not reduce

Paper review - Set the MLP as Discriminator

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 147 (2019) 400–406
Procedia Computer Science
www.elsevier.com/locate/procedia
2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018

Stock Market Prediction Based on Generative Adversarial Network

Kang Zhang[a], Guoqiang Zhong[a,*], Junyu Dong[a], Shengke Wang[a], Yong Wang[a]
[a]Department of Computer Science and Technology, Ocean University of China, Qingdao, 266100, China

- Input 5 days, output 1 day
- 7 features
- LSTM as Generator
- MLP as Discriminator
- Classic GAN

It looks normal that D loss is larger than G loss in this case, but D loss is better to decrease through the process.

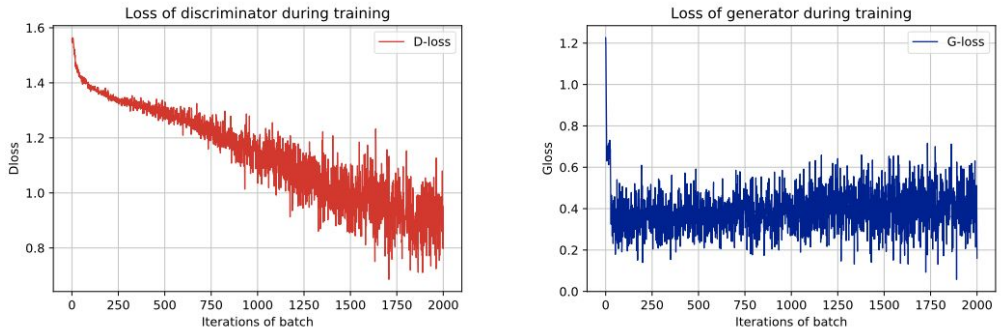


Fig. 4. Losses of the discriminator and generator during training.

Table 2. The average evaluation on five stock data sets.

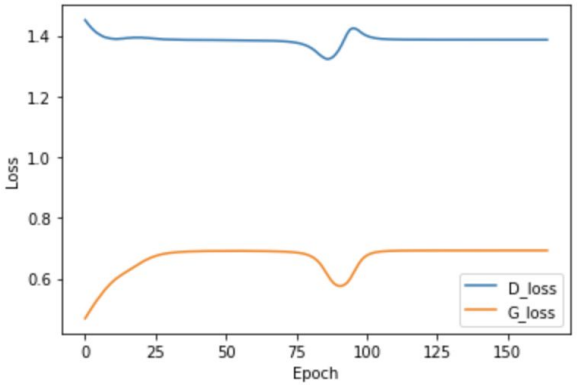
Method	MAE	RMSE	MAPE	AR
Our GAN	**3.0401**	**4.1026**	**0.0137**	**0.7554**
LSTM	4.1228	5.4131	0.0145	0.6859
ANN	7.3029	9.1757	0.0808	0.5249
SVR	4.9285	8.2261	0.0452	0.7266

Reference:
https://reader.elsevier.com/reader/sd/pii/S1877050919302789?token=B294C2B4282B0B67929465
27DDCE625AEE6A7B868F7C96A5BE9C7BD65239880760F8E36BA0BD3EDE90A274911BA2D83D

**Output: 3 days*

Set the MLP as Discriminator (II)

RMSE: 27.62




```
Model: "My-MLP-model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_14 (Dense)             (None, 1024)              34816
_________________________________________________________________
leaky_re_lu_16 (LeakyReLU)   (None, 1024)              0
_________________________________________________________________
dropout (Dropout)            (None, 1024)              0
_________________________________________________________________
dense_15 (Dense)             (None, 512)               524800
_________________________________________________________________
leaky_re_lu_17 (LeakyReLU)   (None, 512)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_16 (Dense)             (None, 256)               131328
_________________________________________________________________
leaky_re_lu_18 (LeakyReLU)   (None, 256)               0
_________________________________________________________________
dense_17 (Dense)             (None, 1)                 257
=================================================================
```

-> The result of MLP is even worse than any result of CNN

What we have tried on CNN model

- According to the trace of loss through the process, we thought the problem is on the ***weak Discriminator***
- So we have tried adding some other layers to our CNN model:
  - Batchnorm()
  - Dropout()
  - Initializer()

-> The result did not improve

- Next:
  - Try to adjust the Generator

*Output: 3 days

Adjust the Generator - LSTM model

*Basic GAN

Previous

Model: "Generator-LSTM"

Layer (type)	Output Shape	Param #
bidirectional_5 (Bidirection	(None, 128)	51712
dense_20 (Dense)	(None, 3)	387

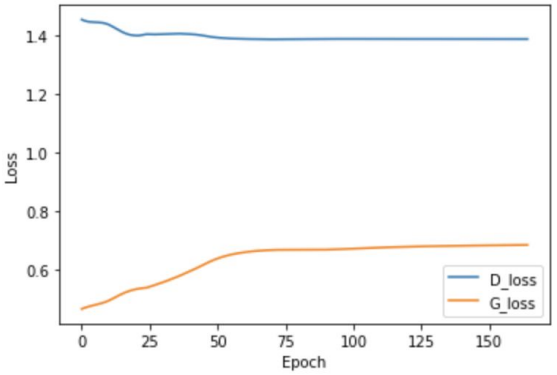
- **Add one more layer**
- **Increase the number of neuron**

New

Model: "Generator - new"

Layer (type)	Output Shape	Param #
bidirectional_10 (Bidirectio	(None, 512)	600064
dense_32 (Dense)	(None, 128)	65664
dense_33 (Dense)	(None, 3)	387

RMSE: 2.57



-> The result improves a lot, strikes disappear

*Output: 3 days

Adjust the Generator - LSTM model

***WGAN-GP (Different loss function compare to GAN)**

Previous

Model: "Generator-LSTM"

Layer (type)	Output Shape	Param #
bidirectional_5 (Bidirection	(None, 128)	51712
dense_20 (Dense)	(None, 3)	387

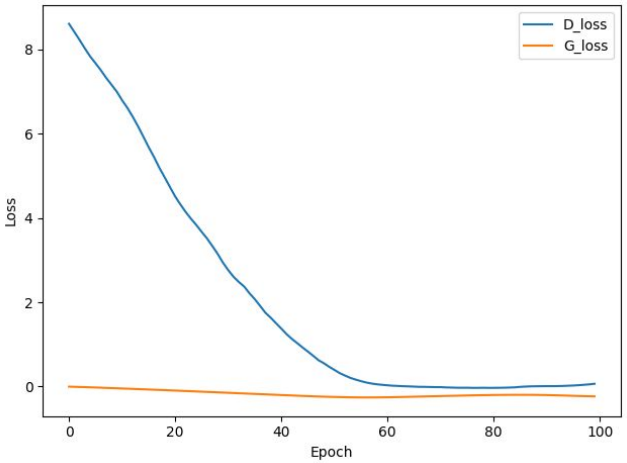
- **Add one more layer**
- **Increase the number of neuron**

New

Model: "Generator - new"

Layer (type)	Output Shape	Param #
bidirectional_10 (Bidirectio	(None, 512)	600064
dense_32 (Dense)	(None, 128)	65664
dense_33 (Dense)	(None, 3)	387

RMSE: 1.80



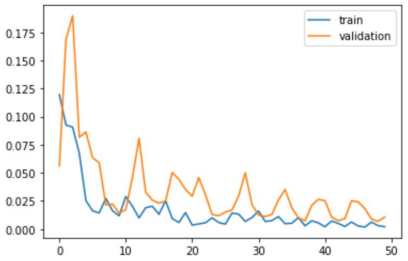
-> The result improves a lot, strikes disappear

**Output: 3 days*

Basic LSTM(Baseline model)

Same LSTM structure with GAN

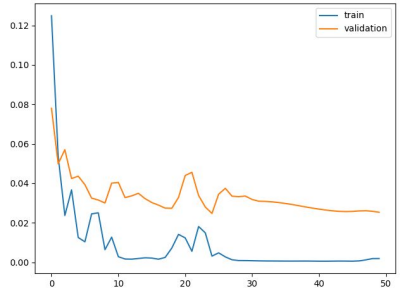
Layer (type)	Output Shape	Param #
bidirectional_11 (Bidirectio	(None, 512)	600064
dense_66 (Dense)	(None, 128)	65664
dense_67 (Dense)	(None, 3)	387



RMSE: 3.51

Previous LSTM structure

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional	(None, 128)	51712
dense (Dense)	(None, 3)	387



RMSE: 2.67

Conclusion for new structure

- 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

Question??

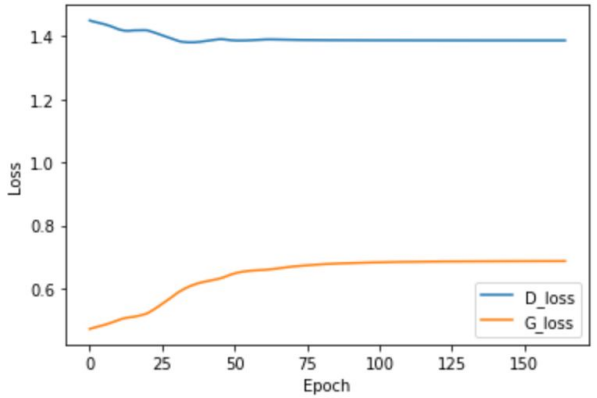
- About the baseline: Basic LSTM, does this LSTM structure should be the same as the one we put in the GAN Generator?

- How to improve model stability?
  - We have tried Batchnorm and Initializer

**Output: 3 days*

Basic GAN

RMSE: 3.60



Layer (type)	Output Shape	Param #
bidirectional_12 (Bidirectio	(None, 512)	600064
dense_68 (Dense)	(None, 128)	65664
dense_69 (Dense)	(None, 3)	387

Layer (type)	Output Shape	Param #
conv1d_45 (Conv1D)	(None, 17, 16)	64
conv1d_46 (Conv1D)	(None, 9, 64)	3136
conv1d_47 (Conv1D)	(None, 5, 128)	41088
flatten_15 (Flatten)	(None, 640)	0
dense_70 (Dense)	(None, 220)	140800
leaky_re_lu_69 (LeakyReLU)	(None, 220)	0
dense_71 (Dense)	(None, 220)	48400
dense_72 (Dense)	(None, 1)	221

Comparison different model for the new structure

Change the output step (7 -> 3)

- We thought the strikes might come from too many output steps

  -> tried to change the output steps from 7 to 3

- It didn't improve the strikes phenomenon

- We set the output step as 3 for the following models

  (All the previous model versions have been repeated)