A Neural Network Filter For Complex Spatio-Temporal Patterns*

Jinwen Ma[1], Dezhi Huang[2]
1. Dept of Information Science, School of Mathematical Sciences and National Laboratory on Machine Perception, Peking University, Beijing, 100871, China
2. Dept of Computer Science and Technology, Tsinghua University, Beijing, 100084, China

Abstract—This paper proposes a four-layer neural network filter for complex spatio-temporal patterns (or sequences). For any given complex spatio-temporal pattern, it can be constructed according to the order of the spatio-temporal pattern. Moreover, it is demonstrated by the simulation results.

I. INTRODUCTION

Filtering and recognition of spatio-temporal patterns (or sequences) is very important in the applications for both signal processing and pattern recognition. With the development of neural network, there have been many approaches using neural network models and training algorithms to process spatio-temporal patterns, e.g. [1]-[7]. However, the main task of them is to learn and retrieve spatio-temporal patterns from some initial information, e.g., small parts of spatio-temporal patterns. This is generally different from the task of filtering and recognition of spatio-temporal patterns. As a matter of fact, there are only a few neural network approaches which can be applied to filtering and recognition of spatio-temporal patterns directly.

One important neural network approach meeting this demand is the avalanche matched filter(AMF) which was proposed by Hecht-Nielsen[3] as a generalization of Grossberg's outstar avalanche[1]. By a training scheme, the AMF can learn a spatio-temporal pattern and recognize it in a noise environment in the sense of nearest matching. However, in using the AMF there is a major disadvantage that the spatio-temporal pattern cannot be recognized if its phase is changed. Another neural network approach is to convert the spatio-temporal signal into a spatial signal by ignoring the temporal component and treating the entire signal as a spatial pattern. Then, the filtering and recognition of these spatial patterns can be realized by conventional neural networks such as back-propagation[8], radial basis functions[9], and time-delayed neural network[10]. But this neural network approach has a disadvantage that the training time is long when solving large scale problems. Recently, a topological and temporal correlator network is proposed to solve the problem of spatio-temporal pattern learning, recognition, and retrieval[7]. It is a synthetical network based on a Kohonen's self-organizing map and a fuzzy ART network. However, the number of the processing neurons in some layer varies with the complexity of spatio-temporal patterns, which makes difficult to implement the network.

In this paper, we propose a neural network based filter for any given complex spatio-temporal pattern under a general noise environment. It is a four-layered forward neural network designed from the order of the spatio-temporal pattern and the pattern itself. Moreover, we substantiate the filter by some simulation results.

II. THE ORDER OF A SPATIO-TEMPORAL PATTERN

The order of a patio-temporal pattern is a key index to represent its complexity. In studying how to learn and generate spatio-temporal patterns, the order of a spatio-temporal pattern has been already introduced in the literature, e.g., [4]-[6]. Here, in preparation for our neural network approach, we make a mathematical study on the order of a spatio-temporal pattern. Since the purpose of our approach is to filter and recognize spatio-temporal patterns, the following definition of the order of a spatio-temporal pattern is slightly different from the old ones. We now begin with the definition of a spatio-temporal pattern.

Definition 1. *A spatio-temporal pattern $\mathcal{S}$ is defined as*

$$\mathcal{S} = P_1 P_2 \cdots P_m,$$

where m is a positive integer or infinity, and $P_i = [p_{i1}, p_{i2}, \cdots, p_{in}]^T \in E \subset R^n$ for $i = 1, \cdots, m$. If $P_i \in \{-1,1\}^n$ ($\{0,1\}^n$), $\mathcal{S}$ is called a bipolar(binary) spatio-temporal pattern.

Generally, m is called the length of the spatio-temporal pattern and these P_i are called the spatial pattern of $\mathcal{S}$. When $\mathcal{S}$ is periodic, i.e., P_i is periodic, it is further called a spatio-temporal cycle. If its minimum cycle is $\{P_1, \cdots, P_{m_0}\}$, we further represent it by

*This work was supported by the Natural Science Foundation of China for Projects 19701022, 60071004 and the Natural Science Foundation of Guangdong Province for Project 970377.

$S^c = P_1, \cdots, P_{m_0} P_1$. And m_0 is called the length of the spatio-temporal cycle.

Definition 2. *When a spatio-temporal pattern $\mathcal{S}$ is not periodic, the order of $\mathcal{S}$ is defined by*

$$\begin{aligned} r(\mathcal{S}) &= min\{k : P_i P_{i+1} \cdots P_{i+k-1} \neq P_j P_{j+1} \cdots P_{j+k-1} \\ &\quad for\ \ all\ \ i,j \leq m-k+1, and\ \ i \neq j.\} \end{aligned} \tag{1}$$

Clearly, the order of $\mathcal{S}$ is the minimum of the number k which enable all the possible k-step blocks in $\mathcal{S}$ are different. Surely, it is a positive integer in the range $[1, m]$. For clarity, $r(\mathcal{S})$-step blocks of $\mathcal{S}$ are called basic blocks of $\mathcal{S}$. When the order of $\mathcal{S}$ is just one, it is called a simple spatio-temporal pattern. In this case, all the spatial patterns are different. When the order of $\mathcal{S}$ is larger than one, it is called a complex spatio-temporal pattern. Obviously, some spatial patterns in a complex spatio-temporal pattern must be repeated.

Similarly, we can define the order of a spatio-temporal cycle as follows.

Definition 3. *For a spatio-temporal cycle $\mathcal{S}^c = P_1, \cdots, P_m P_1$, its order is defined by*

$$\begin{aligned} r(\mathcal{S}^c) &= min\{k : P_i P_{i+1} \cdots P_{i+k-1} \neq P_j P_{j+1} \cdots P_{j+k-1} \\ &\quad for\ \ all\ \ i,j \leq m-1, and\ \ i \neq j.\} \end{aligned} \tag{2}$$

That is, the order of a spatio-temporal cycle is the minimum of the number k which enable all the first m sequential k-step blocks in the spatio-temporal cycle are different. Similarly, $r(\mathcal{S})$-step blocks of $\mathcal{S}^c$ are called basic blocks of $\mathcal{S}^c$. When the order of $\mathcal{S}^c$ is just one, it is called a simple spatio-temporal pattern. That is, all the spatial patterns in the minimum cycle of a simple spatio-temporal cycle are different, i.e., $P_1, \cdots, P_m$ are different. Otherwise, when the order of $\mathcal{S}^c$ is larger than one, it is called a complex spatio-temporal cycle. In some cases, e.g., for a simple spatio-temporal cycle, the order of $\mathcal{S}^c$ is equal to that of the spatio-temporal pattern $\mathcal{S}_b = P_1, \cdots, P_m$. However, they are not equal in the general case. A simple example is $\mathcal{S}^c = 112211$. It can be easily found that $r(\mathcal{S}^c) = 3$, but $r(\mathcal{S}_b) = r(11221) = 2$.

We further study the relation between a spatio-temporal pattern and its basic blocks. Actually, when a spatio-temporal pattern $\mathcal{S}$ is given, we certainly have the set of its basic blocks. For convenience, this set is called the basic block set and denoted by $\mathcal{B}_\mathcal{S}$. Certainly, if $\mathcal{B}_\mathcal{S}$ only corresponds to the true spatio-temporal sequence $\mathcal{S}$, that is, $\mathcal{B}_\mathcal{S}$ corresponds to a unique spatio-temporal sequence, we can recognize $\mathcal{S}$ by equivalently recognizing its basic blocks in $\mathcal{B}_\mathcal{S}$. Clearly, this is not true when $\mathcal{S}$ is a simple spatio-temporal pattern or cycle. However, this is always true when $\mathcal{S}$ is a complex spatio-temporal pattern or cycle, which can be proved by the following theorem.

Theorem 1. *Suppose that $\mathcal{S}_1$ and $\mathcal{S}_2$ are both complex spatio-temporal patterns or cycles. If $\mathcal{S}_1$ and $\mathcal{S}_2$ are different, i.e., they differ at either length or some component position(s), $\mathcal{B}_{\mathcal{S}_1} \neq \mathcal{B}_{\mathcal{S}_2}$.*

The proof will be given in [11].

According to Theorem 1, we certainly have that the map from $\mathcal{S}$ to $\mathcal{B}_\mathcal{S}$ is one to one in the cases of both complex spatio-temporal pattern and cycle. That is, $\mathcal{B}_\mathcal{S}$ uniquely corresponds to its true spatio-temporal pattern or cycle when $\mathcal{S}$ is complex. Thus, a complex spatio-temporal pattern or cycle can be recognized from its basic blocks. We will use this idea to design the neural network spatio-temporal filter.

III. THE NEURAL NETWORK SPATIO-TEMPORAL FILTER

Suppose that $\mathcal{S}$ is a given complex spatio-temporal pattern of finite length (i.e., m is finite) or a given complex spatio-temporal cycle and its order is $k (> 1)$. Moreover, we let M be the number of different spatial patterns in $\mathcal{S}$, and let N be the number of the basic blocks of $\mathcal{S}$, i.e., $N = |\mathcal{B}_\mathcal{S}|$. We now propose our neural network spatio-temporal filter for $\mathcal{S}$.

As sketched in Figure 1, it is a four-layer forward neural network. The first or input layer consists of n input neurons corresponding to the dimension of the spatial patterns in $\mathcal{S}$. These n input neurons only receive and transmit the input signals to the second layer at each time.

The second layer consists of M neurons $U_1, U_2, \cdots, U_M$, which correspond to the M different spatial patterns $SP_1, \cdots, SP_M$, respectively, in $\mathcal{S}$. Each neuron in the second layer represents an individual spatial pattern and its aim is to recognize this spatial pattern from the input pattern in a noise environment. That is, it should be such a binary neuron that when the input pattern is similar to the spatial pattern, the neuron is activated (the state is 1); otherwise, the neuron is inhibited (the state is 0). Let $SP_i = [sp_{i1}, \cdots, sp_{in}]^T$, we can design the corresponding neuron U_i in the two cases of binary and real input patterns as follows.

(i). The Binary Input Case.

We first introduce the perceptive neuron which has been defined and constructed in [12]. As well-known, a binary or MP neuron is a processing element with n input signals $x_1, x_2, \cdots, x_n$ and an output signal y. There is a weight w_i on the connection from each input signal x_i to the neuron. And there is also a threshold value θ for the neuron. For an input signal pattern $X = [x_1, x_2, \cdots, x_n]^T$, the output signal y of the neuron is computed by

$$y = Sgn(H(x)) = \begin{cases} 1 & \text{if } H(x) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where

$$H(x) = \sum_{i=1}^{n} w_i x_i - \theta.$$

Let $X = [x_1, x_2, \cdots, x_n]^T$ be a n-dim binary input pattern, i.e., $X \in \{0,1\}^n$, and $C = [c_1, c_2, \cdots, c_n]^T$ be a fixed n-dim binary pattern, we define

$$d_H(X,C) = \sum_{i=1}^{n} |x_i - c_i| \tag{4}$$

as the Hamming distance between X and C. We then define the t-neighborhood of C over the n-dim binary space $\{0,1\}^n$ as follows:

$$R_t(C) = \{X : d_H(X,C) \le t\}. \tag{5}$$

The definition of the perceptive neuron is given as follows.

Definition 1 *If a binary neuron with a fixed weight vector $W = [w_1, w_2, \cdots, w_n]^T$ and a fixed threshold value θ, satisfies the following input-output relation:*

$$y(X) = Sgn(\sum_{i=1}^{n} w_i x_i - \theta) = \begin{cases} 1 & \text{if } X \in R_t(C) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

it is called a t-neighborhood perceptive neuron of (pattern) C.

For a binary neuron, we can consider that it perceives the pattern when its output is one, and it does not perceive the pattern when its output is zero. From Definition 1, the perceptive neuron perceives or recognizes a unique pattern under a noisy environment. The perceptive field of a t-neighborhood perceptive neuron of C is just the t-neighborhood of C. Actually, the perceptive neuron can be easily constructed according to the following theorem[12].

Theorem 2. *Suppose that $C = [c_1, c_2, \cdots, c_n]^T$ is a constant binary pattern, and that $d_H(C) = \sum_{i=1}^{n} c_i$ is the Hamming weight of C. If a neuron is constructed by*

$$w_i = (-1)^{1+c_i}, \quad i = 1, \cdots, n \tag{7}$$
$$\theta = d_H(C) - (t+1) \tag{8}$$

then it is a t-neighborhood perceptive neuron of C.

By the perceptive neuron, we can easily define the neuron U_i for the spatial pattern SP_i in the case of binary input signals. Certainly, SP_i is also a binary pattern in this case. In fact, we can simply let it be a t_i-neighborhood perceptive neuron of (pattern) SP_i. t_i is selected according to the noise environment.

When the input signals and the spatial patterns are bipolar, we can still construct the perceptive neuron by Eq.(7,8) under the transformation $v = 2u - 1 (u \in \{0,1\}, v \in \{-1,1\})$ and have the similar results.

(ii). The Real Input Case.

In the case of real input signals, each spatial pattern SP_i is a real pattern in R^n. When the input pattern X and the spatial pattern SP_i are normalized, i.e., $\| X \| = \| SP_i \| = 1$, we can design the neuron N_i by a binary neuron where $W = SP_i$ and $\theta = 1 - \varepsilon$ where $\varepsilon \ge 0$. That is, when $X^T SP_i = \sum_{j=1}^{n} x_j sp_{ij}$ is near one, i.e., larger than $1 - \varepsilon$, the neuron is activated and we consider the input pattern is recognized as SP_i; otherwise, the neuron is inhibited and we consider the input pattern cannot be recognized as SP_i. ε is selected according to the noise environment.

When the input pattern X and the spatial pattern SP_i are not normalized, we can design U_i by a special second order binary neuron, called matching neuron. Actually, a second order binary neuron is defined by $\mathbf{W}$ which is an $(n+1)$-order real symmetric matrix. For an input signal pattern X, the output signal y of the second order binary neuron is computed by

$$y = Sgn(H(x)) = \begin{cases} 1 & \text{if } H(x) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where

$$H(X) = \sum_{i,j=0}^{n} w_{ij} x_i x_j, \qquad x_0 = 1.$$

Then, a matching neuron of the spatial pattern SP_i is defined by such a second order binary neuron that

$$w_{lj} = \begin{cases} -1 & \text{if } l = j, \\ sp_{il} & \text{if } j = 0, \\ sp_{ij} & \text{if } l = 0, \\ \theta - \sum_{j=1}^{n} sp_{ij}^2, & \text{if } l = j = 0, \end{cases}$$

where θ is a small positive number. That is, $H(X) = \theta - \sum_{j=1}^{n} (x_j - sp_{ij})^2 = \theta - \| X - SP_i \|^2$. When X is close to SP_i, i.e., the MSE between X and SP_i is less than θ, the neuron is activated and we consider the input pattern is recognized as SP_i; otherwise, the neuron is inhibited and we consider the input pattern cannot be recognized as SP_i. Thus, we can simply design U_i by such a matching neuron of SP_i, where θ is selected according to the noise environment.

For the both cases, when the parameters are properly selected, we can always have that for any input pattern there is at most one neuron in the second layer which is activated. We will assume that this is true in our design of the neurons in the second layer.

We now turn to the third layer of the neural network which consists of N neurons $V_1, \cdots, V_N$, corresponding to the N basic blocks $B_1, \cdots, B_N$, respectively, in $\mathcal{B}_S$. Clearly, each V_i should be able to detect whether B_i appears in the input spatial pattern sequence or not. In order to so, we design V_i as a Shift Register Matching (SRM) neuron of B_i, which is sketched in Figure 2. It consists of a receiving box, shift register, k matching units and a decision binary neuron. We further describe its structure and function with $B_i = SP_{i_1} \cdots SP_{i_k}$, where the index numbers of these spatial patterns are subject to those of the corresponding neurons in the second layer. Since some spatial patterns may be repeated in B_i, i_j may be also repeated. We suppose that all the different index numbers of the spatial patterns in B_i are $j_1, \cdots, j_{k_i}$. Then, only these $U_{j_1}, \cdots, U_{j_{k_i}}$ in the second layer connect to the receiving box of this SRM neuron.

At each time with an input pattern, if some U_{j_l} is activated, it sends a signal to the receiving box where the index number j_l is obtained and transmitted to the first left block of the shift register. Otherwise, the receiving box has no input signal and send the number zero to the first left block of the shift register. In the meantime, the index number stored in each block of the shift register is shifted to its right block. Here we assume that each block of the register keeps the number zero at the beginning. The index number in the j-th block is then transmitted to the j-th matching unit in the upper. The j-th matching unit has stored the index number i_j and make a decision whether the input index number is i_j or not. If the input index number is i_j, the matching unit will send a positive signal 1 to the binary decision neuron; otherwise, it will send a zero signal to the binary decision neuron. All the matching units connect to the binary decision neuron with a unit weight. And the threshold value of the binary decision neuron is near k. Clearly, when the index numbers in the k blocks from left to right approximately matches those of the spatial patterns of B_i from the last to the first, the binary decision neuron will be activated and the output is one. Otherwise, it will be inhibited and the output is zero. Thus, the SRM neuron will detect the basic block at all the time as the input spatio-temporal pattern enters.

The fourth layer is just one output neuron. The aim of this output neuron is to detect whether all or almost the basic blocks appear in the input spatial pattern sequence or not. So we can similarly design it as a binary neuron with one unit weight connection from each V_i and the threshold value near N. Moreover, it should have the capability of summing in a period of time.

We further describe how to operate the neural network filter for $\mathcal{S}$. When an input spatio-temporal pattern $X_1 \cdots X_m$ is provided to the input layer of the filter sequentially, each spatial pattern X_t is then transmitted to all the neurons in the second layer in a parallel way. When a neuron in the second layer is activated by X_t, it will send a signal each of the related SRM neurons synchronously. In the third layer at each time, if a SRM neuron has received a signal from some neuron in the second layer, the index number of this preceding neuron will enter the left block of the register while the old index numbers in the left $k-1$ blocks of the register will be shifted their right blocks, respectively. Then, the new index numbers will be transmitted to the decision units, respectively, and the decision results will be further transmitted to the binary neuron which will finally send a positive signal 1 or no signal to the output neuron of the filter. If a SRM neuron has not received a signal from each neuron in the second layer, a spacial index number 0 will enter the left block of the register and the other process keeps the same. During the period of the m times, if a SRM neuron has been activated one or more times, i.e., it has sent one or more positive signals to the output neuron, its contribution to the output neuron is considered only as one positive signal. Summing up all the signals in the m times, the output neuron will make the final decision. That is, if its output is one, the input spatio-temporal pattern is recognized as $\mathcal{S}$; it is zero, the input spatio-temporal pattern cannot be recognized as $\mathcal{S}$.

We finally analyze the function of the neural network spatio-temporal filter for $\mathcal{S}$ in a noisy environment. We first consider the case that $\mathcal{S}$ is a spatio-temporal pattern, i.e., $\mathcal{S} = P_1 \cdots P_m$. When the input spatio-temporal pattern $\mathcal{X} = X_1 \cdots X_m$ is just $\mathcal{S} = P_1 \cdots P_m$, we can easily find that the output of the filter is one. In fact, as each P_i enters the network sequentially, the corresponding neuron in the second layer will be activated and send the signal to the related SRM neurons synchronously. According to the function of the SMRM neuron, it will be certainly activated if the corresponding basic block enters the network sequentially. Thus, as $P_1 \cdots P_m$ enters the network sequentially, each basic block enters the network and the corresponding SRM neuron is activated at some time. That is, all the SRM neurons will be activated during the period of m times. So the output neuron will be finally activated and give the positive result.

In a noisy environment, the input spatio-temporal pattern $\mathcal{X}$ is near $\mathcal{S}$ as a whole, but different from it. That is, there exists some errors on the spatial patterns in comparison with those of $\mathcal{S}$. Then, we need to filter the errors in $\mathcal{X}$. Actually, there are three filtering processes in our neural network filter. If there are a small number of errors in a spatial pattern X_t in comparison with its real spatial pattern SP_i, they can be filtered by the SP neuron U_i as the parameters of the neurons in the second layer are properly selected. Sometimes, there may be so many errors in some X_t that the neurons in the second layer cannot filter them. That is, X_t is recognized as some other spatial pattern but SP_i, or cannot be recognized and all the neurons in the second layer are inhibited. Then, there appears a wrong spatial pattern in certain k-step blocks of the processed $\mathcal{X}$[1] in comparison with the corresponding basic blocks. If there exist a small number of wrong spatial patterns in a k-step block of the processed $\mathcal{X}$ in comparison with the corresponding basic block, the errors can be filtered by the SRM neuron for this basic block if the threshold value is properly selected. Moreover, if there exists too many wrong spatial patterns in a k-step block of the processed $\mathcal{X}$ in comparison with the corresponding basic block, the SRM neurons cannot filter those errors. That is, the corresponding SRM neuron will not be activated and thus cannot contribute to the output neuron. However, since $\mathcal{S}$ is near $\mathcal{S}$, there exits only a small number of these wrong

[1] Note that each spatial pattern X_i has been processed by the neurons in the second layer.

k-step blocks. The errors can be filtered by the output neuron if the threshold value is properly selected. Therefore, the neural network filter can recognize $\mathcal{S}$ in a noisy environment.

We further consider the case that $\mathcal{S}$ is a spatio-temporal cycle, i.e., $\mathcal{S} = P_1 \cdots P_{m_0} P_1$. In this situation, we design the neural network spatio-temporal filter as for the spatio-temporal pattern $\mathcal{S}' = P_1 \cdots P_{m_0} P_1 \cdots P_{k-1}$, i.e., $m = m_0 + k - 1$. Clearly, when the input spatio-temporal cycle enters the neural network spatio-temporal filter in such a way as $\mathcal{S}'$, we have the same result as above. Moreover, since the output neurons all check whether all or most of the basic blocks of $\mathcal{S}'$ appears in the input pattern sequence in a period of m times and there is not any other requirement, we can enter the input spatio-temporal cycle from any spatial pattern. Therefore, the neural network filter can recognized $\mathcal{S}$ in a noisy environment even if its phase is changed.

Additionally, if $\mathcal{S}$ is a given simple spatio-temporal pattern or cycle, all the 2-step blocks of $\mathcal{S}$ are certainly different. Moreover, $\mathcal{S}$ is uniquely defined by the set of its 2-step blocks. Thus, we can consider $\mathcal{S}$ as a second order spatio-temporal pattern or cycle to design the neural network spatio-temporal filter for it.

In a summary, the neural network spatio-temporal filter for a spatio-temporal pattern or cycle is constructed with the binary neurons for the spatial patterns and the SRM neurons for the basic blocks. It is able to store and recognize the spatio-temporal pattern or cycle in a noisy environment. Moreover, in the case of the spatio-temporal cycle, it can recognize the cycle even when its phase is changed.

IV. THE SIMULATION RESULTS

In this section, some simulation experiments are carried out to demonstrate our proposed neural network spatio-temporal filter. We begin with a description of the spatio-temporal pattern used in our simulation. Our simulation experiments are undertaken on the spatial pattern set of ten Arabic numerals $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ which are expressed by 7x7 pixels in Figure 3. That is, each number i is expressed by a binary matrix $S_i = (s_{ij})_{7\times 7}$, where $s_{ij} = 1$ represents the black pixel. Essentially, we can consider it as a vector, i.e., $vec[S_i]$. For those ten spatial patterns, we define the minimum Hamming distance of each sample pattern S_i to the other ones by

$$\begin{aligned} & d_i^* = min_{j\neq i} d_H(S_i, S_j) \\ = \quad & min\{d_H(S_i, S_j) : j = 1, \cdots, i-1, i+1, \cdots, 10\}. \end{aligned}$$

As is well-known in coding theory, $d_1^*, d_2^*, \cdots, d_{10}^*$ really give the bounds of radiuses of error-correcting hyperspheres of the spatial patterns(codes) in 49-dim binary space. In fact, the reasonable radius of error-correcting hypersphere of each S_i should be no more than $t_i^* = [\frac{d_i^*-1}{2}]$. (Here $[x]$ denotes the integer part of the real number x). For a filtering or recognition system, only when the radius of the error-correcting hypersphere of each S_i is just t_i^*, the error probability of recognition reaches the minimum in a noisy environment.

Based on the Hamming distances between these sample patterns, we have

$$\begin{aligned} & (t_1^*, t_2^*, t_3^*, t_4^*, t_5^*, t_6^*, t_7^*, t_8^*, t_9^*, t_10^*) \\ & = (3, 6, 5, 4, 9, 4, 5, 8, 3, 4). \end{aligned}$$

Then, we can design the binary neuron U_i in the second layer of the neural network filter as the t_i^*-neighborhood perceptive neuron of S_i according to Theorem 2.

Furthermore, we design the neural network filter for the spatio-temporal pattern $\mathcal{S} = \{7, 8, 8, 3, 6, 5, 6, 4, 9, 2, 0, 3, 6, 1, 6, 4, 7\}$ whose order is 3 and the number of the basic blocks is 15. Then, the filter is a four-layer neural network with 49 input neurons, 10 spatial pattern neurons (as designed above), 15 SRM neurons, and one output neuron. Each SRM neuron is designed as in Section III with the threshold value of the binary decision neuron 2.5. The threshold value of the output neuron is designed as 12.5.

We then generate four input spatial-temporal patterns under certain noise environment, which are given in Figure 4. We run the neural network filter on these four input spatio-temporal patterns. We have found that the neural network filter recognizes the input spatio-temporal patterns (a), (b), and (c), which contain a small number of errors, but does not recognize the input spatio-temporal pattern (d) which contains a large number of errors.

V. CONCLUSION

We have investigated the problem of filtering and recognition of complex spatio-temporal patterns or cycles through a neural network system in a general noise environment. By analysis, we have proved that the identification of a complex spatio-temporal pattern or cycle is equivalent to the identification of the set of its basic blocks. Based on this equivalence, we have established the neural network filter for a complex spatio-temporal pattern or cycle. According to the temporal correlations of the spatial patterns, the filter can be designed to remove the noise and recognize the pattern or cycle. Moreover, the neural network spatio-temporal filter is demonstrated by the simulation results.

References

[1] S. Grossberg, "Some networks that can learn, remember, and reproduce any number of complicated space-time patterns," *Int. J. Math. and Mechan.*, vol.16, pp:53-91, 1969.

[2] C. Staneley and W.L. Kilmer, "A wave model of temporal sequence learning," *Int. J. Man-Machine Study*, vol.7, pp: 397-412, 1975.

[3] R. Hecht-Nielsen, "Neural network nearest matched filter clasification of spatiotemporal patterns," *Applied Optics*, vol.26, pp:1892-1899, 1987.

[4] I. Guyon et al., "Storgae and retrieval of sequence in neural networks," *Phys. Rev. A.* vol.38, pp:6365-6372, 1988.

[5] D. Wang and M.A. Arbib, "Complex temporal sequence learning based on short-term memory," *Proc. of IEEE,* vol.78, pp:1356-1543, 1990.

[6] J. Ma, "The memory capacity of recurrent neural networks for storing spatio-temporal sequences," . *Proc. of Int. Conf. On Neural Information Processing*, 1998, pp:947-950.

[7] N. Srinivasa and N. Ahuja, "A topological and temporal correlator network for spatiotemporal pattern learning, recognition, and recall," *IEEE Trans. Neural Networks*, vol.10, pp:356-371, 2000.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, Cambridge, MA: MIT Press, vol.1, 1986.

[9] J. Moody and C. Darken, "fast learning in networks of locally-tuned processing units," *Neural Computations,* vol.1, pp:263-269, 1989.

[10] K.J. Lang, A. H. Waibel, and G. E. Hinton, "A time-delay neural network architecture from isolated word recognition," *Neural Networks*, vol.3, pp:23-44, 1990.

[11] J. Ma and D. Huang, "A neural network approach to filtering and recognition of complex spatio-temporal patterns," submitted to *IEEE Transactions on Signal Processing.*

[12] Jinwen Ma, "A neural network approach to real-time pattern recognition," *International Journal of Pattern Recognition and Artificial Intelligence,* vol.15, No.6, pp:937-47, 2001

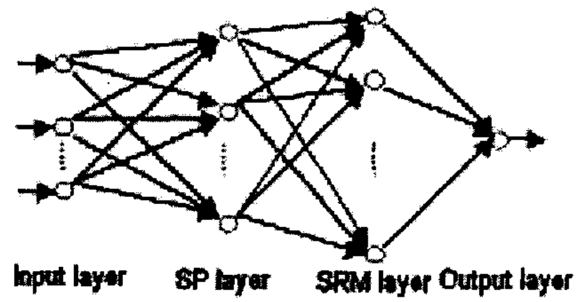

Figure 1. The sketch of the neural network filter

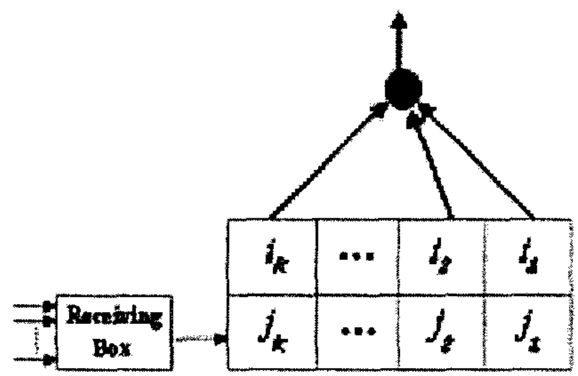

Figure 2. The sketch of the SRM neuron

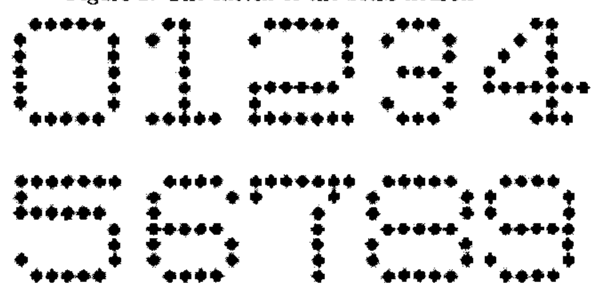

Figure 3. The set the spatial patterns $\{0,1,2,3,4,5,6,7,8,9\}$

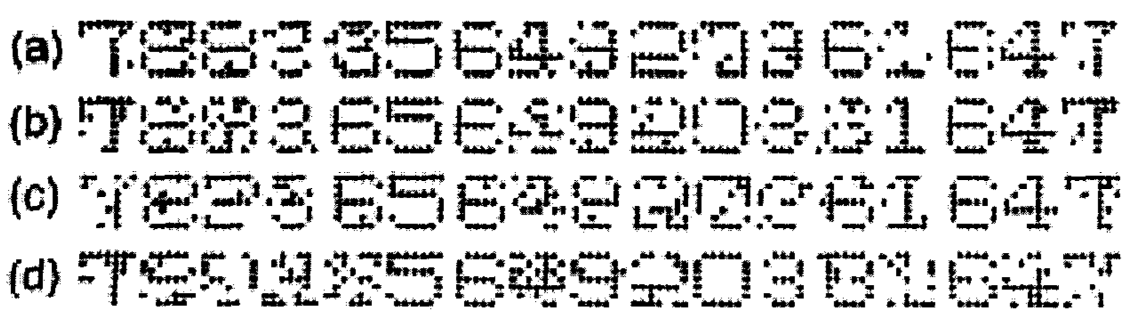

Figure 4. The four spatial-temporal patterns with noise.