Deep Learning Project: China Equity Index Prediction Contest

Cyril de Lavergne, Martin Studer, Oscar Bergqvist
Department of Mathematics
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
{cdldl, msstuder,cowbergqvist}@connect.ust.hk

Abstract

The goal of the report use state of the art techniques to predict returns in equity index high frequency trading. We implement the Rocket-algorithm, a Resnet, and an Attention LSTM and also fit a ensemble of those methods. The ensemble performs best with an average correlation of 0.04008.

1 Introduction

While deep learning methods are able to obtain state of the art results on tasks such as processing of visual or auditive information or playing Go, classical statistical methods are still state of the world of asset management. This is often explained by the fact that data sets in asset management are too noisy, small, and there might not be a clear relationship between the data and the variable to be predicted. The goal of this report is to try to challenge this status quo in the world of equity index high frequency trading.
We use a total of three different approaches. The first approach is the novel so called Rocket-algorithm. It uses suitable random feature transforms and feeds them into a XGBoost algorithm (section 3.1). The second approach is using a Resnet (section 3.2), while the third approach is to use an Attention LSTM (section 3.3). We use the predictions obtained by all 3 models to fit an ensemble model (section 3.4) hoping that this improves performance. In section 4 we present our results and in section 5 we present our conclusions.

2 Stationarity conditions

Statistical analysis of asset prices involves the stationarity hypothesis that when time is shifted, the joint probability distribution does not change. The weak stationary assumption holds when asset prices first two moments are time-invariant. Asset prices indeed involve trend that need to be removed for time series to be stationary. Differentiation is the standard in to remove trend from time series. We [4] denote asset prices as C and returns are therefore obtained as such:

$$r(t, \Delta t) = ln(C(t + \Delta t)) - ln(C(t))$$

t is defined as a amount of time between two snapshot of prices. In our paper, we consider it to be the close to close asset return.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

3 Deep Learning Models

3.1 Rocket

Random Convolutional Kernels as a transform, and using the transformed features as the input to XGBoost is well established [3],[20]. It presents a method for unsupervised learning of convolutional kernels for a feature transform for time series input, based on a multilayer convolutional architecture with dilation increasing exponentially in each successive layer. The method is demonstrated using the output features as the input for XGBoost.

3.2 Resnet

After the celebrated victory of AlexNet at the LSVRC2012 classification contest, deep Residual Network [10] was arguably the most groundbreaking work in the computer vision/deep learning community in the last few years. ResNet makes it possible to train up to hundreds or even thousands of layers and still achieves compelling performance.

According to the universal approximation theorem, given enough capacity, we know that a feedforward network with a single layer is sufficient to represent any function. However, the layer might be massive and the network is prone to overfitting the data. Therefore, there is a common trend in the research community that our network architecture needs to go deeper.

Deep networks are hard to train because of the notorious vanishing gradient problem — as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitely small.

The core idea of ResNet is introducing a so-called “identity shortcut connection” that skips one or more layers as in Figure 1.

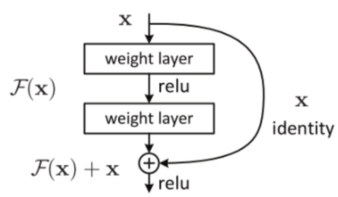

Figure 1: Resnet CNN architecture

3.3 Attention LSTM

RNN based architectures are hard to parallelize and can have difficulty learning long-range dependencies within the input and output sequences. RNN have also long-range dependencies issues. The Transformer models all these dependencies using attention

Instead of using one sweep of attention, the Transformer [2],[8],[12], [14] uses multiple “heads” to capture different aspects of inputs. Position encoding is used to maintain the order of features and the decoder masks the “future” tokens when decoding a certain word.

In addition to attention, the Transformer uses layer normalization and residual connections to make optimization easier. The key feature of layer normalization is that it normalizes the inputs across the features. Batch normalization however are difficult to apply to recurrent connections and statistics are computed across the batch and are the same for each example in the batch.

Intuitively, the attention mechanism allows the decoder to “look back” at the entire sentence and selectively extract the information it needs during decoding.

A dual-stage attention-based recurrent neural network (DA-RNN) [1], [19] to capture the long-term temporal dependencies appropriately. In the first stage, we introduce an input attention mechanism to adaptively extract relevant driving series (a.k.a., input features) at each time step by referring to the previous encoder hidden state. In the second stage, we use a temporal attention mechanism to select relevant encoder hidden states across all time steps.

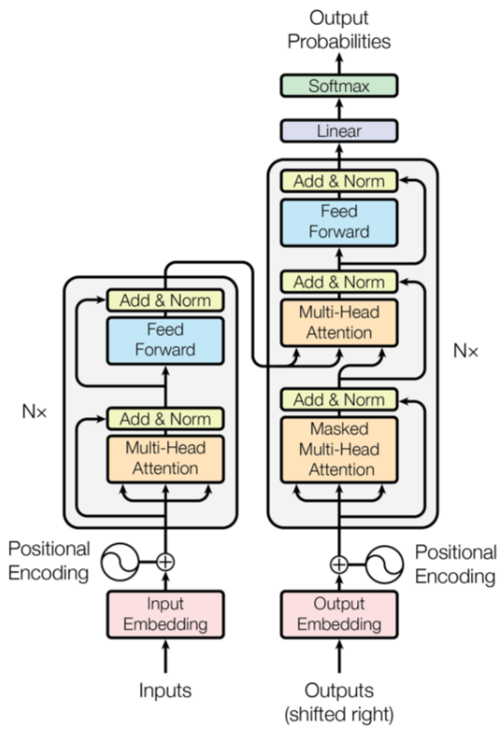

Figure 1: The Transformer - model architecture.

Figure 2: Transformer architecture

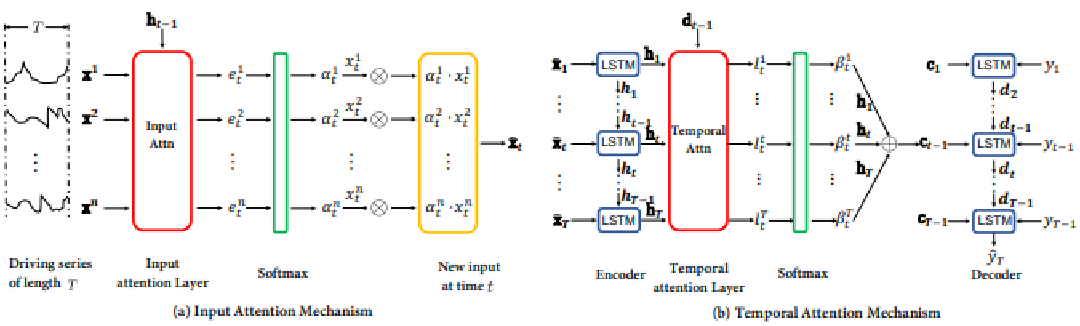

Figure 3: Transformer architecture

4 Results

Rocket results are as follows:

```
correlation checked finished!
average: 0.027140986199349605, std: 0.03843341234853268.
19/12/12 16:04:12:55 Evaluation process finished.
Time Elapsed: 30s
```

Figure 4: Rocket results

Resnet results are as follows:

```
average: 0.02574631533045734, std: 0.03262362564948854.
19/12/12 16:04:12:09 Evaluation process finished.
Time Elapsed: 30s
```

Figure 5: Resnet results

Attention LSTM results are as follows:

```
average: 0.03559141052332065, std: 0.0507807467405701.
19/12/12 16:04:12:12 Evaluation process finished.
Time Elapsed: 28s
```

Figure 6: Attention LSTM results

We ensemble the predictions of the 3 algorithms described above namely rocket, resnet and attention-lstm and we obtain the following results with a correlation below 30% of the linear y_{hat}:

```
correlation checked finished!
average: 0.04008246119629151, std: 0.03540989689250112.
19/12/10 14:02:12:12 Evaluation process finished.
Time Elapsed: 121s
```

Figure 7: Final Results

We have checked the cumulative performance of our strategy using the sign of our predictions times y_{10min}, it leads to the overall performance in Figure 8.

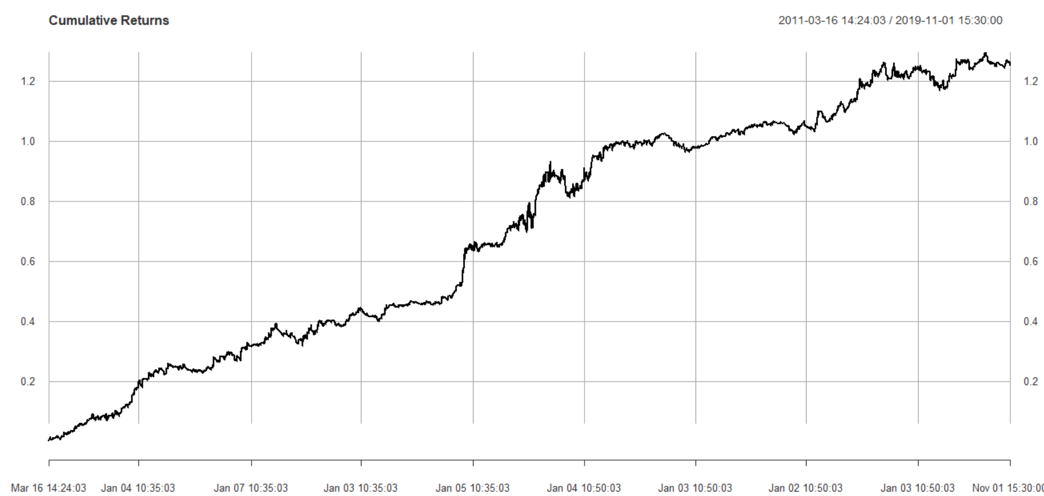

Figure 8: Cumulative Results

Note that cumulative returns were divided by 200 which is 3seconds * 200 = 10minutes. We have also not taken into account eventual transaction costs, spread and slippage due to lack of data.

5 Conclusion

In summary we can say that Rocket and Resnet both gave decent results, while the Attention LSTM performed even a bit better. However, the ensemble of all methods was superior compared to each method individually.
We have shown that diverse state of the art algorithms can in fact with great accuracy, especially when combined, provide superior results to predict the Chinese financial markets.

6 Discussion

- Cost-sensitive methods: class weights and sample weights, cross-entropy modifications [6] [7]
- Anomaly detection inputs or threshold enhancing performance with [9], [16], [18]
- RandAugment [5]
- Bigan instead of stacked denoising autoencoder [11]
- Lightweight and Dynamic Convolutions [17]
- Optimal cell for both CNN and RNN [13], [15]

7 Contribution

1. Rocket and report: Martin
2. Resnet and report: Oscar
3. Attention-LSTM and report: Cyril

References

[1] Financial time series forecast using dual attention rnn. `https://github.com/Regressionist/Stock-prediction-Dual-Attention-based-RNN-`.

[2] Implementation of different rnn architectures for time series prediction for currency markets. `https://github.com/vinit97/RNN-Models-for-Time-Series-Prediction`.

[3] Geoffrey I. Webb Angus Dempster, Francois Petitjean. Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels. 2019.

[4] Rama Cont. Empirical properties of asset returns: stylized facts and statistical issues. 2000.

[5] Jonathon Shlens-Quoc V. Le Ekin D. Cubuk, Barret Zoph. Randaugment: Practical automated data augmentation with a reduced search space. 2019.

[6] Yue Geng and Xinyu Luo. Time series classification from scratch with deep neural networks: A strong baseline. 2016.

[7] Yue Geng and Xinyu Luo. Cost-sensitive convolution based neural networks for imbalanced time-series classification. 2018.

[8] Isaac Godfried. Attention for time series forecasting and classification.

[9] Yujing Wang Chao Yi Congrui Huang-Xiaoyu Kou Tony Xing Mao Yang Jie Tong Qi Zhang Hansheng Ren, Bixiong Xu. Time-series anomaly detection service at microsoft. 2019.

[10] Jonathan Weber Lhassane Idoumghar Pierre-Alain Muller Hassan Ismail Fawaz, Germain Forestier. Deep learning for time series classification: a review. 2019.

[11] Trevor Darrell Jeff Donahue, Philipp Krähenbühl. Adversarial feature learning. 2016.

[12] Sangyeon Kim and Myungjoo Kang. Financial series prediction using attention lstm. 2019.

[13] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[14] Huicheng Liu. Leveraging financial news for stock trend prediction with attention-based recurrent neural network. 2018.

[15] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *ICML*, 2018.

[16] Sanjay Chawla Raghavendra Chalapathy. Deep learning for anomaly detection: A survey. 2019.

[17] Wan-Lei Zhao Chang-Hui Liang Run-Qing Chen, Guang-Hui Shi. Pay less attention with lightweight and dynamic convolutions. 2019.

[18] Wan-Lei Zhao Chang-Hui Liang Run-Qing Chen, Guang-Hui Shi. Sequential vae-lstm for anomaly detection on time series. 2019.

[19] Haifeng Chen Wei Cheng Guofei Jiang-Garrison W. Cottrell Yao Qin1, Dongjin Song. A dual-stage attention-based recurrent neural network for time series prediction. 2017.

[20] Aaron Courville Yoshua Bengio† and Pascal Vincent. Representation learning: A review and new perspectives. 2014.