[20] Pollack J.B., "The Induction of Dynamic Recognizers," *Machine Learning*, Vol. 7 (1991), pp. 227-252.

[21] Tsung, F-S., Cottrell, G.W. and Selverston, A.I., "Some experiments on learning stable network oscillations," *International Joint Conference on Neural Networks, San Diego, IEEE*, (1990).

[22] Tsung, F-S. and Cottrell, G.W., "Learning Temporal Signals in Phase Space," *International Symposium on Nonlinear Theory and Its Application, Hawaii*, (1993).

[23] Selverston, A.I., and Moulins, M., "Oscillatory neural networks," *Ann. Rev. Physiol.*, Vol. 47, (1985), pp. 29-48.

[24] Siegelmann, H.T., Sontag, E.D., "On the Computational Power of Neural Networks," in *Proceedings of the Fifth ACM Workshop on Computational Learning Theory*, Pittsburgh, PA, (1992).

[25] Wang X., "Period Doublings to Chaos in a Simple Neural Network: An Analytical Proof," *Complex Systems*, Vol. 5(4) (1991), pp. 425-441.

[26] Watrous, R.L. and Kuhn, G.M., "Induction of Finite-State Languages Using Second-Order Recurrent Networks," *Neural Computation*, Vol. 4 (1992), pp. 406-414.

[27] Wiggins, S., *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, Springer-Verlag, New York, 1990.

[28] Williams, R.J. and Zipser, D., "A Learning Algorithm for Continually Running Fully Recurrent Networks," *Neural Computation*, Vol. 1 (1989), pp. 270-280.

Second-Order Recurrent Neural Networks," *Neural Computation*, Vol. 4 (1992), 393-405.

[9] Giles, C.L., Sun, G.Z., Chen, H.H., Lee, Y.C., Chen, D., "Higher Order Recurrent Networks and Grammatical Inference," *Advances in Neural Information Processing Systems 2*, D.S. Touretzky (ed), Morgan Kaufmann, San Mateo, CA, (1990), p. 380.

[10] Hertz, J., Krough A., and Palmer R.G., *Introduction to The Theory of Neural Computation*, Addison-Wesley Publishing Company, Menlo Park, California, 1991.

[11] Kalman, R.E., Farb, P.L., and Arbib, M.A., *Topics in Mathematical Systems Theory,* McGraw-Hill, New York, 1969.

[12] Kolen, J.F., "Fool's Gold: Extracting Finite State Machines From Recurrent Network Dynamics," *Advances in Neural Information Processing Systems 6*, J.D. Cowan, G. Tesauro, J Alspector (eds.), Morgan Kaufmann, San Mateo, CA, (1994), p. 501.

[13] Kristan, W.B.,Jr., "Generation of rhythmic motor patterns," *Information Processing in the Nervous System*, (1980), pp. 241-261.

[14] Lewis, R.L., Papadimitriou, C.H., *Elements of the Theory of Computation*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

[15] Milnor, J., "On the Concept of Attractor," *Communications in Mathematical Physics*, Vol. 99 (1985), pp. 177-195.

[16] Munkres, J.R., *Topology: a first course*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.

[17] Narendra, K.S. and Parthasarathy, K., "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, Vol. 1, (1990), pp. 4-27.

[18] Nerode, A., "Linear Automata Transformations," *Proceedings of the American Mathematical Society,* Vol. 9, (1958), pp. 541-544.

[19] Omlin, C., and Giles, C., "Constructing Deterministic Finite-State Automata in Sparse Recurrent Neural Networks," in *IEEE International Conference on Neural Networks (ICNN'94)*, (1994), pp. 1732-1737.

sequence will have an accumulation point. But, since the distance from points in the sequence to points in V goes to zero by the fact that they are chosen from partition states of vanishing size, that accumulation point is also an accumulation point of V, contradicting the assumption that U and V are disjoint. So we have that there are only a finite number of partition parameters which yield partition states containing points in both U and V.

References

[1] Arbib, M.A., and Zeiger, H.P., "On the Relevance of Abstract Algebra to Control Theory," *Automatica,* Vol. 5 (1969), pp. 589-606.

[2] Casey, M., (1993) "Computation Dynamics in Discrete-Time Recurrent Neural Networks," in UCSD's Institute for Neural Computation's *Proceedings of the Annual Research Symposium*, June (1993), pp 78-95.

[3] Casey, M., "Relaxing The Symmetric Weight Condition for Convergent Dynamics in in Discrete-Time Recurrent Networks," Tech. Report UCSD-MATH-95-2, University of California, San Diego, La Jolla, March 1995.

[4] Cleermans, A., McClelland, J.L., and Servan-Schreiber, D., (1989) "Encoding Sequential Structure in Simple Recurrent Networks," Tech. Rept. CMU-CS-88-183, Carnegie-Mellon University, Pittsburgh, November 1988.

[5] Cummins, F., "Representation of Temporal Patterns in Recurrent Neural Networks," Preprint, submitted to *The* 15^{th} *Annual Conference of the Cognitive Science Society.* (1993).

[6] Devaney, R.L., *An Introduction to Chaotic Dynamical Systems*, Addison-Wesley Publishing Company, Inc., New York, 1987.

[7] Elman, J.L., "Representation and Structure in Connectionist Models," Tech. Rept. 8903, Center for Research in Language, University of California, San Diego, La Jolla, CA, August 1989.

[8] Giles, C.L., Miller, C.B., Chen, D., Chen, H.H., Sun, G.Z., Lee, Y.C., "Learning and Extracting Finite State Automata with

Lemma 7.2 *Let f be a continuous function of $\mathcal{I}^N$ into itself, and let B be an closed set of $\mathcal{I}^N$ with nonempty interior and define T to be the set of all $x \in B$ such that no ϵ-pseudo-orbit of x ever leaves B. Then the closure of T is a trapping region.*

Proof: To see that T is open, let $x \in T$ and let N be an ϵ-neighborhood of $f(x)$. Note that N is contained in B, since otherwise x could map out of B in one time step, which would contradict our assumption that x is in T, and N is in T since all points in N are in a possible ϵ-pseudo-orbit of x. By continuity of f, $f^{-1}(N) \bigcap B$ contains a neighborhood of x (since $f^{-1}(N)$ is open) and $f^{-1}(N) \bigcap B$ is in T since $f(f^{-1}(N) \bigcap B)$ is contained in N, which is contained in T.

Let C be the closure of T. To see that $f(C)$ is contained in T, and hence that C is a trapping region, first observe that if $x \in T$, then $f(x)$ is in T by the definition of T. Now suppose that x is in the boundary of C and let N be an ϵ-neighborhood of $f(x)$. Then $f^{-1}(N) \bigcap T$ is nonempty by the continuity of f and the openness of T, so there exists a $y \in f^{-1}(N) \bigcap T$. $f(y) \in N$, therefore $f(x)$ lies in an ϵ-pseudo-orbit of y. Hence $f(x)$ is in T since y is in T. ■

Lemma 7.3 (The Wandering Lemma) *Suppose some closed set B with nonempty interior contains no behaviors. Then for any $\epsilon > 0$ and for any point $x \in B$ there is some finite number N such that the N^{th} point in some ϵ-pseudo-orbit of x will leave B.*

Proof: Assume there exists an $x \in B$ such that no ϵ-pseudo-orbit of x ever leaves B, then T (from the previous lemma) is nonempty, and therefore its closure is a trapping region, which is a contradiction since the existence of a trapping region implies the existence of at least one behavior. ■

Lemma 7.4 *Let U and V be disjoint compact subsets of $\mathcal{I}^N$, then if we divide $\mathcal{I}^N$ into smaller N-dimensional cubes, called partition states, having sides of length 2^m, there exists a finite m such that none of the partition states contain points in both U and V.*

Proof: Consider a sequence of points generated by increasing the partition parameter by powers of two and then choosing one point in U from any partition state that contains points in both U and V. If we can define an infinite sequence this way, then since U is compact our

the problems which it faces.

Acknowledgements

I would like to thank Professor Gary Cottrell, Dr. Kimberly Claffy and two anonymous reviewers for making helpful comments during the preparation of this work. I would like to thank Professor Michael Freedman for his inspiration and guidance, and I would like to thank Professor Arnold J. Mandell for his nurturing and many useful discussions over the last several years.

7 Appendix

Lemma 7.1 *A behavior, $\mathcal{A}$, of a continuous map, f of $\mathcal{I}^N$ into itself, is an open set.*

Proof: Denote the closure of $\mathcal{A}$ by C. First we show that $\mathcal{A}$ is equal to $\mathcal{B}$ the union of the ϵ-pseudoperiodic points of period n for all $n \geq 0$ if the map is restricted to C. $\mathcal{A}$ is contained in $\mathcal{B}$ since $\mathcal{A}$ is a subset of $\mathcal{R}_\epsilon(f)$ and $\mathcal{B}$ is equal to $\mathcal{R}_\epsilon(f)$ when f is restricted to C. To see that $\mathcal{B}$ is contained in $\mathcal{A}$ suppose that there is some point $x \in \mathcal{B}$ which is not contained in $\mathcal{A}$. Since $\mathcal{B}$ is contained in C and C is contained in the ϵ-pseudobasin of attraction of $\mathcal{A}$, since x is not in $\mathcal{A}$, all of x's ϵ-pseudo-orbits are eventually contained in $\mathcal{A}$. But if x is in $\mathcal{B}$, then x is ϵ-pseudoperiodic for some n, and hence not all of its ϵ-pseudo-orbits are contained in $\mathcal{A}$, which is a contradiction. It follows from these conclusions that $\mathcal{A}$ is equal to $\mathcal{B}$, so it is left to show that $\mathcal{B}$ is open.

$\mathcal{B}$ is the union of the ϵ-pseudoperiodic points of period n for all $n \geq 0$, so it will be open if every ϵ-pseudoperiodic point has an open neighborhood containing only pseudoperiodic points. To see that this is true, let x be a point which is ϵ-pseudoperiodic of period n, and $\underline{x} = \{x, x_1, \ldots, x_{n-1}, x_n = x\}$ be its ϵ-pseudoperiodic orbit. By the continuity of f, we may choose an open neighborhood, M, of x which is mapped within the ϵ-ball about x_1, So any point $y \in M$ has an ϵ-pseudo-orbit of the form $\{y, x_1, \ldots, x_{n-1}, x_n = x\}$, so if y is within ϵ of $f(x_{n-1})$, then y is ϵ-pseudoperiodic. Let L denote the ϵ-ball about $f(x_{n-1})$, then we have that $x \in L$, so $x \in (L \bigcap M)$. Hence, if we choose a $y \in (L \bigcap M)$ it will be ϵ-pseudoperiodic. $(L \bigcap M)$ is the open neighborhood of ϵ-pseudoperiodic points that we seek, hence $\mathcal{B}$ is open. ■

(see Williams and Zipser (1989) [28]), and roughly what distribution of training strings to use. Our approach also allows for exploration of DFA learnability without resorting to empirical results from training attempts. We also describe a finite series of tests that determine whether the RNN solution will generalize to all possible unseen inputs.

We feel that restricting our problem domain to modeling DFA does not severely limit the applicability of our theory, because many of the problems which RNNs have solved can be recast as DFA problems. For example, learning Reber grammars as in Cleermans *et al.* (1989) [4], and many of the problems in Williams and Zipser (1989) [28] are essentially DFA problems. Furthermore, it is not difficult to see, following the work of Siegelmann and Sontag (1991) [24], that DFA are the only class of computational models that can be robustly modeled by unaugmented RNNs. Another class of problems that RNNs can solve is pattern production, which has been studied by Tsung, Cottrell and others (Williams and Zipser (1989) [28], Tsung, Cottrell and Selverston (1990) [21], Tsung and Cottrell (1993) [22], Narendra and Parthasarathy (1990) [17]).

The difficulty of learning a DFA from examples of desired behavior (see Giles *et al.* (1992) [8]) serves as evidence that careful studies of behavior and useful classification of problems solved by organisms is a nontrivial aspect of the analogous approach to understanding computation in nervous systems. Once we find a task representation of the tasks performed by "intelligent" organisms (the analogue of the DFA diagram) that enables one to infer internal dynamics from studying the representation, we should be able to predict the organization of nervous system behaviors (on some level) by understanding the mapping from nervous system behavior to organism behavior, and by understanding the logical structure of the problem to be solved (nontrivial even in the case of DFA). One is then in a better position to measure nervous systems in order to find those aspects of nervous system behavior that are essential for computation as opposed to those essential for biological maintenance or some other function. The fact that more than one pattern of nervous system behavior may produce the same pattern of organism behavior (Kristan (1980) [13] and Selverston and Moulins (1985) [23] are examples) suggests that even with the added insight of knowing (on some level) what we seek, it is likely to still be nontrivial to map that onto actual nervous system functioning that is amenable to measurement. But there is always hope that the nervous system has a small number of standard "tricks" or methods for solving

have returned to q_i. It follows that q_i is in an m-cycle of $\mathcal{M}$'s minimal diagram. ■

The robustness assumption is not essential to the argument. It is sufficient that the attractor have a dense orbit. If we are not requiring robustness, then we also might want to consider more general invariant sets than attractors, and, again, a dense orbit is sufficient for this result. ($x_{t+1} = 4x_t(1 - x_t)$, for example, does not have a chaotic attracting set, but rather has only a chaotic invariant set.)

The examples which Kolen gives in Kolen (1994) [12] are all maps with dense orbits in their invariant sets, and hence if they were a part of an RNN modeling a DFA, the invariant sets would correspond to loops in the DFA diagram. Sensitivity to initial conditions itself does not pose any problem for DFA extraction. If the reachable points in a partition state are mapped to more than one other partition state and the RNN models some DFA, then the target partition states must be equivalent.

6 Conclusions

We now have an outline for using the cognitive science approach to understanding the workings of RNNs. First, we gain a clear understanding of the problem to be solved by casting it in the form of a language recognition problem and by finding the minimal DFA that recognizes the language. (Of course, this is a very difficult problem in general.) The fact that many problems that people have trained RNNs to solve can be so described (a notable exception being general pattern production, see Hertz et al. (1991) [10] pp. 63–67) suggests that minimal DFA are a good representation of many problems of interest to connectionists. Second, one analyzes the DFA in a manner that is suited to making connections between the problem and the dynamics of a family of dynamical systems that can model the DFA. Next, one studies the mapping between hidden unit activations and the production of the desired output (trivial for the RNNs studied here, but more complex when one moves to Elman networks, for example). Finally, one measures the RNN to make sure that it has the dynamics necessary for the computation.

Furthermore, the approach outlined in this paper sometimes allows for informed guesses about how to train RNNs with respect to the size a RNN must be in order to model a DFA, whether to use teacher-forcing

nondeterministic FSM with few states or a deterministic FSM with a large number of states. He further argues that no matter how small one partitions the state space, sensitivity to initial conditions will eventually force the extracted state to split into multiple trajectories independent of the future input sequence, which suggests a nondeterministic interpretation of the state.

One way to resolve this problem is to require that the extracted FSM be deterministic. In this case, for any behavior that is a part of the computation [5], will correspond to a loop in the minimal DFA diagram, so there is no problem with sensitivity to initial conditions causing states to split. We will need another definition before formally stating and proving this fact.

Definition 5.1 *A point in the RNN phase space is said to be reachable if it is in the image of the e-map , e-map($\mathcal{I}^N$), or in I-map(e-map($\mathcal{I}^N$)) for some $I \in \{0,1\}^*$. A behavior is called reachable if any of its points are reachable.*

Theorem 5.2 *Suppose that a RNN robustly models a DFA, $\mathcal{M}$, and one of its I-maps has a behavior which is reachable, then the behavior corresponds to a cycle in $\mathcal{M}$'s minimal state diagram.*

In order for the single attractor to represent more than one state, its I-map would have to satisfy the condition that $(I\text{-map})^n$ has n distinct attractors for some $n > 1$, as in the case of an attracting periodic orbit or an attractor where each point in an attracting periodic orbit bifurcates into an invariant circle with a dense orbit (which one would find after a Hopf bifurcation of the n^{th} composition of the map with itself).

Proof: If the behavior is reachable, then there is some point, x, in the behavior which is in S-map(e-map($\mathcal{I}^N$)) for some string S. It follows that x is in Q_i for some i, else the RNN will not model $\mathcal{M}$. Suppose further that there is no number of I strings which $\mathcal{M}$ can read in order to return to q_i (i.e. q_i is not in an n-cycle for the I-map). But, since x is in a behavior of the I-map, it is ϵ-pseudoperiodic for I-map, and hence there is an m and a pseudo-orbit of x such that the m_{th} iterate of x is x, and hence in Q_i. It follows that if the RNN models $\mathcal{M}$ and $\mathcal{M}$ starts in state q_i, then after $\mathcal{M}$ reads m copies of I, $\mathcal{M}$ will

[5] since the Q_i will not fill the state space when there are more than one of them, it is possible for the maps to have behaviors in the complement of the union of the Q_is which have nothing to do with the computation being performed

process to correct the problem of having points in one partition state being sent to more than one partition state. To correct it we make all partition states that are in the range of one partition state equivalent. We need to check that this equivalence is well-defined, which follows from the following theorem:

Theorem 5.1 *For RNNs that robustly model a given DFA, for sufficiently low levels of noise in the model and sufficiently large q, it is sufficient to consider only partitions created by dividing each unit's range into q equal partitions to always succeed in extracting the DFA that is being modeled.*

This may be a nontrivial statement given that the basins of dynamical systems can have very complicated fractal structure.

Proof: Theorem 3.1 says that there are a finite number of mutually disjoint, closed (hence compact) subsets of $\mathcal{I}^N$, Q_i, that correspond to the states of the DFA. By lemma 7.4 of the appendix, we can separate any of the Q_i from the others since their union is a compact set if the the partion parameter is a power of two greater than or equal to some 2^{N_i}. Hence, if we choose the largest of these N_i then no partition state containing points in one of the Q_i will contain points in any of the others, and therefore they will be separated from one another. ■

Theorem 3.1 then implies that for a sufficiently large partition parameter the partition states containing points in any Q_i will not contain points in any Q_j for i not equal to j.

To see that the second of the conjectures in Giles *et al.* (1992) [8], that the extracted minimal DFA will be independent of the choice of the partition parameter for well-trained RNNs, is false, choose a DFA with n states, for $n > 2$, that can be modeled by a one-dimensional RNN. In order to extract the correct DFA, the partition must separate the Q_i in the RNN phase space. Hence, ther second conjecture is false since we can choose $q = 2$. We replace this second conjecture with the conjecture that there may be only a finite number of partition parameters that yield different minimal DFA for a RNN robustly modeling a DFA, in which case their conjecture is true for q sufficiently large.

John Kolen (1994) [12] has recently raised questions concerning finite state machine extraction in RNNs which have chaotic dynamics. The problem which has been suggested is that extraction systems will fail for systems displaying sensitive dependence on initial conditions (Devaney (1987) [6]) since such systems can be interpreted as either a

ject region (hence we correctly reject words containing two consecutive zeros). This is shown in the figure by showing the image of a circle initially in the region of phase space to which the e-map sends all states. The 1-map complements the dynamics of the 0-map so that the composition of the two (the 01-map, will have a fake period two orbit in the accept region, since points on the upper right side of the phase space are mapped to points in the lower right side. Note that if one considers the 01-map to be a dynamical system and iterates it like any other, (i.e. without watching the state variables after applying only the 0-map) then a fake periodic orbit can be a fixed point, as it is in this case.

5 DFA Extraction

Our theory provides a theoretical foundation for studying DFA extraction as defined in Giles *et al.* (1992) [8], and in particular, allows us to study two of their conjectures. The first of their conjectures is that during training the network begins to partition its state space into fairly well separated, distinct regions or clusters which represent corresponding states in some finite state automaton. Another of their conjectures is that for what they call "well-trained" RNNs, the extracted minimal DFA are independent of the choice of partition parameter. Note that one must have a theory such as ours, that allows for an understanding of all of the infinitely many possible RNN solutions to a given language recognition task in order to productively study these conjectures.

Giles *et al.* (1992) [8] define DFA extraction as a four part procedure: (1) cluster DFA states, (2) construct a transition diagram by connecting these states with the alphabet labeled arcs, (3) combine transitions to form the full digraph loops, and (4) reduce the digraph to a minimal representation. They partition the state space by dividing each unit's range into q partitions of equal width.

The first important step in discussing DFA extraction is to note that there are examples of RNNs that robustly model DFA and values of q for which the extraction procedure described in Giles *et al.* (1992) [8] will not produce a DFA, let alone one that reduces to the unique minimal DFA that the RNN models. The problem with their procedure is that one of the RNN's i-maps may send points in one partition state to more than one other partition state and, when this happens, the corresponding DFA digraph will not define a function of the partition states to themselves. DFA extraction will work once we modify the

to cause errors while checking network performance in this way. The reason they cause difficulty is that iterated substrings of length greater

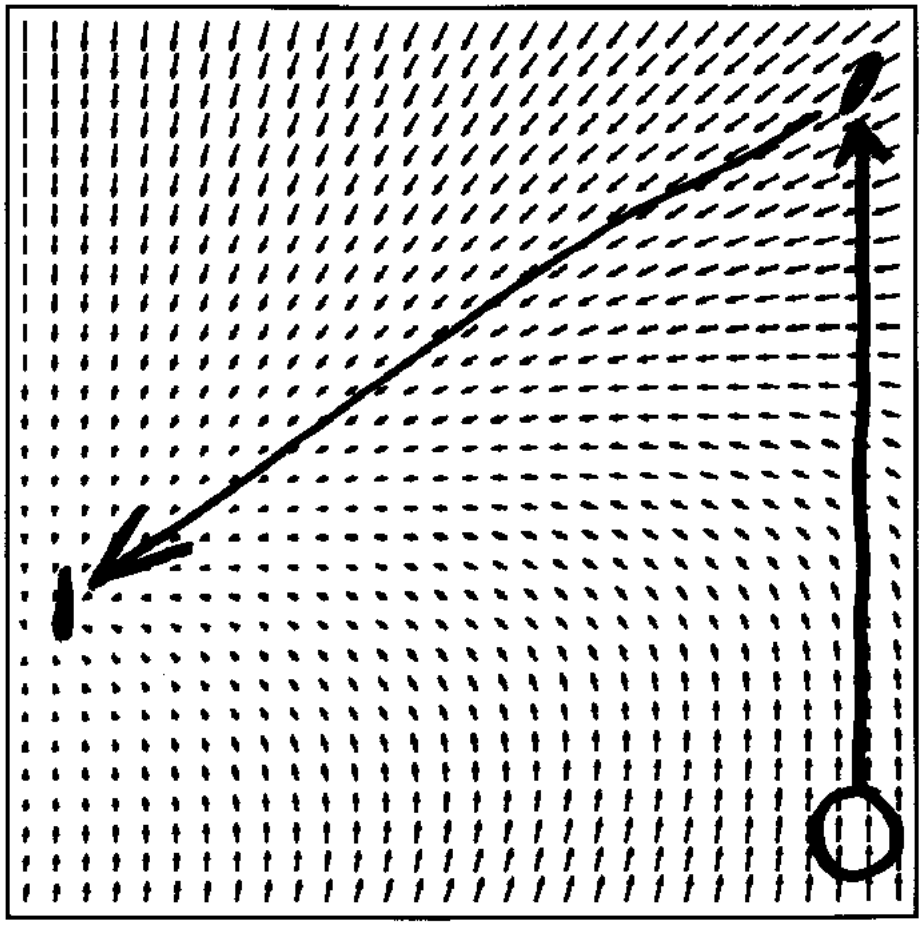

Figure 6: (C) Phase space dynamics of the 1-map.

than one are less likely to appear in both training and testing data sets. For a RNN recognizing L_3, we must check that the 01-map is hysteretic, since the minimal state DFA for L_3 has two fake periodic orbits.

Again, we study a solution which uses two-units. The e-map puts the network in the part of the phase space that will go to the reject region after two iterations of the 0-map or one iteration of the 1-map. The 0-map has only one fixed state, and its corresponding fixed point is in the reject region of phase space. To recognize the language then, it must use transient computation. Points near the bottom right side of the phase space iterate to points in the upper right side of the phase space (still in the accept region), which are then mapped to the re-

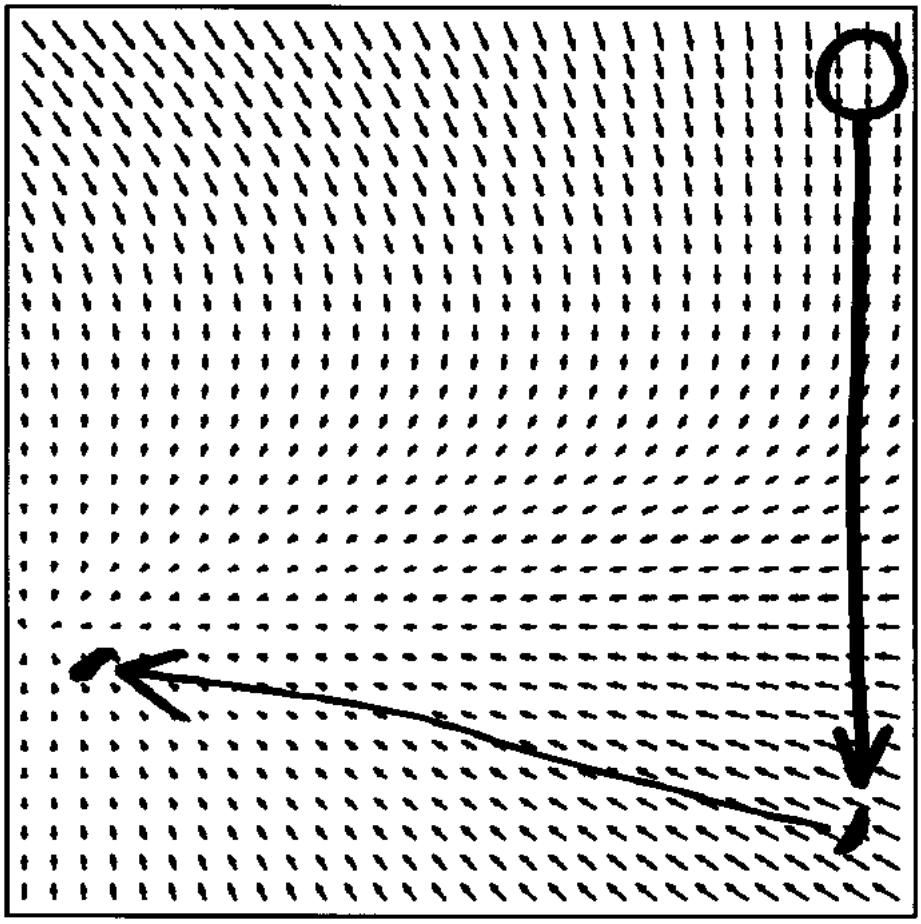

Figure 6: (B) Phase space dynamics of the 0-map.

Our theory allows for a complete understanding of the RNN's ability to generalize to strings of arbitrary length by determining the properties that the individual i-maps and compositions of these maps must have to ensure that the network will recognize the entire language rather than simply recognizing some finite subset of the language. Furthermore, our theory makes it clear that one needs to check the stability of fake periodic orbits, which is not obvious when looking at the grammar description.

As noted in Giles *et al.* (1992) [8] and Watrous and Kuhn (1992) [26], the lack of stability of fixed states is likely to cause errors in network performance when the RNN is checked on long, unseen strings. Lack of stability is an even more likely source of error for periodic orbits and fake periodic orbits since this lack of dynamic stability is less likely

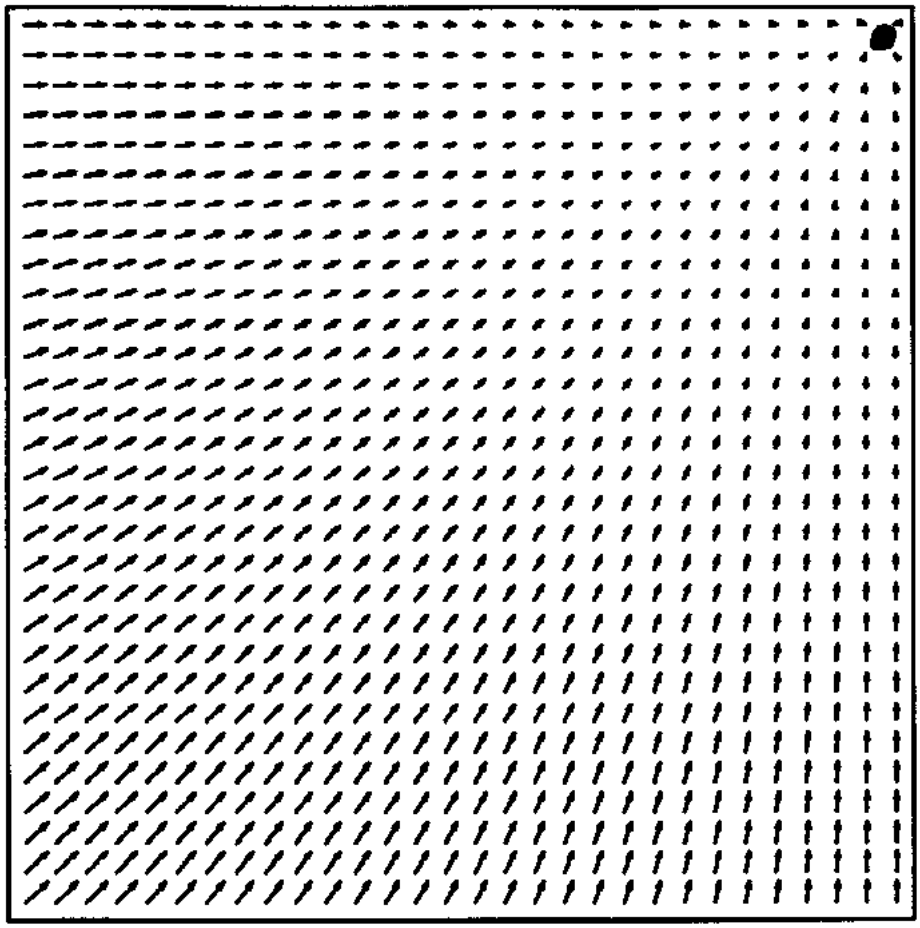

Figure 6: (A) Phase space dynamics of the e-map of a two-unit RNN that recognizes the language L_3.

near to itself (as it will when it is near a saddle node bifurcation), or if it has another type of near invariant set then it can have transients which don't move far from one another under many iterations of the map. If the training set does not contain very long strings, then these near invariant sets will allow for correct network performance on short strings. One way to help avoid this problem while training a network may be to add noise to the phase space variables during training since that would generally cause the near invariant sets to more quickly fail to effectively serve where true attractors are needed.

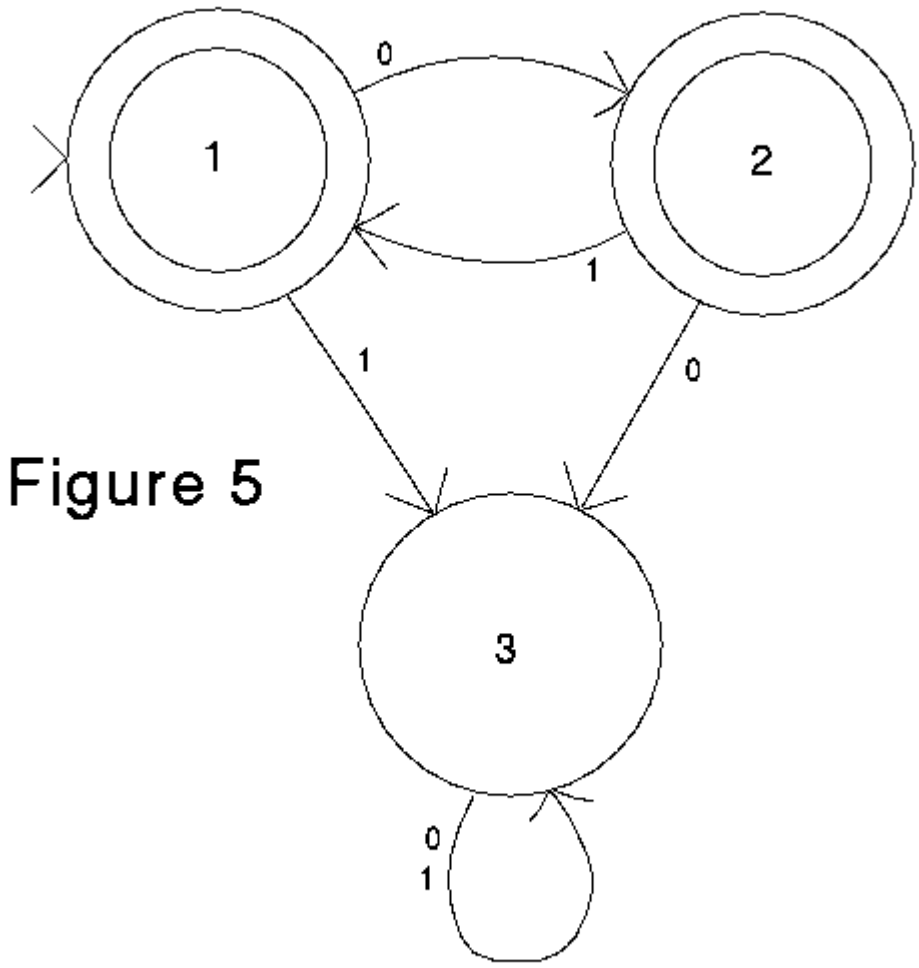

Figure 5

Figure 5: The minimal FSA which recognizes the language containing all words with no 00 or 11 substrings that begin with 0.

biological reasons.

It is possible to recognize L_2 with a one-dimensional RNN with activation functions more complicated than sigmoids. If, however, one attempted to use teacher-forcing during training, the problem would not be solvable in one dimension. Teacher-forcing would restrict the RNN to mapping between two points (one representing the fact that the FSA is in a nonfinal state, and the other representing a final state), and therefore it would be necessary for the 1-map to map the nonfinal state point to itself and also to the final state point, which is impossible. If we move to a two-dimensional network then this problem with teacher-forcing disappears.

4.3 Example 3

Our final example language, L_3, is the set of all strings that begin with 0 and have no 00 or 11 substrings. Figure 5 shows the minimal DFA for L_3, where states q_1 and q_2 are transient states and q_3 is a fixed state of both the 1- and 0-map. We also see that $\{q_1, q_2\}$ and $\{q_3, q_3\}$ are fake periodic orbits for the 01-map. As noted in Giles *et al.* (1992) [8] and Watrous and Kuhn (1992) [26], in order to check a RNN's ability to generalize to unseen strings, it is important to check the dynamic stability of the RNN. Our theory allows for a much more precise and general determination of this stability. While Giles *et al.* (1992) [8] equate generalization to long unseen strings with fixed point behavior, our theory shows that more general attractors may be used for the same purpose. More importantly, since Giles *et al.* (1992) [8] and Watrous and Kuhn (1992) [26] check for fixed point behavior by simply reading a long previously unseen string (Watrous and Kuhn (1992) [26] use strings of length 100), they falsely assume that if the RNN produces the correct output for a string of length N for N sufficiently large, it will necessarily produce the correct output for strings of arbitrary length.

It is possible and seems to be fairly common for a RNN that has been trained to recognize a language L which requires hysteretic computation to use transients of its i-maps rather than true attractors to correctly classify strings in L. With sufficiently large amounts of noise there will be some chance that the network will fail on any symbol after some large number. It seems very unlikely that anything both specific and useful can be said about this relationship since it results from the interplay between the noise and the way that the i-maps approach the identity function, but the idea is that if a function maps a point very

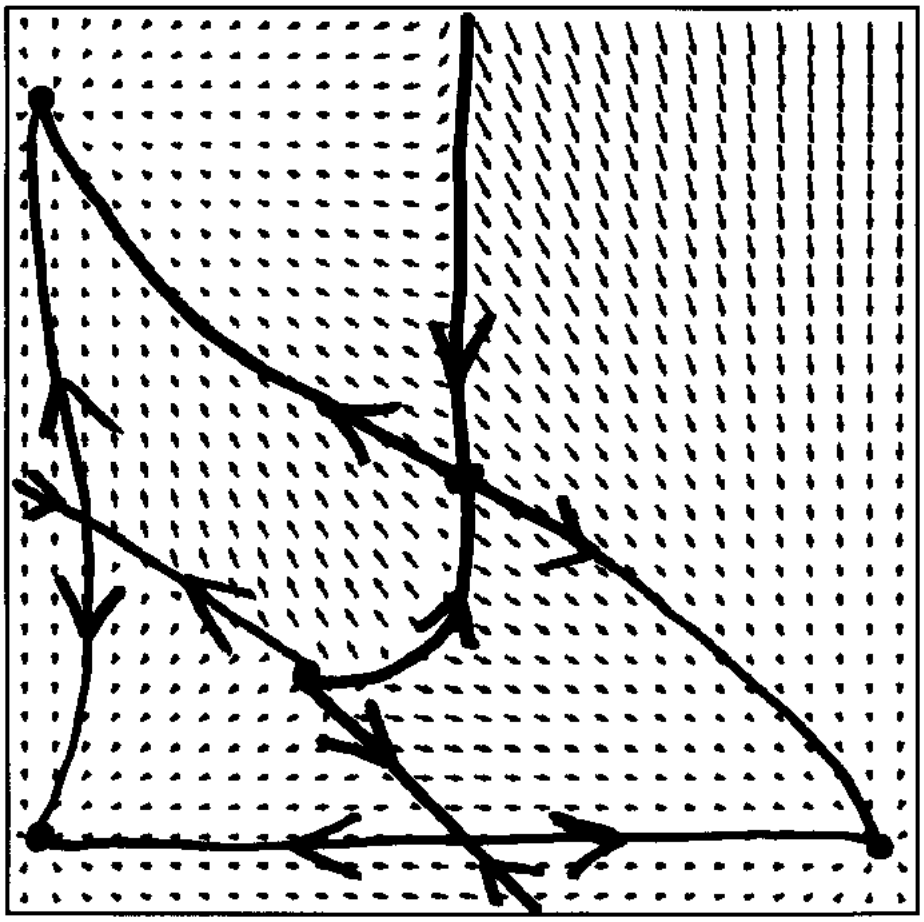

Figure 4: (C) Phase space dynamics of the 1-map.

RNN found could have had complexities that the simple prediction did not consider. Besides the possibility of having many behaviors in the same class, it is also possible that rather than using fixed points, the RNN could have had "small" chaotic attractors (small in the sense that they stay in appropriate future basins of attraction), that might lead one to believe that those dynamics are somehow necessary for the computation. So not only might there be many internal states corresponding to the same computational state, but any of those individual behaviors might also be complicated. What might be true when we observe more complicated dynamics is that, while they were not necessary from a computational perspective, they were necessary for topological reasons from the perspective of the map dynamics, or when observing nervous system behavior that they were necessary for

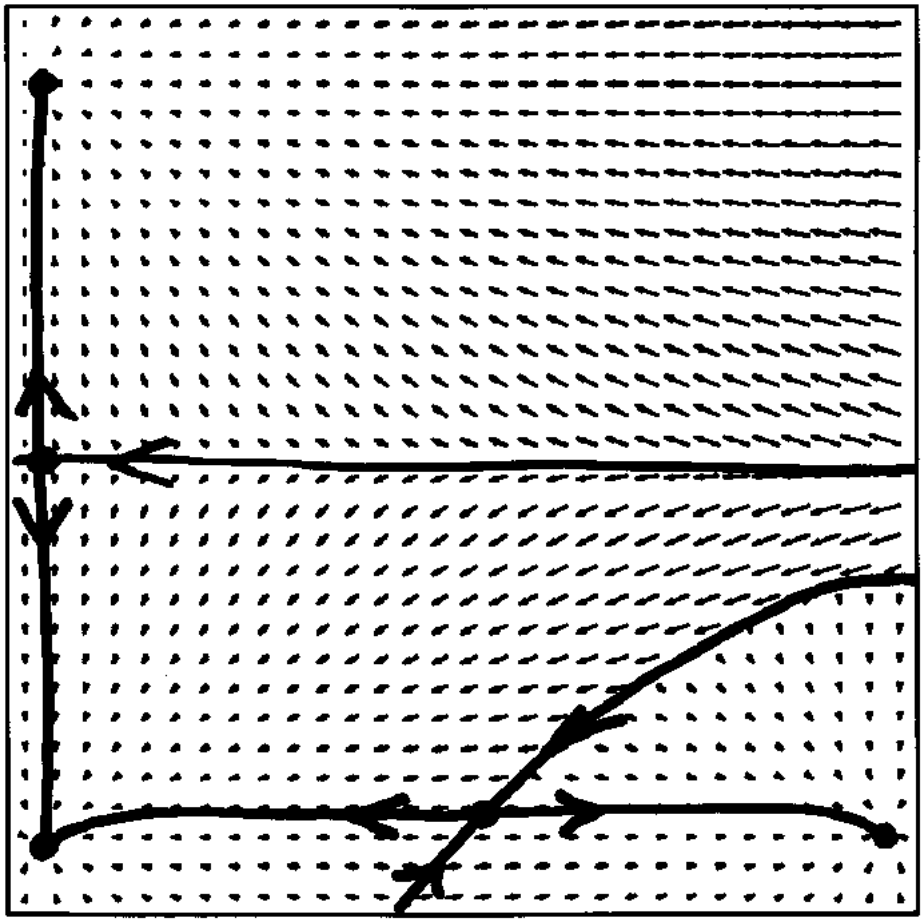

Figure 4: (B) Phase space dynamics of the 0-map.

unlikely that such a solution will be found while evolving the RNN. It is also clear from this example that Theorem 3.2 does not hold for RNNs which simply model rather than robustly model a DFA.

One can use the intermediate value theorem to prove that a one-unit RNN with a sigmoidal discriminant function (as a one-dimensional monotone dynamical system) cannot have these dynamics, so we know that the two-unit RNN solution shown in the figures is minimal with respect to network size. We emphasize here that the solution that the

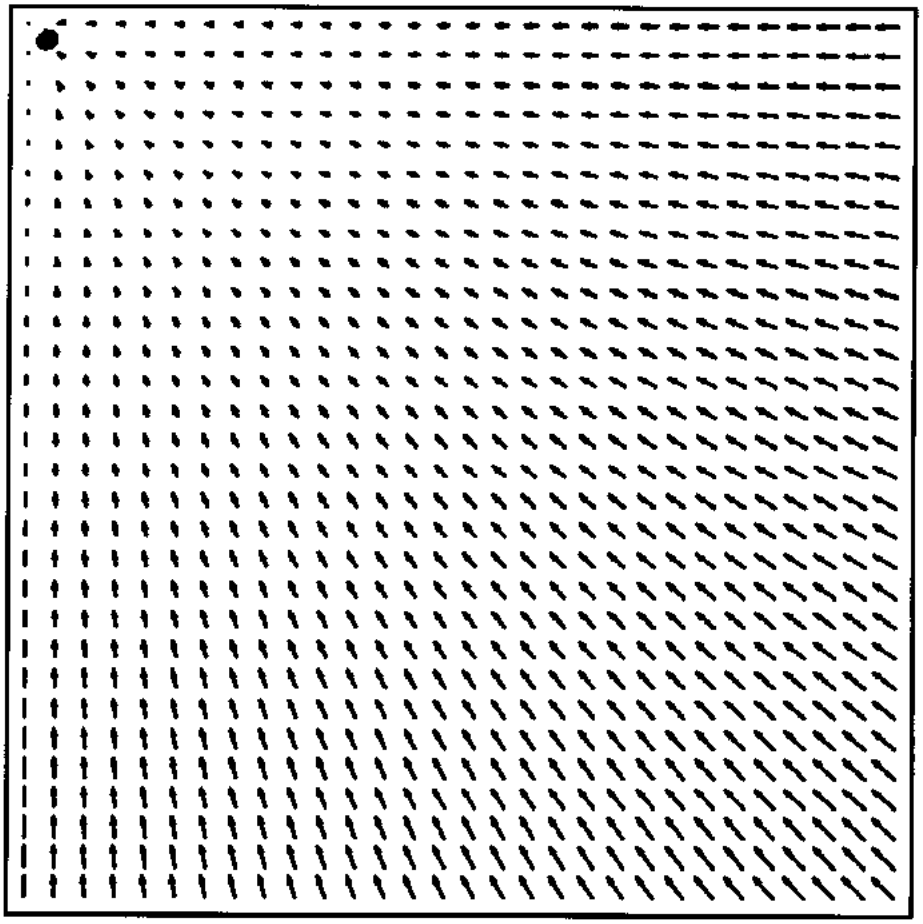

Figure 4: (A) Phase space dynamics of the e-map of the network that recognizes the language accepted by the automata in Figure 3.

the 0-map the periodic orbit of the 1-map lies exactly above them. This RNN clearly models the DFA, but it does not robustly model the DFA since if there is any perturbation to the orbit, then even a string of several zeros with moderate perturbations at each step is likely to fail to produce the correct output. Furthermore, if we perturb the maps at all and move the fixed points or the periodic orbit, then the RNN will fail to model the DFA. From this is follows that it is exceedingly

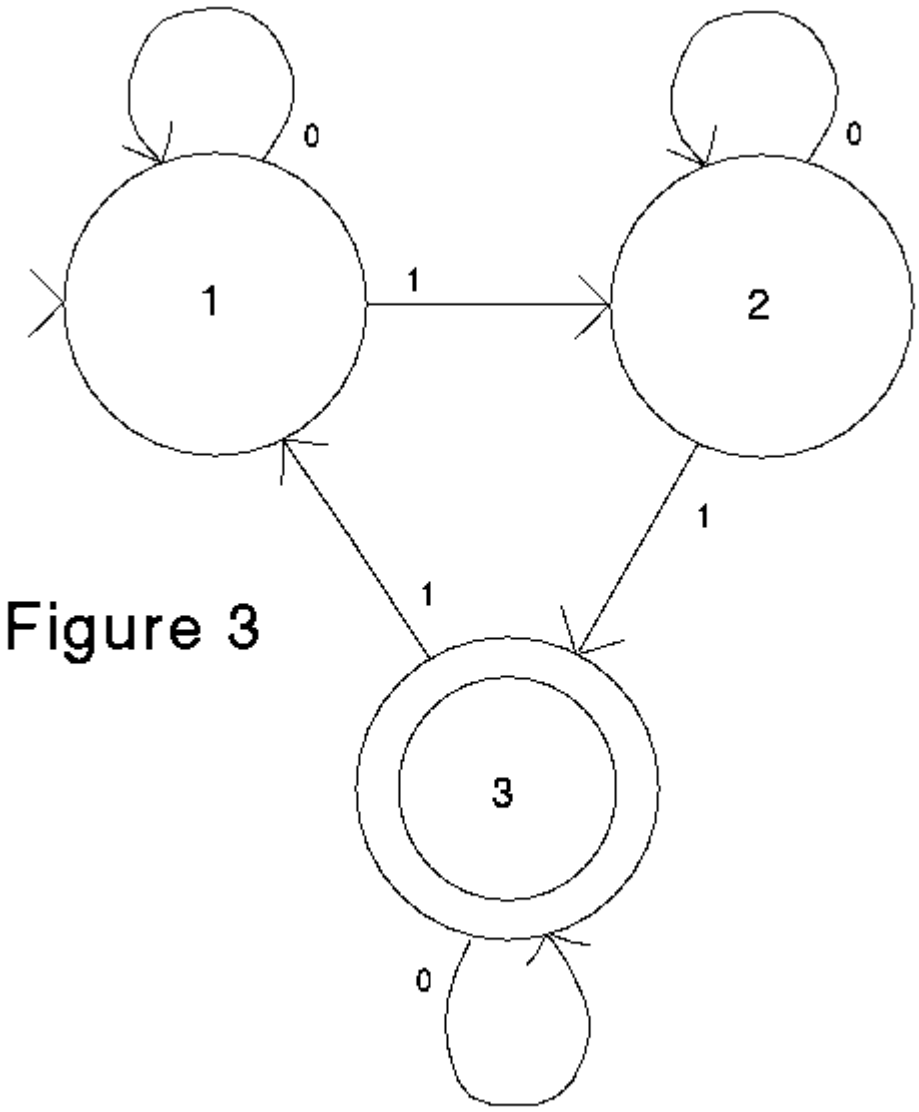

Figure 3

Figure 3: The minimal FSA which recognizes the language containing all words in {0,1}* which contain $3k + 2$ 1's for some k $\geq$ 0.

$k \geq 0$. Figure 3 shows the minimal DFA for this language.

The 0-map for this RNN is hysteretic. The DFA's states are a period three state orbit of the 1-map. Hence, to model this DFA, a RNN might have a 0-map that has three attracting fixed points, and a 1-map with an attracting period three orbit, where two points of this orbit are in the reject region, and the other is in the accept region, and the attracting fixed points of the 0-map coincide, or nearly coincide, with this period three orbit. This prediction does, in fact, turn out to be correct for the ten training trials that we have run. One of these solutions is seen in Figure 4.

The e-map puts the network in the part of the phase space which will go to the accept region after two iterations of the 1-map. The 1-map has three fixed states, corresponding to the three fixed points in the picture, and two saddle points. It will be seen that the fixed points nearly "sit above" the periodic orbit of the 1-map. The 1-maphas a period three orbit. It is more instructive to look at the dynamics of the third iterate of the 1-map($(1\text{-map})^3$), rather than looking directly at the 1-map and trying to see that the arrows will all match up to give this orbit. The attracting periodic orbit shows up as fixed points for $(1\text{-map})^3$, and the unstable period three orbit contains the saddle points. The existence of a periodic orbit implies (by Brower's plane translation theorem) the existence of a fixed point. In this case, the fixed point is the point at the intersection of the three basin boundaries (stable manifolds). If two of the fixed points of the 0-map were in one basin of $(1\text{-map})^3$, then the network would not be able to model the FSA since a word containing many zeros between each one would put the 1-map's period three orbit "out of phase" with the FSA.

To see the utility of using ϵ-pseudo-orbits rather than true orbits for our theory we will now consider a solution to modeling this DFA which uses unstable invariant sets and discuss its properties. Suppose that e-mapis a constant map, mapping all points to the fixed point of the 0-mapthat is in the upper left-hand corner of the phase space. Further suppose that the fixed points of the 0-map are unstable (repelling rather than attracting) and that the periodic orbit of the 1-map is unstable and maps the three fixed points of the 0-map to one another. So we have that rather than approximately sitting above the fixed points of

as it should be since the basin of attraction of the fixed point of the 0-map which is in the accept region would contain the fixed point of the 1-map which is in the accept region.

The 1-map has two attracting fixed points (one in the accept, and one in the reject region - corresponding to q_2 and q_3 respectively), and one saddle-type fixed point. Again, the curve which separates the basins of attraction for the two fixed points is the stable manifold of the saddle point. Note that the "accept" fixed point for the 1-map is in the basin of attraction of the "reject" fixed point of the 0-map, as it must be, since, for example, 111110 is not in the language.

Thus we see that observing phase space dynamics can give a clear understanding of how the RNN operates. Observing a time series of the activations or clusters of activations gives a incomplete understanding of the performance of the network since, for example, it says nothing of the robustness of the solution or whether or not it will generalize to strings of arbitrary length. Furthermore, traces from phase space say little about the "difficulty" of evolving the ability to perform the task since some DFA with n states require more complicated dynamical systems to model them than do other n-state DFA.

Another common approach to understanding RNN solutions is to study connectivity strength patterns. This approach is analogous to a common neurobiological approach used to understand networks of neurons, and has many problems. Some of those problems stem from the fact that there exist many nontrivial symmetries that can act upon the system while preserving the dynamics, some of which are discussed in Casey (1995) [3]. Other problems derive from the fact that this approach is analogous to attempting to look at the parameters of a high dimensional dynamical system and infer from them the dynamics. If the system were linear, this would be equivalent to looking at a matrix and directly, i.e. without explicit calculation, determine its eigenvalues (at least up to multiplication by a positive real number). Even for 2×2 matrices this is an unrealistic hope in general. For nonlinear systems it is even more rare that one can gain insight into the dynamics from observing the parameters, even after considerable effort to determine dependencies between parameters and dynamics, as in Casey (1995) [3] or Wang (1991) [25].

4.2 Example 2

A problem that essentially uses only hysteretic computation is recognizing the language, L_2, containing the words with $3k+2$ 1's for some

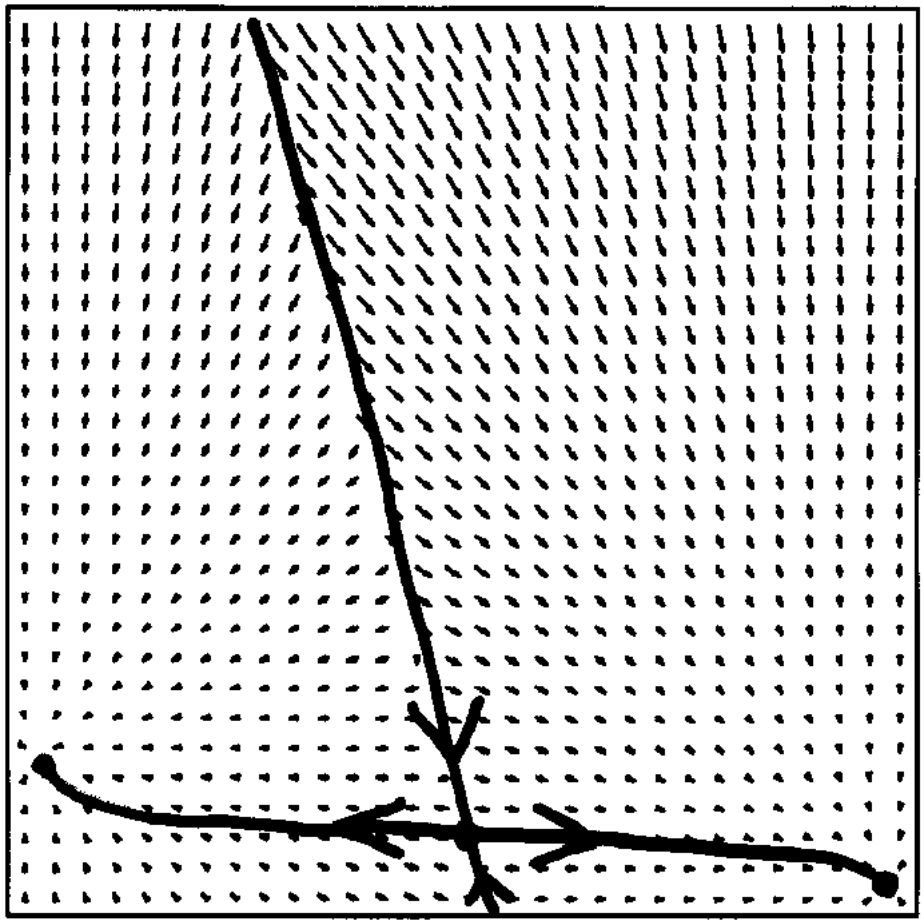

Figure 2: (C) Phase space dynamics of the 1-map.

symbol sets the network to accept, as it should.

The 0-map has two attracting fixed points (one in the accept, and one in the reject region, corresponding to q_1 and q_3 respectively), and one saddle-type fixed point. The curve that separates the basins of attraction for the two fixed points is the stable manifold of the saddle point, and is an example of an unstable invariant set (so one never "sees" it under iteration of the 0-map). The unstable manifold nevertheless can be found, and plays an important role in computation, since changing its shape and position can make the network fail to solve the problem because it forms the boundaries of the basins of attraction for the attracting fixed points. If we bent the right part of the curve down until it intersected the middle of the bottom side of the phase space, for example, then 0000001111111100000 would not be rejected,

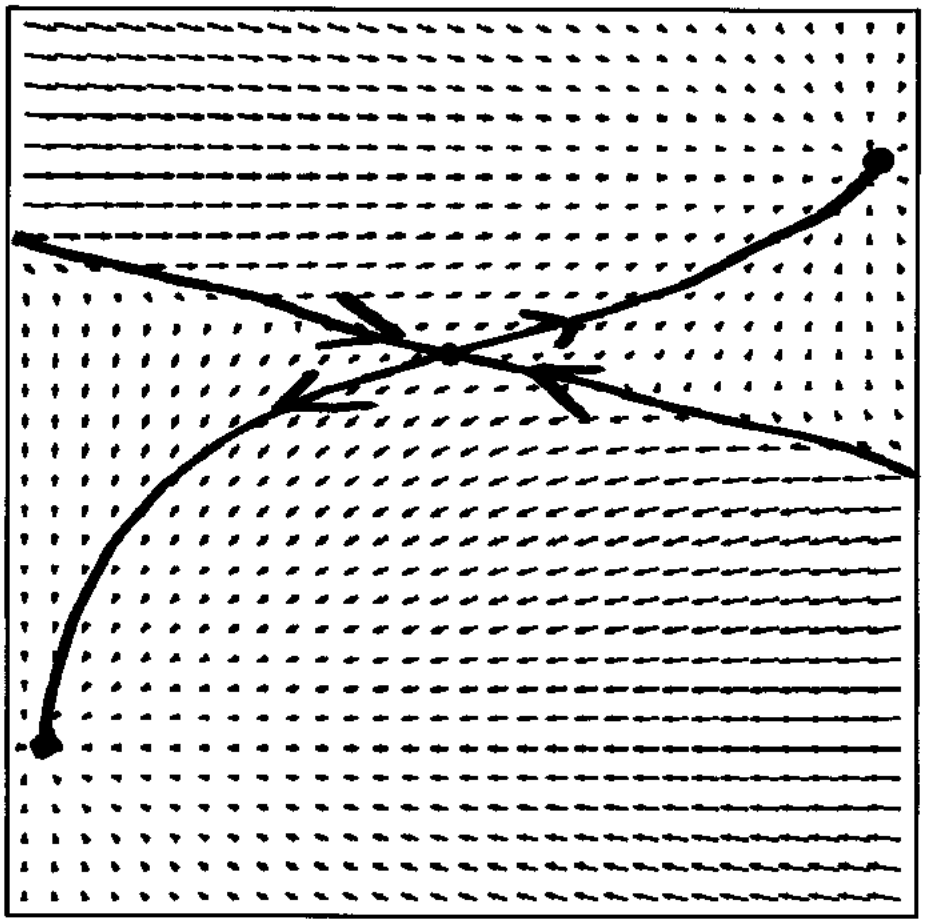

Figure 2: (B) Phase space dynamics of the 0-mapof the network that recognizes the 0*1* language.

The arrows in Figure 2 show the direction that the point at the base of the arrow moves under iteration of the map. The distance that the point is moved is not shown. In the case of the e-map, the point moves very close to the fixed point of the map (the point at which all arrows point). Note that the right side of the phase space (as you are looking at it) in the figure is the accept region (i.e. the region that projects to the interval $[1-\epsilon, 1]$) and the left side is the reject region, so the e

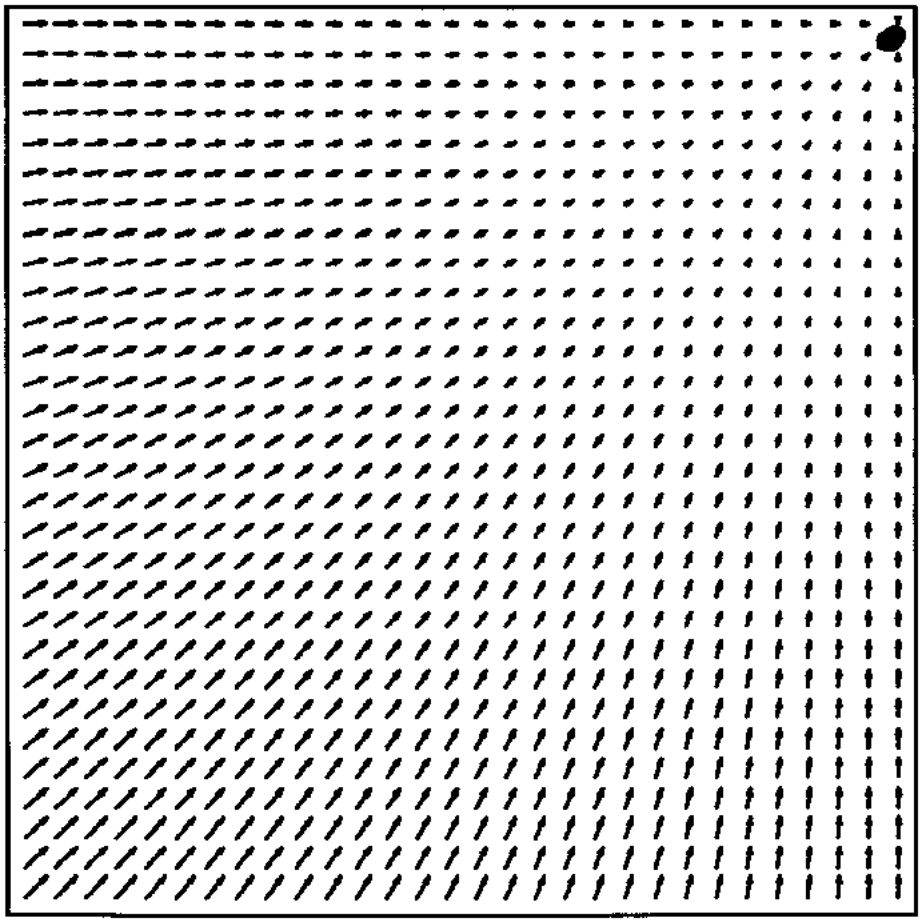

Figure 2: (A) a representation of the phase space dynamics of the e-map of a two-unit network that recognizes the 0*1* language.

4.1 Example 1

A problem that uses hysteretic computation is recognizing the 0*1* language[4]. The minimal DFA that recognizes the 0*1* language is shown in Figure 1. We see q_1 and q_3 are fixed states of the 0-map and q_2 and q_3 are fixed states of the 1-map.

[4] 1* could also be examined, but its simplicity makes it less instructive.

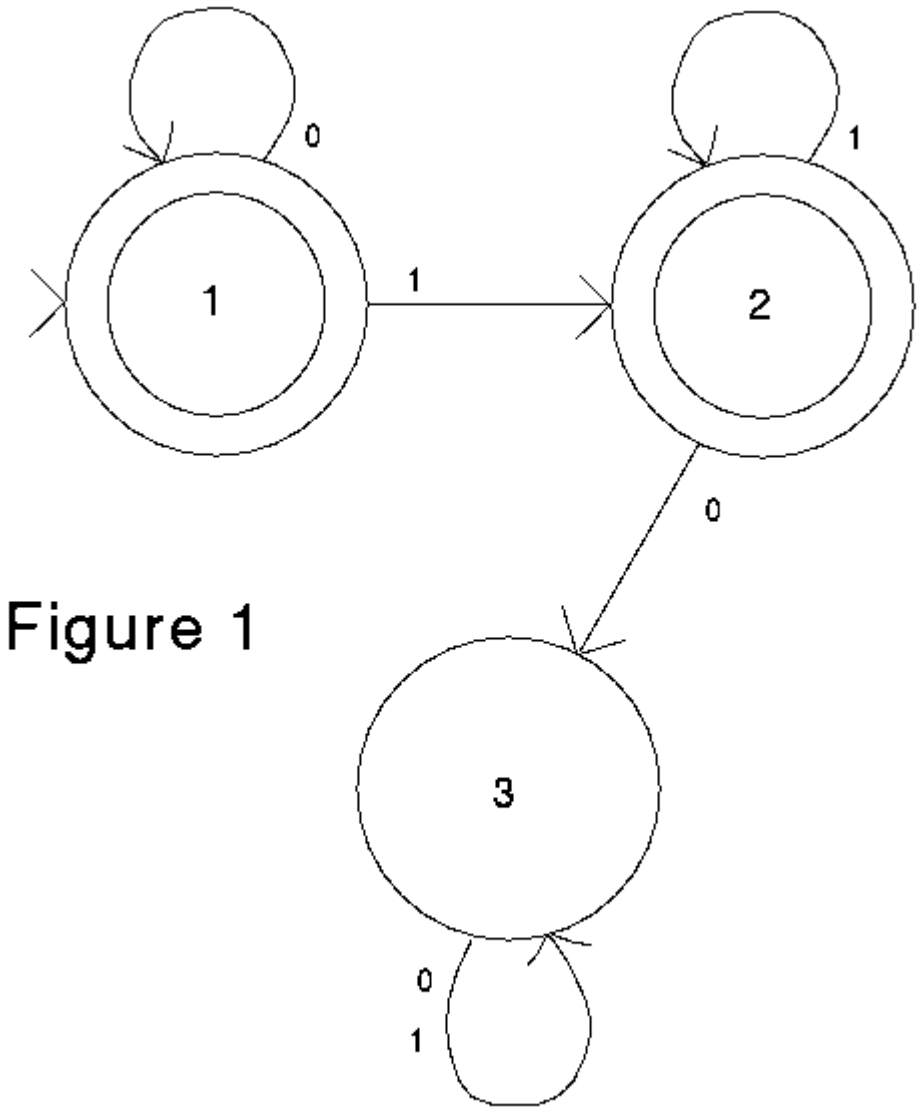

Figure 1

Figure 1: The diagram for the minimal DFA that recognizes 0*1*.

tation, but can lead to difficulties when used to learn transient computation. One potential problem with teacher-forcing is that it can restrict activation dynamics to a subset of the phase space, making the desired computation impossible. This possibility will be discussed with the examples.

The theorems in this sections show that RNNs that model DFA have many constraints on their dynamics. These constraints outline an algorithm for studying RNNs modeling known DFA. First, one examines the minimal state diagram of the DFA to find all cycles. By theorem 3.2, these cycles correspond to behaviors required of the trained RNN that correctly recognizes the language. One can then estimate the size of the RNN needed to model the DFA, and determine whether or not to use teacher-forcing during training by considering the number of behaviors the RNN must have. One should make sure that there are a sufficient number of training strings that contain iterated substrings corresponding to the cycles of the minimal DFA diagram. After training, one examines the stability of invariant sets of the I-map, including their ϵ-pseudobasins of attraction[3]. Finally, one checks that transient states (which are finite in number) map between appropriate ϵ-pseudobasins of attraction of the behaviors. If the trained RNN performs correctly during these tests, then it will generalize to all unseen strings.

4 Examples

For the following problems, similar to the format in Giles *et al.* (1992) [8], the RNN receives a unary encoding, with one input line for each character in $\{e, 0, 1\}$, where e marks the beginning of a string and corresponds to the empty string (rather than denoting the end of a string as it does in Giles *et al.* (1992) [8]). The task is to recognize a regular language by robustly modeling a given DFA. The accept and reject regions are the regions of the phase space that project within δ of 1.0 or 0.0 on the target unit axis, for some $0 < \delta < 0.5$.

[3]This involves finding the "shapes" of unstable invariant sets, which is very difficult for high dimensional mappings.

DFA will have changed but the state of the network will remain the same, contradicting theorem 3.1. This idea will be further investigated in the section on DFA extraction.

The distinction between transient and periodic states is important because periodic states are relatively well understood, and for RNNs with fixed discriminant functions, the dimension of the network needed to have a given combination of fixed and periodic states should be an answerable if difficult question. Transients, however, are not well-understood, and the size of the RNN required to have certain kinds of transients is sensitive to the shape of the discriminant function. Empirically, large numbers of hidden units are not necessary for some DFA with large numbers of transient states, while the number of hidden units roughly scales with the number of fixed states and distinct periodic orbit states. For example, I conjecture that an autonomous RNN with a sigmoidal discriminant function and n units can have at most 2^n attracting fixed points. It is easy to create an n unit RNN having 2^n attracting fixed points, but it is tricky to show that this is also an upper bound.

Here we note that the size of the RNN needed to solve a given problem can depend critically on the discriminant function chosen. If we allow the discriminant functions to be arbitrary by letting each input vector correspond to an arbitrary dynamical system, then it is always possible to model a DFA using only one-dimensional maps. It is easy, given a minimal DFA diagram, to construct by hand a family of dynamical systems that will robustly model that DFA. This fact suggests a more judicious choice of a family of functions to use to model DFA while inferring a DFA from examples; specifically, a family of functions whose complexity depends on degrees of freedom in choosing the one-dimensional maps, rather than in the dimension of the maps. The use of one-dimensional maps would greatly simplify the analysis we describe for determining the RNN's performance. The use of one-dimensional maps may also reduce the scaling of the complexity of the RNN with the complexity of the DFA since increasing the dimension of the RNN increases the number of parameters cubically for second-order RNNs, but the number of parameters should increase only linearly with the degrees of freedom for a well-chosen family of one-dimensional maps. This question requires further investigation.

Another question that may be investigated using our work is whether or not teacher-forcing should be used to learn a known DFA. Teacher-forcing seems to work well for learning to perform hysteretic compu-

of the corresponding I-maps will be contraction mappings, and hence have invariant sets consisting of unique attracting fixed points. By our generalization of Theorem 3.2, if a RNN models a hysteretic DFA, then at least one I-map will undergo a bifurcation during training. Hence, RNNs that model DFA that are hysteretic with respect to many distinct strings will have high geometric complexity. It can be shown that, at least in one-dimensional dynamical systems, one needs many degrees of freedom in order to produce families of dynamical systems having many behaviors, and, therefore, requires many parameters in the generating functions. It is reasonable to assume that the same is true in higher dimensions, but very little is known about the relationship between the number of behaviors of a dynamical system and other properties of the mappings in dimensions greater than one.

At this point, it is useful to note that a fake periodic orbit can exist when its individual i-maps are contraction mappings, so the existence of a single fake periodic orbit for an I-map should not make it more difficult to learn to model a DFA, which is another reason to call them fake orbits.

Theorem 3.3 *If a minimal DFA has a state that is a transient state of the i-map then the i-map must move a point from the ϵ-pseudobasin of attraction of one behavior to the ϵ-pseudobasin of attraction of a behavior not in the same class as the first, for some j-map, a composition of j-maps or the projection map, where the basins of attraction of the projection map are the accept and reject regions.*

This is the weakest useful statement we can make about transient states. It is useful because it states the important feature that the i-map must act productively on transients. We prove this by contradiction: if the i-maps never move points from one ϵ-pseudobasin of attraction to another, then they will never produce a discernible change in the output of the RNN, contradicting the minimality of the DFA. It is the weakest statement because it does not characterize the action of the I-maps on points in phase space corresponding to an arbitrary transient state, which we can characterize in terms of intersections of iterated pre-images of ϵ-pseudobasins of attraction under the i-maps.

Hence, a RNN uses transient computation if it has an i-map with at least one transient state. The proof is to notice that if we are in a transient state for an i-map that is a recurrent point from the perspective of i-map dynamics, and if we then read a string of i's to iterate the i-map and get back to the initial state, then the state of the

There are also the obvious generalizations to the period n case, $n \geq 2$, where we consider $(i\text{-map})^n$ (the n-fold composition of the i-map with itself) instead of the i-map alone and the fake periodic case, where we consider general I-maps.

It follows from theorem 3.2 that if an i-map has a period n state orbit and a period m state orbit, disjoint from one another (n may or may not equal m) then the RNN is hysteretic. (Note that the existence of periodic states only depends on the DFA, but the conclusion that the RNN is hysteretic is a statement about the RNN that models the DFA.) Hysteresis implies the existence of unstable invariant sets. Hence, if hysteresis is required for a computation, then one can deduce that the weights of the RNN must be outside of some neighborhood of the origin (since in a neighborhood of the origin, all of the corresponding i-maps are contraction mappings, and hence, by the contraction mapping theorem, have unique behaviors). Hence, if we use the minimum distance from the origin in weight space as a measure of difficulty of modeling a DFA or a measure of geometric complexity of the RNN, then computations requiring hysteresis are more difficult to learn than those which do not require it.

The benefit of using the minimum distance from the origin in weight space to measure of geometric complexity when talking about RNNs is that it gives us an idea of how far the weights need to travel in order to achieve a solution. It is possible then that it will predict that the number of learning trials that are required to learn to model a DFA with high geometric complexity will, on the average, be higher than the number needed for a DFA with low geometric complexity. There are, however, many problems with this measure. First, a means of calculating the minimum distance is not presently known, and it is unlikely to be an easy calculation even if our knowledge of RNN dynamics were much greater than it is. It is much more reasonable to hope to calculate lower bounds on the geometric complexity (as above) than to hope to calculate the actual minimal distance. These are questions which require much further investigation, but preliminary experiments suggest that even this limited understanding of this measure suggest that it may help understand the speed at which a RNN can learn to model a given DFA relative to similar DFA, and to understand why some RNNs will converge during training and others will not.

A related measure of geometric complexity of the RNN that may be useful is the minimal number of behaviors for the I-maps. If we assume that we initialize weights to sufficiently small random values, then all

state of the i-map. For this reason, the following definition will be useful:

Definition 3.1 *When there is more than one distinct behavior corresponding to a single fixed state, we say that these behaviors are in the same class.*

The following theorem goes deeper into the representation issue by giving a coordinate-independent property of the i-maps of the RNN modeling a DFA. This property can be used as a measure of complexity of the DFA in terms of the complexity of the RNN that models it.

Theorem 3.2 *Suppose that a RNN robustly models a given DFA, then an i-map must have at least one distinct behavior for each of its fixed states.*

Furthermore, each behavior must be contained in either the accept or reject region of phase space according to whether or not its corresponding fixed state is also a final state of the automaton. Note that this theorem is not true if the RNN does not *robustly* model the DFA. We will give a counterexample to that conjecture with the examples.

Proof: The proof of this theorem relies on the observation that we may test the RNN on a string of an arbitrary number of i's, and reading such a string corresponds to iterating the i-map an arbitrary number of times. Suppose that the i-map has more than one fixed state (which is a statement about the DFA), but the i-map has only one behavior. By theorem 3.1 we know that the fixed states have corresponding disjoint, closed sets Q_i and Q_j. It is shown in the appendix, in lemma 7.1 that behaviors are open sets, so the complement of the behavior is closed. By the wandering lemma we know that some ϵ-pseudo-orbit of the RNN will eventually attain the behavior under iteration of the i-map. If the behavior contains points in both Q_i and Q_j, then there is an ϵ-pseudo-orbit of the i-map that will change the state of the RNN to a point corresponding to whichever of the fixed states the DFA is not in, contradicting theorem 3.1. Hence, the behavior is contained in either Q_i or Q_j. Without loss of generality, we may assume it is contained in Q_i. Hence, if the DFA is in state q_j and we read a string of N i's for N sufficiently large, then the RNN will be in Q_i and therefore the RNN fails to model the DFA. Simple induction on the number of fixed states completes the proof. ■

learning algorithms, and, we shall see, is essential for proving our second theorem.

Proof: The Q_i have nonempty interiors by the fact that the RNN is assumed to robustly model the DFA. To see that the Q_i must be mutually disjoint, assume that there is some point, x, in the intersection of two of these sets, Q_i and Q_j for i not equal to j, but that the RNN still robustly models the DFA. Since the DFA is assumed to be minimal, there exists a string, which we call F, such that after reading F, the DFA will be in a final state if the DFA was in q_i, and in a nonfinal state if the DFA was in q_j. Now assume that the RNN is in state x. After reading F, the RNN will be in either the accept or reject region or neither. Therefore, the RNN fails to model the DFA. Since the accept and reject regions are defined to be closed and the i-maps are continuous, we can make this same argument for an accumulation point of any point in one of the Q_i, and, therefore, the Q_i are closed. ■

The fact that the Q_i will be closed follows from the definition of the accept and reject regions as closed sets and the continuity of the i-maps. If we were to define the accept and reject regions as open sets, then the Q_i would be open. If we did not require that the i-maps were continuous, then we would not be able to conclude that the Q_i must be closed. We have chosen to define the accept and reject regions as closed sets so that, when combined with our assumption that the phase space is bounded, we may conclude that the Q_i are compact. Compactness of the Q_i will be used in the applications to DFA extraction, but is not essential since we can work with the compactifications of the Q_i (which will still be disjoint by the robustness assumption) and then see that the conclusion holds for the interiors of the compactified Q_i.

Robustness is not necessary for the proof of 3.1 as stated, except in showing that the Q_i have nonempty interiors. Robustness is, however, necessary for our next theorem and desirable for other reasons given below.

Lemma 3.1 (The Wandering Lemma) *Suppose some closed set B with nonempty interior contains no behaviors. Then for any $\epsilon > 0$ and for any point $x \in B$ there is some finite number N such that the N^{th} point in some ϵ-pseudo-orbit of x will leave B.*

The proof of this lemma is in the appendix.

There may be redundancy in a solution in the sense that there may be several behaviors in the network which correspond to a single fixed

Definition 2.18 *An I-map is said to be hysteretic if it has more than one ϵ-pseudo-attractor, and a RNN is hysteretic if for some string of inputs I, its I-map is hysteretic.*

Hence a hysteretic map is a dynamical system that embodies several different robust, nonlinear oscillations that are selected based upon initial conditions.

Definition 2.19 *A transient of an I-map is a point not in $\mathcal{R}_\epsilon(I - map)$.*

Using the above definitions, we divide computation dynamics into categories: hysteretic and transient computation.

Definition 2.20 *Hysteretic computation is computation that can occur using periodic orbits (of periods $\geq$ 1, where a periodic orbit of period one is a fixed point) of the I-map of a RNN.*

Definition 2.21 *Transient computation is computation performed using transients of the I-map.*

3 Results

Our first theorem shows why clustering and DFA extraction succeed and provides a theoretical foundation for representation issues related to clustering and extraction.

Theorem 3.1 *The state space of a RNN robustly modeling a given connected, minimal DFA must have mutually disjoint, closed sets Q_i with nonempty interiors, corresponding to the states q_i of the DFA.*

The correspondence, of course is that the RNN initialized with any point in Q_i will produce the same output as the DFA initialized to state q_i for all possible input. An anonymous reviewer has pointed out to the author that these properties (excepting nonempty interior) would follow from a more algebraic approach where points in the phase space are defined to be equivalent if they produce the same output (in $\{0,1\}$) for all possible input strings. This type of approach is an old one and is described in Arbib and Zeiger (1969) [1], Kalman, Farb and Arbib (1969) [11], and Nerode (1959) [18]. As discussed earlier, however, robustness is a very important property since it allows one to focus on understanding RNN solutions which can be found using

See Wiggins (1990) [27] or Devaney (1987) [6] for definitions and background from dynamical systems theory. Since we are concerned with systems operating in the presence of noise, we will need to define an attractor for such systems, and to do that we need a notion of recurrence. For ϵ-pseudo-orbits, the recurrent points are those that are ϵ-pseudoperiodic:

Definition 2.15 *A point, x, is said to be ϵ-pseudoperiodic for a map f if there exists an ϵ-pseudo-orbit, $\underline{x}$, such that $x = x_n$ for some $n > 0$. For the least such n, we say that a point is ϵ-pseudoperiodic of period n.*

One set that is often useful for talking about ϵ-pseudo-orbits is the chain recurrent set, defined to be the set of all points which are ϵ-pseudoperiodic for all $\epsilon > 0$. Since we will always have a finite amount of noise in our systems, the following will be more useful:

Definition 2.16 *For a fixed $\epsilon > 0$ and a mapping f the ϵ-chain recurrent set, $\mathcal{R}_\epsilon(f)$, is the set of all points which are ϵ-pseudoperiodic for f.*

For the general topology used in the following definitions and proofs of theorems, one can consult Munkres (1975) [16]. In particular, one should be familiar with open and closed sets, accumulations points and dense sets.

Intuitively, the attractors of a dynamical system are the sets of points to which most points evolve under iteration, but there is no one universally accepted definition of an attractor. Milnor (1985) [15] reviews some of the existing definitions and proposes another, but none of these are well-suited to noisy systems. The following definition seems to correspond to the intuitive notion of an attractor for systems with noise:

Definition 2.17 *A subset of $\mathcal{R}_\epsilon(f)$, $\mathcal{A}$, is an ϵ-pseudo-attractor or behavior if*

1) it is a subset of $\mathcal{R}_\epsilon(f)$ having a dense ϵ-pseudo-orbit which is maximal in the sense that there is no $\underline{x}$ which is dense in a set $\mathcal{B}$ of which $\mathcal{A}$ is a proper subset, and

2) $\mathcal{A}$ is a proper subset the ϵ-pseudobasin of attraction, which is the set of all points whose ϵ-pseudo-orbits are eventually contained in $\mathcal{A}$, and is denoted $\rho(\mathcal{A})$.

Intuitively then, an ϵ-pseudo-attractor corresponds to an oscillation or a steady state of the system which is stable in the sense that a slightly perturbed system will settle to one of these asymptotic behaviors.

perturbations of the organism from sources other than those which are directly related to the computation in which we are interested, and we would like to consider only systems which can still perform correctly in these circumstances. In other words, it is desirable, for example, to consider only systems which can still add even though the wind is blowing. A second reason to consider solutions with finite amount of noise is that if we are implementing the system on a digital computer then there will be noise in the system due to round-off errors. So we would like to restrict our attention to solutions which work in spite of round-off errors. A third reason to consider robust solutions is that if we are using a learning algorithm to find the solution, then we would like to consider solutions which will be well behaved with respect to small perturbations in the weights since these are the solutions which can be found (except possibly while using teacher-forcing, but a solution found in this way is likely to be destroyed when teacher-forcing is turned off). As will be seen in the example, in order to model a DFA with unstable invariant sets it is often necessary to "match-up" invariant sets for the various i-maps, and therefore if we perturb one of these maps and move the invariant set slightly then the RNN will no longer model the DFA.

To rule out these solutions which may require noiseless operation of the neural network or physical system, we will require that the computation be robust in the following sense:

Definition 2.12 *The system is operating in a noisy environment if, for some fixed $\epsilon > 0$, the state variables of the RNN, $\mathbf{x}(t)$, may be replaced by any point $\mathbf{y}(t)$ such that $|\mathbf{y}(\mathbf{t}) - \mathbf{x}(\mathbf{t})| < \epsilon$ at each time step, $\mathbf{t}$ (where $|\mathbf{y} - \mathbf{x}|$ denotes the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$).*

That is, we are concerned with ϵ-pseudo-orbits rather than the true orbits of our state variables under the action of the maps.

Definition 2.13 *Fix an $\epsilon > 0$, let f be a map of $\mathcal{I}^N$ into itself and let $x \in \mathcal{I}^N$. An ϵ-pseudo-orbit of x under f is a sequence of points $\underline{x} = \{x_0, x_1, \ldots\}$ such that $|f(x_{i-1}) - x_i| < \epsilon$ for all $i > 0$, where $x_0 = x$.*

Unless noted otherwise, whenever we speak of the iteration of a point we will assume that the system is operating in a noisy environment.

Definition 2.14 *If a RNN models a DFA, $\mathcal{M}$, and continues to produces the correct output when operating in a noisy environment for some fixed $\epsilon > 0$, then it is said to robustly models $\mathcal{M}$ or, if L is the language associated with $\mathcal{M}$, recognizes L.*

Definition 2.6 *If I is the composition of a symbol, i, with itself n-times, then we say that Q is a period n state orbit of the i-map.*

Definition 2.7 *Call a state which is not in a period n state orbit ($n \geq 1$) for an i-map a transient state of the i-map.*

Definition 2.8 *For all other I, i.e. those which contain at least two distinct symbols, we say that an n-cycle, Q, is a fake period n orbit of the I-map.*

Such orbits are called "fake," because they can be made of transient states of the intermediate i_j-maps.

Definition 2.9 *We say that a cycle is reduced if the q_i are distinct excepting q_1 and q_n).*

In order to speak of the RNN modeling a DFA, we need to have some way of connecting the state of the system to the state of the DFA. In particular, we need to have some way of determining whether the state of the system corresponds to a final or nonfinal state of the DFA. To this end, we need the following definitions.

Definition 2.10 *Let A and R be disjoint, closed subsets of $\mathcal{I}^N$. Call A the accept and R the reject regions of the RNN or simply the accept and reject regions.*

If necessary, we can easily generalize this definition to DFA with more than two types of output, thereby extending the theory to modeled systems that have many distinct outputs.

Definition 2.11 *A RNN is said to model a DFA if for any valid input string S the RNN is in the accept region if the DFA is in a final state after reading S, and in the reject region if the DFA is in a nonfinal state after reading S. To be valid, S must start with e and be followed by any string in $\{0,1\}^*$. If $S = e$, then the RNN must be in the accept or reject region according to whether the initial state of the DFA is a final or nonfinal state.*

Under this definition it is possible for a RNN to model a DFA using unstable invariant sets of the individual I-maps. An example will be given with the rest of the examples, but before continuing we will explain why it is desirable to consider only solutions with a finite amount of noise. The first thing is that for organisms behaving in and interacting with a complicated environment there will inevitably be

with a unary encoding, changing the input is equivalent to changing all of the weights in the network when the RNN is viewed as a dynamical system. Hence, a second-order network with M binary input lines can be viewed as $M+1$ independent dynamical systems[2]. We now generalize the definition of an i-map:

Definition 2.2 *For the string of symbols $I = i_1 i_2 \cdots i_n$, define the I-map to be the composition of the i_j-maps.*

We now turn to DFA. Here and following, when applicable, we shall use the notation of Lewis and Papadimitriou (1981) [14], including the definition of a DFA.

Definition 2.3 *A* **deterministic finite state automata (DFA)** *is a quintuple $M = (Q,\Sigma,\delta,s,F)$ where*
Q is a finite set of **states**,
Σ is an alphabet,
$s \in Q$ is the **initial state**,
$F \subseteq Q$ is the set of **final states**,
and δ, the **transition function**, *is a function from $Q \times \Sigma$ to Q.*

In this paper we assume that all DFA models are connected and minimal with respect to the number of states, for the language that they recognize.

DFA can be viewed as language recognition devices by saying that a string is accepted if after beginning in the initial state and then reading that string the DFA is in a final state, or rejecting

Definition 2.4 *For $n \geq 1$, define an n-cycle of the I-map to be an ordered set of n states $Q = \{q_1, \ldots, q_n\}$ such that when I is read, the DFA will pass through these states, beginning and ending with q_1.*

Note that in this definition we are combining the I-map of the RNN and the states of the DFA, which are not related *a priori*, but we will show that the two correspond in a well-defined manner.

Definition 2.5 *If I is a single symbol, i, then we call a 1-cycle, $\{q_1\}$, a fixed state of the i-map.*

[2]The dynamical systems are independent in the sense that their weights (parameters) are independent. For first-order networks, changing the input amounts to changing only the bias term, which means that we still have M distinct dynamical systems, but they are no longer independent.

DFA (Siegelmann and Sontag (1991) [24], Giles *et al.* (1992) [8]); in particular, RNNs of the type studied in Williams and Zipser (1989) [28] and Giles *et al.* (1992) [8] can be used to model DFA. Omlin and Giles (1994) [19] provide a constructive proof that an N+2-unit second-order RNN is capable of modeling any given N-state DFA. Rather than constructing specific solutions to the problem of modeling various DFA, we solve the converse problem by showing what all solutions must have in common and thereby give a sort of minimal complexity in terms of dynamic properties of the RNN. We have greatly benefited from the framework for solving problems with complicated temporal constraints that Giles *et al.* (1992) [8] have constructed. Hence, when it is useful to speak of a particular type of model, we will use their general framework, though our results are more general and apply to the first-order and the simple recurrent network (as in Elman (1989) [7]) cases, and easily extend to problems with any finite number of inputs or outputs.

More precisely, we use the following equations for the examples:

$$x_k(t+1) = F(\mathbf{x}(t), \mathbf{y}(t)) = S(\sum_{i=0}^{N} \sum_{j=0}^{M} W_{ij}^k x_i(t) y_j(t))$$

where $\mathbf{x}(t)$ is the state vector of activations of the network's N units, $\mathbf{y}(t)$ is the vector of M inputs to the network, and W_{ij}^k is the second-order weight between the i^{th} unit and the j^{th} input for the k^{th} unit. S is a fixed discriminant function. For the examples, we will use the logistic squashing function $\frac{1}{1+e^{-x}}$, but the details of the models in our examples are not important. In order to use our theory the important requirements of a model are that there are as many distinct discrete-time dynamical systems as there are possible inputs (as is the case for first- and second-order RNNs, Elman networks and many variations of these basic RNN architectures).

2 Definitions

Let $\mathcal{I}$ = [0,1] and $i \in$ {e,0,1}.

Definition 2.1 *Consider an N-unit RNN with a fixed input vector, i. This mapping from the unit N-cube ($\mathcal{I}^N$) into itself will be called the i-map.*

We will view an i-map as a dynamical system, with the weights of the RNN as parameters of the system. For a second-order network

the behavior of the system and then attempting to infer system structure in order to elucidate its inner workings. Our work is consistent in spirit with this approach, since we start out assuming that we have a complete understanding of the task[1] (the task in this case is to "act like" a DFA), and then we infer structure in the mediating system.

In the case of RNNs, it is useful to speak of structure in terms of coordinate-independent dynamical properties, as we do here, since coordinate-dependent properties of the system (e.g., sizes and shapes of clusters and distances between them) are confounded by symmetries and redundancies of possible solutions, so no good, general theory can be generated around such ideas. Knowledge about the system performing the calculation might eliminate some of the potential solutions and therefore give a more precise description of solutions to the problem, and in particular what all solutions must have in common. That is, one may be able to use knowledge about the specific family of systems (e.g. the RNN equations or information from the neurosciences) to give a more detailed description of the system and its behavior by eliminating some of the possible solutions.

The main result of this paper is a description of how to study a DFA's minimal state diagram in order to determine up to some equivalence what the system must be doing. Our theory identifies the aspects of a DFA that are important in terms of relating it to the RNN that models it. More generally, we provide a theory of how to determine the salient features of a behavior in order to say which parts of the behavior imply the greatest internal complexity; knowledge which serves to focus exploring the evolution of behaviors in biological organisms or learnability in RNNs.

Yet another way to view this work is as a discussion of representations, an issue central to cognitive science. In particular, we address the representation of relationships and dependencies used during some computations. We give representations in terms of properties of the individual dynamical systems (those dynamical systems we get by giving the system a constant input) and in terms of abstract sets in the RNN phase space, rather than in terms of traces of phase space activities or in terms of the connectivity strengths between units, which may change drastically as the model changes.

Many types of RNNs have been shown to be capable of modeling

[1]This assumption, of course, means that we first hypothesize the full task structure underlying an organism's behavior, which we can never determine empirically if there are an infinite number of potential situations to which the organism must react, and then exploring the complexity that each of the hypotheses would imply in the system.

1 General Discussion

In spite of the interest and progress in the study of recurrent neural networks (RNNs) in the last several years, RNNs are still often seen largely as black boxes without a solid theoretical understanding of how they compute. Many authors have directly or indirectly approached the problem of understanding their inner workings (Cleermans *et al.* (1989) [4], Cummins (1993) [5], Giles *et al.* (1992) [8]) but have provided only a limited understanding for several special cases of the problem. Rather than answering the question for several classes of RNNs modeling specific DFA, we address the deeper question: "what *must* a RNN do in order to model a given DFA?" and we present a unified approach to understanding a wide variety of problems that can be formulated in terms of DFA. By describing what all trained networks have in common, this approach allows one to effectively predict RNN dynamics in a network that models any given DFA. As we shall see, this knowledge of the necessary properties of trained RNN also makes it clear what properties the RNN must have to correctly respond to all possible unseen inputs.

Recently proposed measures of RNNs and their dynamics aim to elucidate the workings of RNNs. For example, Cleermans *et al.* (1989) [4], Elman (1989) [7]and Giles *et al.* (1990) [9], have proposed that cluster analysis of hidden unit activation dynamics yields an adequate understanding of RNN behavior. Others (Pollack (1991) [20]) have taken time series from hidden unit phase space. Still others, for example Williams and Zipser (1989) [28], have used connectivity strengths between units. In this paper we will show that none of these measures are sufficient for understanding RNN behavior in general, and we will describe more useful measures. This paper builds upon ideas first discussed in Casey (1993) [2], but during the preparation of this work, we found that Cummins (1993) [5] had independently discovered some similar results, but our theory includes his results as a special case, answers several questions he poses, and demonstrates why he got the results that he did by clarifying which "environmental constraints" are important.

Cognitive scientists would like to understand complicated natural and artificial systems that were not explicitly designed, e.g. biological organisms or artificial systems with many parameters, that produce some interesting (e.g. "intelligent") behavior. The cognitive science approach to understanding these systems involves carefully observing

How Discrete-Time Recurrent Neural Networks Work

Mike Casey*

Department of Mathematics 0112
UC San Diego
9500 Gilman Dr.
La Jolla, CA 92093-0112

April 11, 1995

Abstract

Second-order recurrent neural networks (RNNs) can learn to model deterministic finite state automata (DFA), and for this task, we examine the extent to which perfect knowledge of task structure constrains possible activation dynamics during computation in RNNs. We show that if an RNN is initialized with small random weights, then it must undergo a bifurcation at some point during learning if it is to model a DFA with properties we will describe later. We also show how to effectively predict activation space dynamics, and thereby allow one to understand RNN computation dynamics in spite of complexity in activation dynamics.

The approach we outline can be used to attempt an understanding of the workings of systems that produce a behavior that is of interest where the systems have not been explicitly designed, e.g. when a learning algorithm has been used or when the system is not man-made (the case of a biological organism).

*Supported by NDSEG Graduate Fellowship, INCOR grant under Professor Michael Freedman and the San Diego Supercomputer Center. e-mail: mcasey@ucsd.edu

How Discrete-Time Recurrent Neural Networks Work

Mike Casey

Department of Mathematics 0112
UC San Diego
9500 Gilman Dr.
La Jolla, CA 92093-0112
E-mail: **mcasey@ucsd.edu**

Technical Report INC-9503
Institute for Neural Computation
University of California, San Diego
9500 Gilman Drive DEPT 0523