# Project 2: Paper Replication Study

**Chen Liu**
Department of Mathematics
cliudh@connect.ust.hk

## 1 Introduction

This report contains the replication results of Jiang et al. [2]. This paper focuses on using Convolutional Neural Networks(CNN) to predict the future return. Different from previous works, this paper attempt predict the future return via the image of historical price charts,moving average lines(MA) and volume bars(VB). While previous works tend to use the trade information and orderbook information. It makes this work relatively novel. In Figure. 1, some examples from the datasets are present. The images are in binary form with one channel only. And the goal is to use the char to predict the return in the future. In Jiang et al. [2], it studies 5-day return, 20-day return and 60-day. In this report, I only reproduce the result of 20-day return. The input size of each image is $64 \times 60$. This paper choose to convert the return into binary label, i.e. 1 for positive return and 0 for negative return. We follow the instruction of the paper and use the cross entropy loss with the following form,

$$L_{CE}(y, \hat{y}) = -ylog(\hat{y}) - (1-y)log(\hat{y}) \tag{1}$$

Here $y$ denotes the true label and $\hat{y}$ denotes the prediction. For data split, I follow the original paper and the instruction to take the first 7 files for train and validation. In detail I use 70% for train and 30% for test. For this part, I extract the data and convert to npy form. The remaining data is used for testing. Due to the memory constraints, it is hard to feed them simultaneously. So I test every file independently and combine the test results.

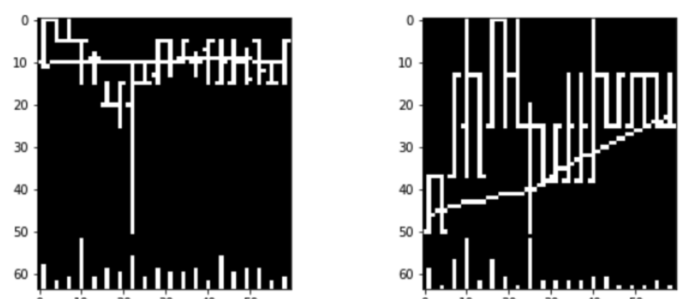


Figure 1: This figure presents the example of the images from the given dataset.

## 2 Architecture

For the architecture, I slightly alter the structure of the original paper. I still use 3 convolutional layers with 1 fully connected layer. The convolutional kernel follows the paper with kernel size $5 \times 3$. The stride and dilation are $2 \times 1$ and padding is $5 \times 3$. For my setting the input dimension is 18176, and the fully connected layer uses dropout [6]. I do not use Batch Normalization [1] here. For the activation function, I use Leaky Relu with parameter 0.1. The strcuture is shown in Figure. 2.

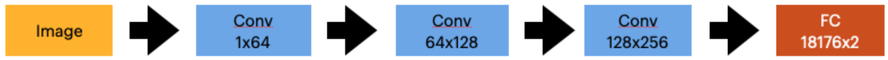


Figure 2: This figure presents the structure.

## 3 Experiment

**Setting** For the replication, I use the Pytorch [4] and some other Python packages. I choose AdamW [3] optimizer in Pytorch. For replication, the learning rate is set as 1e-5 and the weight decay is set 5e-4. To further analyze the method, I also make some ablation studies on the learning rate and weight decay. For the training process, I set a maximum epoch of 15. And I use early stop here. Following the original paper, I stop the training when the validation accuracy does not improve for two consecutive epochs. In addition, I calculate the top 1 classification accuracy. To increase the training speed, I use batch size as 1000. All the images are rescaled to $[0, 1]$ before training.

**Result and analysis** For the main result, the original paper reports a 53.3% accuracy. Using the same setting, I get the result as 55.95% on test data. For the ablation, I pick some other learning rate 1e-3, 5e-4, 1e-4, 5e-5 and 1e-5 to test the sensitivity to learning rate. The results are included in Figure. 3. It is obvious that the model is robust to learning rate.

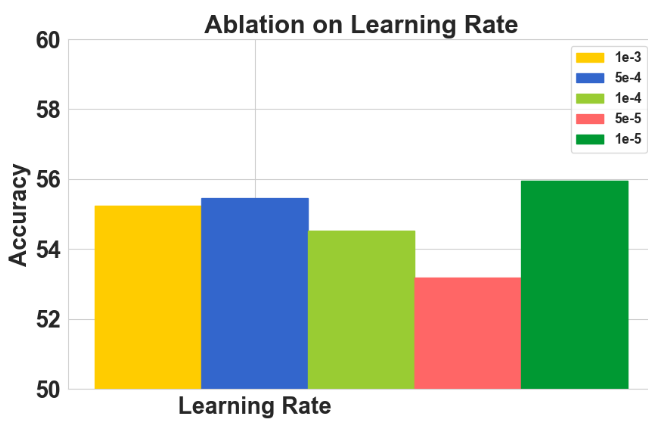


Figure 3: This figure presents the ablation on learning rate.

In addition, I pick some other weight decay 1e-3, 5e-4, 1e-4 ,5e-5 to test the sensitivity to weight decay. We can observe that the change weight decay does not affect the performance significantly. The results are included in Figure. 4.

**Visualization** For the interpretation, the original paper gives some visualization based on Grad-Cam [5]. Here I pick 4 images from the test set. The visualization of positive return is shown in Figure. 3. We can find that the model puts emphasis on the change of curves and can capture the bars.

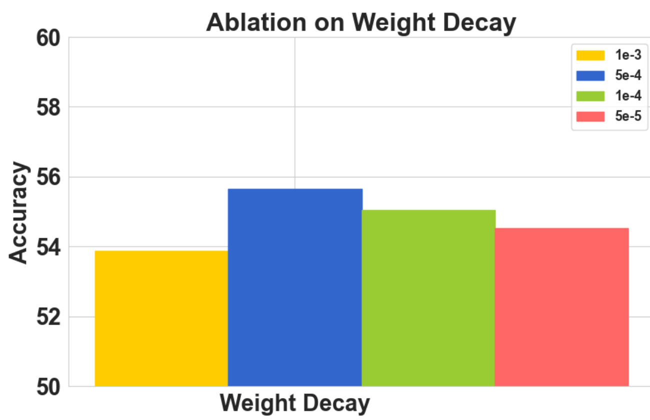


Figure 4: This figure presents the ablation on weight decay.

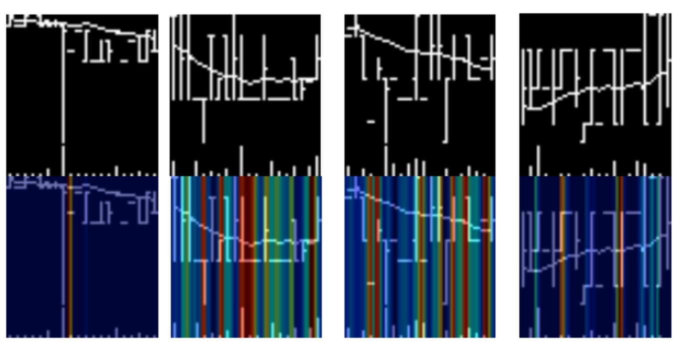

Figure 5: This figure presents the visualization.

## References


[1] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[2] Jingwen Jiang, Bryan T Kelly, and Dacheng Xiu. (re-) imag (in) ing price trends. *Chicago Booth Research Paper*, (21-01), 2020.

[3] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[5] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[6] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.