Artificial Intelligence Review **15:** 165–187, 2001.

On Chaos and Neural Networks: The Backpropagation Paradigm

K. BERTELS[1], L. NEUBERG[1], S. VASSILIADIS[2] and D.G. PECHANEK[3]
[1]*University of Namur, Dept. of Business Administration, Rempart de la Vierge 8, 5000 Namur, Belgium;* [2]*T.U. Delft, Electrical Engineering Department, Mekelweg 4, 2628 CD Delft, The Netherlands;* [3]*IBM Microelectronics Division, Research Triangle Park, North Carolina 27709, USA*

Abstract. In training feed-forward neural networks using the backpropagation algorithm, a sensitivity to the values of the parameters of the algorithm has been observed. In particular, it has been observed that this sensitivity with respect to the values of the parameters, such as the learning rate, plays an important role in the final outcome. In this tutorial paper, we will look at neural networks from a dynamical systems point of view and examine its properties. To this purpose, we collect results regarding chaos theory as well as the backpropagation algorithm and establish a relationship between them. We study in detail as an example the learning of the exclusive OR, an elementary Boolean function. The following conclusions hold for our XOR neural network: no chaos appears for learning rates lower than 5, when chaos occurs, it disappears as learning progresses. For non-chaotic learning rates, the network learns faster than for other learning rates for which chaos occurs.

1. Introduction

In this tutorial paper, we study the appearance of chaos during the learning process of a specific kind of neural networks, namely error backpropagation networks. We try to establish under what conditions this chaos occurs and also what role it plays in the learning process. Before actually analysing the learning process, we will first extensively introduce the reader to the basic notions of chaos theory and explain what mathematical tools are available to characterise it.

Before starting, we want to emphasise the following limitations of the results as discussed in this paper. The first limitation is that we are only dealing with what could be considered a toy-problem. However, neural networks are always seen as black boxes. When we want to study what goes on inside this box, it is a sound scientific principle to reduce the complexity of your phenomenon as much as you can without eliminating those properties that you want to study.[1]

The second limitation refers to the values of the learning rates that were used in the simulations. It could be argued that learning rates larger than 1 are highly unusual and not of any practical relevance. However, it is only after studying extensively the issue of chaos, that we know whether chaos occurs for all kinds of learning rates and what its influence is on the learning process and whether it can be controlled. These and other issues need to be addressed before one can advance any kind of conclusion. The objective of this paper is to try to answer these questions.

The paper is organised as follows. We first briefly describe the algorithm in section 2. We consequently introduce the basic notions of chaos theory (section 3) and illustrate their essence using the Verhulst equation, a simple one dimensional dynamical system (section 4). In section 5, we discuss how chaos can be characterised numerically. In section 6, we then turn to the backpropagation algorithm and graphically as well as numerically reveal the presence of chaos and describe how the order of chaos evolves during learning.

2. The Backpropagation Algorithm

The backpropagation algorithm is a well known learning algorithm for feed-forward neural networks. We will therefore restrict our discussion of the algorithm to a definition of the equations used for learning.

The following equations allow to compute the output of any node (hidden or output) (McClelland and Rumelhart 1988):

$$A_{pj} = \sum_{i=1}^{m} W_{ji} O_{pi} - U_j \tag{1}$$

$$O_{pj} = f_j(A_{pj}) = \frac{1}{1 + e^{-A_{pj}/t}} \tag{2}$$

where A_{pj} is the activation for the input pattern p presented to node j, W_{ji} is the weight of the connections between nodes i and j, O_{pi} is the output of node i presented to node j as input and U_j is the threshold of node j. The transfer function O_{pj} is also called the sigmoid transfer function. The parameter t is called the temperature. The smaller the value, the more the transfer function looks like a step-function and the higher the temperature, the flatter it becomes. The following equations are used to train the network:

- To compute the necessary weight modifications, a rule, denoted as the **delta rule**, is used which is as follows:

$$\Delta_p W_{ji} = \beta \delta_{pj} O_{pi} \tag{3}$$

  where O_{pi} is the value of the ith incoming connection, β is the learning rate and δ_{pj} is the error at the jth node for the input pattern p. The error δ_{pj} needs to be calculated separately for the hidden and the output nodes.

- For the output nodes, the error can be computed in the following way:

$$\delta_{pj} = (t_{pj} - O_{pj})f'(A_{pj}) \tag{4}$$

where the term $(t_{pj} - O_{pj})$ computes the difference between the expected t_{pj} and actual output O_{pj} of the network. This error is then multiplied by the first derivative of the transfer function. This allows to sanction nodes that generate an uncertain output (an activation value close to 0) because there the first derivative will be high and consequently the weight change will be larger. The inverse is true for nodes that generate large activation values. There the first derivative will be close to zero and consequently will result in small weight changes.

- For the hidden nodes, the error can be computed in the following way:

$$\delta_{pj} = (\sum_{k=1}^{n} \delta_{pk} W_{kj})f'(A_{pj}) \tag{5}$$

where $\delta_{pk} W_{kj}$ represents the error of the connected neurons of the above layer, multiplied by the corresponding weights, which is propagated throughout the network. For exactly the same reasons as mentioned above, the first derivative of the transfer function is included.

One of the problems associated with the backpropagation algorithm is its parameterisation. Beforehand, the value of a number of parameters need to be specified. It has been found that very small variations in these values can make the difference between good, average or bad performance (Weiss and Kulikowski 1991). This also implies that one can never be sure to have found the optimal solution. Furthermore, the backpropagation algorithm can converge in a local minimum or oscillate between two (or more) different solutions (Aleksander and Morton 1991). Because, the rule for weight-modifications bears some structural resemblance with a well known chaotic equation, a possible explanation for the hypersensitive and sometimes problematic behaviour of the backpropagation algorithm may be found in chaos theory.

We investigate the backpropagation algorithm from this perspective and reveal the presence and evolution of chaos during learning. We first introduce the basic notions of chaos theory (section 3) and illustrate their essence using the Verhulst equation, a simple one dimensional dynamical system (section 4). Consequently, in section 5, we discuss how chaos can be characterised numerically. In section 6, we then turn to the backpropagation algorithm and graphically as well as numerically reveal the presence of chaos and describe how the order of chaos evolves during learning.

3. Definition of Chaos

Only dynamical systems can exhibit chaotic behaviour. In (Broer et al. 1991), a dynamical system is defined as a triple (M, T, ϕ), consisting of a phase space, a time set T and an evolution operator ϕ: $M \times T \rightarrow M$ where the phase space **M** represents all the possible states of a system at any particular moment, the time set **T** is the time lag in which the system is defined and the evolution operator ϕ specifies deterministically from any give state the future of the system.

Furthermore, chaotic phenomena can appear in dynamical systems which have in addition the following characteristics (Feigenbaum 1980):

- non-linear: only non-linear equations exhibit the kind of dynamical behaviour where phenomena such as chaos arise. In the linear system, one merely has superpositions.
- recursive: the evolution operator ϕ generates a time series representing the evolution of the system over time. If **f** is the operator and the starting point is x_0 (to take a one-dimensional example), then $x_1 = f(x_0)$, $x_2 = f(x_1)$, $x_3 = f(x_2)$, $\ldots$, $x_{n+1} = f(x_n)$. The nth element, x_n then is $f^n(x_0)^2$ and the order of the corresponding polynomial is 2^{n-1}.

Feigenbaum (1980) also points out that this complex behaviour is mainly generated by its recursive nature, rather than by the specific operation performed by the function.

There is very little agreement on what constitute the properties of chaotic systems. However, one element is always considered to be most characteristic, namely sensitivity to initial conditions (SIC) (Broer et al. 1991). In general, this means that the slightest difference in initial conditions of a dynamical system will cause the system to take a completely different orbit towards a potentially different final state. We will primarily focus on SIC to describe the essence of chaos. We will also observe that this SIC covers a multitude of other phenomena for which we can compute a number of statistics. We also limit ourselves to the period doubling bifurcation (which will be explained later) as a road to chaos.[3]

To determine whether or not there is sensitivity to initial conditions, the concept of strange attractors is very important. Attractors can be defined as a final state to which all trajectories converge. We might say that it constitutes the final solution of the non-linear differential equation in which the system will remain. We distinguish between regular attractors, corresponding to periodic and quasi-periodic solutions (or limit cycles) and irregular or strange attractors corresponding to aperiodic solutions. The nature of this attractor now determines whether or not there is sensitivity to initial conditions. The periodicity of the limit cycle represents the number of iterations the system needs to reproduce itself. Consequently, by observing the system for a certain lapse of time, we will find that the system behaves in a regular and orderly way. From any given state, we can consequently predict any future state of the system. What is now the implication of this with respect to the initial conditions? When the attractor is periodic, the final state of the system will always be the attractor, irrespective of the initial conditions. Consequently, the information that is represented in the initial values of the system is destroyed as any nearby initial situation will always lead to the same final state.

When, on the contrary, the attractor is aperiodic, a totally different behaviour of the system will be observed. Where in the periodic case, the system was perfectly predictable and orderly in its behaviour, in the aperiodic case, the system behaves in an ostensibly random way. The aperiodicity of the attractor has as major implication that during whatever time length the system is observed, it will be virtually impossible to detect some kind of sequence that will permit to make any kind of prediction. In the periodic case, the system exhibits a high degree of self-similarity: after a certain number of iterations (which depends on the periodicity), the system reproduces itself. In the aperiodic case, this self similarity diminishes and finally disappears after a number of iterations. This implies that it severely limits the possibility to predict from any given moment a future state of the system. Periodic attractors are also called simple attractors and aperiodic ones are called strange attractors (Broer and Takens 1992).

4. The Verhulst Equation: From Bifurcation to Chaos

In the previous section, we have given a description of chaos and have introduced a number of concepts. We will now illustrate these concepts by means of the Verhulst equation which is a classical example of the bifurcation cascade process as a route to chaos.

The Verhulst Equation is cited in almost any book on chaotic systems as a classical example of a non-linear dynamical system (see for instance Bergé et al. (1992) and Lorenz (1989)). It is especially interesting to discuss its chaotic properties because of its unidimensionality. The equation has the following quadratic structure:

$$x_{t+1} = \alpha x_t(1 - x_t) \tag{6}$$

The equation was introduced by the Belgian professor Verhulst to explain the population growth in terms of its birth and death rate. x_t represents the normalised population at time t and α is a parameter. A steady state solution can be found by solving

$$x^*_{t+1} = \alpha x^*_t(1 - x^*_t) \tag{7}$$

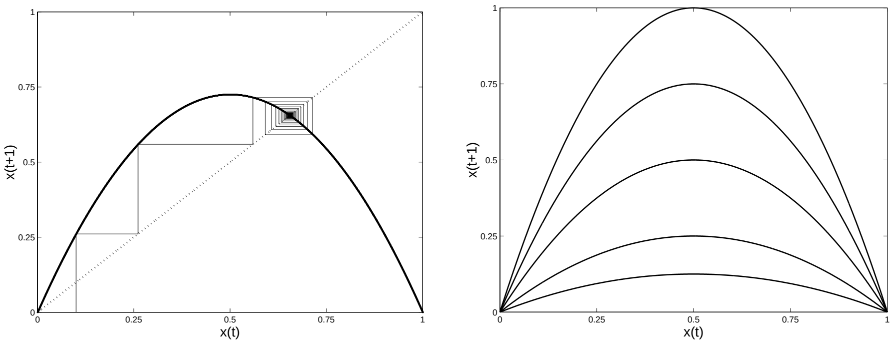

Figure 1. Graphical solution procedure for α = 2.9 and mapping for α = 0.5, 1, 2, 3, 4.

which has two solutions, the trivial solution $x^* = 0$ and $x^* = 1 - 1/\alpha$. One might expect consequently, that whatever the initial value of x_0, we would always end up in either of these solutions. However, the dynamics of this equation are more complex and need to be investigated further. A graphical solution procedure for the logistic function is given in Figure 1 (left figure) and the mapping for different values of α (right figure). One should interpret the different graphs in the following way. We start at the x-axis for any given value of x, e.g. x^*. We proceed vertically to the function being depicted, $f(x^*)$. From $f(x^*)$ we go horizontally to the 45°-line. Along this line $x_t = x_{t+1}$ so that the point on this line gives the starting point for the next iteration. The next value can thus be found by proceeding again vertically to the function, yielding $f(f(x^*))$, denoted by $f^2(x^*)$. This can be repeated for any number of iterations until the obtained solution does not change anymore. A solution of a dynamical system is said to be stable when, irrespective of the initial situation, the system will always end up in the same particular state. By contrast, a solution is said to be unstable whenever a slight deviation from the initial situation will make the system evolve towards another solution.

For discrete time dynamical systems, the asymptotic stability of the fixed point x^* depends on whether the slope of f, evaluated at the fixed point, lies within the unit circle, i.e., whether $|\delta f(x^*)/\delta x| = |\lambda| < 1$ and the equilibrium condition becomes $\lambda = 1$. As can be seen from Figure 1 (right figure), the slope of the graph at the fixed point increases as α increases.

Consequently, there will be a value for α for which the fixed point x^* becomes unstable, giving rise to a bifurcation. The slope of the graph of equation 6 is

$$\frac{df(x_t)}{dx_{t\,|x_t=x^*}} = \alpha(1-2x) = 2-\alpha$$

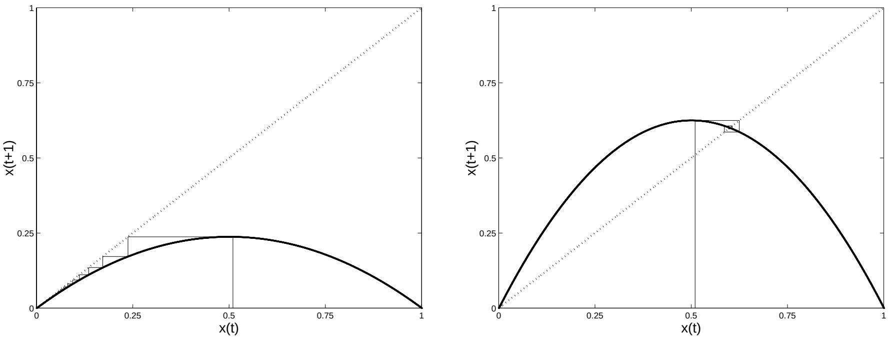

Figure 2. Graphical solution procedure for $\alpha = 0.95$ and 2.5.

For $\alpha = 3$, the stable fixed point becomes unstable and a new stable fixed point of period-2 occurs.

In function of the value of α, the system may have the following solutions:

- **the origin**: from Figure 2 (left figure), for $\alpha = 0.95$, we can easily see that any value for x will, after a limited number of iterations, converge to the origin. The absolute value of the first derivative of this fixed point is zero and thus less than one, implying its stability.[4] This fixed point will remain stable and thus attracts all other values.
- **a fixed point** (see Figure 2 (right figure) for $\alpha = 2.5$): for $1 < \alpha < 3$, the origin is still a fixed point, but the absolute value of the slope of f is at this point equal to α, which is larger than one. Consequently, this fixed point is an unstable equilibrium point. That means that starting at 0, the system will remain at this value, but for any x_0-value differing marginally from 0, the system will move away from this unstable, fixed point and will converge to the second fixed point, $x^* = 1 - 1/\alpha$. The absolute value of the slope at this fixed point is now less than 1 and will be the new attractor of the system. In other words, x^* is an asymptotic fixed point to which the system will converge for any initial value.
- **a limit cycle**: Figure 3 (left figure) shows the evolution of the system for $\alpha = 3.4$. In addition to the curve $x_t - x_{t+1}$, we have now also added the period-2 curve relating x_t with x_{t+2}. At $\alpha = 3$, the previous fixed, stable point now becomes unstable and a bifurcation arises. For $\alpha > 3$, the system needs two iterations to reproduce itself, implying the existence of two fixed stable points. When a system's solution oscillates between two or more values, it is said to have entered a limit cycle (in this case a period 2 limit cycle). This means that, for instance for the period 2 case, the system will eventually produce the sequence of values x_1^*, x_2^*, x_1^*, x_2^*, ... As α increases, the two stable fixed points will become unstable (the

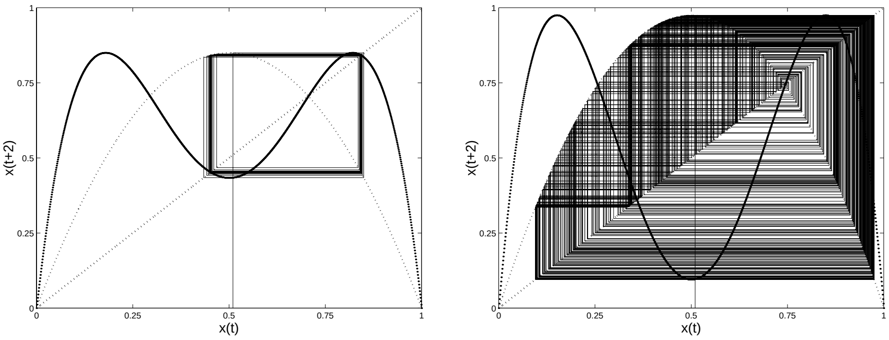

Figure 3. Graphical solution procedure for $\alpha = 3.4$ and 3.9.

absolute values of the first derivatives become = 1), giving rise to another bifurcation in each of the fixed points. A new limit cycle of period 4 will emerge. The limit cycle then becomes x_1^{**}, x_2^{**}, x_3^{**}, x_4^{**}, x_1^{**}, x_2^{**}, x_3^{**}, x_4^{**}, ... For consecutive increases of α, the periodicity of the limit cycles will then become 8, 16, 32, ...

It was shown by Feigenbaum (1980) that this period doubling cycle will tend to infinity at a value for $\alpha \cong 3.56999 \ldots$, called the critical value. From then on, phenomena other than infinite periodicities may occur.

- **chaotic** (Figure 3 (right figure) for $\alpha = 3.9$): as soon as α exceeds this critical value, it is said that the system enters a chaotic state where a number of other phenomena are observed. A quasi infinite number of fixed points leads to aperiodic behaviour. Also, limit cycles of unpair periodicity appear in regions called *windows* where the number of fixed points suddenly decreases. Examples of these low-order periodicities are the period 3 limit cycle for $\alpha = 3.839$, the period 5 limit cycle at $\alpha = 3.74$, etc. ... The fractal nature of the Verhulst equation is also remarkable in this region.[5] The asymptotic periodic orbits of period 6 and 12 (for α equal to respectively 3.845 and 3.849) result in the period doubling of period 3 similar to the doubling of the pair periods. The parameter regime for $\alpha_c < \alpha < 4$ is called the chaotic regime. The simultaneous presence of periodic cycles of order k and of aperiodic cycles is called *deterministic chaos*.

Another way of representing these different solutions for distinct values of α is by means of the Feigenbaum bifurcation diagram (see Figure 4) which shows the asymptotic behaviour of the system. On the x-axis, we plot the different values of α and on the y-axis the value of the system's solution after a number of iterations. This diagram clearly reveals the period doubling. From Figure 4, it can be observed that for values of α between 1 and 3,

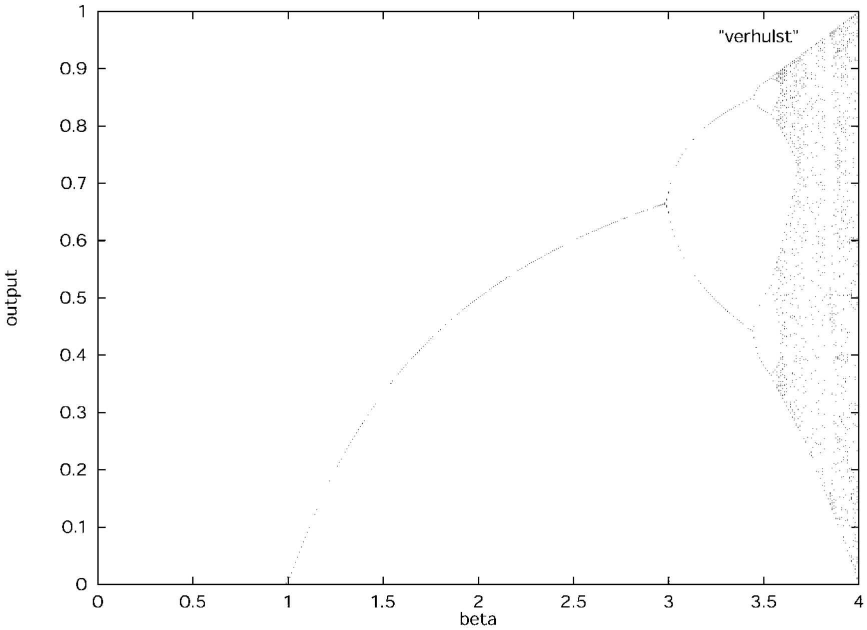

Figure 4. Bifurcation diagram for the Logistic equation.

always one end-state will be found. When α reaches the value 3, the system's state becomes unstable and then moves into two possible new states. This bifurcation process goes on till α reaches the value 3.56999 after which the chaotic behaviour emerges.

In the next section, we present a number of descriptive and numerical tools that allow us to detect whether or not a particular system exhibits chaotic behaviour. These tools are the phase space, the embedding dimension, the correlation dimension, spectral analysis and the Kolmogorov Entropy.[6] They relate to chaos in the following way. Sensitivity to initial conditions means that the system has an aperiodic attractor to which all final states converge. By using phase spaces, we can often describe some topological properties of the attractor and the periodicity or aperiodicity of the attractor can be revealed by applying a Fourier transformation. In the former case, a particular frequency will be found and in the latter case a continuous broadband spectrum will be observed.[7] Finally, the fact that all solutions are still located on this aperiodic attractor, means that there is some kind of limitation to the final outcome, which will be reflected in the finite Kolmogorov Entropy and in the correlation dimension.

5. Descriptive and Numerical Tools to Study Chaos

Besides numerical experiments, the sensitivity to initial conditions can also be revealed by means of other tools that allow us to shed more light on the underlying chaotic process. These tools are presented in this section and will be explained from a ‘user’ point of view.

5.1. *Phase space*

One of the ways in which one may study the dynamical behaviour of a system and detect the presence of attractors is by means of a phase space diagram which is a graphical representation of the evolution of a system towards a solution. Each point of that space then determines unequivocally the state of the system at a given time. The axes may represent the position and speed of the system and the corresponding space trajectory or orbit is the evolution of the system.

5.2. *Embedding dimension*

When studying a dynamical system, we may not always have an idea of what the system looks like nor which variables influence its behaviour. However, quite often one disposes of at least a one dimensional observation. Takens has developed a technique that allows to reconstruct the properties of the dynamical system under study solely on the basis of this one dimensional observation (Takens 1981). We thus create an artificial dynamical system which, as Takens proved, has the same topological properties as the original system. This boils down to finding an appropriate dimension in which to represent the system, called the embedding dimension.

One proceeds in the following way. We assume that the time series is produced by a set of deterministic equations such as $x^i_{t+1} = f_i(x_t)$ where $x \in \Re^n$, $i = 1, \dots, n$. As we said previously, we do not know the structural form of the system nor its true dimension (n). Nor are we certain that the time series contains the real x^i_t-values, so we represent the observed values by $\overline{x}^i_t$ and we assume that the following relation holds: $\overline{x}^i_t = h(x_t)$ which means that the observed values in some way depend on the true value vector x^i_t. We now take the last element of the time series and combine it with its **m** predecessors to form a vector of length m, $X^m_t = (\overline{x}^j_t, \overline{x}^j_{t-1}, \dots, \overline{x}^j_{t-m+1})$. We do this for every element in the time series, eliminating of course the first $m - 1$ elements. We thus obtain a series of m-dimensional vectors X^m_j. The length m is called the embedding dimension and each vector is called a m-history, describing a point in m-dimensional space. Takens proved that this artificially reconstructed object is topologically equivalent to the real system if the following conditions hold:

- the variables x^i of the true system are located on the attractor.
- the functions $f_i(x)$ and $h_i(x)$ are smooth functions.
- $m > 2n - 1$.

This is a very important result because it allows us to calculate three other very important measures, namely the correlation dimension, the Kolmogorov Entropy and Lyapunov exponents.

In order to illustrate this somewhat abstract procedure, we apply it to the well known chaotic system, the Lorenz System which is composed of the following set of coupled non linear differential equations:

$$\begin{cases} \frac{dX}{dt} = a(Y - X) \\ \frac{dY}{dt} = bX - Y - XZ \\ \frac{dZ}{dt} = XY - cZ \end{cases}$$

For our purposes, we are not so much interested in the exact semantics of the variables but rather in the overall behaviour of the system. We computed 15,000 values (where $dt = 0.05$) for X, Y and Z using the following parameter values: $a = 10$, $b = 10$ and $c = -4$. When reconstructing the attractor, we assume that we only dispose of a one dimensional observation of the system, namely on the X-variable. To simplify things a bit and using the fact that the Lorenz system is of dimension 3, we fix the embedding dimension to 3.

In order to avoid autocorrelation (which would merely generate an attractor where all points are centered around the diagonal (plane)), we only use 1 out of every 5 observations. Some of the original values and their constructed vectors are given in Table 1. If we now plot both the original X, Y and Z-values (see Figure 5(a)) and the artificially reconstructed system X, Y' and Z'-values (see Figure 5(b)), we can indeed see that topologically the two are equivalent.

5.3. *Correlation dimension*

As we said previously, a fundamental characteristic of chaotic systems is its sensitivity to initial conditions which basically means that two orbits starting from very close initial points, soon become uncorrelated. However, this does not imply that the dynamical system can be at any position in phase space. Every orbit will always be located at an attractor and therefore all the points on the attractor are spatially related. Grassberger and Procaccia have developed a technique which measures the degree of spatial correlation between 2 points in phase space (Grassberger and Procaccia 1983).

It is computed in the following way. The starting point is the artificial dynamical system as described in the previous section. Two points $\overline{x}_t^i$ and $\overline{x}_t^j$

Table 1. Creation of the embedding vector of dimension 3

Observation I	Observation II	Vector (1)	Vector (2)	Vector (3)
0.034168	0.347151	0.034168	0.087486	0.274364
0.038980	0.439782	0.898230	2.947274	–
0.046739	0.557647		–	–
0.057089	0.707582			
0.070430	**0.898230**			
0.087486	1.140478			
0.109239	1.447933			
0.136953	1.837410			
0.172245	2.329264			
0.217174	**2.947274**			
0.274364	3.717394			

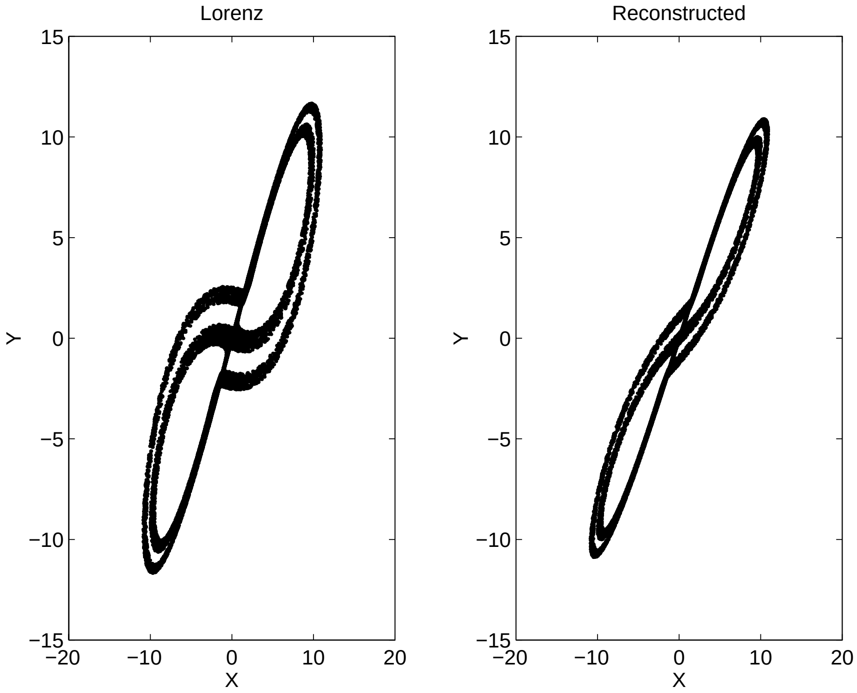

Figure 5. Reconstruction of the Lorenz attractor.

are said to be spatially correlated if the Euclidian distance is less than the radius **r** of an *m*-dimensional sphere centered on one of the two points, or put more formally:

$$||\overline{x}_t^i - \overline{x}_t^j|| < r$$

The spatial correlation between all points on the attractor is then measured by:

$$C(r, m) = \lim_{T\to\infty} \frac{1}{T^2} \sum_{i,j=1}^{T} H(r - ||\overline{x}_t^i(\overline{x}_t^j)||) \tag{8}$$

where *T* = length of the series of the constructed *m*-vectors x_t^m, ||. || is the Euclidean norm and

$$H = H(y) = \begin{cases} 1 \text{ if } y > 0 \\ 0 \text{ if otherwise} \end{cases}$$

C(*r*,*m*) is called the correlation integral from which we can compute the correlation dimension in the following way:

$$D^c(m) = \lim_{r\to\infty} \frac{\ln[C(r, m)]}{\ln[r]}^8$$

The value of *C*(*r*,*m*), and consequently of $D^c(m)$, depends on the length of the *m*-vector. It was shown, however, by Grassberger and Procaccia that for deterministic systems the correlation dimension converges to a fixed value, independent of the embedding dimension used for calculation. We therefore will compute the correlation integral *C*(*r*,*m*) for different embedding dimensions and we can then plot, on a log-log scale, the obtained values for *C*(*r*,*m*) given a particular radius.

An example of these computations for the Lorenz system is given in Figure 6. Each graph connects the different Correlation integrals with the different radii. What we can observe is that the slopes of these lines do not change anymore, even though the embedding dimension changes. It is this ‘constant’ slope which yields the value of the $D^c(m)$. In the case of pure stochastic systems, the slope will permanently increase in function of the embedding dimension.

5.4. *Spectral analysis*

A well known numerical tool is spectral analysis which allows us to analyse the periodicity of dynamical systems. The technique used is called the Fourier Transform.

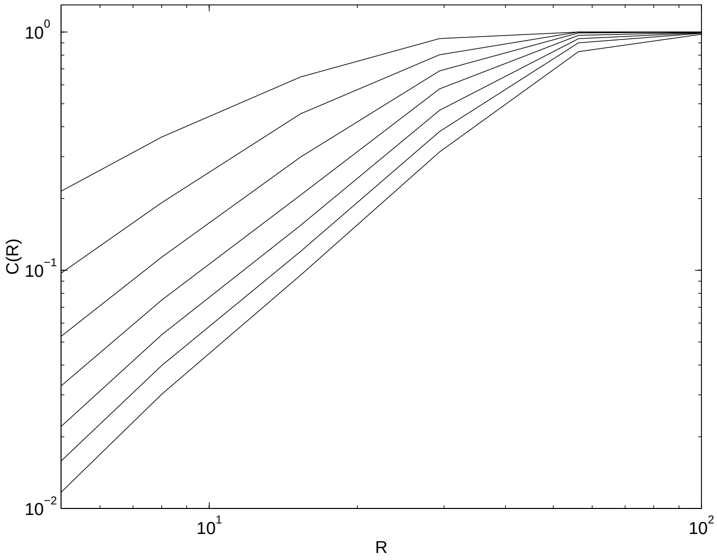

Figure 6. Function of the Correlation integral $C(r,m)$ and the radii R.

When a time series does not converge, it can periodically oscillate, or may appear to be random. The Fourier analysis allows us to distinguish between periodic or quasi-periodic series and random ones. However, we must be careful with this distinction. It may very well be that a time series appears random without being so. This may be caused by the fact that, for instance, the length of the observation interval is smaller than the periodicity of the system.

In the discrete case, we have two possible situations. The first one being either a periodic or quasi-periodic signal where we can observe important peaks in the power spectrum on the frequencies of the periods, and eventually some smaller peaks, for the harmonic frequencies of the main frequencies. The second possibility is the random case, in which the power spectrum seems continuous without special peaks. In that case, the power spectrum reflects broad band noise.

5.5. *Kolmogorov Entropy*

In the phase space of a dynamical system exhibiting sensitivity to initial conditions, two initially close points will evolve on divergent trajectories. We can think of this in terms of information creation when we observe that the two points that were initially indistinguishable become clearly distinguishable after a certain period of time. So we might say that information is added or created. A measure for the asymptotic rate of information creation for an iterative transformation is the Kolmogorov entropy **K**. Again, this measure is difficult to compute unless one has a priori knowledge on the probabilities for the system to be in a particular state.

Grassberger and Procaccia (1983) have shown that the Kolmogorov entropy can be estimated by the following expression:

$$K_2 = lim_{m\to\infty} lim_{r\to O} \frac{1}{\Delta t} \log \frac{C^m(r)}{C^{m+1}(r)} \tag{9}$$

where C^m is the correlation integral of a time series with embedding dimension **m** and the Kolmogorov Entropy **K** represents the upper bond for K_2. This expression is easier to compute. In the periodic and semi-periodic case, the entropy will be zero. A chaotic system will be characterised by a finite entropy, a truly random system will have an infinite entropy.

6. Chaotic Aspects of the Backpropagation Algorithm

We will look at the backpropagation algorithm and study the dynamical Properties of its learning process. This implies evidently that we will primarily focus on the delta rule. As specified in the beginning of this paper, the delta rule for the output node is

$$\Delta_p W_{ji} = \beta(t_{pj} - O_{pj}) O_{pi} (1 - O_{pi}) O_{pi}$$

and for the hidden nodes

$$\Delta_p W_{ji} = \beta(\sum_{k=1}^{n} \delta_{pk} W_{kj}) O_{pi} (1 - O_{pi}) O_{pi}$$

These delta rules are the basic equations driving the learning process of the neural network and we can immediately see that they are non-linear. As the backpropagation algorithm is recursive by nature, the necessary conditions for a chaotic system are satisfied. Given the above, it would therefore seem natural to observe similar phenomena such as bifurcations and chaos as was found for the Verhulst equation.

In Van der Maas et al. (1990), a bifurcation diagram for the backpropagation algorithm is created using the sum of the absolute values of the weights for different learning rates. The parameter regime indicates that bifurcations occur for learning rates inferior to 2.3 and a window appears between β = 2.9 and β = 3.2. For learning rates superior to 3.3, the weights grow exponentially. They furthermore illustrate the presence of chaos by means of a phase space diagram, a power spectrum and the calculation of Lyapunov exponents. They emphasise that for the full range of values the network successfully learned.

A number of other papers discuss the chaotic aspects of neural networks other than backpropagation networks. In Aihara et al. (1990) a simple one neuron neural network is analysed that has a number of properties of biological neurons, such as the squid giant axons. The authors find a similar behaviour of alternating periodic and chaotic sequences of neuron responses. In Derrida and Meir (1988), it is shown that feed forward neural networks in general have a chaotic behaviour because the distance between two arbitrarily close configurations always increases, which may be interpreted as sensitivity to initial conditions. Similar results are discussed in Sompolinsky and Crisanti (1988) where a continuous time dynamical model of a non-linear network with random asymmetric couplings is studied. For these networks, phenomena such as oscillations, bifurcations and chaos are observed.

The following issues have not been addressed in the above mentioned papers and, while focusing on the Xor-function, will be so in the remainder of the paper: what is the parameter regime for the XOR-function, how does chaos evolve during the learning process, does chaos facilitates learning or not? In the section to follow we present in detail our findings.

6.1. *Bifurcation diagrams*

In Figure 7, we show the bifurcation diagrams for the XOR-function where the output-values of the network for each of the XOR-input pairs are plotted against different β-values for a temperature equal to 0.6. The bifurcation diagrams, constructed at different moments during the learning process, allow us to observe how the order of chaos evolves over time.

For different values of the learning rate β, we allowed the network to learn during a limited number of iterations, up to 40,000. For each run, we kept a record of the last 200 output values, generated by the network. These values are then plotted against the corresponding β-value, resulting in a bifurcation diagram.

From these diagrams, we can make the following observations. First, it is clear from the bifurcation diagrams in Figure 7 that we can distinguish between three possible states in which the neural network can be for each value of β:

- convergence of the network resulting in correct output values equal to 1 and 0 (in fact, we use the values 0.9 and 0.1 because of the asymptotic properties of the sigmoid function) (e.g. in Figure 7 (middle): for $\beta = 4$ to 14),
- finite periodicity corresponding to a finite number of values which do not correspond to the desired output (e.g. in Figure 7 (middle): for $\beta =$ 14 to 43),

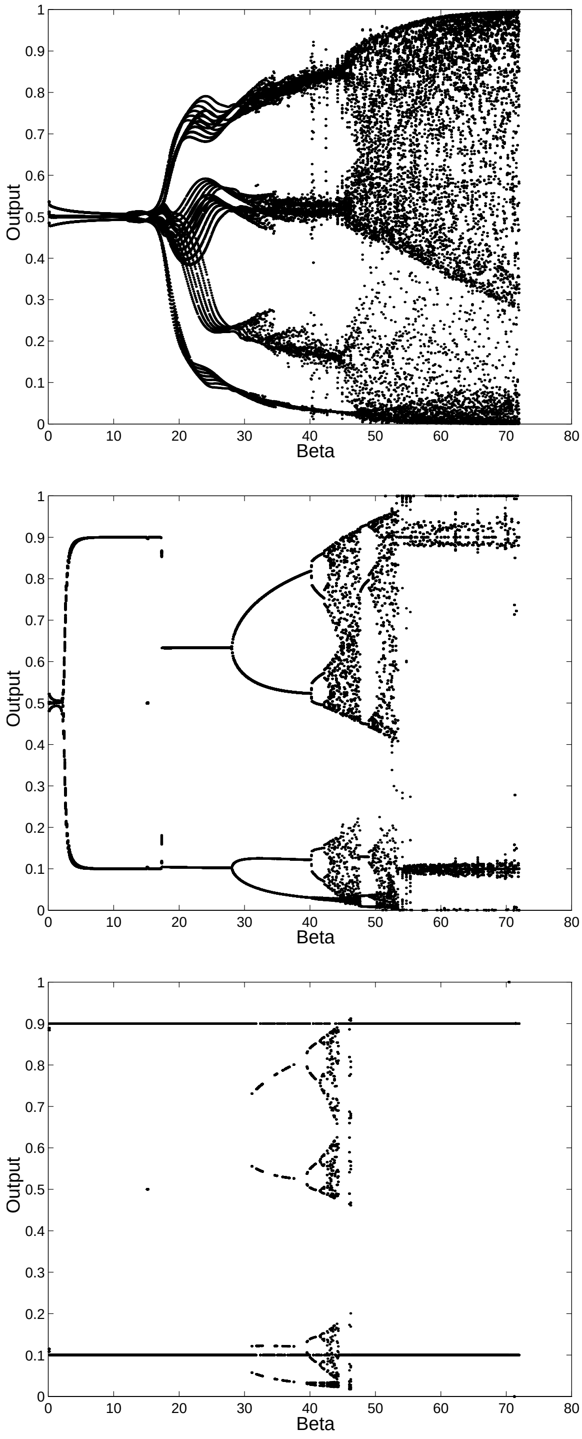

Figure 7. Bifurcation diagram for temp = 0.6 after 100, 1000 and 40,000 iterations.

- chaos having an unperiodic series of output values obviously not corresponding to the desired output (e.g. in Figure 7 (middle): for $\beta > 43$).

Secondly, as the number of iterations increases, the neural network converges for more β-values. After 1,000 iterations, we only have convergence for β-values around 4 and 14 whereas, after 40,000 iterations, convergence is achieved for values between 1 and 30, 48 and 75. Thirdly, as the number of iterations for the learning process increases, the order of chaos diminishes. When looking again at Figure 7, we see that, even in the chaotic zone, more and more values are correctly computed. We will see that this observation is supported by the computation of the Kolmogorov entropy.

6.2. *Phase space*

The phase space is a space in which each possible state of the system is represented unequivocally by a point in that space where each co-ordinate corresponds with a state variable of the system. If a dynamical system has a strange attractor, a phase space diagram will reveal its presence if the dimension of the attractor is less than the projection dimension. Some kind of structure will appear whenever some kind of deterministic system is involved (Broer and Takens 1992), which of course is the case for the backpropagation algorithm. However, this phase plan projection is nothing but an indication of chaos and its topographical characteristics are difficult to interpret.

In Figure 8, we show phase spaces for different values of β. For $\beta = 25$, which corresponds to the non chaotic zone, a relatively regular geometric object emerges. For $\beta = 45$ and 50, we find a much more irregular object. This is considered to be an indication of chaos (Moon 1987).

6.3. *Fourier power spectrum*

As was mentioned previously, chaotic systems are characterised by a broadband Fourier power spectrum in which no particular frequency can be found. We therefore expect to see for values inside the chaotic zone a broadband spectrum. We computed this power spectrum of the output values produced by the network for β-values 25, 45 and 50. As can be seen in Figure 9, in the periodic case the three peaks refer to the period-2 (left figure) and the continuous broadband spectrum (middle figure) to chaos (right figure).

6.4. *Kolmogorov Entropy*

We finally computed the Kolmogorov Entropy for the XOR-function using the Grassberger-Procaccia approximation. As was explained above, the entropy should be finite but non-zero in order to have a chaotic system. Not

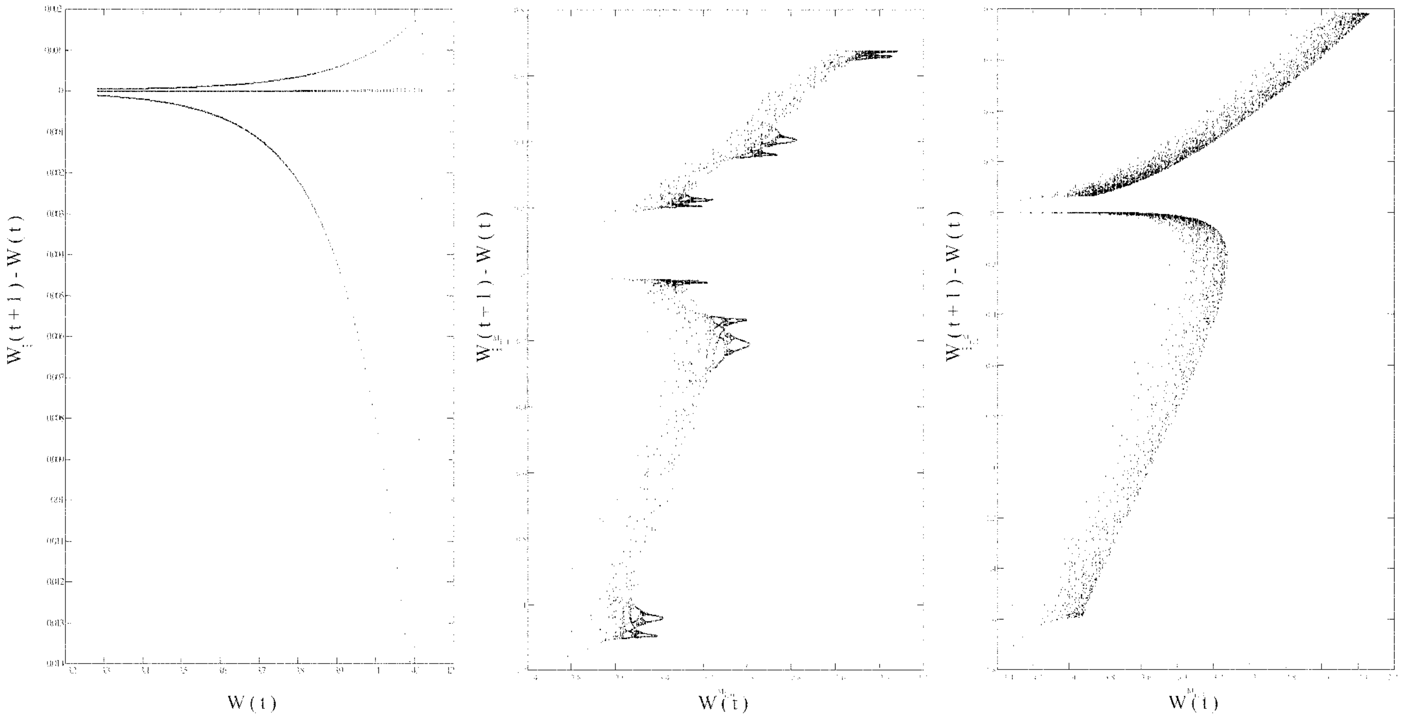

Figure 8. Phase spaces for temp = 0.6 and β = 25 (left), 45 (middle) and 50 (right).

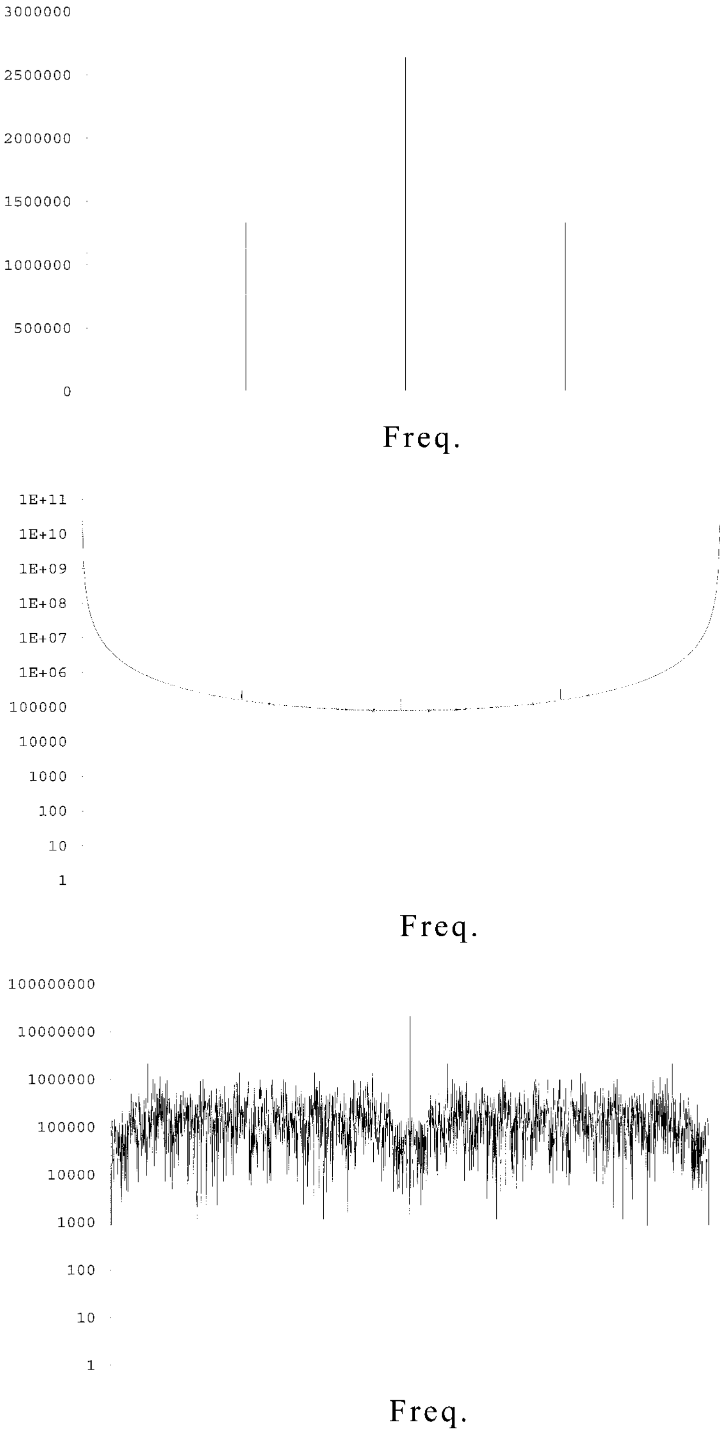

Figure 9. Power spectrum for temp = 0.6 and β = 25 (left), 45 (middle) and 50 (right).

Table 2. Kolmogorov entropy

Iterations	β = 25	β = 50	β = 45
0–10K	0	0.08	0.09
10–20K	0	0.05	0.08
20–30K	0	0.03	0.08
30–40K	0	0.01	0.08
40–50K	0	0	0.08

only did we compute the entropy for the same β-values (25, 45 and 50) for a temperature of 0.6, but we also calculated the evolution of the entropy as the network learns. The results are shown in Table 2 and clearly indicate that the process becomes less chaotic. For β = 45, there is initially a slight decrease in the entropy, but then this stabilises, implying that not all of the chaotic behaviour disappears. This is a confirmation of what we visually could observe in the different bifurcation diagrams (see Figure 7).

7. Conclusion

In this tutorial paper, we have investigated the backpropagation algorithm as a non linear dynamical system having a number of interesting behavioural characteristics. We have established the presence of chaos by means of the bifurcation diagram and the computation of both the Fourier spectrum and the entropy. The following conclusions hold: Firstly, for low values of the learning rate, no chaos occurs. Secondly, for larger learning rates, the learning process is clearly chaotic. Thirdly, the backpropagation network converges faster for those values of the learning rate for which no chaos occurs. This may be an indication that chaos does not prohibit a neural network to learn but increases the number of iterations needed in order to converge.

The main question now of course is: what are the implications for the use of neural networks? As was said in the beginning, the starting point of the research was to study the backpropagation algorithm as a dynamical system and to investigate the properties of its learning behaviour. Primary focus was therefore not on purely operational issues. Trying to give a more operational interpretation of the obtained results, the following might hold. As long as low, and therefore ’normal’ learning rates are used, the network behaves non-chaotic which seems to be necessary for fast learning. Prior to initiating a learning process for any kind of problem, it might be interesting to visually study the parameter regime of the network, given any set of inputs for a given problem. This would allow to find parameter values for the learning rate and the temperature for which no chaos occurs in order to assure fast learning. However, in order to reliably use such a parameterisation strategy, it is clear that further research needs to be done.

Notes

[1] For similar reasons the XOR-function has been chosen in (McClelland and Rumelhart 1988) as a representative example of an interesting problem to be solved.

[2] $f^n(x_0)$ represents the *n*th iteration of function *f*.

[3] Other, known and more or less understood ways to chaos are resonance overlap and intermittence (Broer and Verhulst 1992).

[4] Remark that we only look at the nonnegative values for *x*.

[5] This means that similar geometric structures are found at different scales in the bifurcation diagram, shown in Figure 4.

[6] In this paper, we will not compute Lyapunov exponents that measure the rate of divergence of the trajectories of two nearby initial points. Besides certain algorithmic difficulties with respect to their calculation, there are enough other measures to characterise chaos.

[7] A broadband spectrum means that the power spectrum covers the whole frequency spectrum without any distinguishable peak.

[8] It was also shown by Grassberger and Procaccia that $D^c(m)$ is a good approximation of the Hausdorff dimension ($D^c \leq D^H$), also known for its non-integer values as the fractal dimension (Grassberger and Procaccia 1983).

References

Aihara, Takabe & Toyoda (1990). Chaotic Neural Networks. *Physics Letters A* **144**(6, 7): 333–340.

Aleksander & Morton (1991). *An Introduction to Neural Computing*. Chapman & Hall: London, 240 p.

Bergé, Pomeau & Vidal (1992). *L'ordre dans le chaos*. Hermann: Paris, 352 p.

Broer, Dumortier, van Strien & Takens (1991). *Structures in Dynamics*. North-Holland: Amsterdam, 309 p.

Broer & Takens (1992). *Wegen naar Chaos en Vreemde Aantrekking*. In Broer & Verhulst (eds.) *Dynamische systemen en chaos*, 1–76. Epsilon Uitgaven: Utrecht.

Broer & Verhulst (1992). *Dynamische systemen en chaos*. Epsilon Uitgaven: Utrecht, 349 p.

Derrida & Meir (1988). Chaotic Behavior of a Layered Neural Network. *Physical Review A* (September) **38**(6): 3116–3119.

Feigenbaum (1980). *Universal Behavior in Nonlinear Systems*, 4–27. Los Alamos Science.

Grassberger & Procaccia (1983). Estimation of the Kolmogorov Entropy from a Choatic Signal. *Physical Review A* **28**: 2591–2593.

Lorenz (1989). *Nonlinear Dynamical Economics and Chaotic Motion*. Springer Verlag, 248 p.

Moon (1987). *Chaotic Vibrations*. Wiley & Sons: New York.

McClelland & Rumelhart (1988). *Parallel Distributed Processing, Explorations in the micro-structure of Cognition*, vols. 1 and 2. MIT Press: Cambridge USA.

Sompolinsky & Crisanti (1988). Chaos in Random Neural Networks. *Physical Review Letters* **61**(3): 259–262.

Takens (1981). Detecting Strange Attractors in Turbulence. In Rand & Young (eds.) *Dynamical Systems and Turbulence*. Springer: Berlin/Heidelberg/New York.

Van der Maas, Verschure & Molenaar (1990). A Note on Chaotic Behavior in Simple Neural Networks. *Neural Networks* **3**: 119–122.

Weisss & Kulikowski (1991). *Computer Systems that Learn*. Morgan Kaufman Publ., San Mateo, 223 p.