Re-Imaging Futures Market Data

Janiszzz

June 2023

1 Introduction

This paper uses convolutional neural network (CNN) to model the predictive correlation between the stock's OHLC image and future earnings. The most basic idea is to use computer vision technology to mimic traditional technical analysis. The use of CNN aims to generate unsupervised predictions from images without the need for researchers to manually design predictive features.

The input of CNN is usually an image. In the setup of this article, the image is a graph of past-market information (opening price, highest price, lowest price, closing price, and trading volume), represented as a black and white pixel value matrix.

The main price chart uses OHLC bars (represented in black). The highest and lowest prices are represented by the top and bottom of the middle vertical bar, while the opening and closing prices are represented by small horizontal lines on the left and right sides of the bar, respectively. In the image in this article, a day occupies an area of 3 pixels wide (1 pixel wide for the center, opening, and closing markers).

1. The main component of the image is an OHLC strip with continuous intervals of 5, 20, or 60 days (approximately weekly, monthly, and quarterly price trajectories, respectively). Therefore, the width of an n-day image is 3n pixels.

2. All images have the same pixel size because of the same number of days.

3. If there is missing data in the image, leave the pixel column corresponding to the number of missing days blank.

4. The image uses black as the background color and white as the background color, which simplifies data storage requirements because black pixels are represented by (0,0,0) and so that the generated image is sparse.

5. The image has added a moving average with a window length equal to the number of days in the image. The daily moving average reports each day using one pixel in the middle column and draws a line by connecting these points.

6. The image also adds a set of daily trading volume columns. When including trading volume data, the trading volume is displayed in the bottom fifth of the image, while the top four fifths contain the main OHLC chart.

This article considers predicting returns as a classification problem. If the future return is positive, the label of the image is defined as 1, otherwise it is defined as 0. This paper considers three input options, including Market data images in the past 5, 20 or 60 days. For positive or non positive returns in the next 5, 20 or 60 days, the image tag value is 1 or 0. The network output is a probability value p, which represents the probability that the image belongs to the future positive return category. This article categorizes stocks into decimal investment portfolios and calculates the investment performance of long short portfolios.

In addition, the article also carried out a variety of robustness tests, including comparing CNN trading strategies with classic artificially constructed predictors (such as momentum, beta, size, etc.), traditional technology analysis, and Transfer learning in other countries' markets.

The original paper uses panel data of US stock returns from 1993 to 2019, and I will replicate their model in the Chinese futures market and implement trading strategies.

2 Data

I employee China's futures market data, and train the network during the period from 2016-01-04 to 2021-01-01, about 1218 trading days, and backtest during the period from 2021-01-02 to 2023-01-13, about 494 trading days. Considering our dataset, it is almost a 3:1 division. Inside the training period, I randomly divide it into training set and valid set, in a ratio of 7:3. The random division would not affect the time effect, not only because CNN is a cross-sectional method, but also I care about only the time-relatively prediction.

All futures varieties are mixed together for training, aiming to find common pattern in futures price and gain general prediction. I drop Fiberboard futures(FB), because it has poor liquidity and had relaunched on 2019-12-02, which disturbed its price pattern.

2.1 Data-preprocessing

Before doing anything, I need to handle the "Futures Jump". Futures prices are not continuous because there are multiple futures varieties with different maturities at the same time, and if I choose the most liquid variety to invest in, it means that I need to conduct a whole position change every quarter, when the last one is going to mature. In real world, this happens several days before the maturity. "Futures Jump" would cause bias in our invest strategy.

To build a continuous index of futures price, I scale the after-price by base period price, i.e. the first main contract in our data period. The index is like a Laspeyres index. The advantages of this method is, it can remain the return consistent, which keeps the prediction stabilize. In backtesting, I just need to deduct month-changing transaction cost at checkpoint.

Algorithm 1 Futures price index construction

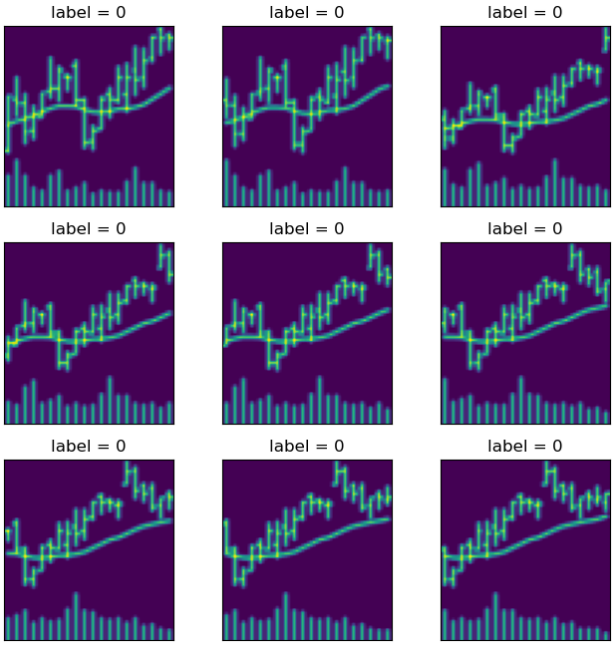
function MAIN CONTRACT ADJUST
  $factor \leftarrow 1$
  $adjustedPrice = [\,]$
  $monthChangingCheckpoint = [\,]$
  for *date* in *tradeDate* **do**
    if *main contract changes* **then**
      $factor \leftarrow \frac{mainContractPrice[date-1]}{mainContractPrice[date]} * factor$
      $monthChangingCheckpoint.append(date)$
    end if
    $adjustedPrice.append(mainContractPrice[date] * factor)$
  end for
return $adjustedPrice, monthChangingCheckpoint$
end function

2.2 Drawing OHLC figures

In this part, I employee mplfinance package in Python to draw OHLC figures. However, it is originally to draw figures depend on your screen's resolution. So after proportional

drawing(just print a bigger picture in the same proportion of the picture I need), I use data preprocessing class in Pytorch to re-scale the size of figures. The function would not loss information of pictures by its design. Figure 2.1 shows some examples of days step = 20.

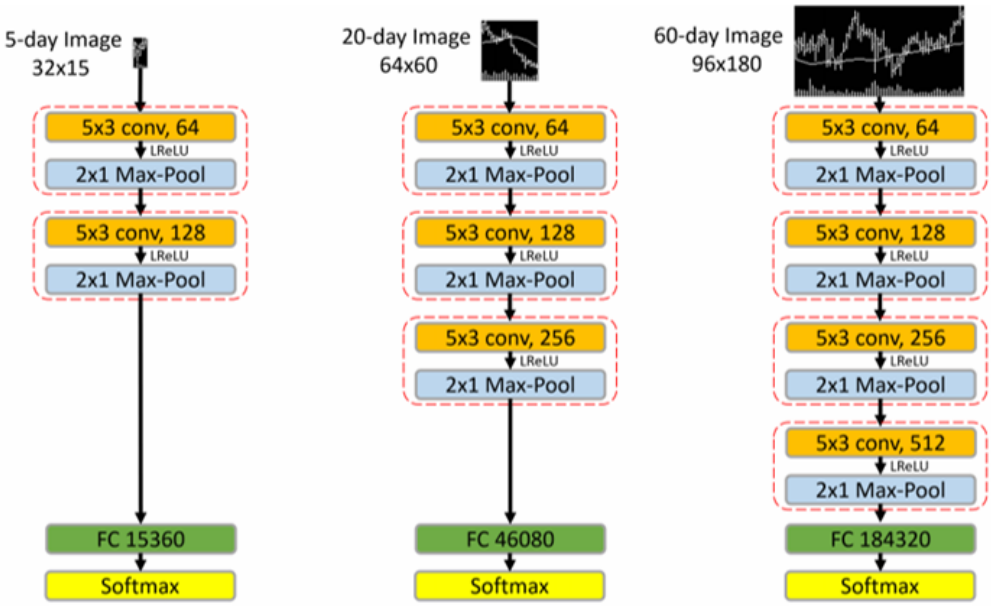
Figure 2.1: OHLC pictures



3 Training and Performance

I copy the network structure in Xiu et al(2019). That restrict the hyper-parameters I can modify. In training, I mainly modify learning rate by warm-up strategy.

Figure 3.1: Diagrams of Networks

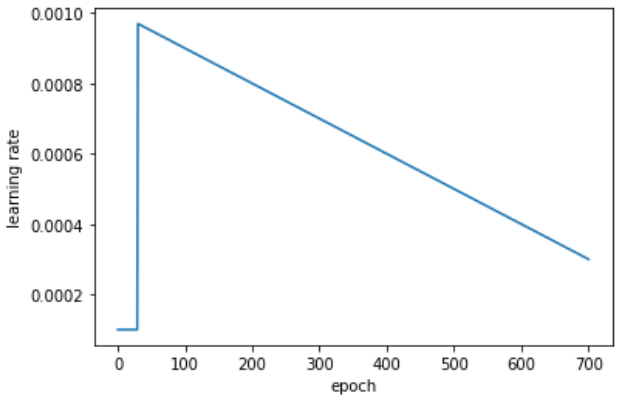


3.1 Warm-up learning rate

At the beginning of training, the weights of the model are randomly initialized. At this time, if a large Learning rate is selected, it may cause instability of the model. Choosing Warm-up to preheat the Learning rate can make the Learning rate within several epochs or steps that start training smaller. Under the preheat primary learning rate, the model can slowly become stable. After the model is relatively stable, select the preset Learning rate for training, Make the convergence speed of the model faster and the model effect better.

In our study, I do not use build-in learning rate scheduler in Pytorch, but realize a simple linear adjustment strategy.Figure below shows the learning rate changing process.

Figure 3.2: Learning rate changes with epoch



Algorithm 2 Warm-up learning rate

```
for t←0 to epochs do
    if t<30 then
        lr←c
    else
        lr←(epochs−t)/epochs∗c
    end if
end for
```

3.2 Loss Performance

Here shows the loss curve of train set and valid set. Notice that loss data is a arithmetic sum of loss function output and hasn't been scaled, so its absolute value depends. The results shows in a window length of 5/20, network performs poor and overfitting obviously. The training loss declines with the valid loss increase. It may be because of the network is shallow and cannot identify the correct pattern. Compared with the two, network with a window length of 60 performs well. Finally I choose the 60-model with epoch of 80 for inference.

Figure 3.3: Training Loss - window length 5

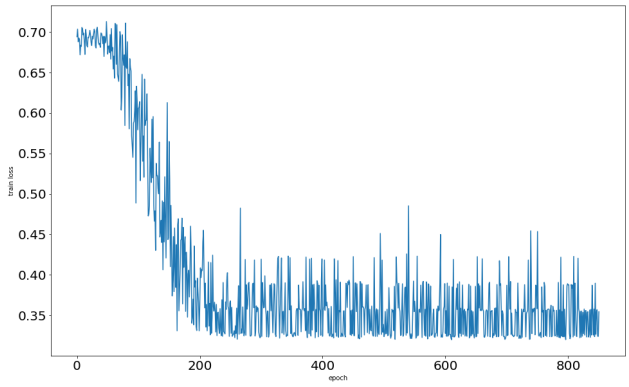


Figure 3.4: Valid Loss - window length 5

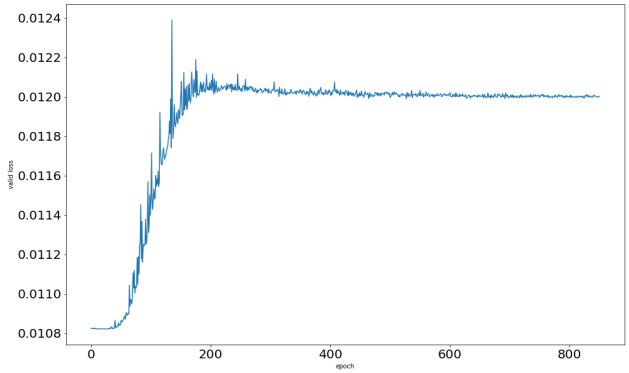


Figure 3.5: Training Loss - window length 20

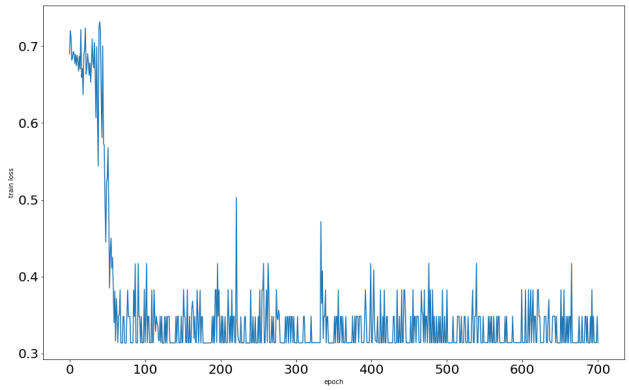


Figure 3.6: Valid Loss - window length 20

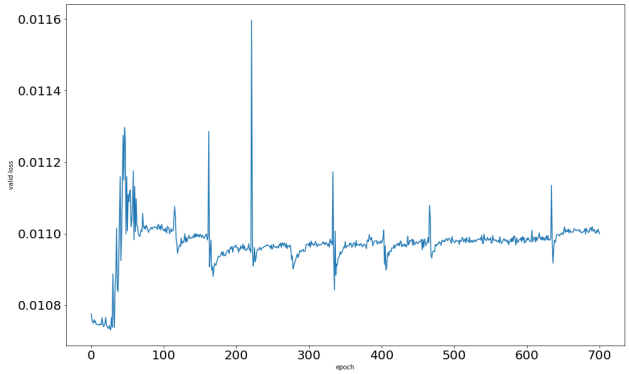


Figure 3.7: Training Loss - window length 60

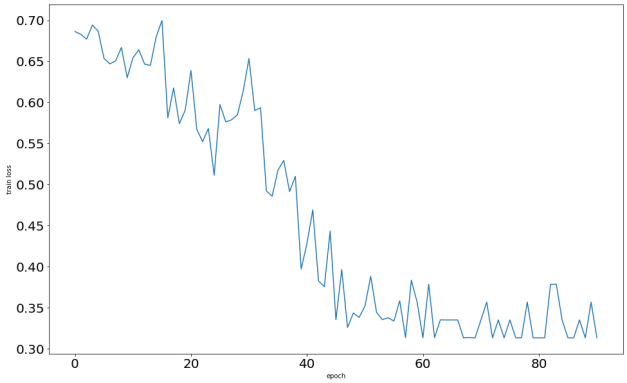
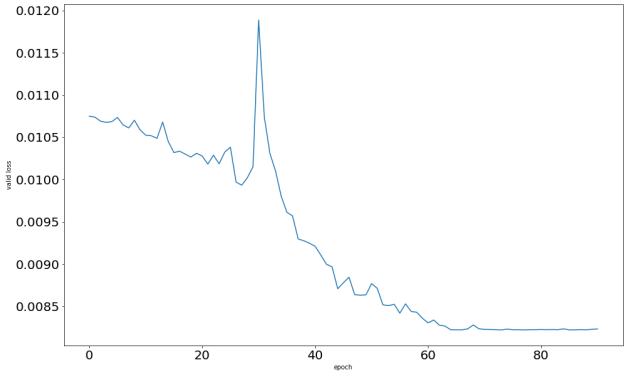


Figure 3.8: Valid Loss - window length 60

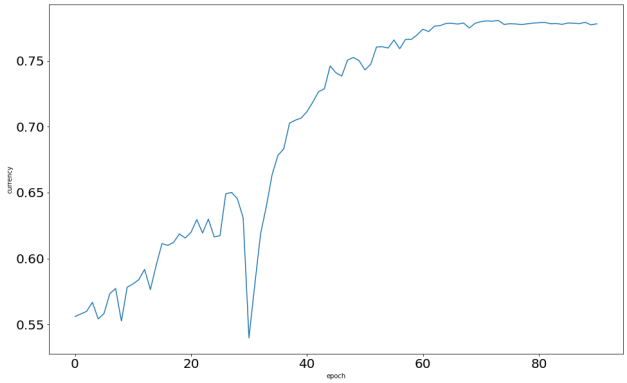


4 Backtesting

4.1 Factor Testing

After training, the network would like to output a float between [0,1] for each picture, presenting the prob. of the price go up and down. In detail, because we choose the 60-60 network, the model would predict the status of future 60 days' return. We divide the prob. into 0 and 1 tow types according to if it was bigger than 0.5, then the valid currency curve is:

Figure 4.1: Valid Currency - window length 60



I employee the classical single portfolio sort for the prob., which is treated as the factor exposure. Here shows the result, in which I take a decile division with 6 futures in each portfolio. It shows the factor does carry a significant profit.

Table 4.1: Portfolio Return of Single sort

	L	2	3	4	5	6	7	8	9	H	HML
mean of return	0.013	0.024	0.018	0.026	0.029	0.030	0.030	0.040	0.039	0.041	0.027
t-stat	3.544	6.440	4.895	7.474	7.800	8.194	8.194	11.769	12.508	7.818	3.285
market adjusted return	-0.016	-0.006	-0.011	-0.001	0.000	0.001	0.002	0.010	0.010	0.011	0.027
market adjusted t-stat	-4.754	-1.847	-3.573	-0.470	-0.069	0.365	0.711	3.456	3.687	2.384	3.285
average factor(prob. of go up)	0.000	0.001	0.028	0.223	0.638	0.922	0.991	0.999	1.000	1.000	1.000

4.2 Strategy Return

After cross-sectional selection of futures by using factor value, I also build long-short portfolio for backtesting. In backtesting, I choose a position adjustment strategy with loop. Because the model always predict return after 60 days, so we generate a portfolio in each day for next 60 days. And on day 61, we release it and gain pl. So inside a loop we hold portfolios range from 1 to 60, most of the time, we hold 60 portfolios. Inside each portfolio, all of futures positions are equal-weighted, i.e. I divide the initial capital into 60 parts in each portfolio and 720 parts in each futures in max. Here the figure show the strategy and its returns. I supposed there are 5 bp of transaction cost each day. Notice: the return curve is from 2021-05 to 2022-05, because of arrangement of data to avoid future functions.

Figure 4.2: Investment Strategy

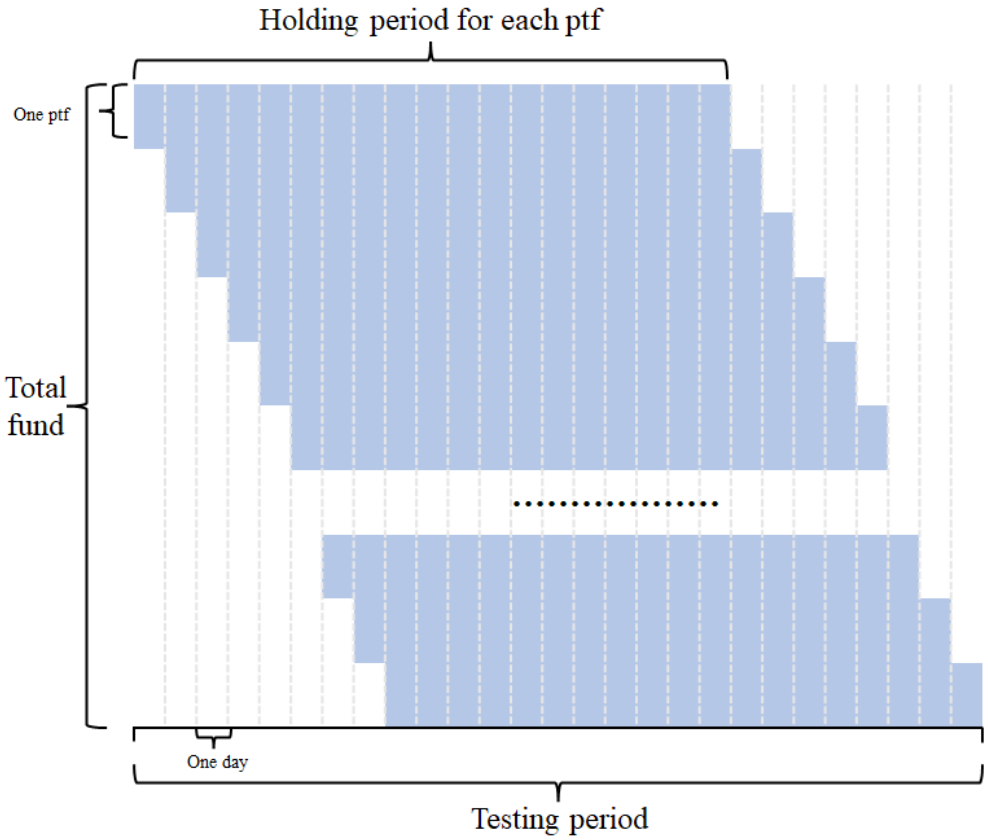


Figure 4.3: Strategy Return

Table 4.2: Daily Strategy Book Return Describe

	Mean Return	Std	Cumulative Return	Sharp Ratio	Annulized Sharp Ratio
market index	0.06%	0.01	13.86%	0.07	1.03
long-short ptf return	0.04%	0.00	8.77%	0.42	6.60

References

[1] Jiang, J. , Kelly, B. T. , Xiu, D. . (2020). (re-)imag(in)ing price trends. SSRN Electronic Journal.