IMPERIAL COLLEGE LONDON

E4.34
SO17
ISE4.29

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2003

MSc and EEE/ISE PART IV: M.Eng. and ACGI

Examined by the following coursework alone — + short oral exam

**ANALYSIS OF NEURAL NETWORK MODELS**

Time allowed: 0:15 hours

**There are ZERO questions on this paper.**

**Answer ZERO questions.**

**Any special instructions for invigilators and information for candidates are on page 1.**

| | | |
|---|---|---|
| Examiners responsible | First Marker(s) : | De Wilde, P. |
| | Second Marker(s) : | Mamdani, E.H. |

(Corrected) June 03
Copy.

# Neural Network Models Exam Coursework 2002-2003 Deadline April 28th 2003 at 10.00 am *

This exam is based on sections 1, 4, and 5 of the paper *Nonlinear Dynamics and Chaos in Information Processing Neural Networks*, A. B. Potapov and M. K. Ali, Differential Equations and Dynamical Systems, Vol. 9, Nos. 3&4, July&October 2001, p.259–319. The paper is available from the undergraduate office on level 6. It describes several neural network models that have a chaotic behaviour. They are extensions of the analog neural networks that you have seen in the course. Read section 1. It is not necessary to look up references for this coursework exam. Ignore anything about Hamiltonian neural networks.

Recall the fundamental set of equations for an analog neural network:

$$\mu_i \frac{dx_i(t)}{dt} = -x_i(t) + \sum_{j=1}^{N} w_{ij} f_j(x_j(t)) + I_i(t), \quad i = 1, \ldots, N, \quad \mu_i > 0,$$

where I have used $x$ for the state variables, and $w$ for the weights, the same convention as in this paper. Every trajectory of this system eventually reaches a set of points which it visits time and again, ad infinitum. This set of points can be just a single point in state space. This is the case that we have used to store patterns. It can also be a curve in state space, this is called a closed orbit. Finally, it can also be a set of points different from a single point and a closed orbit. This is called a chaotic set. A neural network shows chaotic behaviour if some of its attractors are chaotic sets. You do not really need this definition of chaos to understand the paper, the intuition I gave you in the lectures is sufficient.

The main part of your coursework is based on section 4. How can chaos help in the exploration of possibilities (2. *creation of information*) (1/20)? How would you transform a chaotic signal into a deterministic answer (5.) (1/20)?

The first neural network model introduced is equation (7). Why is $z$ called self-feedback and $\beta$ the damping term? The equation (6) that is referred to is the system of dynamical equations for an analog network cited

[*] Post it in the appropriate locked box in the undergraduate office, level 6.

chaos → exploration 1

chaos → determinism 1

above. At this stage, just make sure you understand the equations. The claims made by the authors will be investigated later. Equation (8) introduces another neural network model. For sigmoid function, read tanh. Ignore TSP. Equations (9) and (10) describe the third neural network model. In (10), $F$ should read $f$, and $(1+|E|)^{-1}$ is $\frac{1}{1+|E|}$. What is the range of the neuron states $x_i$? The fourth and final neural network model is described in (11) and (12). Ignore anything about bifurcation diagrams and logistic maps.

Now read the rest of section 4, concentrating on the variety of dynamical behaviour described rather than the mathematics. If the four neural networks were used to store patterns, and if the Hebbian rule was used to determine the weights, which network would be the easiest to analyze the dynamical behaviour of, and to control the chaos in order to avoid spurious 5 patterns (5/20)? Answer this question using the theory you have seen in the course, and the information about the models given in section 4. A reasoned answer will take about a page. The purpose is not to guess the right answer, but to give a proper justification.

Now concentrate on the network you have chosen. Implement a three-neuron system, and attempt to store three patterns. To study the dynamics, you only have to implement the equations, there are no differential equations to be solved. The difficult part is finding the weights, and the values of other parameters. You will have to experiment to find these. Are there spurious 2 states (2/20)? An exhaustive search is not required. Show, via simulations, how you use the chaos in the pattern retrieval (6/20). 6 Make clear what settings you found accidentally, and which ones you found via reasoning. If there are some accidental settings that you understand after doing the simulations, you can pretend you found them by reasoning! The more you show you were in control of your simulations, the higher your marks will be on this part of the coursework.

To make sure your simulations differ from those of your friends, choose the three most significant digits of one of the variables you use equal to the three least significant digits of your CID number (or the day of the month of your birthday if you have no CID number).

**The Challenge**. If you want to get top marks, you have to do this challenge. However, you will get better marks for a good report without the challenge than for a mediocre report with the challenge solved.

Read section 5 of the paper. Make sure you understand what is chaotic in (14), from the text and figure 1. Do not worry about Lyapunov exponents. Then read about the cart-pole balancing task. Without implementing it, 2 describe a learning algorithm for the cart-pole balancing (2/20), and explain where chaos could arise (3/20). 3 Your learning algorithm should be different from the reinforcement learning described in section 6 of the article.

You could organize your work as follows.

day 1 Read the paper, looking up anything you don't understand in your lecture notes. Answer the questions that can be answered without programming. Plan what you are going to program.

day 2 Do the programming, and debug your program.

day 3 Run the simulations, and collect the results in a form that you can present in your report. Simulations can be in any programming language, on any machine. The use of Matlab may simplify plotting data, but will not give you any other advantages.

day 4 Write the report. It should be maximum six pages (single sided) a4, in a font not smaller than 10 point. You will not get marks for anything exceeding six pages, even if it is appendices. Font size in tables and figures should be at least 10 point, or the tables and figures will not be marked. Describe the problem, and how you have solved it. Describe your simulations, but do not give programme listings. Do not give references to the literature. Make sure you do and answer everything that is asked for in the coursework. Do not bind the report, but staple the pages together. Mention your name, and indicate for what degree (e.g. MEng Elec. Eng., MEng ISE, MSc) you are studying.

day 5 Check the consistency and quality of your work. Make last minute changes if necessary. If you feel confident and have the time, tackle the challenge. Resist the temptation to spend more than five 8-hour days of intensive effort on your coursework. You will not be compensated for it in marks. Just as an exam paper requires a concentrated effort over a few hours, this coursework requires a concentrated effort over a few days.

Do not forget to attend on the "exam" day. This day will be advertised in your exam schedule. Bring a copy of your report with you, and your college security card. I will ask you one or two questions based on what you have written in your report, to make sure that you have written it yourself. No preparation is necessary.

Good luck.

Dr. P. De Wilde

ory neural

ed Neural
Springer-

ernational

H. Cohen,
nic Move-
ork, 1988.
in chains
l Applied

s, Boston,

A. Baxter,
uron ring,

tern gen-

pled bio-

and J. P.
tic input,

cillations

r-Verlag,

ns, New

Orleans,
uno.edu.

37N
37D45
(92B20)
68T05

Differential Equations and Dynamical Systems,
Vol. 9, Nos. 3 & 4, July & October 2001, pp. 259–319.

# Nonlinear Dynamics and Chaos in Information Processing Neural Networks

A. B. Potapov and M. K. Ali

3-LETH-P

**Abstract**

We consider a number of possible roles of complex dynamics and chaos in information processing by neural networks. First, we review the working principles of some well-known neural networks, and then discuss a number of approaches to utilization of chaos in neural networks. Our main goal is to present a novel view of the problem of chaos in information processing. We demonstrate that chaos emerges naturally in controls when a neural network forms a controlling part of a more complex system. We show that such neural networks can enhance efficiency by using chaos for explorations in a method known as Reinforcement Learning. A discussion on Hamiltonian neural networks is also included.

## 1. Introduction

Artificial neural networks (ANNs) are widely used now in many engineering applications and research problems [17, 18, 45, 46, 59, 58, 79, 95, 97]. Since ANNs are efficient tools for information processing, there is an ongoing quest for improving their performance, widening the areas of applications and finding new working principles. About ten tears ago, attempts were made to enhance the performance of ANNs on the basis of their complex or chaotic temporal

[0] AMS (MOS) 2000 Subject classifications: 37D45, 92B20.

0

behaviors (see e.g., [93, 34] for the review). The question is: can chaos be useful for information processing? There are arguments both in favor and against the question of usefulness of chaos. On one hand, activities of the brain demonstrate complex and possibly chaotic temporal behaviors, that suggests that maybe the brain uses chaos for sustaining life. On the other hand, neural networks that are currently in use for practical purposes are designed to be nonchaotic on the presumption that chaos is not needed for information processing (though chaotic information processing systems exist, e.g., [5]).

The purpose of this paper is to reflect on the underlying principles of operation of existing neural networks from the point view of nonlinear dynamics, to discuss a number of attempts to endow networks with chaotic behavior, and to present a new perspective of the role of chaos in neural networks. If we consider a neural network as an element of a larger system interacting with the world, then dynamical chaos can emerge in rather simple models. A number of such models are known, for example, in artificial intelligence. Moreover, systems interacting with their surroundings need a source of 'initiatives' to pursue exploration and learning from experience. Dynamical chaos can serve as a source of such initiatives. In this work, we also discuss Hamiltonian neural networks that have received little attention so far in information processing. Hamiltonian neural networks have the advantage that their quantum analogs can be studied.

The paper is organized as follows. First, in Section 2, we discuss definitions and various viewpoints of neural networks. In Section 3, we consider working principles of some existing neural networks to explain why, in the opinions of users of these networks, complex dynamics and chaos are not necessary. In Section 4, we give reasons why chaos may be useful. In Sections 5, 6, and 7 we discuss different approaches to the problem of chaos and complex behavior in neural networks.

## 2. What is an Artificial Neural Network?

In spite of the growing number of publications in the field, there is no consensus on the precise definition of ANNs. The reason for this is that there are too many types of them. Sometimes a more general term called "connectionism" is used for ANNs. The term connectionism means a methodology of making a complex system by a combination of connected elements that are similar or identical [29]. The basic nonlinear elements of an ANN are called formal neurons are "clumped" and play the role of parameters rather than dynamic variables. Freeman's model of olfactory system [35] also falls into this category. Some networks of this type will be considered in the next section.

## 4. Complex Dynamics as an Attempt to Improve ANN

Artificial neural networks appear to be universal tools for approximations, pattern recognition, etc. Nonetheless, difficulties may arise during their application. For functional networks, for example, problems arise when one tries to approximate very complex functional dependence: a too simple network can not give proper approximation, and a too complex one amplifies noise. This problem is called "overfitting" and it is usually solved by introducing a special "second level" structure of a network called the modular and ensemble networks [82].

For conventional dynamical neural networks considered above, the problems that arise most often are (1) convergence to wrong attractors (false memories), (2) too slow convergence to attractors and (3) failure to reproduce the activities of the real brain.

Dynamics indeed has a potential for information processing. For example, an interesting idea about its application has been proposed by A.Dmitriev et al. [4]: an image is stored as a long cycle of a dynamical system, and each step of the cycle corresponds to an image pixel.

Another type of networks with complex dynamics emerged from the analogy between neuron or small groups of neurons and an oscillator. Such a neural network is a lattice of coupled oscillators. It has been shown that information may be stored and processed in the phases of the oscillators. Such an oscillatory network may implement the principles of a Hopfield network [52] or other types of networks, e.g., [69, 48].

The idea that neural networks can work in chaotic regimes has also been proposed in a number of papers. A good review of the basic approaches in this field for early 1990s can be found in [34, 93], and we shall not repeat all that has already been reported. We shall describe several models focusing on the basic directions of studies in this field. In order to ascertain the usefulness of chaos, it is instructive to pay attention to the following properties of chaos [28, 40]:

1. *Local instability.* Local instability might be useful for preventing false memories and staying off trajectories that lead to undesirable locations in the phase space. This effect of chaos may be useful only during a transitional period while the trajectory converges to a "true memory." After such a transition, the dynamics should again become stable.

2. *Creation of information.* This may be the most attractive property of dynamical chaos. The fact that a chaotic system behaves unexpectedly can be used to search a new way for solving a problem. In other words, chaos can help in exploration of possibilities. However, modern neural network architectures can not use this property, because their desired behaviors are predictable: always the same output for a given input. There were attempts to use chaotic signals in algorithms for random minimization during learning of multilayer perceptrons, but deterministic methods had the upper hand. Neural networks capable of exploration during their learning are still to be found.

3. *Wandering along attractor.* This property is closely related to instability. There were attempts to use it as a tool for memory search with the help of chaotic attractor. The trajectory wanders between images, and the idea was to use this property for matching input data with the stored images. Several experiments have been done and preliminary results obtained, but the corresponding pattern recognition algorithm has not been developed.

4. *Resemblance with complex behavior of the brain.* A number of attempts have been made, mainly by Freeman and his colleagues [34, 31, 32, 99] to create a neural network with the structure resembling that of a part of the brain, mostly the olfactory bulb. The behavior of the brain may strongly change in time, and chaotic systems also can demonstrate a variety of behaviors. The behavior of models used resembled experimental signals from the brain.

5. Finally, chaos may be used without any special role: we can create a multiple-attractor system with periodic or chaotic attractors instead of a fixed point [10]. The kind or number of a chaotic attractor can be considered as a result of recognition, though a special decoding subsystem must be added to transform a chaotic signal into the deterministic network answer.

In the following section, we shall consider several neural networks with chaos.

nting false
ions in the
ransitional
h a transi-

roperty of
tedly can
ds, chaos
l network
viors are
tempts to
arning of
d. Neural
found.

to insta-
with the
and the
images.
, but the

attempts
, 99] to
e brain,
ange in
rs. The
in.

reate a
a fixed
ed as a
dded to

s with

## 4.1. Transient Chaos vs False Memories

### 4.1.1. Chaotic Modifications of the Hopfield-Tank Model

One of the drawbacks of the Hopfield-Tank model is that the dynamics can be trapped in local minima. To overcome this difficulty, a number of workers [87, 62] have used transient chaos and noise. Recently Kwok and Smith [62] have presented a unified approach to such neural networks. It is worth reporting one of the examples to illustrate the basic ideas involved.

Chen and Aihara [24] have used transient chaos in a network described by

$$\begin{aligned} x_i(t+1) &= kx_i(t) + \alpha\left( \sum_{j=1, j\neq i}^{N} w_{ij} f(x_j(t)) + I_i \right) - z(t)(f(x_i(t)) - I_0) \\ z(t+1) &= (1-\beta)z(t) \end{aligned} \tag{7}$$

where $z(t) \geq 0$ is the self-feedback term and $0 \leq \beta \leq 1$ is the damping term. These equations of motion can be obtained from the equations of motion similar course to (6) by time discretization. Due to the last term with $z(t)$ the system can be chaotic. The dynamics starts with a large value of $z(t)$ to ensure the existence of chaos, and then $z(t)$ is reduced according to (7) and the system can converge to the attractor. So, the main purpose of using chaos in this type of work is to overcome the difficulties associated with spurious states.

Another network with transient chaos for optimization problems has been proposed in [23]. The network is again a mapping, but now with a delay

$$x_i(t+1) = \sum_{j=1}^{N} w_{ij} f(x_j(t)) + I_i + g(x_i(t) - x_i(t-1)), \tag{8}$$

where $f$ is again a sigmoid function and $g(x) = axe^{-b|x|}$. In a stationary state the last term vanishes and therefore the fixed points of (8) coincide with that of usual Hopfield-Tank model (6).

The term $g$ makes the equations of motion more complex and ensures the chaotic wandering. Due to it the spurious stable states of the original Hopfield-Tank network in numerical simulations became unstable, though trajectory sometimes spent some time near them. To ensure convergence to the global minima, authors of [23] proposed special control scheme for parameters $a$ and $b$. The experiments showed that the network successfully solved the TSP problem.

Another modification of the Hopfield-Tank model with additive chaos or noise was studied in [44].

### 4.1.2. The Hopfield-type Network with Transient Chaos

This network was proposed in [55]. The idea is to replace a simple threshold neuron with an one-dimensional dynamical system $x_{t+1} = f(x_t, \mu)$—a neuron with its own internal dynamics.

The resulting network of mappings is controlled by the system's energy $E$ via the parameter $\mu$. If the energy is high, the dynamics of the mappings is chaotic, and when the energy becomes low, the trajectory of the mapping tends to one of two fixed points, close to ±1. Therefore, during transient phase the system's dynamics is chaotic, and close to the energy minimum it turns stable. The equations of motion for each "neuron" has the form

$$x_i(t+1) = f(x_i(t), E_i), \tag{9}$$

where

$$f(x, E) = \{K(E)(x + |E|)\} \bmod 2 - 1, \qquad K(E) = 2(1 + |E|)^{-1}. \tag{10}$$

(chaotic behavior for $|E| < 1$ and regular otherwise). The parameter $E$ is called the local energy

$$E_i(t) = \lambda \sum_{j=1}^{N} w_{ij} x_j(t).$$

The matrix $w$, as in the original Hopfield model, is constructed according to the Hebbian rule (5). The definition of the local energy $E_i$ involves the parameter $\lambda$, which describes neuron interactions: when $\lambda = 0$—the network splits into $N$ independent mappings, and when $\lambda = \infty$, one obtains the usual Hopfield model.

Numerical experiments show, that for large values of $\lambda$ the system behaves as the Hopfield model, but for moderate values (for the specific example considered in [55], $\lambda \leq 5$) there is a difference: almost all the false memories become unstable, and the system either remains chaotic for a very long time or converges to one of the stored images. So, in this case chaos helps in getting rid of false memories.

ve chaos or

e threshold
—a neuron

s energy $E$
iappings is
ping tends
phase the
rns stable.

(9)

$^{-1}$. (10)

eter $E$ is

ng to the
irameter
plits into
Hopfield

behaves
ble con-
emories
time or
getting

### 4.1.3. Coupled Map Lattice with Nonstationary Synchronous Clusters (CML)

These networks were described in [54]. They are based upon the globally coupled map lattice with the equations of motion

$$x_i(t+1) = (1-\epsilon_i)f_i(x_i(t)) + \frac{\epsilon_i}{N}\sum_{j=1}^{N} f_j(x_j(t)), \qquad (11)$$

$$f_i(x) = \alpha_i x^3 - \alpha_i x + x, \qquad 2 \le \alpha_i \le 4. \qquad (12)$$

There are two versions of this system: $\alpha$-version, when $\epsilon_i = \epsilon$, $\alpha_i$ are different, and $\epsilon$-version, when all $\alpha_i = \alpha$, $\epsilon_i$ are different. Bifurcation diagram for the mapping (12) resembles that for the logistic map. The encoding and decoding are very simple, all positive values $x > 0$ are associated with the value $+1$, and negative values with $-1$.

The idea of the CML-based pattern recognition arose due to the fact that in the CML (11) with $\alpha_i = \alpha$ and $\epsilon_i = \epsilon$ there is a domain of parameters $(\alpha, \epsilon)$, where the network splits into synchronized clusters corresponding to periodic behavior. For these parameters the system possesses rather high "information preservation," that is initial data strongly influence the subsequent behaviors, and there is high correlation between $x(t)$ and $x(0)$. For larger $\alpha$ or smaller $\epsilon$ in chaotic state this property is lost.

The idea is to define a local energy functional $E_i = -x_i \sum w_{ij} x_j$ and use chaos instead of annealing to break unwanted correlations. The matrix $w$ is formed according to the Hebbian rule using the patterns $\xi^{(k)}$ to be stored, $w_{ij} = \sum_k \xi_i^{(k)} \xi_j^{(k)}$. Then in the $\alpha$ -version, the dynamics of $x$ is augmented by the dynamics of $\alpha$:

$$\alpha_i(t+1) = \begin{cases} \alpha_i(t) + (\alpha_i(t) - \alpha_{\min}) \tanh(\beta E_i) & \text{every 16 steps,} \\ \alpha_i(t) & \text{otherwise.} \end{cases}$$

The value of $\alpha_{\min}$ corresponds to the clustering phase. This causes diminishing of $\alpha$, that is entering more ordered phase when the local energy is low enough.

Computer experiments show that this system works as an associative memory, and its memory capacity is about $0.18N$. Similar characteristics are obtained for the $\epsilon$-version of the algorithm.

Another version of pattern recognition with the transition chaos→order has been proposed in [67].

We note that this network does not use chaos for recognition. It uses chaos only for some short transient period although the normal dynamics of the system is non-stationary.

#### 4.1.4. What if Transient Never Ends or a Novelty Filter

There is always a possibility that chaotic transient states, under some combinations of input parameters, can give rise to a chaotic attractor or a very long transient period. Skarda and Freeman [85] supposed that such a state can mean "I don't know," that is, a neural network faces something that has not been learned (see also [55]). Such a state, like the inconsistency in bottom-up and top-down patterns of ART maps, can be used in principle as a novelty filter to initiate the learning phase. To our knowledge examples of networks with such filtering have not been published.

### 4.2. Chaos in Memory Search

When a trajectory moves along a chaotic attractor, it moves sequentially from one part to another. If we associate various parts of the attractor with different patterns, then the trajectory will wander between them. In principle, this wandering can be used for the recognition or association purposes: if a trajectory spends most of its time near one of the patterns, then the latter can be considered as "recognized", and if in the sequence of visited patterns there are stable combinations, those patterns may be considered as "associated" with one another. Note that sequences of patterns can be stored into a Hopfield-type networks. There is a possibility that chaos may help vary these combinations to learn new ones or to allow one pattern to participate in a number of associations simultaneously. These ideas have not been implemented completely, but some preliminary results have been obtained.

To study the linking of stored memories with one another, Tsuda [93] proposed a model that basically resembles a Hopfield-type network. Initially, patterns are stored by the Hebbian rule, but afterwards the connection matrix is dynamically modified. It has been shown that association of patterns with one another takes place, but there is not enough control over the process.

Another example of a model with associative dynamics in chaos has been proposed in [1] and references therein. The equations of motion had the form

$$
\begin{aligned}
x_i(t+1) &= k_f x_i(t) + \sum_{j=1}^{N} w_{ij} f(x_j(t) + y_j(t)) \\
y_i(t+1) &= k_r y_i(t) - \alpha f(x_j(t) + y_j(t)) + a_i,
\end{aligned}
$$

where $0 \leq x, y \leq 1$, $f(x) = 1/(1 + e^{-x/\epsilon})$, and patterns are stored in the network with the help of the Hebbian rule, $w_{ij} = \sum_k (2\xi_i^{(k)} - 1)(2\xi_j^{(k)} - 1)$.

The results of numerical experiments show that trajectories of the system visit neighborhoods of the stored patterns, and after a coarse-graining transformation some of the states reproduce the stored patterns. But the network does not give any definite pattern that can be considered as the resulting output. The presented results show some dependence on the behavior of initial conditions, but it is not clear, whether they correspond to (i) different attractors, (ii) transient processes or (iii) insufficiently long trajectories to establish a good statistics.

The chaotic wandering between stored patterns has also been studied in [26] and called "chaotic memory scanning."

A different idea for using chaotic regimes for recognition purposes has been proposed in [86, 87, 88, 89]. It has been observed, that patterns can be recognized by the shortest time required for full or phase synchronizations of an unknown pattern with known ones. The strength of this approach is that the procedure is general and can be applied to chaotic as well as non-chaotic neural networks. In this approach, one does not need fixed points or stationary final states for pattern recognition. The weak point is that the dynamics of known patterns needs to be run in parallel with that of the unknown pattern. It has also been found that virtual basins of attraction can be created around fixed points by root finders. What is good about this approach is that it utilizes the dense set of periodic orbits of chaotic NNs and thereby increases the capacity enormously. The work that needs to be done is to find ways to map patterns to state variables of the network.

## 4.3. Chaos Instead of a Fixed Point

As we have mentioned in the beginning of Section 4, the multiple-attractor and attractor manipulation networks can be implemented with attractors other than fixed points. One of the simplest ways to construct a multiple-attractor network is described in [10]. For attractor manipulation networks there is no such simple example. This may be because a complex bifurcation structure of chaotic attractors may cause severe difficulty in distinguishing attractors for different parameter values. Nonetheless, one of the most famous chaotic networks belongs to the attractor manipulation class. This is the Freeman's model of olfactory bulb.

The studies of olfactory system have been the primary goal of W. Freeman and his colleagues for several decades, see [99, 31, 32, 30, 85] and references therein. After some years of biological studies of olfactory bulb, they concluded that studying only the structure of neurons and their connections is not enough to understand the neural mechanisms responsible for olfaction. For this reason they developed a number of mathematical models for information processing in olfactory bulb. The dynamics of the models are in qualitative agreement with the experimental EEG measurements, and is chaotic.

The model is rather complex. Each "memory unit" is described by about 10 second-order differential equations, which describe the "specialization" of neurons within every unit. All equations have similar form $\ddot{x}_i + A\dot{x}_i + Bx_i = G_i$, where the right hand side terms are different for different types of neurons [99]. Some of the $G_i$ include input terms $X$, while others have delayed input (dependent on past values of $x$). There are neurons responsible for connections with other memory modules, and for them $G_i$ include the term $\sum K[i,j]Q(x[j])$, where $Q$ is a sigmoid-like function and $x[j]$ is similar "connection" neuron from the $j$-th memory module.

Information in this network is stored into the connections $K[i,j]$. They can take only two values, $K_{\min}$ and $K_{\max}$. Initially, all connections are set to $K_{\min}$, and to store pattern, for which modules $i$ and $j$ are both "active", the corresponding $K[i,j] = K[j,i]$ is set to $K_{\max}$ (Hebbian rule).

The network works as follows: When there are no external signals, the network oscillates chaotically on an attractor. If an external stimulus is presented to the network, the system stabilizes onto individual parts of the attractor. To explain the main idea, one can consider a multilobed attractor such as the Lorenz attractor which has two lobes as the network's "basal activity state." By applying an input, the network can be stabilized onto one of the two lobes. The dynamics would still be chaotic but confined to one of the lobes only. Then the presence of the trajectory at this lobe can be decoded into the network output, e.g., with the help of local time averages.

Other chaotic models of olfactory bulb also exist, e.g., [6], though they are not in very good agreement with biological experiments.

### 4.4. Recurrent Neural Networks as Generators of Chaos

There are works that consider neural networks just as models for some biological phenomena or convenient forms of dynamical systems. There is no question of computation, approximation or associative memory, e.g., [34, 2, 3, 61, 70] and some other. This class of neural networks falls out of our scope. Indeed, it is not hard to construct a neural network, e.g., in a form of recurrent perceptron (Section 3), with chaotic dynamics. Such works do not answer the question about the role of chaos in information processing. Nonetheless, if there is such a role, those neural networks may serve as a convenient generators of chaos [75].

## 4.5. What's Wrong with Chaotic Networks?

We have mentioned only some of the works related to chaotic neural networks just to illustrate major directions of studies. A common feature of all chaotic networks is that they are not used in practical applications. The only exception is the experiment described in [100]. In contrast, multilayer perceptrons are widely used, while chaotic networks have remained only as objects of theoretical studies for about 15 years. What is the reason for this? From our point of view, the reason is the way in which neural networks are currently used. Current uses of neural networks may be called "isolated computations." A neural network's task is only to generate definite, always the same output for a given input. Chaotic dynamics, which is unstable, can only make such a computation unreliable. Therefore, there is no apparent room for chaos in such a scheme.

In contrast with such neural networks, the brain always works as a part of the body. It is involved in continuous processing of information coming from the outside world, and it guides the body to perform actions that change the environment. Therefore, the brain operates as a part of a closed loop: brain—actions—world—sensing—brain. It is possible that accounting for the embodiment of the brain can explain the advantages of operating in chaotic mode. Also chaos may just emerge as a consequence of positive feedback in the aforementioned loop.

Note that the problem of "embodiment of intelligence" has been intensively discussed in Artificial Intelligence during the last 15 years, see [73] for a review. The related approach called "behavior based robotics" or "embodied cognitive science" led to a number of efficient practical solutions and new theoretical concepts. Moreover, with examples of small robots controlled by a rather simple neural networks, it has been shown that closing the loop through the world may lead to a very complex, probably chaotic behaviors [73].

Another natural source of complex temporal behaviors may be special implementations of neural networks: using ensembles of coupled oscillators to perform computation, e.g., [52, 48, 69]. In Section 7 we shall discuss this point in more details.

## 5. Closing the Loop: Chaos in a Combination of Controlling Neural Network and Controlled System

As we said in Sections 3, 4, one of the simplest ways to obtain chaos is to take a functional network, say a feed-forward multilayer perceptron approximating equations of motion of a chaotic system, and feed its output back to the input. The result is a chaotic dynamical system. However, it does not perform useful information processing. A useful task can arise if we place between output and input of a network a system that needs controlling (often called the "plant"). In this section we present some rather simple examples that demonstrate how chaos can appear in a controller coupled with a controlled system. Chaotic signals can be registered at any part of the combined controller-plant system. This notion may partially explain the appearance of chaos in the activities of the real brain.

Here we are referring only to chaos emerging in the course of information processing. In the next section, we will consider the importance of chaos in the learning process of a controller-plant system.

Let us consider a dynamical system

$$\dot{x} = \lambda x + f, \qquad \lambda > 0. \tag{13}$$

For $f = 0$ it has an unstable equilibrium point $x = 0$. Suppose that we can control this system by applying the "force" $f$ at the discrete moments of time $t_k = \tau k$. Then the force remains the same until the next switching. The absolute value $|f| = f_0$ always remains the same, we can only change its direction. Our goal is to keep the trajectory in the vicinity of the point $x = 0$. So, at every $t_k$ we know $x(t_k)$, and we have to make a decision regarding the direction at which the force should be applied.

This is a simple problem. Let us write $x_k = x(t_k)$. Since $f_k = f(x_k)$ remains constant until $t_{k+1}$, (13) gives: $x_{k+1} = e^{\lambda\tau} x_k + (e^{\lambda\tau} - 1)\frac{f_k}{\lambda}$. It is easy

be special
oscillators
iscuss this

is to take
ximating
the input.
rm useful
utput and
"plant").
trate how
Chaotic
t system.
ivities of

ormation
chaos in

(13)

t we can
s of time
ng. The
ange its
t $x = 0$.
ding the

$= f(x_k)$
t is easy

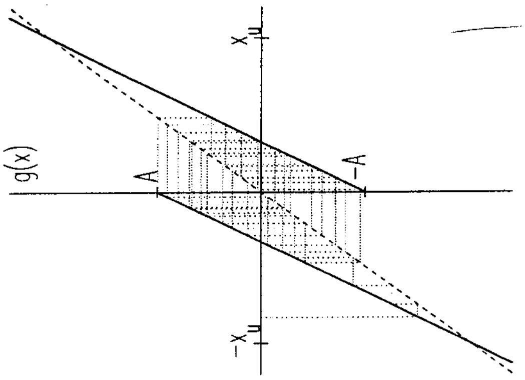

**Figure 1.** Mapping resulting from discrete control of unstable fixed point and an example trajectory.

to check that the choice $f_k = -f_0\mathrm{sgn}(x)$ solves the problem, and we obtain the following one-dimensional mapping

$$x_{k+1} = g(x_k), \quad g(x) = \begin{cases} e^{\lambda\tau}x - A, & x \geq 0 \\ e^{\lambda\tau}x + A, & x < 0 \end{cases}, \quad A = \frac{\left(e^{\lambda\tau} - 1\right) f_0}{\lambda}. \quad (14)$$

The plot of $g(x)$ is shown in Fig. 1. It can be seen that the trajectory remains near the unstable point provided $|x(0)| < f_0/\lambda$. Since $dg(x)/dx = e^{\lambda\tau} > 1$, a chaotic attractor is born with the Lyapunov exponent equal to $\lambda$.

The control of the system can be performed by a "network" with a single threshold neuron that receives input $x_k$ and generates the signal ±1 , showing the direction of the force. As the attractor is chaotic, the sequence of the neuron outputs will look random. The source of this randomness is a discrete control of an unstable equilibrium.

This example explains the main idea, but it has two obvious shortcomings: (i) there is no learning, and (ii) there is no true need for the use of a neural network. Let us consider more complex examples related with discrete controls. Numerous examples of such problems can be found, e.g., in the literature on machine learning [65].

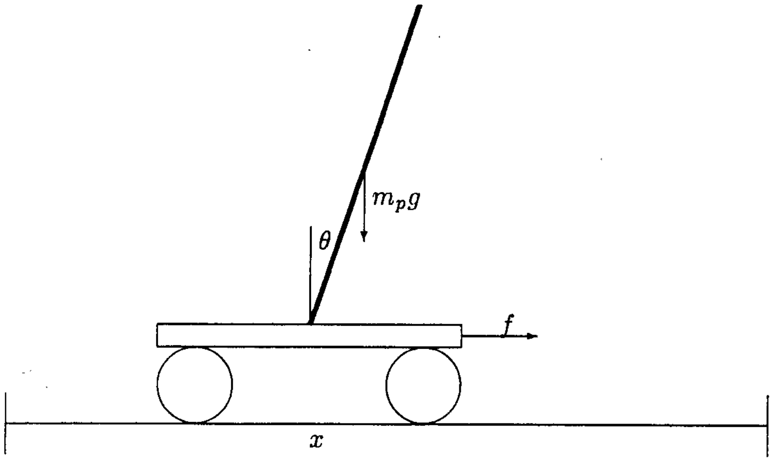

**Figure 2**. The cart-pole balancing task. Controller should choose the proper direction for $f$ after each time interval $\tau$ such that the angle $\theta$ for the pole will remain within $[-\theta_{\max}, \theta_{\max}]$, and the cart never hits the ends of the track, $-x_{\max} < x < x_{\max}$. In the beginning the cart is positioned at the middle of the track $x = 0$ and the pole is set at some angle $\theta_0$ which is within the admissible limits.

The easiest generalization is an inverted pendulum that need to be kept close to its highest point. However, it can be shown that this problem reduces to the above example — the unstable manifold of the fixed point of the inverted pendulum is one-dimensional.

A more interesting problem is that of cart-pole balancing, one of the well-known benchmark problems in machine learning [64, 12, 46]. There is a cart that can move along the line from $-x_{\max}$ to $x_{\max}$. A pole is attached to the cart with one end such that it can rotate in the vertical plane parallel to the line of motion of the cart. If the pole is set almost vertical, while falling, it moves the cart. If one pushes the cart, the push affects the pole dynamics as well. That is, by moving the cart, one can change the position of the pole. The state of the cart-pole system is determined by $x$ (coordinate of the cart), $\dot{x}$ (velocity of the cart), $\theta$ (inclination angle of the pole from the vertical), and $\dot{\theta}$ (angular speed of the pole), see Fig. 2. The task of control is as follows: After every time interval $\tau$, the controller receives the values of the cart-pole state variables $x$, $\dot{x}$, $\theta$ , $\dot{\theta}$. The controller can apply a force equal to $\pm f$ to the cart that acts during the next $\tau$-interval. The task is to keep the angle $\theta$ within the limits $[-\theta_{\max}, \theta_{\max}]$, and the position of the cart $x$ within $[-x_{\max}, x_{\max}]$.

At the first glance this problem seems equivalent to the control of the inverted pendulum. However, if one tries to apply the same control algorithm, the cart very soon hits the end of the track. So, it is necessary to control the cart position as well, but there is no "obvious" control policy that should be learned. In the next section, we consider the details of such learning. Here, we only mention that the policy for neural network controller has been obtained [77], and the resulting regime is chaotic with one positive Lyapunov exponent. We registered (see Fig. 3) activity patterns from several neurons of the controller and calculated their autocorrelation functions. They look chaotic, though the neural network architecture (a Kohonen-type network) does not possess any complex dynamics. Similar results were obtained for another model control task — stabilization of an unstable chemical equilibrium [77].

So, chaos may arise in complex feedback loops where a neural network plays the role of a learning controller. One may ask: can chaos be useful for the information processing or for learning? The results of our models show that the answer is yes, but to explain it we need to describe the idea of reinforcement learning.

# 6. Chaos and Reinforcement Learning

## 6.1. What is Reinforcement Learning?

In most books on neural networks two types of learning are considered, supervised and unsupervised learning. If for every training input $X$ the correct output $Y$ is known, and this knowledge can be used for updating the weights of the network, the learning is called supervised learning or learning with a teacher. Examples of supervised learning include all functional and most dynamical networks. If learning proceeds without a teacher that can provide the correcting signals, then it is called unsupervised learning. For examples of unsupervised learning see, for example, Kohonen and ART networks. In addition to these two types of learning, an intermediate situation is possible when some evaluation of the network performance can be done, but the correct answer is unknown. Usually, such an evaluation of performance comes in the form of a scalar 'reward' $r$ : for success $r > 0$, for failure $r < 0$, and for neutral case $r = 0$. The corresponding learning is called reinforcement learning or learning with a critic. This type of learning is rarely used in traditional neural networks. However, it is very valuable in situations where one knows the