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

G. Tóth, P. Földesy, 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.

Shigeru 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:

$$\frac{dA}{dt} = c_1 A + c_2 I + c_3 + D_A \frac{d^2 A}{dx^2} - g_A A \qquad \frac{dI}{dt} = c_4 A + c_5 + D_I \frac{d^2 I}{dx^2} - g_I I \qquad (1)$$

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_i, g_i, D_i 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:

$$\frac{dA_i}{dt} = aA_i + bI_i + c + \mu(A_{i-1} - 2A_i + A_{i+1}) \qquad \frac{dI_i}{dt} = dA_i + eI_i + f + \nu(I_{i-1} - 2I_i + I_{i+1}) \qquad (2)$$

The parameters *a, b, c, d, e, f,* μ and ν can be easily expressed in terms of c_i, g_i, and D_i.

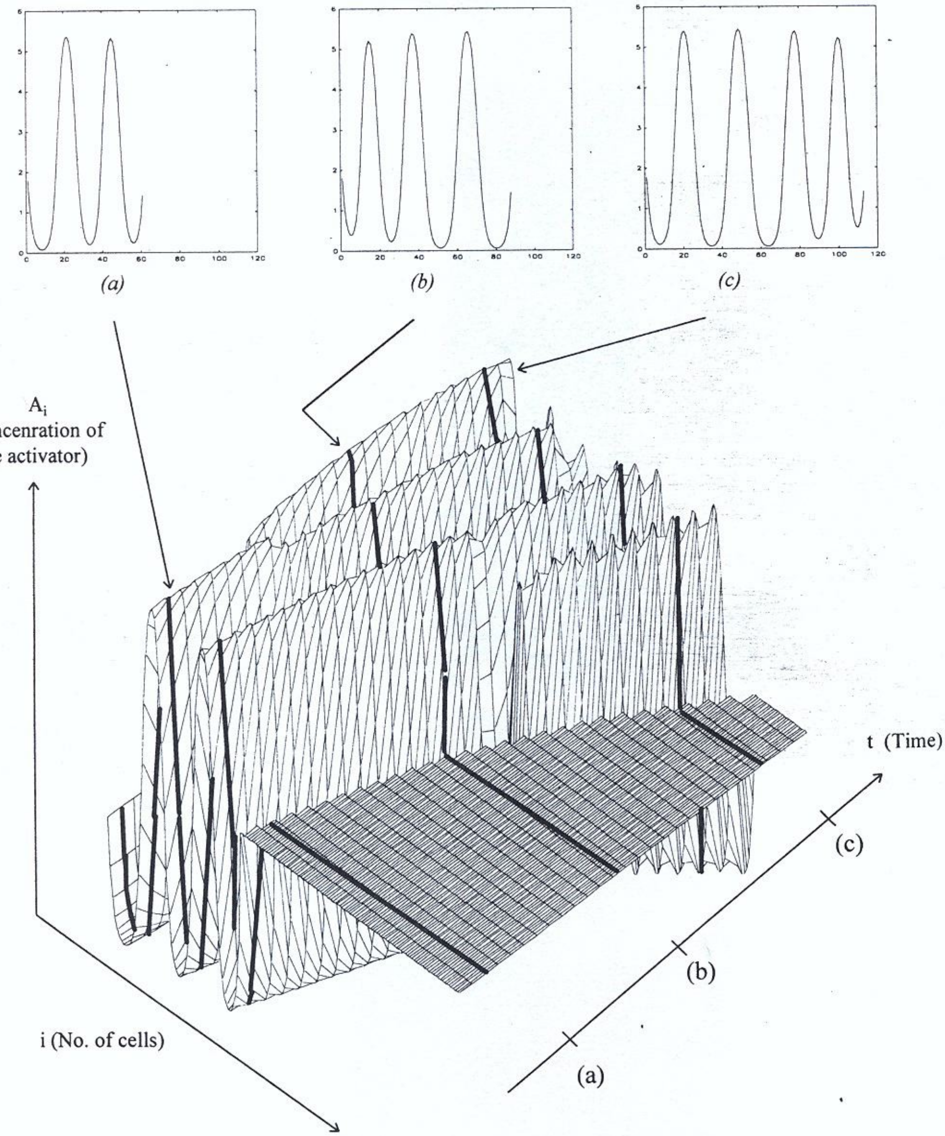


Figure 1. Stripes in an in
increases by 5
(a) Two stripes
and 115 unit l

The cellular neural network te

A1to1=[μ (a-2μ+1) μ]
A1to2=[0 d 0]

The phenomenon found in Ar
Turing-patterns. (The initial state
at the right edge of the cell array.
times. An example is shown in F
one appears.

3. Complex valued CNN te

Complex neural cells and net
templates were used. In the reali
variables are represented by their
The state equation of the com

$$\frac{d\left(X_{R,ij} + jX_{I,ij}\right)}{dt} = -\left(X_{R,ij} + jX\right.$$

where $j=\sqrt{-1}$. The complex

$$\frac{dX_{R,ij}}{dt} = -X_{R,ij} + \sum_{(k,l)\in Nr(i,j)} \left(A\right.$$

$$\frac{dX_{I,ij}}{dt} = -X_{I,ij} + \sum_{(k,l)\in Nr(i,j)} \left(A\right.$$

The templates of the complex
below:

layer	feedback
Complex:	A_R+jA_I
Real:	Self:
	From Ima
Imaginary:	Self:
	From Rea

4. Pattern generation with

Here an application of the c
patterns and we will compare th
the case of complex-valued CNN
Next we design a CNN to g
starting from *any* initial state at 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:

$$\begin{array}{lll} A1to1=[\,\mu \quad (a-2\mu+1) \quad \mu\,] & A2to1=[\,0 \quad b \quad 0\,] & I_1=c \\ A1to2=[\,0 \quad d \quad 0\,] & A2to2=[\,\nu \quad (e-2\nu+1) \quad \nu\,] & I_2=f \end{array} \quad (3)$$

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:

$$\frac{d\left(X_{R,ij}+jX_{I,ij}\right)}{dt}=-\left(X_{R,ij}+jX_{I,ij}\right)+\sum_{(k,l)\in Nr(i,j)}\left(A_R+jA_I\right)_{kl}\left(Y_{R,kl}+jY_{I,kl}\right)+\sum_{(k,l)\in Nr(i,j)}\left(A_R+jA_I\right)_{kl}\left(U_{R,kl}+jU_{I,kl}\right) \quad (4)$$

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

$$\begin{aligned} \frac{dX_{R,ij}}{dt}&=-X_{R,ij}+\sum_{(k,l)\in Nr(i,j)}\left(A_RY_{R,kl}-A_IY_{I,kl}\right)+\sum_{(k,l)\in Nr(i,j)}\left(B_RU_{R,kl}-B_IU_{I,kl}\right)+I_R \\ \frac{dX_{I,ij}}{dt}&=-X_{I,ij}+\sum_{(k,l)\in Nr(i,j)}\left(A_IY_{R,kl}+A_RY_{I,kl}\right)+\sum_{(k,l)\in Nr(i,j)}\left(B_IU_{R,kl}+B_RU_{I,kl}\right)+I_I \end{aligned} \quad (5)$$

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

layer	feedback		control	current
Complex:	A_R+jA_I		B_R+jB_I	I_R+jI_I
Real:	Self:	A_R	B_R	I_R
	From Imaginary:	$-A_I$	$-B_I$	
Imaginary:	Self:	A_R	B_R	I_I
	From Real:	A_I	B_I	

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 3×1, in the case of the real-valued CNN it is 5×1.

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 '*' is (capacitance *C=1* and resistor *R=1*):

$$\frac{d}{dt} v_{xi}(t) = a_i * v_{xi}(t) \tag{6}$$

The relation between the a_i convolution mask and the *A* template is:

$$A = [a_r \quad \dots \quad a_2 \quad a_1 \quad (a_o+1) \quad a_{-1} \quad a_{-2} \quad \dots \quad a_{-r}] \tag{7}$$

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] :

$$v[\omega](t)=v[\omega](0)e^{a[\omega]t} \tag{8}$$

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 ω_0. Practically, in a physical system the initial state contains all of the frequency components. If only the frequency components around ω_0 increase then a complex harmonic with frequency ω_0 will appear.

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

$$A(\omega_0)=[\ a\cdot e^{-j\omega 0} \quad (b+1) \quad a\cdot e^{j\omega 0}\ ] \qquad B=[\ 0\ ] \qquad I=0 \tag{9}$$

where *a, b,* ω_0 are positive constants. In this case $a[\omega]$ is maximum at $\omega=\omega_0$. We require that $a[\omega]>0$ only around ω_0. This can be achieved with the proper settings of *a* and *b*: $max\ a[\omega]=a[\omega_0]=2a+b$ must be a small positive value.

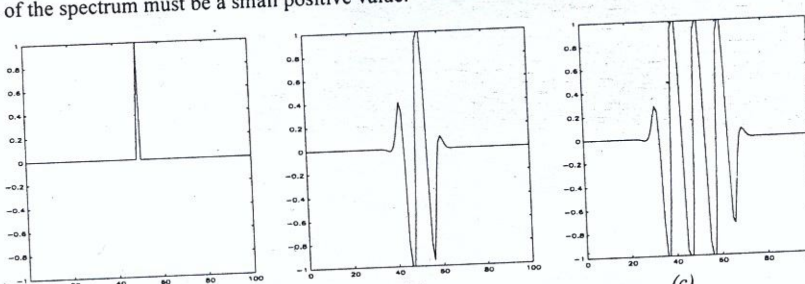
The following templates are capable of generating sine waves with period-length *L=10:*

$$A=[\ 0.202-0.147j \quad 0.6 \quad 0.202+0.147j\ ] \qquad B=[\ 0\ ] \qquad I=0 \tag{10}$$

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 5×1 templates:

$$A=[\ -a \quad 4a\cdot\cos(\omega_0) \quad (b+1) \quad 4a\cdot\cos(\omega_0) \quad -a\ ] \qquad B=[\ 0\ ] \qquad I=0 \tag{11}$$

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.



(a) (b) (c)

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 τ can be seen. After 100τ the whole area is filled with sine peaks.

uppose that we are in the linear
on denoted by '*' is (capacitance

(6)

(7)

of the state v_{xi} at time t. Then

(8)

ne network will increase only the
tem the initial state contains all of
ise then a complex harmonic with

(9)

$_0$. We require that $a[\omega]>0$ only
$\omega]=a[\omega_0$ $+b$ must be a small

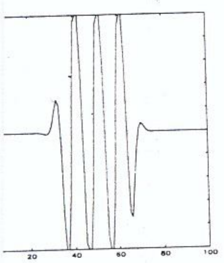
gth $L=10$:

(10)

ate. The pattern generation can be

(11)

s must be chosen knowing that the



(c)

10 units) designing in the
l the output of the real layer
hole area is filled with sine

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
1	2	3	4	5	1	2	3	1	2	3	4	5	1	2	3	1	2	3	4	1	2	1	2	3	1. Counting from left to right
5	4	3	2	1	3	2	1	5	4	3	2	1	3	2	1	4	3	2	1	2	1	3	2	1	2. Counting from right to left
6	6	6	6	6	4	4	4	6	6	6	6	6	4	4	4	5	5	5	5	3	3	4	4	4	3. Addition of row 2 and 3
6	6	6	6	6	4	4	4	6	6	6	6	6	4	4	4	6	6	6	6	4	4	6	6	6	4. Required value for row 4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	–1	–1	–1	–1	–1	–1	–2	–2	–2	5. Subtraction of row 5 from 4

Table I. *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 algoritm detected the desired stripes.*

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

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 no, we must compare the elements of the fourth row with $5+1=6$ under a black stripe and with $3+1=4$ 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 of 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 countation 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 countation at the border of the stripes.

The third layer summs the results of the previous two layers (step 3). This layer also performs the comparsion to the desired length (step 4). The base of the comparsion 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.

References

[1] A. M. Turing, "The chemical basis of morphogenesis", *Philos. Trans. R. Soc. London*, ser. B, vol. 237, pp. 37-72, 1952

[2] S. Kondo, R. Asai, "A reaction-diffusion wave on the skin of the marine angelfish Pomacanthus", *Nature*, vol. 376, pp. 765-768, 1995

[3] L. O. Chua, L. Yang, "Cellular neural network: Theory and Application", *IEEE Trans Circuits Syst.*, vol. 35, pp. 1257-1290,1988

[4] L. O. Chua, T. Roska, "The CNN Paradigm", *IEEE Trans. Circuits Syst.*, vol. 40(I), pp. 147-156, 1993

[5] T. Roska, L. O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, F. Puffer, K. Lotz, "Simulating nonlinear waves and partial differential equations via CNN-Part I: Basic techniques", *IEEE Trans. Circuits Syst.*, vol. 42(I), pp. 807-815, 1995

[6] T. Kozek, L. O. Chua, T. Roska, D. Wolf, R. Tetzlaff, F. Puffer, K. Lotz, "Simulating nonlinear waves and partial differential equations via CNN-Part II: Typical examples", *IEEE Trans. Circuits Syst.*, vol. 42(I), pp. 816-820, 1995

[7] L. O. Chua, M. Hasler, G. Moschytz, J. Neirynck, "Autonomous Cellular Neural Networks: A Unified Paradigm for Pattern Formation and Active Wave Propagation", *IEEE Trans. Circuits Syst.*, vol. I., pp. 559-577, 1995

[8] K. R. Crounse, L. O. Chua, "Method for Image Processing and Pattern Formation in Cellular Neural Networks: A Tutorial", *IEEE Trans. Circuits Syst.*, vol. 42(I), pp. 583-601, 1995

[9] P. Thiran, K. R. Crounse, L. O. Chua, M. Hastler, "Pattern Formation Properties of Autonomous Cellular Neural Networks", *IEEE Trans. Circuits Syst.*, vol. 42(I), pp. 757-775, 1995

[10] N. N. Aizenberg, I. N. Aizenberg, "CNN based on multi-values neuron as a model of associative memory for grey-scale images", *Proc. of IEEE CNNA'92 Conference*, pp. 36-41

[11] P. Szolgay, G. Vörös, Gy. Erôss, " On the implementation of the cellular neural network paradigm in mechanical vibrating systems", *IEEE Trans. Circuits Syst.*, vol. 40, pp. 222-227, 1993

[12] N. N. Aizenberg, Y. L. Ivaskiv, "Multi-valued treshold logic", Naukova Dumka Publisher House, Kiev, 1977

[13] L. Nemes, G. Tóth, T. Roska and A. Radványi, "Analogic CNN algorithms for 3D interpolation-approximation and object rotation using controlled switched templates", Int. Journal of Circuit Theory and Application, vol. II., May 1996

Getting Order in
by Self-Organizati

Radu Dogaru[*],
[*]Applied Electronics De
Bvd. Iuliu Ma
[$]Institute of Microelectronic
E

ABSTRACT: The effects of
experimentally investigated for
autonomous oscillatory neural n
CNN's. Allowing interconnecti
systems, higher degree of order
observations we have introduce
entropy and self-organization. T
phenomena being described in
systems. Within the CNN's fra
exploited as new classes of con
paper. Such applications inclu
and intelligent chaotic synchron

1. Introduction

In the last decades, an increasing num
what is by now known as the field of C
nonlinear dynamic systems is obvious w
different nonlinear behaviors to do diffe
non-linear input-output mappings (as use
complex fixed-point convergence (as in t
used for image processing tasks) and to e
recent solutions mentioned in literature
models [3]).

The problem of doing intelligent com
and evolution which are assumed to be sp
consider a system as being intelligent in
which allows it to generally solve the p
from the particular problems is effectivel
Such an example called „intelligent chao
and states to vary continuously, CNN's
framework for doing VLSI friendly co
different examples of applications were
especially for problems in the field of
dynamics, but recently a lot interesting
which are examples of oscillatory dyna
may also exhibit chaos, and using dis
produced in small size neural networks
using chain structures , efficiently ciphe
implement in both analog and digita
transmission systems or even new neura

However, what role plays adaptation
seems by now that very few CNN mod
binary states cellular automata, the eff
[9] but only for simple computational
non-friendly, adaptation and particula

0-7803-3261-X/96/$5.00 © 1996