IN-60
43048
P 29

An Empirical Investigation of Sparse Distributed Memory Using Discrete Speech Recognition

Douglas G. Danforth

March 1990

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS Technical Report 90.18

NASA Cooperative Agreement Number NCC 2-408 and NCC 2-387

RIACS

Research Institute for Advanced Computer Science
An Institute of the Universities Space Research Association

AN EMPIRICAL INVESTIGATION OF SPARSE DISTRIBUTED MEMORY USING DISCRETE SPEECH RECOGNITION

Douglas G. Danforth

Research Institute for Advanced Computer Science
Mail Stop 230-5, NASA Ames Research Center
Moffett Field, California 94035
U.S.A.

RIACS Technical Report 90.18

March 1990

ABSTRACT

An experimental investigation of Sparse Distributed Memory (SDM) (Kanerva, 1988) is presented. SDM is an associative memory which can be thought of as a 3-layer Artificial Neural Network. It uses massive parallelism, associates very large patterns, and is trained rapidly. The theory of SDM was developed for uncorrelated bit patterns. In this paper the behavior of SDM is examined when the constraint of random input is violated and the memory is presented with highly-correlated data for classification tasks. Experiments from the domain of discrete-word speech recognition are used. These experiments lead, in a step-by-step manner, to factors which improve the memory's ability to recall and to generalize. It is shown that generalization can be enhanced with appropriate application of: (1) the form of encoding of class labels, (2) the placement of hard locations within the memory, (3) the activation rule of hard locations, and (4) the write rule used to modify the memory. Comparisons are made between SDM, a class-mean model, and the Nearest Neighbor rule. For single-talker digit recognition a form of SDM, called the Selected Coordinate Design, attains 99.3% correct generalization.

Work reported herein was supported in part by Cooperative Agreements NC 2-408 and NCC 2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA).

Table of Contents

AN EMPIRICAL INVESTIGATION OF SPARSE DISTRIBUTED MEMORY USING DISCRETE SPEECH RECOGNITION

Douglas G. Danforth

LEARNING SYSTEMS DIVISION
RESEARCH INSTITUTE FOR ADVANCED COMPUTER SCIENCE (RIACS)
Mail Stop 230-5
NASA Ames Research Center
Moffett Field, California 94035
U.S.A.

1. Introduction

In Pentti Kanerva's lovely book (Kanerva, 1988) a model for associative memory is developed based on the mathematical properties of large binary vector spaces. In that work, a point in the binary space acts as a sensory pattern or as an address to a location within memory. For large input dimensions it is not possible to implement a memory which has a unique hard location for every input pattern but only a sparse subset. The properties of such sparse memories were investigated by Kanerva when the addresses of locations are randomly and uniformly selected from the set of all possible addresses and also when the input patterns to the memory are selected from a uniform random distribution.

The Learning Systems Division of RIACS is involved in exploring the theory and applications of associative memories and Sparse Distributed Memory (SDM) in particular. As such it is important to us to understand the behavior of SDM when presented with non-random data. Non-uniformity is the rule rather than the exception for most phenomena. The elegance and mathematical formulation of SDM affords a solid reference point for this investigation.

This is the first report by the Learning Systems Division on the behavior of SDM in the domain of Automatic Speech Recognition. Previous work has dealt with Text-to-Phoneme generation (Joglekar,1989), issues in visual shape recognition (Olshausen,1988), capacity of the memory (Keeler,1987), as well as others.* Presented here is an incremental step-by-step analysis of how the basic SDM model can be modified to enhance its generalization capabilities for classification tasks. Data is taken from speech generated by a single talker. Experiments are used to investigate the theory of associative memories and the question of generalization from specific instances. A theoretical analysis of these findings will be presented elsewhere.

It is stressed at the outset that this study was not an attempt to build an optimal speech recognizer but rather to investigate SDM. An attempt to maximize absolute score would have entailed maximizing the "front-end" as well as the "back-end" of the recognizer. Little effort was placed on maximizing the quality of the signal produced by the front-end. As you will see, the data for the DIGIT64 database can be discriminated very well.

2. The basic SDM model

I adopt an Artificial Neural Net (ANN) nomenclature to describe the basic SDM model here rather than that of an associative memory in an effort to acquaint a larger body of readers to the SDM technology.

* To obtain an SDM publications list, write to: Attention SDM Publications, RIACS Learning Systems Division, NASA Ames Research Center, MS 230-5, Moffett Field, CA 94035. Telephone (415) 604-4991, Email sdmpubs@riacs.edu.

SDM contains an input, hidden, and output layer. Each node in the input layer is binary valued (0 or 1). A binary input pattern is thought of as an *address* into the memory. Weights between the input and hidden layer are fixed at time of creation and are also binary valued. The binary weights leading into a hidden node are an address called the location of the node. They are chosen randomly from the set of all possible 2^{n_1} locations where n_1 is the number of nodes in the input layer. There are n_2 locations (hidden nodes). The weights from a hidden node to the n_3 output nodes are variable and are implemented as counters which are incremented or decremented by 1 during training.

Operation of the memory entails activation, reading, and writing. Activation of a hidden node occurs when an input pattern is within a specified Hamming distance of the node's location. A node fires or it doesn't. This all-or-nothing rule transforms an input pattern of size n_1 into an a far larger activation pattern of size n_2 (up to 1/2 million with some experiments on a connection machine (Rogers,1989)). The activation pattern encodes, as single bits, higher order interdependencies between bits of the first pattern.

Training memory to respond (writing) with a desired binary output pattern of size n_3 entails incrementing the corresponding counter of each activated node if the desired output bit is 1 and decrementing if the desired output bit is 0.

Testing (reading from) memory entails pooling the counters for each bit of the activated nodes and then quantizing the resultant sums about 0 to produce a binary output pattern (the basic model includes a global threshold to accommodate global bias).

This concludes the description of the basic SDM model.

3. Speech processing

A SUN-386i workstation with an Ariel Corporation DSP-16 board (Texas Instruments TMS320C25 signal processor running at 40 MHz) was used for SDM simulation and speech processing. The DSP-16 can sample up to 50 KHz with a resolution of 16 bits. All results presented in this report used 10 KHz sampling. A Realistic 33-2001 Dynamic Mike was used for input. Recording was done in a single-user office environment (computer cooling fans audible).

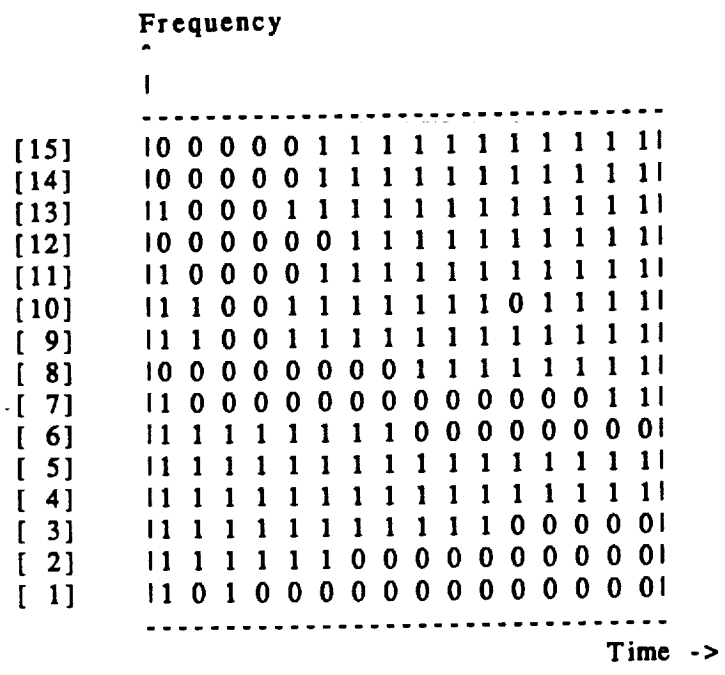


Figure 1: Typical speech pattern

Front-end processing used 16 bandpass filters distributed on a Mel frequency scale which were low-pass filtered and sampled every 6.4 msecs. Amplitude differences between consecutive frequencies were calculated and quantized to 1-bit. A simple energy-based end-point detection scheme was used and the utterance cut into 16 equalize-size segments. Majority rule was used to assign a single bit to each segment at each frequency difference. This yielded a total of 240 (15*16) bits. The pattern was padded with 16 additional bits to investigate SDM's ability to perform auto-associative recall. The resultant pattern contained a total of 256 bits.

A database, DIGIT64, was collected from a single talker which consisted of 64 repetitions of the digits {0-9}.

4. Accretive testing and training

An accretive (slowly growning) training strategy was used in all experiments. An utterance (256-bit pattern) was presented to the memory for recognition and its correct or incorrect response recorded. That same utterance was then used to train the memory and a new utterance was selected for testing. This process was repeated to the end of the database. Each utterance was thus used once for testing and once for training (in that order). This sequential presentation process allows one to construct a curve showing the rapidity of learning. This sequential adjustment also corresponds to the way an organism must deal with its environment. An organism can not afford the luxury of analyzing a large body of data before it makes its decision on how to respond. These arguments also apply to real-time speech recognition systems that must adapt to their current talker and noise level.

The percentage correct results reported in this paper refer to the average of the last half of the learning curve.

Total time for activation, reading, and writing of a single utterance is about 1.1 seconds when programmed in the MAINSAIL language running on a SUN-386i for an SDM memory with 1,024 hidden nodes.

5. Nearest neighbor and quantized mean models

Two bench-mark models were used for comparison with SDM. Both models were accretively tested and trained. The first was the Nearest Neighbor (NN) rule (Cover,1967). Cover showed that for continuous spaces the decision to classify an unknown point of the space as that of the label belonging to the nearest observation point of a sample asymptotically approaches at most twice the Bayes error as the sample size goes to infinity. For discrete spaces, the error rate for the nearest neighbor rule asymptotically equals the Bayes rule. Figure 2 shows the error curve for the 1 nearest neighbor (1-NN) rule. All nearest neighbor rules which based their decision on more than just the single nearest neighbor (k-NN) had worse performance.

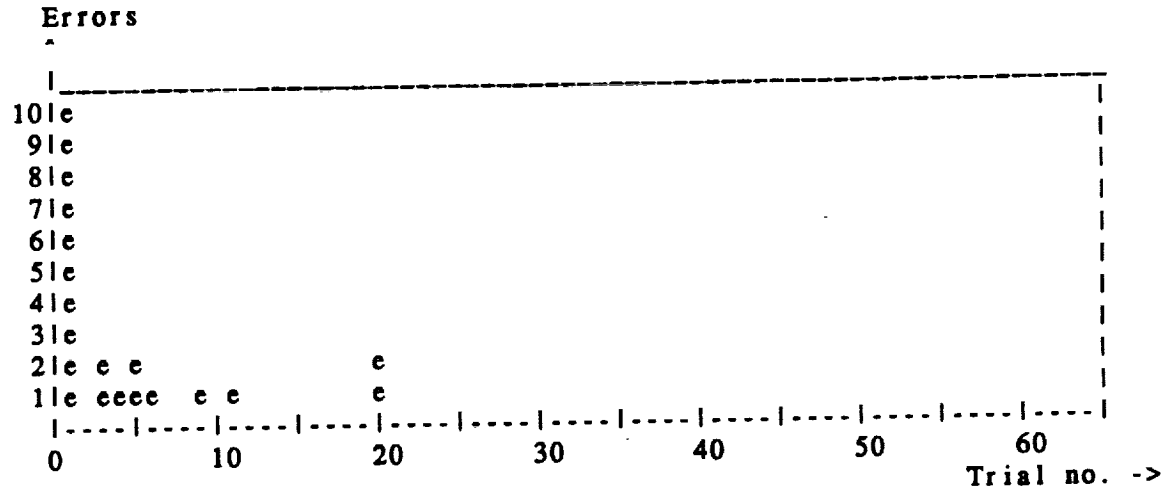


Figure 2: 1-Nearest Neighbor model
DIGIT64 database
n1=256, 0<n2<640, n3=16, k=1, pcc = 100.0%

As one can see, the 1-NN rule made no errors in the last half (320 utterances) of the DIGIT64 database. This shows that the front-end encoding scheme was sufficient to allow excellent discrimination for the digit vocabulary. Separability of the data was possible.*

The second model used for comparison was a quantized Mean model where the (running) average pattern for each digit was calculated and each element quantized to 1 bit. This is a very parsimonious model that needs only one structure for each of the 10 digits. The Nearest Neighbor model, in contrast, stores one structure for each observation (640 for the DIGIT64 database).

The model gives an indication of how much discriminatory information is contained within the class means. The model also approximately corresponds to a 2-layer Artificial Neural Network that has 256 input nodes and 10 output nodes. Comparison is made not between the input and the weights but between input and the quantized weights to determine which output node is most strongly activated. Modification of the weights simply entails adding (with +1 and -1 values) the input pattern to the previous weights for a given class (output node).

* It should be mentioned that an SDM, repeated cycled through the data, was able to learn this data set without error, however, such a technique says little about its ability to generalize, which is the main thrust of this work.

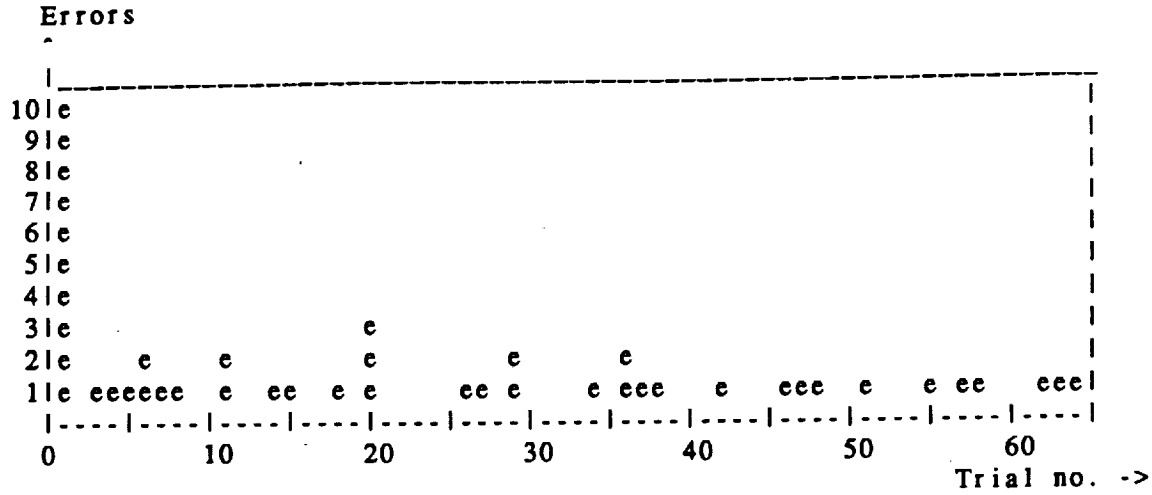


Figure 3: Quantized Mean model
DIGIT64 database
n1=256, n2=10, n3=256, k=1, pcc = 95.0%

It can be seen from figure 3 that the quantized Mean model had a 95.0% recognition rate over the last half of the data set indicating that the class means of the data contain a substantial proportion of the information needed to discriminate between the classes. The term "pcc" in the caption of the figures stands for percent correctly classified (which can be considered an estimate of the probability of correct classification).

6. Area addressing

The basic SDM model uses a fixed radius to determine activation. In this work the closest k locations to an input pattern determined the activation set. The size of this "area" (number of nodes activated) was varied from experiment to experiment and is specified by the variable "k" in the figure captions. To implement area addressing in an ANN it would be necessary to have competitive inhibition between hidden nodes. Area addressing guarantees that k locations always will be activated. It has the effect of dynamically increasing or decreasing the size of the activation radius depending upon the region of the input space considered.

7. Experiment D1 - Integer output codes

The first SDM experiment was constructed to test the significance of the encoding scheme for word labels. The output class labels were coded as 16 bit integers with numeric value equal to digit+1. In base 2, digit "zero" was coded as 0..01, "one" as 0..10, "two" as 0..11, etc (code 0..00 was used for "don't know" and was the default for an initially zero memory). The memory had 1024 hard locations which were randomly chosen. The recognition rate was 49.6%.

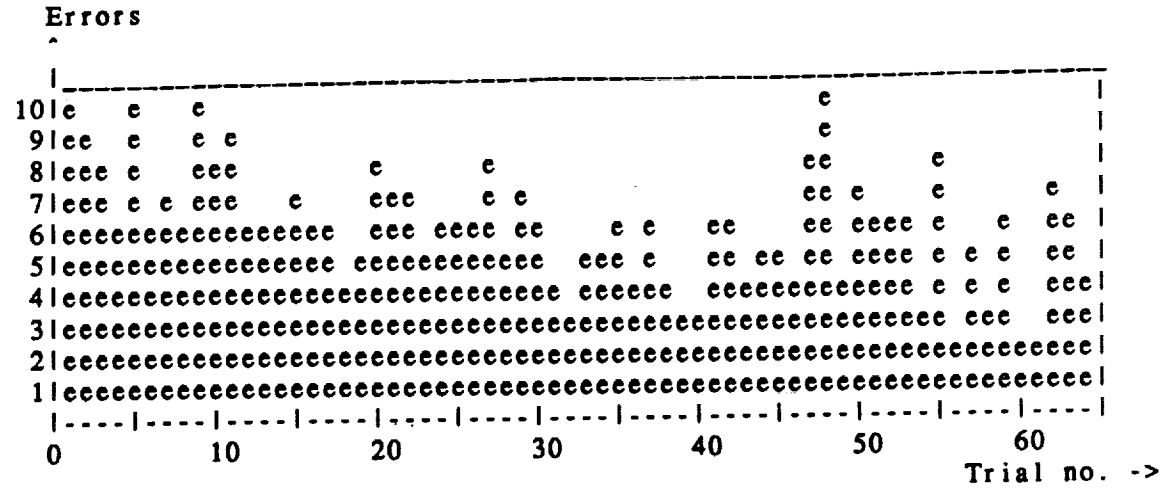


Figure 4: Integer encoding of class names
DIGIT64 database, random hard locations
n1=256, n2=1024, n3=16, k=23, pcc = 49.6%

8. Experiment D2 - Hadamard output codes

The encoding scheme was then changed to use Hadamard bit codes (Harwit,1979) which form an orthogonal basis set. That is, for 16 bits there are 16 basis vectors each of which are exactly 8 bits away in Hamming metric from each other. The encoding rule for the ith component of the kth vector, x, in N dimensions that was used was:

$$x_{k,i} = -1^{B_k \cdot B_i}$$

where B_k and B_i are the bit representations (0,1) of k and i taken as N-bit bit vectors and $B_k \cdot B_i$ is their inner product. If the inner product is *even* then the element of x is +1 otherwise it is -1 (I was lead to this encoding scheme by a theoretical analysis of the "inverse" of the basic SDM model which will be reported elsewhere). Each digit class was assigned a basis vector from this set. On reading from memory the 0-thresholded output sums were compared with each class vector and the closest was deemed the winner of the recognition.

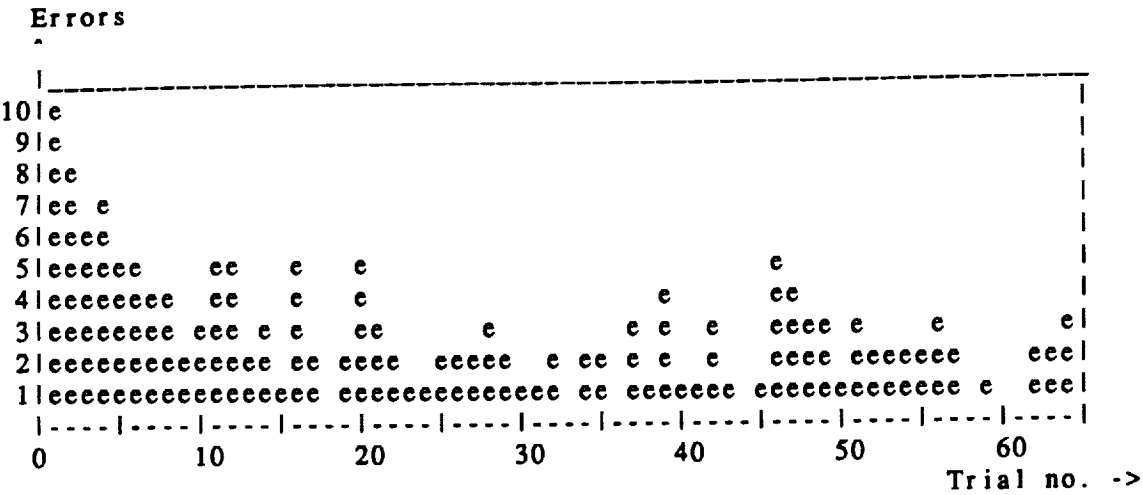


Figure 5: Hadamard encoding of class names
DIGIT64 database, random hard locations
n1=256, n2=1024, n3=16, k=23, pcc = 81.2%

The recognition rate with the Hadamard encoding rose to 81.2% showing that the form of class label encoding is important.

It should be noted that Hadamard labeling combined with winner-take-all effectively implements an $n_3/4$ error correcting code (out of 16 bits 4 can be corrected).

9. Memory efficiency

A measure of the effectiveness of memory is how often a memory location is activated. Locations that are never activated serve no purpose (future data may activate these locations and so one must not enforce total memory efficiency). Figure 6 depicts the fraction of hard locations activated equal to or less than the specified number of activations.

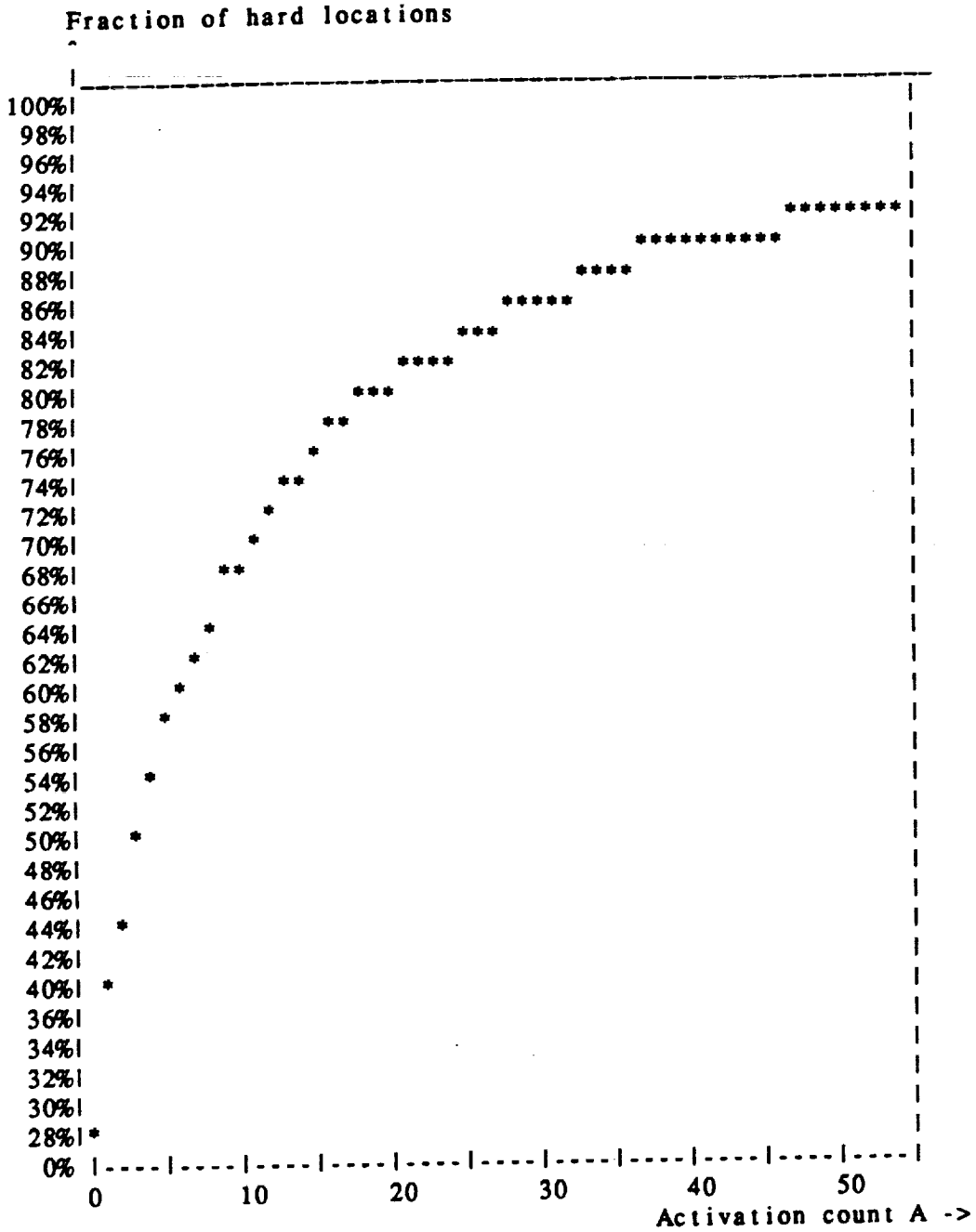


Figure 6: Fraction of hard locations activated less than or equal to A
DIGIT64 database, random hard locations
n1=256, n2=1024, n3=16, k=23

It was found that 28.4% of the memory was unused (0 activation), 50% of the memory was activated 16 or less times, and the most frequently activated location occurred 307 times (out of 640 possible). The distribution of activations changes with the activation rule (see for example the Selected Coordinate Design of experiment D6).

10. Experiment D3 - Dependence on hidden layer size

To determine the degree of dependence of the error rate on the number of hidden nodes in an SDM an experiment was constructed that increased the number of hard locations from 10 up to 20,000 in quasi-logarithmic steps. The results are depicted in figure 7.

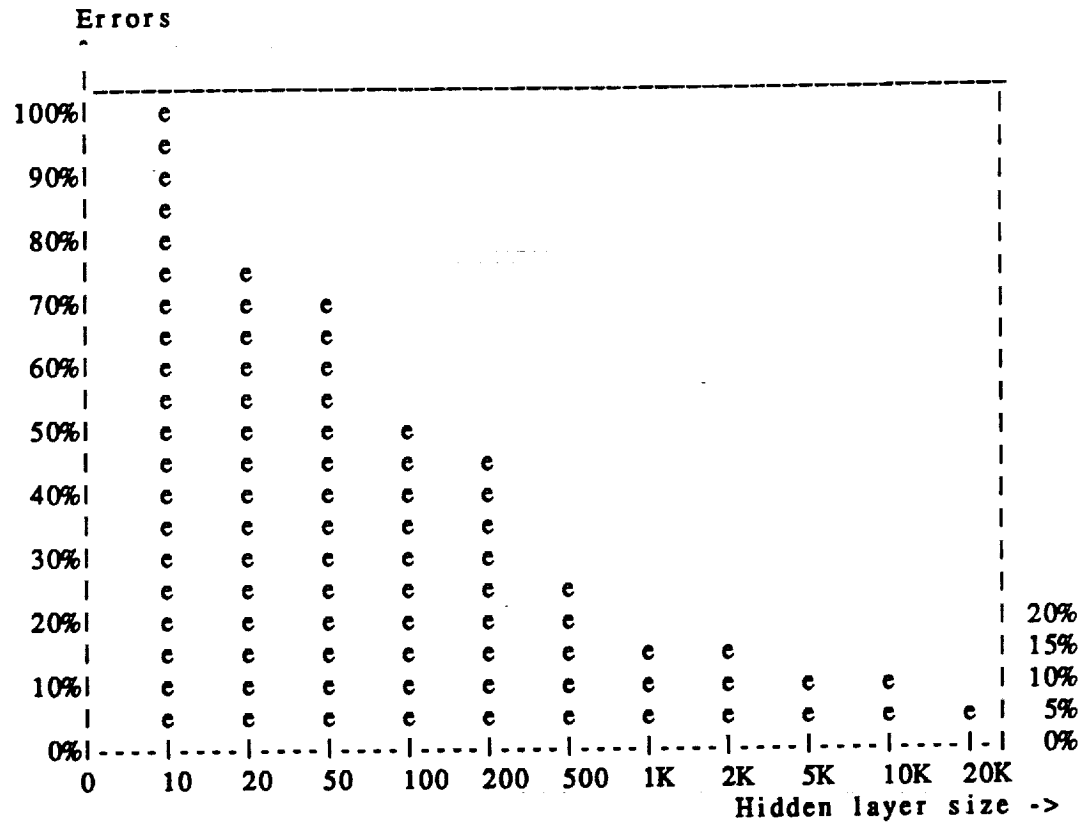


Figure 7: Error graph as a function of hidden layer size
DIGIT64 database, random hard locations
n1=256, n2=variable, n3=16, k=16

The error rate at 20K is 5.2% (94.8% correct) and is decreasing very slowly. This error rate is essentially the same as that obtained by the quantized mean model that used only 10 nodes. The interpretation of these results is given in the next section.

11. Indifference distance and correlated data

One property of uniformly random binary vectors used by Kanerva is "indifference". That is, out of n bits most vectors differ from each other by about n/2 bits. For large n this indifference becomes very pronounced. About any vector almost no other randomly chosen vector is very close to it. If the locations in the memory and the patterns presented to the memory both follow a random uniform distribution then the separation between pattern-pattern, pattern-location, and location-location will all have the same distribution and locations will *cover* the input pattern space.

When uniformity of input is violated there is the possibility that many patterns will fall between locations. The resolution of a uniform distributed memory may not be sufficient to resolve variations in highly compacted or correlated input patterns.

Examination of the *within-class* Hamming distance of the DIGIT64 database revealed that the average

separation of patterns was about 40 bits (see Appendix B). Patterns chosen randomly would have had a mean of 128 bits so the 40 bit separation was actually 11 standard deviations away from random. The probability that 2 points chosen at random would fall within 40 bits is less than 10^{-27}. For an input pattern to fall with high probability within 40 bits of any hard location it would be necessary to have at least 10^{27} locations.

The previous arguments plus the experimental results leads one to the understanding that a *redistribution* of locations within the memory is necessary to obtain good recognition (once the input distribution is approximately modeled then including more locations will enhance the density of coverage).

The condition upon which Kanerva derived his basic SDM model of uniform input and uniform memory locations can now be generalized to:

DISTRIBUTION PRINCIPLE

Place the available memory locations in accord with the distribution of the input patterns.

This principle was not specified by Kanerva in his book but has been present in his thinking (personal communication) and discussions over the years. Jim Keeler (1988) first mentioned this principle in writing in the context of the capacity of SDM in comparison with the Hopfield-type Neural Networks.

12. Experiment D4 - Hidden node distributions

In order to test the distribution principle a collection of 850 templates were generated by reading from the introductory portion of a paper on speech recognition (Baker,1984). The words modeled speech (but not specifically digits). These templates were used as the hard locations of SDM (n_2=850). The system was accretively tested and trained using the DIGIT64 database which resulted in a recognition score of 92.1%.

These results show that a speech-distribution of 850 locations was able to exceeded the recognition accuracy of 10,000 randomly chosen ones supporting the distribution principle.

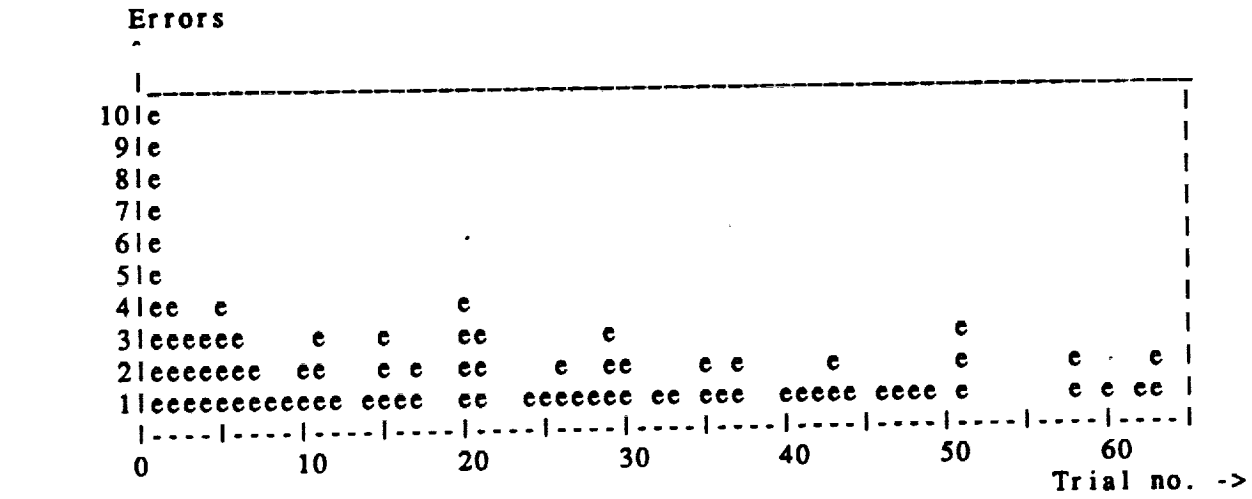


Figure 8: Hard locations chosen from speech DIGIT64 database, correlated hard locations n1=256, n2=850, n3=16, k=6, thresholded, pcc = 92.1%

That preconditioning the memory with speech-like sounds aided in the ability of the memory to learn the digit's sound:name associations is suggestive of processes used in human language learning. A child is immersed in language for many months after birth before object:name associations are reliably formed. Could it be that part of this process entails modifying the child's memory by laying down a covering over auditory-pattern experience on which associations are to built?

13. Thresholding the output layer

With orthogonal codes, such as Hadamard, the counter values of a location maintain the frequency count of each class that activated the location. That is, the counter values are the linear sum of orthogonal codes. That they are orthogonal means that the weighting (frequency of occurrence) of each code can be recovered from the counters simply by taking the inner product of a code with the counters. The value of the inner product is the frequency times n_3 (the number of bits needed to represent the code which is the dimensionality of the output layer).

This is also true when a read is performed and all activated hard locations add their counters to a global "sums" register. The inner product of each code with the global sums reveals a frequency count of each code. These frequencies estimate the conditional class probabilities given the input pattern. If these frequencies are the true conditional class probabilities then the Bayes rule, that would minimize the error of classification, is to choose the class with the maximum probability. One can approximate this rule by choosing the class with the maximum frequency.

If one thresholds the global sums before performing the inner product then the frequency of each class is distorted and a simple maximum frequency argument no longer applies.

Experiments indicate that one gets slightly better results if one does not threshold the output sums. Experiment D4 was run a second time without thresholding which yielded a performance of 94.6% correct (effectively equaling a memory with 20,000 random-address locations).

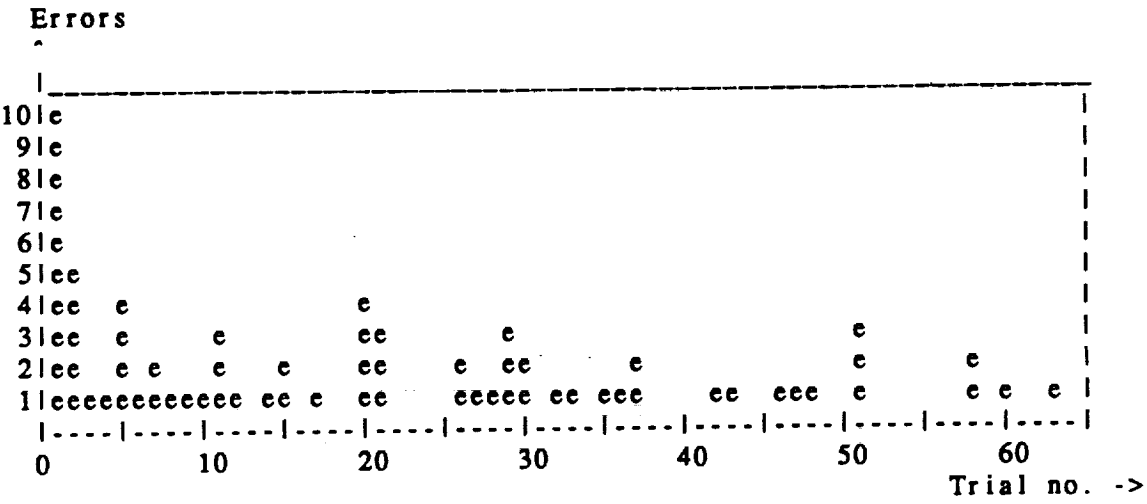


Figure 9: Hard locations chosen from speech, no thresholding
DIGIT64 database, correlated hard locations
n1=256, n2=850, n3=16, k=6, pcc = 94.6%

14. Experiment D5 - Incremental address adjustment

Since the placement of location addresses made a substantial difference another experiment was performed that slightly modified the hard locations of experiment D4. The address of a location was changed slightly by moving toward the input pattern (X) which activated it. The rate of change was

governed by the parameter ρ (rate of learning) which could vary between 0 and 1. The memory was run in auto-associative mode. Counters (C) were represented in floating point and were quantized to produce a new modified address (A). Quantization is indicated below by square brackets "[]" such that [positive]=1, [non-positive]=0. For vectors quantization occurs on an element by element basis. If a location with address A is activated by X then:

$$C := C + \rho(X - C) \quad (\textit{new counters})$$

$$A := [C] \quad (\textit{new address})$$

Adjustment occurred during every write operation. For ρ =.3, recognition rose to 97.5%.

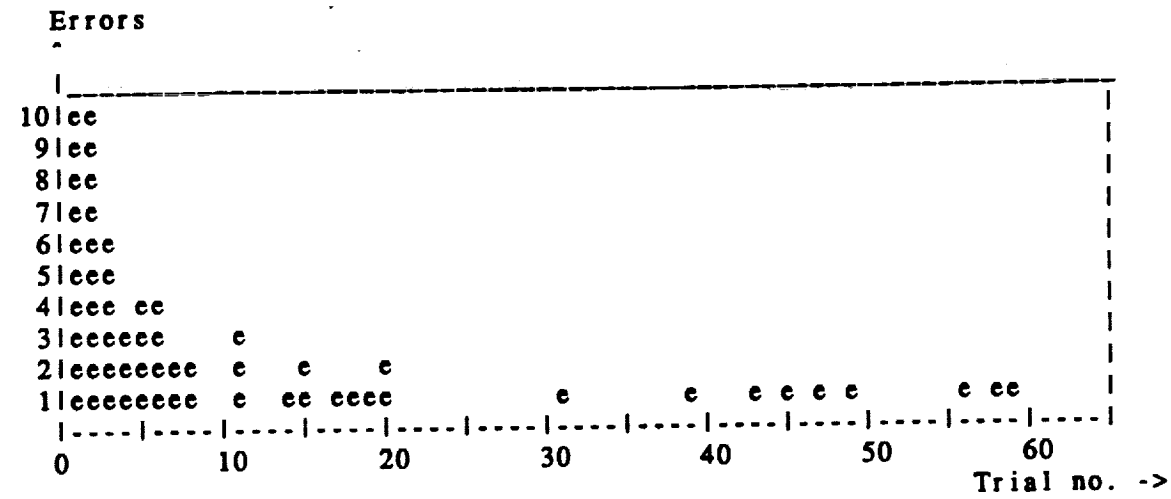


Figure 10: Incremental address adjustment
DIGIT64 database
n1=256, n2=850, n3=256, k=6, ρ=.3, pcc = 97.5%

Starting with speech-like addresses for locations and then modifying them slightly to correspond more closely to the actual input digits produced an associative memory that exceeded the simple quantized mean model (2-layer network).* This shows that further information can be extracted from the data in excess of that obtained by the mean model by distributing, in the input space, more than one location for each class.

A slight variation on this incremental address adjustment model was also tried where the fraction of motion was governed by the clearity of the signal present at the read address (global sums) verses the clearity of the signal stored within the counters of an activated location. Clearity was defined as the sum of the absolute value of the vector elements. Comparable results to those reported here were obtained.

* One might expect this since repositioning locations to correspond more closely to the input patterns has the flavor of a Nearest Neighbor rule.

15. Experiment D6 - The Selected Coordinate Design

In the previous experiments, each memory location examined all 256 bits of the input pattern to make its decision to activate or not. An alternative model, the Selected Coordinate Design,* has been put forth by Louis Jaeckel (1989) in which hard locations examine only a small subset of the input bits. These "selected coordinates" are chosen at random and have random bit values. ** A location is activated if an input pattern exactly matches the selected coordinates.

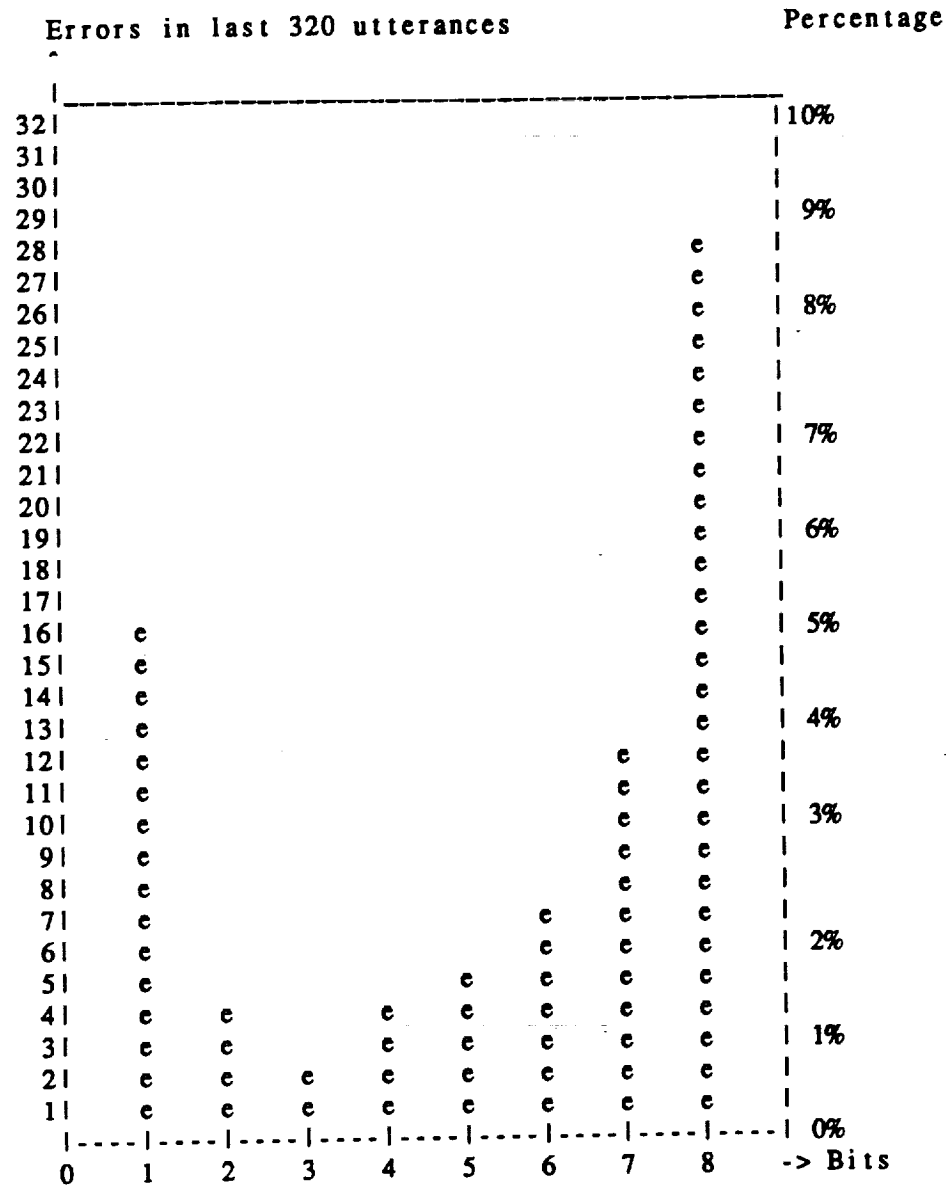


Figure 11: Errors as a function of the number of selected coordinates
DIGIT64 database, no thresholding, polarity learning
n1=256, n2=2048, n3=16, k=variable

* Patent pending

** The cerebellum of the brain seems to follow a similar design where the number of connections between a granule cell and the mossy fiber inputs are few, ranging from 4 to 6 (Marr,1969).

The Selected Coordinate Design (SCD) was coupled with a technique introduced by Prager (1989a,1989b). This technique, which I call the polarity rule, adjusts counter values depending upon whether the polarity of a bit-sum, constructed during a read, has the correct sign or not. If the polarity is wrong then the bit-counter of all activated nodes is adjusted just enough so that a future read with the same input pattern will give the correct sign for that bit.

The number of bits, b, in an SCD is a free parameter of the model. To explore the dependence of classification on it, a series of experiments were run where b was varied from 1 to 8. The results are depicted in figure 11.

The first thing one notices is the pronounced minimum at 3 selected coordinates. That so few bits could perform so well came as a revelation (it is believed at this time that the data complexity will determine the number of bits needed and that 3 does not have any universal significants). The 2 errors at this value correspond to a 99.3% correct recognition rate. Figure 12 shows the error curve for this Selected Coordinate Design of 3 bits.

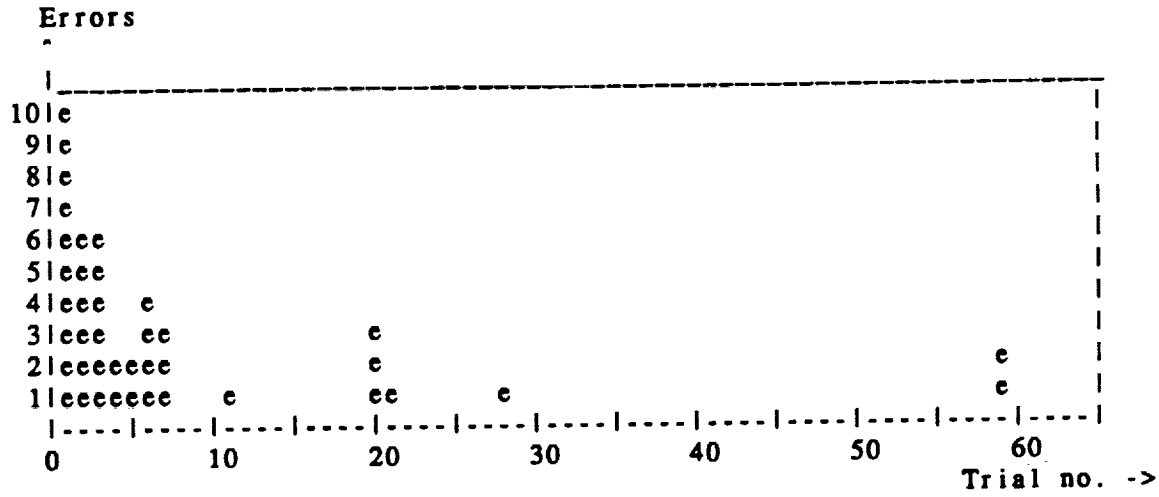


Figure 12: Selected Coordinate Design
DIGIT64 database, no thresholding, polarity learning
n1=256, n2=2048, n3=16, k=variable, b=3, pcc = 99.3%

One can get a sense of the efficiency of a Selected Coordinate Design by examining the fraction of unused memory locations as a function of the number of selected coordinates. This information is presented in figure 13.

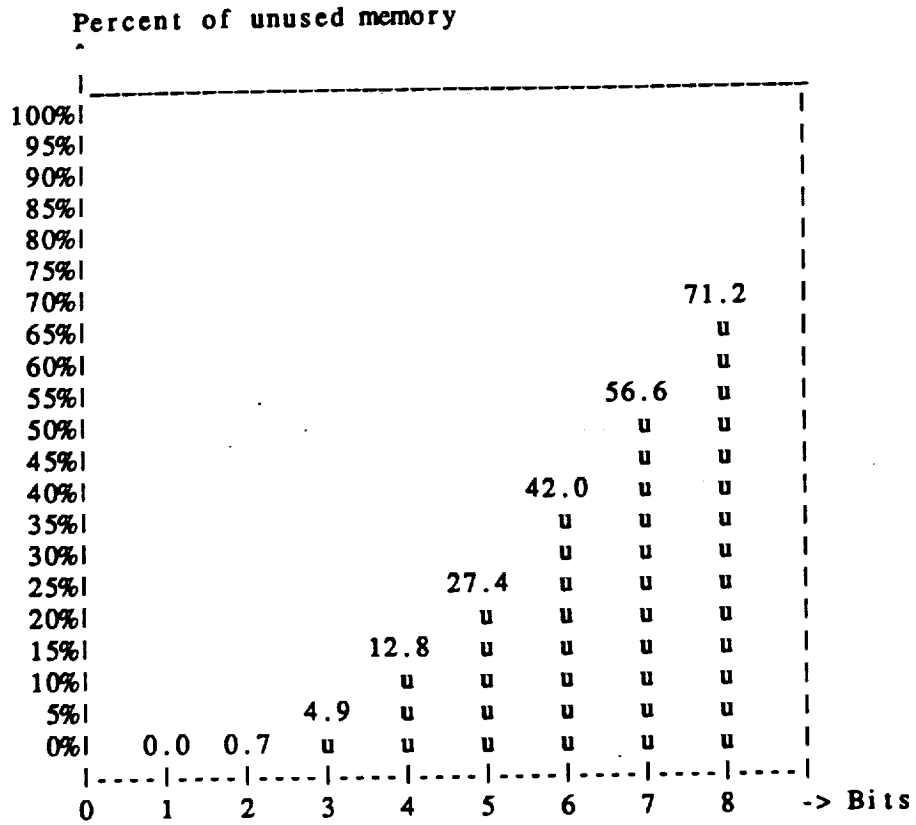


Figure 13: Unused memory verses number of selected coordinates
DIGIT64 database, no thresholding, polarity learning
n1=256, n2=2048, n3=16, k=variable

It would be expected that the unused portion of memory would grow exponentially as a function of the number of selected coordinates if the input data were uniformly random. This expectation follows from the fact that each additional selected coordinate decreases by a factor of 2 the probability of an address being activated.

It can be seen from figure 13 that, although the curve is increasing, it is not exponential. In fact the rate of increase is linear in the range from 4 to 8 bits (14.6% per bit).

One might ask whether increasing the amount of memory, to compensate for the loss of efficiency, while increasing the number of selected coordinates would yield better recognition results. Errors in the early part of the learning curve are indeed reduced by doing this, however, the asymptotic results do not seem to improve (using 16,384 hard locations and 8 selected coordinates an SCD made 4 errors in the last half of the learning curve).

A second amazing fact that can be extracted from figure 12 is that a single bit (selected coordinate) was as good at generalization as the quantized mean model (5% error). This fact is not a coincidence since (for equal class probabilities) the class mean (unquantized) is equivalent to a 1-bit Selected Coordinate Design. This fact was realized only after these experiments were run. It will not be shown here but the following correspondence holds:

2–layer ANN <=> class means <=> 1–bit SCD

where <=> means "is equivalent to". It seems that a b-bit Selected Coordinate Design affectively approximates the input probability distribution as a superposition of bth order marginals of the distribution.

Experiments have also been carried out where a distribution in the number of selected coordinates in the memory have been implemented. When the *average* number of selected coordinates is b the recognition rate is essentially the same as that for an SCD which has only b selected coordinates.

The behavior of the Selected Coordinate Design coupled with polarity learning is much more robust than any of the other forms of Sparse Distributed Memory tested. It consistently performs well. One may ask why this should be so when the amount of information available to each hard location is actually far less than that in the basic SDM model.

A tentative explanation is that the uniform treatment of bits by the Hamming distance used in the basic SDM model does not capitalize on local interdependencies. A hard location treats all patterns within its vicinity equally, independent of where bit errors occur, attending only to the number of errors. In contrast, a hard location in the Selected Coordinate Design attends to *which* bits are in error for patterns within its vicinity.

If one considers the speech patterns in Appendix A, the non-attendance to where errors occur would be equivalent to allowing these errors to be moved around at random from one part of a pattern to another part. For speech, this treats high or low frequency errors or front or back errors as equivalent, which they are not. So, although the SCD model attends to less information within a pattern, it is potentially more discriminatory.

16. Summary

A Sparse Distributed Memory was constructed that used discrete-word speech patterns as input and word labels as output. The properties of the memory were investigated for varying label codes, hard location distributions, and activation rules. By adopting the best of each of the techniques it was possible to improve the recognition score on single-talker digit-recognition from 49.6% to 99.3%.

	D1	D2	D3	D4	D5	D6
SDM Model						
Basic	X	X	X	X	X	
SCD(1)						X
Addresses						
Random	X	X	X			(2)
Speech				X		
Adjusted					X	
Layer sizes						
n1	256	256	256	256	256	256
n2	1024	1024	(3)	850	850	2048
n3	16	16	16	16	16	16
Area size	23	23	16	6	6	(4)
Output encoding						
Integer	X					
Hadamard		X	X	X	X	X
Output thresholded						
Yes	X	X	X	X	X	
No				X		X
Training rule						
Inc/Dec	X	X	X	X	X	
Polarity						X
Generalization rate	49.6%	81.2%	94.8%	94.6%	97.5%	99.3%

Table 1: Summary of parameter settings used in experiments

(1) Selected coordinate design.
(2) 3 bits randomly selected, value randomly chosen.
(3) Number of hard locations varied from 10 to 20,000.
(4) Variable number of locations selected.

It appears crucial that the weights in the first to second layer of the memory be distributed in accord with the metric-determined distribution of the input data and that, for classification, the labels be as independent as possible.

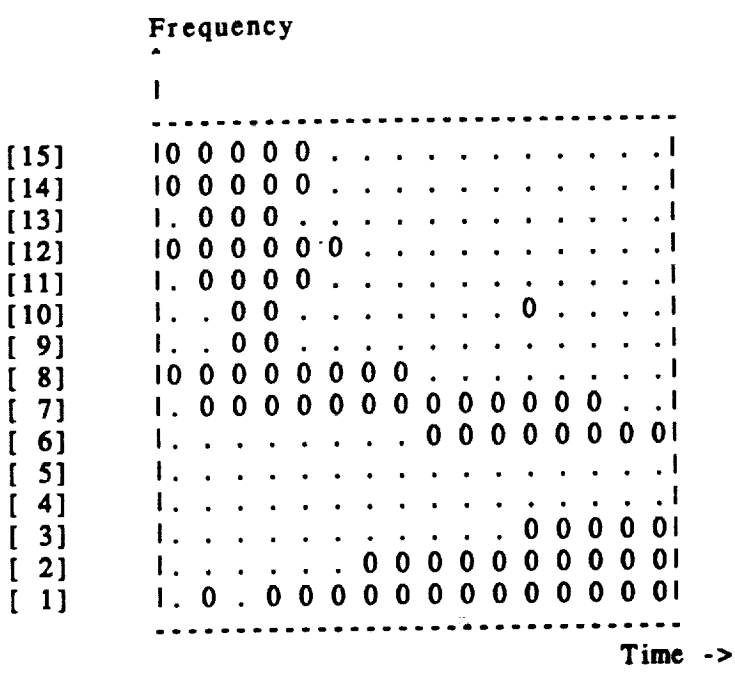
The experience gained from these collection of experiments points to the need to be able to construct memories that are "locally discriminatory" and adaptive to the input patterns presented to them.

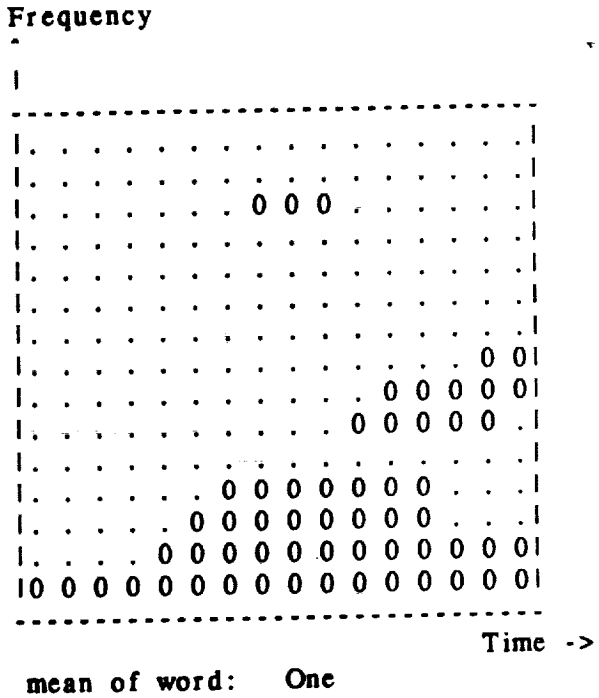
It was seen that random placement of addresses was not as good as guided address placement. It was also seen that attention to local bit interdependencies was better than uniform treatment of bit errors.

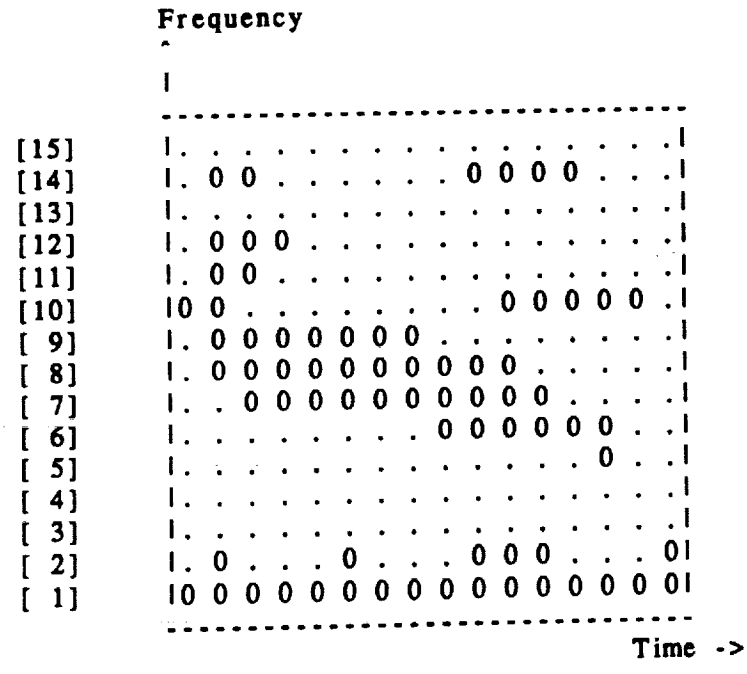
Combining adaptive placement with selective attention should lead to even more efficient, higher calibre, associative memories.

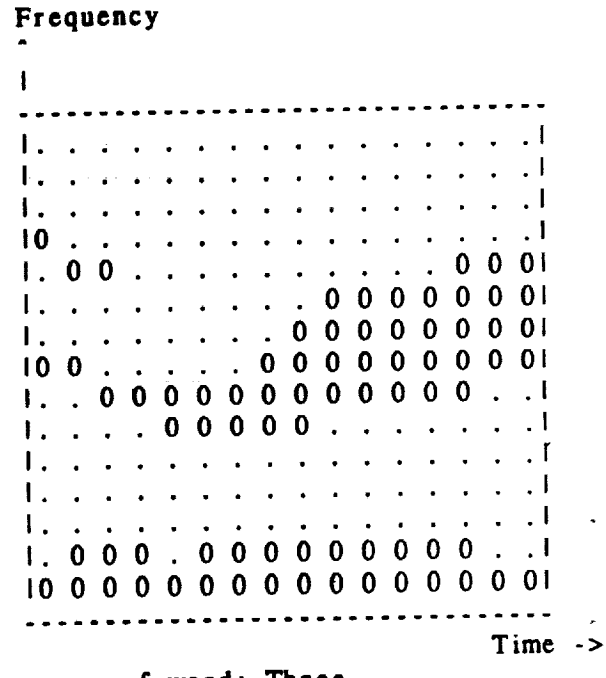
17. Appendix A - pictorial representation of average digit templates

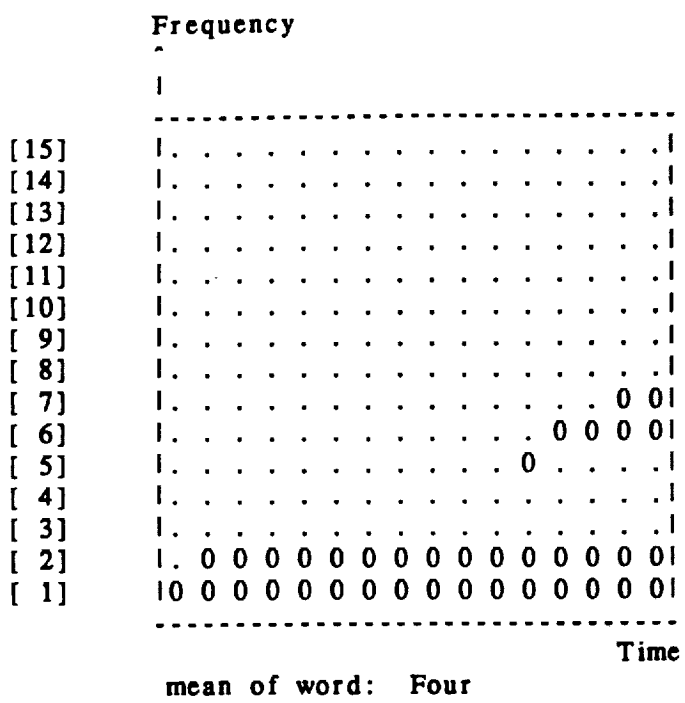
Below is presented the quantized mean patterns of the digits taken from the DIGIT64 database. A value of "0" indicates that, out of the 64 repetitions of the pattern, the bit 0 occurred more often than the bit 1 in this location. For visual clearity a period "." is used to represent a bit value of 1.

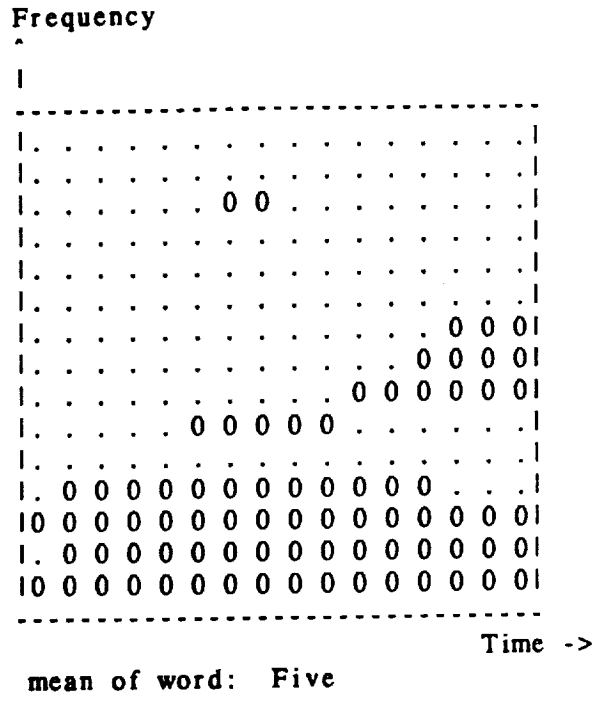


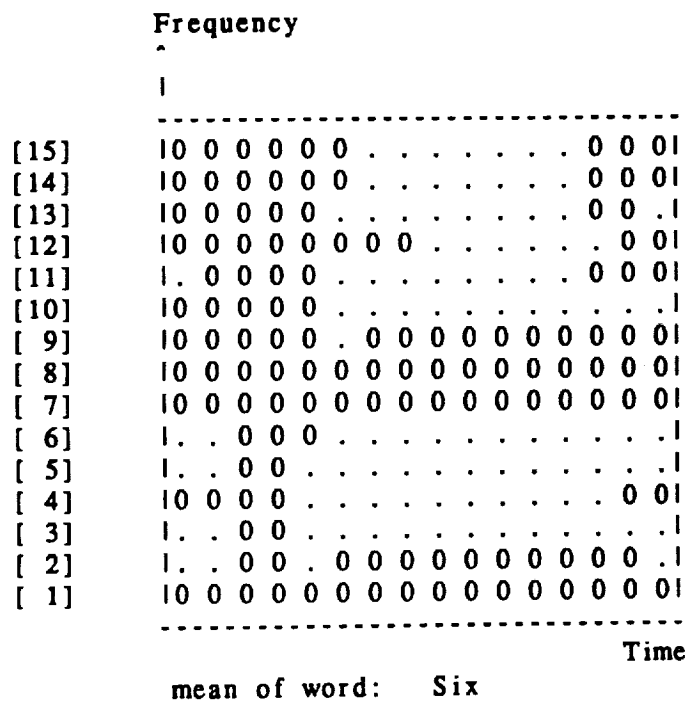


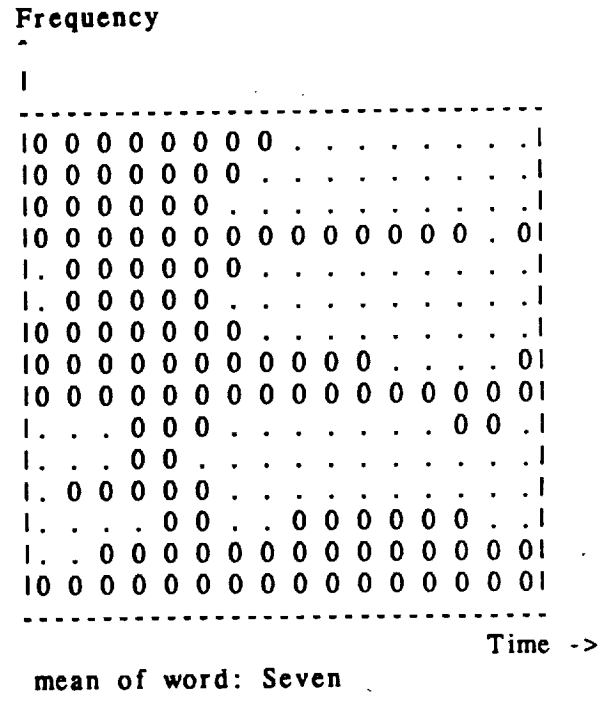


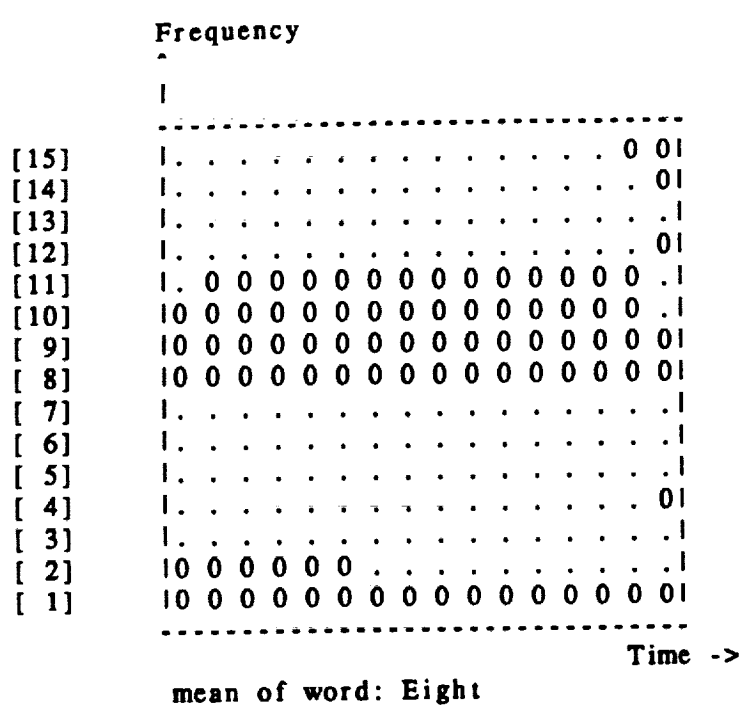


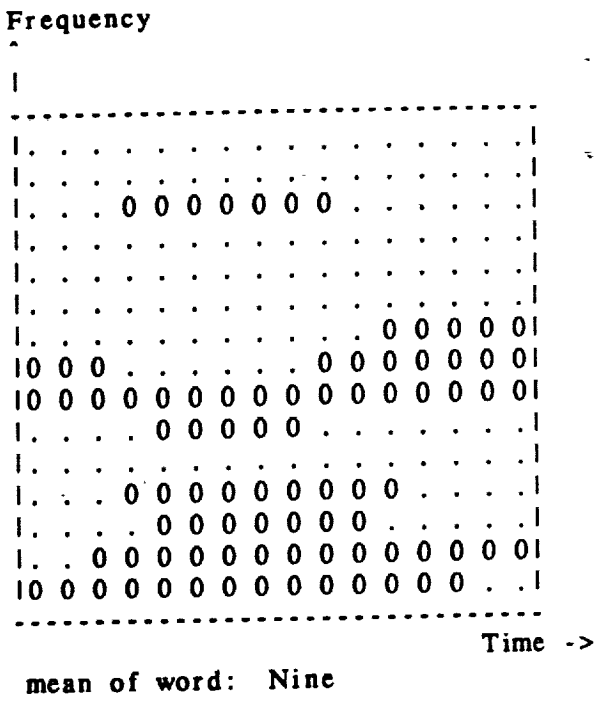

Frequency
[15]
[14]
[13]
[12]
[11]
[10]
[9]
[8]
[7]
[6]
[5]
[4]
[3]
[2]
[1]
Time ->
mean of word: Four


Frequency
Time ->
mean of word: Five


Frequency
[15]
[14]
[13]
[12]
[11]
[10]
[9]
[8]
[7]
[6]
[5]
[4]
[3]
[2]
[1]
Time ->
mean of word: Six


Frequency
Time ->
mean of word: Seven


Frequency
[15]
[14]
[13]
[12]
[11]
[10]
[9]
[8]
[7]
[6]
[5]
[4]
[3]
[2]
[1]
Time ->
mean of word: Eight

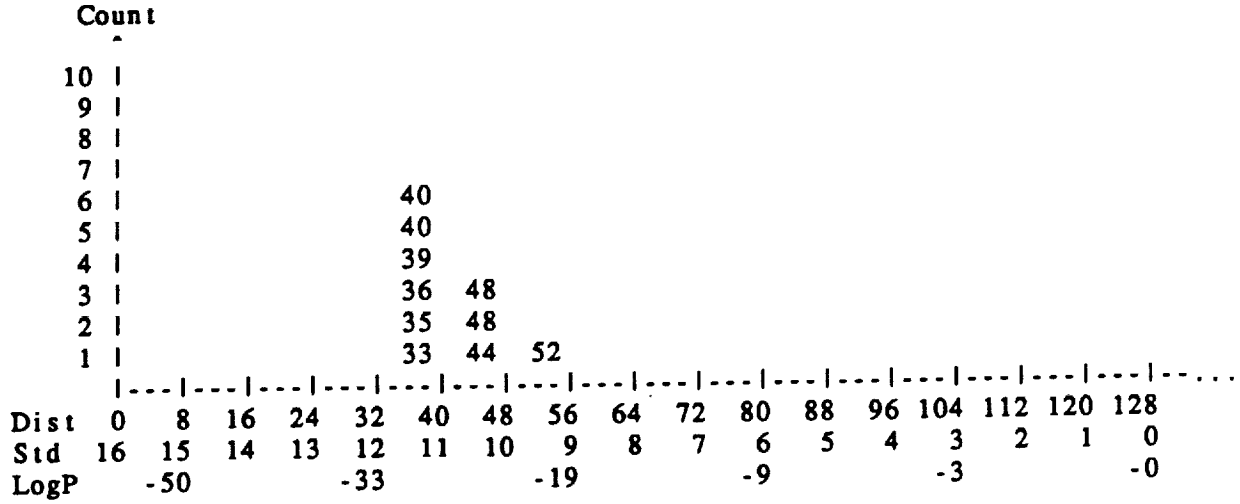

Frequency
Time ->
mean of word: Nine

18. Appendix B - Distribution of distances of digits

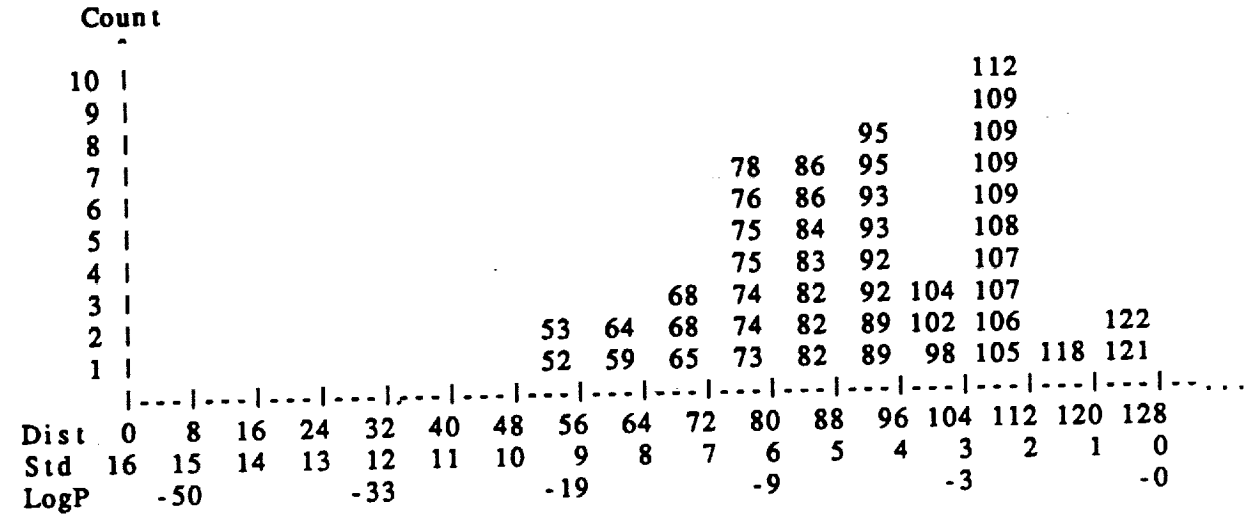
Below is presented the average Hamming distance of patterns within-digits and between-digits for the DIGIT64 database.

	Zero	One	Two	Three	Four	Five	Six	Seven	Eight	Nine
Zero	40									
One	82	39								
Two	68	89	48							
Three	86	83	74	40						
Four	74	52	75	73	35					
Five	98	53	104	84	64	36				
Six	89	122	93	86	109	121	48			
Seven	76	109	93	107	102	106	82	52		
Eight	105	109	82	78	92	118	92	112	33	
Nine	95	65	95	68	75	59	108	107	109	44

Average Hamming distance of digit observations



Histogram of average distances WITHIN-DIGITS



Histogram of average distances BETWEEN-DIGITS

19. Appendix C - Words used in speech-determined addresses

Below are the 850 words (not distinct) used to create and load the hard locations of experiments D4 and D5. All numeric references and words such as "to", "two", "too", "four", and "for" have been deleted from the original text.

THE FREEDOM OF SPEECH JANET M BAKER PHD DRAGON SYSTEMS INC CHAPEL BRIDGE PARK CHAPEL STREET NEWTON MA USA ABSTRACT SKILLFULLY CAST A NEW COMMUNICATIONS LINE TETHERS AND SECURES THE STRUCTURE A NEWLY CREATED WEB OF CAPABILITIES BLOSSOM ABOUT IT THE CULTURAL CHANGES WROUGHT BY ESTABLISHING EAST WEST TRADE ROUTES OR INEXPENSIVE TELEPHONES ARE IRREVERSIBLE THE FREEDOM WE ENJOY IN SPEAKING OTHER PEOPLE WILL SOON EXTEND OUR SPEAKING WITH MACHINES AS WELL HOW WE EXERCISE THIS NEW FREEDOM WILL DETERMINE NEW WAYS WORK AND PLAY THIS ESSAY EXPLORES THE PROGRESS OF SPEECH PAST PRESENT AND FUTURE AND THE CREATIVE CHALLENGES NOW SPINNING ABOUT US CODED COMMUNICATIONS TIME-SPACE COMPRESSIONS HUMAN SPEECH HAS PROBABLY BEEN EVOLVING SINCE THE DAWN OF HUMANITY THE REFINEMENTS OF SOUND AND STRUCTURE HAVE ESTABLISHED LANGUAGE COMMUNICATE THE IDEAS AND INFORMATION WE SHARE AS A SOCIETY SPEECH IS A UNIQUELY EFFECTIVE REAL-TIME INTERACTIVE COMMUNICATION MODE FREELY PRODUCED AND CONSUMED BY US ALL IN PRIVATE CONVERSATION AS WELL AS PUBLIC ORATORY SPEECH OPERATES AS A BROADCAST MEDIUM DISTRIBUTING INFORMATION OVER SHORT MODERATE DISTANCES UNCONSTRAINED BY VISUAL BARRIERS IN TIMES LONG PAST SPEECH PROVIDED THE PRIMARY VEHICLE CONVEYING IN STORY AND SONG THE NEWS AND TRADITIONS OF THE DAY AS TOLD BY BARDS AND OTHERS IN THEIR TRAVELS THE CODIFICATION OF SOUND INTO SPOKEN LANGUAGE PERMITS PEOPLE MORE COMPLEX COMMUNICATIONS AND CONSEQUENTLY MORE CHOICES IN CONDUCTING THEIR LIVES WITH DISTINCTIVE INDIVIDUALITY INDEED THE VALUE OF THIS FREEDOM IS REFLECTED IN THE SEVERITY OF PUNISHMENT METED OUT AT THE TOWER OF BABEL AS THE PRICE OF PRIDE THE TOWER'S PEOPLES FOUND THEMSELVES EACH SPEAKING DIFFERENT LANGUAGES AND UNABLE COMMUNICATE WITH SPEECH FURTHER CEMENTING THIS BARRIER THEY WERE THEN SENT ABROAD SEPARATE LANDS PROMETHEUS BOUND NUMBER THE PRIMARY SCIENCE I INVENTED THEM AND HOW SET DOWN WORDS IN WRITING THE ALL-REMEMBERING SKILL MOTHER OF MANY ARTS WE LEARN FROM AESCHYLUS THAT HUMANITY WAS GIFTED WITH WRITING FIRE AND OTHER TREASURES STOLEN FROM THE GODS BY PROMETHEUS FALLEN FROM THEIR GRACE INFURIATED ZEUS RETALIATED BY CHAINING PROMETHEUS A MOUNTAINTOP AND UNLEASHING THE EVILS OF PANDORA'S BOX UPON HUMANITY WHETHER BESTOWED BY PROMETHEUS' HAND OR NOT WE KNOW THAT WRITTEN LANGUAGE IS RELATIVELY RECENT PROGRESSING FROM THE PHOENICIAN ALPHABET MESOPOTAMIAN CUNEIFORM AND EGYPTIAN HIEROGLYPHICS TOOK ABOUT YEARS IN THE PAST YEAR TIME-FRAME WRITTEN INFORMATION COULD THE FIRST TIME PROVIDE A RELIABLE AND PERMANENT RECORD AND AFFORD ACCURATE COMMUNICATION WITH PEOPLE AT OTHER TIMES AND OTHER PLACES AS CLAY TABLETS GAVE WAY LESS CUMBERSOME AND COSTLY PAPER SIMPLE INSCRIBED PRONOUNCEMENTS THE MASSES COULD BECOME MORE DETAILED WITH LESS EFFORT AND MORE PRIVATE THE GREATER CARE AND COST OCCASIONED BY CONSTRUCTING COMPACT COMPOSITIONS IN COMPARISON WITH SPOKEN MESSAGES WERE BALANCED BY THEIR UNIQUELY ACCURATE PORTABLE TRANSMISSION AND DISSEMINATION OF KNOWLEDGE AND INFORMATION SINCE GUTENBERG'S INVENTION OF THE PRINTING PRESS IN THE POOR HAVE GAINED ACCESS THIS WRITTEN KNOWLEDGE AS WELL AS THE PRIVILEGED FEW AS THE WRITTEN WORD HAS BECOME EVER MORE BROADLY DISTRIBUTED WORLD LITERACY HAS GROWN IN RESPONSE THE EASE OF PRODUCING PRINT AS WELL AS CONSUMING IT HAS RESULTED TODAY IN MORE WRITTEN LESS WELL THAN EVER

BEFORE IN ADDITION THE STEADY DEMAND EVER MORE PRINTED PUBLICATIONS THE DESIRE PERSONAL COPIES OF TEXT ON DEMAND HAS TOTALLY ECLIPSED CARBON PAPER IN THE FLOOD OF PHOTOCOPIES THE LONG-PROMISED PAPERLESS OFFICE ELUDES US ONCE AGAIN LONG-DISTANCE COMMUNICATIONS ONCE EXEMPLIFIED BY ROMAN HILL-TOP SIGNAL FIRES AMERICAN INDIAN TOM-TOM DRUMS AND MECHANICAL SEMAPHORE TOWERS HAVE GIVEN WAY IN REMARKABLY SHORT ORDER ORBITING SATELLITE STATIONS FROM CARRIER PIGEON PONY EXPRESS TRAINS TRUCKS AND PLANES THE ADVENT OF EACH NEW MEANS OF TRANSPORTATION HAS SHRUNK THE TIME AND SWELLED THE FLOW OF INFORMATION IN THE PAST YEARS WE HAVE WITNESSED THE INVENTION OF THE TELEGRAPH TRANSMIT ELECTRICALLY CODED SIGNALS REPRESENTING LETTERS OVER LONG DISTANCES BY WIRE USING SPECIALIZED KEYSETS AND SKILLED MORSE CODE OPERATORS THE TELEGRAPH MADE RAPID REMOTE MESSAGING VIABLE THE FIRST TIME AND JUST AS TYPEWRITERS WERE COMMING INTO VOGUE THE INVENTION OF THE TELEPHONE FINALLY SUCCEEDED IN TRANSMITTING SPEECH ITSELF BY FIRST TRANSFORMING THE SOUND PRESSURE WAVEFORM INTO FLUCTUATING ELECTRICAL CURRENT AND THEN BACK AGAIN SPURRED ON BY THE FREEDOM OF RAPID REMOTE SPEECH NETWORKS SPRUNG UP WHILE INEXPENSIVE HANDSETS PROLIFERATED THROUGHOUT THE PUBLIC AND PRIVATE SECTORS TEN YEARS LATER IN HEINRICH HERTZ PRODUCED THE FIRST RADIO WAVES WITHIN THE NEXT DECADE MARCONI DEMONSTRATED A RADIO TRANSMISSION AND ESTABLISHED MARCONI'S WIRELESS TELEGRAPH PROMOTE SHIP-TO-SHORE COMMUNICATIONS ONLY A SHORT WHILE THEREAFTER THE FIRST TRANSATLANTIC COMMUNICATIONS WERE DEMONSTRATED THOUGH DUE IN PART THE UNEXPECTED ASSISTANCE RENDERED BY IONOSPHERIC REFLECTIONS WHICH COMPENSATED THE EARTH'S CURVATURE IN SIMULTANEOUS SOUND AND PICTURE WIRELESS BROADCASTS CREATED TELEVISION REGULAR BBC TRANSMISSIONS COMMENCED IN THE PORTABILITY AND AFFORDABILITY OF THE TELEVISION ONCE AGAIN ESTABLISHED A MECHANISM WIDESPREAD INFORMATION FLOW PEOPLE RICH AND POOR THE PAST YEARS SATELLITES HAVE COME OF AGE EXTENDING OUR REACH YET FURTHER YET IN ALL THESE PATHWAYS AND TRANSMISSIONS SO CLEVER WE OBSERVE THAT LANGUAGE SPOKEN AND LANGUAGE WRITTEN DESPITE THEIR PARALLELS ARE AS SEPARATE TODAY AS THEY WERE YEARS AGO THE ULTIMATE ALCHEMY IN WHICH WE ARE NOW ENGAGED TRANSMUTES THE SPOKEN AND WRITTEN BRINGING LANGUAGE FULL-CIRCLE AT LAST

References

Baker, J. M. (1984). The Freedom of Speech. International Conference on Speech Technology 23-25 October, Metropole Hotel, Brighton, UK. IFS (Conferences) Ltd, Kempston, Bedford, UK.

Cover, T. M. and P. E. Hart (1967). Nearest neighbor pattern classification, *IEEE Trans. Info. Theory*, IT-13, 21-27.

Harwit, M. and N. J. A. Sloane (1979). *Hadamard Transform Optics*. Academic Press, New York.

Jaeckel, L. A. (1989). An Alternative Design for a Sparse Distributed Memory. RIACS Technical Report 89.28.

Joglekar, U. (1989). Learning to Read Aloud: A Neural Network Approach Using Sparse Distributed Memory. RIACS Technical Report 89.27.

Kanerva, P. (1988). *Sparse Distributed Memory*. Cambridge, MA: MIT Press.

Keeler, J. D. (1987). Capacity for Patterns and Sequences in Kanerva's SDM as Compared to Other Associative Memory Models. RIACS Technical Report 87.29.

Keeler, J. D. (1988). Comparison Between Kanerva's SDM and Hopfield-type Neural Networks. *Cognitive Science*, 12,299–329.

Marr, D. (1969). A theory of cerebellar cortex. *Journal of Physiology* 202:437-470.

Olshausen, B. (1988). A Survey of Visual Preprocessing and Shape Representation Techniques. RIACS Technical Report 88.35.

Prager, R. W. and F. Fallside (1989a). The modified Kanerva model for automatic speech recognition. *Computer Speech* and *Language*, 3,61–81.

Prager, R. W. and T. J. W. Clark (1989b). The Modified Kanerva model: Results for Real Time Word Recognition. IEE First International Conference on Artificial Neural Networks, Savoy Place, London, 16-18 October 1989.

Rogers, D. (1988). Kanerva's Sparse Distributed Memory: An Associative Memory Algorithm well-suited to the connection machine. RIACS Technical Report 88.32.