Specialized Neural Network for Extracting Financial Trading Signals: The Alpha Discovery Neural Network

Abstract

Genetic programming (GP) is currently the leading method for automated feature generation in financial applications. It utilizes reverse Polish notation to denote features and subsequently performs an evolutionary procedure. Nevertheless, with the advancements in deep learning, more effective feature extraction instruments have become accessible. This research introduces the Alpha Discovery Neural Network (ADNN), a customized neural network architecture designed to autonomously generate a variety of financial technical indicators using established knowledge. Our primary contributions are threefold. Firstly, we employ domain-specific expertise in quantitative trading to formulate sampling guidelines and the objective function. Secondly, we substitute genetic programming with pre-training and model pruning techniques to enable a more streamlined evolutionary process. Thirdly, the feature extraction components within ADNN can be interchanged with various other feature extractors, resulting in the creation of diverse functions. Empirical findings demonstrate that ADNN can produce more distinct and informative features in comparison to GP, thereby effectively augmenting the existing pool of factors. Fully connected and recurrent networks demonstrate superior performance in extracting information from financial time series compared to convolutional neural networks. In practical scenarios, the features generated by ADNN consistently enhance the revenue, Sharpe ratio, and maximum drawdown of multi-factor strategies when contrasted with investment strategies that do not incorporate these factors.

1 Introduction

Predicting the future returns of stocks is a paramount and demanding endeavor in the field of quantitative trading. Numerous factors, including historical price, volume, and a company's financial information, can be employed to forecast the future returns of stocks. Typically, researchers categorize features derived from price and volume as technical indicators, while those derived from a company's financial data are classified as fundamental data. Various well-known multi-factor models have been introduced to address this task, and numerous established technical and fundamental factors have been developed. For instance, the Fama-French Three-Factor Model utilizes three crucial factors that furnish the majority of the information required to elucidate stock returns. Subsequently, the Fama-French Five-Factor Model and numerous other factors have been formulated by domain experts. Nonetheless, two limitations exist. Firstly, recruiting human specialists is quite costly. Secondly, humans are unable to create certain nonlinear features from data with high dimensionality. Consequently, both academic scholars and institutional investors have increasingly focused on the task of automated financial feature engineering.

Feature engineering is a procedure that uncovers the connections between features and expands the feature space by deducing or generating novel features. During this operation, new features can be created by combining pre-existing features. A more explicit explanation is that algorithms employ operators, hyper-parameters, and existing features to construct a new feature. Occasionally, feature construction and feature selection can be integrated into a single process. These methodologies encompass wrapper, filtering, and embedded techniques. Filtering is straightforward but yields suboptimal results; it merely employs certain criteria to select a feature and can sometimes aid in overseeing the feature construction process. The wrapper method exhibits strong performance by directly utilizing the model's outcomes as an objective function. Consequently, it can treat an independently trained model as a newly generated feature. Nevertheless, a substantial quantity of computational resources and time are necessary. Embedded is an approach that employs generalized factors and a pruning method to choose or amalgamate features, serving as an intermediate option between filtering and wrapper techniques.

2 Related Work

With the progression of deep learning, an increasing number of researchers are utilizing neural networks to derive features from raw data and subsequently incorporating a fully connected layer to modify the feature's output. Similarly, a trained model signifies a newly developed feature. Researchers have leveraged it on pattern recognition tasks, employing a CNN model to construct facial descriptors, and this method generates features that possess considerably more information than the previous method. Experiments have been conducted on this task, employing a deeper and wider convolutional neural network. Recurrent neural networks have been used to pre-locate feature-rich regions and successfully construct more refined features. In a text classification task, recurrent neural networks have been utilized to build a rule-based classifier among text data, wherein each classifier represents a portion of the text. A network structure that uses both a recurrent neural network and a convolutional neural network to extract text information has been proposed. Utilizing a neural network's robust fitting capability, we can generate highly informative features by customizing the network architecture for diverse industries. In financial feature engineering tasks, researchers have commenced employing neural networks to provide an embedding representation of financial time series. More specifically, LSTM has been utilized to embed various stock time series, followed by adversarial training to perform binary classification on a stock's future return. Well-designed LSTM has been adopted to extract features from unstructured news data, subsequently forming a continuous embedding. The experimental outcomes indicate that these unstructured data can furnish substantial information and are highly beneficial for event-driven trading. A Skip-gram architecture has been employed to learn stock embedding, inspired by a valuable knowledge repository formed by fund managers' collective investment behaviors. This embedding can more effectively represent the varying affinities across technical indicators. Adopting a similar concept, we employ a neural network to provide a concise embedding of extended financial time series.

3 Methodology

The ADNN's network architecture is structured in a specific way. The primary contributions of this innovative network structure are: 1) ADNN employs Spearman Correlation as its loss function, mirroring the practices of human quantitative investment. Furthermore, the sampling guidelines adhere to economic principles. 2) A significant, derivable kennel function is introduced as a substitute for the non-derivable operator. 3) We utilize pre-training and pruning in place of the GP's evolutionary process, resulting in enhanced efficiency.

In each back-propagation cycle, ADNN randomly selects data from a certain number of trading days and subsequently computes the Spearman Coefficient between the factor value and factor return for each of those days. The number of days should be greater than 3, and incorporating information from multiple trading days enables the neural network to achieve a more consistent convergence. Quantitative investors prioritize the relative strength of each stock on a given trading day over its absolute strength. Therefore, performing calculations for each trading day and employing the Spearman Coefficient as the loss function is justifiable.

We posit that there are a certain number of stocks pertaining to a given trading day in each batch. The input tensor has a specific shape because there are a certain number of samples, and five categories of time series: the opening price, high price, low price, closing price, and volume. Each time series has an input length. We also designate the output tensor as the factor value, possessing a particular shape. The factor return tensor has a specific shape, denoting the profit we can obtain from this asset over an extended duration. The holding period's length is defined. Here, we presume that all feature extractors are Multi-layer Perceptrons (MLPs), simplifying the provision of a general mathematical description. In the experimental section, we will present the experimental outcomes based on more intricate and varied feature extractors.

4 Experiments

We utilize daily trading data from the Chinese A-share stock market, encompassing the daily opening, high, low, closing prices, and trading volume over the preceding 30 trading days. The raw data is standardized using its time-series mean and standard deviation derived from the training set. Both the mean and standard deviation are computed from the training set. We endeavor to employ these inputs to forecast the stock return for the subsequent 5 trading days (utilizing 3-15 trading days is advisable). Furthermore, we must adhere to market regulations when devising a trading strategy.

Extensive experiments have been performed to identify appropriate hyper-parameters. For each experiment, 250 trading days constitute the training set, the ensuing 30 trading days serve as the validation set, and the subsequent 90 trading days function as the testing set. The generated factors maintain a high Information Coefficient (IC) throughout the subsequent 90 trading days. Most significantly, we emphasize a counter-intuitive configuration: the training period should not surpass 250 trading days due to the non-stationary nature of financial features. If we mandate a feature to function effectively over an extended duration, we will only encounter this feature in an over-fitting scenario. Consequently, we devise a rolling forecast framework wherein we automatically identify potent features for each trading day. Each autonomously generated feature will have its own period of prominence on that particular trading day. Moreover, these factors not only perform effectively on this single day but also maintain their efficacy for several trading days, exhibiting a gradual decline.

To ensure an equitable comparison, the identical configuration is implemented for the GP algorithm. The logic of this algorithm references related work. Moreover, the input data's period and type must be consistent. In this paper, we scrutinize the performance of the constructed features from diverse angles. Typically, institutional investors employ the Information Coefficient (IC), to quantify the amount of information conveyed by a feature. For diversity, cross-entropy is utilized to gauge the distance between the distributions of two distinct features on the same trading day.

5 Results

The network structure can equip ADNN with different deep neural networks. In order to show the general situation, we equip ADNN with 4 fully-connected layers. Each layer has 128 neural, tanh activate function, L2 Regularization, and dropout technic. This general and simple setting is enough to beat the GP. We put forward three schemes help to show how ADNN beat the GP. Only GP means only using genetic programming, Only ADNN means only use ADNN to construct factors, GP&ADNN means use GP's value to initialize ADNN and then construct factors. All the experiments are conducted out of the sample.

Table 1 shows that Only ADNN is better than Only GP, which means ADNN outperforms GP on this task. And we also find that GP&ADNN is the best, it means that our method can even improve the performance of GP.

Table 1: The performance of different schemes.

Object	Information Coefficient	Diversity
Only GP	0.094	17.21
GP&ADNN	0.122	25.44
Only ADNN	0.107	21.65

In real practice, we should leverage the constructed factors to form a multi-factor strategy and compare its performance with GP. The specific strategy setting is same as section 3.4, and we have repeated this experiment on different periods of time. The long-term backtest result is shown in Table 2, Only ADNN always has better performance than the Only GP. It shows that ADNN has also beaten the SOTA in real practice. Similar to the conculsions made above, if we combine these two methods together, the combined factors' strategy has the best performance in backtesting.

Table 2: Strategy's absolute return for each scheme.

Time	Only GP	GP&ADNN	Only ADNN	ZZ500
Train:2015.01-2015.12 Test: 2016.02-2016.03	+2.59%	+5.74%	+4.52%	+1.67%
Train:2016.01-2016.12 Test: 2017.02-2017.03	+5.40%	+10.26%	+8.33%	+2.53%
Train:2017.01-2017.12 Test: 2018.02-2018.03	-5.27%	-4.95%	-4.16%	-6.98%
Train:2018.01-2018.12 Test: 2019.02-2019.03	+13.00%	+15.62%	+15.41%	+13.75%

All the results shown above is based on the most basic feature extractors. So will there be more powerful feature extractors to discover knowledge from financial time series? And what is the suitable input data structure for financial time series?

Table 3 shows that, basically, all neural networks can produce more diversified features than using GP. But temporal extractors are especially better at producing diversified features, such as LSTM and Transformer. As for TCN, the author who put forward this network structure proves its ability to capture the temporal rules buried in data. However, there is a huge difference. TCN relies on a convolution neural network, but LSTM and Transformer still contain recurrent neural networks (Normally, the transformer uses a recurrent neural network to embedded the input data). The existence of a recurrent neural network structure may contribute to the difference in diversity. For Le-net and Resnet, they don't provide us with more informative features. It looks like that the convolution network structure is not suitable to extract information from the financial time series.

Table 3: The higher are the information coefficient (IC) and diversity, the better is their performance. Normally, a good feature's long-term IC should be higher than 0.05, but it cannot be higher than 0.2 in an A-share market.

Type	Network	IC	Diversity	Time
Baseline	GP	0.072	17.532	0.215 hours
Vanilla	FCN	0.124	22.151	0.785 hours
	Le-net	0.123	20.194	1.365 hours
Spatial	Resnet-50	0.108	21.403	3.450 hours
	LSTM	0.170	24.469	1.300 hours
Temporal	TCN	0.105	21.139	2.725 hours
	Transformer	0.111	25.257	4.151 hours

In practical applications, we integrate conventional factors with those generated by ADNN to formulate a quantitative investment strategy. Our objective is to ascertain whether ADNN can enhance the factor pool and improve upon the traditional multi-factor strategy.

We establish a commonly employed multi-factor strategy to assess its performance in a real-world context. Within the training set, samples whose returns rank in the top 30% for each trading day are designated as 1, while those ranking in the bottom 30% are labeled as 0. The remaining samples in the training set are discarded. Following the training of these features using XGBoost in binary logistics mode, the prediction outcome reflects the probability of a stock exhibiting exceptional performance in the subsequent 5 trading days. It designates the 50 features constructed by human experts as PK 50, the features constructed by ADNN as New 50, and the features constructed by both GP and PK as GP-PK 50. In separate experiments, we use XGBoost to pre-train both PK 50 and New 50 in the training set and then using the weight score from XGBoost to choose the 50 most important features as Combined 50. This feature selection process only happens once, and only be conducted in training set.

Table 4 shows the results of the backtesting.

Table 4: Back testing starts from Jan 2019 to June 2019. The investment target is all A-share, except for the stock can't be traded during this period of time. Strategy's commission fee is 0.5%. SR refers to Sharpe Ratio, MD represents Max- Drawdown.

Type	Target	Group	Revenue	MD	SR
	ZZ500	Stock Index	19.60%	13,50%	1.982
Baseline	HS300	Stock Index	18.60%	20.30%	1.606
	PK	PK 50	24.70%	18.90%	2.314
		GP 50	17.60%	25.30%	1.435
GP		GP-PK 50	25.40%	14.80%	2.672
		New 50	20.60%	15.80%	2.189
Vanilla	FCN	Combined 50	29.60%	15.70%	3.167
		New 50	18.00%	16.90%	1.800
	Le-net	Combined 50	27.50%	16.40%	2.921
Spatial		New 50	19.90%	15.40%	1.962
	Resnet-50	Combined 50	29.30%	17.20%	2.787
		New 50	19.50%	13.00%	2.205
	LSTM	Combined 50	29.90%	15.00%	3.289
Temporal		New 50	22.40%	14.70%	2.440
	TCN	Combined 50	26.90%	16.80%	2.729
		New 50	21.10%	15.90%	2.203
	Transformer	Combined 50	27.20%	15.10%	2.806

As shown in Table 4, HS300 and ZZ500 are important stock indices in the A-share stock market. Revenue represents the annualized excess return, by longing portfolio and shorting the index. The max drawdown is the worst loss of the excess return from its peak. The Sharpe ratio is the annually adjusted excess return divided by a certain level of risk. These indicators can show the strategy's performance from the perspective of both return and risk.

For the New 50, although they have higher IC than the PK 50, their overall performance is not always better than PK 50. Because the overall performance of a multi-factor strategy is determined by both diversity and information volume (IC), we guess the diversity of PK 50 is remarkably higher than the diversity of New 50. We also did experiment to verify this guess. Thus, although every single new factor is better than the old factor, their overall performance not always be better. ADNN's diversity is larger than the GP, but for further research, making ADNN's diversity even larger is still badly needed. In the real world use case, all investors have their own reliable and secret factor pool, what they want is that the new constructed factors can bring in margin benefits. Thus, they will use both new and old factors to do trading. That's the reason why Combined 50 can represent ADNN's contribution in the real situation. In all cases, Combined 50 is better than PK 50 and GP-PK 50, which means that the ADNN not only perform better than GP, but also can enrich investors' factor pool.

6 Conclusion

In this research, we introduce the Alpha Discovery Neural Network (ADNN), a system capable of autonomously generating financial features from raw data. We have meticulously crafted its network architecture in accordance with economic principles and furnished it with a variety of sophisticated feature extractors. Empirical results indicate that ADNN can generate features that are more informative and diverse than those produced by the benchmark method in this specific application. In practical scenarios, ADNN also demonstrates superior revenue, Sharpe ratio, and maximum drawdown compared to genetic programming. Furthermore, different feature extractors assume distinct roles. We have conducted numerous experiments to validate this observation and endeavor to comprehend its functionality. For future research, we intend to employ this framework to automatically generate valuable features based on companies' fundamental data and sentiment data.