Dynamic Recurrent Neural Networks: Theory and Applications

C. Lee Giles, Gary M. Kuhn, and Ronald J. Williams, *Member, IEEE*

INTEREST in dynamical recurrent neural networks is not new. The earliest work seems to be that of [6]. In Section II of their paper (The Theory: Nets without Circles) they developed models for feed-forward networks which had time dependences and time delays. However, these networks were constructed from the threshold logic variety of neuron. In Section III (The Theory: Nets with Circles) McCulloch and Pitts extend their networks to those with "circulating" or "dynamic memory." These networks had feedback.

Kleene [4] was motivated by this work. In his own words: "Finally, we repeat that we are investigating McCulloch-Pitts nerve nets only partly for their own sake as providing a simplified model of nervous activity, but also as an illustration of the general theory of automata, including robots, computing machines and the like." Kleene did not find section III of MCulloch and Pitts easy to understand and was forced to proceed independently. He modeled the McCulloch-Pitts networks as finite automata and their language as a regular language and proved the equivalence of the two. His work is usually referenced as the first work on finite automata [11].

Minsky extended the McCulloch-Pitts model to include more conventional types of time dependencies, static memories and flip-flops [7], [8] and proved how networks of McCulloch-Pitts neurons are equivalent to finite-state machines in general.

In a mathematical sense, the training of dynamical recurrent neural networks is also not new. One is reminded of the bourgeois gentilhomme of Molière, who was surprised when he was told that he had been speaking prose all his life. Similarly, practitioners of training with feedforward nets may be surprised to be told that they have been doing a kind of recurrent training all along!

If we take first-order feedforward networks as an example, the gradient-based weight change for weight w_{ji} from unit i to unit j is usually shown as something like

$$\Delta w_{ji} \propto \delta_j y_i.$$

This weight change factors into $\delta_j = \frac{-\partial E}{\partial x_j}$ and $y_i = \frac{\partial x_j}{\partial w_{ji}}$, where E is a training error, x_j is the weighted sum into node j of the network, and $y = f(x)$ is often the logistic function $(1 + e^{-x})^{-1}$.

Anyone who can count would say that there is only one term, namely y_i, in the second factor of this chain rule. Would they be right?

No. Since $x_j = \sum_i w_{ji} y_i$ it follows that

$$\frac{\partial}{\partial w_{ji}} x_j = w_{ji} \frac{\partial}{\partial w_{ji}} y_i + y_i$$

is the derivative of a product and has *two* terms.

The point is that the first term on the right hand side of the equation $w_{ji} \frac{\partial}{\partial w_{ji}} y_i$, has been there all along in the feedforward case. But because there was no recurrence (no loop from the output of a unit back to its input), a change in a weight on a connection *from* unit i could not affect the output of unit i. So the practitioners of training with feedforward nets have in fact been doing a kind of recurrent training, namely a kind where the amount of recurrence is...zero!

This special issue illustrates both the scientific trends of the early work in recurrent neural networks, and the mathematics of training when at least some recurrent terms of the network derivatives can be non-zero. The following is a brief description of each of the papers. We have organized this description into two parts. The first part contains the papers that are mainly theoretical, and the second part contains the papers that are mainly applications. The order of papers is alphabetical by first author.

I. Theoretical Papers

Bengio, Simard, and Frasconi set up three conditions for a parametric dynamical system to learn and store relevant state information: that the system be able to store information for an arbitrary duration of time; that the system be resistant to noise; and that the parameters of the system be trainable in reasonable amount of time. They show that gradient-based training methods fail to meet the first two criteria and as a consequence ineffective for learning long-term time dependencies. They then show the performance for both state retention and training time for the training algorithms of backpropagation, pseudo-Newton, time-weighted pseudo-Newton, discrete error propagation, multi-grid random search and simulated annealing, on three state retention problems; latch memory, 2-sequences, and parity. In general, the non-gradient methods produced better networks for memory retention, but took much longer to train.

The paper by Bianchini, Gori, and Maggini analyses the problem of optimal learning in recurrent networks, by proposing some sufficient conditions which guarantee that error surfaces are free of local minima. Formal relations are established between feedforward and recurrent networks, so that examples of local minima for feedforward networks can be

1045–9227/94$04.00 © 1994 IEEE

associated with analogous ones in recurrent networks. Also, the constructive role of the analysis is shown, for the designing of networks suitable for a given task.

Nerrand *et al.* show the importance of choosing the appropriate training algorithm for the training of a dynamical system in the presence of noise. Through simulations of nonlinear processes with and without noise, they show that directed (teacher forcing) algorithms seem to be better for prediction when modeling a noiseless system or one that has white noise added to the system's state variables. However, when the white noise is added to the output, undirected training algorithms appear to be better predictors. Thus, some knowledge of the locations of noise in a dynamical system can be very helpful in choosing a model for training and prediction.

The article by Olurotimi first proves that any recurrent network is equivalent to a kind of layered network in which the hidden layer has no recurrent connections and the output layer units have self-recurrent connections as well as feedback connections to the hidden units. This result is then used to yield a method for learning the weights in any fully recurrent network using only feedforward learning, under the assumption that a reasonable state representation can be constructed from the input-output data, for example, by using n-fold derivatives of the output for suitable n.

The article by Piche provides both a tutorial overview of the two basic strategies used for computing error gradients in recurrent neural networks (or, more generally, in any discrete-time dynamical system) and an analysis of the computational requirements of the various forms such algorithms may take. Also presented are examples illustrating the application of these techniques to nonlinear controller design and adaptive noise cancellation.

The paper by Principe, Kuo, and Celebi is a vector space study of short-term memory structures in dynamic recurrent networks. For the special case of gamma memory structures, which are another example of what Back and Tsoi call "LRGF architectures" [1], the study shows the following advantage of recurrence: the recurrent system is able to control the angle between the signal vector and its projection in the memory space. In a feedforward system that angle would always be fixed.

The article by Srinivasan, Prasad, and Rao provides several interesting and novel results, including: 1) a convergence proof for both backpropagation through time and the algorithm known variously as dynamic backpropagation [10] forward propagation [5], or real-time recurrent learning [15]; 2) a proof that truncated backpropagation through time is sufficient for convergence; and 3) a description of the explicit form that backpropagation through time takes for ARMA models.

The paper by Tsoi and Back provides an overview of the architectures of locally-recurrent locally-feedback networks. The architectures reviewed: Back, Tsoi [1]; Frasconi, Gori, Soda [2]; de Vries and Principe [14]; and Poddar and Unnikrishnan [12]; are compared as to whether the synaptic type is simple or dynamic and whether the feedback location is in any or all of the synapses, activations and outputs. From this comparison, it is evident that other architectural models are possible. In addition, a collection of issues are raised regarding universal approximated properties, location and placement of feedback, differences between locally and globally recurrent networks, state-dependent models, structural robustness, etc.

II. Applications Papers

The paper by Connor, Martin, and Atlas applies a robust learning algorithm to the training of a dynamic recurrent neural network. The robust algorithm is reminiscent of the probabilistic weighting of data for hidden Markov model or EM training, in that outliers are probabilistically filtered while the parameters for the network are estimated. In the chosen application, the robust algorithm yields considerable improvement on prediction of the Puget Power Electric Demand time series.

The paper by Parlos, Chong, and Atiya describes work on the training and validation of a dynamic recurrent neural network for control of a heat exchanger. Batch training was successfully supplemented by on-line training, to achieve further reduction in mean square error. This paper is an example of the kind of close-to-real system identification that is often under-reported in the literature.

The article by Kechriotis, Zervas, and Manolakos presents results on the use of recurrent neural networks as adaptive communication channel equalizers. This is a potentially important application area, and their simulations show that small recurrent nets can outperform both traditional linear filter equalizers and multilayer feedforward net equalizers of larger size. Because very small recurrent networks are used, on-line gradient algorithms that scale poorly with network size can still be effective in this application.

Neurocontrol of nonlinear dynamical systems with Kalman-filter trained recurrent networks is discussed in the paper by Puskorius and Feldkamp. They show how dynamic recurrent neural networks can be trained with parameter-based extended Kalman filters (EKF). They illustrate this training by the application of Kalman-trained neural networks for a wide range of control problems: the cart-pole problem, the bioreactor benchmark and engine idle speed control.

The paper by Robinson shows that a dynamic recurrent neural network can make a good probability estimator for use in phonetically based speech recognition. Some details of the training algorithm are quite unexpected. The probability values output by the network can be used by a hidden Markov model in the place of values traditionally calculated by Gaussian Mixtures.

The article by Sastry, Santharam, and Unnikrishnan studies a form of recurrent network that is essentially a feedforward net in which each node is followed by a single-pole adaptive filter. Such a net thus has recurrent connections of a limited form and is an example of what Back and Tsoi [1] call a locally recurrent, globally feedforward architecture. Also presented are a gradient-based algorithm for determining the weights that ignores dependencies on past time and weak convergence results for this algorithm. A significant portion of this article is devoted to the study of the behavior of the architecture and learning algorithm when applied to the identification and control of dynamical systems. This study yields favorable results on several problems first considered by Narendra and Parthasarathy [10].

Discrete recurrent networks for grammatical inference are described by Zeng, Goodman, and Smyth. This neural network is an extension of a previously discussed hybrid neural network structure of Giles and Sun [3], [13] that connects a recurrent neural network to a discrete stack. A common error function enables the network and stack to be trained concurrently. When trained this neural system emulates a neural network pushdown automaton. What is different from previous work is that a new pseudo-gradient training algorithm trains the network to use the stack and to form the internal networks states that are extremely stable and can classify long unseen strings. These stable internal states enable the networks learned internal representation of its pushdown automaton to be readily extracted.

REFERENCES

[1] A. D. Back and A.C. Tsoi, "FIR and IIR synapses, a new neural network architecture for time series modelling," *Neural Computation*, vol. 3, no. 3, pp. 337–350, 1991.

[2] P. Frasconi, M. Gori and G. Soda, "Local feedback multilayered networks," *Neural Computation*, vol. 4, pp. 120–130, 1992.

[3] C. L. Giles, G. Z. Sun, H.H. Chen, Y. C. Lee, and D. Chen, "Higher order recurrent networks and grammatical inference," in *Advances in Neural Information Processing Systems 2,* D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann Publishers, 1990, pp. 380-387.

[4] S.C. Kleene, "Representation of events in nerve nets and finite automata," *Automata Studies*, C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton University Press, 1956, pp. 3–42.

[5] G. M. Kuhn, R. L. Watrous, and B. Ladendorf, "Connected recognition with a recurrent network," *Speech Communication*, vol. 9, no. 1, pp. 41–48, 1990.

[6] W. S. McCulloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[7] M. L. Minsky, *Computation: Finite and Infinite Machines.* chap. 3, "Neural networks: Automata made up of parts," Englewood Cliffs, NJ: Prentice-Hall, Inc., 1967, pp. 32–66.

[8] M. L. Minsky, "Neural-analog networks and the brain-model problem," Ph.D. thesis, Princeton University, 1954.

[9] M. L. Minsky and S. A. Papert, *Perceptrons.* Cambridge, MA: MIT Press, 1967.

[10] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. on Neural Networks*, vol. 1, no. 1, p. 4, 1990.

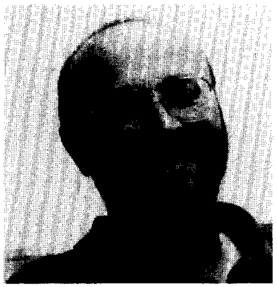
[11] D. Perrin, "Finite automata," in *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, J. van Leeuwen, Ed. Cambridge, MA: The MIT Press, 1990.

[12] P. Poddar and K. P. Unnikrishnan, "Nonlinear prediction of speech signals using memory neuron networks," in *Neural Networks for Signal Processing: Proceedings of the 1991 IEEE Workshop*, B. H. Juang, S. Y. Kung, and C. A. Camm, Eds. Piscataway, NJ: IEEE Press, 1991, pp. 395–404.

[13] G. Z. Sun and C. L. Giles and H. H. Chen and Y. C. Lee, "The neural network pushdown automaton: model, stack, and learning simulations," Technical Report, UMIACS-TR-93-77, Institute for Advanced Computer Studies, University of Maryland, College Park, MD.

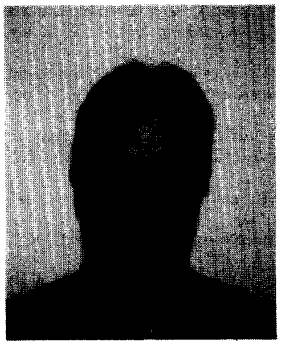
[14] B. de Vries and J. Principe, "The gamma model—A new neural network for temporal processing," *Neural Networks*, vol. 5, no. 4, pp. 565–576, 1992.

[15] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks,"*Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.



C. Lee Giles is a Senior Research Scientist at NEC Research Institute, Princeton, NJ and an Adjunct Associate Professor at the Institute for Advanced Computer Studies at the University of Maryland, College Park, MD. His current research interests are in recurrent neural networks and their computational capabilities and use in hybrid systems and in adaptive optical computing and processing. He serves on many related conference program committees and has helped organize many related meetings and workshops and has been an advisor and reviewer to government and university programs in both optical computing and processing and in neural networks. He was one of the founding members of the Governors Board of the International Neural Network Society. He has served or is currently serving on the editorial boards of Applied Optics, IEEE Transactions on Neural Networks, Journal of Parallel and Distributed Computing, Neural Networks, Optical Computing and Processing, and Academic Press. He is a member of IEEE, the International Neural Network Society and the Optical Society of America. Previously, he was a Program Manager at the Air Force Office of Scientific Research in Washington, D.C. where he initiated and managed research programs in Neural Networks and in Optical Computing. Before that he was a research scientist at the Naval Research Laboratory, Washington, D.C. and an Assistant Professor of Electrical and Computer Engineering at Clarkson University. Before graduate school he was a research engineer at Ford Motor Scientific Research Laboratory. His advanced degrees include a Ph.D. in Optical Sciences from the University of Arizona, a M.S. in Physics from the University of Michigan, a B.S. in Engineering Physics from the University of Tennessee and a B.A. from Rhodes College.

Gary M. Kuhn, (M'78), received the Baccalaureat es Lettres in Rouen, France, in 1967, the B.A. from St. Lawrence University in 1968, the M.A. from Middlebury College in 1969, and the Ph.D. in Linguistics from the University of Connecticut in 1977. He worked on a reading machine for the blind, as a Research Assistant at the Haskins Laboratories, New Haven, CT, from 1969 to 1976, and as a consultant to Kurzweil Computer Products, Inc., in 1976. From 1976 to 1992 he worked at the Institute for Defense Analyses, Communications Research Division, Princeton, NJ, with a sabbatical at the Joint Speech Research Unit, Cheltenham, England, in 1984. Since 1992 he has been employed by Siemens Corporate Research, Princeton, NJ, where his activities are in the area of signal processing and neural networks. He has served as a member of the Speech Technical Committee of the Acoustical Society of America, as a thesis advisor in Computer Science at the University of Pennsylvania, as a founding member of the Technical Committee on Neural Networks for Signal Processing of the IEEE Signal Processing Society, and as an organizer or Co-Chair of the 1991–1993 IEEE Workshops on Neural Networks for Signal Processing. Dr. Kuhn is a member of the Acoustical Society of America, the European Speech Communication Association, the International Society of Phonetic Sciences, the International Neural Network Society, and the IEEE Signal Processing Society. His current research is in the areas of pattern recognition and system identification.



Ronald J. Williams earned a B.S. in mathematics from the California Institute of Technology in 1966 and an M.A. and Ph.D. in mathematics from the University of California at San Diego in 1972 and 1975, respectively. He is currently an Associate Professor of Computer Science at Northeastern University, a position he has held since 1986. From 1983 to 1986 he was a member of the Parallel Distributed Processing Research Group at UCSD's Institute for Cognitive Science, where he studied learning algorithms for artificial neural networks. He has served as an Associate Editor of the IEEE Transactions on Neural Networks and is currently an Associate Editor of Adaptive Behavior. His current research interests focus on learning control methods using dynamic programming-based reinforcement learning algorithms.