CONTEXT DYNAMICS IN NEURAL SEQUENTIAL LEARNING

Kevin G. Kirby
Department of Computer Science and Engineering
Wright State University
Dayton, Ohio 45435

ABSTRACT

A new neural architecture was developed for efficient learning of spatiotemporal dynamics. This architecture reduces the learning problem to two subproblems: (1) the formation of a "context" containing compressed input histories, and (2) the classification of context by an associational algorithm. The first subproblem was handled by introducing a nonlinear dynamical system into the neural network, which can be a low-connectivity random net or a continuous reaction-diffusion system. This enables the solution of the second subproblem to become simpler, requiring only a variant of the classical perceptron learning algorithm. A theoretical framework was developed in which the learning capabilities were analyzed in terms of finite automata theory. A computer simulation system was developed and used to show efficient learning of the sequential parity problem. Further simulations clarified the role of the context subsystem and demonstrated promising non-connectionist architectures for this problem.

INTRODUCTION

The canonical problem in neural network research is the following: *Given a finite subset of the graph of a function* f, *guess what* f *is.* The subset of the graph we are given is called a training set. Elements of the domain are spatial patterns. The algorithm must process this set to come up with a representation for f. The representation can be used to compute f on elements of the domain not appearing in the training set, and this is called "generalization". Often we are interested in adding a temporal dimension. This could mean real time, or merely the information included in *sequences* of input patterns provided to the system. In other words, an output pattern is no longer determined only by an input pattern, but potentially by an unbounded sequence of previous input patterns. If we let X and Y be our input and output sets, respectively, we pass from the interpolation of a function $f: X \rightarrow Y$ to the simulation of a dynamical system $\delta: Q \times X \rightarrow Q$ where Q is the set of states of our observed system. The job of the learning system is to construct an internal model of this observed system. Training sets are now *sequences* of inputs paired with *sequences* of output. A good simulation of δ permits good generalization. Connectionist systems can be harnessed for such computation by using recurrent networks. Indeed, backpropagation formally generalizes fairly easily to the recurrent case, although such an extension seems to be fairly demanding of computational resources (Williams and Zipser 1989). This is a very active area of connectionist research (Giles, et al. 1990, Pineda 1989). Of course, this is by no means the only sense of "temporal" learning. One can also phrase the problem as one of learning a sequence of actions in a network with scalar feedback; this is the work of Klopf (1988).

Proceedings of the 4th Florida Artificial Intelligence Research Symposium, 1991, by the Florida AI Research Society

Our architecture is sketched schematically in Figure 1. We assume a discrete time scale. Spatial input patterns arrive on input lines, and are sent to an output layer of conventional linear threshold neurons, and to an internal subsystem. This internal subsystem has explicit or implicit recurrent connections, and is used to store the state of the system. Unlike the state units in the work of Jordan (1986), the state representation is arbitrary. We call this a *context reverberation (CR) subsystem.* In the work of Gallant and King (1988). enhancing an earlier model of Rosenblatt (1961). a totally connected net of linear threshold neurons was used for a similar function. The output of a sequential system depends on current input plus state. Our architecture captures this dependence in a very straightforward way; output units receive signals from the input units and the CR subsystem.

The important research problem here is this: how can the CR-subsystem send an effective representation of input history to the output layer? Let us define *context* as the dynamical state of the CR subsystem. This concept can be clarified by considering a spectrum of approaches to connectionist architectures for sequential problems, shown in Figure 2. At one extreme, time-delay neural networks send exact delayed copies of the input signals to the output layer. A conventional learning algorithm learns to produce correct output from inputs $\mathbf{x}(t)$, $\mathbf{x}(t-1)$, $\cdots$ $\mathbf{x}(t-TMAX)$, where the input temporal window $[t\text{-}TMAX...t]$ is determined in advance. Hence context in this case exactly corresponds to history. On the other extreme, some sequential problems depend on time and not on history. A clock neural network calculates outputs from the input signal $\mathbf{x}(t)$ coupled with an encoding of t.

In many problems, we need an unbounded temporal window, plus some access to encoded time, but do not want to maintain the overhead of a very large number of time-delayed inputs. The usual approach would be to designate some neurons as state units, and give them recurrent connections. These connections can be learned by "recurrent back-prop", for example. But our architecture differs at a deep level from that approach. In such recurrent adaptation algorithms, the idea is to create a homomorphism from the dynamics of the observed system to the dynamics of the neural net. If δ is the dynamics of the system that we want to model (a finite automaton

that computes parity, for example), we would to set up the weights of our neural net so that its dynamics is given by δ', where there is a mapping h from external states Q to net states Q' that preserves the dynamics. This means that these two dynamics are coordinated by the relation $\delta'(h(q),x) = h(\delta(q,x))$ for all inputs x and states q of the observed system. This homomorphism of finite automata corresponds to the simulation relation: the learning system is supposed to *simulate* the observed system.

Our model constructs a representation of the observed system in a different, innovative sense. We have a mapping ϕ that collapses many states of our CR-subsystem onto each state q of the observed system. This is depicted in Figure 3. The equivalence classes of states, $\phi^{-1}(q)$, correspond to the ellipses in the bottom half of the figure. The set of these classes is called the "quotient space" under the mapping ϕ. It is this quotient space that, as learning proceeds, should come to represent the known system. As time proceeds, each class flows through the CR state space (bottom) tracking the transitions in the observed state space (above). In reality, since the complexity of the CR dynamics is so great, occasionally we will find that generalization off a training set of sequences is poor over long time periods. Referring to the figure, this happens when the dynamics of the CR system does not track the quotient structure. The trajectories wander out of their proper equivalence classes, so the learned internal model would break down a few time steps past the end of the training data. A good CR architecture will make this a rare occurrence.

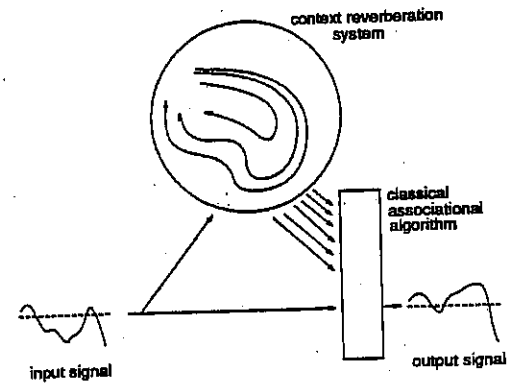


Figure 1. The context reverberation architecture.

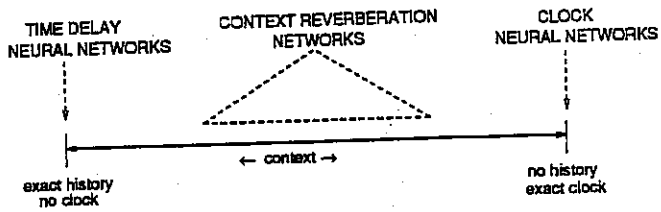


Figure 2. The spectrum of context usage.

LOW-CONNECTIVITY ARCHITECTURES FOR CONTEXT REVERBERATION

Having discussed the role of the CR subsystem, we now turn to its implementation. We first investigated a connectionist architecture, in which the CR subsystem contained a number of linear threshold units arranged in a grid, each connected to others within a limited neighborhood. (This contrasts with the total (i.e., $O(n^2)$) connectivity in the nets of Hopfield, Anderson, and others.) Synaptic weights are fixed and randomized. We had known from the work of Gallant and King (1988) that a totally connected layer of random hidden units could learn sequential problems with some success. But dozens of units in a such a highly connected system will result in hundreds of wires. Our first step showed that for the so-called "robot plan task", low connectivity was more efficient (Kirby and Day 1990). An "autopsy" of the networks (from large sequences of randomly generated trials) showed that the most successful ones were those that had highly irregular dynamical trajectories. We can see this if we plot the intensity of the firing states of the CR net versus time, for an arbitrary clamped input. (Intensity here means we add up the +1/-1 (firing/silent) values.) This is shown in the two plots at the bottom of Figure 4. The first plot shows a "good" CR net, which learned the task quickly, and the bottom a "bad" net that failed to learn. This shows that aperiodicity (or, strictly speaking, periods of length much longer than the time scales of interest) is an attribute correlated with good CR performance. One way to control this is to adjust the "gain" signal, which amplifies the input signals coming in to the hidden net. This is shown in the plot at the top of Figure 4. Low gain yields better results.

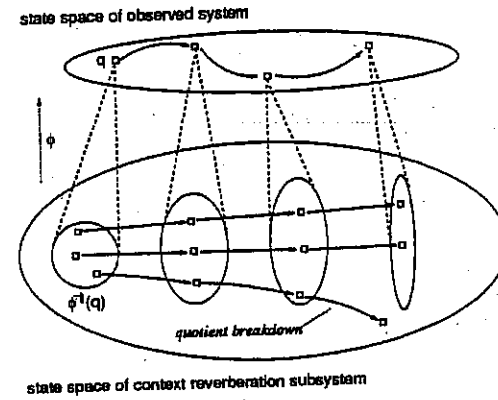


Figure 3. The over-representation of states by context.

The robot plan task, however, merely involves memorizing a set of sequence pairs, and does not address the issue of generalization. To go beyond this, we had our network learn the sequential

parity automaton (Figure 5.) The output at time t is the parity of the string of binary inputs from time 0 to time t. We used connectionist CR nets with various neighborhood sizes, and plotted the learning times in Figure 6. The perceptron algorithm was used to make the input/context associations. The solid lines plot the number of perceptron epochs (passes through the training set), when trying to learn parity from 20 sequences of duration equal to 6 time units. We used 128 CR units, and averaged over 10 different randomly seeded configurations. We only plot the result for cases when every instance learning the training data perfectly (zero error). The dashed lines show the performance of an unseen test set of bit sequences. This shows that not only is lower connectivity dramatically more efficient in terms of required connections, but it is even more efficient in absolute learning time. Preliminary results from a genetic modification scheme, which remove unchanging context units and swap weight values, showed modest improvements in learning rates with no significant change in generalization ability.

NON-CONNECTIONIST EXTENSIONS

We can conclude from the experiments just described that the *locality* of the connections in the CR architecture is a feature to be exploited. This is an economic issue; fewer connections require less space and ease hardware implementations. But it also allows us to more rigorously investigate the dynamical properties of CR systems. In this section we discuss our work in relation to work in the dynamical systems disciplines. Insights from these disciplines are important, because they help us understand and extend the capabilities of the CR architecture.

Our connectionist CR-subsystem uses a random network. Kauffman (1989) showed that networks of totally connected random boolean units exhibit an exponential growth in limit cycle length as the number of units increases. (A limit cycle is one period of the state trajectory.) In other words, a the state trajectories of even a small net are highly aperiodic. This is termed a "chaotic" phase, as opposed to the so-called "ordered phase" when cycle lengths increase polynomially with the number of units. Reducing from global to local connectivity slows this growth. With only 2-neighbor connectivity the periodicity grows as $\sqrt{N}$, too slowly for effective use as a context reverberation net. Kürten (1988) studies threshold units with different local connectivities and shows that whereas low-connectivity systems in which neighbors are chosen randomly have exponential growth in cycle lengths, nearest-neighbor systems may show linear growth. The low-connectivity nets studied were the 3-neighbor "honeycomb" lattice, and the self+4-neighbor square lattice. The zero-threshold honeycomb lattice shows linear growth in cycle length, and the zero-threshold square lattice with self-feedback shows exponential growth. Adding a unit threshold moves this system back into an ordered phase. Our research suggests that we should seek a chaotic phase in our context-reverberation systems. So if we are committed to using a local net of conventional neurons, the connectivity level (i.e., neighborhood size) should be at least 4. For connectivity less than this, learning should be impossible. We have experimentally confirmed this; the curve in Figure 6 goes to infinity on the left, when connectivity approaches 3.

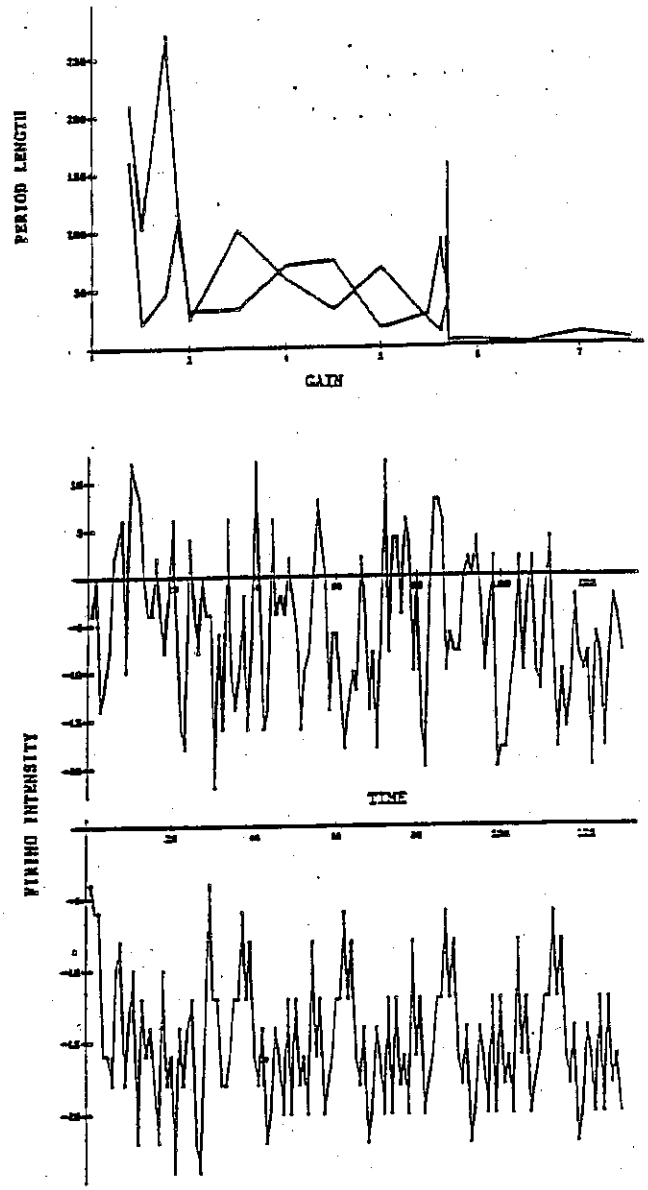


Figure 4. Periodicity and dependence of period on gain

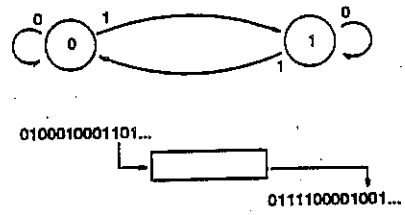


Figure 5. The state transition graph for sequential parity.

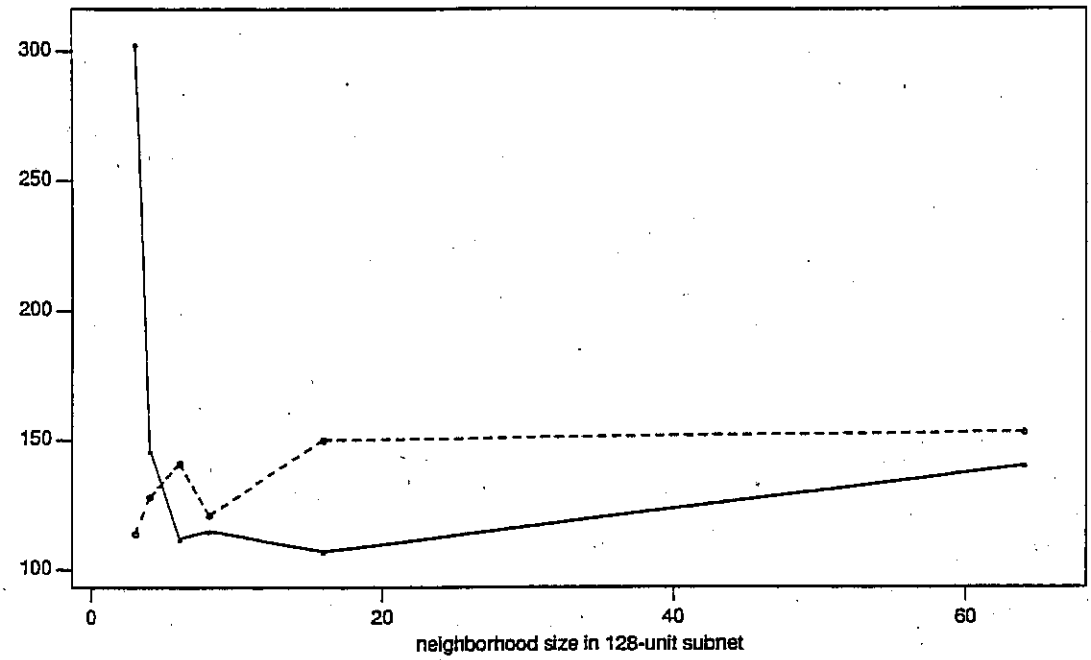


Figure 6. Learning times (epochs, solid) and generalization accuracy (test set fraction incorrect ×1000, dashed) versus connectivity for the sequential parity problem, for neighborhood sizes of 3, 4, 6, 8, 16 and 64.

Given the efficiency of low-connectivity threshold lattice automata as chaotic reverberation subsystems, we can investigate the potential of continuous local dynamical systems for setting up context. The general case is a reaction-diffusion equation, of the form

$$\frac{\partial u(\mu,t)}{\partial t} = \nabla \cdot D(\mu)\nabla u(\mu,t) + R[u(\mu,t)] \qquad (1)$$

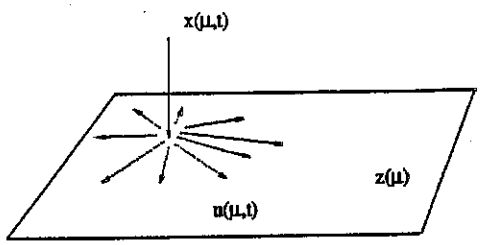
Here $u(\mu,t)$ is an excitation signal diffusing across a space with coordinates μ. D is the diffusion coefficient, which may vary across the space. R is the reaction term, a function of the excitation level. In the two terms we have the two ingredients necessary for effective context reverberation: local communication (via the diffusion term), and local computation (via the reaction term). In one dimension, in analogy to the discrete set of threshold units we used for learning the parity problem in the previous section), we can compartmentalize the system to create a ring of compartments. The diffusion term with discretized compartments becomes $\sum_{j=k\pm1} d_{jk}(u_j - u_k)$. This system was introduced by Alan Turing (1952) to study the destabilizing effect of diffusion in morphogenesis. (Turing used two diffusing signals.) Othmer and Scriven (1971) examined how the dynamical properties of this reaction-diffusion system depended on the topology, studying rings and lattices in what was a continuous analog to the studies of Kürten (1988) on lattice automata cited above.

Can we expect such continuous dynamics to improve our CR-subsystems? We claim that it should be possible by using a neuronal model based on the Turing morphogenesis equations. This model is called the *reaction-diffusion neuron* (Kirby and Conrad 1984; Kirby, Conrad and Kampfner 1981), a continuous extension of a discrete linear unit used in a Darwinian brain model. These neurons take input signals and map them into continuous gradients, which evolve by reaction-diffusion equations. Gradients are read by spatially fixed sensors, whose response induces the firing of the neuron. With suitable dynamics one reaction-diffusion neuron can play the role of an entire CR-subnetwork.

Once we allow continuous dynamics to enhance the CR system, we can consider adaptation of this dynamics. Recall that in the experiments discussed so far, the CR-subsystem was random and fixed, except possibly for sporadic localized genetic modification. This allows the context dynamics to evolve concurrently with the representation. We can view this idea of context as analogous to some phenomena observed in other research. In the continuous case we can have formation of topological features such as those that arise in Turing-type morphogenesis systems, e.g., the stripes of Meinhardt and Gierer (1980). A well-known neural analog is in the work of Amari (1977) on pattern formation in neural fields. A neural field changes the representation of neurons from a set of finite units to a manifold of mathematical points. Firing signals do not propagate along connections, but spread out along the manifold, governed by equations of the form:

$$\tau \frac{\partial u(\mu,t)}{\partial t} = -u(\mu,t) + \int w(\mu,\mu') f[u(\mu',t)]\, d\mu' + \text{input term} \qquad (2)$$

The weighting "matrix" $w(\mu,\mu')$ internal to the neural field is fixed in advance; only the input weights change (according to a Hebbian algorithm). Efferents from the neural field in a sense use the acquired patterns as state information, since the topographic arrangement on the field is also a kind of repository for input history. Let us call the kind of input context in these morphogenesis and neural field models *topographic context*. This contrasts with the concept of *scrambled context* used by our CR systems to represent history.



$$u = Ix - hy + \int \nabla \cdot D \nabla u + R(u)\ dt$$

$$y = \alpha\left[\int uz\ d^2\mu - \theta \right]$$

Figure 7. The dynamics of the reaction-diffusion neuron.

CONCLUSIONS

Scrambled and topographic context promise to be important notions in the theory of sequential learning. We have shown that the requirements for a good context-reverberation subsystem do not include the high connectivity required for connectionist solutions to other problems. This may encourage molecular electronic hardware implementations (Hong 1986). Low-connectivity nets of linear threshold functions with nearest-neighbor topology provide long cycle lengths and are an effective means for providing scrambled context information to a single-layer learning algorithm for learning the parity dynamics. We believe that such a result is encouraging for the study of non-symbolic non-connectionist continuous systems for solving "real" artificial intelligence problems. Far from merely providing a new technology for machine learning, the CR system has produced fertile ideas enabling a more profound understanding of the learning problem itself.

ACKNOWLEDGEMENTS

This work was supported by the Air Force Office of Scientific Research. The author acknowledges many valuable discussions with Louis Tamborino, Michael Conrad, Qiang Gan, and Nancy Day.

REFERENCES

Amari, S. 1977. "Dynamics of Pattern Formation in Lateral-Inhibition Type Neural Fields," **Biological Cybernetics.** Vol. 27, pp. 77-87.

Doya, K. and S. Yoshizawa. 1990. "Memorizing Oscillatory Patterns in The Analog Neuron Network," **Proc. IEEE/INNS Conference on Neural Networks,** pp. I27-I32.

Giles, C.L., G.Z. Sun, H.H. Chen, Y.C. Lee, and D. Chen. 1990. "Higher-Order Recurrent Networks and Grammatical Inference," **Neural Information Processing Systems 2,** D. Touretzky, Ed., San Mateo, California, Morgan-Kaufmann, pp. 380-387.

Hong, F.T. 1986. "The Bacteriorhodopsin Model Membrane System as a Prototype Molecular Computing Element," **Biosystems.** Vol. 19, pp. 223-236.

Jordan, M.I. 1986. "Serial Order: A Parallel, Distributed Processing Approach," Institute for Cognitive Science Report 8604, University of California, San Diego.

Gallant, S.I., and D.J. King. 1988. "Experiments with Sequential Associative Memories," **Cognitive Science Society Conference,** Montreal.

Kauffman, S.A. 1989. "Principles of Adaptation in Complex Systems," In **Lectures in the Sciences of Complexity,** D. Stein, Ed., Addison-Wesley.

Kürten, K.E. 1988. "Dynamical Properties of Threshold Automata with Nearest-Neighbor Interactions on a Regular Lattice," **Proc. IEEE International Conference on Neural Networks,** pp. I37-I43.

Kirby, K.G., and M. Conrad. 1984. "The Enzymatic Neuron as a Reaction-Diffusion Network of Cyclic Nucleotides," **Bulletin of Mathematical Biology.** Vol. 46, pp. 765-783.

Kirby, K.G., M. Conrad, and R. Kampfner. 1991. "Evolutionary Learning in Reaction-Diffusion Neurons," **Applied Mathematics and Computation.** (To appear.)

Kirby, K.G. 1989. "Information Processing in the Lorenz-Turing Neuron," **Proc. IEEE Engineering in Medicine and Biology Conference,** Molecular Electronics Track, pp. 1358-1359.

Kirby, K.G., and N. Day. 1990. "The Neurodynamics of Context-Reverberation Learning," **Proc. IEEE Engineering in Medicine and Biology Conference,** Molecular Electronics Track, pp. 1781-1782.

Klopf, A.H. 1988. "A Neuronal Model of Classical Conditioning," **Psychobiology.** Vol. 16, pp. 85-125.

Meinhardt, H., and A. Gierer. 1980. "Generation and Regeneration of Sequences of Structures During Morphogenesis," **Journal of Theoretical Biology.** Vol. 85, pp. 429-450.

Othmer, H.G., and L.E. Scriven. 1971. "Instability and Dynamic Pattern in Cellular Networks," **Journal of Theoretical Biology.** Vol. 32, pp. 507-537.

Pineda, F.J. 1989. "Recurrent Backpropagation and the Dynamical Approach to Adaptive Neural Computation," **Neural Computation.** Vol. 1, pp. 161-172.

Rosenblatt, F. 1961. **Principles of Neurodynamics,** Washington, DC, Spartan Press.

Turing, A.M., 1952. "The Chemical Basis of Morphogenesis," **Phil. Trans. Royal Soc. B** Vol. 237, pp. 37-72.

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