MAFS 6010Z Project 2: (Re-)Imag(in)ing Price Trends

XIA Yiqiao yxiaat@ust.hk, Department of Mathematics, HKUST

1. Introduction

This project is to replicate Jingwen Jiang, Bryan Kelly and Dacheng Xiu (2021) and make price trend prediction for 20-day return. The author believed that CNN models could detect more information in images, which enable it to produce accurate forecasts that are differentiated from traditional stock-level predictors. I managed to replicate the initial stage of the paper, and the model scored mean **accuracy 0.51** and mean **loss 0.74.**

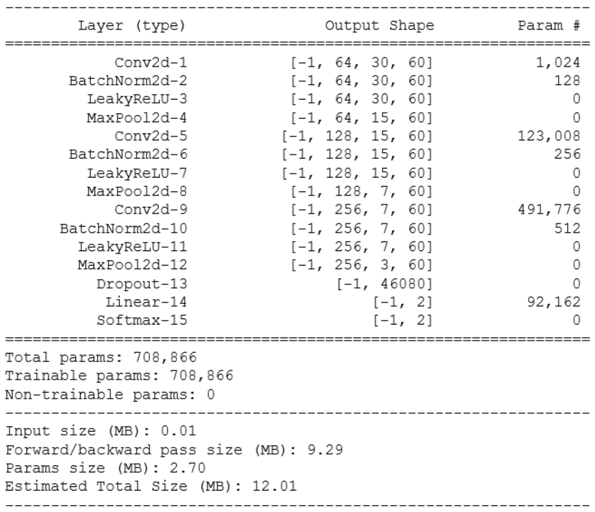
2. Pre-processing

Data Labeling

- $label(Ret_{20d}) = \begin{cases} 0 \ \ if \ Ret_{20d} > 0 \\ 1 \ \ else \end{cases}$

Data Split

- The first 8-year (1992-1999) data are split into **70% train data** and **30% validation data** randomly.
- The rest 20-year (2000-2019) data are used as test data.



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1           [-1, 64, 30, 60]           1,024
       BatchNorm2d-2           [-1, 64, 30, 60]             128
         LeakyReLU-3           [-1, 64, 30, 60]               0
         MaxPool2d-4           [-1, 64, 15, 60]               0
            Conv2d-5          [-1, 128, 15, 60]         123,008
       BatchNorm2d-6          [-1, 128, 15, 60]             256
         LeakyReLU-7          [-1, 128, 15, 60]               0
         MaxPool2d-8           [-1, 128, 7, 60]               0
            Conv2d-9           [-1, 256, 7, 60]         491,776
      BatchNorm2d-10           [-1, 256, 7, 60]             512
        LeakyReLU-11           [-1, 256, 7, 60]               0
        MaxPool2d-12           [-1, 256, 3, 60]               0
          Dropout-13                [-1, 46080]               0
           Linear-14                    [-1, 2]          92,162
          Softmax-15                    [-1, 2]               0
================================================================
Total params: 708,866
Trainable params: 708,866
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 9.29
Params size (MB): 2.70
Estimated Total Size (MB): 12.01
----------------------------------------------------------------
```

CNN Model Summary

3. Model Construction

Blocks

- Each block is consisted of 1 convolution function, 1 leaky ReLu function and 1 Max Pool function.
- Three blocks are added to the model.

Dropout

- There is a 50% dropout in the full connected layer.

Parameter settings

- Padding = (16, 1) for the first convolution function, and (2,1) for others.
- Stride = (5,3) for all convolution functions.
- Kernel size = (5, 3) for all convolution functions.
- Kernel size = (2, 1) for all Max Pool functions.
- k = 0:01 for Leaky ReLu functions.

4. Model Training

Initialization

- **Xavier initializer** is applied for weights in each layer to promote faster convergence.

Loss Function

- Cross-entropy loss is used for model valuation, which is defined as

$$L(y, \hat{y}) = -ylog(\hat{y}) - (1-y)\log(1-\hat{y})$$

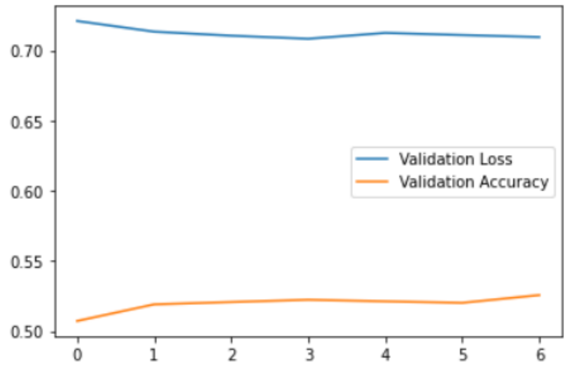
Batch

- Data are split into batches with **batch size 128** for training validation and test purpose.
- For each layer, there is a **batch normalization** process to reduce covariate shift.

Early Stopping

- Model is trained by batch, and model parameters are **saved for every 20 batches.**
- For every 200 batches, the main function evaluate the model using the validation data and records the mean validation loss. If the validation loss is **above** the best historical validation loss by more than **3 times**, the training process will be early stopped.

5. Model Performance – Loss and Accuracy



Accuracy and Loss for CNN Model

6. Analysis and Conclusion

The number of total parameters, as well as parameter settings are consistent with the description in the paper. Overall, I successfully replicated the paper, while the out-of-sample accuracy is slightly below the author's practice.

After checking the data, the 2% accuracy deviation may due to some **dirty data that are mislabeled** (e.g. N/A and 0 return). Another reason may be that **the model might be improved by increasing the patience of early stopping**. The paper ended training after 2 epochs, while my model ended within 1/3 epoch.

7. References

[1] Jiang, Jingwen and Kelly, Bryan T. and Xiu, Dacheng, (Re-)Imag(in)ing Price Trends (December 1, 2020). Chicago Booth Research Paper No. 21-01

[2] Gu, Shihao, Bryan Kelly, and Dacheng Xiu, 2020, Empirical asset pricing via machine learning, Review of Financial Studies 33, 2223-2273.