Distance preserving 1D Turing-wave models via CNN, implementation of complex-valued CNN and solving a simple inverse pattern problem (detection)

G. Tóth, P. Foldesy, and T. Roska

Analogical and Neural Computing Laboratory, Computer and Automation Institute of the Hungarian Academy of Sciences, Kende u. 13, Budapest H-1111, Hungary

Abstract - In the first part of this paper a CNN implementation of a reaction-diffusion system is described that produces distance preserving periodic Turing patterns. In the second part the CNN with complex-valued templates are introduced, presenting an application for pattern generation. In the third part a method for black-and-white pattern detection will be described.

1. Introduction

The implementation of PDEs using CNN has been reported recently [5,6,11]. Turing-patterns are introduced by A. M. Turing [1] in 1952. They appear in physics, chemistry, biology. The hypothetical molecular mechanism is called reaction-diffusion system, and develops periodic patterns from the initially inhomogeneous state. Practically, the initial state always contains inhomogeneity, and this is enough to start pattern generation.

Stigers Kondo and Rihito Asai [2] used Turing-patterns to simulate the behavior of the skin of the marine angelfish Pomacanthus. On the skin of this fish the width of the stripes is independent of the length of the fish. As the fish grows, new vertical stripes appear between two old stripes, so these stripes are not fixed in the skin. Unlike mammal skin patterns which simply enlarge proportionally during body growth, these stripes maintain the spaces between the lines.

In this paper, in section 2 we show a CNN model of this phenomenon. In sections 3 and 4 we show how a complex-valued CNN can be implemented with two layers, and used for pattern generation. In section 5 we show a single 1D pattern detection mechanism.

2. CNN model of a one-dimensional Turing-type reaction-diffusion system found in Angelfish

2.1. Stripes arising from Turing-type reaction-diffusion equations

In [2], the following reaction-diffusion equations were identified as the governing equations for forming patterns:

$$\frac{dA}{dt} = c_1 A + c_2 I + c_3 + D_A \frac{d^2 A}{dx^2} - g_A A$$

$$\frac{dI}{dt} = c_4 A + c_5 + D_I \frac{d^2 I}{dx^2} - g_I I$$  (1)

Here x is the coordinate for the one-dimensional space, A and I are the concentrations of the two so-called morphogens, the Activator and Inhibitor molecules. Parameters $c_1$, $g_A$, $D_A$ are constants. In [1] it is proved that one of the spatial frequency components of the concentrations grows faster than the others and will eventually dominate. In other words, a spatial sine wave appears. This feature of the equations can be used for sine wave generation.

2.2. Realization using a 1D double-layer first-order CNN

The equations (1) are discretized in space in order to model it with a 1D cellular neural network [5]. The general form of the discretized equations is:
\[
\frac{dA_i}{dt} = aA_i + bI_i + c + \mu(A_{i-1} - 2A_i + A_{i+1}) \\
\frac{dI_i}{dt} = dA_i + eI_i + f + v(I_{i-1} - 2I_i + I_{i+1})
\]  

(2)

The parameters \(a, b, c, d, e, f, \mu,\) and \(v\) can be easily expressed in terms of \(c_b, g_e\) and \(D_i\).

**Figure 1.** Stripes in an image increase by 50% per second. (a) Two stripes and 115 unit length.

The cellular neural network is:

\[
\begin{align*}
A_{to1} &= [\mu \ (a-2\mu+1) \ 0] \\
A_{to2} &= [0 \ d \ 0]
\end{align*}
\]

The phenomenon found in Angenent's Turing-patterns. (The initial state is the image, at the right edge of the cell array.) An example is shown in Figure 1 where one stripe appears.

3. Complex valued CNN templates

Complex neural cells and networks were used. In the real case, variables are represented by their complex conjugates.

The state equation of the complex valued cellular neural network is:

\[
\frac{d(X_{R,ij} + jX_{I,ij})}{dt} = -(X_{R,ij} + jX_{I,ij}) + \sum_{(k,l) \in \mathcal{N}(i,j)} A_{R,kl} X_{R,kl} + A_{I,kl} X_{I,kl}
\]

where \(j = \sqrt{-1}\). The complex valued template is given by

\[
\frac{dX_{R,ij}}{dt} = -X_{R,ij} + \sum_{(k,l) \in \mathcal{N}(i,j)} A_{R,kl} X_{R,kl} + A_{I,kl} X_{I,kl}
\]

\[
\frac{dX_{I,ij}}{dt} = -X_{I,ij} + \sum_{(k,l) \in \mathcal{N}(i,j)} A_{R,kl} X_{I,kl} + A_{I,kl} X_{R,kl}
\]

The templates of the complex valued CNN are below:

<table>
<thead>
<tr>
<th>layer</th>
<th>feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex:</td>
<td>(A_R + jA_I)</td>
</tr>
<tr>
<td>Real:</td>
<td>Self:</td>
</tr>
<tr>
<td>Imaginary:</td>
<td>Self:</td>
</tr>
</tbody>
</table>

4. Pattern generation with CNN

Here an application of the complex valued CNN was used to generate patterns and we will compare the results. In the case of complex-valued CNN system, the initial state at the cell array can be any distribution, as long as starting from any initial state at the cell array
\[ v(I_{r+1} - 2I_i + I_{i-1}) \] (2)

\( i \) and \( D_t \)

Figure 1. Stripes in an increasing cell array. At start the length of the cell block is 60. After a given time it increases by 5%. (One inner cell of a 20 cell block is duplicated.) Three snapshots are shown. (a) Two stripes and 60 unit long array. (b) Three stripes and 90 unit long array. (c) Four stripes and 115 unit long array.

The cellular neural network templates implementing equations (2) are:

\[
\begin{align*}
A_{101} &= [\mu (a-2\mu+1) \mu] & A_{201} &= [0 b 0] & I_1 &= c \\
A_{102} &= [0 d 0] & A_{202} &= [v (e-2v+1) v] & I_2 &= f
\end{align*}
\] (3)

The phenomenon found in Angelfish can be modeled in the following way. We start the network that produces Turing-patterns. (The initial state is a random noise.) When the network reaches the steady state a new cell is added at the right edge of the cell array. Then we start the network and wait again. This sequence can be repeated several times. An example is shown in Figure 1. As the array size changes from 51 to 52 the second peak splits and a new one appears.

3. Complex valued CNN templates

Complex neural cells and networks were introduced in [10,12], where so-called multi-valued cells and complex templates were used. In the realization presented here both the templates and the states are complex. All complex variables are represented by their real and imaginary parts, and the implementation uses 2 standard CNN layers.

The state equation of the complex-valued CNN is:

\[
\frac{d(X_{R(i,j)} + jX_{I(i,j)})}{dt} = -(X_{R(i,j)}^2 + jX_{I(i,j)}) + \sum_{(k,l)\in N(i,j)} (A_{R} + jA_{I})_{kl} (Y_{R(k,l)} + jY_{I(k,l)}) + \sum_{(k,l)\in N(i,j)} (A_{R} + jA_{I})_{kl} (U_{R(k,l)} + jU_{I(k,l)})
\] (4)

where \( j = \sqrt{-1} \). The complex equation can be separated into two real equations:

\[
\begin{align*}
\frac{dX_{R(i,j)}}{dt} &= -X_{R(i,j)} + \sum_{(k,l)\in N(i,j)} (A_{R}Y_{R(k,l)} - A_{I}Y_{I(k,l)}) + \sum_{(k,l)\in N(i,j)} (B_{R}U_{R(k,l)} - B_{I}U_{I(k,l)}) + I_{R} \\
\frac{dX_{I(i,j)}}{dt} &= -X_{I(i,j)} + \sum_{(k,l)\in N(i,j)} (A_{R}Y_{I(k,l)} + A_{I}Y_{R(k,l)}) + \sum_{(k,l)\in N(i,j)} (B_{R}U_{I(k,l)} + B_{I}U_{R(k,l)}) + I_{I}
\end{align*}
\] (5)

The templates of the complex-valued CNN and its realization by a double-layer real-valued CNN are given below:

<table>
<thead>
<tr>
<th>layer</th>
<th>feedback</th>
<th>control</th>
<th>current</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex:</td>
<td>( A_{R}^* + jA_{I} )</td>
<td>( B_{R} + jB_{I} )</td>
<td>( I_{R} + jI_{I} )</td>
</tr>
<tr>
<td>Real:</td>
<td>Self:</td>
<td>( A_{R} )</td>
<td>( B_{R} )</td>
</tr>
<tr>
<td></td>
<td>From Imaginary:</td>
<td>(-A_{I})</td>
<td>(-B_{I})</td>
</tr>
<tr>
<td>Imaginary:</td>
<td>Self:</td>
<td>( A_{R} )</td>
<td>( B_{R} )</td>
</tr>
<tr>
<td></td>
<td>From Real:</td>
<td>( A_{I} )</td>
<td>( B_{I} )</td>
</tr>
</tbody>
</table>

4. Pattern generation with complex-valued templates

Here an application of the complex-valued CNN templates will be used to generate one-dimensional periodic patterns and we will compare the solution to the single-layer real-valued CNN implementation. We will see that in the case of complex-valued CNN the template size is only \( 3 \times 1 \), in the case of the real-valued CNN it is \( 5 \times 1 \).

Next we design a CNN to generate sine waves. The \( A \) template has a band-pass filter spectrum to achieve that starting from any initial state at the end only one spatial harmonic remains ([8-9]).
An autonomous CNN is described, that has a zero \(B\) template and \(I\) bias. Suppose that we are in the linear domain of the output nonlinearity. The cell's state equation using spatial convolution denoted by \('\ast\) is (capacitance \(C=I\) and resistor \(R=I\)):
\[
\frac{d}{dt} v_{xi}(t) = a_i \ast v_{xi}(t)
\]  
(6)

The relation between the \(a_i\) convolution mask and the \(A\) template is:
\[
A = \begin{bmatrix}
| a_r & \cdots & a_2 & a_1 & (a_o + 1) & a_{-1} & a_{-2} & \cdots & a_{-r} \\
\end{bmatrix}
\]  
(7)

Let \(a[\omega]\) be the spatial spectrum of \(a_i\) and \(v[\omega](t)\) the spatial spectrum of the state \(v_{xi}\) at time \(t\). Then \(v[\omega](t)\) can be expressed in the following way [8-9]:
\[
v[\omega](t) = v[\omega](0)e^{i\omega ct}
\]  
(8)

If the real part of \(a[\omega]\) is positive only in interval \([\omega_0 - \Delta\omega, \omega_0 + \Delta\omega]\) then the network will increase only the amplitude of the frequency components around \(\omega_0\). Practically, in a physical system the initial state contains all of the frequency components. If only the frequency components around \(\omega_0\) increase then a complex harmonic with frequency \(\omega_0\) will appear.

The following complex-valued template (9) has this property.

\[
A(\omega_0) = [0, \ e^{j\omega_0}, \ (b+1) \ e^{j\omega_0}, \ 0, \ e^{j\omega_0}, \ 0]
\]  
(9)

where \(a, b, \omega_0\) are positive constants. In this case \(a[\omega]\) is maximum at \(\omega = \omega_0\). We require that \(a[\omega] > 0\) only around \(\omega_0\). This can be achieved with the proper settings of \(a\) and \(b\): \(\max a[\omega] = a[0\omega] = 2a + 2a = 4a\) must be a small positive value.

The following templates are capable of generating sine waves with period-length \(L=10\):
\[
A = [0.202 - 0.147, \ 0.6, \ 0.202 + 0.147] \quad B = [0] \quad I = 0
\]  
(10)

In Figure 2 the generation process can be seen in case of a peak initial state. The pattern generation can be realized also by a real-valued CNN with \(5 \times 1\) templates:
\[
A = [-a \ 4a \cos(\omega_0), \ (b+1) \ 4a \cos(\omega_0) - a] \quad B = [0] \quad I = 0
\]  
(11)

where \(a\) and \(b\) are positive constants. As in the previous case, the constants must be chosen knowing that the maximum of the spectrum must be a small positive value.

Figure 2. Generating spatial sine waves (with period-length of 10 units) designing in the frequency domain. (a) The initial state (that is a peak), and the output of the real layer after (b) 30 and (c) 60 \(\tau\) can be seen. After 100\(\tau\) the whole area is filled with sine peaks.
5. 1D black and white pattern detection with 3×1 templates

In this section a new method for pattern detection on a one-dimensional black-and-white image is given. The main problem is to measure the length of a black or a white stripe with the CNN that contains only locally interconnected cells. Thus we have to measure a stripe with length of 20-30 units with a CNN that has a template size of 3 units. First an algorithm independent of the implementation will be presented then the realization with CNN will be described. The solution proposed here uses a series given by a recursive formula.

As mentioned above, first an algorithm independent of the implementation will be explained. Suppose, that we want to detect a stripe series containing black stripes with length of 5 units and 3 units space between them. (Thus the required pattern consists of black and white stripes with length of 5 and 3 units, respectively.)

![1D input image Table 1. Algorithm for stripe detection. The pattern to detect contains black stripes with length of 5 units and 3 units space between them. If there is a zero in the last row then there the algorithm detected the desired stripes.]

The steps of the detecting algorithm are the following (See also Table 1):

**Step 1** Count from left to right and put the result in row 1. Restart counting, if the actual element and the previous one of the black and white input image are different (i.e., if the border of a single-colored area is reached).

**Step 2** Count from right to left in a similar way and put the result in row 2.

**Step 3** Add row 1 and the row 2 and put the result in row 3. There is under a black or a white area its length plus one. To decide whether there are the desired stripes or not, we must compare the elements of the fourth row with \( 5+l=6 \) under a black stripe and with \( 3+l=4 \) under a white stripe.

**Step 4** Write in row 4 the value with which we should compare the elements of row 3 to detect the stripes with the desired length.

**Step 5** Subtract the row 5 from row 4 and put the result in row 5. If it is zero then there the required stripe series was found.

Next, the implementation in CNN will be described. It will be a three-layer network, although it can be implemented in CNN Universal Machine, too.

The first and the second layer performs the counting in both two directions (step 1 and step 2). A non-linear template (so-called switched type [13]) is used for this task, where the variable element of the \( A \) matrix depends on the actual cell states. This template starts the counting at the border of the stripes.

The third layer sums the results of the previous two layers (step 3). This layer also performs the comparison to the desired length (step 4). The base of the comparison can be tuned as required, by varying the interaction between this and the previous layers.

The third layer has a special output nonlinearity, which detects zero state of the cells with a given tolerance. The binary output of the this layer shows, where the input has exactly the required stripes.
Getting Order in Chaotic Systems by Self-Organization

Radu Dogaru,
*Applied Electronics Department,
Bvd. Iuliu Maniu 9,
**Institute of Microelectronics,
București, Romania

ABSTRACT: The effects of the CNN’s ability to exhibit autonomously oscillatory neural networks (CNN’s) have been extensively investigated. In these systems, a higher degree of order is observed. To describe these phenomena being observed in CNN’s, allowing interconnections on a three-dimensional space, we introduce a new concept of entropy and self-organization. The CNN’s framework is exploited as new classes of chaotic systems. Within the CNN’s framework, these phenomena are described as new classes of chaotic systems. Within the CNN’s framework, these phenomena are described as new classes of chaotic systems.