LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

LA-UR -87-3441 CONF-8710189--1

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE IMPROVED INPUT REPRESENTATION FOR ENHANCEMENT OF NEURAL NETWORK PERFORMANCE

LA-UR--87-3441

DE88 001830

AUTHOR(S) C. H. Aldrich
Z. G. An
K. Lee
Y. C. Lee

SUBMITTED TO For publication in Complex Adaptive System Proceeding of workshop at Santa Fe Institute

MASTER

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Improved Input Representation for Enhancement of Neural Network Performance

C. H. ALDRICH, Z. G. AN, K. LEE and Y. C. LEE
(Los Alamos National Laboratory, Los Alamos, NM 87545)

INTRODUCTION

Auto-associate and hetero-associate memory have received major attention as important applications for neural network or "connectionist" computing[1-6]. The potential importance of this application has attracted numerous investigations into optical hardware implementations[7-13]. An important consideration for the implementation of associative memory is the storage capacity of the network. For a Hopfield net, the memory capacity for uncorrelated patterns is approximately $.25N/log(N)$, where N is the number of neurons. In general, the capacity for information storage is proportional to the number of synapses[15]. For fully connected networks the number of synapses scales as N^{m+1} , where N is the number of neurons and m is the order of the network. Higher order networks have a much greater storage capacity than a Hopfield net for an equivalent number of neurons. Simply increasing the number of neurons will not always increase the storage capacity. This will be shown by the simulation results of this paper. Since in most applications, the patterns are coherent with some correlation, the storage capacity is significantly worse than for uncorrelated patterns. The performance of an associate memory network depends significantly on the representation of the data. For example, it has already been recognized that bipolar representation of neurons with -1 and +1 states out- perform neurons with on and off states of +1 and 0 respectively. This paper will show that a simple modification of the pattern vector to have zero bias will provide even more significant increase for the performance of an associative memory network.

The higher order algorithm of Lee et al.[16] is used for the numerical simulation studies of this paper. To the lowest order this algorithm reduces to the Hopfield model for auto-associative memory and the bidirectional associative memory (BAM) of Kosko[6] for hetero-associative memory model respectively.

ALGORITHM

For auto-associative memory, the iterative dynamical equation is:

$$S_{i_1}^{n+1} = W\Big[\sum_{i_2 \ldots i_l} T_{i_1, i_2 \ldots i_l} S_{i_2}^n \ldots S_{i_l}^n\Big]$$

where

$$W[x] = +1 \quad \text{for} \quad x \geq 0$$
$$W[x] = -1 \quad \text{for} \quad x < 0$$

The iterative dynamical equations for hetero-associative memory are:

$$S_{i_1}^{n+1} = W\Big[\sum_{i_2 \ldots i_l, j_1 \ldots j_m} T_{i_1 \ldots i_l j_1 \ldots j_m} S_{i_2}^n \ldots S_{i_l}^n U_{j_1}^n \ldots U_{j_m}^n\Big]$$

$$U_{j_1}^{n+1} = W\Big[\sum_{i_1 \ldots i_l, j_2 \ldots j_m} T_{i_1, i_2 \ldots i_l, j_1, j_2 \ldots j_m} S_{i_1}^n S_{i_2}^n \ldots S_{i_l}^n U_{j_2}^n \ldots U_{j_m}^n\Big]$$

where $T_{i_1, \ldots j_1, \ldots}$ is the memory tensor constructed from the assumption of Hebbian learning. For auto-associative memory the memory tensor is:

$$T_{i_1, i_2 \ldots i_l} = \sum_p \xi_{i_1}^p \xi_{i_2}^p \ldots \xi_{i_l}^p$$

For hetero-associative memory the memory tensor is:

$$T_{i_1,i_2\ldots i_l,j_1\ldots j_m} = \sum_p \xi^p_{i_1}\xi^p_{i_2}\cdots\xi^p_{i_l}\eta^p_{j_1}\cdots\eta^p_{j_m}$$

The summation over p is the sum over all patterns. The most efficient implementation of the iterative dynamical equations depends on the nature of the application. For a learning situation where the system is required to extract features and to generalize, the number of patterns is sufficiently large such that the storage requirement for all the patterns exceeds the storage requirement for the tensor elements. For associative memory applications, on the other hand, to avoid degradation of classification accuracy, the number of patterns stored on the neural network is limited and the storage requirements for all the individual patterns is less than the storage requirement for the memory tensor. In this instance, it is more efficient to implement the iterative dynamic equation in the form of dot products. For hetero- associative memory, the dynamical equations are:

$$S^{n+1}_{i_1} = W[\sum_p \xi^p_{i_1}(\underline{\xi}^p \cdot \underline{S}^n)^{l-1}(\underline{\eta}^p \cdot \underline{U}^n)^m]$$

$$U^{n+1}_{j_1} = W[\sum_p \eta^p_{j_1}(\underline{\xi}^p \cdot \underline{S}^n)^{l}(\underline{\eta}^p \cdot \underline{U}^n)^{m-1}]$$

For auto-associative memory, the dynamical equations reduce to:

$$S^{n+1}_{i_1} = W[\sum_p \xi^p_{i_1}(\underline{\xi}^p \cdot \underline{S}^n)^{l-1}]$$

For the applications of pattern recognition or retrieval, the bipolar representation of patterns usually lead to a bias. As will be shown in this paper, this lead to degraded performance of the network. To remove the bias, the bipolar representation will be transformed into an alpha-beta representation where

$$N_p\beta - N_m\alpha = 0$$

$$N_m\alpha^2 + N_p\beta^2 = N$$

N_m and N_p are the number of neurons for pattern $\underline{\xi}^p$ in the -1 and +1 states respectively. The second equation insures that all pattern vectors are of the same magnitude. The new dynamical equations are:

$$S^{n+1}_{i_1} = W[\sum_p \xi^p_{i_1}(\bar{\underline{\xi}}^p \cdot \bar{\underline{S}}^n)^{l-1}(\bar{\underline{\eta}}^p \cdot \bar{\underline{U}}^n)^m]$$

$$U^{n+1}_{j_1} = W[\sum_p \eta^p_{j_1}(\bar{\underline{\xi}}^p \cdot \bar{\underline{S}}^n)^{l}(\bar{\underline{\eta}}^p \cdot \bar{\underline{U}}^n)^{m-1}]$$

The $\bar{}$ indicates the transformed state. The transformed state replace the -1 and +1 with -alpha and beta respectively. Each pattern has its own corresponding value of alpha and beta, and the alpha and beta values for $\underline{S}$ and $\underline{U}$ are re-computed for each iteration.

SIMULATION RESULTS

These modified algorithms have been implemented on the Sun workstation with a very friendly user interface. Figure(1) is a sample display of the window interface. By using the mouse a user can choose the simulation model, the simulation resolution (8x8, 16x16, or 32x32 pixels) and the order of the simulation. Other features are included for printing, storing, retrieving and deleting patterns to make the use of the interface easier and more intuitive. The particular sample is running the algorithm in the auto-associative memory mode. Using the mouse, the pattern displayed on the upper right corner can be arbitrarily modified turning pixels on and off. Patterns constructed in this way can be added to the network memory which is displayed on the left. Each of these stored patterns can be recalled, modified and either restored to storage or used as an input pattern (perhaps with the addition of "noise" pixels or warped in shape) for the neural net. The pattern on the lower grid is the input interface to the network. For the example shown, each pattern vector has 256 pixels or input neurons (16 x 16). The bipolar representation is active, the letter "Q" was retrieved from storage and placed on the lower grid. The dot product formulation was used with dot product raised to the eighth power. The output for each iteration is also displayed on the lower left grid pattern. For the particular example, the iteration has converged to the alphabet "O" in memory. A ninth order neural network is required to retrieve the letter "Q" without error.

Simulation studies have been conducted with the system described above using three different set of patterns. These three sets are shown in Figure(2). The first two sets of letters have 16x16 resolution or 256 input neurons. The third set has a resolution of 8x8 or 64 neurons. The third pattern set is constructed from the second pattern set where four pixels are mapped onto one pixel. Pixels in the third set are turned on if any one of the four pixels on the corresponding mapping are on. The simulation results are shown in table (I). Only one set is in the neural network memory at a given time. For each set, the order of the network is determined for the perfect retrieval of each patterns with corresponding uncorrupted pattern as input. Performance can be compared for each pattern set in memory and for the two different representations.

The first and fourth column compared the performance of the alpha-beta representation versus the bipolar representation for the perfect retrieval of the first pattern set. Very high order is required for the bipolar representation, order of 7-15, as compared to order of 2-5 for the alpha-beta representation. The simulation clearly showed the problem of bias for the bipolar representation. At network order lower than that given in column one, the network converges to a very strong attractor which turns every pixel off or to -1.

Comparison between the first and second column shows large improvement for the bipolar representation, consistent with the second pattern set being much less biased. The corresponding comparison in the alpha-beta representation is between columns four and five. The performance degraded slightly for the second pattern set due to greater overlap among the patterns.

The best performance for the bipolar representation is for 64 neurons. This contrasts with the conventional wisdom that storage capacity increases with number of input neurons. This rule of thumb is only true for storage of unbiased random patterns. If the stored patterns requires only a small number of neurons, the excess neurons contribute noise and not increased storage capacity. This points to a problem for using neural network to recognize a small object in a large scene. The signal from the small object will be dominated by the background scene. To overcome this problem a network must have the ability to focus attention on the small object. That is local connections on the network has to be enhanced over global connections.

The comparison for the alpha-beta representation for the three sets of pattern shows a similar level of performance. The best performance is achieved with the first set since the overlap among these patterns is the smallest. Recognition system requiring high resolution will always be biased for the bipolar representation. The alpha-beta representation overcomes this problem. Comparison between the two representations has been made in the hetero-associative mode. Figure (3) shows the input patterns and the corresponding output patterns. Figure (4) illustrates that the network successfully retrieved the association of "f" with "F" for a noisy "f" input. As expected the simulation results showed significant increased performance with lower network order for the alpha-beta versus the bipolar representation. The gain in performance is comparable to that acheived for auto-associative memory.

REFERENCES:

(1.) J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in Proc. Nat. Academy Sci., USA, vol. 79, 2554-2558 (1982)

(2.) T. J. Sejnowski and C. R. Rosenberg, Complex System 1, 145, (1987)

(3.) T. Kohonen, Self-organization and Associative Memory, Springer-Verlag, 1984.

(4.) J. A. Anderson, R. M. Golden, and G. L. Murphy, "Concepts in Distributed Systems," Optical and Hybrid Computing, SPIE 634, 260-276, (1986)

(5.) H. H. Chen, Y. C. Lee, T. Maxwell, G. Z. Sun, H. Y. Lee, and L. Giles, "A Higher-order Correlation Model for Associative Memory", in Neural Network for Computing, J.S. Denker, editor, (AIP, 1986)

(6.) B. Kosko, "Constructing and Associative Memory", BYTE 12, 137, (1987)

(7.) N. Farhat, D. Psaltis, A. Prata, and E. Pack, "Optical implementation of the Hopfield Model", Appl. Opt. 24, 1469-1475, (1985)

(8.) D. Psaltis and N. Farhat, "Optical Image Processing Based on an Associative-Memory Model for Neural Nets with Thresholding and Feedback", Opt. Lett. 10, 98, (1985)

(9.) R. A. Athale, H.H. Szu, and C. B. Friedlander, "Optical Implementation of Associative Memory with Controlled Nonlinearity in the Correlation Domain", Opt. Lett. 11, 482, (1986)

(10.) Y. Owechko, G. J. Dunning, E. Marom, and B. H. Soffer, "Holographic Associative Memory with Nonlinearities in the Correlation Domain", Appl. Optics 20, 1900-1910, (1987)

(11.) G. Eichmann and M. Stojancic, "Superresolving Signal and Image Restoration Using a Linear Associative Memory", Appl. Opt. 26, 1911-1918, (1987)

(12.) D. Cassasent and S. A. Liebowitz, "Model-based Knowledge-based Optical Processors", Appl. Opt. 26, 1935-1942, (1987)

(13.) D. Psaltis and J. Hong, "Shift-invariant Optical Associative Memories", Optical Engineering 26, 10-15, (1987)

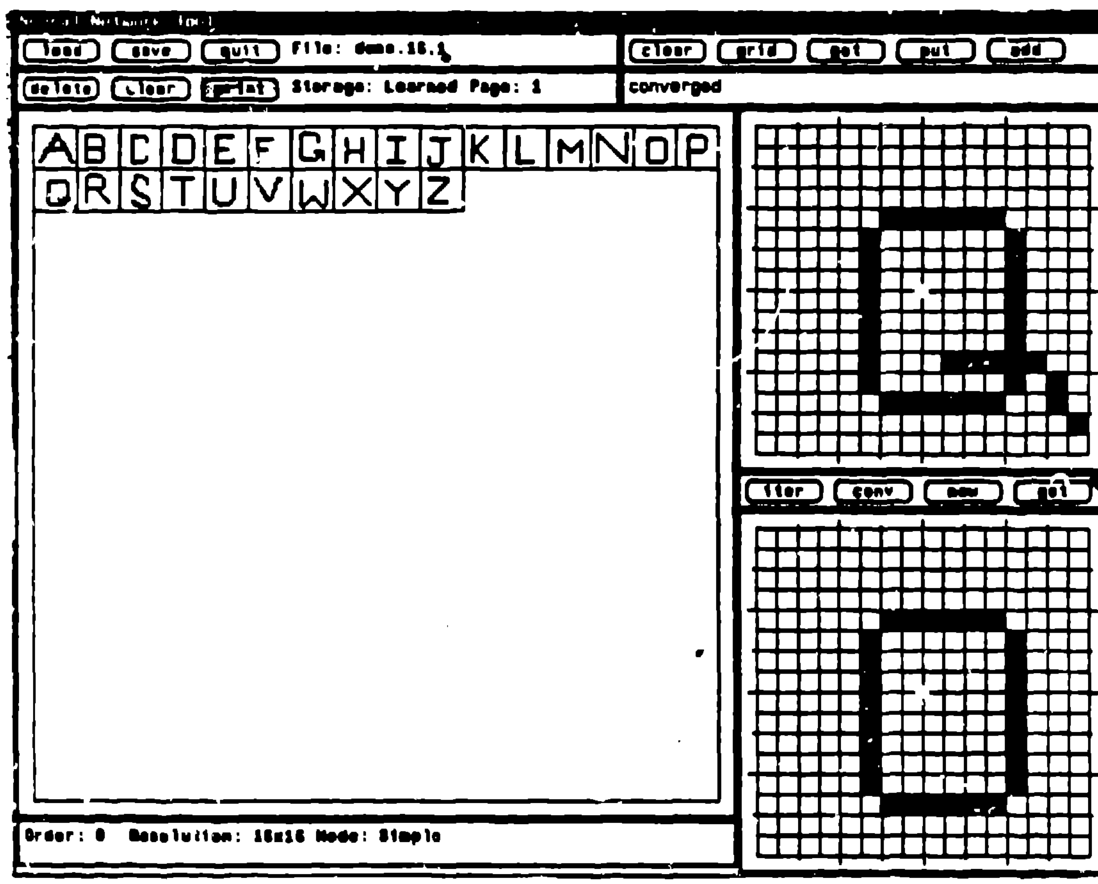
(14.) . D. J. Amit, H. Gutfreund, and H. Somploinsky, Physical Rev. Lett. 55, 1530, (1986)

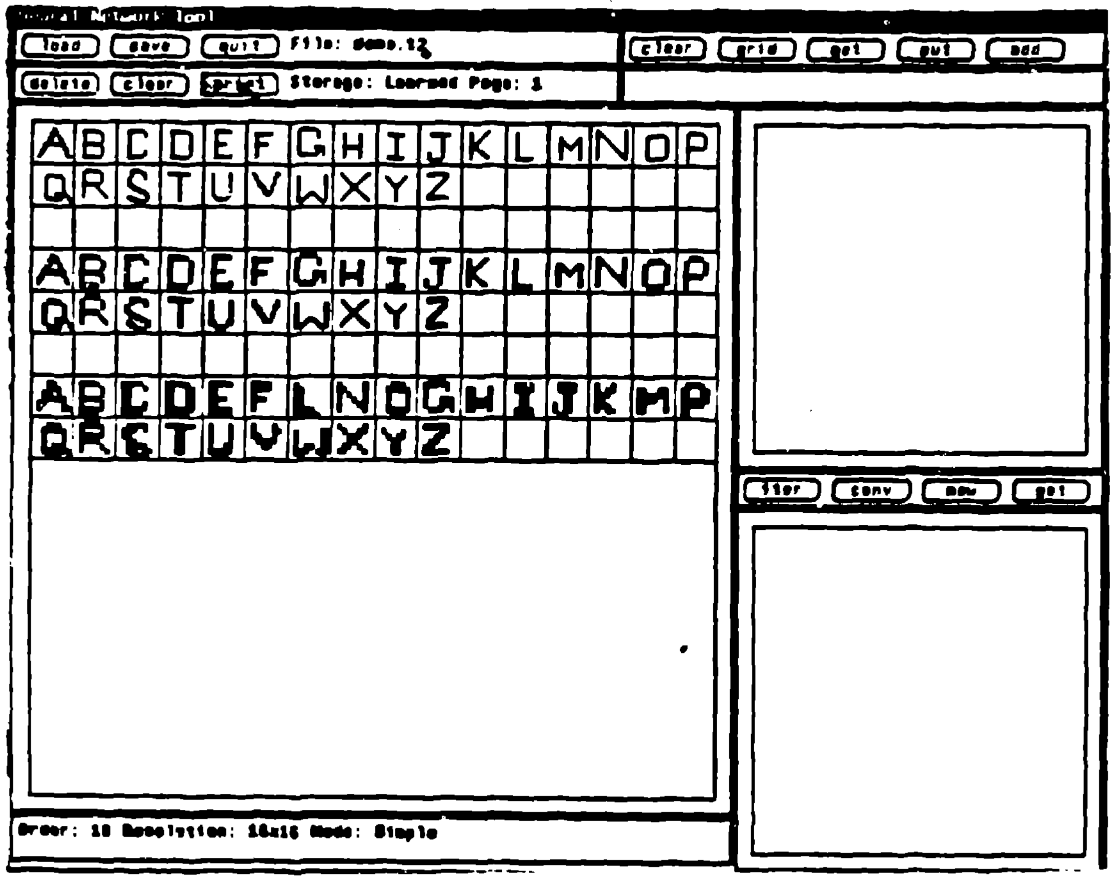
(15.) P. Peretto and J. J. Niez, "Long Term Memory Storage Capacity of Multiconnected Neural Networks", Biological Cybernetics 54, 53-63 (1986)

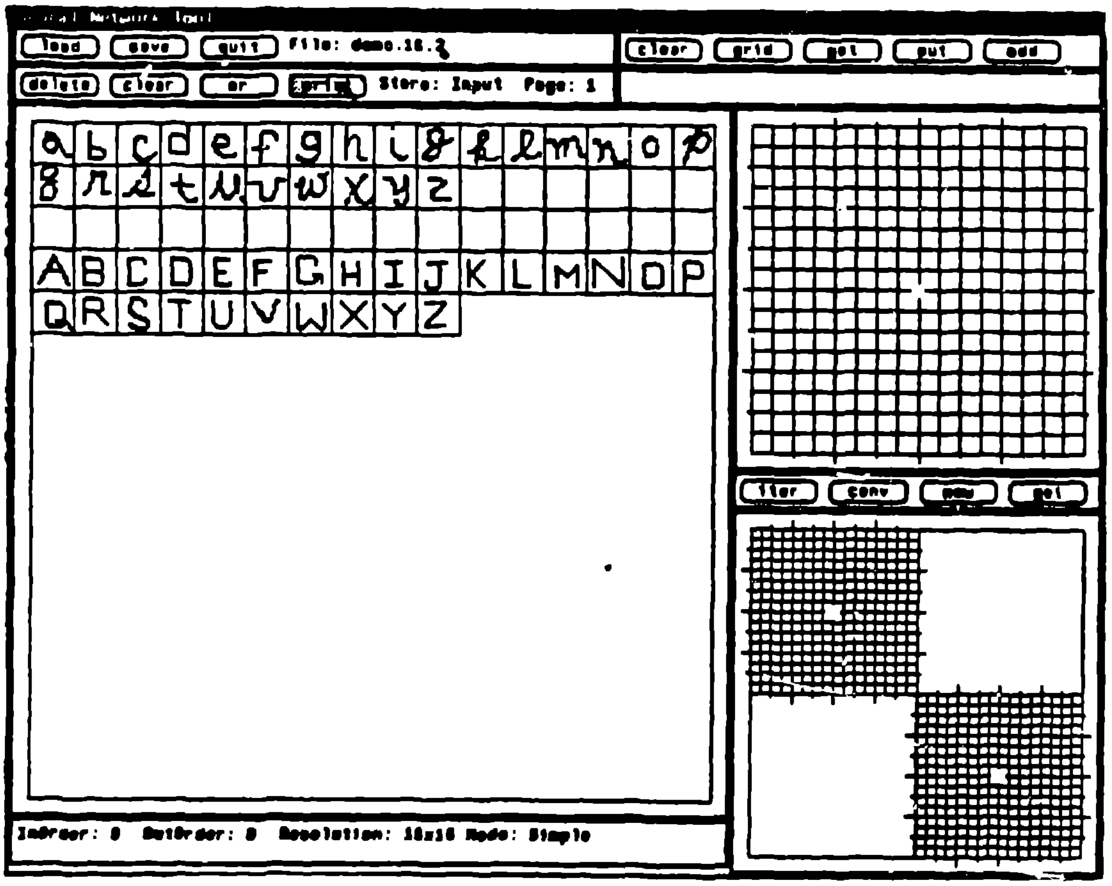
(16.) Y. C. Lee, G. Doolen, H. H. Chen, G. S. Sun, T. Maxwell, H. Lee, and L. Giles, "Machine Learning Using a Higher-order Correlation Network", Physica 22D, 276, (1986)

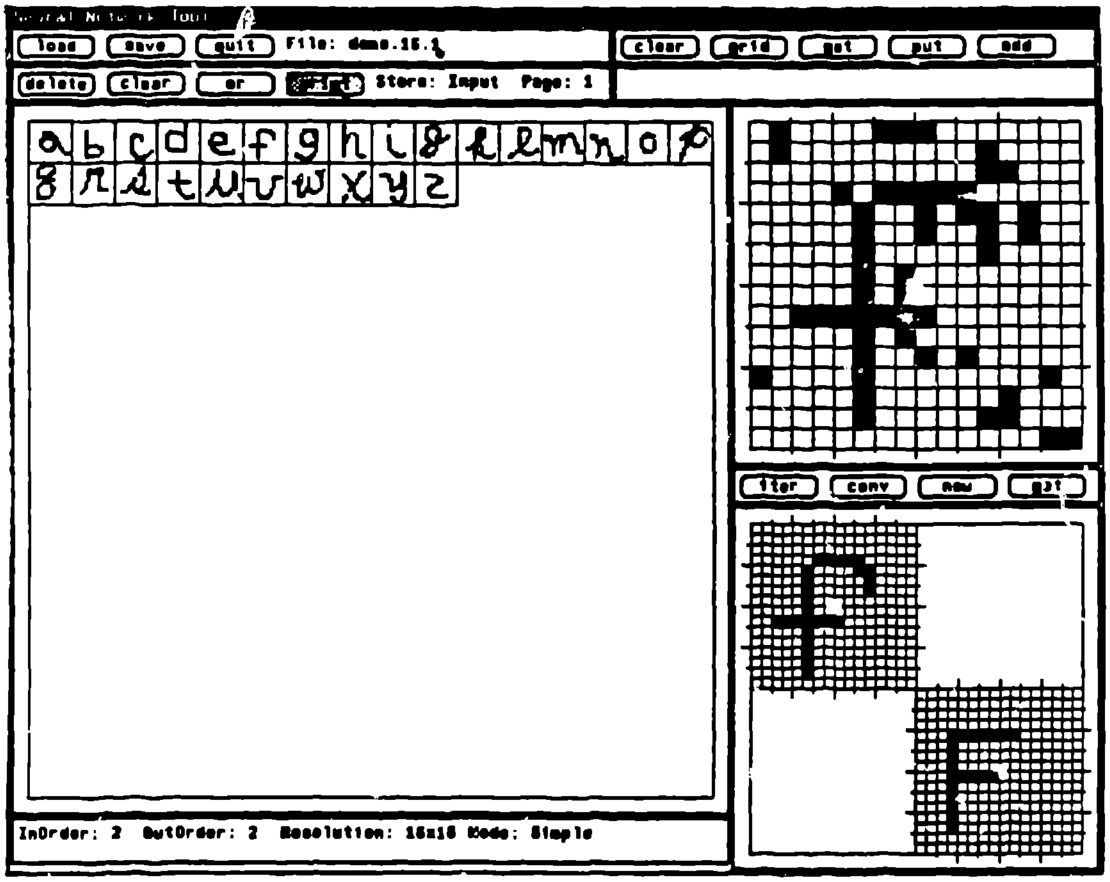
	Bipolar			Alpha-Beta		
Letter	16 × 16	16 × 16	8 × 8	16 × 16	16 × 16	8 × 8
A	7	4	4	2	2	3
B	9	5	4	3	4	4
C	14	7	3	4	5	2
D	13	7	4	5	5	4
E	15	8	4	5	5	4
F	10	6	3	2	3	3
G	9	4	3	3	3	3
H	11	6	3	3	3	2
I	10	6	2	2	3	2
J	9	6	3	2	3	2
K	10	6	3	3	3	3
L	7	6	5	4	4	4
M	10	5	4	2	3	3
N	7	4	3	2	3	3
O	7	5	2	2	3	2
P	11	5	4	3	3	3
Q	9	5	4	3	3	3
R	9	5	4	3	3	3
S	8	5	4	2	3	3
T	11	6	5	2	3	3
U	10	5	4	3	3	4
V	9	5	[illegible]	2	2	2
W	8	4	3	2	2	2
X	9	5	3	2	2	2
Y	11	6	3	2	2	2
Z	10	5	3	2	3	2

TABLE 1. Minimum Network Order


load
save
quit
File: demo.16.1
clear
grid
get
put
add
delete
clear
print
Storage: Learned Page: 1
converged
A B C D E F G H I J K L M N O P
Q R S T U V W X Y Z
iter
conv
new
get
Order: 0 Resolution: 16x16 Mode: Simple


load
save
quit
clear
grid
put
add
delete
clear
Storage: Learned Page: 1
ABCDEFGHIJKLMNOP
QRSTUVWXYZ
ABCDEFGHIJKLMNOP
QRSTUVWXYZ
ABCDEFLNOGHIJKMP
QRSTUVWXYZ
conv
new
Order: 10 Resolution: 16x16 Mode: Simple


load
save
quit
clear
grid
get
put
add
delete
clear
Store: Input Page: 1
conv
Resolution: 16x16 Mode: Simple


load
save
quit
File: demo.16.1
clear
grid
get
put
add
delete
clear
or
Store: Input
Page: 1
conv
new
InOrder: 2 OutOrder: 2 Resolution: 16x16 Mode: Simple