My project is about the prediction of stock price with RNN. The reason why I choose this project \_ it’s mainly because of my father. He is an investor , who has been taught a lot of lessons in the stock market.

One thing you may not know about the Chinese stock market is: It somethimes follows the trend of the stock market in United states. Which means, when American investors lose money , Chinese investors will also lose their money. When American investors win some money ——，Chinese investor will still lose their money —— but not that much.

HOw brave my father is , right ?

The reason why I choose RNN, the **Recurrent Neural** Network is, It’s designed for timeseries data. timeseries data is easy to understand, right? Like the temperature, it changes over time. Much of our data are inherently sequential. It will reach to it’s lowest point at about 2 am and get to the highest ppoint at 2pm also because of the sunlight. So as the stock data. Your product is a hit,so the stock price is higher. There is a strong need of your service, so the stock price is higher… It’s explainable.

What is different from MLP is. We do not treat it as indivial data. We treat it as a series. They are related to each other. We give index to the data and feed them in the system It can be thought of as multiple copies of the same network, each passing a message to a successor

This is what I do

1. I chose the stock data from Amazon and decided the variable(Open', 'High', 'Low')→Close
2. Decided the basic variables for my model. In this project, I use stock price data from the prior 60 days to predict the close price in 5 days.
3. Choose different models
   1. stacking LSTM layers and recurrent dropout.
   2. a recurrent neural network with one LSTM layer.
   3. recurrent dropout.

The reason why I choose LSTM(Long short-term memory) is : For simple RNN , there are some problems. We have multiple layers , and each paramter is sent from layer to layer. Just imagine we have 5 layers. When the calculcation comes to the final layer, it has all the information from all these 5 layers. It’s a hugh system , right? But unluckily, the information form the very first layer will have little inpact. The network just make it inpact little because it has to remember informationfrom other layers!

So that’s what LSTM do. It automatcally forget something unnecessary, which makes the model have a better long-term memory.

As for the recurrent dropout. It’s another way of forget things.

1. Train the model and get the results
2. Evaluate the model