MSBD 5013 Final Project
Paper Replication: (Re-)Imag(in)ing Price Trends

CEN Xinxin, MA Xiaoran and WU Xiang {xcenab, xmabi, xwucb}@connect.ust.hk
Department of Computer Science and Engineering, HKUST

1. Introduction

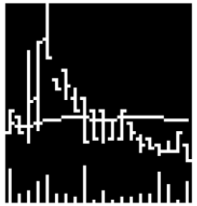
In our project, we implemented different image classification models to predict future returns of stocks. We used images of stock-level price charts as raw predictor data. We followed the paper *(Re-)Imag(in)ing Price Trends* to decide the data split strategy, model structure, evaluation metrics and robustness checking methods. Other than CNN models, we also investigated a traditional method based on HOG feature and Logistic Regression model. Finally, we compared all the models and analyzed the results.

2. About Dataset

The sample period is from 1993 to 2019. Each sample is an image which concisely embed a variety of a stock patterns of consecutive 20 days. To be specific, each image consists of three components: OHLC Chart, moving average lines, and volume bars. The images of 20-days version are of size 64×60. Images labels takes value 1 for positive returns and 0 for non-positive returns of the subsequent 20-days.

3. HOG-Based Method

HOG (Histogram of Oriented Gradients) is a feature descriptor used in image processing, mainly for object detection. Getting the HOG feature descriptor consists of several steps. The first is dividing the image into cell of 8×8 pixels, and calculating the gradient for each pixel of cell. The second step is to group the gradients into nine bins using the direction of the gradient. If the direction is between two bins, then the votes is divided proportionally to each bins. Since we want histogram values to be independent of lighting, we also applied L2-norm in each block consisting of 2×2 cells. The result feature vector has dimension of 1512, which is much smaller than the original image. The images below shows an example: the left one is the original image, and the right one is the visualization of HOG descriptors of each cell.



After transforming images into HOG feature descriptors, we use Logistic Regression as the classifier. The classification performance is recorded in the last row of the resulting table. The performance is apparently worse than that of the CNN models overall.

4. CNN-Based Method

◆ Data Preparation

Images labels take value 1 for positive returns ('up'), 0 for nonpositive returns ('down') and 2 for 'NaN' value. We remove any samples whose returns are 'NaN' or label marks are 2, because the requirement is a two-class classification problem.

◆ Model Design

We replicate a baseline CNN architectures which has three convolution blocks, a fully connected layer and other details which refer to the original paper.

◆ Workflow Design

Firstly, we divide the entire data into training, validation and testing dataset. Specifically, we use the first eight-year sample (randomly select 70% for training and 30% for validation) to train and validate the model. The remaining nineteen years of data are used for testing. Then, in the train process, we use cross entropy loss and accuracy to measure our task. Moreover, we use more evaluation metrics like Spearman Correlation, Pearson Correlation, and AUC. We also use other techniques, such as Xavier initialization, dropout, batch normalization, early stopping and so on, to deal with the over-fitting issues and improve the model performance, which all mentioned in the paper. Finally, for the extension task one, we do ablation studies and test robustness following what the original paper mentioned, and also replicate a table as Table 18 partially shown in the paper. We also post the result of HOG-Based Method to make a comparison.

		Loss		Acc.		Correlation		ROC
		V	T	V	T	Spearman	Pearson	Auc
Baseline		**0.689**	**0.695**	**0.541**	**0.525**	**0.064**	**0.040**	**0.535**
Filters (64)	32	0.689	0.694	0.538	0.519	0.060	0.039	0.532
Layers (3)	2	0.690	0.698	0.541	0.521	0.057	0.037	0.531
Dropout (0.50)	0.00	0.693	0.697	0.531	0.520	0.047	0.032	0.526
BN (yes)	no	0.688	0.695	0.542	0.520	0.056	0.037	0.531
Xavier (yes)	no	0.688	0.693	0.542	0.523	0.059	0.039	0.532
Activation (LReLU)	ReLU	0.696	0.704	0.533	0.513	0.049	0.031	0.526
Max Pool Size (2×1)	2×2	0.689	0.694	0.538	0.520	0.059	0.038	0.532
Filter Size (5×3)	3×3	0.696	0.703	0.529	0.508	0.045	0.029	0.524
Dilation/Stride (2,1)/(3,1)	(1,1)/(3,1)	0.689	0.694	0.539	0.520	0.055	0.037	0.544
HOG Feature + Logistic Regression		0.691	0.693	0.525	0.511	0.038	0.022	0.521

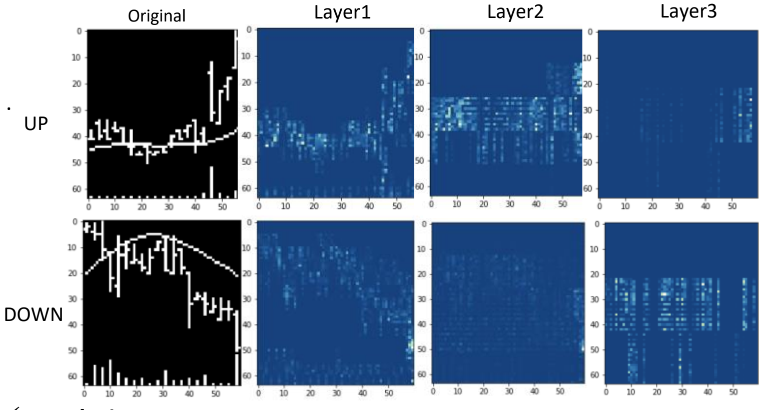
5. Interpretability of the CNN Model

✓ Grad-CAM

Grad-CAM is a visualization method to interpret CNN model. It is gradient-weighted class activation mapping. According to the output vector, Grad-CAM use backward to get the gradient corresponding to each pixel on each feature map, then average each gradient map to get the weight of each feature map, and finally the final class activation graph can be obtained through the activation function.

✓ Results

For extension task two, we illustrate our CNN baseline model and draw both "up" and "down" images in 2019 respectively. The following figure shows the original images and Grad-CAM heat map at each of the CNN's three layers.



✓ Analysis

From the results, we can see that CNN model focus on high volatility. In the Layer1, we find that model pays special attention to open and close prices. And in the Layer2 and Layer3, we also find that "up" sample is activated in the upper regions of the image and "down" sample is activated in the bottom of the image.

6. Contribution

- Coding: all three members make contributions to the modeling part.
- Poster: MA Xiaoran: part1-3, WU Xiang: part4, CEN Xinxin: part5