COMPUTING PROJECT IN ALGO FINANCE
DETAILED PROPOSAL

NAME : ZHAN Hui
UNIVERSITY NUMBER : 3035249227
SUPERVISOR : Dr. Hilton K.H. Chan
TITLE : Stock Price Pattern Recognition and Trading Signal Release Based on Neural Networks
PLANNED SUBMISSION SEMESTER : Sem. 1, year 2016-17

Stock Price Pattern Recognition and Trading Signal Release Based on Neural Networks

CONTENTS

1. INTRODUCTION

As stated by efficient market hypothesis, the price of share fully captures all known information about the share. The market maps all of the information to a certain share price by a set of complex mechanisms without explicit rules. The history people study the methods predicting share price has been continued a great long period, during which many techniques are proposed to stimulate the market and forecast the stock price. Among the great deal of analysis methods, artificial neural network (ANN) is a recent focused machine learning method and widely applied into current market, to discover/recognize stock price patterns, predict stock trends and forecast share price.

It has been more than three decades that ANNs are applied to financial data analyzing, and there are numerous researches to improve and extend neural network to make it a more powerful financial analyzing measure. NN is a confirmed efficient analysis technique in either accuracy and executing speed. An important advantage of NN [4] points out, NN can extract rules without having them explicitly formalized, we only input the data into machines and then get the result without necessarily know the rules. That makes me judge NN as a rational technique to work out financial market problems, because market is a such mystery that no experts understand its rule pretty surely.

Stock price patterns, which could be accurately searched out by NN, always represent the specific conditions of the stock, and cluing to the trend of future changes in stock price. It is obvious that stock price patterns are useful to help researchers and invertors understand the stock states and make specific trading decisions.

In this project, I want to apply several neural networks techniques to discover and recognize short-term stock price patterns (of Hong Kong market, HKEX) automatically and instantly, and release specific trading signals. In the following sections of this proposal, I will enumerate some main stream research topic combining NN and its derivations with financial data analysis of recent 2 decades in section 2; the methodology of I approach my proposal in section 3, containing the data, technique, tool and evaluation measures I will use; work schedule and deliverables of this project in section 4 and 5 respectively.

2. LITERATURE REVIEW

2.1 Artificial Neural Networks (Back Propagation NN)

Artificial neural networks which created in 1950s, well-known by people in mid of 1980s for back propagation NN (BPN) proposed, and applied to financial data study in latter 1980s, is a recent focused, efficient and powerful analyzing tool to help people making trading decisions by several static share information. The most basic and widely used ANN in financial analyzing is BPN, which is commonly consisted of one input layer, single layer or multi-layer of hidden neurons and one output layer, the following Fig. 1 shows the structure of a four layers BPN. And in most practice of applying ANN, researchers would allocate one input neuron to deal with one specific data item (like volume, opening price), and one output neuron to represent one kind of result (like up-forward trend, good-stock-performance)

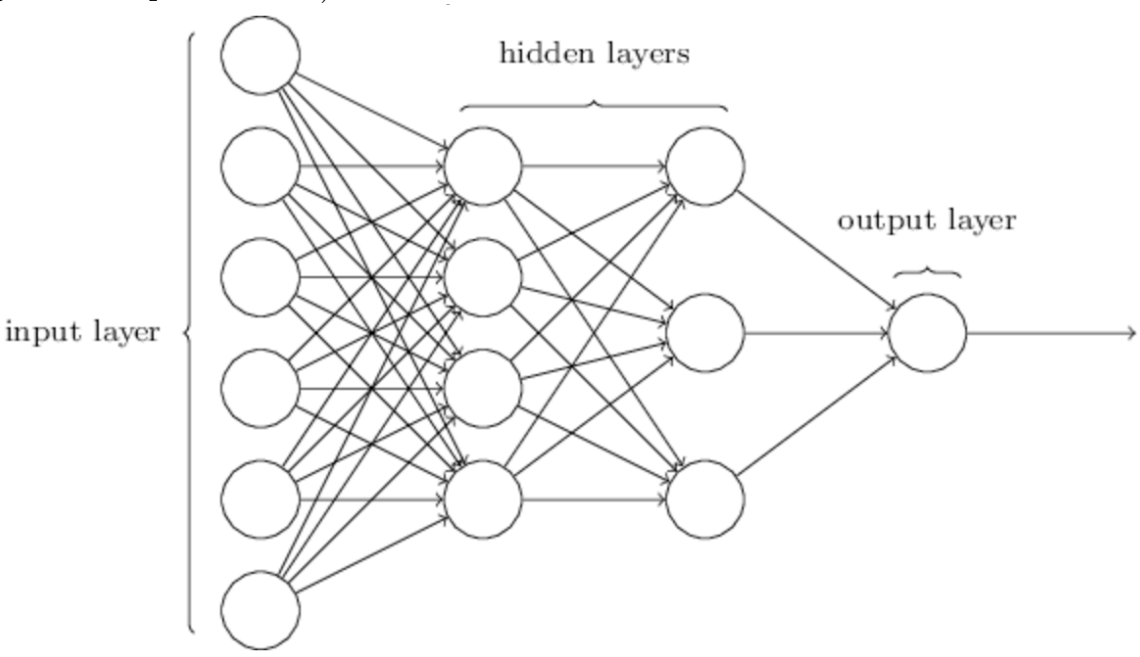

Fig. 1 Multi-hidden layer neural networks structure in [19]

The [2] implement a four-layer BPN to classify firms in stock market, with economic factors, long-term optimism, short-term optimism and other 6 kinds of data of firms as inputs and the running conditions of the firm as outputs, which could precisely label the firms as well running or poorly running firms. In [3], the authors structure a modular BPN system consists of 4 BPNs to forecast the trading time of a stock, they use 4 separate learning data sets for forecasting and each modular BPN takes one of them. To help BPN make better trading decisions, [20] implement a dynamic window system to check the result of BPN.

2.2 Time-Delayed NN and Recurrent NN

BPN is a widely applied NN algorithm in analyzing, however, it still has limitation. A major limitation of BPN is that it can learn only an input-output mapping of static (or spatial) patterns that are independent of time, to overcome this limitation, two methods applying the time property are proposed: using of time-delayed links; use recurrence links, [6]. Fig. 2 shows the common structure of time-delayed neural network (TDNN) and recurrent neural network (RNN).

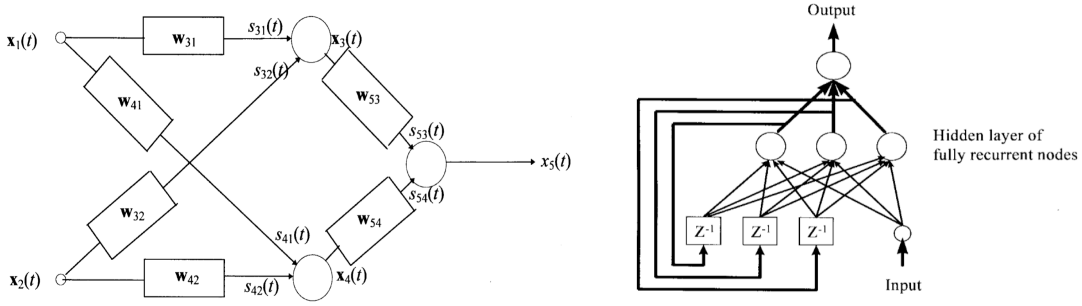

Fig. 2 Common struture of TDNN (left) and RNN (right) in [11]

The authors of [11] implement TDNN and compare its performance with other two NNs, giving a piece of detailed elaboration and evaluation. [6] uses the hybrid approaches, the derivations combine TDNN with genetic algorithm and Adaptive TDNN (ATNN) with genetic algorithm, to detect stock price temporal patterns, and eventually finding the GA-TDNN and GA-ATNN outperform than standard ATNN, TDNN and RNN.

[1] proposes RNN approach to recognizing stock price patterns and gets a superb performance, then the authors also give the evaluation measures to judge RNN outputs. It confirms the stock price patterns would repeat in time.

2.3 Self-Organizing Map and Rival Penalized Competitive Learning (RPCL)

Differing from other ANNs which are supervised learning process, self-organizing map (SOM) and rival penalized competitive learning (RPCL) NN are unsupervised learning. A supervised learning machine is ones whose every label results would be judged by correct answers during training period, if there are no correct answers that means it is unsupervised. SOM and RPCLNN are kinds of unsupervised learning machine which should do stock patterns discovery first before recognizing patterns. As for the structure of SOM and RPCLNN, both of them are two-layer networks, namely input layer and competitive layer, every different class would have special output in competitive layer naturally. Fig. 3 shows such structure. The difference between is SOM would modify the winner neuron to a better value, however, RPCLNN would also do such modify and additionally it would also modify other neurons to keep them even away form the

winner neuron.

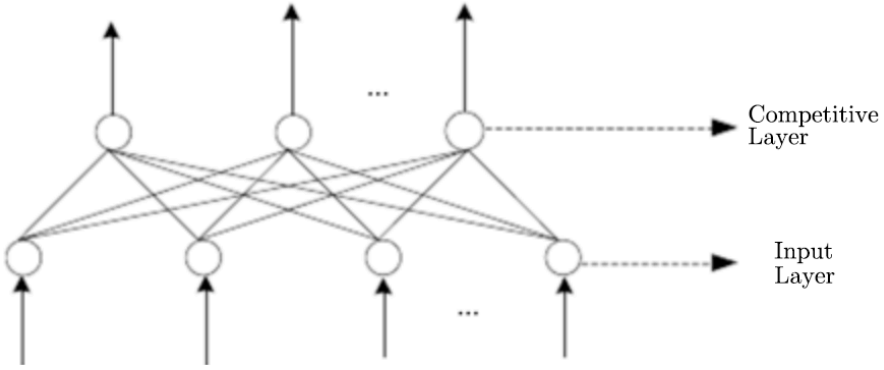

Fig. 3 Common structure of SOM and RPCLNN

[16] applies a SOM to discover 36 kinds of natural clustering chart patterns and recognize stock price patterns using this classification, and extra implement a system to release buy/sell signal based on stock patterns. In [15], authors do not use the result of competitive layer directly but connect competitive layer to a hidden-output-layer structured network, using another neural network to hybrid SOM and getting final result. Like [15], [14] also make a derivation of SOM to predict stock price, it hybrid SOM with a support vector machine, getting a better result.

In [18], the authors implement two networks, SOM and RPCLNN, to discover stock price patterns, and they find the performance of RPCLNN is better than SOM because RPCLNN could make the dissimilarity of classes bigger than that of SOM.

2.4 Other techniques (PNN, DNN)

Except the widely used neural networks have already been introduced, there still exist some other derivations.

[11] introduces a kind of NN, probabilistic neural network (PNN), which is an algorithm for approximating the Bayesian decision tree, and applying it to predict stock trend. After compared with other techniques, they find PNN extreme implementation simplicity and low false signal rate even the stock with low predictability, however, it just suits for stocks without long history for its high storage requirement.

In [7], authors proposed a derivation combine neural networks with dynamic-programming match system, to improve the ability of searching pattern in database.

3. METHODOLOGY

3.1 Inputs and Data Collection

[5] applies a neural network to forecast stock price, and the inputs it uses are interesting. In this paper, authors use 63 indicators to train NN, including fundamentals (volume, yield, price, etc.), technical values (MA, volume trends, etc.), gold/oil price, foreign exchange rates, interesting rates and economic statistics (import, export), and so on. However, finally they find many of them are useless, they still find around 30 useful inputs.

2.2 mentions a limitation of BPN, "it can learn only an input-output mapping of static (or spatial) patterns that are independent of time", and giving two solutions, RNN and TDNN. However, [9] gives another solution by generating new inputs to contain time information. It divides one pattern into n segments by $n-1$ points, and calculating the change rate of each point to contain the time span information.

In this project, I plan to select these initial data items:

1) real time data: the real time price, volume, time period (divide whole daily trading time in to segments per half-hours);
2) fundamental data: daily open, high, low, close, adjusted close and volume;
3) technical values: volume trend, short-term MA and long-term MA;
4) time information: step follow [9] to initially generate new features contain time information;
5) environment data: HSBC index, foreign exchange rates (HKD-USD, HKD-CNY exchange rates) and their trends, commodities price (gold, oil, coal) and their trends.

As for the source of these data, most of the raw daily data of 2) to 5) could be crawled from Yahoo finance and Bloomberg, 3) needs some simple calculation, 5) needs some further calculation and a separated neural networks would be implemented if need. The data in 1), could get from some charged application or free opened APIs from some financial organizations.

3.2 Techniques Selection

This project could be divided into four stages:

1) Preprocessing stage: applying a structure of modular RPCLNNs to discover the patterns of environment data items (into 9, 3x3, classes), and do other data preprocessing works, including value normalization, fault detection, technical values calculation, smooth price values by PIP;

2) Stock pattern discovery stage: applying RPCLNN to discover the stock price patterns (into 36, 6x6, classes), and then manually filter and label patterns as training data (the result of this stage could directly apply to stage 4);
3) Stock patterns recognize stage: applying BPN or RNN to recognizing stock patterns;
4) Trading signal release stage: applying decision tree to classify stock situation and release specific strategy buy/sell signals.

3.3 Evaluation

The evaluation of this project should include two parts:

1) Self evaluation: overall accuracy; precision, recall and f-measure for each class. The points to evaluate, evaluating the RPCLNN of discovering exchange rates and commodities price trends at the end of stage 1; evaluating the RPCLNN of discovering stock price patterns at the end of the stage 2; evaluating the NN of recognizing stock price patterns at the end of stage 3 (all of these evaluations need test data).
2) Compare evaluation: this NN system differs from others by the inputs, so should implement a standard BPN only use fundamental data as train data, and evaluating the result versus that of stage 3.

3.4 Tool

This project uses python as development language, and would use several python packages as tools:

1) Pandas: an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python.
2) Keras: a minimalist, highly modular neural networks library.

4. WORK SCHEDULE

4.1 Milestones

	Tasks	Completion time	Hours
1	Finish preprocessing module	15th Apr. 2016	100
2	Finish stock patterns discovery module	15th May 2016	100
3	Finish stock patterns recognition module	15th June 2016	100
4	Finish trading signals release module	15th July 2016	100
5	Evaluation each modules	1st Aug. 2016	60
6	Build website and make poster	15th Aug. 2016	40
7	Finish dissertation	15th Sept. 2016	100
			Total: 600

5. DELIVERABLES

At the end of project, the following items will be submitted:

	Item	Note
1	Detailed dissertation proposal	
2	Dissertation webpage	Before submit 2 and 3.
3	Intention to submit dissertation	
4	Interim report	
5	Initial presentation	
6	Poster	For exhibition usage.
7	Dissertation	

6. REFERENCES

[1] Kamijo, K. I., & Tanigawa, T. (1990, June). Stock price pattern recognition-a recurrent neural network approach. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on* (pp. 215-221). IEEE.

[2] Yoon, Y., & Swales, G. (1991, January). Predicting stock price performance: A neural network approach. In *System Sciences, 1991. Proceedings of the Twenty-Fourth Annual Hawaii International Conference on* (Vol. 4, pp. 156-162). IEEE.

[3] Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990, June). Stock market prediction system with modular neural networks. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on* (pp. 1-6). IEEE.

[4] Lawrence, R. (1997). Using neural networks to forecast stock market prices. *University of Manitoba*.

[5] VanEyden, R. J. (1996). *The application of neural networks in the forecasting of share prices*. Finance & Technology.

[6] Kim, H. J., & Shin, K. S. (2007). A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets. *Applied Soft Computing*, *7*(2), 569-576.

[7] Tanigawa, T., & Kamijo, K. I. (1992, June). Stock price pattern matching system-dynamic programming neural networks approach. In *Neural Networks, 1992. IJCNN., International Joint Conference on* (Vol. 2, pp. 465-471). IEEE.

[8] Tsang, P. M., Kwok, P., Choy, S. O., Kwan, R., Ng, S. C., Mak, J., ... & Wong, T. L. (2007). Design and implementation of NN5 for Hong Kong stock price forecasting. *Engineering Applications of Artificial Intelligence*, *20*(4), 453-461.

[9] Guo, X., Liang, X., & Li, X. (2007, August). A stock pattern recognition algorithm based on neural networks. In *Natural Computation, 2007. ICNC 2007. Third International Conference on* (Vol. 2, pp. 518-522). IEEE.

[10] Nygren, K. (2004). Stock prediction–a neural network approach. *Master's Thesis, Royal Institute of Technology, KTH, Stockholm*.

[11] Saad, E. W., Prokhorov, D. V., & Wunsch, D. C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *Neural Networks, IEEE Transactions on*, *9*(6), 1456-1470.

[12] Zhou, B., & Hu, J. (2009, December). A dynamic pattern recognition approach based on neural network for stock time-series. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* (pp. 1552-1555). IEEE.

[13] Fu, T. C., Chung, F. L., Ng, V., & Luk, R. (2001, August). Pattern discovery from stock time series using self-organizing maps. In *Workshop Notes of KDD2001 Workshop on Temporal Data Mining* (pp. 26-29).

[14] Hsu, S. H., Hsieh, J. P. A., Chih, T. C., & Hsu, K. C. (2009). A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. *Expert Systems with Applications*, *36*(4), 7947-7951.

[15] Afolabi, M. O., & Olude, O. (2007, January). Predicting stock prices using a hybrid Kohonen self organizing map (SOM). In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on* (pp. 48-48). IEEE.

[16] Chen, S. H., & He, H. (2004). Searching financial patterns with self-organizing maps. In *Computational intelligence in economics and finance* (pp. 203-216). Springer Berlin

Heidelberg.

[17] Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*,*78*(9), 1464-1480.

[18] Guo, X., Liang, X., & Li, N. (2007). Automatically recognizing stock patterns using RPCL neural networks.

[19] Nielsen, M. A. (2015). Neural Networks and Deep Learning. *URL: http://neuralnetworksanddeeplearning. com/. (visited: 01.11. 2014)*.

[20] Chang, P. C., Liu, C. H., Lin, J. L., Fan, C. Y., & Ng, C. S. (2009). A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications*, *36*(3), 6889-6898.