Stock Price Pattern Recognition
— A Recurrent Neural Network Approach —

Ken-ichi Kamijo and Tetsuji Tanigawa

C&C Information Technology Research Laboratories,
NEC Corporation

1-1, Miyazaki 4-Chome, Miyamae-ku, Kawasaki, Kanagawa 213, Japan

Abstract

This study was undertaken to apply recurrent neural networks to the recognition of stock price patterns, and to develop a new method for evaluating the networks. In stock tradings, *triangle* patterns indicate an important clue to the trend of future change in stock prices, but the patterns are not clearly defined by rule-based approaches. From stock price data for all *names* of corporations listed in The First Section of Tokyo Stock Exchange, an expert called *chart reader* extracted sixteen *triangles.* These patterns were divided into two groups, 15 training patterns and one test pattern. Using stock data during past 3 years for 16 names, 16 experiments for the recognition were carried out, where the groups were cyclically used. The experiments revealed that the given test *triangle* was accurately recognized in 15 out of 16 experiments, and that the number of the mismatching patterns was 1.06 per name on the average.

A new method was developed for evaluating recurrent networks with context transition performances, in particular, temporal transition performances. The method for the *triangle* sequences is applicable to decrease in mismatching patterns. By applying a cluster analysis to context vectors generated in the networks at recognition stage, a transition chart for context vector categorization was obtained for each stock price sequence. The finishing categories for the context vectors in the charts indicated that this method was effective in decreasing mismatching patterns.

1 Introduction

The purpose of this study is to propose a recurrent neural network model for stock price pattern recognition, and to develop a new method for evaluating the network. In stock trading with technical analysis[1], price patterns in Japanese-style stock charts[2], such as *triangles*, indicate an important clue to the trend of

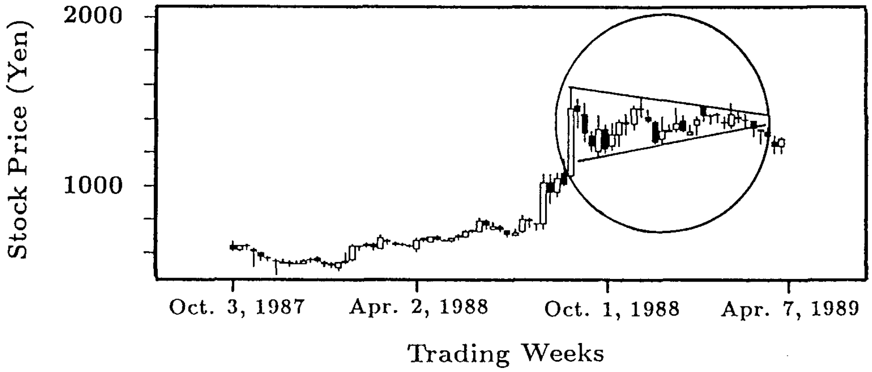


Figure 1: Candlesticks Chart and *Triangle* Pattern example

triangle pattern is enclosed by a large open circle in Fig.1. The candlestick is a symbol for describing opening, closing, high and low prices for a week at the same time. In case of a white (black) candlestick, the opening(closing) price is lower than the closing(opening), and the top and the bottom of the candlestick represent the closing(opening) and the opening(closing) prices, respectively. The top and the bottom of the line running through the candlestick depict high and low prices, respectively. Two oblique lines in the *triangle* pattern are called *resistance lines*. They are assistant lines which experts draw when judging whether a stock price pattern is a *triangle*.

The *triangle* refers to the beginning of a sudden stock price rise. Then, high and low prices appear mutually and the stock price oscillates. The resistance lines joining peaks and troughs converge. Since the resistance line is hand-drawn and the oscillation is vaguely defined, it is very difficult to formulate the *triangle* by means of a simple statistical model. Moreover, there is a difference in the degree of convergences among *triangle* patterns.

2.2 Data Collection

One expert extracted *triangles* by drawing resistance lines in candlesticks charts during the past three years for 1,152 names of joint stock corporations listed in The First Section of Tokyo Stock Exchange. The 16 extracted patterns were clearly judged to be *triangles* by the expert. Also, beginning and ending weeks for these patterns were assigned by the expert. In the selected patterns, there is a difference in name and starting week. A *triangle* period varies from 13 to 35 weeks. Because the resistance line was marked on the basis of high and low prices for the candlesticks, these prices were used as the input data to the author's proposed neural network model.

2.3 Normalization

In general, the stock price data have bias due to differences in name and time spans. Eliminating this bias requires stock price data normalization. To accomplish this, the authors used the variation in stock price average rate every week as a normalized value. The stock price average, obtained by exponential smoothing, was adopted. Let C_t be a closing price at the t-th week. Then, average A_t is given by

$$\begin{aligned} A_t &= s\,C_t + (1-s)A_{t-1} \\ &= s\,C_t + s(1-s)C_{t-1} + s(1-s)^2 C_{t-2} + \ldots, \end{aligned} \tag{1}$$

$$s = \frac{2}{13}, \tag{2}$$

where s is a constant well-known in the security market, and the denominator is the number of recursive calculations. Therefore, normalized value $\tilde{V}_t$ is given by

$$\tilde{V}_t = \frac{A_t - A_{t-1}}{A_{t-1}}. \tag{3}$$

Since both high and low prices are important information for deciding whether there is a *triangle* pattern or not, it is necessary to utilize the prices as input data. The authors used a dissociation from stock price

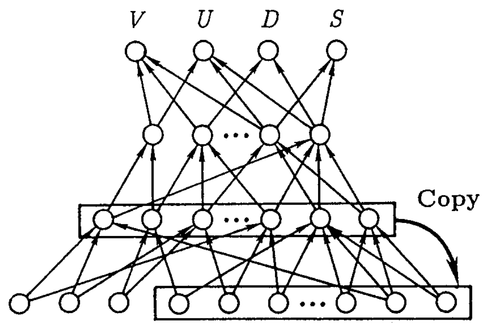


Figure 2: Network Architecture

average as a normalized value. Let H_t and L_t be a high price and a low price at time t, respectively. Then, dissociation from a stock price average for the high price, $\tilde{U}_t$,and that from a stock price average for the low price, $\tilde{D}_t$, are given by

$$\tilde{U}_t = \frac{A_t - H_t}{A_t}, \tag{4}$$

$$\tilde{D}_t = \frac{A_t - L_t}{A_t}, \tag{5}$$

respectively. Thus, a variation rate $\tilde{V}_t$ for stock price average, a dissociation $\tilde{U}_t$ for high price and a dissociation $\tilde{D}_t$ for low price were collected as normalized stock price data.

Although the frequency distribution of stock price data is roughly approximated by the normal distribution, there are different averages and variances for each normalized stock price data.

Since the sigmoid function is used as the output function for the input layer, all the input data is linearly transformed, so that the data are involved within the domain ranging from 0 to 1. For this purpose, the average μ and the variance σ^2 were calculated for each normalized value, from the data for 16 names during past three years, and all the data were transformed by means of the linear function, so that the interval $[\mu-\sigma, \mu+\sigma]$ corresponds to the interval $[0, 1]$. Moreover, V_t, U_t and D_t are obtained by linear transformations of $\tilde{V}_t$, $\tilde{U}_t$ and $\tilde{D}_t$, respectively. Consequently, about 68 percent of all the data was involved in the interval $[0, 1]$.

3 Neural Network Model

The present network model has a recurrent connection structure, similar to that proposed by Elman [3]. The structure does not rigidly constrain the length of input sequences and represents a finite state grammar implicitly[4]. For *triangle* pattern recognition, it is probable that variable oscillations were represented by the transition to a recursive state of internal grammar, and that nonlinear expansion and contraction were represented by both the transition to a sub-state and the jumping over states.

The network has a four-layer architecture, consisting of one input layer, two hidden layers and one output layer, as illustrated in Fig.2, for discriminating nonlinear patterns. Because of possibility that the teacher signal is beyond the interval $[0, 1]$, the output value for each unit in the output layer is given by the following linear function

$$f_1(x) = 0.4x, \quad (x \equiv \sum_{i=1}^{m} w_i y_i - \theta) \tag{6}$$

where y_i is the output value of the i-th unit in a previous layer, w_i is the weight on the connection from the i-th unit, θ is the threshold and m is the number of unit in the previous layer. An output function for other layers is the sigmoid function:

$$f_2(x) = \frac{1}{1 + e^{-x}}. \tag{7}$$

The input layer consists of two sets of units. The first set represents current stock data V, U, D. The second set of input units is called the *context* layer, and its units are used to represent the temporal context by holding a copy of the first hidden units' activity level at the previous time step. The output layer consists of prediction units for stock data and a *triangle* unit S which represents the *triangle* pattern termination. In this paper, the unit numbers for the first and the second hidden layers were set to 64 and 24, respectively.

4 Training

The proposed network, with the back propagation learning procedure[5], was trained to acquire features of the *triangles* retroactively, by using V_t, U_t, D_t as input data and V_{t-1}, U_{t-1}, D_{t-1} and S_{t-1} as teacher signals at each point in time t. All the initial values for the context layer are zero. The training data for the *triangle* unit S_t, which extracts the *triangle* pattern termination, is set to be 0.5 during a period ranging from the beginning of the *triangle* pattern to the appearance of the earliest peak. Otherwise, it is to be zero. Periods differ for *triangles* ranging from 1 to 4.

Sixteen stock price patterns were divided into two groups, fifteen training patterns and one test pattern. Sixteen experiments were carried out for the recognition, using these groups cyclically. In each experiment, the network was trained for fifteen *triangles* in random sequence. This training was iterated 2,000 times. After the iterations were complete, the error of the trained network was very small and its variation rate hardly changed.

5 Results

After training in each experiment, it was attempted to recognize the *triangle* for stock price data for 16 names during the past three years. The recognition started at the current week, then the network searched for *triangles* retroactively. When stock price data at any week is presented, the squared error was calculated between the prediction data and real data at a point in time one week past, transferring the values in the first hidden layer to the context layer. The error was retroactively accumulated. If the activation value of *triangle* unit S is beyond some threshold (in these experiments, 0.35) and the error values per unit and per week are

Table 1: Recognition Results

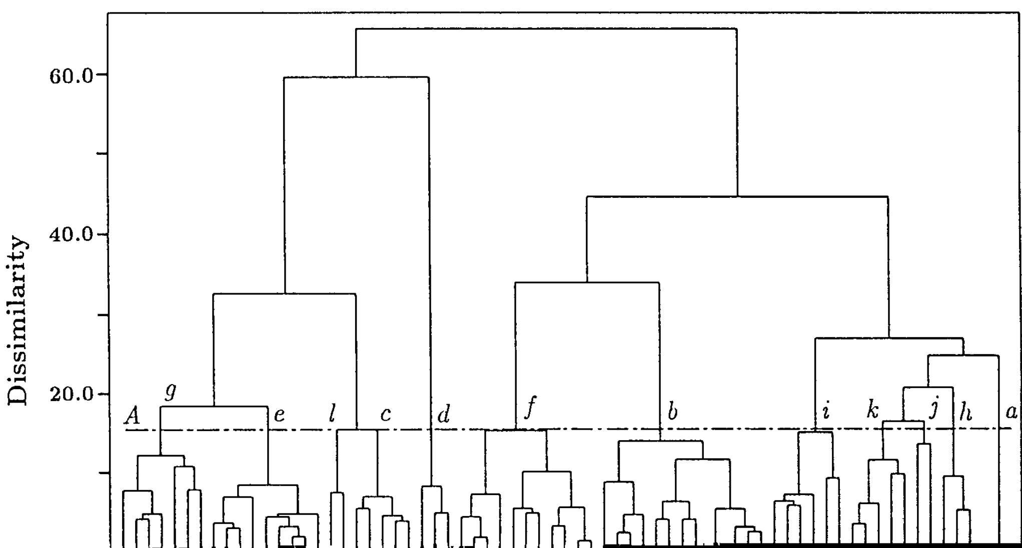
Experiment	*triangles* Training(%)	*triangles* Test	Mismatch* (total)
1	100	yes	1.06(17)
2	100	yes	1.31(21)
3	100	yes	0.69(11)
4	100	yes	1.88(30)
5	100	yes	0.75(12)
6	100	yes	0.88(14)
7	100	yes	1.44(23)
8	100	yes	0.69(11)
9	100	yes	0.88(14)
10	100	yes	1.06(17)
11	100	yes	1.00(16)
12	100	yes	0.63(10)
13	100	yes	1.88(30)
14	100	yes	0.63(10)
15	100	yes	0.63(10)
16	100	no	1.56(25)
Average	100 %	93.8 %	1.06(16.9)

*: pattern(s) per name

Each leaf in the dendrogram denotes a context vector to any input just presented in stock pattern sequences. Twelve clusters were made by cutting the tree at distance dash-dotted line A in Fig.3. Then, they are alphabetically named, that is, categories a to l. By describing a transition chart of cluster classification with the context vectors for each stock price sequence, it is possible to verify performance for recurrent networks, that is, the ability of the *triangle* recognition. Then, the temporal transition charts were made by using the network for Experiment 4.

Fig.4(a) illustrates a training *triangle* chart and its context transition chart for the *triangle*. The transition chart indicates that the context category is retroactively transited. The same charts as in Fig.4(a) for a test *triangle* and a mismatching pattern are shown in Figs.4(b) and 4(c), respectively. It became clear that the context category transited recursively, according to *triangle*'s oscillation, and that there are local category jumps at non-linear time-elasticity points. Moreover, in this case, the mismatching pattern recognition finished according to a different category from others.

To verify the ability of classifications, the finishing categories were checked for all *triangles* and all mismatching patterns in each experiment. Using these pattern's context vectors, a cluster analysis was carried out with 20 categories. The results concerning Experiments 1, 3, 4, 13 and 15 are shown in Table 2, and are similar to the results concerning the rest experiments. It is demonstrated that *triangles* are divided into a few major categories and some minor categories, and that the mismatching patterns are partially eliminated by checking the finishing categories. For example, in case of Experiment 13, training patterns were divided into two major categories and two minor ones. The mismatching patterns were divided into four categories identical to the training pattern categories(1st, 2nd, 3rd and 4th categories), and remaining three categories(5th, 6th and 7th categories). Therefore, the mismatching patterns in the remaining categories



[redacted]

Table 2: Number of Pattern in Finishing Categories

Experiment	Pattern Category	Finishing Category 1st	2nd	3rd	4th	5th	6th	7th
	training	5	4	2	2	1	1	-
1	test	1	-	-	-	-	-	-
	mismatch	3	5	-	1	-	8	-
	training	11	2	1	1	-	-	-
3	test	-	-	-	-	1	-	-
	mismatch	3	-	1	3	-	4	-
	training	10	2	1	1	1	-	-
4	test	1	-	-	-	-	-	-
	mismatch	3	5	6	12	-	1	3
	training	5	5	3	2	-	-	-
13	test	1	-	-	-	-	-	-
	mismatch	2	1	4	3	5	6	7
	training	9	3	1	1	1	-	-
15	test	1	-	-	-	-	-	-
	mismatch	4	3	-	1	1	1	-

nential smoothing and the dissociations from the average of high and low prices were utilized as normalized stock data. From these experiments, it was confirmed that the given test pattern was accurately recognized in 15 out of 16 experiments, and that the number of the mismatching patterns was 1.06 per name on the average in 16 experiments.

A new method for examining recurrent networks was established by searching for the temporal latent state transition. By applying a cluster analysis to context vectors generated in the networks at recognition stage, a transition chart for context categorization was obtained for each pattern sequence. It was found that the oscillations and the non-linear time-elasticity were represented by the recursive transition of context category and by the local jumps to other categories. Moreover, the finishing categories for context vectors in the charts are effective in *triangles* classification and partial elimination of mismatching patterns. It is hoped to develop a method for determining the *triangle* period and to refine the method for eliminating mismatching patterns.

Acknowledgment

The authors would like to thank Daiwa Securities Co. Ltd. for providing data related to stock and chart analysis. Meaningful discussions were held about applicability of neural networks to the securities domain, with Mr. M. Kinouchi, Mr. Y. Chiba, Daiwa Institute of Research Ltd., Mr. N. Kajihara and Mr. M. Asogawa, C&C System Research Laboratories, NEC Corporation.

References

[1] M. J. Pring, "Technical Analysis Explained", McGraw-Hill, New York(1985).

[2] S. Nison, "Learning Japanese-style 'Candlesticks' Charting", *Futures*, Vol.47, No.13, pp.46-47(1989).

[3] J. L. Elman, "Finding Structure in Time", Technical Report 8801, Center for Research in Language, University of California, San Diego(1988).

[4] D. Servan-Schreiber, A. Cleeremans and J. L. McClelland, "Encoding Sequential Structure in Simple Recurrent Networks", CMU-CS-88-183, Carnegie Mellon University(1988).

[5] D. E. Rumelhart, H. E. Hinton and R. J. Williams, "Learning Internal Representations by Error Propagation", in D. E. Rumelhart, J. L. McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Vol.I, The MIT Press, Cambridge(1986).

[6] H. C. Romesburg, "Cluster Analysis for Researchers", Lifetime Learning Publications, Belmont(1984).