Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER 131312
UNCLASSIFIED	

TITLE

PERIODIC AND CHAOTIC SIGNAL GENERATION IN A RECURSIVE ARTIFICIAL NEURAL NETWORK

System Number:

Patron Number:

Requester:

Notes: Paper #4 contained in Parent Sysnum #129006

DSIS Use only:

Deliver to: DK

Periodic and Chaotic Signal Generation in a Recursive Artificial Neural Network

Simon A. Barton
Defence Research Establishment Suffield,
Box 4000, Medicine Hat, Alberta, T1A 8K6, Canada

Abstract

Feigenbaum showed that a large class of nonlinear functions may recursively generate values that exhibit periodicity. The periodic length doubles as the magnitude of some parameter in the function increases, and the system finally becomes aperiodic, or chaotic.

A recursively connected network of nonlinear processing nodes, such as those using the sigmoidal function, may therefore be expected to exhibit chaotic behaviour under some circumstances. Hopfield showed that some recursive artificial neural networks are stable; i.e. they settle to a constant output pattern.

We show that a simple recursive network of only 3 sigmoidal nodes can exhibit cyclical behaviour whose periodicity depends on the magnitude of a node response parameter, s. High values of s produce chaotic signals. From the numerically estimated values of s for which each period doubling occurs, we obtain a value of 4.669 for Feigenbaum's universal constant delta (4.669201..), thus confirming that this is a chaotic system. Chaotic and complex cyclical patterns can also arise in networks containing many nodes.

The behaviour of artificial neural networks of recursive nonlinear processing elements can be rapidly moved from stability, through complex periodicity, to chaotic activity and back again, by only slight and subtle changes in a parameter that controls the steepness of the nonlinear node response functions. The practical implications for artificial neural network applications, and the possible biological significance of this effect are discussed.

1 Introduction

Consider a nonlinear function of a variable x, whose shape is determined by a parameter λ; i.e. $f(x;\lambda)$. Feigenbaum (Ref. 1) showed that for a large class of such functions, values generated recursively exhibit a periodicity that doubles as the magnitude of λ increases past well-defined values. Beyond a critical value of λ, recursive values of f become aperiodic and are said to be chaotic.

Iterative or recursive signal generation may be written

$$x_{i+1} = f(x_i;\lambda) \tag{1}$$

Thus if $x_{i+1} = x_i$ the period is 1, and x_i is a fixed point of f. If $x_{i+2} = x_i$ the period is 2, if $x_{i+4} = x_i$ the period is 4, and so on. If λ_n is the value of λ for which the n-th periodicity of x first occurs, Feigenbaum showed that λ_n converges geometrically in n to λ_∞, for which the system is aperiodic. He defined

$$\delta_n = \frac{\lambda_{n+1} - \lambda_n}{\lambda_{n+2} - \lambda_{n+1}} \tag{2}$$

and found that δ_n approaches a universal constant

$$\lim_{n\to\infty} \delta_n = \delta = 4.6692.. \tag{3}$$

for *all* nonlinear systems that exhibit period doubling.

A recursively connected network of nonlinear processing nodes, such as one using a sigmoid transfer function, can therefore be expected to generate periodic and chaotic signals under certain circumstances.

Recursive artificial neural networks are often referred to as Hopfield nets because of the pioneering work done by J.J. Hopfield. His original model (Ref. 2) employed a completely connected set of simple linear threshold elements (LTEs) as nodes. The output of each node was either on or off (0 or 1), depending on whether or not the sum of the weighted input signals was above the node threshold. The signals passing between nodes were modified by a matrix of weights, which were iteratively adjusted to minimize a Liapunov function. Hopfield's later work (Ref. 3) employed sigmoidal nodes. The output of a sigmoidal node is a continuous, nonlinear, monotonically increasing function of the summed input. For both types of network, Hopfield showed that stable network states, giving a constant output, were possible.

Bruck (Ref. 4) proved that Hopfield nets with LTEs *always* converge to stable or cyclic states of period 2 or 4, if the matrix of connection weights

between the nodes is symmetric or antisymmetric. He termed these "the only interesting cases", but also showed that for an arbitrary weight matrix there exists a network of order n that has a cycle of period $2^{n/3}$. For example, with 30 LTE nodes there exists a network of period 1024. Bruck did not consider sigmoidal nodes.

The presence and role of periodicity and chaos in biological systems has been discussed by Freeman and coworkers (Refs. 5-7), and a simulation of biological neurons by Chay (Ref. 8) also showed the presence of complex oscillations and chaos. A mathematical proof of the existence of period-doubling to chaos in sigmoidal neurons was also given by Wang (Ref. 9). The generation of chaotic behaviour in neural networks has been widely discussed (Refs. 10-15).

In this work we show that a simple recursive network of only 3 sigmoidal nodes can exhibit cyclical behaviour, with period doubling observed from 1,2,.. 1024. The periodicity depends on the magnitude of a node response parameter s, which determines the steepness of the sigmoidal curves. For higher values of s, chaotic behaviour results. From the values of s at which each period doubling occurs, we obtain an estimate (4.669) of Feigenbaum's universal constant δ, showing that the process does indeed lead to a chaotic system.

Beyond the first chaotic region, there exist smaller regions of s, for which cyclic behaviour and period doubling reoccur. We have observed regions of periodicity 7, 14, 28, 56,.., also 9, 18, 36, 72.., and 8, 16, 32, 64.. For even higher values of s, for which the sigmoids become almost step functions (quasi-LTEs), cycles of unpredictable period occur (3, 6, 5, 13, 4..).

2 Recursive Networks

We consider networks that are composed of highly connected processing nodes, the output of each node n being a sigmoidal function f_n of the total input signal x_n. We define

$$f_n(x_n; s, x_{0n}) \equiv [1 + \exp(-s(x_n - x_{0n}))]^{-1} \tag{4}$$

The function f_n is parametric in a steepness factor s and an offset x_{0n}. Figure 1 shows the sigmoidal curves for $s = 4.0$ and 8.0, with an offset of 0.5. The offset plays the same role as a bias or threshold.

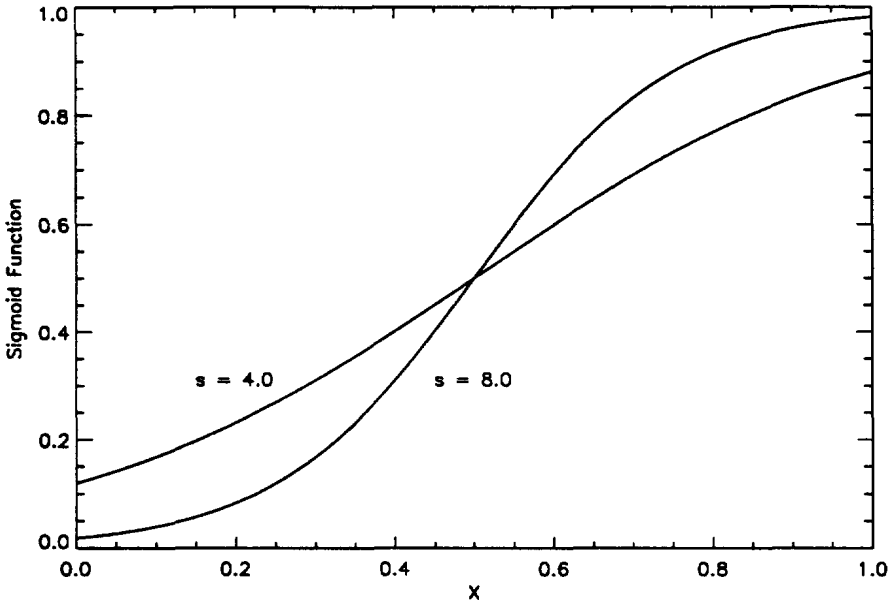

Figure 1: Sigmoid Functions with s = 4.0, 8.0, and $x_0 = 0.5$.

The total input signal to the n-th node at cycle t is

$$x_n^{(t)} = \sum_{m=1}^{N} W_{nm} f_m^{(t-1)} \tag{5}$$

i.e. it is the weighted sum of the outputs from the previous cycle of all the N nodes in the network. If the network is not completely connected, some of the weights will be zero. Also, the diagonal elements of the weight matrix $\mathbf{W}$ will be zero for nodes that do not have direct feedback.

In this work, we set $W_{nn} = 0$ for all n, and apply the following normalisation to the weights at each node (c is a constant)

$$\sum_m |W_{nm}| \equiv c \tag{6}$$

In principle, the steepness parameter s could be different for each node, but in this work the same value of s is applied to every node.

The recursive output of the network in matrix form is

$$\mathbf{f}^{(t)} = [1 + \exp(-s(\mathbf{W}\mathbf{f}^{(t-1)} - \mathbf{x}_0))]^{-1} \tag{7}$$

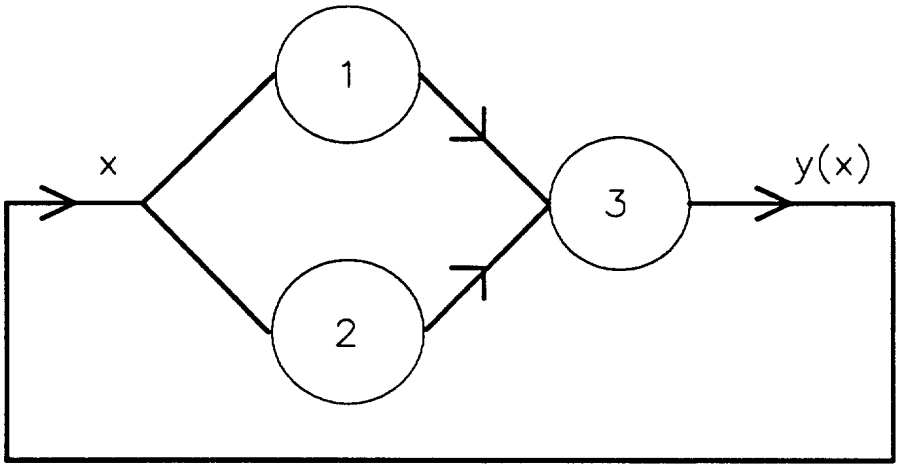

Figure 2: A Simple 3-Node Recursive Network.

The weights and offsets may themselves change from cycle to cycle. There are many weight updating schemes on the literature, most of which attempt to generate stable network states in response to given initial conditions; i.e. the vector **f** becomes fixed after a number of cycles. References 16 and 17 are collections that contain many of the important papers on weight updating algorithms. In this work, the weights are not altered from cycle to cycle.

In the next subsection we show how a simple 3-node system can be made to generate periodic and chaotic output.

2.1 A 3-node recursive network

Consider the simple recursive network of 3 nodes shown in Figure 2. Initially, we consider the function $y(x)$ that is generated by the group of 3 nodes in response to the input x. The output $y(x^{(t)})$ actually occurs after 2 network cycles; i.e.

$$y(x^{(t)}) = f_3^{(t+2)} \tag{8}$$

However, to demonstrate the property of period-doubling to chaos in a way that can be easily related to Feigenbaum's work (Ref. 1), we consider the iterative generation of the output of the 3-node group to be

$$x^{(t)} \rightarrow y(x^{(t)}) \rightarrow x^{(t+1)} \rightarrow y(x^{(t+1)}) \rightarrow \cdots \tag{9}$$

2.2 Generation of symmetric peak functions

The sigmoidal curves generated by nodes 1 and 2 of Figure 2 can be combined at node 3 in such a way that $y(x)$ is a symmetric peak function. This can be done by setting $W_{31} = -W_{32}$. Since nodes 1 and 2 are not connected, the weight matrix is

$$\mathbf{W} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ -\frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} \tag{10}$$

$\mathbf{W}$ satisfies the normalization condition (Eq. 6), and the node outputs are

$$f_n = [1 + \exp(-s(x - x_{0n}))]^{-1} \qquad (n = 1, 2) \tag{11}$$

and

$$f_3 = [1 + \exp(-s(p(x) - x_{03}))]^{-1} = y(x) \tag{12}$$

where

$$p(x) = \frac{1}{2}[f_2(x; s, x_{02}) - f_1(x; s, x_{01})] \tag{13}$$

If the two sigmoids f_1 and f_2 have different offsets, $p(x)$ is a peak function. Figure 3 shows $p(x)$ for several values of the steepness parameter (s = 10, 20, 30), with $x_{01} = 0.65$ and $x_{02} = 0.60$.

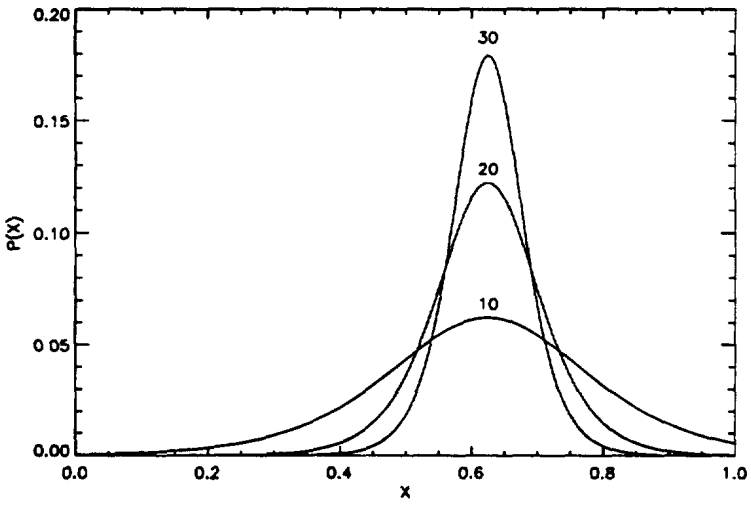

Figure 3: $p(x)$ with s = 10, 20, 30, and $x_{01} = 0.65$, $x_{02} = 0.60$.

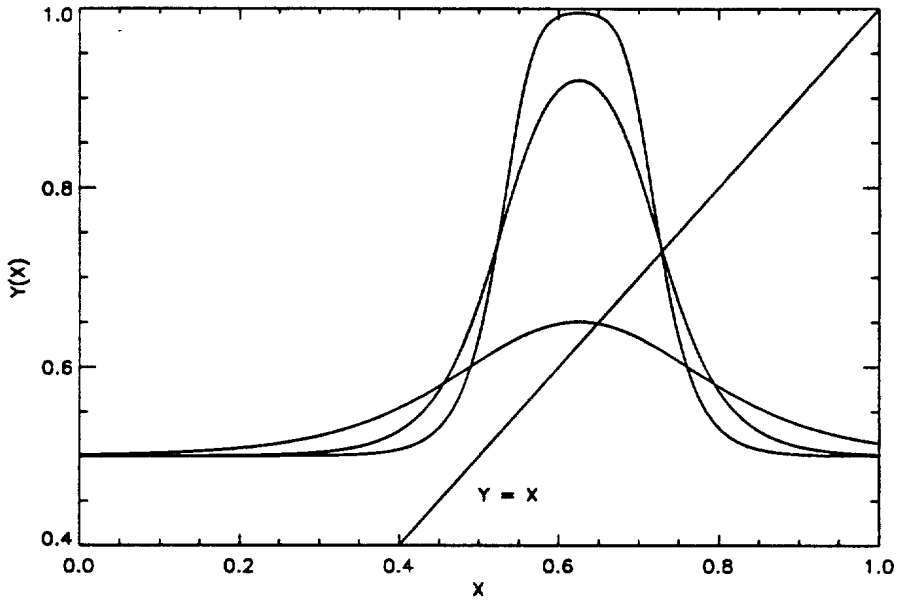

Figure 4: $y(x)$ with $s = 10$, 20, 30, and $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$. The line $y = x$ is also shown.

The function $y(x)$ also depends parametrically on the offset for node 3, x_{03}. Figure 4 shows $y(x)$ with $x_{03} = 0.0$, and using the same parameters as those used in Figure 3. It can be shown that the asymptotic values of $y(x)$ are 0.5, and that as s increases the maximum approaches 1.0; i.e. for $x_{03} = 0.0$

$$0.5 \leq y(x) \leq 1.0 \qquad (-\infty < x < \infty) \qquad (14)$$

If x_{03} is not zero, the baseline of $y(x)$ moves up ($x_{03} < 0$) or down ($x_{03} > 0$) the y-axis.

2.3 Periodicity and chaos

Figure 4 shows the interception of the line $y = x$ with curves of $y(x; s)$ for 3 values of the steepness parameter s.

Feigenbaum showed (Ref. 1) that when the magnitude of the slope of $y(x)$ at $y = x$ is less than 1, the point of interception is a stable fixed point (x_f); i.e. iterations from all initial values of x converge to x_f. Thus, the lowest curve in Figure 4, with $s = 10$, will generate recursive values of about

0.65 for x and $y(x)$, for *any* initial value of x. This can be seen from the following procedure:

1. choose any initial value on the x-axis; e.g. $x_1 = 0.2$;
2. find $y(x_1)$ on the curve;
3. move horizontally to the line $y = x$ to find x_2;
4. iterate steps 2 and 3.

The iteration converges rapidly to the interception point on the curve with $s = 10$.

It is possible for a curve to have *more than one* stable fixed point. For example, consider the curve of Figure 5, which has been displaced to the right by setting $x_{01} = 0.8$ and $x_{02} = 0.75$, and has $s = 15.0$. In this case there are *two* stable fixed points, near 0.54 and 0.79.

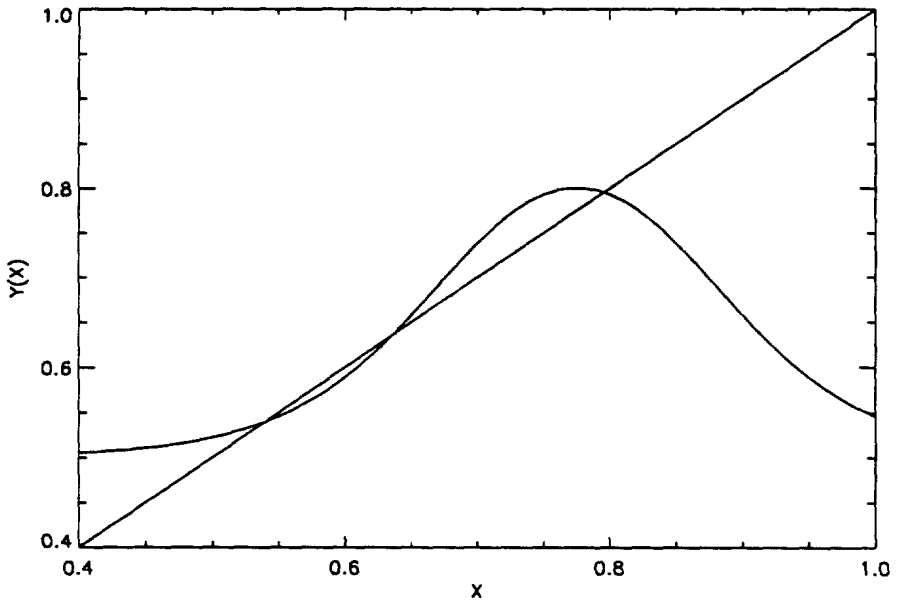

Figure 5: $y(x)$ with 2 stable fixed points; $s = 15$, $x_{01} = 0.8$, $x_{02} = 0.75$, $x_{03} = 0.0$. There is also an unstable point.

In Figure 5 there is also an *unstable* fixed point near 0.64. At this point the slope of $y(x)$ is > 1, and iterations diverge away from the point to either of the stable points. Therefore this system can stabilize to either of two values, depending on the initial value. There are certain regions of x for which the iterative result is extremely sensitive to the initial value; e.g. near 0.64 and 0.92 in Figure 5.

When many sigmoidal nodes provide input to another node, the output function can contain many peaks. Therefore, large Hopfield networks can stabilize to any of many fixed patterns, and which pattern results may be very sensitive to the initial conditions.

Consider the cases in Figure 4 where $|y'| > 1$ at $y = x$; i.e. with $s = 20$ and 30. Now the iterations diverge away from $y = x$. This is an unstable fixed point. As s increases from 10, there is a value for which the iteration settles into a cycle of period 2.

This *period doubling* occurs because the curve $y(y(x))$, which we write $y^{(2)}(x)$, has 2 stable fixed points. An example is shown in Figure 6, for $s = 15.0$.

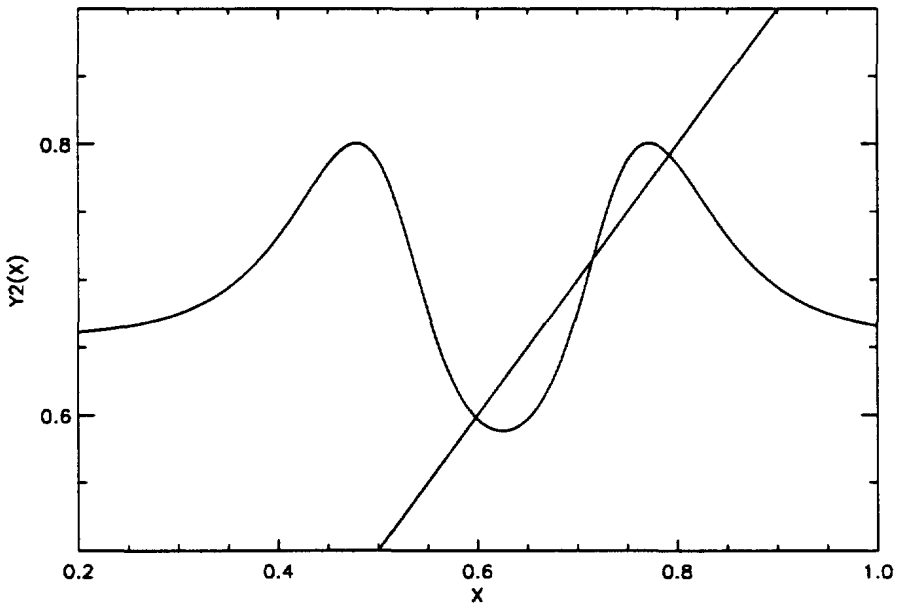

Figure 6: $y^{(2)}(x)$ with 2 stable fixed points; $s = 15$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$. There is also an unstable point.

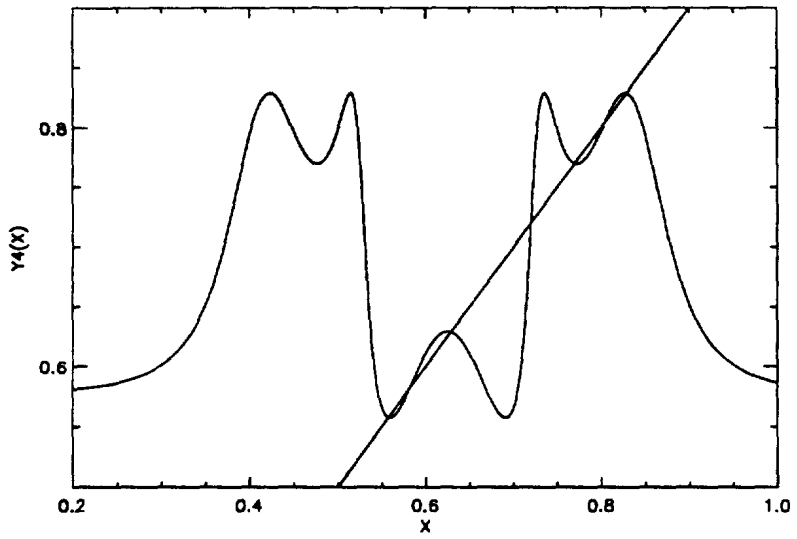

Figure 7: $y^{(4)}(x)$ with 4 stable fixed points; $s = 16$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$. There are also 3 unstable points.

As s increases, the two stable points of $y^{(2)}$ become unstable, 4 new stable points first occur for $y^{(4)}$ (Figure 7, with $s = 16.0$), then 8 for $y^{(8)}$ (Figure 8, $s = 16.5$) and so on. Period doubling occurs *ad infinitum*, but the spacing between successive values of s at which doubling occurs becomes geometrically smaller. Thus there is a rapid convergence to a value s_∞, beyond which the system is aperiodic, or chaotic.

Table 1 lists the values of s_n, found numerically, at which the n-th period doubling was first observed to occur. Table 1 also gives the values of Feigenbaum's constants δ_n, calculated by substituting s_n for λ_n in Eq. 2. This shows a rapid convergence to Feigenbaum's value of δ (4.669201..).

It is quite difficult to obtain accurate numerical estimates of s_n, for several reasons. First, for any value of s that is very near a period doubling transition, it takes hundreds of thousands of iterations of the x-values to stabilize to a precise fixed cycle. For example, Figure 9 shows the slow convergence of iterated x-values when $s = 12.9$, near the first period doubling. As s approaches s_1 (12.9735..) the convergence gets slower, and as s increases further a stable oscillator of period 2 eventually emerges, as shown in Figure 10 for $s = 13.1$.

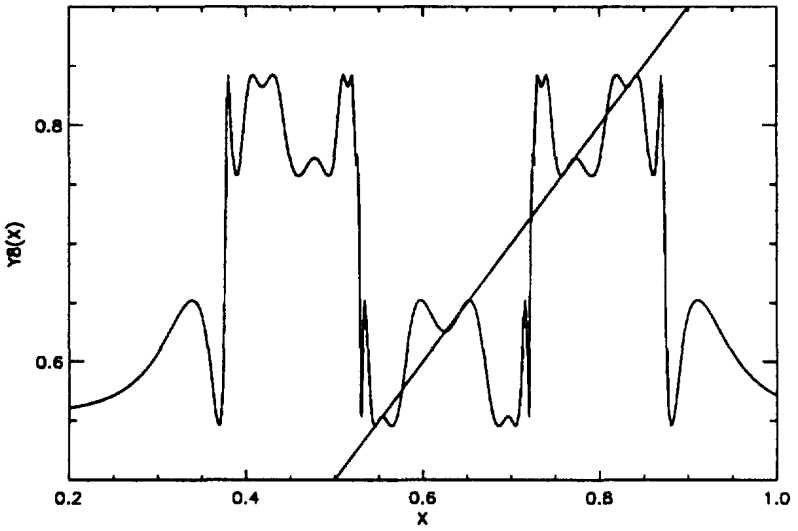

Figure 8: $y^{(8)}(x)$ with 8 stable fixed points; $s = 16.5$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$. There are also 7 unstable points.

n	s_n	period	δ_n
1	12.9735	2	2.503
2	15.4103	4	4.124
3	16.3838	8	4.579
4	16.61984	16	4.646
5	16.6713832	32	4.665
6	16.6824763	64	4.669
7	16.6848542	128	-
8	16.68536348	256	-

Table 1: Values of s at which period doubling first occurs

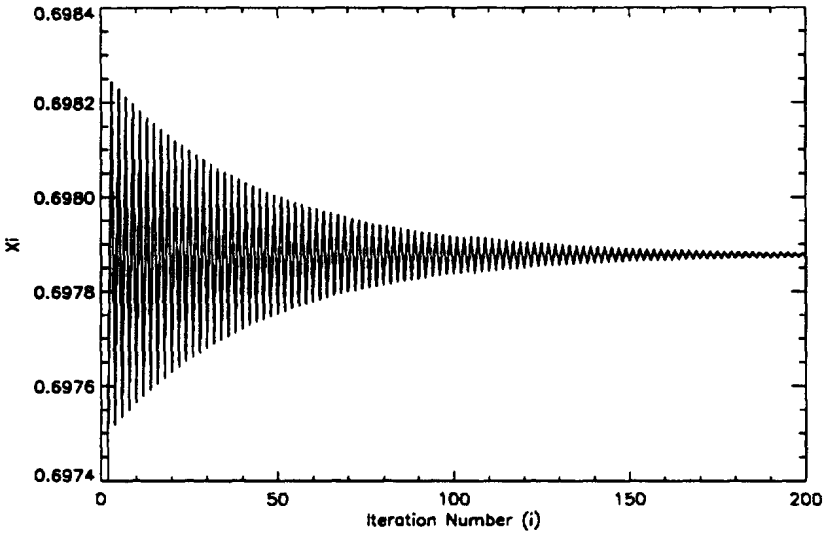

Figure 9: Slow convergence of x_i when $s = 12.9$, near the first period doubling .

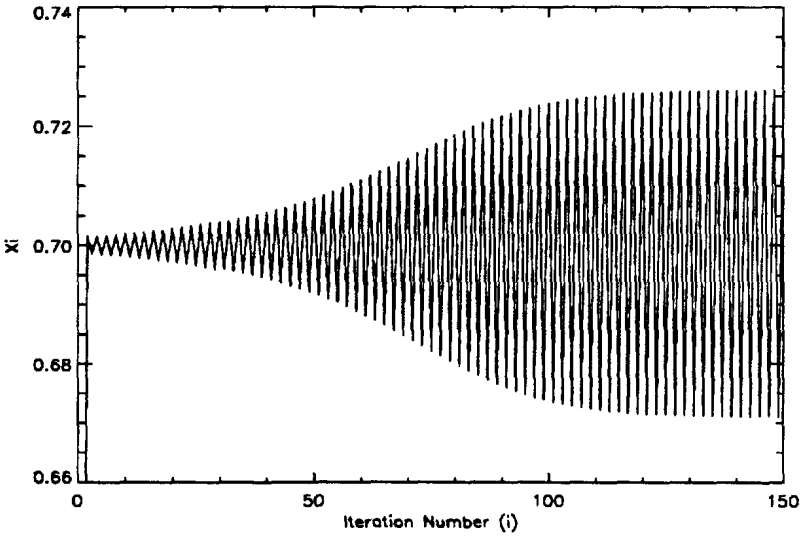

Figure 10: Convergence to a cycle of period 2 ($s = 13.1$).

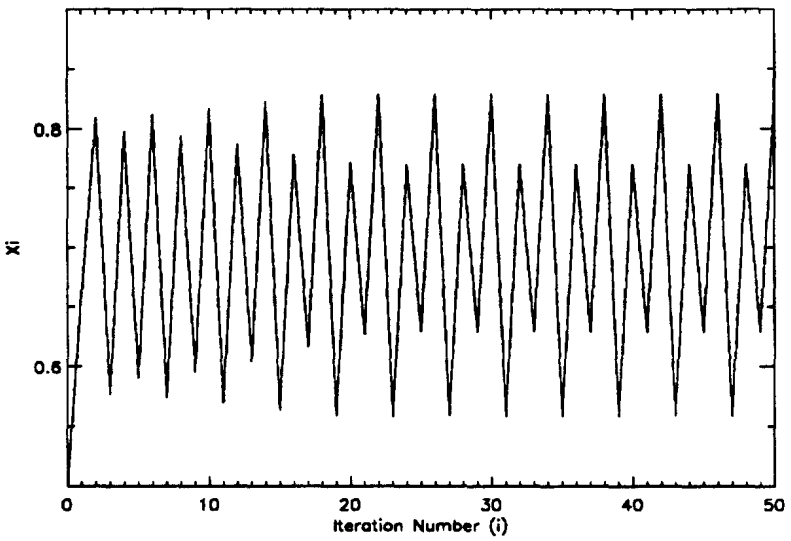

Figure 11: Cycle of period 4 ($s = 16.0$).

The closer s is to a period-doubling transition, the greater is the number of iterations required to determine the stable cycle, and so some criteria must be chosen to define the point at which period doubling is accepted to have occurred. Our algorithm performs the following two tests for a cycle of period N:

$$|x_i - x_{i-N}| < \Delta \tag{15}$$

and

$$|x_i - x_{i-M}| > \epsilon \qquad (M < N) \tag{16}$$

We used Δ = 1.0E-12 and ϵ = 1.0E-08. The second condition, Eq. 16, establishes the uniqueness of at least one other member of a cycle of period N. After every iteration, the new point must be checked against up to N previous points. Thus, if for example the period = 1024, thousands of checks must be performed on hundreds of thousands of iterations. This is quite CPU intensive, taking several minutes on a typical Unix workstation. We used a DECstation 5000/125. All calculations were done in double precision, using a C program.

Figures 11 and 12 show example of cycles of periods 4 and 8 respectively. They show that when the value of s is *not* near a transition, a stable cycle is quickly established after only a few iterations.

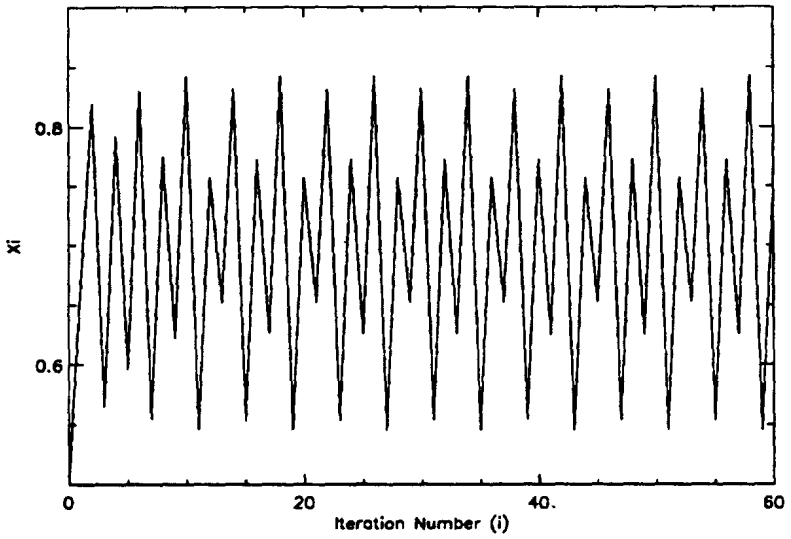

Figure 12: Cycle of period 8 ($s = 16.5$).

Table 1 shows values of s that generate period doublings up to 256. Cycles of period 512 and 1024 were also observed with $s = 16.6854726$ and 16.6854960 respectively, but no accurate estimates of s_9 and s_{10} were attempted numerically because the differences are approaching the limit of double precision accuracy.

Table 1 also indicates that a region of chaotic (aperiodic) signal generation would be expected for values of $s > 16.7$, and in fact no periodicity was observed immediately beyond this value. Figure 13 shows an example of chaotic iterative x values for $s = 17.0$. The values shown are an arbitrary set of 200 consecutive iterations.

Iterative periodicity and chaos can be represented as trajectories in xy-space. If we take each input signal x_i and its output y_i (i.e. x_{i+1}) to be a coordinate pair, then the movement of these points represents a trajectory. Figure 14 shows the trajectory for the cycle of period 8 that was shown in Figure 12. Figure 14 is actually a plot of 101 points from the 400th to 500th iterations. This shows that the system is constrained to move in a closed 8-point trajectory. Each of the 8 points must, of course, lie on the curve $y(x)$. Figure 15 is the trajectory for the chaotic system shown in Figure 13. In this case, the trajectory is *not* closed. Plotting more points results in completely filling the curve $y(x)$ to which the trajectory is confined.

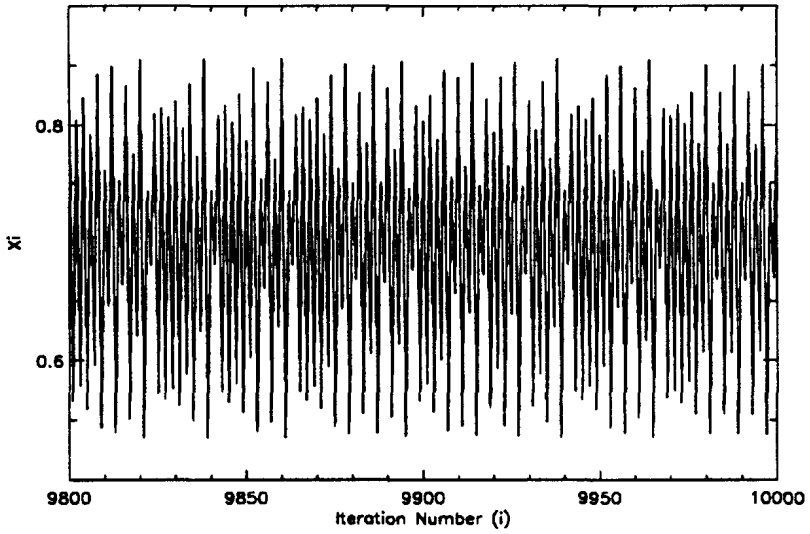

Figure 13: Chaotic signal generation ($s = 17.0$).

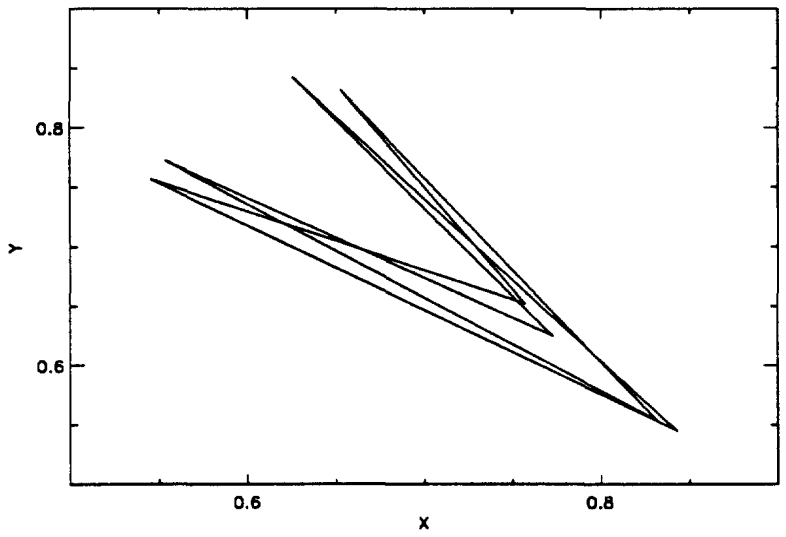

Figure 14: Trajectory of a cycle of period 8 ($s = 16.5$).

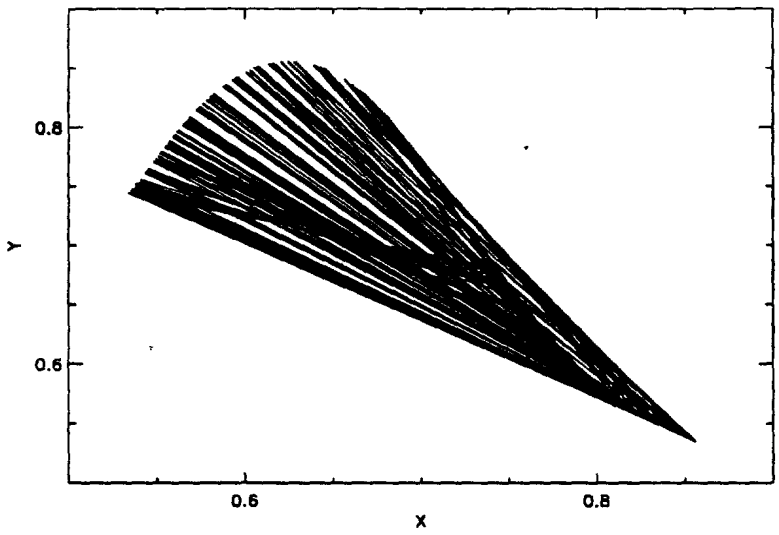

Figure 15: Chaotic trajectory ($s = 17.0$).

As s increases, the system remains chaotic until a cycle of period 7 appears near $s = 20.0$. Period doubling then reoccurs as s increases, as summarized in Table 2. The way in which this region of periodicity emerges from a chaotic region can be understood by looking at the curve $y^{(7)}(x)$ for $s = 20.0$, where the period is 7. Figure 16 shows that this curve has 7 points where $|y'| < 1$, where the line $y = x$ intercepts maxima or minima. Figure 17 shows the same curve, but with $s = 19.0$, just before the onset of periodicity.

n	s_n	period
0	19.996	7
1	20.090	14
2	20.131	28
3	20.140	56
4	20.1415	112
5	20.1419	224

Table 2: Approximate values of s in second period doubling region

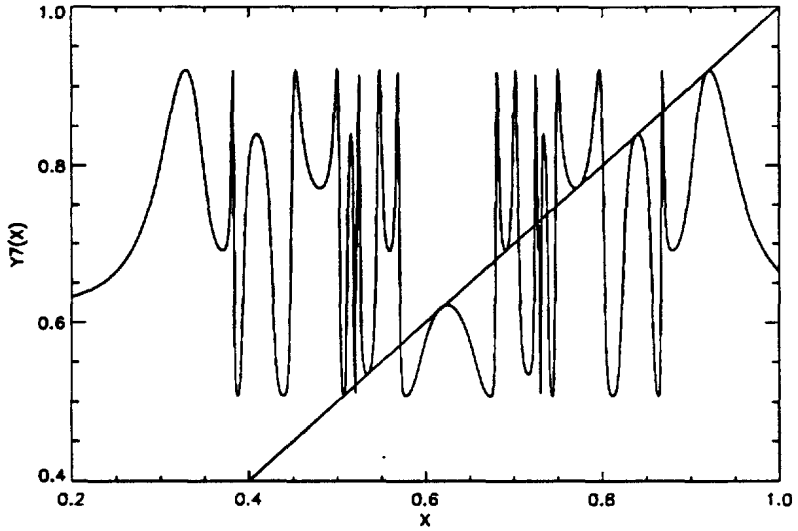

Figure 16: $y^{(7)}(x)$ with 7 stable fixed points; $s = 20.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

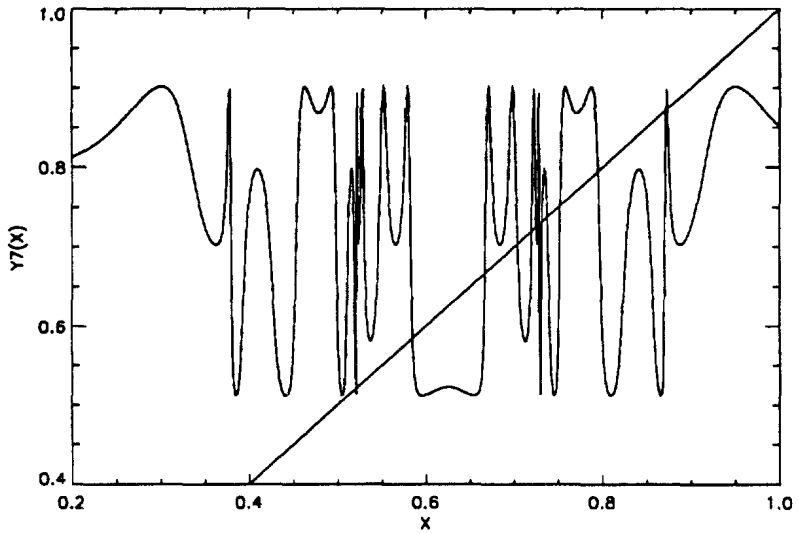

Figure 17: $y^{(7)}(x)$ with no stable fixed points; $s = 19.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

n	s_n	period
0	20.600	9
1	20.602	18
2	20.6032	36
3	20.60351	72
4	20.60358	144
0	20.95	8
1	21.00	16
2	21.024	32
3	21.029	64
4	21.030	128

Table 3: Approximate values of s in 3rd and 4th period doubling regions

The second region of periodicity spans a much smaller range of s than the first, and at $s = 20.15$ the system is again chaotic. However, because of the emergence of new maxima and minima in the curves $y^{(n)}(x)$, further regions of period doubling occur. At $s = 20.60$, a cycle of period 9 is established, period doubling reoccurs, then again at $s = 20.95$ a cycle of period 8 appears. These 3rd and 4th regions of period doubling are summarized in Table 3.

Beyond the 4th period doubling region the system is chaotic until about $s = 24$, where a cycle of period 3 appears. Now the sigmoids are so steep that they are almost step functions. At $s = 23.0$ (Figure 18) $y^{(3)}(x)$ has no value for which $|y'| < 1$ where $y = x$, but when $s = 25.0$ (Figure 19) there are three stable fixed points. This system of period 3 remains stable as s increases from 24 to 47; i.e. no period doubling occurs. This is due to the flat extrema produced by the quasi-step functions.

At $s = 48$, a cycle of period 6 occurs. However, this is not a process of period doubling leading to chaos, but rather a series of abrupt and unpredictable transitions. As s increases, cycles of period 5, 13, 4, 6, 5 and 7 appear for $s = 50$, 51, 52, 61, 62 and 70 respectively. Figures 20 and 21, showing $y^{(3)}(x)$ for $s = 47$ and 49 respectively, demonstrate the origin of these sharp transitions, arising near discontinuities in the steep quasi-step functions. Figure 22 shows the 6 fixed points for $s = 49$, and Figures 23 to 25 are examples of cycles with periods 5, 13 and 4. In this domain of s the fixed points jump erratically from one flat region of a y iterate to another.

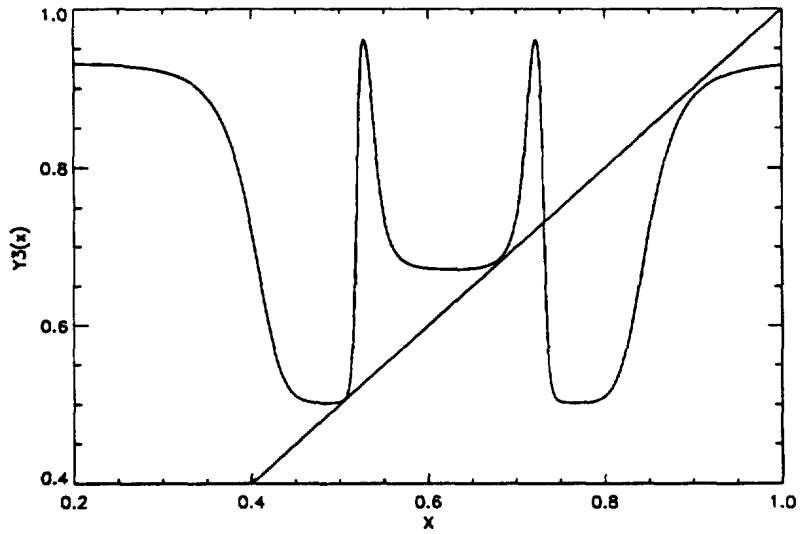

Figure 18: $y^{(3)}(x)$ with no stable fixed points; $s = 23.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

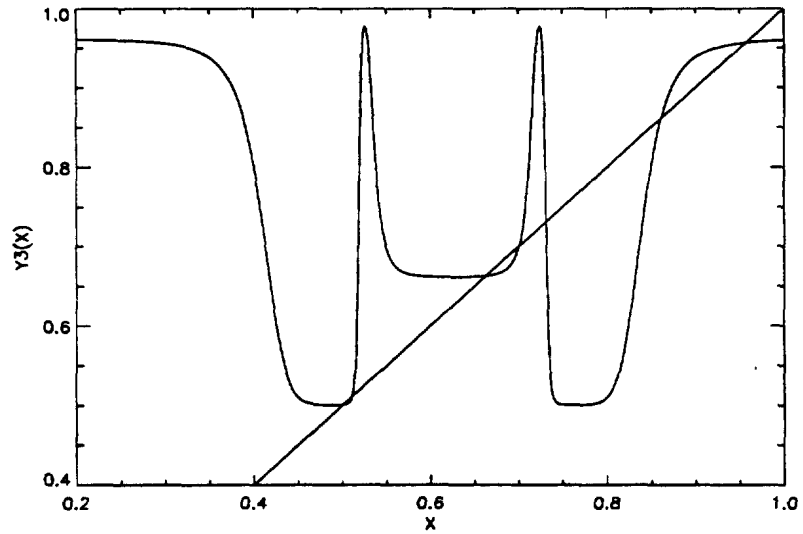

Figure 19: $y^{(3)}(x)$ with 3 stable fixed points; $s = 25.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

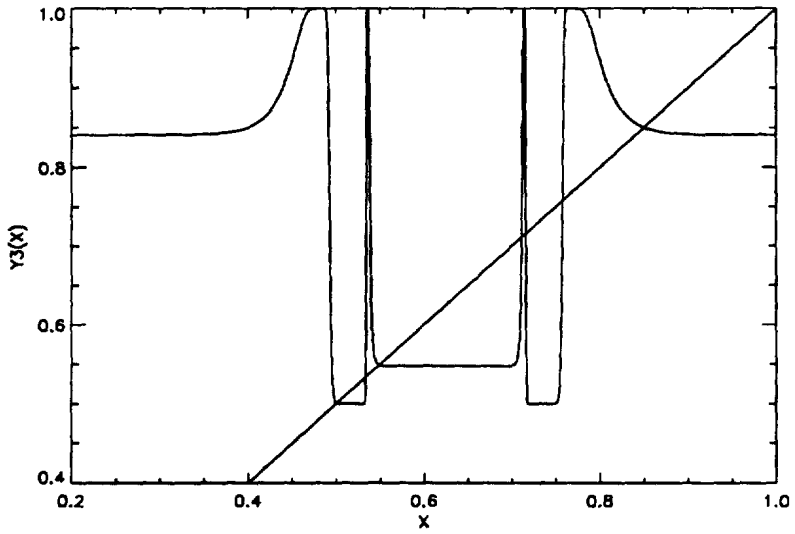

Figure 20: $y^{(3)}(x)$ with 3 stable fixed points; $s = 47.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

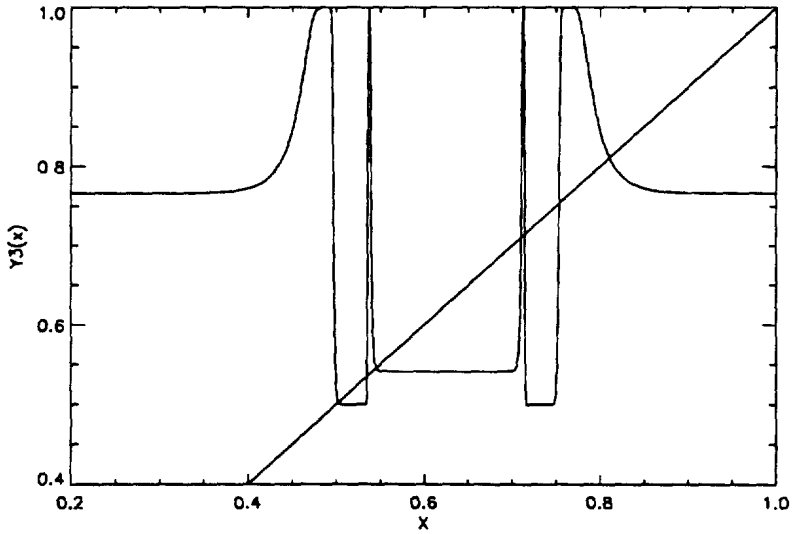

Figure 21: $y^{(3)}(x)$ with no stable fixed points; $s = 49.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

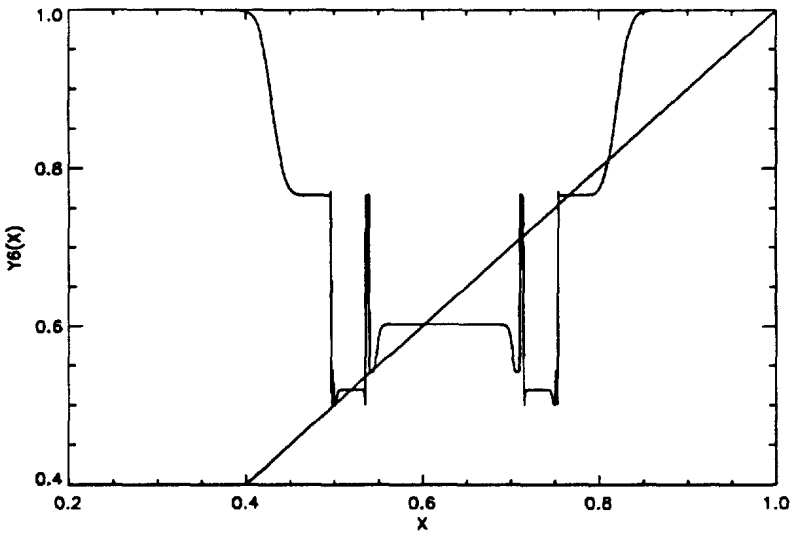

Figure 22: $y^{(6)}(x)$ with 6 stable fixed points; $s = 49.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

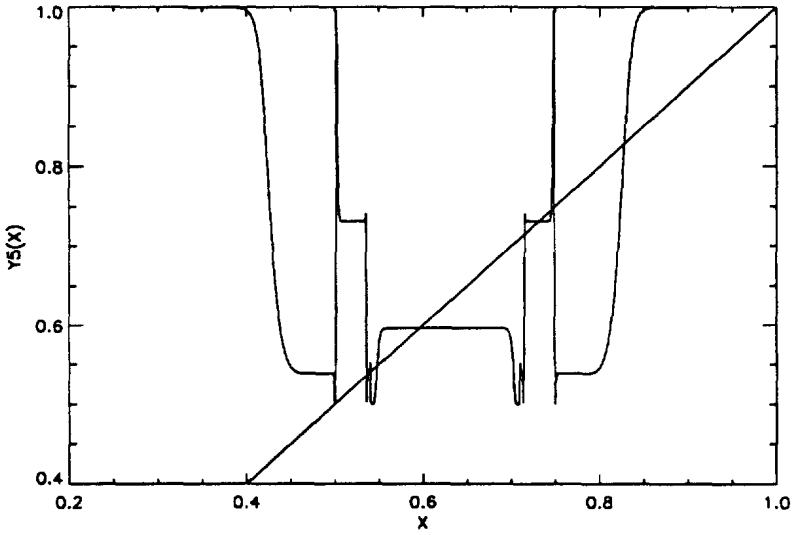

Figure 23: $y^{(5)}(x)$ with 5 stable fixed points; $s = 50.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

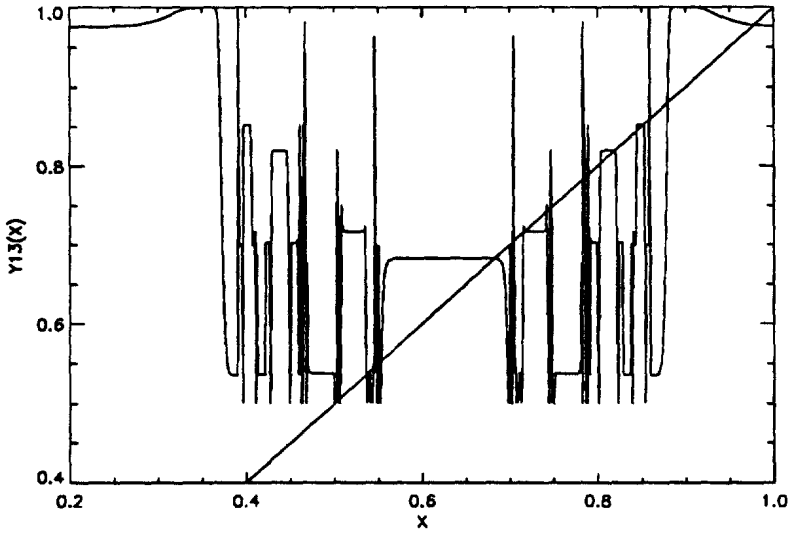

Figure 24: $y^{(13)}(x)$ with 13 stable fixed points; $s = 51.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

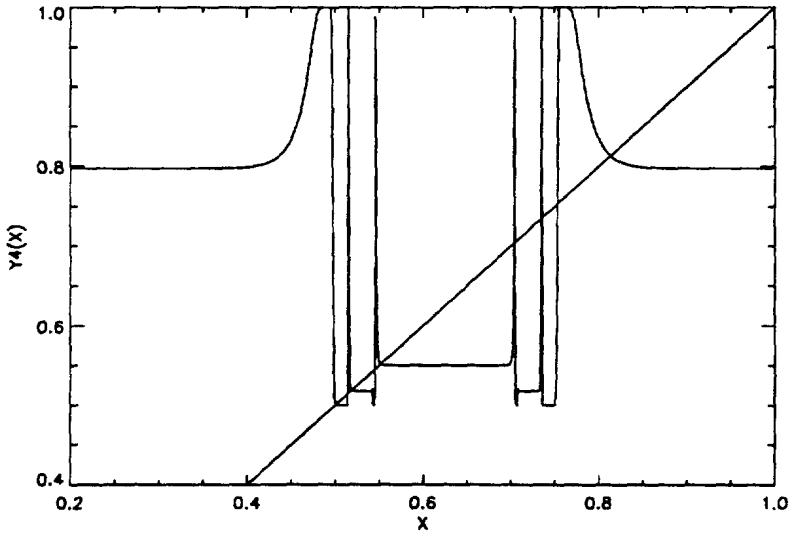

Figure 25: $y^{(4)}(x)$ with 4 stable fixed points; $s = 60.0$, $x_{01} = 0.65$, $x_{02} = 0.60$, $x_{03} = 0.0$.

2.4 Multinode recursive networks

We have simulated several completely connected multinode recursive networks. The nodes had no self-feedback ($W_{nn} = 0$). The weights for each node were chosen randomly between -1 and +1, and then normalized (Equation 6), with the normalization constant generally chosen to be the number of nodes in the network. The offsets, contained in the vector $\mathbf{x}_0$, were chosen randomly between 0 and 1.

The state at time t of a recursive network of N nodes is given by the vector $\mathbf{f}^{(t)}$. It is calculated from the state at the previous cycle using Equation 7. The initial components of the state vector for each network were chosen randomly between 0 and 1.

When a recursive network operates cyclically, periodicity must be detected by testing all the components of the state vector with all the components from previous cycles (cf. Equations 15 and 16). For example, for a 100 node network, the outputs from all the 100 nodes must be identical to those from a previous cycle for the system to be termed periodic.

From the work described in the previous section on 3-node networks, we expected that periodicity and chaos should occur for some random multinode networks, and that it should be possible to control this behaviour by adjusting the node steepness parameter s for the whole network.

Initial work on nets containing 20 and 100 nodes support this. A 20-node network showed cycles of period 1, 7, 14 as s increased from 1.0 to 2.1. However, beyond about 2.194, we could detect no periodicity. The 20-node system then appears to be chaotic. This aperiodicity persists until s reaches about 6.0, where a cycle of 1 occurs. For higher values of s, erratic periodicity occurs, much as in the 3-node case, with cycles of period 38, 1, 142, 1, 31, 33, 13,.. being observed in the region $s = 10$ to 20.

For the 100-node network, the system appears to be stable, i.e. converges to a cycle of period 1, for $s < 0.69$. Above this value we have detected no periodicity. The transition from stable cyclic behaviour to chaos seems to be much more abrupt for larger networks.

3 Conclusions

The iterative behaviour of a recursive network of sigmoidal nodes can converge to a stable or periodic state, or it may be chaotic. The behaviour of a 3-node system can be understood using the Feigenbaum model of a nonlinear system driven by single parameter, in this case the node response parameter s. As s increases, a stable system becomes periodic, and then exhibits period doubling to chaos. Beyond the first chaotic region, other regions of periodicity and chaos occur, and because of the step-like functions generated by higher s values, a region of erratic periodicity follows.

The behaviour of large recursive networks is much more difficult to analyse. However, it seems that the node response parameter s can be used to ensure stability. Below a certain value of s, a multi-node network will converge to a stable state. The onset of chaotic signal generation appears to be much more abrupt from our preliminary studies of networks of 20 and 100 nodes. The choice of the range for the weights and offsets, and the initial random network state vector, has a big effect on the behaviour of these larger systems. So far we have only studied a few cases. Much work remains to be done to understand more clearly the process of stability, periodicity and chaos in large recursive networks.

Depending on the exact initial state, a multinode net may converge iteratively to one of many possible stable states, and this has been proposed as a mechanism for pattern recognition. It should be possible to avoid iterative chaos in pattern recognition networks by adjusting the node steepness parameter.

On the other hand, there is strong evidence that chaotic signals are generated in at least some biological neural systems. In particular, Freeman *et al* (References 5-7) present evidence that the mammalian olfactory system generates chaotic signals during exhalation, i.e. with no input signal. Inhalation of a recognized odour appears to be associated with a cyclical periodic signal in the olfactory bulb. Chaotic signal generation may play a vital role in this neural system. Since the convergence to a particular pattern in a recursive neural system is critically dependent on the initial state of the system, chaos may provide an unbiased baseline upon which the incoming signal is imposed.

One can speculate on other uses of chaos in biological systems. If a system must respond in an unpredictable way, for example an insect such as a fly or mosquito that is disturbed by some movement, the generation of chaotic signals may assist its survival. Also, during the learning process, an

organism must generate many random movements in order for the neural system to learn the effects of a changing environment. In other words it learns by trial and error.

Some of the possibilities may be explored by constructing a simulated neural network that controls a simple model creature, and then allowing it to evolve in a simulated environment. The development of behaviour patterns may then be studied using different synaptic weight modification schemes. Our work in this area has the long-term goal of assisting the development of autonomous land vehicles, initially focussing on obstacle avoidance using input from an array of sensors.

4 References

1. Feigenbaum, M.J., "Universal Behaviour in Nonlinear Systems", Los Alamos Science, Summer 1980, pp. 4-27.

2. Hopfield, J.J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proc. Natl. Acad. Sci. USA, **79** (1982), pp. 2554-2558.

3. Hopfield, J.J., "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons", Proc. Natl. Acad. Sci. USA, **81** (1984), pp. 3088- 3092.

4. Bruck, J., "On the Convergence Properties of the Hopfield Model", Proc. IEEE, **78** (1990), pp. 1579-1585.

5. Skarda, C.A. and Freeman, W.J., "How Brains Make Chaos in Order to Make Sense of the World", Behavioural and Brain Sciences, **10** (1987), pp. 161-195.

6. Yao, Y. and Freeman, W.J., "Model of Biological Recognition with Spatially Chaotic Dynamics", Neural Networks, **3** (1990), pp. 153-170.

7. Freeman, W.J., "The Physiology of Perception", Scientific American, February 1991, pp. 78-85.

8. Chay, T.R., "Complex Oscillations and Chaos in a Simple Neuron Model", Proc. IJCNN, Vol. II, July 1991, pp. 657-662.

9. Wang, X, "Period Doublings to Chaos in a Simple Neural Network", Proc. IJCNN, Vol. II, July 1991, pp. 333-338.

10. Kurten, K.E. and Clark, J.W., "Chaos in Neural Systems", Physics Lett., **114A** (1986), pp. 413-418.

11. Kurten, K.E., "Critical Phenomena in Model Neural Networks", Physics Lett. A, **129** (1988), pp. 157-160.

12. Riedel, U., Kuhn, R. and van Hemmen, J.L., "Temporal Sequences and Chaos in Neural Nets", Phys. Rev. A, **38**, (1988), pp. 1105-1108.

13. Sompolinsky, H. and Crisanti, A., "Chaos in Random Neural Networks", Phys. Rev. Lett., **61** (1988), pp. 259-262.

14. Bauer, M. and Martiennssen, W., "Quasi-Periodicity Route to Chaos in Neural Networks", Europhys. Lett., **10** (1989), pp. 427-431.

15. van der Maas, H.L.J., Verschure, P.F.M.J. and Molenaar, P.C.M., "A Note on Chaotic Behaviour in Simple Neural Networks", Neural Networks, **3** (1990), pp. 119-122.

16. "Neurocomputing, Foundations of Research", Eds. Anderson J.A. and Rosenfeld, E., MIT Press, Cambridge MA, 1988.

17. "Neurocomputing 2, Directions for Research", Eds. Anderson J.A. and Rosenfeld, E., MIT Press, Cambridge MA, 1990.