Computation with chaotic patterns

Peter Andras

Neural Systems Group, University of Newcastle upon Tyne, Department of Psychology,
Ridley Building, NE1 7RU Newcastle, UK

Abstract

The paper is focused on how chaotic patterns, occurring in nature, might be used by biological organisms to perform computations. This issue is investigated in the context of neural systems. As a simple model of chaotic patterns, the world of Sierpinski triangles is analysed. The paper introduces the Sierpinski basis functions and the Sierpinski brain, which is able to perform classification and approximation. © 2002 Published by Elsevier Science B.V.

Keywords: Chaos; Computation; Emergent patterns; Neural networks; Sierpinski triangle

1. Introduction

Research results show that the emergence of chaotic patterns in biological organisms is usual [4]. Such cellular patterns also emerge in the neural systems of animals [4]. Other results indicate that the activity of various functional components of the neural system has a chaotic nature (e.g., activity within the olfactory bulb [6]). A new approach to the analysis of the role of chaotic neural phenomena is presented here. This approach relates the chaotic patterns to natural computation performed by biological organisms.

As simple model of biological chaotic patterns, the world of Sierpinski triangles [3] (see Fig. 1) is analysed. (The Sierpinski triangles are generated by adding the midlines to the original triangle, and by repeating this operation with the resulting triangles, except those which are formed by newly added lines, i.e., the resulting middle triangles are ignored.) The paper introduces the Sierpinski basis functions that are obtained by counting the number of intersections of two Sierpinski triangles with certain restrictions. It is shown that it is possible to perform classification and approximation tasks by using such Sierpinski triangles.

E-mail address: peter.andras@ncl.ac.uk (P. Andras).

0925-2312/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0925-2312(02)00347-8
To link these results to neural chaos, we propose simple neural network architectures, which use biologically realistic units and give rise to Sierpinski triangles in their spatio-temporal activity space. The paper presents briefly the Sierpinski brain [1], which combines such basic neural circuits to perform classification and approximation tasks. The learning within the Sierpinski brain is described on the basis of two natural principles: energy minimization and survival of the fittest. From these two principles emerges a learning procedure that leads to a robust solution of the classification or approximation task. The proposed interpretation of neural chaos suggests that attractors emerging in the space of spatio-temporal neural activity patterns might serve as building blocks for neural computations.

2. Approximation with Sierpinski basis functions

Usually neural computational problems appear formulated as classification, prediction or control problems, all of which can be reformulated in terms of approximation of some functions. We show in this section how to solve approximation problems using simple chaotic objects.

We start by considering two sets of Sierpinski triangles. The triangles of the first set have their baseline between the points (0, 0) and (1, 0), and their third vertex is (x, 1). The triangles of the second set have their baseline between the points (0, 1) and (1, 1), and their third vertex is (w, 0). We use the notation $S^n_x$ for the partial Sierpinski triangle obtained after adding the $n$th order midlines. The triangles of the two sets are denoted by $S_x$ and $T_w$, where $x$ and $w$ are the abscissa value of the third vertex. A Sierpinski basis function $s_x(w)$ is defined as the normalized number of intersections between the triangles $S_x$ and $T_w$. The normalization means that the actual numbers are divided by the largest possible number of intersections, which is obtained for symmetrical triangles (i.e., for $w = x$). Practically we approximate the Sierpinski basis functions taking the normalized functions obtained by considering the partial triangles $S^n_x$ and $T^n_w$. Fig. 2 shows an example of approximated Sierpinski basis functions.

In [5] it is shown that a parametric family of functions has universal approximation property with respect to the continuous functions if the functions have a finite integral.
over $\mathbb{R}$. To show this we decompose the Sierpinski basis functions into nine component functions defined by the combinations of edges of the two triangles. Mathematical analysis of the resulting functions indicates that the finite integral condition is satisfied for all of them [2]. Consequently, we can solve a wide range of approximation problems using linear combinations of Sierpinski basis functions. This means that by employing our simple chaotic objects it is possible to solve a large class of computational tasks.

7. Neural network formulation

We present the Sierpinski neural networks in this section. These networks produce a spatio-temporal activity pattern equivalent to Sierpinski triangles. The Sierpinski neural networks are built up by spiking excitatory and inhibitory neurons. The structure of the network is presented in Fig. 3.

The neurons are inhibitory neurons in the upper row, periodic bursters in the middle row, and integrate-and-fire excitatory neurons in the lower row. The burster-inhibitory complex selects one of the three pairs of the burster neurons (e.g., $a_x$ and $a_y$). At each time only one pair of bursters is firing. The $z_x$ and $z_y$ neurons integrate the incoming firing frequencies for a longer time period. The later the incoming signal, the smaller its contribution to the final sum. The integration process is formulated mathematically as

$$f_{z_x} = \sum_{i=1}^{n} \lambda_i f_i,$$

where $\lambda_i = 1/2^i$ and the $f_i$’s randomly take the values $f_{ax}$, $f_{bx}$, $f_{cx}$. The formula is similar for the $z_y$ neuron.

The Sierpinski neural network produces the Sierpinski triangle as the spatio-temporal pattern of firing rates of the $z_x$ and $z_y$ neurons. This is because the calculation of the $f_{z_x}$ and $f_{z_y}$ firing rates follows the calculation of the points of a Sierpinski triangle by a random generation method [3]. Having neural networks that produce the Sierpinski triangles in their spatio-temporal activity patterns allows us to combine them to obtain Sierpinski basis functions. Fig. 4 shows the interaction network that combines the output of two Sierpinski triangles.
The neuron $C$ receives excitatory input from two Sierpinski networks and detects the coincident firing of the $zx$ and $zx'$, respectively, $zy$ and $zy'$ neurons. If both pairs fire they signal an intersection point of the two Sierpinski triangles. The neuron $C$ counts the simultaneous coincident firings of the $(zx, zx')$ and $(zy, zy')$ neuron pairs for a time period. The output of the $C$ neuron represents the normalized result of this counting, i.e., the neuron $C$ computes the value of a corresponding Sierpinski basis function.

4. Performing computational tasks with neural chaos

Using several Sierpinski neural networks and interaction networks we can build a Sierpinski brain [1] that can perform simple computational tasks. The key question is
how to learn to perform these tasks. We suppose the presence of three burster neuron populations, $P_0$, $P_1$, and $P_{xw}$, of which neurons represent corresponding triangle vertex coordinates (i.e., 0, 1, and other $x$ and $w$ numbers). We further suppose that the neurons of the $P_{xw}$ population may change their bursting rate.

The learning in the Sierpinski brain is based on two natural principles: (1) the energy minimization principle, and (2) the survival of the fittest principle. The activity of the neurons within the Sierpinski neural networks is independent of the represented Sierpinski triangle (i.e. they all have the same representational size). The activity of the $C$ neurons in the interaction networks depends on the number of intersections of the Sierpinski triangles represented by the networks. Consequently the activity of the $C$ neurons is where the energy minimization principle may be applied. This means that the number of intersections between the Sierpinski triangles represented by the active networks should be reduced. This is formulated mathematically as

$$\min_a \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j s_i(x_j),$$

where the $a_i$'s are the numbers of $P_{xw}$ neurons representing the value $x_i$.

To reformulate the second principle in a mathematical way we consider principles of development of neural wiring. Those neural connections, which are regularly used survive, and neurons tend to participate in networks where they receive inputs from, and provide outputs to other neurons. In other words the neurons tend to join and form neural circuits, which are used relatively frequently by the neural system. Admitting the recruitment of new burster neurons into the structures of the Sierpinski brain, this leads to the mathematical formulation of the second principle as

$$\max_a \sum_{i=1}^{N} a_i,$$

Putting together the two learning principles we get the full optimization formula that describes the learning

$$\min_a \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j s_i(x_j) - \sum_{i=1}^{N} a_i.$$  

This resembles the support vector optimization formula [7] which leads to the minimal complexity solution of the considered problem. This shows that it is possible to perform computational tasks very efficiently by using emerging neural chaos as building blocks of the computational process.

5. Conclusions

A new interpretation of neural chaos is introduced in this paper that shows how emerging neural chaos might be used to perform neural computation. The new interpretation suggests that the activity of individual neurons is organized into spatio-temporal activity patterns that represent neural objects. The neural computation is performed by
the interaction of these neural objects that may take the form of chaotic firing patterns. We indicated how such a neural system can obtain robust, minimal complexity solution of problems formulated in terms of approximation tasks.

The suggested interpretation of neural activity points to new ways and directions of analysis of lower and higher level neural activity. These new directions are: the analysis of emerging chaotic neural activity patterns, the generative neural models of such patterns, and the study of interactions of such neural objects.

References


