

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 3 hours per response, including the time for reviewing instructions, searching existing data sources, gathering the data needed, reviewing the collection of information, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank) 2. REPORT DATE 7/28/94 3. REPORT TYPE AND DATES COVERED Final - 5/1/91-4/30/94

4. TITLE AND SUBTITLE Connectionist Models for Intelligent Computation (21) 5. FUNDING NUMBERS G AFOSR-91-0257
61102F
2305/B3

6. AUTHOR(S) H.H. Chen Y.C. Lee

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Maryland Colege Park, Maryland 20742 B AFOSR-TR-95-G 467

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/PKA 110 Duncan Avenue, Suite B115 Bolling AFB, DC 20332-0001 nm 10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFOSR-91-0257

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED for Public Release: Distribution Unlimited 12b. DISTRIBUTION CODE UL

13. ABSTRACT (Maximum 200 words) This final report covers the work done by our group of neural network computing at the University of Maryland for the past three years. We studied the neural network's capability of processing temporal or sequential data. Recurrent neural networks were used to perform inference on grammars. An external memory stack was constructed to work with the neural network to perform inferences on context free languages. And finally, a spatially homogeneous locally connected recurrent neural network that could simulate any given turing machine, including the universal Turing machine was devised. It is capable of performing universal computations and demonstrated the universal power of recurrent neural network architectures. To train these sequential neural net machine, we have investigated the forward propagating learning algorithms. DTIC QUALITY INSPECTED 8

14. SUBJECT TERMS Nerual network, temporal data, sequential data, recurrent 15. NUMBER OF PAGES 8 16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified 20. LIMITATION OF ABSTRACT UL SAC

FINAL REPORT

**CONNECTIONIST MODELS
FOR INTELLIGENT COMPUTATION**

Covering Period

May 1, 1991 —April 30, 1994

Submitted to

Air Force Office of Scientific Research/

by

Laboratory for Plasma Research
University of Maryland
College Park, Maryland 20742

Principal Investigators: H. H. Chen
Y. C. Lee

Approved for public release,
distribution unlimited

19950724 043

ABSTRACT

This final report covers the work done by our group of neural network computing at University of Maryland for the past three years under the sponsorship of AFOSR. During this grant period, we studied the neural network's capability of processing temporal or sequential data. Recurrent neural networks were used to perform inference on grammars. An external memory stack was constructed to work with the neural network to perform inferences on context free languages. And finally, a spatially homogeneous, locally connected, recurrent neural network that could simulate any given Turing machine, including the universal Turing machine was devised. It is capable of performing universal computations and demonstrated the universal power of recurrent neural network architectures. In order to train these sequential neural net machines, we have investigated the forward propagating learning algorithms. A fast learning algorithm is proposed that could reduce the computation complexity from $O(N^4T)$ to $O(N^3T)$. This algorithm was tested on a continuous temporal problem that will be the next phase of our research effort.

I. INTRODUCTION

Artificial neural networks are very powerful constructs. It is supposed to simulate the brain structure of human to perform intelligent tasks. However, most of the current research in this area seem to treat neural net as only that of a functional mapper. It is used to conduct an input- output mapping. We think that most intelligent tasks involve the processing of temperol or sequential signals. Therefore, neural networks must be able to extract generation rules from sequential patterns, i.e. grammatical inferences. In the past three years, our project dealt with topics surrounding such issues. Neural networks with recursive connections were chosen to process such sequential data that are correlated in both space and time. The recurrent connections that serve as a memory of history of the sequence and empower the neural network to extract temporal orders out of the sequential patterns. A recurrent network in itself behaves like a finite state machine if we can cluster the neuronal state and quantizt them. Indeed, numerical simulations showed that recurrent neural net can be easily trained to do just that. Perfect finite state machine can be extracted from only a handful amount of data from a sequence. Theoretical analysis also established that the computational power of a finite state machine would increase tremendously if it is coupled with a memory stack. The resulting pushdown automata could recognize an extended set of language called context free language which is much more expressive than the finite state machine grammar. Even more computational power could be achieved by replacing the stack memory with an infinite tape and empower the neural network finite state controller to erase and write on the tape and thus becoming a Turing machine. Once established that neural net can fully simulate universal Turing machine, we have no doubt that neural net should have the full power to simulate any given intelligent task.

Availability Codes	
Dist	Avail and/or Special
A-1	

II. Simulation of Finite State Machine

As in most applications of neural nets, the topology of the connection weights in the net is crucial to the success of the applications. We studied the issue regarding the order of connections to the simulation of finite state machines and pushdown automata. Specifically, we established that:

Theorem 1: For any given finite state machine(FSM) with N states and M input symbols, there are at least one *second order* connected recurrent neural net (RNN) with N state neurons and M input neurons that can *directly* simulate the FSM.

This theorem ensures us that RNN can be used to learn regular grammar since the existence of the solution is guaranteed. On the other hand, we can also show that

Theorem 2: There are *some* FSM structure that *can not be directly* simulated by any RNN with *first order* connections without hidden neurons.

A notable example is the four state loop transition diagram of a dual parity finite state grammar. However, this does not imply that first order recurrent net cannot learn the dual parity grammar. Because we can also prove the following

Theorem 3: There are at least one first order RNN with at most NM neurons that can simulate a given FSM indirectly. That is, it can simulate an equivalent FSM to the given FSM and this equivalent machine can be obtained automatically.

Theorem 4: In a second order RNN with S state neurons and M input neurons, the probability of it to simulate a given finite state machine with N states and M input symbols is given by

$$P = \left[\frac{L(N, S)}{2^N} \right]^{SM}$$

where $L(N,S)$ is the number of dichotomies that can be implemented by a S -dimensional perceptron for an N -input pattern.

These theorems give us good guidance in our choice of different RNN connection topologies for various grammatical inference tasks. Similar results for neural net simulation of PDA were also obtained.

Theorem 5: For any given deterministic pushdown automata with N states, M symbols and M stack symbols, there exists a third order RNN coupled with an external stack memory that can simulate it completely.

Likewise we also have an estimate of the capacity of a K-th ordered RNN to simulate finite state machines.

Theorem 6: The capacity of a K-th order RNN with N recurrent neurons, $C(N,K)$, can be inferred from the recursion formula:

$$C(N, K) = C(N - 1, K) + C(N - 1, K - 1)$$

III. Simulation of Pushdown Automata with enhanced RNN:

In previous work we have developed a RNN pushdown that uses a third order connected recurrent neural net controller to operate an analog stack memory. Many context free grammars were learned by this construct from a limited set of positive and negative samples. A typical example is renthesis balance checker.

It came to our attention later that several hard grammar such as the Palindrome is very difficult to learn by the above system. To solve this problem, we devised an enhanced recurrent neural net with "full order" connections. It turns out that this enhanced neural net can learn grammars such as the Palindrome very easily. It automatically figured out some very tricky transition rules associated with such grammar. After quantization, the learned rules are again exact and the generalization to test other samples of this grammar is again infinite.

IV. Neural net Turing Machine

Turing machine is the most powerful sequential machine that is capable of universal computations. The ability of the neural net to simulate Turing machines is therefore an important issue in neural computing. In the past year, we have succeeded in constructing just such a neural net Turing machine. The finite state controller and the tape symbols are represented by neurons arranged into a row of columns. Each neuron is locally connected to other neurons in the same and neighboring columns. The detailed values of these weights and the specific construction of the network is beyond this report. We shall summarize our results in the following two theorems.

Theorem 1: Given an arbitrary deterministic Turing machine with M symbols and N states, there exists a neural network with $M+2N+1$ rows of neurons and a set of second order locally connected weights that can simulate it in 2-to-1 time.

Theorem 2: Given an arbitrary deterministic Turing machine with M symbols and N states, there exists a neural net with $M+N+2$ rows of neurons and two sets of second order locally connected weights that can simulate it in real time.

Currently, we are studying the training of the neural net Turing machine to recognize some

simple grammars, for example, the parenthesis checker.

V. Green Function Method for Fast On-line Training of Recurrent Neural Networks

In processing temporal or sequential signals, recurrent neural network is found to be able to capture most of the complex temporal orders and correlations. However, a particular pressing issue concerning recurrent net is the lack of an efficient on-line training algorithm especially when we are dealing with applications that would require large number of internal states or neurons.

Among the currently popular training algorithms, the error back propagation method is not an on-line algorithms since we have to wait until the signals propagate all the way to the output layer to obtain the error message and then propagate back the error to each layer for weight corrections. The Williams and Zipser error prediction forward propagation algorithm is indeed on-line. However, it is very expensive since it needs $O(N^4T)$ number of calculations for each updating of the weights. Recently, Toomerian and Barhen modified their adjoint operator approach into an on-line algorithm and claimed that it only needs $O(N^3T)$ number of calculations. However, a careful examination of their scheme revealed some flaw in their derivation and therefore invalidated their claim.

In the past year, we developed an alternative approach in which we tried to avoid the redundant calculations presented in the forward propagation method and use a common Green function to integrate the error sensitivity matrix. We also exploit the special form of the driving term in the equation to reduce the number of calculations. The combined effect is an algorithm that is truly $O(N^3T)$.

VI. Controlling Chaos with Neural Networks

Many of the everyday signal processing problems are temporal in nature. They are the continuous or analog counterpart of the symbolic sequential patterns. The extraction of temporal orders from such continuous temporal signals would found many real world applications in signal processing. As a preliminary study of the neural net capability in this respect, we studied the interesting chaos control problem.

The control of chaos means that to stabilize a chaotic system settling around an unstable fixed point or periodic orbit. The system we chosed is the two dimensional Henon map:

$$X_{t+1} = A - X_t^2 + BY_t$$

$$Y_{t+1} = X_t$$

where the parameters are chosen as $A=1.29$ and $B=0.3$ and is in a typical chaotic regime.

One of the unstable fixed point for this attractor can be found as

$$X_F = Y_F = \frac{1}{2} [B - 1 + \sqrt{(B - 1)^2 + 4A}] = 0.838486$$

Our objective is to construct a neural net controller that can be trained to locate the unstable fixed point automatically and to guide the chaotic system to this point and settled there indefinitely. Our study seems successful. The neural net is easily trained to accomplish this goal with an objective function given by:

$$E_t = \frac{1}{2} [(X_t - \bar{X}_\tau)^2 + (Y_t - \bar{Y}_\tau)^2]$$

It measures the deviation of the orbit from an averaged orbit. In the vicinity of the fixed point, the orbit is in general sticky and therefore contribute heavily to the averaging of orbits. the minization of the objective function therefore requires the system to stay around the fixed point.

We also add noises to the system either in the mapping itself or in the emulating neural net weights. The system turns out to be rather robust against noises. It learned to control the system very smoothly without the appearance of uncontrollable outbursts seen in the original OGY model. It also located the fixed point by itself. Since chaotic system has the characteristics of moving around the whole chaotic attractor in an ergotic fashion, the system is bound to travel by the fixed point location frequently. This makes the controllability of a chaotic system even better than a regular system.

VII. Future Directions:

In the previous research, we have developed knowledge of extracting temporal orders mainly from discrete symbolic sequences. Sequential machines that generate these sequences can be constructed from a few hundred of positive and negative samples each about ten symbols or less. The constructed machine are usually complete and exact capable to generalize to the infinite number of member sequences that belong to the same grammar. The rich phenomena associated with these discrete symbolic sequences should be a subset of what could be described in an analog or continuous sequences. In one sense, a continuous sequence is a discrete sequence with an infinite number of different discrete symbols. It is therefore much more intricate to deal with. However, our preliminary study of such problems indicated that neural net could be easily adapted to the analog situation. Instead of a discrete recurrent neural net that simulate a finite state machine, we should use an infinitesimally incremented constructed so that it simulated a differential or integral system.

An especially interesting realm of research could be directed toward the chaotic system. It is known that very simple system could exhibit very complicated orbit. In a study of automata, it was

pointed out by Wolfram that it can generate sequences with arbitrary complexity. An understanding of controlling these subset of dynamical systems would be the most fruitful endeavor in the neural net research.

Publications

1. Y. D. Liu, G.Z. Sun, H.H. Chen, Y.C. Lee and C.Lee Giles, "Grammatical Inference and Neural Network State Machines", Proceeding IJCNN, Vol. 1, p.285-288, Washington DC (1990).
2. G.Z. Sun, H.H. Chen, C.L.Giles, Y.C. Lee and D. Chen, "Connectionist Pushdown Automata That Learn Context-Free Grammars", Proceeding IJCNN, Vol. 1, P. 577-580 Washington DC (1990)
3. G.Z. Sun, H.H. Chen, Y.C. Lee and C.L.Giles,"Recurrent Neural Networks, Hidden Markov Models and Stochastic Grammars", Proceeding International Joint Conference on Neural Networks, San Diego Vol. I, p. 729, (1990).
4. Y.C. Lee, S. Qian, R.Jones, C. Barnes, G Flake, M. O'Rourke, K. Lee P.Lewis, H.H. Chen, G.Z. Sun, Y.Q. Zhang, D. Chen and C.L.Giles, " Adaptive Stochastic Cellular Automata: Theory", Physica D. (1990).
5. S. Qian, Y.C. Lee, R.Jones, C. Barnes, G. Flake, M. O'Rourke, K. Lee, P. Lewis, H.H. Chen, G.Z. Sun, Y.Q. Zhang, D. Chen, and C.L.Giles," Adaptive Stochastic Cellular Automata: Applications", Physica D, (1990).
6. S. Qian, Y.C.Lee, R. Jones, C. Barnes, and K. Lee, " Function Approximation with an Orthogonal Basis Net", Proceeding IJCNN, San Diego, Vol III, p. 605, (1990).
- 7.R. D. Jones, Y.C.Lee, C. W. Barnes, G. Flake, K. Lee, P.Lewis, and S. Qian, "Function Approximation and Time Series Prediction with Neural Networks", Proceeding IJCNN, San diego, Vol. I, p. 649, (1990).
8. C. Lee 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, Ed. David Touretzky, p.380, (1990).
9. G.Z. Sun, H.H. Chen and Y.C. Lee, "Green Function Method for Fast On-Line Learning Algorithm of Recurrent Neural Networks" Submitted to NIPS meeting (1991).
10. G.Z. Sun, H.H. Chen, Y.C. Lee, "Turing Equivalence of Neural Networks with Second Order Connection Weights" Submitted to NIPS meeting (1991).
11. D. Chen, H.H. Chen, G.Z. Sun, C. L. Giles and Y.C. Lee " Simulating Finite State Automata by Neural Networks" in preparations.

12. Y.D. Liu, Y.C. Lee H.H. Chen and G.Z. Sun 'Nonlinear Resampling Transformation for Automatic Speech Recognition' IEEE Workshop on Neural Networks for Signal Processing, Princeton, Ed. B.H. Juang and S.Y. Kung, p.319, (1991).
13. G.Z. Sun, H.H. Chen and Y.C. Lee, "A Fast On-line Algorithm for Recurrent Neural networks", Proceeding IJCNN, Seattle, Vol. II, p. 13, (1991)
14. D. Chen, H.H.Chen, G.Z. Sun, Y.C. Lee c>L. Giles,and C.B. Miller, "Second Order Recurrent Net for Grammatical Inference" Proceeding IJCNN, Seattle, Vol. II p. 273 (1991).
15. G.Z. Sun, H.H.Chen, Y.C. Lee and C.L.Giles "Turing Equivalence of Neural Networks with Second Order Connection Weights" Proceeding IJCNN, Seattle, Vol. II, p. 357 (1991).
16. C.L.Giles, C.B. Miller, D. Chen, H.H.Chen, G.Z. Sun and Y.C.Lee "Learning and Extracting Finite State Automata with Second Order Recurrent Neural Networks" accepted in Neural Computation, Vol, 4, No. 3, (1991).
17. Y.D. Liu, Y.C. Lee, H. H. Chen and G.Z. Sun, " Discriminative Training Algorithm for Predictive Neural Network Models", Proceeding of IJCNN, Vol. IV, p. 685-690, Baltimore, MD. June, (1992).
18. Y.D. Liu, Y.C. Lee H.H. Chen, and G.Z. Sun, "Speech Recognition using Dynamic Time Warping with Neural Network Trained Templates", Proceedings of IJCNN, Vol. II, P. 326-331., Baltimore, MD. June (1992).
19. S. Das, C.L. Giles and G.Z. Sun, "Learning Context Free Grammars: Capabilities and Limitations of a Neural Network with an External Stack Memory", Proceedings if the Fourth Annual Conference of the Cognitive Science Society, July 29, (1992)
20. D. Chen, C. L. Giles, G.Z. Sun, H.H. Chen, Y.C. Lee and M.W. Goudreau, "Constructive Learning of Recurrent Neural Networks", in Computaional Learning Theory and Natural Learniong Systems III, MIT Press, Ed. T. Petsche and S. Judd and S. Hanson. (1993).
21. S. Das, C. L. Giles and G.Z. Sun, " Using Hints to Successfully Learn Context Free Grammars with a Neural Net Pushdown Automaton", in Advances in Neural Information Processing Systems 5, p.65,(1993).
22. G. Flake, G.Z. Sun Y.C. Lee and H.H. Chen, "Exploiting Chaos to Control the Future", NIPS (1993).