Thesis Proposal: Chaos in Classes of Artificial Neural Networks

Ralph Bean
Department of Computer Science
Rochester Institute of Technology
ralph.bean@gmail.com

November 14, 2008

Thesis Committe.
Advisor: Dr. Roger Gaborski
Observer: Dr. Peter Anderson *(Unconfirmed)*
Reader: Thomas Borelli

Contents

1 Introduction **3**

2 Artificial Neural Networks (ANNs) **4**

3 Chaos **4**

4 Chaos in ANNs **4**
4.1 Autonomous ANNs 4
4.1.1 Single-neuron Networks 5
4.1.2 Multi-neuron Networks 6
4.2 Nonautonomous ANNs 9

5 Genetic Algorithm **10**
5.1 Goal 10
5.2 Representation 11
5.3 Crossover 12
5.4 Mutation 12
5.5 Fitness Metric 12
5.5.1 Lyapunov Spectrum 12
5.5.2 Constraint Satisfaction 13
5.6 Distributed Implementation 13

6 Proposed Timeline **14**

1 Introduction

The computational tasks of nonlinear classification and control are of high importance to those interested in vision and acoustics. The information of interest in real world signals is most often embedded intrinsically in their temporal structure. Examples include distinguishing between and producing the subtleties of human speech and predicting natural time series such as weather patterns and radioactive decay [17, 6]

For many tasks, humans outcompete state-of-the-art computational techniques hands-down in both accuracy and flexibility.

Consequently, techniques regarding recurrent neural networks are of interest to computing research as a potential solution which could harness the hard work already done for us computer scientists by the process of biological evolution itself.

A large amount of theory has been developed with regard to feed-forward neural networks from how to train them in both supervised and unsupervised paradigms as well as why and how they work. Things are not as nice when we consider networks that have recurrent connections or feed-back connections. Doya [4] provides an excellent description of the problems confronting gradient descent for training recurrent networks.

The primary inspiration for this proposal came in 2001 from the work of Jaeger [16]. The Echo State Network (ESN) described there combined a randomly generated recurrent network with a linear readout mechanism trained to classify the state of the recurrent network. This proposed thesis treats itself as a part of a process attempting to theoretically explain the experimental success of Jaeger and others work [14].

One attempt to do so was authored by Verstraten [17] who was the first to use Lyapunov exponents as a measurement of chaos in the context of Jaeger's work. It must be stated though that Verstraeten's application was built on a misunderstanding of Lyapunov exponents. In [17], only the first iteration of the map defined by the network was considered in calculating Lyapunov exponents. Lyapunov exponents involve taking the limit of infinite compositions of the map with itself which, for transcendental activation functions like the hyperbolic tangent function most often considered for artificial neural networks, is incalculable.

The point of this proposed thesis is to clarify notions of the edge of chaos as they apply to artifical neural networks. Sections 2 and 3 briefly describe some background in artificial neural networks and dynamical systems theory.

The thrust of the thesis is a proof of the absence of chaotic motion in all autonomous artificial neural networks in section 4 and a proposal for a genetic algorithm to shed light on the common characteristics of nonautonomous artificial neural networks which do exhibit the chaotic motion in section 5. Section 6 is a timeline for completing the proposed work.

2 Artificial Neural Networks (ANNs)

Artificial Neural Networks are a computational model of natural brains that consist of a network of "neurons". These artificial neuron models activate more or less in response to each other and a series of inputs as governed by a system of equations [16, 4, 7]. They are perhaps best contrasted with the Biological Neural Network or Spike/Pulsed Neural Network computing models which are more biologically realistic but more computationally expensive and more theoretically unwieldy [13, 8, 10].

3 Chaos

Dynamical systems theory is a branch of mathematics that deals with systems of change rules and the ultimate fate of solutions to those systems. A system which has orbits which do not leave a particular interval and are not periodic are called chaotic – and chaotic systems have many interesting properties [1, 2, 5, 9].

4 Chaos in ANNs

4.1 Autonomous ANNs

The author's stated purpose at the beginning of the summer was to "derive constraints on the parameters of a neural network such that the dynamics of the network under no input (i.e. the baseline behavior) are chaotic." In July, a proof was developed showing that there are no such solutions in the parameter space of an artificial (discrete-time) neural network which makes use of the sigmoid (hyperbolic tangent) activation function alone.

4.1.1 Single-neuron Networks

Chaos in one-dimensional discrete-time dynamical systems is much more well understood than in any other non-trivial class of systems. Proof that a new one-dimensional system is chaotic may be done by application of Sarkovskii's Theorem which, in its condensed form, states that any one-dimensional discrete-time dynamical system which has a periodic orbit of prime period 3, has periodic orbits of every other prime period [5]. The natural numbers are countably infinite. An infinite number of distinct periodic orbits on an open interval implies density (much like the argument for density of the rationals) and density of periodic orbits is the very topological definition of chaos.

Our one-neuron recurrent network can be expressed as a dynamical system:

$$x_{t+1} = f(x_t) = tanh(w * x_t + b)$$

where x_t is the state of the neuron at time t, w is the weight of the connection from this neuron to itself, and b is the bias of the neuron towards activation. We have a one-dimensional phase space and a two-dimensional parameter space.

One method is to use Sarkovskii's Thereom on the above equation and solve for $x = f(f(f(x)))$ and say "See, the equality only holds if w and b take on 'this' relationship." However, due to our neuron's activation function being transcendental and consequently, non-algebraic, $x = f(f(f(x)))$ eluded manipulation. (even intractable for Mathematica).

An entirely different (and more often used) method of proving the existence of chaos in a dynamical system is to show that it is topologically conjugate to another known chaotic dynamical system such as the shift map, the smale horsehoe map, or the logistic map [5, 9]. These three are all defined quite differently but have nonetheless been. shown to be conjugate to one another. To show topological conjugacy. between two dynamical systems $f(x)$ and $g(x)$, one must provide a. function, a homeomorphism, $h(x)$ such that $h^{-1} \circ f = g \circ h$. One might hope to provide such a homeomorphism between our single-neuron artificial. neural network and the logistic map, $x_{t+1} = \mu x(1 - x)$. Many attempts at this failed. Is there such an $h(x)$?

Claim: A network of one recurrently connected neuron in the absence on online input does not have a chaotic baseline for any combination of parameters.

Proof: Let our neural network's change function be denoted by $g(x)$ and the logistic map be denoted by $f(x)$ Suppose there exists an h such that h

is a homeomorphism between f and g. Then h must map every dynamicaly unique orbit of f onto the orbits of g. Observe that the logistic map has a critical point ($f' = 0$) at $x = \frac{1}{2}$ for all μ. This. critical point divides the domain of f into two distinct subintervals. which map to a common range. More. specifically, $f(\frac{1}{2} + \delta_1) = f(\frac{1}{2} - \delta_2)$. where $\delta_1 \neq \delta_2$ and $\delta_1 \leq \delta_2$ without loss of generality for infinitely many (although distinct). pairs of δ_1 and δ_2.

Now note that $g'(x) = (1 - tanh^2(wx + b))w \neq 0$ for all x except the trivial case where $w = 0$... Consequently, for every $x \neq -\frac{b}{w}$, there can be no y. such that $g(x) = g(y)$.

$$
\begin{aligned}
f(\frac{1}{2} + \delta_1) &= f(\frac{1}{2} - \delta_2) \\
h(f(\frac{1}{2} + \delta_1)) &= h(f(\frac{1}{2} - \delta_2)) \\
h(f(\frac{1}{2} + \delta_1)) &= g(h^{-1}(\frac{1}{2} + \delta_1) \\
g(h^{-1}(\frac{1}{2} + \delta_1)) &= g(h^{-1}(\frac{1}{2} + \delta_2) \\
\delta_1 &= \delta_2
\end{aligned}
$$

Contradiction. The only assumption was existence of h therefore there is no such h. So, f and g are not homeomorphic and therefore g is not chaotic, regardless of parameter choice.

4.1.2 Multi-neuron Networks

We are left crestfallen! Can the same be the case for networks of multiple neurons? Initially we might guess not. Henri Poincaré proved continuous (not our discrete) dynamical systems may not be chaotic in either one or two dimensions but may be in three or above. We will proceed with proof by contradiction assuming the h as above but we need first to consider the topological possibilities of this more general case.

First of all, forgetting topology, is a homeomorphism h between f the one dimensional logistic map, and our multi-neuron network g *algebraically* possible? Well, $f : \mathbb{R}^1 \to \mathbb{R}^1$, $g : \mathbb{R}^n \to \mathbb{R}^n$, where n is the number of neurons in our network and $n \neq 1$. There is no pair $< l, m >$ such that $h : \mathbb{R}^l \to \mathbb{R}^m$ is valid to consider in $h^{-1} \circ f = g \circ h$.

Instead, let's reconsider our choice of f. Other multi-dimensional discrete systems are known to be chaotic, but the logistic map is most friendly so we will stay close to home. Let f now denote an n dimensional trivial coupling of logistic maps where a separate, independent map controls the dynamics of each of the n components.

Claim: A recurrently connected network of n neurons in the absence of online input does not have a chaotic baseline for any combination of parameters.

Proof: Let our n-neuron neural network's change function be denoted by $g(x)$ and the trivial coupling of logistic maps be denoted by $f(x)$ where $f(x) =$

$$\begin{pmatrix} \mu_1 * x_1 * (1 - x_1) \\ \mu_2 * x_2 * (1 - x_2) \\ \vdots \\ \mu_n * x_n * (1 - x_n) \end{pmatrix}$$

and where $g(x) =$

$$\begin{pmatrix} tanh(a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n) \\ tanh(a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n) \\ \vdots \\ tanh(a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n) \end{pmatrix}$$

Does there exist a homeomorphism h? As before, such an h will have to map every dynamically unique orbit of f onto g and g onto f. As before, f has at least one critical point where f' equals zero and orbits on either side map to dynamically unique regions of the phase space which contributes to the 'mixing' quality of chaotic motion.

Is any component of $g'(x)$ possibly zero at some x? We can no longer deal with one-dimensional derivatives and will instead now employ the Jacobian of $g(x)$, $J(g(x))$ which gives a matrix, the entries of which constitute the derivative of every equation of the system with respect to every system variable. $J(g(x)) =$

$$\begin{pmatrix} a_{11}tanh'(a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n) & \cdots & a_{1n}tanh'(a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n) \\ \vdots & \ddots & \vdots \\ a_{n1}tanh'(a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n) & \cdots & a_{nn}tanh'(a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n) \end{pmatrix}$$

We can deduce some information about the critical points of g by looking at the eigenvalues of $J(g(x))$ which characterize the tangent space of g. Since we are looking for zero-valued eigenvalues, we can simplify our approach and look simply at the determinant which is equal to the product of the eigenvalues. We won't know which eigenvalue takes on a zero-value but we are only interested so far in the existence of such a value.

To do this, we introduce a new theorem and its proof which gives the determinant of $J(g(x))$ denoted $|J(g(x))|$.

Claim: $|J(g(x))| = |A|\Pi_{i=1}^{n} tanh'(a_i \cdot x)$ Where $|A|$ is the determinant of the weight matrix A and $a_i \cdot x$ is the dot-product of the i^{th} row of A with the state x.

The proof will proceed by induction. The base case is from a one-neuron network where the weight matrix $A = [a]$, the weight of the neuron connecting back to itself. $g(x) = tanh(ax)$, $J(g(x)) = [atanh'(ax)]$, The determinant of A, $|A| = a$. The determinant of $|J(g(x))| = atanh'(ax) = |A|tanh'(a_1 \cdot x)$.

For the inductive step, suppose the claim is true for networks of size i, we will inspect and see if it is true for networks of size $i + 1$.

We use the cofactor expansion method for finding the determinant of a matrix down the arbitrarily chosen l^{th} column.

$$|J(g(x))| =$$

$$a_{1,l} tanh'(a_1 \cdot x)|J_{1,l}| -$$

$$\dots$$

$$+a_{i+1,l} tanh'(a_{i+1} \cdot x)|J_{i+1,l}|$$

The terms $J_{k,l}$ above are the $k - j$ minors of $J(g(x))$ and are matrices of size i. Knowing this we can substitute in our assumption.

$$|J(g(x))| =$$

$$a_{1,l} tanh'(a_1 \cdot x)(|A_{1,l}|\Pi_{j=1}^{i} tanh'(a_j \cdot x)) -$$

$$\dots$$

$$+a_{i+1,l} tanh'(a_{i+1} \cdot x)(|A_{i+1,l}|\Pi_{j=1}^{i} tanh'(a_j \cdot x))$$

In the above sum of products, each Π term contains the $tanh'$ of dot-product of i of the $i + 1$ rows of A with x. The $tanh'$ of the one extra left out row, for each product, is included in the extracted cofactor. Knowing this we can safely factor out the product of the $tanh'$ of the $i + 1$ rows of A, giving $|J(g(x))|$

$$= (a_{1,l}|A_{1,l}| - a_{2,l}|A_{2,l}| + \cdots + a_{i+1,l}|A_{i+1,l}|)\Pi_{j=1}^{i+1} tanh'(a_j \cdot x)$$
$$= |A|\Pi_{j=1}^{i+1} tanh'(a_j \cdot x)$$

The inductive step is shown. *q.e.d.*

Now given this formula for the determinant of $J(g(x))$, what can we say about its eigenvalues, for $g(x)$, its coresponding critical points and its candidacy for conjugacy with $f(x)$?

None of the $tanh'$ terms will ever be zero.

Of note, $|J(g(x))| = 0$ when $|A| = 0$. This is, however, not of interest. Neural networks represented by singular weight matrices have disconnected subnetworks. No information is shared between them. Without loss of generality we can say that we might as well have considered one subnetwork or the other [1].

The nonexistance of a homeomorphism h between f and g follows from the demonstration of a lack of a critical point for g. Therefore g does not exhibit chaotic motion under any parameterization.

4.2 Nonautonomous ANNs

The above demonstration of autonomous ANNs' lack of chaotic solutions leads us to consider the more theoretically cumbersome case of nonautonomous ANNs. Dynamical systems theory has been geared towards the study of autonomous systems [9]. However, one can rethink simpler nonautonomous sytems in terms of more complex autonomous ones by considering the product space of the nonautomous sytem and its input signal. This system (of necessarily higher dimension) is autonomous in the formal sense and is now a valid object for measurement under autonomous dynamical systems theoretic metrics such as the Lyapunov spectrum.

It is worth citing [11] who characterizes the lyapunov exponents of nonautonomous sytems without the above consideration. There, in the DST divergence condition of chaotic systems, the 'small differences in the initial state

[1] Our motivation here is to show the nonexistence of a homeomorphism with a system exhibiting chaotic motion. The existence of a zero-valued derivative is a necessary condition for this, but not sufficient. The existence of a zero-valued derivative for artificial neural systems with singular weight matrices is a trivial byproduct of their networks being disconnected and will likely not meet other necessary conditions for a homeomorphism's existence

of the system' is taken to mean 'small differences in the online signal'. While this is practically relevant to building robust systems, it is not compatible with formal notions of chaos. In the proposed thesis here, we attempt to address neural systems from a more orthodox DST ground.

Previously, in the 'off-line' case for ANNs, we asked the question 'under what conditions or constraints is the behavior of an ANN to be considered chaotic?' An ANN is constituted only of its activation function and weight matrix. We assumed the hyperbolic tangent (tanh) activation then as we do now and we take the weight matrix to be composed of real numbers. Now, in the on-line and nonautonomous we must also consider an input signal which we will take to be continuous and sampled at some regular constant interval dt and which can be described as a function $f(x)$ and some initial offset x_0.

Whereas Maass measured the sensitivity of neural systems to small differences in the input, namely the input signal [11], here we will take the input signal as a static part of the system itself and measure the sensitivity (read: Lyapunov exponents) of the system and take the 'small differences' to be, as in dynamical systems theory, small differences in the initial state of the neural system.

An attempt at an analogy to a behaviorist study of humans: whereas Maass' approach would measure the difference of the same subjects' reaction to stimuli which differ only slightly, the approach proposed here would measure the difference of the same subjects' reaction to the *same* stimuli when primed by slightly different sets of stimuli preceeding the measurement.

5 Genetic Algorithm

A purely analytic approach was taken in the above section on autonomous artificial neural networks. The introduction of an input signal into the work presents challenges which make analytic results beyond the scope of a M.S. thesis. Consequently, a genetic algorithm is proposed with the express purpose of suggesting direction for future analytic research.

5.1 Goal

Typically, genetic algorithms are employed to approximate solutions to problems which are difficult to compute but easy to verify. A classic example is the travelling salesman problem: Given a graph, what is the shortest path that visits every vertex and that starts and stops on the same vertex. Computing this solution for arbitrarily large graphs is intractable but approximation with genetic algorithms is satisfactorilly fast. This is possible by virtue of the ease of quantifying the 'fitness' of potential solution which is, in the case of the travelling salesman problem, the shortness of the path.

In this proposed thesis, we are trying to find artificial neural networks and associated input signals that behave chaotically. This can be quantified by approximating the Lyapunov spectrum of the system, a task that is more computationally expensive than just measuring the shortness of a path on a graph, but a task we can compute nonetheless.

Genetic algorithms have been often applied to neural networks with the goal of creating a network that solves a particular problem [4]. Typically, the fitness is how well the network achieves the desired behavior in a simulation. In our case this is not enough and is where the proposed thesis takes on its novel character.

In Maass' publications [14], networks are randomly searched for that meet the necessary criteria. Jaeger provides a stochastic method for constructing networks [16]. The goal of this genetic algorithm will be to evolve, not a neural network, but a system of equations that define constraints on a neural network and its input, that define a *family* of neural networks and inputs. The hope is that the product of simulated evolution will provide clues as to just what it is about certain neural networks and their associated inputs that produces chaotic behavior and what role chaos can play in computing theory. This thesis began analytically and here moves towards experiment, but experiment directed expressly at future analytic attempts.

5.2 Representation

Organisms in the proposed genetic algorithm consist of a description of the size of the network, n, a variable-length conjunction of variable-length constraints on the weights of the network, and a variable-length expression of the input signal, a function $f(x)$ and an intial offset x_0.

The most managable genetic representation of variable length individuals is a binary tree. Elements of the tree are either binary operators, variable symbolic operands, or constants. Two trees will be needed to describe individuals: one for the system of contraints on the weight matrix, and another for the expression of the input signal.

Operators include all the arithmetic operators: addition, subtraction, multiplication, division, and exponentiation as well as trigonometric functions for the input signal only.

Symbolic operands apply to constraints on the network only (not the input signal) and refer to indices in the networks' weight matrices.

5.3 Crossover

Crossover is more complicated for variable-length-individual genetic algorithms, the above tree representation was chosen to facilitate a more simple crossover implementation.

Crossover can be achieved by randomly picking a branch in both parent trees and splicing together the respective roots and branches to make children. Additionally, [3] describes a new technique for crossover as applied to genetic programming (another variable-length-individual GA technique) which will be applied here as well and compared to the simpler first proposed method.

5.4 Mutation

Mutation on individuals represented as trees will amount to a random choice of a few operations: Random pruning of subtrees, random generation of an addition of new subtrees, and random flipping of bits in symbolic variables and constants.

5.5 Fitness Metric

The fitness metric is the most complex aspect of this approach and encapsulates the majority of both the work and theory.

5.5.1 Lyapunov Spectrum

Given a neural network and an associated input signal, a good metric for experimentally determining how chaotic the system is is to measure the largest Lyapunov exponent from the Lyapunov spectrum which describes the eigenvalues of the rates of expansion of a hypersphere over the infinite iteration of the system. In practice, this is theoretically incalculable but can be reasonably approximated by actually iterating the system on some randomly generated points.

A module for iterating dynamical systems and measuring lyapunov exponents constitutes a large part of the thesis.

5.5.2 Constraint Satisfaction

Our fitness metric will, however, not be given a neural network and an associated input signal, but instead a system of contsraints that describe a family of neural networks.

A constraint propagation algorithm [15] will be applied to determine the number of degrees of freedom a family of networks has. The free variables will be randomly assigned for a large number (order 100) networks. Each network will have its lyapunov spectrum measured and the mean value of the largest exponent from each system will serve as the fitness metric for 'how chaotic are the networks in this family'.

A module for solving the constraint satisfaction problem and managing instances of the family also constitute a large part of the proposed thesis.

5.6 Distributed Implementation

The constraint satisfaction problem and lyapunov spectrum measurement described above are both computationally expensive tasks. To reduce runtime, the code portion of the proposed thesis will be implemented in python and distributed across idle machines in the RIT Department of Computer Science over ssh.

Consequently, the authoring of an ssh job manager constitutes another, somewhat less major but nonetheless critical part of the proposed thesis.

6 Proposed Timeline

The following timeline allows two quarters to complete code, experiments and the final thesis document.

Completed - No chaos in autonomous networks

Completed - Thesis proposal *(Pending approval and revision).*

12/15/08 - Lyapunov Approximation module completed, begin work on CSP module.

01/15/09 - Constraint Satisfaction module completed, begin work on GA suite.

02/15/09 - Distributed GA suite completed, begin running experiments, tweaking as necessary.

04/01/09 - Finish experiments. Begin final analysis and composition of thesis.

05/??/09 - Defend.

References

[1] K. Alligood, T. Sauer, J.A. Yorke *Chaos: An Introduction to Dynamical Systems*, Springer-Verlag, 1997.

[2] D. Arrowsmith and C.M. Place, *An Introduction to Dynamical Systems*, Cambridge University Press, 1990.

[3] J. Clegg, J. A. Walker, and J. F. Miller, *A new crossover technique for Cartesian genetic programming*, Proceedings of the 9th annual conference on Genetic and evolutionary computation, 1580-1587, 2007.

[4] K. Doya, *Bifurcations in the Learning of Recurrent Neural Networks*, Proceedings of the 1992 IEEE International Symposium on Circuits and Systems, 2777-2780, 1992.

[5] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, Westview Press, 2003.

[6] A. Weigend and N. Gershenfeld, *Time Series Prediction. Forecasting the Future and Understanding the Past*, Addison-Wesley, 1994.

[7] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, The MIT Press, 2001.

[8] C. Rocsoreanu, A. Georgescu, and N. Giurgiteanu, *The Fitzhugh-nagumo model: Bifurcations and Dynamics*, Kluwer, 2000.

[9] M. W. Hirsch, S. Smale, R. L. Devaney, *Differential Equations, Dynamical System, and an Introduction to Chaos*, Elsevier, 2004.

[10] E. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*, The MIT Press, 2007.

[11] R. Legenstein and W. Maass, *What makes a dynamical system computationally powerful?* Haykin, Principe, Sejnowski, and McWhirter: New Directions in Statistical Signal Processing: From Systems to Brain 2005.

[12] T. Natschläger, W. Maass, H. Markram, *The "Liquid Computer": A Novel Strategy for Real-Time Computing on Time Series* Special Issue on Foundations of Information Processing of TELEMATIK, vol. 8, num. 1, p. 39-43, 2002

[13] W. Maass and C. M. Bishop, *Pulsed Neural Networks*, The MIT Press, 1999.

[14] W. Maass, T. Natschläger, and H. Markram, *A Model for Real-Time Computation in Generic Neural Microcircuits*, Proc. of NIPS 2002, Advances in Neural Information Processing Systems, volume 15, pages 229-236. MIT Press, 2003.

[15] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.

[16] H. Jaeger, *The "echo state" approach to analysing and training recurrent neural networks.* GMD Report 148, GMD - German National Research Institute for Computer Science, 2001.

[17] D. Verstraeten, B. Scharuwen, M. D'Haene, D. Stroobandt, *An experimental unification of reservoir computing methods*, Neural Networks, vol. 20, pgs 391-403, 2007.