International Journal of u- and e- Service, Science and Technology
Vol.8, No. 7 (2015), pp.35-48
http://dx.doi.org/10.14257/ijunesst.2015.8.7.04

Quantitative investment Based on Artificial Neural Network Algorithm

Xia Zhang

Department of Software Technology, Shenzhen Polytechnic
zhangxia@szpt.edu.cn

Abstract

Abstract: Financial investment has become an important issue, there are many trading strategies and parameters based on quantitative models, this paper use neural network algorithm to optimization strategy parameters, various combinations of optimization strategies, as well as the evolution of new strategies to generate better returns. The empirical results show that this method has a stable and substantial return on investment, neural network can be used as an aid for decision making investments in securities.

Keywords: *quantitative investment, trading strategies, Probabilistic Neural Network, Mixed Neural Network, multi-factor model*

1. Introduction

Quantitative investment has become a major investment in the global, investment amount has occupied about 30% share. In 2013 a quantitative investment perfect show of the year, which benefited the most in North America, the entire net capital inflows quantitative funds reached 60 billion U.S. dollars. Quantitative investment is the use of mathematics, physics, geometry, psychology and even bionics knowledge modeling through analysis, conduct financial markets, judgment and process transactions [1].

Research indicates that the current optimal strategy is more excellent in the coming period of performance, you can dynamically adjust trading strategies, namely the choice of the optimal strategy for the current simulation trading, empirical results show that this method has a more stable and substantial return on investment [2], it is an auxiliary securities investment decisions.

Artificial neural network is a new high-tech research area

Since 1980s, involving a variety of disciplines, attracting many neurophysiologists, psychologists, mathematician, computer and information scientists, engineers to research and apply. Artificial neural network has entered the stage of steady development, its features and advantages, mainly in three aspects: a self-learning function, associative memory function, with the ability to find the optimal solution for high-speed. Neural network model mainly consider the network topology connection characteristics of neurons, learning rules. Currently, nearly 40 kinds of neural network model, widely used in the field of artificial neural network, including pattern recognition, signal processing, knowledge engineering, expert systems, optimize the combination of robot control. In 1990, Specht based on radial basis function neural network and Parzen window function density estimation method based on the proposed A new artificial neural networks - probabilistic neural network (Probabilistic Neural Network, PNN), which utilizes Priori probability-Bayes optimal law of the sample and determine the principles for a new classification of samples, but also from multivariate normal points Restrictions cloth and other conditions, and in the course of operation can be calculated PNN posterior probability of a new input sample is classified, so Provide explanations for the results. PNN its simple structure, fast training process and good ability to promote superior

ISSN: 2005-4246 IJUNESST

Potential performance has been widely used in many fields, such as stock forecasts. PNN has to be applied in face recognition, credit evaluation, quality assessment, remote sensing image classification, forecasting river bed morphology and sensor fault diagnosis and other fields. In this paper, the radial basis function - probabilistic neural network model is explored. Then the accuracy and performance of the predictions are studied with comparisons.

2. Multi-Factor Model Based on Neural Network

The use of quantitative investment strategies is a mathematical modeling and optimization process. We research relative gains. Relative income, also known as positive earnings (Alpha), the excess return (ER) or implementation return, it is the total income exceeds the income or part of the risk-free benchmark yield.

For the complexity of the stock market composed of multiple factors, to go through a certain quantization process can be an important factor in the conversion of variable form. The biggest advantage of neural networks is that complex systems without knowing its internal mechanisms [2], just make sure the input and output vectors, a network structure can be constructed to simulate the system. To construct the network structure, the input and output variables of the network must be determined firstly.

Multi-factor quantitative investment model is an important stock selection models, the basic idea is to find the most relevant indicators yields, and find Alpha factor from micro data. Assuming a stock at time t , yield R_t, geometric growth G(R) of the stock, then

$$1 + G(R) = \{\prod_{r=1}^{T} (1 + R_t)\}^{\frac{1}{T}} \tag{1}$$

T represents the current sample.

A basic neural network processing units of neurons is usually in the form of multi-input, single-output nonlinear devices.

This neural network model based on statistical theory, the training convergence time is short and easy to practice, but it limits its network structure only as a classifier output. Priori probability-Bayes theorem, sample and minimum Bayes risk-based decision rules for a new sample into Line classification, resulting in a probabilistic neural network (referred to as PNN).

Radial basis function network (Radial Basis Function Neural Network congregation, referred RBFNN) [26] is raised by J. Moody and Darken in the late 1980s, the hidden layer activation function is the radial basis function. In the structure, RBF network has three single hidden layers feed forward structure. Since it simulates the human brain partial adjustment, mutual acceptance domains covering neural network structure, so RBF network is a network of local approximation can be used for function approximation and pattern recognition. In fact, PNN network, wavelet network, generalization regression neural networks are improved by the radial basis function network.

In this paper, PNN network structure and radial basis function neural network (referred RBFNN) are selected to combine a new network structure: the probability of hybrid radial basis function neural network (referred to as the RBF-PMNN). RBF-PMNN is not a PNN classifier with only features, but also has the function of RBFNN fitting characteristics. On the other hand, due to the network performance affected by the network parameters, the EM algorithm and genetic algorithms are applied to optimize the network parameters.

3. Auto Encoders

In fact deep learning model is a hidden layer (hidden layer number≥2) neural network model. After a layer of a layer of abstraction, the original data are finally classified, But the multilayer neural network model is highly non convex. Because of the existence of a large number of local minima, and poor convergence, it’s very difficult to obtain good learning results In training. Over fitting in the practical application, the phenomenon is serious. Learning model in depth training, a large amount of data is used for unsupervised learning network layer by layer, and then the parameters after training can be used as the initial value of the supervised neural network.

Since the encoder is a kind of unsupervised learning methods commonly used network, is a special kind of neural network model. From the transmission of information, it is a kind of coding method, it makes the information loss as small as possible. When unsupervised network training is over, a label samples can be used for supervised learning, by the back propagation algorithm.

Based on the self encoder, deep learning model is the idea of: The original data by self encoder learning, get the code layer H1, then the code layer H1 encoder to obtain self learning, get the code layer H2, so each layer is obtained by neural network hidden layer depth. Put the encoder coefficient and the result as the initial value of the underlying neural network model, to in-depth learning of the neural network.

In order to make the results obtained from the encoder is more robust, reduce the over fitting, Bengio (2008) proposed a method called self encoder noise reduction method. In each of the training from the encoder. The encoder from the nodes in the input layer is set to a certain probability.

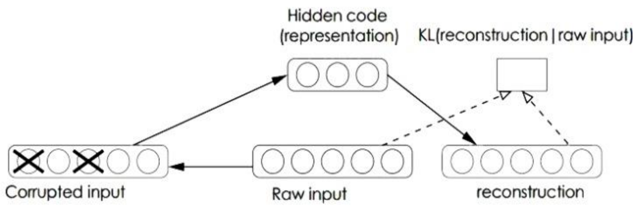


Figure 1. AutoEncoders Schematic

As shown in Figure 1, Need to reacquire update parameters in each iteration, to make the Results through the encoding and decoding can be reconstructed as close as possible to the original input. The practice proves that, through this robust self coder training, deep learning model, is better performance than from encoder model or a restricted Boltzmann machine to obtain the model.

Deep learning model training is divided into two parts. The first step is to use the no label or labels removed data, Noise reduction by self encoder, Layer by layer, to learn hidden layer coefficient in deep web. The second step is to put the encoder noise reduction from learning gain coefficient as the coefficient of initial value network, using the label data, to learn on the deep network supervised learning by the back propagation algorithm.

Iterative algorithm for large data optimization problem

We want to deal with is shaped like a formula optimization problems, it can be iteratively calculated by gradient descent algorithm. Because the prediction error is obtained through training all the data , so the calculation requires all the training data set used. For large-scale data, the gradient descent algorithm iterative speed will be very slow. This can improve the speed of optimization operation using stochastic gradient descent method. Need each iteration gradient facts obtained are cumulative results of partial derivatives of the prediction error of each sample.

Stochastic gradient descent method is done on a single sample parameter update after update after using the parameters to calculate the next sample prediction error for a sample of the updated parameters calculated. Therefore, when using the formula for the n th sample of the parameter is updated, the value of the parameter to spend a sample is calculated after the update. Due to the iteration of each sample, the parameters to be optimized are constantly updated, if the sample amount of iterative computation is very big, the speed of convergence of stochastic gradient descent method will be significantly better than gradient descent method. Moreover, stochastic gradient descent method are more likely to avoid local optimal solution. Compared with the gradient descent method, the problem of stochastic gradient descent method is: The significant differences between samples, each iteration of the parameter update is not towards the "optimal" direction, In the fixed learning rate, the algorithm does not usually convergence, But will value near the oscillation in the optimal finally.

Deep learning Mini batch models commonly used in training (Mini-Batch) descent method is an iterative optimization algorithm based on stochastic gradient descent method established on the. The algorithm is for each parameter update is not in accordance with the single sample, but according to a batch consisting of several samples to calculate the gradient parameter update. Therefore, the mini batch method can be regarded as a compromise between the ordinary gradient descent method and stochastic gradient descent method. On the one hand, the high efficiency of the iteration algorithm; On the other hand, in batches of various types of sample selection equilibrium conditions, parameters are updated at each iteration are approximately toward the direction of the "optimal". At the same time, compared with the stochastic gradient descent method, mini batch down each iteration method are used to calculate a plurality of samples, High efficiency makes full use of the Matlab scientific calculation software to quantify the calculation.

4. Algorithm Principle

Probabilistic neural network using Bayes method for pattern recognition problem, a common solution is given, and the establishment of a Bayes classifier. Bayes methods can be made to infer the input data space, can provide more information on which part of the data in order to select from for guidance.

4.1. Probabilistic Neural Network (PNN)

PNN has the function of statistical classification; it is based on the a priori probability theory Bayes decision Bayes law and the minimum risk. Through non-parametric probability density function estimation. Parzen window function, the calculation process can get the class conditional probability density, then classified sample. This neural network model based on statistical principles has the advantages of short training time, without connection weights training samples, trained samples by a given hidden layer constitutes direct.

As a basic neural network processing units of neurons, usually in the form of a multi-input, single-output nonlinear devices, the general structure of the model shown in Figure 2.

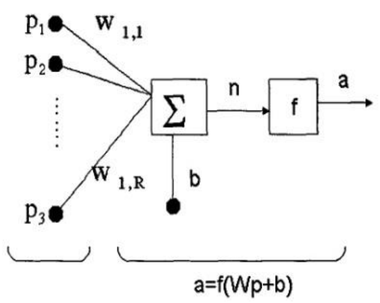


Figure 2. Multiple Input Neurons

p is an input signal, corresponding to the elements w_i of the weight matrix w, b is the threshold, and it is all the weighted input and accumulated to form a net input,

$$n = w_{1.1}p_1 + w_{1.2}p_2 + \cdots + w_{1.R}p_R$$

Neuron output can be written as:

$$a = f(w_p + b)$$

(1) Threshold type:

$$f(n_i) = \begin{cases} 1 & n_i \geq 0 \\ 0 & n_i < 0 \end{cases} \tag{2}$$

(2) A piecewise linear model

$$f(n_i) = \begin{cases} 1 & n_i \geq n_2 \\ an_i + b & n_i \leq 0 \leq n_{2i} \\ 0 & n_i < 0 \end{cases} \tag{3}$$

(3) S-type function:

$$f(n_i) = \frac{1}{1 + \exp(-n_i/c)^2} \tag{4}$$

(4) Competitive function:

$$f(n_i) = \begin{cases} 1 & n_i = \max_j\{n_j\} \\ 0 & \text{other} \end{cases} \tag{5}$$

Among them, the competitive decision-making function has been used in probabilistic neural network output as the output of Bayes decision.

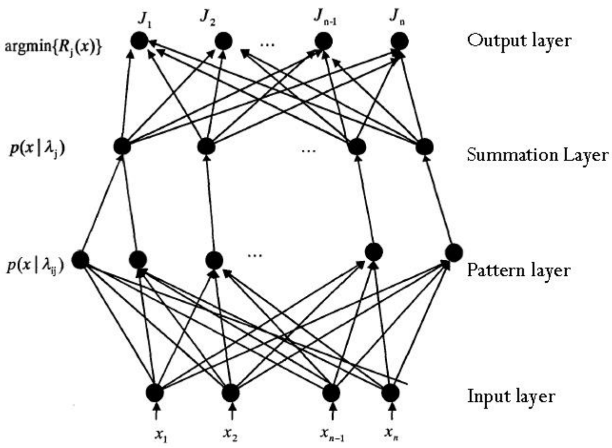


Figure 3. PNN Network Structure

Input layer: the input vector x into the model layer.

Model layer: The total number of nodes is $\sum_{i=1}^{M} G_i = G$

Summation Layer: According $p(x|\lambda_j) = \sum_{i \in G_j} \pi(\lambda_{ij})\, p(x|\lambda_{ij})$. This layer nodes is M,

$$\text{Connection weights} \begin{cases} \pi(\lambda_{ij}), & \text{when the i\_th belongs to the class j} \\ 0, & \text{when the i\_th does not belong to the class j} \end{cases}$$

Output layer (also called competitive layer): get M nodes $p(x|\lambda_i)$ as input by the summation layer output layer, and then get the output in accordance with the model output layer, the network final output is J class.

4.2. Radial Basis Function Neural Network (RBFNN)

Hidden layer activation function is the radial basis function. In the structure, RBF network has three single hidden layer feed-forward structures [6]. Since it simulates the human brain partial adjustment, mutual acceptance domains covering neural network structure, so RBF network is a network of local approximation, it can be used for function approximation and pattern recognition.

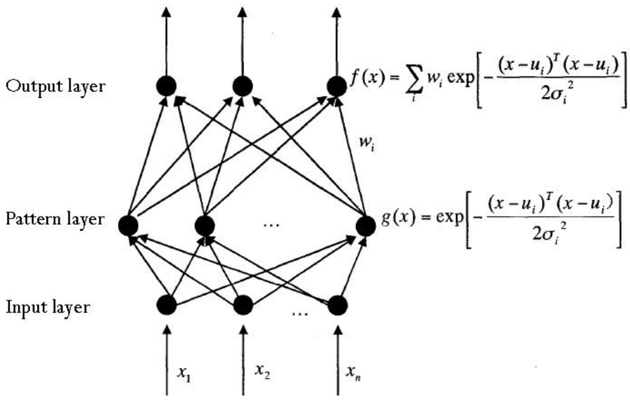


Figure 4. RBFNN Network Structure

Layer 2 with K modes layer unit ($2 \le K \le N$), the choice of the Gaussian kernel function as activation function:

$$g_i(x) = exp\left[-\frac{(x-u_i)^T(x-u_i)}{2\sigma_i^2}\right] \tag{6}$$

u_i and σ_i is the center of the vector and the smoothing factor in the i-th layer pattern node, output layer is also referred to as linear layer, the output function:

$$f(x) = \sum_{i=1}^{K} w_i g_i(x) = \sum_{i=1}^{K} w_i exp\left[-\frac{(x-u_i)^T(x-u_i)}{2\sigma_i^2}\right] \tag{7}$$

w_i is the connection weights from the i-th layer pattern to the output layer unit.

4.3. Hybrid Neural Network Structure (RBF-PMNN)

The difference between RBFNN and PNN:

(1) RBFNN model layer nodes is less than or equal to PNN pattern layer nodes (*i.e.*, number of training samples).

(2) RBFNN through repeated training to determine three parameters: u_i, σ_i, w_i; PNN training only once to determine the u_i.

(3) RBFNN through linear processing output; PNN also need to calculate the class conditional density by category, which through a competitive output.

(4) RBFNN supervised learning through training; PNN by category labeled self-supervised classification.

From the difference between the two can be found, PNN network does not take into account the overlapping and interlacing patterns among different categories, although the training time is short, but limited performance. Therefore, we can combine the advantages of PNN and RBFNN, create a new network structure.

RBFNN and PNN network structure similar to a pattern layer having function to select and activate Parzen window function, such as a Gaussian kernel function $g_i(x)$, and the third layer PNN and RBFNN's output is y^r, y^p ,they can be expressed as:

$$y^r = \sum_{i=1}^{K} w_i g_i(x) \tag{8}$$

$$y^p = \{y_j^p\}_{j-1}^m = \left\{ \sum_{i=1}^{k_j} g_i(x) \right\}_{j-1}^M \tag{9}$$

m is the number of categories of PNN, k is the RBFNN model layer nodes, $k_j$for the j-th node of PNN categories. Combines the advantages of RBFNN and PNN, we can create a new network architecture –probability mix radial basis function neural network (Radial Basis Function - Probabilistic Mixed Neural Network, referred to as the RBF-PMNN).

RBF-PMNN a total of four layers, namely the input layer, pattern layer, summation layer, linear layer (output layer). Training set of data after treatment, arranged according to different categories and different regional or sub regional connectivity typical sample extraction, as the center of the vector model layer. [4]

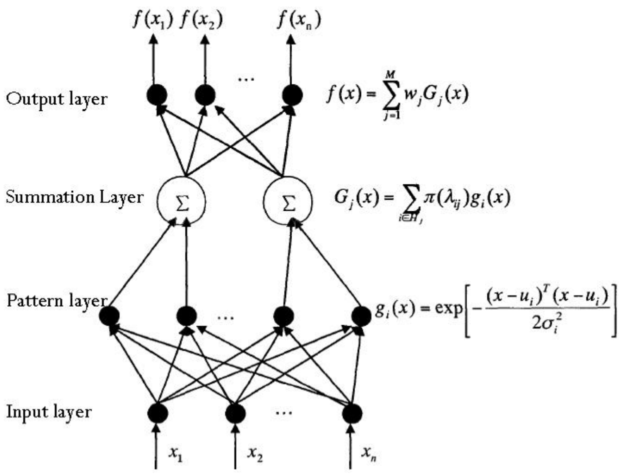


Figure 5. RBF-PMNN Network Structure

Input layer is an N-dimensional input vector x;

Model layer using RBFNN model layer, It has H modes layer unit, G is the training set samples; sample H can be divided into M disjoint categories, H_j is the number of training samples J-class nodes. Pattern layer activation function still options:

$$g_i(x) = exp\left[-\frac{(x-u_i)^T (x-u_i)}{2\sigma_i^2} \right] \tag{10}$$

u_i and σ_i is the center of the vector and the smoothing factor in the i-th layer pattern node.

Summation layer using PNN summation layer.

Output layer is also referred to as linear layer, the output function:

$$G_j(x) = \sum_{i \in H_j} g_i(x) \pi(\lambda_{ij}) = \sum_{i \in H_j} exp\left[-\frac{(x-u_i)^T (x-u_i)}{2\sigma_i^2} \right] \tag{11}$$

$\pi(\lambda_{ij})$ is the i-th observations, it belongings to J class and with probability$\pi(\lambda_{ij})$.

Linear layer using RBFNN's linear layer as the output of the network.

$$f(x) = \sum_{j=1}^{M} w_j G_j(x) = \sum_{i=1}^{M} w_j \pi(\lambda_{ij}) \sum_{i \in H_j} exp\left[-\frac{(x-u_i)^T(x-u_i)}{2\sigma_i^2}\right] \quad (12)$$

w_i is the connection weights of j-th node in linear layer.

After determining the network structure of RBF-PMNN, to determine the parameters of the network through learning and training:

$$u = [u_1, u_2, \dots u_H], \sigma = [\sigma_1, \sigma_2, \dots \sigma_H],$$
$$\text{and } \omega = [\omega_1, \omega_2, \dots \omega_H]$$

4.4. Learning Algorithm Of Radial Basis Function Neural Network Hybrid Probability (RBF- PMNN)

When the network structure of RBF-PMNN determined, the network parameters should be determined by training and learning. There are many learning algorithm used to determine the parameters of the network. Here's an RLS-GD adaptive (Recursive Least Squares - adaptive gradient descending) combined algorithm

u_i and σ_i are determined using an adaptive iterative gradient descent algorithm (adaptive gradient descending, AGD), w_i recursive least squares method to determine the choice of learning algorithm (Recursive Last Squares Algorithm, RLSA), and the combination of the two training adaptive get a joint algorithm RLS-GD.

Joint algorithm steps as follows:

(l) Given initial weight vector $w(0)$, the inverse correlation matrix initial value $p(0)$, smoothing factor $\sigma_i(0)$, error iteration termination value ε.

(2) Randomly selected radial basis function center vectors u_i ($2 \leq i \leq N$, N is the number of training samples).

(3) The beginning of iteration, k = 1.

(4) Calculate G of each class

$$G_j(x,k) = \sum_{i \in H_j} exp\left[-\frac{(x(k)-u_i)^T(x(k)-u_i)}{2\sigma_i(k)^2}\right] \quad (13)$$

(5) Calculate the Kalman gain $g(k)$ and an inverse matrix of the training sample $p(k)$

$$g(k) = \frac{p(k-1)G(x,k)^T}{\lambda + G(x,k)p(k-1)G(x,k)^T} \quad (14)$$

$$p(k) = \frac{1}{\lambda}[p(k-1) - g(k)G(x,k)p(k-1)] \quad (15)$$

(6) Calculate the output node corresponding to the error signal

$$\varepsilon(k) = Y(k) - G(x,k)W(k-1) \quad (16)$$

(7) Update:

$$w(k) = w(k-1) + g(k)[Y(k) - G(x,k)W(k-1)] \quad (17)$$

$$\sigma_i(k) = \sigma_i(k\text{-}1) + \alpha(\sigma_i(k\text{-}1) - \sigma_i(k\text{-}2)) + \eta \sum_{t=1}^{k} \lambda^{k-1}(x(t)-u_i)^T(x(t)-u_i))^T \quad (18)$$

$$(T(t) - y(t)) \frac{w_j(t)}{\sigma_i(k-1)^3} \sum_{i=1}^{H_j} exp[-\frac{(x(t)-u_i)^T(x(t)-u_i)}{2\sigma_i^2}] \quad (19)$$

(8) Update the accumulated error energy J

$$J(k) - \lambda J(k-1) = \frac{1}{2}(Y(k) - G(x,k)W(k))^2 \quad (20)$$

(9) If $j(k) \geq \varepsilon$, then $k = k + 1$, turn (4); otherwise the end of the training.

5. Trading Strategies

Establish a Multi-factor quantitative stock-picking system, the entire system create a database style factor firstly, the factor may be a lot of attention on the market to bring together. After initial processing (*i.e.,* data preprocessing), it can be used to obtain statistics factor income data analysis and stock data, then the system use multiple metrics on the effectiveness of the style factor database for each factor multi-dimensional characterization, and pick out the Alpha which can produce a stable factor. Alpha factor in the completion of the selection, that is currently available to determine the Alpha factor, the next step is to integrate on the Alpha factor, and pick the appropriate stocks as an investment portfolio [3].

First established by RBF-PMNN stock volatility model, namely, RBFNN model layer consists of all the training set consisting of nodes, then the summation layer in accordance with their respective categories together, the different categories online layer obtained by adding the results of the weight of the output layer.

After the establishment of the Alpha factor, a fully automated quantitative investment can be achieved.

Assuming the T-th day's closing price is v, the yield (rate of return) on the T-th day, the number of differential calculation is defined as:

$$r_T = \log v_T - \log v_T = \log(v_T/v_{T-1}) \tag{21}$$

Historical volatility v_T can be written as:

$$v_T = (1-\alpha)\sum_{\tau\leq t} \alpha^{t-\tau}\, r_T^2 \tag{22}$$

T= 1,2 ... Weights a∈(0,1) determine the effects of past yields on volatility, which means a bigger, "memory", the longer and the greater the impact. (10) can be written as the following recursive form:

$$v_T = \alpha v_{T-1} + (1-\alpha)\, r_T^2 \tag{23}$$

$$v_0 = 0 \tag{24}$$

As the floating rate is relatively small, the number of singular characteristics, so it needs to be normalized, it is defined as the following normalization methods [1].

$$x_i = \frac{V_i - min(V_i)}{max(V_i) - min(V_i)} \tag{25}$$

6. Stock -Picking System

Simmons gecko-ended investment theory tells us that in the short-term directional prediction can be invested to capture short-term arbitrage opportunities. Trading strategies based on the depth of learning is learning by means of the depth of a large number of historical transaction data to learn, build predictive models to capture the short-term trading opportunities in the actual transaction. The proposed stock index futures based on the depth of learning first-use policy in the short term deep learning model to predict the ups and downs of stock index futures, stock index futures trading signals is then determined based on forecast results. The first is the depth of learning predictive models. In general, the shorter the time interval to predict if the predictive power of machine learning models will be stronger. We consider that the stock index futures one second grade high frequency data. In selecting input on the prediction model, the choice is the stock price in the short term, as well as changes in the price range, prices and orders committee appointed to sell to buy amount. Change output prediction model is the direction of the short-term future. Consider the ups and downs associated with the transaction, we recommend selecting the next downbeat larger sample as the sample of interest to machine learning models. When the depth of learning predictive model training is good, we can determine the future of the ups and downs according to the output prediction model.

But the forecast results here are not used to build trading strategies directly, for two reasons: first, the change is just a directional prediction, did not specify how much concrete chance of ups and downs, the amplitude is large enough; second, the high frequency fluctuations in the stock is limited, time is very short trading positions is difficult to bring the excess returns. Based on these two considerations, this paper trading strategies have two characteristics. First, a given threshold value, reaches or exceeds the threshold value, only the prediction score will trigger trading signals. Second, the opening and closing of the transaction by the buy and sell signals are triggered.

Although deep learning model is a classification model, it also gives the sample score (score is the output, the logic function is located between the [0,1]). For the change of stock price prediction model, prediction of Score 1 it can give rise and fall prediction Score 2. For example, stock prices rise judgment whether the future, the higher the Score1 sample, the higher probability of future prices. Corresponding, higher Score 2 sample, the higher the probability of future decline. It can set the threshold according to the score, score, which when exceeded a threshold, triggering a sell signal.

The trading strategy triggered by the trading signals. Opening day are short positions in state. According to the buy / sell signal trigger to build multi bin and short positions, positions of indefinite periods of time. If in the holds warehouse, a short signal trigger, and reverse positions immediately establish short positions, otherwise (such as continue to be bullish signal triggered) does not change the positions. If the holders of short positions, a bullish signal trigger, and reverse positions immediately set up multi bin, otherwise (*e.g.*, continue to have short signals triggered) does not change the positions.

Because the market noise and mutation of the, The model can accurately predict only a part of the stock price. In position, it should set a stop loss strategy, when the real-time price breakout stop line R, immediately stop the open. The stop price should use a recent issue trading signals when the stock price as a benchmark to calculate. The specific transaction strategy as shown below.

Table 1. Depth Learning Stock Price Prediction Feature Selection

Select the input variables	Description
Close Price	
Highest Price	
Low Price	
Open Price	
Orders offer poor	Sell offer - buying quote
Average price quote orders	(Buying offer + sale offer) / 2
Volume	
Commission appointed to sell to buy ratio of the amount	log (Commission amount to buy / sell amount Commission)
Orders depth	
Change positions	
The previous day's closing price	

The prediction model and the trading strategy model are based on week-frequency model strategy, specifically, to select the last trading day of each week as establishing stock positions in open trading day. According to the prediction model in front, stocks after T=5 trading day rose of ScoreUp, and the sharp drop of ScoreDown, it can create very convenient similar multi-factor model of stock trading strategies. Specifically to the combined size of 100 as an example, select the day before the 100 stock ScoreUp rose forecast as the largest bullish combination,

Thus at least it can be derived from the following three strategies:

(1) One way to do multi-strategy. Each trading day to buy the top 100 stocks of this system to make Bullish combination, stocks in the Combination to be configured in a manner of the same right to the funds.

(2) Long-short pairing strategy. Each trading day to buy the top 100 stocks to make Bullish portfolio by short selling before the 10% bearish portfolio. Bullish and bearish portfolio total amount of money equal to the combination, the combination of internally configured in the manner of stocks funds right.

(3) The right to hedge funds and other strategies. Each trading day the top 100 stocks to buy to make Bullish combination, while the short CSI 300 stock index futures to hedge. Single transaction gains and losses = 0.8 × (see more portfolio returns + short index futures gains). See perform multiple combinations of stocks configure the funds right manner. The above strategy, the way to do more strategic risk is generally relatively large, long and short pairing strategy due to the need short selling short, there may be some problems on the actual operation, so it take the third trading strategies that hedge funds and other rights strategy where equity portfolio transaction fees for unilateral 3 ‰.

Deep learning network of the first hidden layer has 400 nodes, the second hidden layer has 200 nodes, the output layer has 2 nodes (output ScoreUp and ScoreDown in turn, the price is forecast to rise or fall). The hidden layer training iteration unsupervised learning for 50 times, a number of iterations supervised learning for the 400 time. The model is applied to predict the time-consuming in a single sample within 5ms. Sample forecasting results are shown in Table 2, gives rise to predict ScoreUp and drop prediction ScoreDown is not at the same time, the stock price in the average change in 5 trading days.

Table 2. Depth Learning Stock Predictions

The sample mean of 5 days after the price change				
Prediction score	Before 5%	The top 10%	Before 15%	Before 20%
Score Up	1.97%	1.39%	1.11%	0.92%
Score Down	-0.66%	-0.45%	-0.33%	-0.26%
ScoreUp-ScoreDown	1.97%	1.48%	1.24%	1.09%
ScoreDown-ScoreUp	-0.64%	-0.59%	-0.56%	-0.50%

In order to ensure the authenticity and reliability model, the simulation of real-time tracking method was used to test the model. Determine the positions of the day, its profit and loss analysis, just read in from a database by the end of the trading day closing time data, depth of learning model for forecasting. According to the prediction model of the choice to buy scoring stocks portfolio, go back to the database, read a week after the transaction data, calculation of profit and loss.

In order to consider the combination of different size, *i.e.*, the number of shares for 10, 30, 50, 100, 200, 300, the empirical results of a hedging strategy. All data with the prediction model, when the combination of scale is smaller, the cumulative returns ratio is higher, but the withdrawal will also increase accordingly. This is because, when the rising more prediction ScoreUp, usually the corresponding up greater possibilities. Hedge after withdrawal is still large, is because when the number of stocks in the portfolio is less, the Shanghai and Shenzhen 300 stock index futures hedging effect is not obvious.

In general, the effect of rolling forecast model can forecast model slightly outperforms fixed good. With the combination of the size of the amplification factor effect, reduction, cumulative yields lower, but also a corresponding reduction in retracement. Considering the yield and its stability, portfolio size control around 100 more appropriate.

A month of turnover and price momentum is the mainstream of multi factor model of neutral, comparing the turnover of stock momentum hedge portfolio based on double factor and the deep learning prediction model of hedging portfolio, in the vast majority of the month, the deep learning hedge portfolio returns are better.

In addition to picking up the prediction ScoreUp as Alpha factor, but also picking up the predict ScoreDown as stock selection factor, or the difference between ScoreUp - ScoreDown as picking a comprehensive factor.

Consider these factors on cumulative five-speed performance, that is, each of the combination is relatively large when the top 160 stocks selected, Each factor of IC, IR and other manifestations such as shown in Table 3.

Table 3. Depth Learning Stock Predictions

Factor Name	IC	IR	Winning	Exchange ratio
The momentum for a month	-1.90%	0.75	54%	38%
Turnover of a month	-3.51%	2.03	66%	11%
Depth study up factor	3.92%	2.34	67%	42%
Depth study fell factor	-3.92%	2.62	62%	52%
Depth-learning factor	4.60%	3.23	68%	52%

The depth of learning strategies equity portfolio turnover rate is relatively high, a weekly cycle of multi-factor strategy positions, turnover in more than 40%. From the factor information ratio IR, the depth of each factor prediction models have to learn much bigger factor than the mainstream model of IR. From the industry distribution, the rolling forecast model portfolio size of 100 stocks chosen industry when each phase of a combination of the number of changes. From which we can see the policy portfolio stock turnover is relatively high, and their combination on the issue of choice in each period, there are big changes in the industry. The number of shares each phase combination of the industry median. Pharmaceutical, real estate, electrical components and other sectors of the stock is relatively popular strategy stock.

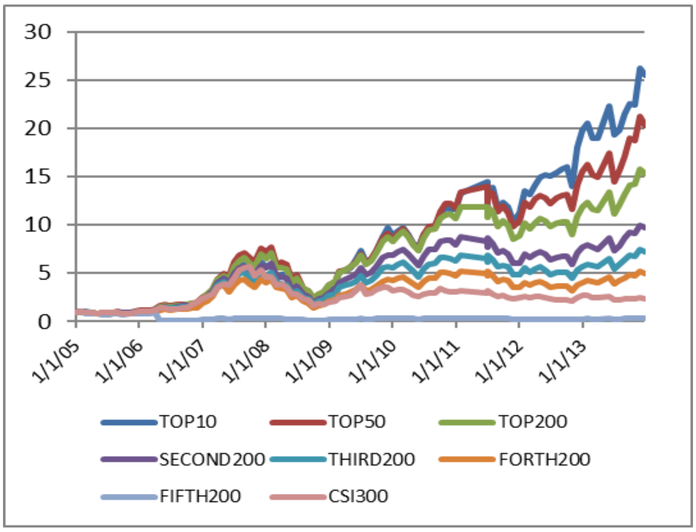


Figure 6. Full Market Stock Picking Net Movements And The Csi 300 Comparison Chart

Extracted from Shanghai and Shenzhen stock market data as the study of the whole sample set between 2005/01/01 to 2013/12/31, all the data from Wind database.

Since the stock market is high frequency trading, with the most recent year's data for training, and then select Alpha factor, according to the factor score, select the top 10, top 50, top 200 back-testing performed to assess the effect of neural network learning.

Full market stock picking (excluding ST, condemned the company, the shares of listed unhappy times a year, suspension) net movements and the CSI 300 comparison chart below, the initial value of 1.0.

Through the above illustration, you can see with a hybrid neural network (RBF-PMNN) stock selection, selected the top ten stocks far ahead of the CSI 300 Index, the effect is very obvious.

7. Conclusion

In this paper, considered the characteristics and the difficulty of predicting the stock market stock market, the hybrid neural network (RBF-PMNN) is proposed to solve the problem. By learning on the stock market depth data mining, a multi-factor stock selection models is established. An effective factor Alpha can produce benefits. Through the stock market data to predict the short-term price trend model, the equity portfolio and obtain excess returns can be filtered by the prediction mode. Compared with the traditional volume and price factors, the depth of learning algorithms by multi-factor can get a better rate of return.

References

[1] C. Yang and S. Sun, " Multi-agent artificial stock market model based on GNP", Computer Applications, vol. 26, no. 5, **(2006)** May, pp. 1217-1219.

[2] S. Dan and X.-Y. Zhang, "Stock market forecasting model based on artificial neural networks", journal of Jilin University(Information Science Edition, vol. 20, no. 4, **(2002)**, pp. 68-70.

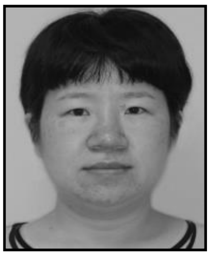
[3] S. Song and Y. Zhang, "MATLAB-based BP neural network model of software reliability", Information aspect, vol. 12, **(2009)**, pp. 15-18.

[4] S. Wang, Z. Li, E. Chen and B. Wang, "Improved BP Network in Intrusion Detection", ACTA ELECTRONICA SINICA, vol. 29, no.5, **(2001)**, pp. 119-123.

[5] Y.-K. Kwon and B.-R. Moon, "A Hybrid Neurogenetic Approach for Stock Forecasting, Neural Networks", IEEE Transactions on , vol.18, no. 3, **(2007)** May, pp. 851-864.

[6] P. Areekul, T. Senjyu, H. Toyama and A. Yona, “A Hybrid ARIMA and Neural Network Model for Short-Term Price Forecasting in Deregulated Market”, Power Systems, IEEE Transactions on, vol. 25, no. 1, **(2010)** February, pp. 524-530.
[7] R. Panxiang, S. Jianpeng, L. Zhaoyu, Y. Lin and D. Wenbo, “Simulation and Research of Boiler Combustion Process Based On the Improved RBF Neural Network”, International Journal of u- and e-Service, Science and Technology, vol. 6, no. 5, **(2013)**, pp. 79-88.
[8] S. Sun, C. Zhang and G. Yu, “A bayesian network approach to traffic flow forecasting, Intelligent Transportation Systems”, IEEE Transactions on , vol. 7, no. 1, **(2006)** March, pp. 124-132.
[9] Q. Zhang and A. Benveniste, “Wavelet networks, Neural Networks”, IEEE Transactions on , vol. 3, no. 6, **(1992)** November, pp. 889-898.
[10] X. Li, G. Chen, Z. Chen and Z. Yuan, “Chaotifying linear Elman networks, Neural Networks”, IEEE Transactions on , vol. 13, no. 5, **(2002)** September, pp. 1193- 1199.
[11] P. Zhang, “Probabilistic model of the three stock price index calculations”, Journal of Qingdao Technological University, vol. 27, no. 5, **(2006)**, pp. 119-123.
[12] H. Xiao-qi and W. Bo, “Fast discovering frequent patterns in financial time series data based on data mining”, University of Shanghai for Science and Technology, vol. 28, no. 4, **(2006)**, pp. 281-286.
[13] J. Cao, K. Yuan and H. X. Li, “Global asymptotical stability of recurrent neural networks with multiple discrete delays and distributed delays”, IEEE Transactions on Neural Networks, vol. 17, no. 6, **(2006)**, pp. 1646–1651.
[14] K. Prasad, D. C. Nigam, A. Lakhotiya and D. Umre, “Character Recognition Using Matlab’s Neural Network Toolbox”, International Journal of u- and e- Service, Science and Technology, vol. 6, no. 1, **(2013)** February, pp. 13-20.
[15] M. Yang and J. Lin, “Mathematica Base and Mathematical Software”, Dalian University of Technology Press, Dalian, China, **(2007)**.
[16] M. M. Gupta, L. Jin and N. Homma, “Static and Dynamic Neural Networks”, Wiley-Interscience, New York, NY, USA, **(2003)**.
[17] M. T. Hagan, H. B. Demut, M. H. Beale, “Nerual Network Design”,China Machine Press, **(2004)** September.
[18] “Science and Technology Products Research Center of Feisi”, Neural Network Theory and the Implementation of Matlab7, Electronic Industry Press, Beijing, China, **(2005)**, pp. 99-108.
[19] L. K. C. Lai and J. N. K. Liu, “Stock forecasting using Support Vector Machine, Machine Learning and Cybernetics (ICMLC)”, International Conference on , vol. 4, **(2010)** July 11-14, pp.1607-1614.
[20] H. Liu, D. Huang and Y. Wang, “Chaotic Dynamics Analysis and Forecast of Stock Time Series”, Computer Science and Society (ISCCS), 2011 International Symposium on, **(2011)** July, pp. 75-78, 16-17.
[21] L. Ran and G. Chaoyun, “The relevance analysis between electrical day peak load and meteorological index based on wavelet denoising and SVM”, Electric Utility Deregulation and Restructuring and Power Technologies, DRPT, Third International Conference on , **(2008)** April 6-9, pp. 788-793.
[22] L. Q. Yu and F. S. Rong, “Stock Market Forecasting Research Based on Neural Network and Pattern Matching, E-Business and E-Government (ICEE)”, 2010 International Conference on, **(2010)** May 7-9, pp. 1940-1943.
[23] W. Ma, Y. Wang and N. Dong, “Study on stock price prediction based on BP Neural Network”, Emergency Management and Management Sciences (ICEMMS), 2010 IEEE International Conference on, **(2010)** August 8-10, pp. 57-60.
[24] M. Gao, Q. Feng and J. Tian, “Building Logistics Cost Forecast Based on Wavelet Neural Network, MultiMedia and Information Technology”, MMIT '08, International Conference on, **(2008)** December 30-31, pp. 117-120.

Author



Zhang Xia, mainly research on e-commerce, artificial intelligence, distributed network, and have published dozens of scientific papers.

