Project 2: Paper Replication Study

1   Project Requirement

This project as a warm-up aims to explore basic techniques in machine learning.

1. Pick up ONE (or more if you like) favourite dataset below to work. If you would like to work on a different problem outside the candidates we proposed, please email course instructor about your proposal.

2. Team work: we encourage you to form small team, up to **FOUR** persons per group, to work on the same problem. Each team just submit **ONE** report, *with a clear remark on each person's contribution*. The report can be in the format of either a *poster*, e.g.

   ```
   https://github.com/yuany-pku/2017_math6380/blob/master/project1/DongLoXia_
                                poster.pptx
   ```

   or *technical report within 8 pages*, e.g. NIPS conference style (preferred format)

   ```
   https://nips.cc/Conferences/2019/PaperInformation/StyleFiles,
   ```

   with source codes such as Python (Jupyter) Notebooks with a detailed documentation.

3. For Kaggle contests, please register your team with name in the format of math4995_lastname, so that we could easily find your results on Kaggle. For example, a team with Shawn Zhu and Kate Wong would be named by math6010z_Zhu_Wong.

4. In the report, show your proposed scientific questions to explore and main results with a careful analysis supporting the results toward answering your problems. If possible, you should include your Kaggle contest score or rating in the report. Remember: scientific analysis and reasoning are more important than merely the performance tables. Separate source codes may be submitted through email as a GitHub link, or a zip file.

5. Submit your report by email or paper version no later than the deadline, to the following address (aifin.hkust@gmail.com) with a title "MAFS6010z: Project 2"

2 Paper replication I: Empirical Asset Pricing via Machine Learning

The fundamental goal of asset pricing is to understand the behavior of risk premiums. However, risk premium is difficult to measure: market efficiency forces return variation to be dominated by unforecastable news that obscures risk premiums. This paper predicts the expected return and identifies informative predictor variables via machine learning methods, which facilitates more reliable investigation into economic mechanisms of asset pricing. Now you are required to replicate some results of this paper based your understanding of it, and write a report about your work.

The requirements of this paper replication project are as follow:

- The machine learning methods used in this paper include linear regression (OLS, elastic net), dimension reduction (PLS, PCR), generalized linear models, trees (gradient boosting trees, random forest) and neural networks. Please try to replicate **at least 6 methods** of them (e.g., OLS, elastic net, PLS, PCR, random forest, neural networks, etc. Please note that if you choose OLS, OLS-3 should also be included; and if you choose neural network, NN1 to NN5 are included. Besides, robust loss function should also be considered. See details in the paper), and analyze your results specifically. Hints on parameter choice are presented in the paper.

- Include the variable importance (section 2.3 of the paper) in your analysis. You do not need to replicate all the figures in section 2.3, but you are encouraged to investigate it carefully.

- Note that this paper uses a 'recursive performance evaluation scheme'. You are also required to evaluate your result by this method. For more details of this method, please refer to the paper and its supplementary material. The PPT presented in class about this project may also be helpful to you.

- As you can know from the paper (section 2.1), predictive characteristics include firm characteristics, sic code and macroeconomic predictors. Firm characteristics and sic code are provided in the original dataset of this paper, and the 8 macroeconomic predictors are constructed following Welch and Goyal (2008), which are not directly provided in the original dataset of this paper. Hence, you may construct the predictors by yourself according to the description in Welch and Goyal (2008), for instance, see
`https://christophj.github.io/replicating/r/replicating-goyal-welch-2008/`.

- The portfolio forecast part of the paper (section 2.4) is not compulsory for you to replicate.

You may access the paper and the supplementary material via:
`https://dachxiu.chicagobooth.edu/download/ML.pdf`
or
`https://academic.oup.com/rfs/article/33/5/2223/5758276`.
Meanings of characteristics of the data are provided in the supplementary material.

The original dataset (4.05GB) can be obtained at
`https://dachxiu.chicagobooth.edu/download/datashare.zip`.

The zip file is about 1.64GB. Please be patient since it may take you about 6 hours to download the data. Another fast access can be via
`https://www.dropbox.com/s/zzgjdubvv23xkfp/datashare.zip?dl=0`

3 Paper replication II: *(Re-)Imag(in)ing Price Trends*

3.1 Background

We are targeting to replicate the following paper by Jingwen Jiang, Bryan Kelly and Dacheng Xiu:
`https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3756587`.

This paper explores convolutional neural networks that flexibly learn price patterns as images that are most predictive of future returns. The raw predictor data are images – stock-level price charts, from which authors model the predictive association between images and future returns using a convolutional neural network (CNN). They claims that by using CNN they can automatically identify context-independent predictive patterns which can gave more accurate return predictions, translate into more profitable investment strategies and are robust to variations.

In the empirical designs, they first embeds 1D time series data in a higher dimensional space, representing it as a 2D image depicting price and volumes. Then they feed each training sample into CNN to estimate the probability of a positive subsequent return over short (5-day), medium (20-day) and long (60-day) horizons. Afterwards, they use CNN-based out-of-sample predictions as signals in a number of asset pricing analyses. Finally, they attempt to interpret the predictive patterns identified by the CNN.

3.2 Replication studies

In this reproduction process, we mainly focus on understanding the data preparation (how to transfer 1D time series data to 2D images representing historical market data), model design (CNN architecture design and mechanism behind it), workflow design (from training to model tuning and finally to prediction), performance evaluation and finally the interpretation part.

1. **Data**

   The sample runs from 1993-2019 based on the fact that daily opening, high, low prices. In the original paper, authors construct datasets consisting three scale of horizons (5-day, 20-day, 60-day), Here we just collect the 20-day version. The total size of data is 8.6G in a zipped file (802.9MB). The download link of data is:

   `https://dachxiu.chicagobooth.edu/download/img_data.zip`

   or a fast access

   `https://www.dropbox.com/s/njehqednn8mycze/img_data.zip?dl=0`

   with iPython image processing demo in

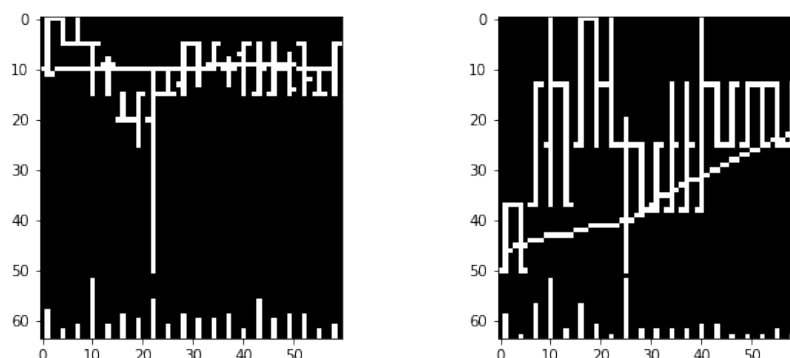   `https://dachxiu.chicagobooth.edu/download/img_demo.html`.

Figure 1: Examples of 20-day Image with volume bar and moving average line

We already transformed the OHLC charts into images following the same procedures introduced in the paper (Section 2). Current images have the same resolution (64 * 60) and added with moving average lines(MA) and volume bars(VB). Some example figures is shown in Figure 1.

Images labels take value **1** for positive returns and **0** for non-positive returns. In addition, we use **2** to mark the NaN value. In the simplest terms, you need to complete a two-class classification problem, and use the CNN model to predict whether the trend is 'down' or 'up' for the input image. For detail of data and label file, please refer to appendix.

2. **Architecture Design**

Why use CNN? Since CNN impose cross-parameter restrictions that dramatically reduce parameterization and embed a number of tools that make the model resilient to deformation and re-positioning of important objects in the image. A core building block consists of three operations: convolution, activation and pooling. In the paper, for 20-day image, they build a baseline CNN architectures with 3 conv blocks and connected with a fully connected layer as a classifier head. You should refer to the design of the conv block in the original paper (including the selection of the size of the convolution kernel, the selection of the convolution method, the design of the pooling layer and the selection of the activation function, etc) Figure 2 shows a diagram of 20-day CNN model proposed in the paper, just for your reference.

3. **Working Flow**

Data split First, consider dividing the entire sample into training, validation and testing samples. In the original paper, they use first seven-year sample (1993-1999) to train and validate model, in which 70% of the sample are randomly selected for training and the remaining 30% for validation. The remaining twenty years of data comprise the out-of-sample test dataset. You should consider follow the same format in case better comparson with the original paper.

Loss and evaluation You can simply treat the prediction analysis as a classification problem. In particular, the label for an image is defined as y = 1 if the subsequent return is
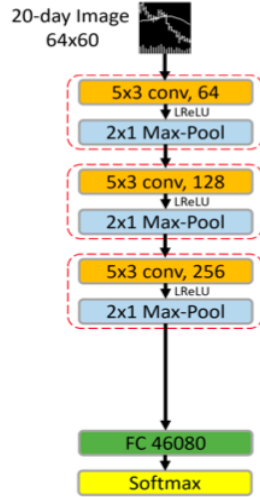
Figure 2: Diagram of CNN model

positive and y = 0 otherwise. The training step minimizes the cross-entropy loss, which is the standard objective function for classification problem, which define as:

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

where $\hat{y}$ is the prediction and y is the ground truth.

To measure the classification accuracy, a true positive (TP) (true negative (TN)) occurs when a predicted probability of greater than 50% coincides with a positive realized return (a probability less than 50% coincides with a negative return, respectively). False positives and negatives (FP and FN) are the complementary outcomes. We calculate classification accuracy as:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

For more evaluation metrics or methods, like Sharpe Ratio, please refer to the original paper.

Training process The author adopt several ways to combat over-fitting issue and aid efficient computation. For example, they applied the Xavier initialization for weights in each layer, which guarantees faster convergence by scale the initial weights. Other techniques like dropout, batch normalization and early stopping may also improve performance. We recommend to refer to the training details mentioned in the paper 3.3 when training the baseline model.

4. **Extensions**

- For ablation studies and testing robustness, we suggest you follow what original paper mentioned in Appendix B. For example, you can perform the same sensitivity analysis

of the CNN prediction model to alternate choices in model architecture (e.g. varying the number of filters in each layer or varying the number of layers, like the paper shows in Table 18)

- Another direction that can be used as an extension is exploring of the interpretability of the CNN model in Chapter 6 of the original paper. Though interpreting a CNN model is quite difficult due to its stacks of non-linear structures, you can imitate what the author did in Part 6.3, using a visualization method (Grad-CAM) to understand how different image examples activate the regions of the CNN to trigger 'up' or 'down' return predictions.

- What's more, we encourage you not being limited to simple binary classification task, since the label files we provided consist more meaningful attributes, containing both categorical and numerical values. For example, you can use the same 20-day horizon images to train your model to predict the return trend of different subsequent **y**-days even the detailed return values. (**y** can be 5, 20 even larger). In this way, you can prove more firmly that using CNN can automatically identity robust and transferable predictive features.