On the relationship between symbolic and neural computation

Whitney Tabor and Dalia Terhesiu
University of Connecticut
406 Babbidge Drive
Storrs, CT 06269
tabor@uconnvm.uconn.edu

To appear as a 2004 AAAI Fall Symposium Series Technical Report
(see http://www.sp.uconn.edu/ ps300vc/papers.html)

Abstract

There is a need to clarify the relationship between traditional symbolic computation and neural network computation. We suggest that traditional context-free grammars are best understood as a special case of neural network computation; the special case derives its power from the presence of certain kinds of symmetries in the weight values. We describe a simple class of stochastic neural networks, Stochastic Linear Dynamical Automata (SLDAs), define Lyapunov Exponents for these networks, and show that they exhibit a significant range of dynamical behaviors—contractive and chaotic, with context free grammars at the boundary between these regimes. Placing context-free languages in this more general context has allowed us, in previous work, to make headway on the challenging problem of designing neural mechanisms that can learn them.

Introduction

Part of the interest in symbol composition lies in the fact that it supports recursive symbolic computation. A hope is that neural networks can exhibit the power of recursive computation, but do so in a way that maintains some of their advantageous properties, like graceful degradation, learning, and human-like generalization. There is also a sense that in addition to affording these desirable qualities, some different kind of insight may be offered if recursive neural network symbol processing can be made robustly successful. We suggest the following as one aspect of this "different kind of insight": symbolic grammars (i.e., Turing Machines) are a special case within the wide gamut of dynamical (i.e., real-valued, recurrent) computing devices. Context free grammars, for example, lie at the boundary between contractive and chaotic systems, which has been associated with complex processing in other work (Wolfram 1984; Langton 1992; Moore 1990; Crutchfield 1994; Kaufmann 1993). Our prior work (Tabor 2002b) suggested this correspondence by identifying contractive neural networks that generate/recognize finite state languages and edge-of-chaos networks that generate/recognize context-free languages. The present work extends these results by including the chaotic case. Along the way, we adopt a method of studying chaos in stochastic systems, demonstrating its well-behavedness. We identify a particularly simple class of networks (Stochastic Linear Dynamical Automata or SLDAs) that provides a convenient domain for studying all three dynamical regimes (contractive, edge of chaos, and chaotic) and comparing them.

History

(Elman 1990; 1991; Pollack 1990) provided promising notes with regard to getting neural networks to learn context-free languages. (Levy, Melnik, & Pollack 2000; Melnik, Levy, & Pollack 2000) recognized the role of fractal sets in (Pollack 1990)'s proposal and were able to prove convergence for a simple case.

(Wiles & Elman 1995; Rodriguez, Wiles, & Elman 1999; Rodriguez 2001) made headway in understanding how symbol counting could work in Elman's Simple Recurrent Networks (SRNs) and (Rodriguez 2001) found that a linear approximation of the solution could be extracted which revealed the network's counting technique. (Gers & Schmidhuber 2001) have created a very robust counting unit which can learn dependencies of great length.

So far, the results only handle linear and quadratic state-growth languages. Exponential state growth languages (where the number of *causal states* (Shalizi, Shalizi, & Crutchfield 2002) grows exponentially with string length; palindrome languages are an example) are more difficult, but clearly central to interesting applications (e.g. human language).

Meanwhile, (Moore 1996) showed how to recognize context-free grammars with dynamical recognizers with piecewise linear activation functions. (Tabor 2000) described a similar mechanism, pushdown dynamical automata (PDDAs), and suggested that such encodings might be a reachable state for complex grammar learning. (Tabor 2002b) provided evidence for this view by using a variant on a classification technique from dynamical systems theory, Lyapunov Exponents (Oseledec 1968; Abarbanel, Gilpin, & Rotenberg 1996). Lyapunov Exponents measure the average expansiveness of the points along a trajectory of a dynamical system. Among discrete, deterministic dynamical systems, the maximal Lyapunov Exponent is negative for periodic attractors and positive for chaotic attractors, so the boundary

between the periodic and chaotic regimes is at 0. Under a proposed extension of the definition to stochastic systems, (Tabor 2002b) showed that the maximal exponent was 0 for probabilistic PDDAs. Moreover, the maximal exponents for SRNs trained (only partially successfully) on stack- and queue-based languages were much closer to 0 than the maximal exponents for finite-state languages.

These results suggest (1) it may be possible to design learning algorithms that converge on zero-Lyapunov solutions like the PDDAs; (2) neural networks seem to fit the picture sketched by (Wolfram 1984; Moore 1990; Langton 1992), in which the computations of Turing machines lie in a special section of dynamical computation space on the border between periodic and chaotic processes (the so-called "edge of chaos")—see also (Siegelmann 1999; Griffeath & Moore 1996).

Regarding (1), (Tabor 2003; 2002a) described a learning network called a Fractal Learner which shows, in simulations, evidence of convergence on certain PDDAs, including exponential state growth cases.

Regarding (2), it must be noted that the results of (Tabor 2002b) only provide evidence for part of the proposed picture. That is, they show neural networks in the contractive and edge of chaos domains. They do not show neural networks in the chaotic domain.

The present paper introduces a construction, Stochastic Linear Dynamical Automata (SLDAs), that helps flesh out the picture for one simple class of neural dynamical systems: one-dimensional (i.e., one-hidden unit) maps with linear activation and gating functions. We (i) sketch a method of proving that the extended definition of Lyapunov Exponents given in (Tabor 2002b) is well-behaved in the sense of being independent of initial state and the randomness of the process and (ii) show that all three dynamical types (periodic, edge-of-chaos, and chaotic) are present among SLDAs. The results suggest a route to checking the proposed correspondences between Turing and dynamical computation.

Construction

Lyapunov Characteristic Exponents (Oseledec 1968; Abarbanel, Gilpin, & Rotenberg 1996) measure the average rate of contraction or divergence of trajectories near the attractor of a dynamical system. Let

$$\vec{h}_{t+1} = f(\vec{h}_t) \tag{1}$$

be a discrete, deterministic dynamical system with N-dimensional state, $\vec{h}$. The Lyapunov Exponents, λ_i for $i = 1, \ldots, N$, of the trajectory starting at $\vec{h}$ are the logarithms of the eigenvalues of the matrix

$$OSL(\vec{h}) = \lim_{t\to\infty} ((Df_t)^T (Df_t))^{\frac{1}{2t}}(\vec{h}) \tag{2}$$

where T denotes transpose, $Df(\vec{h})$ is the Jacobian of f at $\vec{h}$. For $\vec{h}$ in a single attractor, the values of the eigenvalues are essentially independent of the choice of $\vec{h}$ so we may speak of the Lyapunov exponents of the attractor. If all the exponents are negative, then the system is a limit cycle and visits only finitely many points. If at least one exponent is positive and the system is bounded, then the system is chaotic. The case in which the greatest Lyapunov exponent is 0 in a discrete system is a special case which can yield complex behavior (famously for the logistic map at the "edge of chaos"—(Crutchfield & Young 1990)).

To extend (2) to stochastic dynamical systems, we let the definition of eigenvalues depend not only on the initial condition, but also on the specific random sequence of inputs, S, that the autonomously functioning network chooses:

$$OSL(\vec{h}, S) = \lim_{t\to\infty} ((Df_t)^T (Df_t))^{\frac{1}{2t}}(\vec{h}) \tag{3}$$

(Tabor 2002b) hypothesized that the logarithms of the eigenvalues of this matrix are essentially independent of the choice of $\vec{h}$ and corresponding symbol sequences S.

A simple case: Stochastic Linear Dynamical Automata (SLDAs).

Consider the following class of dynamical automata:[1]

$$M = (H, F, P, \Sigma, IM, \vec{h}_0, FR) \tag{4}$$

where $H = \{\vec{h} \in \mathbf{R^N} : 0 \leq h_j \leq 1, j \in \{1, \ldots, N\}\}$, $F = (\vec{m}, \vec{b})$ is the set of K linear functions, $\{F_i : \mathbf{R} \to \mathbf{R} : F_i(\vec{h}) = \vec{m}_i \cdot \vec{h} + \vec{b}_i, 0 \leq h_j, (F_i)_j(\vec{h}) \leq 1, i \in \{1, \ldots, K\}, j \in \{1, \ldots, N\}\}$, P is a partition of H the boundaries of whose compartments are the boundaries of the domains of the F_i, $\Sigma = \{1, \ldots, K\}$ is the alphabet of symbols, IM, the input map, specifies that when $\vec{h}$ is in the domain of F_i then a possible next symbol is i and the corresponding next state is $F_i(\vec{h})$, $\vec{h}_0 \in H$ is the initial state, and $FR \in H$ is the final region.

Here, we focus on a related class of devices

$$MP = (H, F, P, \Sigma, IM, \vec{h}_0, PD) \tag{5}$$

where there is no final region (so the machine starts at $\vec{h}_0$ and runs forever) and where $PD : H \to F$ is a probability function which assigns probabilities to choices when the map is nondeterministic. We say that the probabilities are properly assigned if some function in F has positive probability at every point in H. We call these stochastic maps Stochastic Linear Dynamical Automata (SLDAs).

One-dimensional SLDAs can be depicted graphically in a transparent way. We take languages to be distributions of strings. Table 1 shows a grammar for generating Language 1, a context free language which is not a finite state language. Figure 1 shows an SLDA for processing Language 1.

Sketch of invariance demonstration for SLDAs

(See http://www.sp.uconn.edu/~ps300vc/papers.html#eocnn for details.)

[1] Dynamical automata are defined in (Tabor 2000).

Table 1: Grammar 1, which generates Language 1. Parentheses denote optional constituents, which occur with probability 0.5. A slash between two productions means they each occur with probability 0.5. Generation of a string is accomplished by expanding S. The grammar generates an infinite string by repeatedly generating a finite string and appending it to its previous output.

S → AB/XY
A → a (S) X → x (S)
B → b (S) Y → y (S)

We would like to know if the Lyapunov exponents which we measure for SLDAs are a reliable indicator of the dynamical status of the SLDAs. To this end, we ask if the values of the Lyapunov exponents are independent of the choice of initial state, $\vec{h}_0$ and its corresponding symbol sequence, $S \in \Sigma^\infty$. We show that there is an invariant probability measure on the product space, $H \times S^\infty$. This means that the case at hand falls under a generalization of (Oseledec 1968)'s Mulitplicative Ergodic Theorem to discrete-time, stochastic systems (Arnold 1998). This guarantees the existence of Lyapunov exponents. We also show that the invariant measure is unique. Therefore, the Lyapunov exponents are independent of the initial state, h_0, for almost all choices of initial state (Walters 1986). Note that in the one-dimensional case, there is only one Lyapunov exponent for each system, given by

$$\lambda = lim_{n\to\infty}\frac{1}{n} log\,|m_n| \tag{6}$$

where m_n is the slope of the function applied on the n'th iteration.

SLDAs as Recurrent Neural Networks (RNNs)

An SLDA can be interpreted as a layered neural network. The coefficients of the F_i correspond to first and second-order weights on connections into the second layer. Figure 2 shows the neural network version of the SLDA in Figure 1.

Various dynamical behaviors observed

By analogy with the deterministic case, we call an SLDA whose Lyapunov exponent is negative a *contractive SLDA*, and an SLDA whose Lyapunov exponent is positive a *chaotic SLDA*. Some examples show that there exist contractive, edge-of-chaos, and chaotic devices within the set of 1-dimensional SLDAs.

Example 1. A context free grammar at the edge of chaos. Under the definition discussed in the previous sections, the Lyapunov exponent of SLDA1 is 0. This result can be derived by noting that SLDA1 satisfies the definition of a Push-down Dynamical Automaton (Tabor 2000) and therefore undergoes equal contraction and expansion during the processing of any well-formed sentence. This implies convergence of the Lyapunov exponent on 0 as n $\rightarrow \infty$. Measurement of the value on a corpus of 10000 words produced the value -0.0123. In fact, 0 is the maximum Lyapunov exponent that this system can achieve over all proper assignments of probabilities to transition functions.

Example 2. A finite state contractive mapping. Setting,

$$(\vec{m}, \vec{b}) = \begin{bmatrix} 1/2 & 0 \\ 1/2 & 1/2 \end{bmatrix} \tag{7}$$

where both functions are equally probable for all h yields an exponent of -log 2 (measured value -0.6931). Note that this case generates the same language as the finite state machine with one state and two equally probable paths from the state to itself. Note that the maximum Lyapunov exponent over all assignments of probabilities for this case is negative (= -log 2).

Example 3. A chaotic case. Let

$$(\vec{m}, \vec{b}) = \begin{bmatrix} 1/2 & 0 \\ 1/2 & 1/2 \\ 2 & 0 \\ 2 & -1 \end{bmatrix} \tag{8}$$

with $P = \{[0, 1/2), [1/2, 1]\}$ and the probability of each contractive map equal to $\frac{1}{6}$ and the probability of each expansive map equal to $\frac{2}{3}$ on its interval. Then the Lyapunov exponent is (log 2)/3 (measured value 0.2377). Since this number is greater than 0 and the system is bounded, the automaton is chaotic. Note that the maximum of the Lyapunov exponent over all assignments of probabilities is positive in this case (= log 2).

Conclusions

These cases are consistent with the conjecture of (Tabor 2002b) that contractive dynamical automata correspond to probabilistic finite-state automata while Turing computers lie on the boundary separating contractive from chaotic dynamical automata. An advantage of the current framework is that the full range of dynamical behaviors is exhibited in a very simple class of functions. Desirable and possibly reasonable goals for future work are (i) to fully establish the correspondences just mentioned for SLDAs, (ii) to probe the implications of the sensitivity of the Lyapunov exponents to assignments of probabilities to transition functions, (iii) to provide a symbolic characterization of the chaotic cases, and (iv) to generally establish the convergence of a learning algorithm along the lines of (Tabor 2003). An interesting idea, suggested by the foregoing, is that complex cognitive computation is rare because it depends on the occurrence of rare symmetries in neural parameter space.

References

Abarbanel, H. D. I.; Gilpin, M. E.; and Rotenberg, M. 1996. *Analysis of Observed Chaotic Data*. New York: Springer-Verlag.

Arnold, L. 1998. *Random Dynamical Systems*. Berlin: Springer Verlag.

Crutchfield, J. P., and Young, K. 1990. Computation at the onset of chaos. In Zurek, W. H., ed., *Complexity, Entropy,*

and the Physics of Information. Redwood City, California: Addison-Wesley. 223–70.

Crutchfield, J. P. 1994. The calculi of emergence: Computation, dynamics, and induction. *Physica D* 75:11–54. In the special issue on the Proceedings of the Oji International Seminar, *Complex Systems—from Complex Dynamics to Artificial Reality*.

Elman, J. L. 1990. Finding structure in time. *Cognitive Science* 14:179–211.

Elman, J. L. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning* 7:195–225.

Gers, F. A., and Schmidhuber, J. 2001. Lstm recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks* 12(6):1333–1340.

Griffeath, D., and Moore, C. 1996. Life without death is p-complete. *Complex Systems* 10:437–447.

Kaufmann, S. 1993. *Origins of Order*. Oxford: Oxford University Press.

Langton, C. 1992. Life at the edge of chaos. In Langton, C. G.; Taylor, C.; Farmer, J. D.; and Rasmussen, S., eds., *Artificial Life II*. Addison-Wesley. 41–91. Proceedings of the workshop on artificial life held February, 1990 in Santa Fe, New Mexico. Proceedings Volume X, Santa Fe Institute Studies in the Sciences of Complexity.

Levy, S.; Melnik, O.; and Pollack, J. 2000. Infinite raam: A principled conectionist basis for grammatical competence. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum Associates. 298–303.

Melnik, O.; Levy, S.; and Pollack, J. B. 2000. RAAM for infinite context-free languages. In *Proceedings of the International Joint Conference on Artificial Neural Networks (IJCNN)*. IEEE.

Moore, C. 1990. Unpredictability and undecidability in dynamical systems. *Physical Review Letters* 64:2354–2357.

Moore, C. 1996. Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science* 162(1):23–44.

Oseledec, V. I. 1968. A multiplicative ergodic theorem. lyapunov characteristic numbers for dynamical systems. *Trudy Mosk. Mat. Obsc.* 19:197.

Pollack, J. B. 1990. Recursive distributed representations. *Artificial Intelligence* 46:77–106. Special issue on Connectionist symbol processing edited by G. E. Hinton.

Rodriguez, P.; Wiles, J.; and Elman, J. 1999. A recurrent neural network that learns to count. *Connection Science* 11(1):5–40.

Rodriguez, P. 2001. Simple recurrent networks learn context-free and context-sensitive languages by counting. *Neural Computation* 13(9).

Shalizi, C. R.; Shalizi, K. L.; and Crutchfield, J. P. 2002. Pattern discovery in time series, part i: Theory, algorithm, analysis, and convergence. Santa Fe Working Paper #02-10-060.

Siegelmann, H. T. 1999. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Boston: Birkhäuser.

Tabor, W. 2000. Fractal encoding of context-free grammars in connectionist networks. *Expert Systems: The International Journal of Knowledge Engineering and Neural Networks* 17(1):41–56.

Tabor, W. 2002a. Fractal learning neural networks. Unpublished Manuscript, University of Connecticut.

Tabor, W. 2002b. The value of symbolic computation. *Ecological Psychology* 14(1/2):21–52.

Tabor, W. 2003. Learning exponential state-growth languages by hill climbing. *IEEE Transactions on Neural Networks* 14(2):444–446.

Walters, P. 1986. Unique ergodicity and random matrix products. In Arnold, L., and Wihstutz, V., eds., *Lecture notes in mathematics/Lyapunov exponents*, number 1186, 37–56. Heidelberg: Springer Verlag.

Wiles, J., and Elman, J. 1995. Landscapes in recurrent networks. In Moore, J. D., and Lehman, J. F., eds., *Proceedings of the 17th Annual Cognitive Science Conference*. Lawrence Erlbaum Associates.

Wolfram, S. 1984. Universality and complexity in cellular automata. *Physica D* 10:1–35.

Figure 1: Graphical depiction of SLDA1, a one-dimensional Linear Dynamical Automaton that generates Language 1. The initial state (h_0) is 0.5; $F_1 : [0,1] \rightarrow [0,\frac{1}{3}] = \frac{1}{3}h, F_2 : [0,1] \rightarrow [\frac{2}{3},1] = \frac{1}{3}h + \frac{2}{3}, F_3 : [0,\frac{1}{3}] \rightarrow [0,1] = 3h, F_4 : [\frac{2}{3},1] \rightarrow [0,1] = 3h - 2$; $P_1 = [0,\frac{1}{3}], P_2 = [\frac{1}{3},\frac{2}{3}], P_3 = [\frac{2}{3},1]; PD_1 = (0.25, 0.25, 0.5, 0), PD_2 = (0.5, 0.5, 0, 0), PD_3 = (0.25, 0.25, 0, 0.5)$. $PD_i = (p_1, \ldots p_K)$ means that in P_i, the probability of selecting F_j is p_j, for $1 \leq j \leq K$.

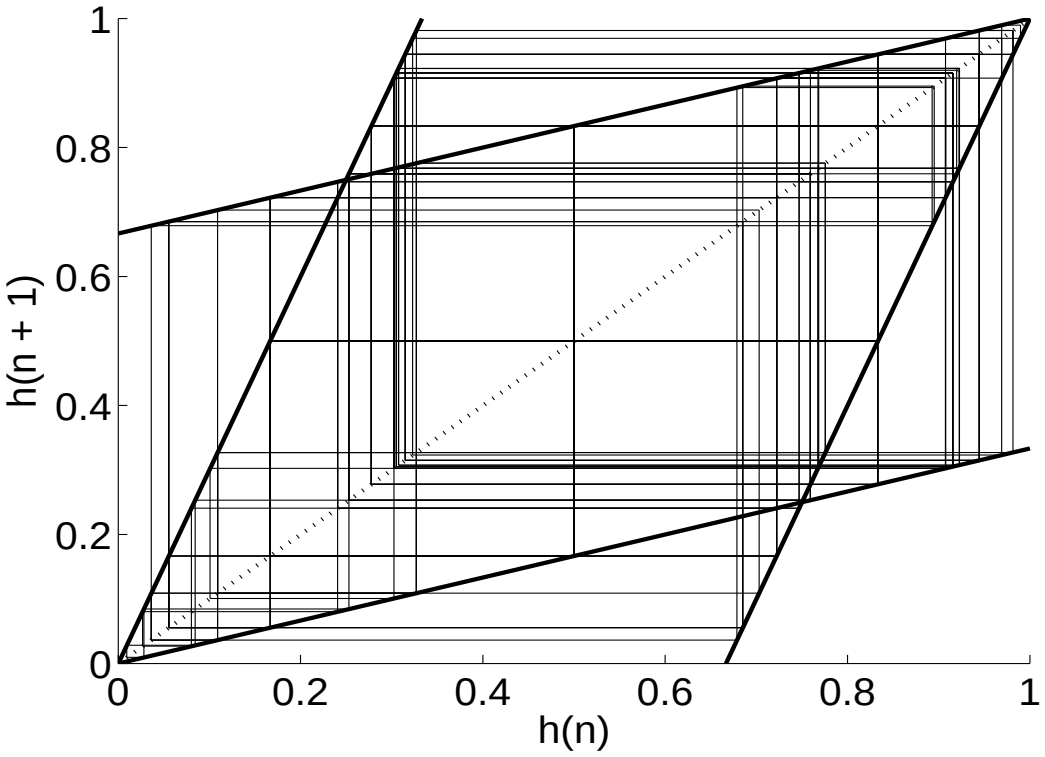

Figure 2: Neural network implementation of SLDA 1. The second order connections from the input units to the hidden layer specify the weight on the hidden unit self-connection. The hidden unit activation function is identity. The output units are threshold units with thresholds indicated on them. The outputs are binary codes. An additional layer can be added to compute probabilities as a normalization of the weighted binary codes. The next word is picked by sampling the output distribution.