An Application of the Associate Hopfield Network for Pattern Matching in Chart Analysis

Weiming Mai and Raymond S. T. Lee *

Abstract: Chart patterns are significant for financial market behavior analysis. Lots of approaches have been proposed to detect specific patterns in financial time series data, most of them can be categorized as distance-based or training-based. In this paper, we applied a trainable continuous Hopfield Neural Network for financial time series pattern matching. The Perceptually Important Points (PIP) segmentation method is used as the data preprocessing procedure to reduce the fluctuation. We conducted a synthetic data experiment on both high-level noisy data and low-level noisy data. The result shows that our proposed method outperforms the Template Based (TB) and Euclidean Distance (ED) and has an advantage over Dynamic Time Warping (DTW) in terms of the processing time. That indicates the Hopfield network has a potential advantage over other distance-based matching methods.

Keywords: pattern matching; chart analysis; learning associate hopfield network; perceptually important points; time series data mining

1. Introduction

Chart analysis is a kind of technical analysis in financial trading, which is different from quantitative analysis. The quantitative analysis intends to predict an exact future price by using machine learning models or deep learning models. Chart analysis aims to predict the trend of the future price according to the price pattern of the historical data. The traders and the financial analyst believe the theory that some specific patterns that have appeared before would appear again. Accordingly, these patterns could be signals for trading decisions. For example, the Head-and-Shoulder (H&S) pattern is believed to be one of the most reliable trend-reversal patterns. This pattern consists of three peaks, the first and third peaks are the shoulders and they represent the small rallies of the stock price. The second peak forms the head and it is the sign that the price would subsequently decline. A neckline can be drawn connecting the bottom of the two shoulders and the pattern can be confirmed normally when the closing price is clearly below this line. Lots of works have been proposed to analyze the relationship of the movement of the financial market and the shape of the chart pattern. Bulkowski [1] studies the character of the chart patterns and summarizes 53 applicable trading patterns. Wan et al. [2] divided the patterns into five categories in terms of their shape on the basis of Bulkowski’s work. In this paper, we adopt several classic patterns to generate synthetic data.

How to find the subsequences that match the query patterns as much as possible [3] has become an important problem in technical analysis. This question can be explained as given a fixed length of the financial time series data, find all the subsequences similar to the stored or expected pattern like H&S. Normally, pattern matching approaches in financial chart analysis can be categorized as template-based, rule-based, training based and distance-based. Most of them need to be segmented as a data preprocessing process and then the similarity between the processed data and the predefined template needs to
be evaluated. Once the similarity reaches or exceeds a given threshold by the analyst, the subsequence can be accepted as a specific tradable pattern.

The existing time series data mining research has been studied thoroughly in [4–6]. Fu et al. proposed a preprocessing method called perceptually important point (PIP) [3,7] to reduce the fluctuant data points and extract a given number of points to represent the subsequence of the time series data. Keogh et al. proposed piecewise aggregate (PAA) method [8] and piecewise linear approximation (PLA) method [9]. The PAA approach divides the time series data into \( N \) equal parts and calculates the mean value of each part to represent the subsequence. The PLA method adopts the sliding window to scan the subsequence in a top-down or bottom-up way to extract several straight lines to segment the data. In addition, Si et al. [10] proposed a segmentation method based on TPs (turning points). Wan et al. conducted several experiments [5] of different segmentation methods on the synthetic data and concluded that the PIP segmentation is robust in different similarity measurement and can preserve the overall shape of the subsequence.

Other than the segmentation methods mentioned above, Leigh and Martins et al. utilized a grid template method [11,12] to represent the 'Bull Flag' pattern. Goumatianos et al. proposed a grid template representation for pattern recognition in the forex market [13]. They introduced the template grid to capture the chart formation and defined a novel similarity measurement based on the template grid.

The similarity measurement is also critical to the matching result. In the work of Fu et al., they introduced a temporal distance (TD) [3] measurement to define the similarity of the segmented sequence and the predefined template. The rule-based (RB) method is also proposed in the same paper, which uses predefined rules to identify patterns. In [14], Zhang et al. designed a real-time pattern matching scheme based on Spearman’s rank correlation coefficient and rule sets. These two methods rely on the rules defined for each pattern thus have a disadvantage when we want to update or increase the query pattern, and it takes time to redefine the rules for the new patterns. The Euclidean distance (ED) method can also be used to calculate the similarity of two patterns and it does not need to be segmented, but from the previous experiments we can see that the ED approach has bad performance regard to some distorted sequences data and does not consider the horizontal and vertical shifts, so the dynamic time warping [15] algorithm (DTW) would be more useful in time-series data processing. However, time sequences are usually long, it would be time-consuming using the distance-based methods without segmentation.

Training-based methods consider the pattern matching process as a classic pattern recognition problem. Traditional classification models like support vector machine (SVM) and back-propagation neural network (BPNN) can be applied in time series data classification, the segmentation process is not necessary in these algorithms. Therefore, SVM and BPNN can preserve more information from the raw data [2]. However, this kind of method may have another drawback: These models always need an amount of training data to achieve a high testing accuracy, and it has to learn multiple classifiers for different patterns. Consequently, it is inefficient to apply in real-time financial pattern matching. In this context, we consider using the continuous Hopfield network as our matching approach.

The Hopfield neural network (HNN) was proposed by Hopfield John J [16] in 1982. The energy function was introduced to study the stability of the network, and it turns out that the HNN has a good memory association ability. The original HNN can only deal with the discrete binary pattern recognition by using Hebb’s rule [17] and its memory capacity is limited to the network size [18]. However, in recent years lots of works [19–26] have studied the memory capacity and invented different kinds of continuous HNN to deal with the continuous value pattern. We leverage HNN’s advantage in warping pattern recognition and the segmentation method in our work, proposing a training-based pattern matching approach, which only needs to be trained on the predefined template pattern.

With the work mentioned above, we treat the financial time series pattern matching problem as a classic pattern recognition problem. In the next section, we will review the related work in financial pattern matching. In Sections 3.1.1 and 3.1.2, we introduced details about
how to leverage the segmentation method and template grid into our matching approach. Section 3.2 presents the algorithm of the learning associate Hopfield network. The content includes the training process and matching procedure of our method. Section 4 describes the experimental data we use and the algorithm that generates the synthetic data. Section 4.2 summarizes the results of the experiments.

2. Related Work

Several current similarity-based pattern matching approaches are reviewed in this section. The TD approach in template-based pattern matching measures the point-to-point similarity between the predefined template and the segmented subsequences. The similarity can be described as a weighted combination of the amplitude distance (AD) and temporal distance (TD). The amplitude distance can capture vertical distortion and the temporal distance reflects the horizontal disparity. AD is defined as follows:

$$AD(SP, Q) = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (sp_k - q_k)^2}$$  \hspace{1cm} (1)

Here, SP and sp_k denote the extracted points with the PIP segmentation method. q_k is the point of the predefined template. TD is defined as follows:

$$TD(SP, Q) = \sqrt{\frac{1}{n-1} \sum_{k=2}^{n} (sp_k - q_k^t)^2}$$  \hspace{1cm} (2)

where sp_k^t and q_k^t denotes the coordinate of the point in time dimension. The similarity measure is in the following form:

$$D(SP, Q) = w_1 \times AD(SP, Q) + (1 - w_1) \times TD(SP, Q)$$  \hspace{1cm} (3)

usually, w_1 is set to be 0.5 in the experiment of [2,3,5], we follow this setting in our experiment. Furthermore, we set a threshold for the similarity measure: Once D(SP, Q) is lower than the preset threshold, the stored pattern with the minimum D would be accepted as the matching pattern.

The ED approach calculates the point-to-point distance between the query template and the sequences without segmentation. Let the predefined pattern be denoted as \(Y(y_1, \cdots, y_n)\) and the time series sequences be denoted as \(X(x_1, \cdots, x_n)\). We can easily get the similarity by the following:

$$ED(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$  \hspace{1cm} (4)

same as above, we set a threshold, once the sequence got a minimum ED(X, Y) that was lower than the threshold, it accepts the corresponding pattern.

DTW has been applied in time series pattern detection by Berndt et al. [15] in 1994. It has been widely used in speech recognition, gesture recognition and time series clustering since it was prevented. In time-series data processing, the length of the two sequences that need to be compared for similarity may not be equal. In such a case, the ED approach can not measure the similarity of two different size sequences of time series data efficiently. Figure 1A is the point-to-point computation of the ED method, Figure 1B is the demonstration of DTW, for each data point in the time series \(T\), it considers the distance between the point and the other all points in the sequence \(S\). DTW is based on the dynamic programming approach, given two sequences \(S(s_1, \cdots, s_n)\) and \(T(t_1, \cdots, t_m)\) we can form an \(n\)-by-\(m\) matrix \(\gamma\), the matrix elements \(d(s_i, t_j)\) represent the Euclidean distance of the points
s_i and t_i. The warping path \( W(w_1, \ldots, w_k) \) shown in Figure 2 maps the elements in \( S \) and \( T \). The dynamic time warping problem can be solved by minimizing the warping path.

\[
DTW(S, T) = \min \sqrt{\sum_{k=1}^{K} w_k} \quad (5)
\]

The dynamic programming formulation is based on the following recursive equation:

\[
\gamma(i, j) = d(s_i, t_j) + \min \{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \} \quad (6)
\]

where \( \gamma(i, j) \) denotes the cumulative distance of \( (i, j) \).

Figure 1. (A) Euclidean distance approach, (B) Dynamic time warping approach.

Figure 2. The warping path in DTW.

In the work of Kim et al. [27], the DTW algorithm is utilized for intraday price pattern matching. They constructed two sets of fixed patterns and used them as the matching templates. However, the computational complexity of DTW is \( O(m \times n) \), which is not efficient for large scale financial time series pattern recognition. In [28], Keogh et al. proposed a scaling up DTW for massive data processing. Their main idea is to reduce the data points by the piecewise linear representation (PLR) segmentation approach.
Training-based methods like SVM are applied in stock trading signal prediction in [29,30]. To train a SVM model, a set of time series labeled as positive and negative samples should be generated. Wan et al. also proposed a hidden semi-Markov model (HSMM) [2] for chart pattern matching. In this paper, we present a non-distance-based matching method by using the learning associate Hopfield network. This model could be trained with fewer samples and it costs less training time than the SVM and BPNN.

3. Materials and Methods

3.1. Pattern Representation

A suitable segmentation or representation of the time sequence is important for the matching results. In the next few subsections we introduce how to make use of the PIP and TG method in our pattern representation.

3.1.1. Perceptually Important Point

The well-known segmentation method PIP would be used in our matching approach. There are three variant ways to measure the distance between the adjacent points in PIP, which are the euclidean distance (PIP-ED), perpendicular distance (PIP-PD) and vertical distance (PIP-VD). The results of Fu et al. [3] illustrate that the PIP-VD would be the best choice in terms of efficiency and effectiveness. The PIP-VD can be calculated as follows:

$$VD(p_3, p_c) = |y_c - y_3| = \left| (y_1 + (y_2 - y_1) \frac{x_c - x_1}{x_2 - x_1}) \right|$$  \hspace{1cm} (7)

where $p_3(x_3, y_3)$ denotes the next chosen PIP, $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ are the exiting PIPs and $p_c(x_c, y_c)$ is the point in the line between $p_1$ and $p_2$. The schematic diagram is illustrated in Figure 3:

Figure 3. The above subgraph is the demonstration of the PIP-VD measure. A 80 days' normalized time series data (black line) is represented by 7 extracted PIPs (red points) using PIP-VD.

Algorithm 1 is the pseudocode for the selection procedure of the PIPs:
Algorithm 1 Perceptually important point identification with VD measure.

**Input:** sequence \( S(s_1, \ldots, s_m) \), predefined template \( P(p_1, \ldots, p_n) \);

**Output:** \( SP: \) PIPs with the length of \( n \);

1. Set \( SP_1 = s_1 \), \( SP_n = s_m \);
2. **repeat**
   3. Select point \( S_j \) with maximum VD distance to the adjacent points in \( SP \);
   4. Add \( S_j \) to \( SP \);
5. **until** \( SP \) is filled.
6. **return** \( SP \);

### 3.1.2. Template Grid

In Goumatinos’ paper [13], they introduced their novel template grid representation methodology: The time series data is encoded by the pattern identification code (PIC), a one-dimensional array that represents the position of the data points in a given template grid. An example is given to illustrate the PIC in Figure 4:

![Figure 4](image)

**Figure 4.** A time-series subsequence is represented by a 10-by-10 grid [13], which with the cell containing the data points of the sequence. The array \([0, 5, 5, 6, 8, 9, 8, 2, 4, 0]\) is the pattern identification code (PIC) of the time series sequence, which is the data points’ vertical positions of the line during the time.

After generating the PIC, each cell of the template grid would be assigned a weight by applying the following approach: the weights of each column are calculated as the equation \( w_{j,c} = 1 - |p - j|D \), where \( j = 1 \cdots M \), \( D = \frac{2M}{(M-p)(M-p+1)+(p-1)} \), \( p \) is the vertical position of the data point, \( M \) is the dimension of the template grid.

We constructed the predefined patterns like H&S, Double Top, Triple Top, Spike Top, etc. using the TG method. In order to generate the stored pattern that could be properly learned by the HNN, we present three different representation: (a) PIP-TG: To reduce the processing time, we firstly utilize PIP to process the data and then extract the data points to generate PIC, such that a simplified TG can be formed. (b) N-equal-part TG: Preset the dimension \( N \) of the template grid, the time sequence would be split evenly into \( N \) parts, each part represents the data point on the TG. For the predefined pattern, we increase the data points of the template to \( N \) and then apply the TG representation. (c) Scaling PIP: After reducing fluctuant points by PIP, simply scale up the number of the data points between the PIPs.

### 3.2. Learning Associative Hopfield Network

The Hopfield network is an important tool for memory retrieval. Traditional discrete HNN [16] can be defined by using Hebb’s rule. In 2011, Zheng et al. [20] firstly proposed the Learning Associative Hopfield Network, which can be applied in continuous-time real-world problems. They adopted the \( m \) energy function method to analyze the retrieval property of the CHNN and proposed sufficient conditions for local and global asymptotic
stability in [19]. Based on that theorem, the given patterns can be assigned as locally asymptotically stable equilibria of the learning associate Hopfield network (LAHN) by error back-propagation algorithm. The neural dynamics of LAHN can be described as following equations:

\[
\dot{x}(t) = A^{-1}(WF(x(t)) + \Theta) \quad (8)
\]
\[
\Delta x(t) = \dot{x}(t) - x(t) \quad (9)
\]
\[
x(t + 1) = x(t) + a\Delta x(t) \quad (10)
\]

where \(x\) denotes the initial neuron states, \(\Theta \in \mathbb{R}^n\) is a real constant vector. \(A \in \mathbb{R}^{n \times n}\) is a positive diagonal matrix, \(W \in \mathbb{R}^{n \times n}\) is the asymmetric weight matrix that can be learned from the error back-propagation algorithm. \(F(x)\) denotes a continuous activation function that is differentiable [20], \(a\) is the growing rate.

In [19], a sufficient condition is given for the local stability of the equilibria.

**Theorem 1.** \(X^{(i)}\) is locally asymptotically stable if \(\lambda_1[H - A\Phi(X^{(i)})] < 0\).

In Theorem 1, \(\Phi(X) = \text{diag}(f_1'(x_1), \cdots, f_n'(x_n))^{-1}\), \(H = W + W^T\), \(\lambda_1(\cdot)\) denotes the maximum eigenvalue of a matrix. In order to train the LAHN, we choose a proper differentiable activation function. The training datasets are the stored patterns, and the label of the stored pattern \((x_1, x_2, \cdots, x_m)\), \(x_i\) is a continuous activation function. The training process of LAHN would be described in Algorithm 2.

In this paper, we propose a pattern matching approach by LAHN that is tailored for our pattern representation method, for different representations we choose different activation functions and different initialization of \(A\) and \(W\). The growing rate \(a\) is also variant with different neuron sizes. The detailed parameter setting would be described in Section 4.2.

**Algorithm 2.** Training process of LAHN.

| Input: | the predefined patterns \(P = (p_1, \cdots, p_m)\), activation function \(f(x)\), diagonal matrix \(A\), the threshold of squared error \(T\), learning rate \(lr\); |
| Output: | the asymmetric weight \(W\) and bias \(\Theta\); |
| Initialize \(W\) and \(\Theta\), squared error (SE) = 0; |
| 2: while \(SE > T\) do |
| 3: for each stored pattern \(p_i\) in \(P\) do |
| 4: \(\text{out} \leftarrow WF(p_i) + \Theta;\) |
| 5: \(y_i \leftarrow Ap_i;\) |
| 6: Get the training loss of each training sample: \(\text{loss} \leftarrow (y_i - \text{out})^2;\) |
| 7: Compute the derivative of \(W\) and \(\Theta;\) |
| 8: \(W \leftarrow W - lr \times \frac{\partial \text{loss}}{\partial W}, \Theta \leftarrow \Theta - lr \times \frac{\partial \text{loss}}{\partial \Theta};\) |
| end for |
| 9: Calculate overall squared error; |
| end while |
| 10: return \(W, \Theta;\) |

3.3. Pattern Matching Based on Segmentation and LAHN

In this section, we will introduce our matching procedure in detail. Eleven basic patterns are chosen from 5 categories (Figure 5) that were summarized by Wan et al. [2]. We take these predefined patterns as stored patterns of the LAHN. As the number of the neuron is fixed and predefined, the basic chart patterns would be firstly processed by the PIP algorithm to extract a fixed number of the data points and form the stored pattern (i.e., the template patterns).
The basic template patterns would be represented by three representation methods described above, and then we design three different LAHNs accordingly. To illustrate our pattern matching process, we take Figure 6 for an example. The classic H&S chart pattern is represented in three different ways. As is shown in Figure 6, the PIP-TG method would extract a fixed number of data points and then TG is used to represent the pattern. According to Zheng’s paper, LAHN has the same characteristic as the traditional Hopfield network: the more neurons in the network, the higher the correct recall percentage. But it brings the higher computational cost at the same time. Therefore, we need to properly design the neuron number of each representation. The figure shows that there are 49 cells in the TG, so the neuron number of the LAHN would be 49. It is the same way for the N-equal-part TG. As for the Scaling-PIP, the neuron number would be the predefined total number of the data points after scaling and there would be 25 neurons in the case of Figure 6. In order to assign the stored pattern as the locally asymptotically stable equilibria, we fine-tune the parameters until the maximum eigenvalue of each $[H - A\Phi(X(i))]$ is lower than 0. We choose the tanh function as our activation function. The form is as follows: $f(x) = \frac{e^{kx} - e^{-kx}}{e^{kx} + e^{-kx}}$ and the exact value of the parameters would be described in Section 4.2.

Each stored pattern can be resized as a 1-dimensional vector, so we can easily get the training label by multiplying the matrix $A$. The training process is quite simple and efficient, after getting a satisfied squared error, a well-trained LAHN can be obtained. For the matching process, the incoming time series data would be segmented by one of the three representation methods as the initial state of the LAHN. The network dynamic evolves following the Equations (8)–(10). To avoid the oscillation, we constrain the growth rate into the range of $(0, 0.5]$. Unlike other distance-based methods, after completing the iteration, the initial warping pattern would asymptotically converge to the most similar stored template pattern. It saves time in calculating the similarity between the input pattern and template patterns. Figure 7 shows the network structure of the LAHN and Figure 8 is the flow chart for training a LAHN.

![Figure 5](image_url) Eleven template patterns, each pattern is represented by 7 PIPs. (H&S, Triangles ascending, Cup with handle, Reverse cup with handle, Triple bottoms, Double tops, Double bottoms, Spike top, Spike bottom, Flag, Wedges).
Figure 6. Three different representations. The value of the TG cell ranging from $[-1, 1]$ is shown as the gray-level image for visualization. There would be only one cell value that was equal to 1 in each column. Here, the dimension of the grid is 12. The scaling PIP is generated by the scaling procedure and it will be described in detail in Section 4.

Figure 7. The structure of LAHN resembles the traditional Hopfield network. The input $x(t)$ denotes the neuron state in time step $t$, the output of the network $\dot{x}(t)$ is the momentum of the current state, we can get the next state (i.e., the input of the network in time step $(t+1)$ by Equation (10), and the stable state could be achieved if we evolve the network recursively.
4. Experiment

4.1. Synthetic Data and Predefined Templates

Our experiment would be conducted on the synthetic dataset following the algorithm in Wan’s paper [5] to generate the synthetic data. It consists of three steps to generate the data, time scaling, time warping, and noise adding. The time scaling is used to enlarge the data of the template pattern, we produce different lengths of the scaling time series during this process. Time warping changes the position of the data points extracted by PIP, so the shape of the synthetic data would be different from the template, but the overall picture is similar. The noise adding process we use is different from the original method, a random value \( r_{nd} \) follows the Gaussian distribution \( \mathcal{N}(0, \sigma^2) \) would be added to each data point if the random value \( r \) is below the threshold (0.7 is used in this paper), we can easily control the noise level by adjusting the value of \( \sigma \). Algorithm 3 describes the process of generating the synthetic data. Later in Section 4.2, the results will show the performance of each model in different noise levels. The predefined template patterns would be shown in Figures A1–A3 in Appendix A.

The accuracy is used to compare the models, which is defined as Equation (11). As described in the last paragraph, each template pattern would be used to generate numbers of corrupted patterns. The correctly recalled patterns would be the final matching pattern.

\[
\text{Accuracy} = \frac{\text{correct recall patterns}}{\text{total patterns}}
\]  

(11)

Figure 8. The training process and pattern matching process of LAHN.
Algorithm 3 Generating synthetic data.

**Input:** A template pattern $p$, scaling number $m$, length of the template $q_{num}$;  
**Output:** Synthetic data of the template pattern;  

1. **Time Scaling** 
   Compute $X \leftarrow (m - q_{num})/(n - 1)$;  
2. for each point $x_i$ in the set $(x_2, \ldots, x_{end})$ do  
   $x_i \leftarrow x_{i-1} + (X + 1)$;  
end for  

3. **Time Warping** 
   for each data point $x_i$ do  
   Randomly change the position of $x_i$ between $x_{i+1}$ and $x_{i-1}$;  
end for  

4. **Noise Adding** 
   for each point $x_i$ do  
   Generate a random value $r$ that follows $U(0, 1)$;  
   if $r < $ threshold then  
   Generate a random value $rnd$ that follows $N(0, \sigma^2)$;  
   $x_i \leftarrow x_i + rnd$;  
end if  
end for  
return $x_i$;  

4.2. Results  
4.2.1. Six Stored Patterns  

Table 1 summarizes the differences of each matching method. The distance-based methods, ED, DTW and PIP-VD, require a predefined threshold to decide whether to accept the matching pattern. Segmentation means the methods require a data preprocessing procedure to extract the important data points. Training represents the methods that need to be trained.  

The proposed matching approach is compared with three different distance-based methods: ED, DTW and PIP-VD. We choose the first six templates as the stored patterns of the LAHN. We generate the synthetic data of the 11 patterns shown in Figure 9. Each template pattern would generate 200 samples. The length of each generated time-series data is the width of the sliding window and we temporally set it to be 49. Table 2 shows the accuracy of each model applied on the first 6 templates (H&S, Tria-A, CWH, Reverse CWH, Trip-B, Doub-T). We conduct the experiment on the normal data and noisy data, the standard deviation $\sigma$ in the noise adding process of the normal data is 0.15 while it is 0.2 in the noisy data. As for the time-warping process, we change the position of data point $x_i$ within the range of $\frac{2}{3}[x_{i-1}, x_{i+1}]$ in the normal dataset, and set the range to $[x_{i-1}, x_{i+1}]$ in the noisy dataset.

Table 1. The characteristics of each models.

|       | Segmentation | Training | Threshold |
|-------|--------------|----------|-----------|
| ED    | ×            | ×        | ✓         |
| DTW   | ×            | ×        | ✓         |
| PIP-VD| ✓            | ×        | ✓         |
| PIP-TG| ✓            | ✓        | ×         |
| N-equal-part TG | ✓        | ✓        | ×         |
| Scaling PIP | ✓        | ✓        | ×         |
Table 2. The overall accuracy of each model on the synthetic data.

| Model            | ED   | DTW  | PIP-VD | PIP-TG | N-Equal-Part TG | Scaling PIP |
|------------------|------|------|--------|--------|---------------|-------------|
| Noisy data       | 0.557| 0.919| 0.806  | 0.803  | 0.825         | 0.913       |
| Normal data      | 0.897| 0.998| 0.868  | 0.874  | 0.940         | 0.960       |

In the above experiment, the threshold of the ED approach is set to be 3, and DTW is 17. We scale the template pattern to the same size as the synthetic data in these two methods. Once the difference measured by the model is lower than the predefined threshold, the distorted time series would be matched to the most similar template pattern. The parameters of the LAHNs are summarized in Table A1. As is shown in Table 2, the ED approach is quite sensitive to the noise level of the data, while the accuracy of the other 5 methods is changed within an acceptable range. In our proposed method, the scaling PIP has the best overall matching accuracy, and the PIP-TG is slightly better than the traditional PIP. As for the training process, it only takes a few seconds to train a LAHN. DTW got the best performance over the other methods, but it requires more processing time as the data size increases.

The results in Figure 10 show that the consuming time of DTW grows with the scale of the time series. However, with the segmentation, the processing time of the other 4 methods (except ED) only has a small change. The N-equal-part method costs a bit more time because the LAHN has more neurons compared to the one used in the other representation.

Table 3 shows the matching results of each template pattern. The synthetic data is generated with three noise levels here, and the position of the data points would be changed within the range $[x_{i-1}, x_{i+1}]$ in the time-warping process. Although the PIP method has a good performance in some patterns such as H&S, Tria-A, and Trip-B, its overall accuracy is not the best. With more information on the time series, the DTW method got the best overall performance. The scaling PIP performs slightly worse than DTW, but as described above, it has an advantage in terms of processing time.
Figure 10. Processing time of each model.

Table 3. The accuracy of each template pattern with different standard deviation.

| Pattern     | ED   | DTW  | PIP  | PIP-TG | N-Equal-Part | Scaling PIP |
|-------------|------|------|------|--------|--------------|-------------|
| H&S         | 0.565| 0.960| 1.000| 1.000  | 0.550        | 0.990       |
| Tria-A      | 0.550| 0.955| 0.985| 0.945  | 0.785        | 0.925       |
| CWH         | 0.715| 0.995| 0.895| 0.895  | 0.935        | 0.915       |
| Reverse CWH | 0.725| 0.995| 0.810| 0.850  | 0.975        | 0.900       |
| Trip-B      | 0.245| 0.920| 0.995| 0.995  | 0.845        | 0.990       |
| Doub-T      | 0.760| 0.990| 0.610| 0.550  | 0.940        | 0.960       |
| Overall     | 0.593| 0.969| 0.883| 0.873  | 0.838        | 0.947       |

| Pattern     | ED   | DTW  | PIP  | PIP-TG | N-Equal-Part | Scaling PIP |
|-------------|------|------|------|--------|--------------|-------------|
| H&S         | 0.535| 0.940| 0.975| 0.975  | 0.555        | 0.890       |
| Tria-A      | 0.445| 0.930| 0.970| 0.930  | 0.630        | 0.875       |
| CWH         | 0.665| 0.955| 0.800| 0.755  | 0.905        | 0.915       |
| Reverse CWH | 0.590| 0.980| 0.690| 0.710  | 0.940        | 0.895       |
| Trip-B      | 0.225| 0.885| 0.975| 0.995  | 0.735        | 0.940       |
| Doub-T      | 0.660| 0.895| 0.515| 0.500  | 0.935        | 0.925       |
| Overall     | 0.520| 0.931| 0.821| 0.811  | 0.783        | 0.907       |

| Pattern     | ED   | DTW  | PIP  | PIP-TG | N-Equal-Part | Scaling PIP |
|-------------|------|------|------|--------|--------------|-------------|
| H&S         | 0.495| 0.845| 0.930| 0.930  | 0.630        | 0.785       |
| Tria-A      | 0.410| 0.855| 0.960| 0.905  | 0.715        | 0.795       |
| CWH         | 0.645| 0.915| 0.625| 0.685  | 0.865        | 0.835       |
| Reverse CWH | 0.665| 0.925| 0.540| 0.540  | 0.930        | 0.760       |
| Trip-B      | 0.165| 0.790| 0.955| 0.905  | 0.715        | 0.945       |
| Doub-T      | 0.715| 0.825| 0.530| 0.505  | 0.910        | 0.890       |
| Overall     | 0.516| 0.859| 0.757| 0.745  | 0.794        | 0.835       |

From Figure 11, we can see that the performance of DTW and scaling PIP barely changed with the noise level, and the accuracy of each pattern is very balanced. It indicates these two models would not have a preference on some specific pattern.
Figure 11. (a) The accuracy of each model with $\sigma = 0.15$. (b) The accuracy of each model with $\sigma = 0.2$. (c) The accuracy of each model with $\sigma = 0.25$.

4.2.2. Samples Analysis

To analyze the matching result more intuitively, we take several samples to illustrate the advantage of our approaches. Figure 12 shows the matching result of a distorted Trip-B pattern. Although the representation of the time series is blurred because of the Gaussian noise, the LAHN retrieves the original stored Trip-B template that most closely resembles the testing samples, while the ED method could not recognize this pattern because the distance-based similarity is higher than the predefined threshold.

Figure 12. The N-equal-part method, the middle subgraph is the input of the LAHN and the third subgraph is the matching template.

Figure 13 shows that Doub-T testing data is misidentified by the ED, DTW and PIP methods. The black line represents the synthetic data generated according to the Doub-T template. By enlarging the data points between the critical points of the PIP, the representation method can provide more information for the LAHN.

4.2.3. Eleven Stored Patterns

It has been verified that the memory capacity of the Hopfield network is confined to the neuron size of the network. In this section we enlarge the stored patterns of LAHN and explore how this could influence the matching performance of the LAHNs. The parameter and threshold setting is the same as the experiment of the six stored patterns, but the testing
data is a little bit different here. The standard deviation of the Gaussian noise in the noisy
data is 0.25, and the range of the time-warping process remains the same.

Figure 13. (a) The DTW algorithm mistakes the Doub-T testing sample (Black) as a Wedge pattern (Red). (b) The ED approach mistakes the Doub-T testing sample (Black) as a H&S pattern (Red). (c) The representation of PIP, (d) is the matching result of PIP, it mistakes the Doub-T testing sample (Black) as a H&S pattern (Red). (e,f) are the representation of scaling PIP and its matching result by LAHN, scaling PIP successfully recognizes the Doub-T pattern.

As can be seen from Figure 14 and Table 4, the overall performance of the Hopfield network based methods is somewhat declined, but the result of scaling PIP changes in an acceptable range. We can also observe that the N-equal-part method has a particularly low accuracy of the H&S pattern, that is mainly because the representation of H&S is quite similar to the Doub-T pattern. The improvement of this method would be later discussed in Section 5.

Figure 14. The overall accuracy on different synthetic data. The ED and DTW performs well when the noise level is low, but the accuracy declines a lot as the noise increases. The results of the scaling PIP are similar to the one in Figure 11. It barely changed with the noise level and that indicates that the scaling PIP is much more robust than the distance-based methods.
Table 4. The overall accuracy of each model on the synthetic data.

|                  | ED   | DTW  | PIP  | PIP-TG | N-Equal-Part | Scaling PIP |
|------------------|------|------|------|--------|---------------|-------------|
| **Normal data**  |      |      |      |        |               |             |
| H&S              | 0.845| 0.985| 0.955| 0.890  | 0.345         | 0.820       |
| Tria-A           | 0.835| 1.000| 0.975| 0.945  | 0.705         | 0.870       |
| CWH              | 0.965| 1.000| 0.580| 0.685  | 0.960         | 0.730       |
| Reverse CWH      | 0.960| 0.990| 0.605| 0.675  | 0.860         | 0.785       |
| Trip-B           | 0.550| 1.000| 0.985| 0.990  | 0.790         | 0.985       |
| Doub-T           | 0.935| 0.955| 0.510| 0.480  | 0.980         | 0.940       |
| Doub-B           | 0.965| 1.000| 0.520| 0.425  | 0.990         | 0.905       |
| Spike-T          | 0.870| 0.965| 0.775| 0.695  | 0.860         | 0.720       |
| Spike-B          | 0.885| 0.920| 0.760| 0.650  | 0.885         | 0.675       |
| Flag             | 0.990| 0.815| 0.905| 0.815  | 1.000         | 0.990       |
| Wedges           | 0.990| 0.995| 0.910| 0.515  | 0.925         | 0.665       |
| **Overall**      | 0.890| 0.966| 0.771| 0.706  | 0.845         | 0.826       |
| **Noisy data**   |      |      |      |        |               |             |
| H&S              | 0.465| 0.875| 0.465| 0.925  | 0.215         | 0.845       |
| Tria-A           | 0.410| 0.820| 0.410| 0.925  | 0.720         | 0.905       |
| CWH              | 0.640| 0.850| 0.640| 0.700  | 0.725         | 0.680       |
| Reverse CWH      | 0.635| 0.865| 0.635| 0.680  | 0.600         | 0.735       |
| Trip-B           | 0.170| 0.810| 0.170| 0.985  | 0.460         | 0.945       |
| Doub-T           | 0.585| 0.655| 0.585| 0.475  | 0.800         | 0.865       |
| Doub-B           | 0.620| 0.835| 0.620| 0.435  | 0.890         | 0.875       |
| Spike-T          | 0.605| 0.780| 0.605| 0.685  | 0.750         | 0.720       |
| Spike-B          | 0.535| 0.670| 0.535| 0.655  | 0.650         | 0.685       |
| Flag             | 0.855| 0.580| 0.855| 0.795  | 0.995         | 0.985       |
| Wedges           | 0.750| 0.835| 0.750| 0.300  | 0.820         | 0.675       |
| **Overall**      | 0.570| 0.780| 0.570| 0.687  | 0.693         | 0.810       |

5. Discussion

We proposed a lightweight pattern matching method by utilizing the learning associative Hopfield network, which is a non-distance-based approach that combines the segmented representation method. By using the synthetic data generated from the 11 traditional trading chart templates, we conduct experiments with different levels of noise and distortion. We discover that the scaling PIP performs better than the ED and the traditional PIP methods. Its matching results are slightly worse than DTW but cost less time during the matching process. Furthermore, when the number of stored patterns does not exceed the memory capacity of the LAHN, the N-equal-part method performs steadily. It can be concluded from the experimental results that the scaling PIP is a robust and efficient matching method and the Hopfield network has a potential advantage over other distance-based matching methods.

The proposed Hopfield network based algorithm can be applied in some pattern matching trading systems to detect the patterns in the daily or hourly financial market. With fewer training datasets and less processing time the trader can efficiently capture the specific signals and make the transaction. Future work could consider how to leverage the characteristics of DTW and the methods we proposed to construct a more accurate and scaling matching method. Moreover, the memory capacity limits the performance of the Hopfield network based pattern matching. Future studies can also explore the retrieval reliability of the LAHN and different segmentation algorithms like PLA, PAA or TP (turning points), which could distinguish different stored templates.

**Author Contributions:** Conceptualization, W.M. and R.S.T.L.; Methodology, W.M. and R.S.T.L.; Formal analysis, W.M.; Visualization, W.M.; Writing—original draft, W.M.; Writing—review and editing, R.S.T.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.
Acknowledgments: This paper was supported by Beijing Normal University-Hong Kong Baptist University United International College (UIC) research grant R202008 and R201948.

Conflicts of Interest: The authors declare no conflict of interest.

The following abbreviations are used in this manuscript:
LAHN   Learning associate Hopfield network
DTW    Dynamic time warping
ED     Euclidean distance
PIP-VD Perceptually important point using vertical distance

Appendix A

Table A1. Parameters setting of the LAHN. $A$ denotes the diagonal elements of the matrix $A$, for simplicity, the elements of $A$ are all equal. $k$ denotes the parameter of the activation function. The first column is the number of neurons and the last column is the learning rate.

| Neurons | $k$  | $A$  | $lr$ |
|---------|------|------|------|
| LAHN$_1$ | 49   | 6.0  | 5    | 0.01 |
| LAHN$_2$ | 144  | 2.5  | 10   | 0.01 |
| LAHN$_3$ | 25   | 5.9  | 1.5  | 0.01 |

Figure A1. The representation of the PIP-TG.
Figure A2. The representation of the N-equal-part TG.

Figure A3. The representation of the Scaling PIP.

References
1. Bulkowski, T.N. *Encyclopedia of Chart Patterns*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 225.
2. Wan, Y.; Si, Y.W. A hidden semi-Markov model for chart pattern matching in financial time series. *Soft Comput.* 2018, 22, 6525–6544. [CrossRef]
3. Fu, T.C.; Chung, F.L.; Luk, R.; Ng, C.M. Stock time series pattern matching: Template-based vs. rule-based approaches. *Eng. Appl. Artif. Intell.* 2007, 20, 347–364. [CrossRef]
4. Fu, T.C. A review on time series data mining. *Eng. Appl. Artif. Intell.* 2011, 24, 164–181. [CrossRef]
5. Wan, Y.; Gong, X.; Si, Y.W. Effect of segmentation on financial time series pattern matching. *Appl. Soft Comput.* 2016, 38, 346–359. [CrossRef]
6. Ali, M.; Alqahtani, A.; Jones, M.W.; Xie, X. Clustering and Classification for Time Series Data in Visual Analytics: A Survey. *IEEE Access* 2019, 7, 181314–181338. [CrossRef]

7. Fu, T.C.; Chung, F.L.; Lu, R.; Ng, C.M. Representing financial time series based on data point importance. *Eng. Appl. Artif. Intell.* 2008, 21, 277–300. [CrossRef]

8. Keogh, E.; Chakrabarti, K.; Pazzani, M.; Mehrotra, S. Dimensionality reduction for fast similarity search in large time series databases. *Knowl. Inf. Syst.* 2001, 3, 263–286. [CrossRef]

9. Keogh, E.; Chu, S.; Hart, D.; Pazzani, M. An online algorithm for segmenting time series. In Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 29 November–2 December 2001; pp. 289–296.

10. Si, Y.W.; Yin, J. OBST-based segmentation approach to financial time series. *Eng. Appl. Artif. Intell.* 2013, 26, 2581–2596. [CrossRef]

11. Leigh, W.; Modani, N.; Purvis, R.; Roberts, T. Stock market trading rule discovery using technical charting heuristics. *Expert Syst. Appl.* 2002, 23, 155–159. [CrossRef]

12. Martins, T.M.; Neves, R.F. Applying genetic algorithms with speciation for optimization of grid template pattern detection in financial markets. *Expert Syst. Appl.* 2020, 147, 113191. [CrossRef]

13. Goumatianos, N.; Christou, I.T.; Lindgren, P.; Prasad, R. An algorithmic framework for frequent intraday pattern recognition and exploitation in forex market. *Knowl. Inf. Syst.* 2017, 53, 767–804. [CrossRef]

14. Zhang, Z.; Jiang, J.; Liu, X.; Lau, R.; Wang, H.; Zhang, R. A real time hybrid pattern matching scheme for stock time series. In Proceedings of the Twenty-First Australasian Conference on Database Technologies, Brisbane, Australia, 18–22 January 2010; Volume 104, pp. 161–170.

15. Berndt, D.J.; Clifford, J. *Using Dynamic Time Warping to Find Patterns in Time Series*; KDD Workshop: Seattle, WA, USA, 1994; Volume 10, pp. 359–370.

16. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* 1982, 79, 2554–2558. [CrossRef] [PubMed]

17. Morris, R.G.M. Do Hebb: The Organization of Behavior, Wiley: New York; 1949. *Brain Res. Bull.* 1999, 50, 437. [CrossRef]

18. Baram, Y. Orthogonal Patterns in A Binary Neural Network. *Appl. Opt.* 1991, 30, 1772–1776.

19. Zheng, P.; Tang, W.; Zhang, J. Efficient Continuous-Time Asymmetric Hopfield Networks for Memory Retrieval. *Neural Comput.* 2010, 22, 1597–1614. [CrossRef] [PubMed]

20. Zheng, P.; Zhang, J.; Tang, W. Learning Associative Memories by Error Backpropagation. *IEEE Trans. Neural Netw.* 2011, 22, 347–355. [CrossRef] [PubMed]

21. Hernández-Solano, Y.; Atencia, M.; Joya, G.; Sandoval, F. A discrete gradient method to enhance the numerical behaviour of Hopfield networks. *Neurocomputing* 2015, 164, 45–55. [CrossRef]

22. Viola, F.; Marco, L.; Giancarlo, R. On the Maximum Storage Capacity of the Hopfield Model. *Front. Comput. Neurosci.* 2016, 10, 144.

23. Cabrera, E.; Sossa, H. Generating exponentially stable states for a Hopfield Neural Network. *Neurocomputing* 2018, 275, 358–365. [CrossRef]

24. Demircigil, M.; Heusel, J.; Löwe, M. On a Model of Associative Memory with Huge Storage Capacity. *J. Stat. Phys.* 2017, 168, 288–299. [CrossRef]

25. Do-Hyun, K.; Jinha, P.; Byungnam, K.; Constantine, D. Enhanced storage capacity with errors in scale-free Hopfield neural networks: An analytical study. *PLoS ONE* 2017, 12, e0184683.

26. Kobayashi, M. O(2)-Valued Hopfield Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* 2019, 30, 3833–3838. [CrossRef]

27. Kim, S.H.; Lee, H.S.; Ko, H.J.; Jeong, S.H.; Byun, H.W.; Oh, K.J. Pattern matching trading system based on the dynamic time warping algorithm. *Sustainability* 2018, 10, 4641. [CrossRef]

28. Keogh, E.J.; Pazzani, M.J. Scaling up Dynamic Time Warping to Massive Datasets. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Prague, Czech Republic, 15–18 September 1999.

29. Tang, H.; Dong, P.; Shi, Y. A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points. *Appl. Soft Comput.* 2019, 78, 685–696. [CrossRef]

30. Luo, L.; You, S.; Xu, Y.; Peng, H. Improving the integration of piece wise linear representation and weighted support vector machine for stock trading signal prediction. *Appl. Soft Comput.* 2017, 56, 199–216. [CrossRef]