Robust chaos in neural networks

A. Potapov *, M.K. Ali

Department of Physics, The University of Lethbridge, Lethbridge, Alberta, Canada T1K 3M4

Received 31 August 2000; accepted 20 October 2000
Communicated by C.R. Doering

Abstract

We consider the problem of creating a robust chaotic neural network. Robustness means that chaos cannot be destroyed by arbitrary small change of parameters [Phys. Rev. Lett. 80 (1998) 3049]. We present such networks of neurons with the activation function $f(x) = |\tanh s(x - c)|$. We show that in a certain range of $s$ and $c$ the dynamical system $x_{k+1} = f(x_k)$ cannot have stable periodic solutions, which proves the robustness. We also prove that chaos remains robust in a network of weakly connected such neurons. In the end, we discuss ways to enhance the statistical properties of data generated by such a map or network. © 2000 Elsevier Science B.V. All rights reserved.

PACS: 05.45; 87.18.Sn
Keywords: Robust chaos; Neural networks

1. Introduction

The problem of chaos in neural networks has received much attention recently. The activities in this field can be divided into three categories: (i) attempts to explain experimentally observed aperiodic behaviour in a single perturbed neuron or in a small assembly of neurons [1,2]; (ii) attempts to explain complex temporal behaviour of the brain and possible roles of chaos in information processing [2–7], and (iii) studies of routes to chaos and properties of chaotic attractors in neural network models [1,8,9]. The present paper falls mainly in the third category and partially in the second.

Artificial neural networks (ANN) provide new and effective solutions to problems in diverse fields [10–12]. In particular, they can serve as generators of chaotic signals. A number of applications, where chaotic signals are used, are mentioned in [13]. Another possibility of using chaos may be related with learning algorithms [10,14,15] in which a random search is used. In such cases, a neural generator of chaos may be a part of a deterministic neural network, which could implement stochastic methods of learning.

In applications, chaos should be robust [13] in the sense that it should not be destroyed by small perturbations of the system parameters. Such a perturbation may even be caused by a change of the computer or compiler in use and can be demonstrated with a very simple example. We considered the logistic map $x_{k+1} = ax_k(1 - x_k)$ for $a \in (3.6, 4.0)$ and calculated its Lyapunov exponents $\lambda$ for $10^4$ values of $a$ using single and double precisions on an Alpha workstation (DEC Fortran-90 compiler, real*4 and real*8 data types). In 23 cases the signs of $\lambda$ were...
different. For example, at $a = 3.967760$ with single precision $\lambda_s \approx 0.568$ and with double $\lambda_d = -0.245$. At $a = 3.649720$, $\lambda_s \approx -0.036$ and $\lambda_d = 0.249$. When we calculated $\lambda$ for the same $a$ on PC (AMD K6-2 processor, GNU C++ compiler, float and double data types) the results again were different: in the first case $\lambda_s \approx -0.216$, $\lambda_d \approx 0.581$, and in the second both values were positive and close to 0.25.

According to [13], a chaotic attractor is robust if, for its parameter values, there exists a neighborhood in the parameter space with no periodic attractor and the chaotic attractor is unique in that neighborhood. The question about the route in which chaos can loose its robustness has been addressed in [16]. In smooth 1-D maps $x_{i+1} = f(x_i)$ there are points $x_i$ where $f'(x_i) = 0$, and if the trajectory passes close enough to $x_i$, then it becomes stable. The idea is that in smooth higher-dimensional maps there is an analog of $x_i$, which is called a spine locus. Barreto et al. [16] discuss the possible structure of these loci and conjecture that typically a slight variation of $n$ parameters can destroy the chaos if $n \geq k$ where $k$ is the number of positive Lyapunov exponents. However, chaos typically cannot be so destroyed if $n < k$.

In this Letter, we will consider the problem of robust chaos in a class of artificial neural networks (ANN). The networks which we shall consider are dynamical systems with equations of motion of a special form. The basic elements of ANN are neurons and connections (to distinguish the elements of ANN from biological neurons, they sometimes are called “formal neurons”). A neuron receives weighted signals $w_{ij}x_j$ from other neurons or from external inputs through the connections, whose strength is characterized by the weights $w_{ij}$. Then the neuron sums up all its inputs $y_i = \sum w_{ij}x_j$ and performs a nonlinear transformation $f(y_i)$ which then serves as the output signal $x_i$. The function $f$ is called the “activation function”. Depending on the details of implementation and the structure of the matrix $w$, the resulting construction may be explicit or implicit function or a dynamical system (numerous examples can be found in [11]).

We shall consider neural networks that are the mappings $\mathbb{R}^n \to \mathbb{R}^n$ (dynamical systems with discrete time $t = 0, 1, 2, \ldots$), $x_i$, $i = 1, \ldots, n$, will be the dynamical variables and $f(y)$ will represent their values at the next time step. The general form of the equations of motion is

$$\dot{x}_i = f\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right), \quad i = 1, \ldots, n,$$

(1)

where $x_i \equiv x_i(t)$, $\dot{x}_i \equiv x_i(t + 1)$. We shall consider only recurrent networks, for which the variables $x_i$ cannot be renumbered such that the matrix $w$ becomes lower triangle, that is, $w_{ij} = 0$ for $j > i$ (otherwise there cannot be any nontrivial dynamics).

The properties of map (1) essentially depend on the activation function $f$. In most works on neural networks, it has a so called “sigmoid” form, for example $f = \tanh x$ or $f = (1 + e^{-x})^{-1}$ (monotonous function that tends to a constant value as $x \to \pm \infty$). In a number of papers nonmonotonous $f(x)$ has been used, e.g., [17–19]. As we shall show, nonmonotonous $f$ can represent a block of “sigmoid” neurons. Constructing a chaotic network from such blocks is equivalent to using a nonmonotonous activation function. Another possibility, which also may be used for generating chaos, is to use a sum of several activation functions. This may represent a “complex neuron” with two or more simple nonlinear elements acting simultaneously. For our purposes it will be enough to use only two:

$$\dot{x}_i = f_1\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right) + f_2\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right),$$

(2)

In such a case, even if both $f_1$ and $f_2$ are sigmoid, their combination can be nonmonotonous, which is convenient for chaos generation.

Our main purpose is to obtain “chaotic blocks”, which would enable us to construct a robust network of required complexity. We shall discuss possible ways of building such blocks with standard “sigmoid” neurons. But numerical calculations show that chaos in them is fragile. This is in a good agreement with the conjecture, that chaos in smooth systems is not robust [13,16].

We have obtained robust chaos with the nonmonotonous and nonsmooth functions $f(x) = |\tanh s(x - c)|$ and $f(x) = |s|x - c|/(1 + s|x - c|)$. Both of these functions can be obtained with the help of (2) or block-building technique from the nonsmooth “gated” sigmoid functions [20], e.g., $f^+(x) = \max(0, \tanh x)$. 


We shall show that, for a certain range of \( s \) and \( c \) values, the arising chaotic attractor satisfies \( |df/dx| > 1 \), and therefore the chaos is robust in this range. In addition, our numerical results indicate that robust chaos probably exists in a wider interval of parameter values. Even when \( |df/dx| < 1 \) on a part of the attractor, \( |df/dx| \geq \epsilon > 0 \), that is, separated from 0, and in terms of [16], there is no spine locus. Experiments show, that when a part of the attractor expands to the domain with \( \epsilon \) show, that when a part of the attractor expands to the domain with \( |df/dx| < 1 \), then for \( s \) not too large the Lyapunov exponent decreases gradually to zero, without abrupt falls caused by windows of periodicity. Then we prove that chaos in a network of weakly connected robust chaotic neurons is also robust. In the end we also discuss the problem of improving the statistical properties of chaotic data.

2. Chaos in the blocks with smooth activation function is fragile

Our main purpose is to create a neural generator of robust chaotic signals. A series of our experiments have shown that it is possible to obtain chaos in network of type (1) with the usual sigmoid \( f \). The problem is that in all of these cases it was easy to leave the domain of chaos by changing only one of \( w_{ij} \). In other words, chaos is not very typical. On the other hand chaos may be much more common if \( f \) is nonmonotonous [18].

Sigmoid-type neurons are typical in neural network models. It can be shown, that combinations of such neurons are equivalent to “complex” neuron with nonmonotonous \( f \). Such combinations or blocks also yield networks in which chaos is rather common. We shall describe two examples of chaotic blocks with a typical choice of \( f(x) = \tanh x \).

2.1. Near-logistic 2-neuron block

An example of how dynamical chaos can arise in a network of 3 sigmoid neurons, can be found, for example, in [21] (a very good explanation of how complex functions arise from a simple activation function can be found in [12]). The idea is that the function \( g(x) = \tanh(s(x - c_1)) - \tanh(s(x - c_2)) \) within a certain range of parameters resembles a logistic map — a well known example of a smooth chaotic system.

The same construction can arise in a usual 2-neuron network with special choice of connection matrix \( w \). Let us consider a network equations in matrix form

\[
\begin{align*}
\dot{x} &= f(wx + b), \quad x, \dot{x}, b \in \mathbb{R}^2,
\end{align*}
\]

where \( \dot{x} \) stands for \( x \) at the next time step. Let us make a change of variable, \( y = Tx \), where \( T \) is a \( 2 \times 2 \) matrix. Then

\[
\begin{align*}
\dot{y} &= T\dot{x} = Tf(wx + b) = Tf(wT^{-1}y + b).
\end{align*}
\]

Let us take, for example,

\[
\begin{align*}
T &= \frac{1}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \\
wT^{-1} &= \begin{pmatrix} s & 0 \\ s & 0 \end{pmatrix}, \\
w &= \frac{1}{2} \begin{pmatrix} s & -s \\ s & -s \end{pmatrix},
\end{align*}
\]

then the resulting equations of motion for \( y \) are

\[
\begin{align*}
\dot{y}_1 &= \frac{1}{2} \left( f(sy_1 + b_1) - f(sy_1 + b_1) \right), \\
\dot{y}_2 &= \frac{1}{2} \left( f(sy_1 + b_1) + f(sy_1 + b_2) \right),
\end{align*}
\]

or, in original variables,

\[
\begin{align*}
\dot{x}_1 &= f\left( \frac{s}{2}x_1 - \frac{s}{2}x_2 + b_1 \right), \\
\dot{x}_2 &= f\left( \frac{s}{2}x_1 - \frac{s}{2}x_2 + b_2 \right).
\end{align*}
\]

Since the matrix \( w \) is singular, this is in fact a one-dimensional dynamical system (it depends only on one variable \( y_1 \) while \( y_2 \) is its function). But the same dynamics can be obtained with a nonsingular matrix, if the second eigenvalue will be a small nonzero number \( \epsilon \).

It is convenient to write this block as a one-dimensional map

\[
\dot{x} = \tanh(s(x - c + d) - \tanh(s(x - c - d)),
\]

where \( c = -(b_1 + b_2)/2s, \quad d = (b_1 - b_2)/2s \). An example of bifurcation diagram for this map, where \( s = 6, \quad d = 0.1 \) and \( c \) is the bifurcation parameter is shown in Fig. 1. Panels (b)-(d) of this figure show the estimated value of Lyapunov exponent for this map on smaller and smaller scales. It is seen, that the domains where chaos seem persistent, contain periodic windows. The situation is similar if we vary \( s \) or \( d \).
Fig. 1. Example of bifurcation diagram (panel (a)) and Lyapunov exponent (panels (b)–(d)) for the neural chaotic block — the mapping
\[ \dot{x} = \tanh s(x - c + d) - \tanh s(x - c - d), \quad s = 6, \quad d = 0.1, \quad c \text{ is the bifurcation parameter.} \]
Inside the regions of chaos there are parameter values, for which the behavior is regular and Lyapunov exponent \( \lambda \leq 0 \). The intervals of \( c \) which seem chaotic on smaller scales also prove to contain islands of regularity (panels (c) and (d)). Even if such islands are not found, it is impossible to guarantee their absence. Therefore, chaos in such blocks is not robust. Situation is similar if we vary \( s \) or \( d \).

It is hard to apply here directly the theory developed for the logistic map because here the bifurcation parameters are different. Nonetheless, numerical experiments show the presence of stable periodic solutions among chaotic ones.

2.2. A block of coupled oscillatory neurons

Another 2-neuron chaotic block, not related directly to the logistic map, can be obtained with the help of “oscillating neurons”. Let us consider a single-neuron dynamical system with the equation of motion
\[ \dot{x} = f(s(c - x)), \quad s > 1, \quad f(x) = \tanh x. \quad (8) \]
System (8) always has a fixed point, the solution of equation \( x_s = \tanh s(c - x_s) \). The derivative at this point
\[
\left. \frac{df}{dx} \right|_{x=x_s} = -\frac{s}{\cosh^2 s(c - x_s)}
= -s(1 - \tanh^2 s(c - x_s))
= -s(1 - x_s^2).
\]
When \( s(1 - x_s^2) > 1 \), this fixed point becomes unstable, and a stable period-2 cycle arises [22]. It is well known, that chaos may arise in a system of coupled nonlinear oscillators, so we tried to obtain a chaotic block this way. To achieve maximum versatility of coupling, we chose the value of \( s \), for which oscillations in a single neuron are observed in the widest range of \( c \). This \( s \) can be obtained from the analysis of bifurcation points.
If we fix \( s \) and consider \( c \) as a bifurcation parameter, then the bifurcation points \( c_b \) can be obtained from the equation \( |f'(x_0)| = 1 \) or

\[
\cosh s(c_b - x_a) = \sqrt{s}, \tag{9}
\]

then we immediately obtain that

\[
x_a = \tanh s(c_b - x_a) = \pm \sqrt{\frac{s-1}{s}}.
\]

From (9) it follows that

\[
s(c_b - x_a) = \ln(\sqrt{s} \pm \sqrt{s-1}) = \pm sr,
\]

\[
r = s^{-1} \ln(\sqrt{s} + \sqrt{s-1}), \tag{10}
\]

and

\[
c_b = \pm \left( \sqrt{\frac{s-1}{s}} + r \right).
\]

One of these points corresponds to arising of the periodic solution, while other — to disappearance. The length of the \( c \) interval, where oscillations exist, is

\[
\Delta c = c_{b+} - c_{b-} = 2 \left( \sqrt{\frac{s-1}{s}} + \frac{1}{s} \ln(\sqrt{s} + \sqrt{s-1}) \right).
\]

There is \( s \) value corresponding to the largest \( \Delta c \), which satisfy \( \frac{d}{ds} \Delta c = 0 \) or

\[
\sqrt{\frac{s}{s-1}} = \ln(\sqrt{s} + \sqrt{s-1}).
\]

Numerical solution of this equation gives \( s_0 \approx 3.2767 \) with \( \Delta c = (s(s-1))^{-1/2} \approx 2.3994 \).

In the coupled system we used this \( s \) value. The resulting equations were

\[
\dot{x}_1 = f(c_1 - sx_1 + ax_2), \tag{11}
\]

\[
\dot{x}_2 = f(c_2 - ax_1 - sx_2). \tag{12}
\]

We indeed observed dynamical chaos for some choices of \( c_i \) and \( a \). An example of bifurcation diagram and plots of the largest Lyapunov exponents are shown in Fig. 2. Despite another form of the diagram, like in the previous case of Fig. 1, chaos is fragile.

2.3. Higher-dimensional networks

We also made a number of attempts to obtain robust chaos in a higher dimension network. The results can be summarized as follows.

The larger the network size \( n \), the easier it is to find chaos if the matrix \( w \) is chosen randomly. However, it is hard to find a high-dimensional attractor with random choice. Usually, the largest dimension of randomly found attractor was about \( n/3 \) for \( n > 5 \). The number of positive Lyapunov exponents also did not exceed \( \sim n/3 \), while the number of parameters was \( \sim n^2 \). The conjecture in [16] implies that in such situation it is very easy to destroy chaos. Our computational results are in good agreement with this conjecture: the domain of chaoticity in parameter space is ragged and chaos is easily destroyed by a change of parameter values. According to [16], the use of block with effective nonmonotonous activation function can be considered as an attempt to diminish the number of parameters (because of restrictions on the choice of \( u_{ij} \)). But it seems that the more important reason for chaos fragility is the smoothness of \( f \) (the conjecture of chaos fragility in [16] is made only for smooth dynamical systems that are at least differentiable). For this reason we tried neurons with the properties close to the tent map, where \( f \) is nondifferentiable in one point.

3. Robust chaos in neural network

3.1. “Nonsmooth neuron” and the pincers map

The most well known example of robust chaos is given by a tent map [23]. For this mapping \( |df/dx| > 1 \), and therefore there is no stabilizing spine locus [16], like in the logistic map or logistic network. The tent map is non-differentiable at its vertex, where it behaves like \( |x| \), and the conjecture in [16] does not hold for it. We tried to build a neural network with similar features by using activation function of the form

\[
f(x) = |\tanh s(x - c)|. \tag{13}
\]

The plot of \( f(x) \) is shown in Fig. 3. It has two “wings” which meet at \( x = c \), where \( f = 0 \). We shall consider two kinds of such tent neurons and show that both of
Fig. 2. Example of bifurcation diagram (panel (a)) and the largest Lyapunov exponent for another chaotic neural block — two coupled oscillating neurons (11)–(12). The values of the parameters are $s = 3.2767$, the coupling strength $a = 0.45$, $c_1 = c$ is the bifurcation parameter, $c_2 = c_1 + 0.3$. Chaos also proves not robust.

them have robust chaotic behaviour for a range of $s$ and $c$. It is clear, that chaos can exist only for $s > 1$ and $c > 0$.

Like in the case of the logistic block (5)–(6), such a transfer function can be implemented as a two-neuron network with singular connection matrix $w$ and the nonsmooth sigmoid activation function

$$f^+(x) = \begin{cases} \tanh x, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

Functions of this type have been used, for example, in [20]. With this definition, we can write $f(x) = f^+(s(x - c)) + f^+(s(c - x))$. In what follows, we shall not specify how such a mapping has been obtained, we will just use the activation function (13).

The bifurcation diagram for the mapping

$$\hat{x} = |\tanh s(x - c)|$$

is shown in Fig. 4 (because of the form of the diagram in Fig. 4(b), we called (14) the "pincers map"). The dependence of Lyapunov exponent on $c$ is shown in Fig. 5. Numerical experiment gives "visibly continuous" dependence of $\lambda(c)$ in the domain where $\lambda > 0$. The continuity hardly can be proved, but we can present evidence that within a certain $c$ interval, $\lambda$ should be strictly positive and therefore chaos should be robust.

The dotted lines in Fig. 4 correspond to $x = c$ (middle line) and the points $x = x_{ri}$, $i = 1, 2$, where $|df(x_{ri})/dx| = 1$ (other two lines). The equation for $x_r$ values, $\cosh s(x_r - c) = \sqrt{s}$, coincides with (9), and therefore $x_{r1} = c - r$, $x_{r2} = c - r$, where $r$ is defined in (10). It can be seen from Fig. 4, that there exists an interval of $c$ values, where the chaotic attractor is located in the domain $x_{r1} < x < x_{r2}$. These intervals correspond to robust chaos, since everywhere on the
Mapping (14) maps the interval \( T \) also mapped into itself (Fig. 3). The mapping takes its sometimes there exist smaller subinterval, which is \( f \).

If \( x \) the point \( f \). this interval maps onto itself (dotted square in Fig. 3). then \( f_x/ \), then \( f \). \( f \). \( f \). \( sc < \).

If \( sc \) an one of the condition for strict robustness

\[
\tanh sc \leq 2c. \tag{16}
\]

If \( s \leq 2 \), (16) always holds. For larger \( s \), we can use the first two terms in the expansion of \( \tanh x \),

\[
s c - \frac{1}{3} (sc)^3 \leq \tanh sc \leq 2c,
\]

\[
c \geq c_{s1} = \frac{\sqrt{3(s - 2)}}{s} \tag{17}
\]

This can be observed on bifurcation diagrams: for \( s \leq 2 \) chaos begins at \( c = 0 \), while for \( s > 2 \) it begins at \( c > 0 \). The value of \( c_{s1} \) (17) is shown in Figs. 4 and 5 as \( \uparrow 1 \).

For guaranteed robustness of chaos, another condition should hold: everywhere within the attractor \( |f'| > 1 \). On the interval \([0, f(0)]\), \( |f'| \) takes its lowest value at \( x = 0 \). Therefore, we need that the point \( x = 0 \) falls within the interval \([x_r, x_l] \), that is, \( x_r = c - r \leq 0 \) or \( c \leq r \). \( c = r \) is marked in Figs. 4 and 5 as \( \uparrow 3 \). Note, that conditions (17) and (18) do not contradict each other, since we always have

\[
\ln(\sqrt{s + \sqrt{s - 1}}) \geq \frac{3(s - 2)}{s}. \tag{18}
\]

Another problem is that for small \( c \) values, the chaotic attractor coexists with a stable fixed point (see Fig. 3(a)). The chaotic attractor is reached not from all initial data, therefore, such chaos also lacks robustness. This fixed point disappears when the right wing of the map is tangent to the line \( f = x \). Consequently, at the point \( x \) of the fixed point disappearance \( f(x) = x \) and \( f'(x) = 1 \), so \( x = x_r2 \). This gives the relations

\[
x_{r2} = c + r = f(c + r) = \tanh sr
\]

\[
= \frac{\sinh sr}{\cosh sr} = \sqrt{\frac{s - 1}{s}},
\]

and another condition for chaos robustness

\[
c > \sqrt{\frac{s - 1}{s}} - r
\]

(this \( c \) value is marked in Figs. 4 and 5 as \( \uparrow 2 \)). Together with (18) it gives the length of \( c \) interval, where chaos is robust

\[
\Delta c = 2r - \sqrt{s - 1/s}.
\]

It achieves its largest value when \( d(\Delta c)/ds = 0 \), which gives \( s \cong 1.30 \) with \( \Delta c \cong 0.32 \) (\( c_{\text{min}} = 0.077 \), \( c_{\text{max}} = 0.402 \)).

As \( s \) grows, \( r \to 0 \), and at some \( s = s_u \) there will be no guaranteed robustness at all with \( \Delta c = 0 \). Numerical solution for the equation \( \Delta c = 0 \) gives \( s_u \cong 2.73 \).
Fig. 4. Examples of bifurcation diagrams for the pincers map (14), corresponding to different $s$ values: 1.1 (a), 1.3 (b — the largest interval of guaranteed robustness), 1.7 (c), 2.0 (d), 2.7 (e — guaranteed robustness disappears), 4.0 (f). Arrows correspond to characteristic $c$ values: 1 — onset of chaos; 2 — disappearing of fixed point; 3 — end of guaranteed robustness of chaos; 4 — value for which $f'(0)f'(c) = 1$; 5 — value for which $f(x) = c$.

However, numerical calculations show that chaos appears to remain robust on larger $c$ interval, and the Lyapunov exponent gradually tends to 0 as $c$ increases. This fact is hard to prove, but the explanation may be that attractor gradually extends to the domain, where $|f'| < 1$, but since $|f'(x)| \approx |f'(0)| = s / \cosh^2(sc)$, the mean value $\langle \ln |f'(x)| \rangle$ still remain positive. We can propose an approximate estimate of the largest possible $c$ value, based on the assumption that chaos disappears close to the $c$ where $|f'(0)||f'(c)| \approx 1$ (the product of the largest and the smallest $|f'(x)|$ values). This gives the condition $\cosh sc = s$ or

$$c_{\text{max}} = s^{-1} \ln \left( s + \sqrt{s^2 - 1} \right).$$
This $c$ value is marked in Figs. 4 and 5 as $\uparrow 4$. For $s < \sim 2$ it indeed proves to be close to the point of chaos disappearance, or, for greater $s > \sim 2.5$, to the point of appearing of windows of periodicity (Figs. 4 and 5).

Another estimate of maximum $c$ can be obtained from the fact that chaos cannot exist when the image of 0 is less than $c$ (in this case the system behaves like a neuron with monotonous activation function, and the most complex behaviour is a period-2 oscillation). This means that chaos cannot exist if $\tanh sc \leq c$.

The numerical solution for the equation $\tanh sc = c$ is marked in Figs. 4 and 5 as $\uparrow 5$, and it is seen, that it always is outside the domain of chaos.
We also studied the behaviour of another inverse tent neuron which uses hyperbolic activation function instead of hyperbolic tangent, and the equation of motion has the form
\[ \dot{x} = f(x), \quad f(x) = \frac{s|x - c|}{1 + s|x - c|}. \]
Qualitatively, it behaves like the previously analyzed neuron, though all estimates are different. Use of this function strongly reduces the amount of computations, but the maximal interval of guaranteed chaos robustness is only \( \Delta c = 0.125 \) at \( s = 4/3 \). The actual interval of chaos robustness in this case also seems essentially wider, and the bifurcations of chaos disappearance are simpler.

### 3.2. Robust chaotic network

When we have a robust chaotic neuron, it is rather simple to organize a robust chaotic network of the form
\[ \dot{x}_i = \tanh \left( x_i + a \sum_{j=1}^{n} w_{ij} x_j - c \right), \]
\[ w_{ij} = 0, \quad |w_{ij}| < 1, \quad a > 0, \quad i = 1, \ldots, n. \]
(19)

Let \( c \) be in the middle of the guaranteed robustness interval: \( c = r - \frac{1}{2} \Delta c \). For brevity, it is convenient to write the \( n \)-dimensional system (19) as a vector system \( \dot{x} = F(x, a) \).

For \( a = 0 \) we have \( n \) independent chaotic maps, and the matrix of derivatives \( DF = \text{diag}(DF_{ii}) \) with all \( |DF_{ii}| \geq \alpha(c) > 1 \),
\[ \alpha(c) = |f'(0)| = s / \cosh^2(s c) \] (20)
(here and below we shall write \( f(x) \) instead of \( \tanh(s(x - c)) \) for brevity).

Now let \( a > 0 \) but small enough, such that inequality
\[ g = a \sum_{j=1}^{n} w_{ij} x_j \leq a \sum_{j=1}^{n} |w_{ij}| \leq \kappa \]
holds (it takes into account that \( |x_i| \leq 1 \)). We shall give the estimate for \( \kappa \) later.

For the matrix of derivatives \( DF \) the following relations are obvious
\[ |DF_{ij}| = |f'(x)| \cdot a|w_{ij}| \leq sa|w_{ij}|, \quad i \neq j, \]
\[ |DF_{ii}| = |f'(x + a \sum w_{ij} x_j)| \geq \alpha(c + g) \geq \alpha(c + \kappa) \]
does not hold (the greater \( \kappa \), the smaller is \( |f'(0)| \)). According to the Gershgorin circles theorem, all eigenvalues \( \mu_k \) of \( DF \) should belong to the circles on the complex plane with the centers at \( P_i = DF_{ii} \) and radii \( R_i = \sum_{j \neq i} |DF_{ij}| \).
Therefore,
\[ |\mu_k| \geq \min_{i} \left\{ |DF_{ii}| - \sum_{j \neq i} |DF_{ij}| \right\} \geq \alpha(c + \kappa) - \kappa, \]
and the sufficient condition for \( |\mu_k| > 1 \) is \( \alpha(c + \kappa) > 1 + \kappa \). Using the expression for \( \alpha(c) \) (20), we can write it as
\[ \cosh(s(c + \kappa)) < \sqrt{s / (1 + s \kappa)}. \]

For \( \kappa = 0 \) this inequality holds because of the choice of \( c \). At the left hand side we have exponentially growing function of \( \kappa \), while at the right hand side — decreasing. Hence, at some \( \kappa = \kappa_{\text{max}} > 0 \) inequality (21) will cease to hold.

It is necessary to note, that in any case \( |f'(0)| > 1 \) must hold, which gives the rough upper bound \( \kappa < \kappa'_{\text{max}} = r - c \) (see previous section). To obtain a lower bound for \( \kappa_{\text{max}} \), let us estimate the dependence on \( \kappa \) in (21) by a linear approximation. To avoid overestimation, it is necessary to linearize the right hand side at \( \kappa = 0 \), while for the left hand side we must take the estimates of the derivative at \( \kappa = \kappa_{\text{max}} \) and for the latter we shall take its upper bound \( \kappa'_{\text{max}} \).
This gives the equation for \( \kappa_{\text{max}} \):
\[ \cosh(s(c + \kappa_{\text{max}})) \sinh(s(c + \kappa'_{\text{max}})) = \sqrt{s} \left( 1 - \frac{s \kappa_{\text{max}}}{2} \right), \]
or, taking into account that \( c + \kappa'_{\text{max}} = r \) and \( \sinh(s r) = \sqrt{s - s^2} \),
\[ \kappa_{\text{max}} = \frac{1}{s} \sqrt{s - \cosh(s c)}} + \frac{1}{2} \frac{1}{s} \sqrt{s}. \] (22)
Therefore, for \( k < \kappa_{\text{max}} \) all eigenvalues of \( DF \) satisfy \( |\mu_k| \gg |\mu_{\text{min}}| = \alpha(c + k) - sk > 1 \). It immediately gives, that any solution of linearized equation (19),
\[
\dot{u} = DF(x)u,
\]
will grow at least as \( |\mu_{\text{min}}|^c \), and hence all \( n \) Lyapunov exponents satisfy \( \lambda_k \geq \ln |\mu_{\text{min}}| > 0 \).

Small variations of \( w_{ij} \) which do not violate these conditions, will not change this conclusion. Therefore, for small enough \( a \) chaos in the network (19) is robust.

Note, that this result gives only sufficient, but not necessary conditions for robustness. Numerical experiments show, that as a rule the interval of robustness is wider, and practical requirement is that \( e^c = c + |\alpha \sum_{j \neq i} w_{ij} x_j| \) belongs to the interval of chaos robustness in a single map.

4. Adaptive neural corrector of probability distribution density

If we consider a time series of the value of one of the variables, which describes the state of the chaotic dynamical system, then this series can be used as pseudorandom numbers. In this case it may be desirable to make these numbers independent and as close as possible to any standard distribution, for example, uniform on \([0, 1]\).

If \( x \) were a random number with the continuous distribution density \( p(x) \), then \( y = F(x) = \int_{-\infty}^{x} p(z) \, dz \) would be a random number with the uniform distribution on \([0, 1]\). A transformation with some approximation \( g(x) \) of \( F(x) \) may give density that is almost uniform. But invariant measures of dynamical systems may be singular or discontinuous, and usually it is impossible to transform such distribution into continuous one by means of a piecewise differentiable function \( g(x) \). Nonetheless, if we consider a simpler problem, to draw a coarse histogram obtained with \( g(x) \) look like that for uniform distribution, then such correction could be done in an adaptive way.

Let the probability distribution density \( p(x) \) be localized on \([x_{\text{min}}, x_{\text{max}}]\), that is, \( p(x) = 0 \) for \( x < x_{\text{min}} \) and \( x > x_{\text{max}} \). Then \( F(x_{\text{min}}) = 0 \), \( F(x_{\text{max}}) = 1 \), and we need an approximation of \( F(x) \) on this interval. Let us choose \( n + 1 \) points \( b_i, i = 0, 1, \ldots, n \), such that \( b_0 = x_{\text{min}}, b_n = x_{\text{max}} \), and \( F(x_i) - F(x_{i-1}) = \Delta F = n^{-1} \). Therefore, the intervals \([b_{i-1}, b_i]\) will have equal probability. Let us consider the piecewise linear transformation
\[
y = \left(i - 1 + \frac{x - b_{i-1}}{b_i - b_{i-1}}\right) \Delta F, \quad \text{if} \ x \in [b_{i-1}, b_i].
\]

For \( y \), the intervals \([y_{i-1}, y_i], y_i = F(b_i)\), will have equal length and equal probability, and the histogram, formed by these intervals, will be uniform. On smaller scales the distribution of \( y \) will remain nonuniform, but it may be not very important in specific applications.

If we are dealing with a set of data \( x_k \), generated by a dynamical system, then we should choose \( b_i \) such that the number of points within each interval \([b_{i-1}, b_i]\) will be equal. The simplest way to do it is to use a large set of \( x_k \). If keeping numerous data is impossible, this problem can be solved “on the fly”, simultaneously with collecting data without keeping them, with the help of stochastic approximation techniques [10].

Suppose that \( b_i \) are fixed and we are making a histogram of \( x_k \). Let us denote by \( h_i \) the share of \( N \) points, which fall into \([b_{i-1}, b_i]\). Suppose that the next number \( x_{N+1} \) comes, and it falls into \( j \)th interval.

Let us denote by \( \tilde{h}_i \) the values of \( h_i \) after adding the next \( x \). The number of points within \( i \)th interval \((i \neq j)\) remains equal to \( Nh_i \), and within \( j \)th interval \( Nh_j + 1 \). So, if we denote \( \beta = (N + 1)^{-1} \),
\[
\tilde{h}_i = \frac{N}{N + 1} h_i + \frac{1}{N + 1} \delta_{ij} = (1 - \beta) h_i + \beta \delta_{ij},
\]
\[(24)\]

In this relation \( \beta \) should also change after adding each \( x \), \( \tilde{\beta} = (\beta^{-1} + 1)^{-1} = \beta/(\beta + 1) \) (it agrees with the requirements of stochastic approximation techniques that \( \sum \beta(t) = \infty, \sum \beta(t)^2 < \infty \)).

Adjusting of \( b_i \) we made in the following way. If two neighboring histogram elements \( h_{i-1} \) and \( h_i \) are not equal, we are shifting the boundary \( b_i \) between them towards the largest, that is \( \Delta b_i \sim h_{i-1} - h_{i-1} \). At the same time after this shift the histogram should change, and we add the correcting terms to \( h_k \) proportional to \( \Delta b \).

During these changes caution is necessary to avoid situations, when \( b_i \) becomes more than \( b_{i+1} \) or \( h_k < 0 \). In a sense, this problem is equivalent to that of the stability of difference schemes for differential
equations, that is, algorithm should be stable. We propose the following implementation.

Let us consider the change of $b_i$ and introduce two values, $\kappa_+$ and $\kappa_-$,

$$\begin{align*}
\kappa_+ &= \gamma \max\{0, h_i - h_{i-1}\} < 1, \\
\kappa_- &= \gamma \max\{0, h_{i-1} - h_i\} < 1,
\end{align*}$$

where $\gamma < 1$ is a parameter. That is, only one of $\kappa$ is nonzero, and both are nonnegative. Then the algorithm for $b_i$ is

$$\begin{align*}
\hat{b}_i &= b_i + \kappa_+(b_{i+1} - b_i) - \kappa_-(b_i - b_{i-1}) \\
&= (1 - \kappa_+ - \kappa_-)b_i + \kappa_-h_{i-1} + \kappa_+b_{i+1},
\end{align*}$$

for $i = 1, \ldots, n - 1$. The correction of $h_i$ is performed assuming that the density inside each segment is uniform and $\sim h_i$. Then

$$\begin{align*}
h_{i-1}' &= (1 - \epsilon\kappa_-)h_{i-1} + \epsilon\kappa_+h_i, \\
h_i' &= (1 - \epsilon\kappa_+)h_i + \epsilon\kappa_-h_{i-1},
\end{align*}$$

Experiments show, that for better results the correction rates for $h$ should be smaller than for $b$, for this purpose a small factor $\epsilon < 1$ is introduced.

Finally, we let $b_0$ and $b_n$ tend to the smallest and the largest of $x_k$:

$$\begin{align*}
\hat{b}_0 &= (1 - \epsilon\gamma)b_0 + \epsilon\gamma \min_k x_k, \\
\hat{b}_n &= (1 - \epsilon\gamma)b_n + \epsilon\gamma \max_k x_k.
\end{align*}$$

$\epsilon$ should be small enough, to ensure $b_0 < b_1, b_n > b_{n-1}$.

The whole scheme can be implemented as a rather large but simple network. In particular, approximations (23) can be done by neurons with piecewise linear activation function $f(x) = 0$ for $x < 0$, $f(x) = x$ for $0 \leq x \leq 1$, $f(x) = 1$ for $x > 1$. Approximation with smooth sigmoid function also possible, but it requires solving a linear system of equations for coefficients (this case is almost identical to the use of radial basis function network [10,12]).

Fig. 6. (a) Histogram ($10^3$ bins) for the map $\hat{x} = |\tanh x(x - c)|$ for $x = 2, c = 0.5$; (b)–(d) Histograms after correction (23) with various number of correction intervals: $n = 10$ (b), 20 (c) and 50 (d). Correction cannot exclude discontinuities, but overall distribution looks more uniform.
This procedure has been applied to the probability distribution for mapping (14) for $s = 2, c = 0.5$. The original distribution is shown in Fig. 6(a), and the 1000-bin histogram of the signal after transformation (23) with $n = 10, 20$ and 50 — in Figs. 6(b)–(d).

As it was told, such transformation cannot exclude discontinuities, but “on average” the distribution looks essentially more uniform (its distribution function $F_1(y)$ looks almost linear).

Another problem may be related with correlations in the series of data generated by a single map (14). To exclude correlations, we used $k$ identical maps, numbered from 1 to $k$, with different initial conditions and the distributor. The first output number it takes from the first map, the second — from the map 2 and so on. After $k$th map it returns to the first and so on. As a result, each individual map generates only each $k$th number in the series. For $k = 10$ we could not detect any correlations.

So, the final scheme of the neural random generator is the following: $k$ isolated maps or networks $\rightarrow$ distributor $\rightarrow$ corrector (23) $\rightarrow$ output number $y_i$.

We checked its properties on the tests, proposed in [24]: random walk test and block test. Correction was done with $n = 20$, $\gamma = 0.1$, $\epsilon = 0.1$, the parameters $b_j$ were determined by $10^7$ data points, $k = 10$ independent maps with $s = 2$ and $c = 0.5$ were used, and the set of $y_i$ passed both tests.

It is hard to say, whether such a neural generator is useful in such applications as Monte Carlo methods, but it seems very probable, that it can serve as a part of a controlling neural network, which should make a random choice to learn various control policies [14,15].

5. Conclusion

In this Letter we have shown how to design a robust chaotic neural network with the help of usual but nonsmooth activation functions. We have given the estimates of the range of parameters for which robust chaos exists and have shown how to improve its statistical characteristics. It is possible that, beyond the usual applications related with chaotic data [13], such robust chaotic elements may be a part of deterministic neural networks, implementing control policies, related to random choices.

Acknowledgements

This work has been supported through a grant to M.K.A. from the Defense Research Establishment Suffield under the Contract No. W7702-8-R745/001/ EDM. The work was carried out using computing facilities provided by Multimedia Advanced Computational Infrastructure (MACI).

References