A PETABYTE SIZE ELECTRONIC LIBRARY USING THE N-GRAM MEMORY ENGINE

Joseph M. Bugajski
Triada, Ltd.
4251 Plymouth Road
Ann Arbor, Michigan 48105
(313) 663-8622
(313) 663-7570
triada@middlec.convex.com

ABSTRACT: A model library containing petabytes of data is proposed by Triada, Ltd., Ann Arbor, Michigan. The library uses the newly patented N-Gram™ Memory Engine (Neurex™), for storage, compression, and retrieval. Neurex splits data into two parts: an hierarchical network of associative memories that store "information" from data, and a permutation operator that preserves sequence. Neurex is expected to offer four advantages in mass storage systems. (1) Neurex representations are dense, fully reversible, hence, less expensive to store. (2) Neurex becomes exponentially more stable with increasing data flow, thus, its contents and the inverting algorithm may be mass produced for low cost distribution. Only a small permutation operator would be recalled from the library to recover data. (3) Neurex may be enhanced to recall patterns using a partial pattern. (4) Neurex nodes are measures of their pattern. Researchers might use nodes in statistical models to avoid costly sorting and counting procedures.

Neurex subsumes a theory of learning and memory that the author believes extends information theory. Its first axiom is a symmetry principle: learning creates memory and memory evidences learning. The theory treats an information store that evolves from a null state to stationarity. A Neurex extracts information from data without *a priori* knowledge; i.e., unlike neural networks, neither feedback nor training is required. The model consists of an energetically conservative field of uniformly distributed events with variable spatial and temporal scale, and an observer walking randomly through this field. A bank of band limited transducers (an "eye"), each transducer in a bank being tuned to a sub-band, outputs signals upon registering events. Output signals are "observed" by another transducer bank (a mid-brain), except the band limit of the second bank is narrower than the band limit of the first bank. The banks are arrayed as n "levels" or "time domains, td." The banks are the hierarchical network (a cortex), and transducers are (associative) memories.

A model Neurex was built and studied. Data were 50 MB to 10 GB samples of text, data base, and images - black/white, grey scale, and high resolution in several spectral bands. Memories at td, $S(m_{td})$, were plotted against outputs of memories at td-1. $S(m_{td})$ was Boltzman distributed, and memory frequencies exhibited Self-Organized Criticality (SOC) [Bak *et al.* (1987) Phys Rev Lett: *59*, 381-384]; i.e., "$1/f^{\beta}$" after long exposures to data. Whereas output signals from level n may be encoded with $B_{output} = O(-\log_2 f^{\beta})$ bits, and input data encoded with $B_{input} = O([S(td)/S(td-1)]^n)$, $B_{output}/B_{input} \ll 1$ always, the Neurex determines a canonical code for data and it is a (lossless) data compressor. Further tests are underway to confirm these results with more data types and larger samples.

1. Introduction

Electronic libraries holding 10^{15} bytes (one petabyte, PB) of information are being planned. The Library of Congress' Global Knowledge Network, NASA's EOS/DIS, the Sequoia earth science project, and seismic data collections at major oil companies may be measured in petabyte units within ten years [1][2][3]. These large libraries will adopt information system technologies that compress data, store and retrieve information from very high density storage devices, and answer queries using knowledge of the information in the library. The Neurex™ memory engine for mass storage applications, being developed by our firm Triada, Ltd., Ann Arbor, Michigan, should provide features large libraries will require. And it is being considered for beta installation by several large libraries. Here we introduce the technology behind Neurex; N-Gram™, learning and memory theory. We review the N-Gram associative memory form that equates information with storage locations. We report results of tests using data samples provided by prospective Neurex users to show that Neurex losslessly compresses data at rates up to 200:1. In the attachments we illustrate the N-Gram learning transform and the Neurex machine.

How will petabytes of information be stored? How will users retrieve information from a petabyte library? Is it possible to just automate card catalogs or expand the scale of file based or database management systems? The first question appears to have been answered. The other questions are actively debated under the rubric of *metadata*.

Data storage technology now can support petabyte storage systems using mini-supercomputers running UNIX and UNITREE, redundant arrays of inexpensive disks (RAID), and petabyte libraries comprising helical scan tape [4][5][6]. A large storage system model is being built at the National Storage Laboratory at the Lawrence Livermore National Laboratory[7][8]. With it data storage technology advances from a role subservient to computers to an egalitarian role in a network of computing devices. But key issues are unsolved, including support for high performance computing [9].

The *metadata* problem requires integrating storage management with data management and current technology does not solve the problem [10]. First, databases do not extend to tertiary stores [11]. Second, unstructured data requires many file names. Suppose text files are .01 MB and image files are 20 MB. The catalog for a 1 PB system then has 1 billion names. 2.5 kilobytes per name requires a 2.5 terabyte card catalog on fast storage. The naming problem can be experienced today firsthand. Issue a global query on Internet. It may be days before the system contacts tens of thousands of nodes and it might not come back [12].

Meta-data is an intelligence modeling problem; data must become information. Researchers are attacking it from two directions. We call one the Turing paradigm; the other we call the connectionist paradigm [13].

The Turing paradigm works from the *top down*. One studies a phenomenon, e.g., intelligence, to deduce an algorithm that will operate on input data and output the phenomenon of interest. Ostensibly a metadata transformation is sought to map data into information by a finite number of instructions that can be executed on a computer in polynomial time, and the program can be self modifying. Artificial intelligence (AI) attempts to provide a complete solution, while database theory (DBT), information retrieval (IR), and information filtering (IF) attack parts of the problem.

Although AI, DBT, IR, and IF have progressed during the past twenty years, a general transform for changing data into information has not been discovered [14]. Notwithstanding the problems inherent in intelligence modelling,

research according to the Turing paradigm is robust and new publications are numerous. [15] is about (AI) implementation issues. [16] is a classic AI reference. [17][18] review problems in image representation and understanding. [19] and accompanying articles review database theory. [20] defines a general IR system model. [21] explains basic concepts in IR and compares these with IF, and [22] reviews an AI application at the U.S. Census Bureau. An intriguing extension of AI learning models, which has a flavor of fuzzy logic and poses interesting issues when juxtaposed with semantic logic, is relevance feedback theory [23]. Finally, no review of AI is complete without referencing Japan's Fifth Generation Language Project [24].

Solutions following the Turing paradigm that employ indexing methods could exacerbate the storage problem and not solve the metadata problem. Database keys and indices within text and images must be in primary memory but primary memory costs are high. If indices measure 10^{10} bytes and more, total system costs could measure ($ U.S.) 10^{7} or more. Indices in tertiary storage expand storage costs and they are useless until data is moved to primary storage.

The connectionist paradigm works from the bottom up and is a branch of cellular automata theory. Cellular automata are "discrete dynamical systems whose behavior is completely specified in terms of a local relation" [25]. The phenomenon exhibited by a cellular automaton is expressed by a behavior rule for the individual components. Hence, a researcher who wants a cellular automaton to act intelligently must discover a local relation that globally will make the automaton seem intelligent. Most current research defines local relations as either the spin glass model of John Hopfield, or the Boltzmann machine model of Terrence Sejnowski [26][27]. An alternative to the energy function models is the autocorrelation model [28]. Kevin Knight surveys the field, and he contrasts the Turing and connectionist paradigms [29]. Three survey works are [30][31][32]. Self-organizing systems and a review of several of the problems mentioned here is in [33]. Marvin Minsky wrote rules for a novel automaton that departs from the connectionist model [34].

The connectionist paradigm also does not solve the metadata problem. First, memory is not invertible and given the continuous functions of the local relations the capacity is unknown in general [35]. Second, neural networks can fall into spurious minima and not yield correct answers [36]. Third, they are not entirely bottom up because behavior derives from *a priori* training procedures. Example: A network taught to recognize type written characters will not recognize hand print. [37] gives a more complete introduction to problems in machine learning including an introduction to the literature of machine learning paradigms.

The above argues that the metadata problem cannot be solved following either the Turing paradigm or the connectionist paradigm. The crux of the metadata problem is that its solution may depend on answering a more profound question, *what is meaning*, which begs another profound question, *what is mind*? [38] Study of these go to the heart of philosophical enquiry dating back to antiquity, and have been investigated by the world's greatest minds: in jargon, the problem is highly non-trivial.

Triada is developing what we believe to be a robust solution to the metadata problem. It is obtained by attacking the metadata problem as a learning transform problem. Learning in our model is a metric tensor that under suitable conditions reversibly maps vectors of data into memories that are forms, i.e., information, and thus departing philosophically from the above paradigms. We study a general model of an observer equipped with a bank of band limited transducers attached to a hierarchical memory structure. The observer randomly walks through a region bounded by its lifetime and containing objects that reflect photons thereby allowing the observer to "see" the objects. The observer's input transducers register events within their frequency band limit by outputting a signal to the discrete learning transform. A set of ordered signals is a vector that is mapped into a memory form by the learning

transform. The set of all forms recorded this way describe the path taken by the observer, and transforming these into their dual space equivalent constitutes a faithful memory of the objects along the path in the neighborhood of the observer. Thus, memories are *p-forms* and electromagnetic events are *n-vectors*. Our conclusion is that information is a form while data is a vector, and the learning tensor is the desired metadata transform, that is, *memory and information are the same phenomenon*. The transform in hand we introduce the Neurex memory engine that embodies it. We present results of tests using a Neurex prototype and discuss the benefits afforded by this new technology. In particular, we will show results indicating 85:1 compression of text and 341:1 of fax image data. We will conclude with a review and talk about future research directions.

2. N-Gram Learning and Memory Theory

The learning transform acting on a field of electromagnetic events and registering differential patterns, or forms, is called a Poisson process [39]. Individual memories accumulate at each level of the memory hierarchy at a rate that decays exponentially, their probability of occurrence within any subregion of the entire region bounded by the observer's lifetime is Poisson distributed, the length of the path required to completely map all objects into the observer's memory is gamma distributed. Because sums of Poisson distributed random variables are Poisson distributed the growth of the entire memory is readily characterized.

Energy values (the memory forms) as memory is well accepted; minimal energy states are memories in both Hopfield and Boltzmann neural networks. Recently Friedland and Rosenfeld recognized a class of objects using an energy function [40]. Their work followed Geman and Geman who showed the Gibbs (Boltzmann) distribution and the characterization of an image as a *Markov Random Field* (MRF) were equivalent, where an image is a pair of matrices, the matrix of grey levels, and its dual, the edge matrix. Eugene Margulis applies a related concept in multiple Poisson models of word distributions in full text documents [41]. He demonstrated empirically that the meanings of particular words are multiply Poison distributed according to distribution parameters π_i and λ_i; where i counts the number of subjects, π_i is the probability the i'th subject is covered in a document, and λ_i is the mean occurrence of a word in the i'th subject.

We hypothesize the existence of measures $\lambda_{\beta,\alpha}$ of local information content, and other measures $\mu_{\tau,\beta}$ of global information content. The measures $\mu_{\tau,\beta}$ are the boundaries of the r volumes that contain the $\lambda_{\beta,\alpha}$, both sets of measures are found during a point-wise continuous random walk through all parts of an energetically conservative data field. Should a path of the walk be restricted to a surface of constant energy then only events with the same information will be found. But, these are elementary results in probability theory where the gamma and Poisson distributions are shown to be related, and the Boltzmann distribution is a special case of the gamma distribution [42][43]. In particular, the sum of t Boltzmann distributed random variables with parameter λ is gamma distributed with parameters (t, λ), and the probability that there are k occurrences of an event, say a particular word appears in an interval of length t is Poisson distributed. The equivalence of Markov and Poisson processes then obtains by [44]. Hence Markov $\Leftrightarrow$ Boltzmann $\Leftrightarrow$ Poisson.

The N-Gram memory model is an elementary implementation of the above ideas. A data stream is input to the N-Gram algorithm. The stream is parsed into sets of words according to rules that are empirically determined to be appropriate for the data type. The processor receiving the input word pattern searches its local memory is to determine if the input word pattern has previously occurred. If it has previously occurred, a counter is incremented and a signal representative of the storage location of the pattern is output to the subsequent processing level. If the pattern has not previously occurred, it is assigned a place in storage, a signal representative of its new location is output to the subsequent processing level, and a counter is incremented to the value 1. The signals output to the next

processing stage are similarly treated.

We want to know the size of the output stream after n levels and we want to know the size of the hierarchical memory after x bytes of data have been read. We first determine the size of the memory structure.

The N-Gram Memory can be represented an arrays of numbers. The numbers may be from the set of integers (**I**), rationals (Q), real (**R**), or complex (**C**). Elements in each row, or level, in the network are mapped into the level immediately above it, and each element in a level is the image of a mapping of elements in the level immediately below it. Let us assume that the level elements are rank ordered by relative frequency from most to least frequent.[1] Let X be a data stream comprised of signals $\mathscr{E}_j$, $0 < j \le \Im$, $\Im$ a nonzero integer, from a nonempty range of signals measured by (real or complex valued) frequencies, $f_i < \mathscr{E} < f_f$, $0 < |f_f - f_i|$. Thus, $\mathscr{E}_j$ is a signal (most commonly, an n bit binary code) representing any frequency in the $j'th$ partition of the range $|f_f - f_i| / \Im$. Define a recognition event in an N-Gram Associative Memory Network as the image of a function $\mathfrak{L}$ from any nonempty string S of signals $\mathscr{E}_j$ along a data stream X. Hence, in the most general case, the N-Gram Associative Memory Network is the codomain of $\mathfrak{L}$ where the domain of $\mathfrak{L}$ is any "piece wise continuous" stream of signals.

Now, let $T = |t_{final} - t_{initial}|$ be any nonzero time interval. Let $\mathfrak{L}$ be any invertible function that rank orders its image by relative frequency, from most to least frequent. Above we said the N-Gram Memory, N, can be represented by an array of size Cl_{max} *by* TD with integer elements. Let the first level of N be the image of $\mathfrak{L}$ operating on a data stream X comprised of signals $\mathscr{E}_j$, where each signal is n bits long. Suppose $\mathfrak{L}$ begins sampling X at time $t_{initial}$ by consistently selecting s, $s \in \mathbf{I}$, $0 < s$, nonoverlapping contiguous signals from X. Hence, every S^1 has word length $W = s$ x n bits. Let $x_1, x_1 \in \mathbf{I}$, be the number of words S^1 sampled by $\mathfrak{L}$ during an interval T. Note, $x^1 = 0$ at time $t_{initial}$. Then the first level of N, M_1, is the set

$$M_1 = \{ m_{1,i} \mid m_{1,i} = \mathfrak{L}(S^1);\ |a| < |m_{1,i}| < |b|;\ \text{a, b, and } m_{1,i} \in \mathbf{R} \},\ \text{where}$$

$||b| - |a|| \ge \lceil Cl_{\max}(1) \rceil$, $Cl_{\max}(1)$ is an empirically determined constant, and $\lceil\ \rceil$ is the greatest integer function.

We call an element $m_{td,i}$ a "memory," and the level number is td, $1 \le td \le TD$. Note, also, that $\mathfrak{L}$ is invertible and its image is discrete and rank ordered, therefore, without loss of generality we define a new function I that substitutes for each $m_{1,i}$ its integer position, i.

Define the second level in N like the first level as the rank ordered image of $\mathfrak{L}$, $m2_{,i} = \mathfrak{L}(S^2)$. Here S^2 contains s^2 contiguous signals $\mathscr{E}$ from a data stream X. Every S^2 is now a digital word of length $W = s^2$ x n bits. Suppose, we define a binary function $\mathfrak{L}^*$, that has as its image the position values i of the elements of the second level M_2 of N, and $\mathfrak{L}^*$ takes as its arguments the two recognition events (position values) of the elements of the first level M_1 of N that are the level one images of the first and second halves of the signal S^2. Let $S^1(x^1_u)$ and $S^1(x^1_v)$ be the first and second halves, respectively, of a signal S^2 from X: u and v are indices. Then,

$i_2 = I(m_{2,i}) = \mathfrak{L}^*[\ \mathfrak{L}(k,l)\ ] = \mathfrak{L}^*[\ \mathfrak{L}(\ m_{1,k}\ ),\ \mathfrak{L}(\ m_{1,l}\ )] =$

$\mathfrak{L}^*[\ \mathfrak{L}(\ S^1(x^1_u)),\ \mathfrak{L}(\ S^1(x^1_v))] = \mathfrak{L}[\ S^1(x^1_u) \wedge S^1(x^1_v)\ ] = \mathfrak{L}[\ S^2\ ]$, where $\wedge$ is the concatenation operator.

Therefore, the second level of memories, M_2 in N, is the set $M_2 = \{ i_2 \mid i_2 = I(m_{2,i}) = \mathfrak{L}\{S^2) = \mathfrak{L}^*(p,q) \}$, where p,q are recognition events in level one, i.e., $p = \mathfrak{L}(S^1(x^1_u)$ and $q = \mathfrak{L}(S^1(x^1_v)$;

[1] If the m_i are integers, i.e., $m_i \in \mathbf{I}$, then $\mathfrak{L}$ is an indexing function. If the elements of the array are real (or rational), i.e., $m_i \in \mathbf{R}$ (Q), and $a = 0$, $b = 1$, and the relation above is $a \le m_i$, then $\mathfrak{L}$ is a correlation function. If the elements are complex $\mathfrak{L}$ is a contraction.

$i_2 \varepsilon \mathbf{I}$; $|a| < |m_{2,i}| < |b|$; a, b, and $m_{2,i} \varepsilon \mathbf{R}$ }; $||b| - |a|| \geq \lceil Cl_{\max}(2)$, and $Cl_{\max}(2) \rceil$, is an empirically determined constant

We can now define any memory level as the ordered set of integers $M_{td} = \{ i_{td} \mid i_{td} = I(m_{td,i}) = \mathcal{GL}\{S^{td}) = \mathcal{GL}^{*}(p,q) \}$, where the signal S^{td} is a binary word of length $W = s^{td}$ x n bits; p,q are recognition events in level td - 1; $i_{td} \varepsilon \mathbf{I}$; $|a| < |m_{td,i}| < |b|$; a, b, and $m_{td,i} \varepsilon \mathbf{R}$ }; $||b| - |a|| \geq \lceil Cl_{\max}(td) \rceil$, and $Cl_{\max}(td)$ is an empirically determined constant.

N-Gram technology is the study of the N-Gram Memory to better understand human knowledge, and to invent and develop more efficient information management systems. We obtain the empirical constant $Cl_{\max}(td)$

$$CL_{\max}(td) = \frac{CL(x_{td})}{(1 - e^{-\lambda x_{td}})} \tag{1}$$

where, λ is the mean of the information density of the data X, $Cl(x^{td})$ are the number of memories accumulated after x^{td} events, and $0 \leq x^{td}$ is the number of nonoverlapping contiguous signals S^{td} from X.

Equation (2) shows a relationship between the relative frequency of a memory at level td, $m_{td,i}$, and its rank in the relative frequency ordered list of memories at that level. This equation is related to (1) by the information mean density value, λ.

$$\begin{gathered} 2\lambda = f^{c,i} N_{c,i}, \\ \textit{whence,} \\ I(m_{TD,i}) = i_{TD} = \lceil \frac{2\lambda}{f^{c}} \rceil, \end{gathered} \tag{2}$$

$f^{c,i}$ is the (relative) frequency of the memory $m_{td,i}$ and c is the class number, therefore, $N_{c,i}$ is the i'th memory at level td. $c = \lceil \log_2(f^{c,i}) \rceil$. The total number of classes, C_{td}, that form at level td is exactly

$$C_{td} = \lceil \frac{1 - f^{0,0}}{2\lambda} \rceil \tag{3}$$

Therefore, the total number of memories at level td, is

$$Cl_{\max}(td) = 2\lambda \sum_{c=1}^{C_{td}} \overline{f^{-c}}, \tag{4}$$

where f^c is the class frequency.

Suppose X has a density λ at every td^2. Then using either (4) or (1), we calculate the number of memories in N formed after it has observed X. The length of X, $|X|$, must be much longer than $Cl_{max}(TD)$, the number of unique signals S_{TD} that occur in X; say that the length of X is greater than an integer N > 10: i.e., let the bit length measure be $|X| > N\ Cl_{max}(TD)$ (s^{td} x n). Thus, the number of memories M contained in a network N is $M = TD \times Cl_{max}(td)$.

The N-Gram algorithm N^*

(i) parses a data stream X into signals S_{td} that are binary words of size W, as defined above,
(ii) maps every S_{td} in X into one and only one element $m_{td,i}$ of N; and
(iii) outputs a data stream $N^*(X) = m_{TD,1}(x)$, where x is the number of signals S_{TD} input to N^*, and the output is ordered as $x = 1,2,3,...$

Each signal S^{TD} has word length W. The length of an output word $N^*(X)$ is $W^* = \lceil \log_2(Cl_{max}(TD)) \rceil$. Hence, the density improvement ratio ρ achieved by N^* as it processes X is simply, $\rho = W/W^*$. If N contains fewer than M memories then the density improvement ratio is degraded by a factor r, where r is of the order $O_{td}(r) \approx 2^{-c+1}$, where td is the lowest level at which $Cl_{td}(x) < Cl_{max}(td)$, and c is the corresponding frequency class. In this case the density improvement becomes $(1-O(r))\rho = W/(W^*+r')$, where $r' = \log_2(Cl_{td}(x))$.

3. Neurex System Tests

The machine embodiment of N-Gram learning and memory theory is called Neurex™ and it is patented [45]. Two prototype Neurex were built and tested using samples of data to (1) test predictions of N-Gram Theory, (2) measure memory populations, and (3) determine performance parameters. They were not designed to benchmark I/O performance nor to reduce data samples for compressed storage. Rather, both were designed to gather statistics to determine the relationship among the size of the memory structure, the amount of density improvement obtained with a given memory structure, the amount of physical storage that would be needed for a memory structure, and the distribution of the memories within lists of memories created by the N-Gram algorithm.

The first prototype was a set of boards with four Inmos Transputers installed in a 500 megabyte solid state disk (SSD) loaned to us by Zitel Corporation. The N-Gram algorithm was written in the "C" programming language. The SSD held a partial N-Gram Memory. The Neurex was linked by serial ports on the Transputers to Transputer boards installed in two IBM AT compatibles. The compatibles provided the programming environment, and they were used to load programs and test software, to supply test data, and to hold statistics gathered during test runs.

The N-Gram algorithm mapped patterns in the input data stream into the N-Gram memory array stored in the Zitel RAMDisk. Two memory classes were created: those having met a predetermined threshold value and which are stored permanently, and those which have not met the threshold and are stored temporarily. Memories that have not met the threshold value, and are thus kept temporarily, are eventually excluded into the output stream. Memories that have met the threshold value are mapped into the next higher level in the memory array to determine more complex features in the data stream. The amount of space available for memories bounded the length of the data stream that could be viewed; i.e., a window was created that reduced the exposure of the Neurex to low frequency data patterns slowing the growth of the permanent memory structure. The prototype permitted periodic measurements

[2] The assumption that the mean information density exists over a range of levels TD, is valid whenever the longest signal S_{TD} is small compared to the "field of view" of an N-Gram associative memory network N.

of the memories accumulated as a function of the number of events.

We also built a prototype consisting of N-Gram algorithm running on a Convex mini-supercomputer. Convex provided time on their laboratory machines and access to tape drives to load large data files. The algorithm was modified to process data in sections where every section contained only those data stream patterns that would be within the section of the memory structure in the primary memory.

Description of Test Data Samples

We tested samples of text, 10 bit four color images, black/white images, travel time data, data base data, a 10 gigabtye sample of 32 bit floating point numbers from a numerical analysis project at NASA Ames, and multiple spectral band data from the LandSat and NOAA 12 satellites. The text sample was 1.5 gigabytes of ASCII coded files from the University of Michigan's collection of weekly USENET Internet service articles. A 1 gigabyte sample three of LandSat scenes was provided by NASA Goddard Space Flight Center. A single scene consists of seven roughly equal sized segments, each of which represents a spectral view of the same area on the surface of the earth as viewed from the LandSat satellite. The black/white fax images were a 3.2 gigabyte sample of bank check images. The relational data base contained typical corporate records. The sample was 4.4 gigabytes long.

Test Results

The tests were designed to measure the information density of the data samples, and to calculate a compression ratio using the above equations.

The information density for each data sample was obtained and it was used to extrapolate compression results shown in Table I. The fax image sample required approximately 500 million memories to achieve a density improvement ratio of 341:1. The text data sample reached 85:1 with a 1.6 billion memories. To obtain a 43:1 density improvement the commercial data base required only 280 million memories. The samples that were most dense with information were the satellite images. We were estimated the size of a memory structure for these high resolution images would be 3.6 billion memories and it would achieve a density improvement of 73:1. The worst performance was with the seismic and floating point matrix samples, however, these were said to be incompressible using standard compression techniques (according to the owners of the data).

Table I: Neurex Data Compression Performance

Data Type	No. Memories	Output Code Word Length	Input Code Word Length	Compression Ratio
ASCII Text	$1.6 * 10^9$	24 bits	2048 bits	85:1
Fax Image	$5.0 * 10^8$	24 bits	8192 bits	341:1
Seismic	$5.2 * 10^7$	24 bits	64 bits	2.7:1
LandSat (8-bit pixels)	$3.6 * 10^9$	28 bits	2048 bits	73:1

NOAA 11 (8 bit pixels)	$3.6 * 10^9$	28 bits	2048 bits	73:1
Commercial Database	$2.8 * 10^8$	24 bits	1024 bits	43:1
Floating Point matrix (32 bit)	$7.0 * 10^7$	26 bits	32 bits	1.23:1

4. Neurex Model Library

A model library with 36 terabyte capacity is illustrated in the attachments. Key to the feasibility of the library are the above compression results and the application of the N-Gram memory form to pattern recognition.

5. Conclusions

The N-Gram learning and memory model holds for a large range of data types. The compression possible with the large memory structure is significantly greater than that achieved using state-of-the-art methods. While additional test are required using data samples that are significantly larger than the memory structure size, given the stationarity and ergodicity of the samples we tested there is no reason to believe a larger sample will produce significantly different results than those given above.

1. J.A. Adams, "Multimedia Repository," *IEEE Spectrum*, p. 29, March 1993.

2. C. D. Benjamin, "The Role of Optical Storage Technology for NASA's Image Storage and Retrieval Systems," in *Storage and Retrieval Systems and Applications* (1990), SPIE, vol. 1248, pp. 10-17.

3. M. Stonebreaker, J. Frew, K. Gardels, J. Meredith, "The Sequoia 2000 Storage Benchmark," in *Proc. 1993 ACM SIGMOD* (Wash., DC), P. Buneman & S. Jajodia, eds., SIGMOD Record, 22, Issue 2, June 1993, p. 3.

4. R. H. Katz, G. A. Gibson, and D. A. Patterson, "Disk System Architecture for High Performance Computing," *Proc. IEEE*, Vol. 77, No. 12, December 1989, pp. 1842-1858.

5. D. Lancaster, "Convex and the Fileserver Market," in *Proc. Seminar Series, Mass Storage and Fileserver Solutions for Networked Computer Systems*, Convex Computer Corporation publication, Richardson TX, 1992.

6. K. Ishii, T. Takeda, K. Itao, and R. Kaneko, "Mass Storage Technology in Networks," in *Storage and Retrieval Systems and Applications* (1990), SPIE, Vol. 1248, pp. 2-9.

7. S. Coleman and S. W. Miller, eds. "Mass Storage System Reference Model: Version 4." IEEE Technical Committee on Mass Storage Systems and Technology, May 1990.

8. S. Louis, S. W. Hyer, "Applying IEEE Storage System Management Standards at the National Storage Laboratory," in *Proc. Twelfth IEEE Symposium on Mass Storage Systems*, IEEE Computer Society Press, Los Alamitos, CA, S. S. Coleman, ed., 1993, 55-62.

9. W. Myers, "Supercomputing 92 reaches down to the workstation," *Computer*, IEEE Comp. Soc., Vol. 26, No.1, January 1993, pp. 113-117.

10. S. S. Coleman, R. W. Watson, R. A. Coyne, and H. Hulen, "The Emerging Storage Management Paradigm," in *Proc. Twelfth IEEE Symposium on Mass Storage Systems*, S. S. Coleman, ed., IEEE Computer Society Press, Los Alamitos, CA, 1993, pp. 101-110.

11. M. Carey, L. Haas, M. Livny, "Tapes Hold Data Too: Challenges of Tuples on a Tertiary Store," in *Proc. ACM SIGMOD*, P. Buneman and S. Jajodia, eds., ACM, NY, 1993, pp. 413-417.

12. J. J. Ordille and B. P. Miller, "Database Challenges In Global Information Systems," in *Proc. 1993 ACM SIGMOD*, P. Buneman and S. Jajodia, eds., ACM, NY., 1993, pp. 417.

13. M. Conrad, "Molecular Computing Paradigms," *Computing*, Vol.25, Nov. 1992, pp. 6-9.

14. M.V. Wilkes, "Artificial Intelligence as the Year 2000 Approaches," *Communications of the ACM*, Vol. 35, No. 8, August 1992, pp. 17-25.

15. B.R. Gaines and M. L. G. Shaw, "Eliciting Knowledge and Transferring It Effectively to a Knowledge-Based System," *IEEE Trans. on Knowledge and Data Engrg*, Vol. 5, No. 1, February 1993, pp. 4-14.

16. P. R. Cohen and E. A. Fiegenbaum, eds., *The Handbook of Artificial Intelligence*, Vols. I-III, William Kaufmann, Inc., Los Altos, CA, 1982.

17. S-K Chang and A. Hsu, "Image Information Systems: Where Do We Go From Here," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 4, No. 5, Oct. 1992, pp. 431-442.

18. C. C. Weems, E. M. Riseman, A. R. Hanson, "Image Understanding Architecture: Exploiting Potential Parallelism in Machine Vision," *Computer*, Vol. 25, No. 2, February 1992, pp. 65-68.

19. A. Silberschatz, M. Stonebreaker, J. D. Ullman, "Database Systems: Achievements and Opportunities," *SIGMOD RECORD*, ACM Press, Vol. 19, No. 4, December 1990, pp. 6-22.

20. J. Tague, A. Salminen and C. McClellan, "Complete Formal Model for Information Retrieval Systems," in *Proc. 14'th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, 1991, pp. 14-20.

21. N. J. Belkin and W. B. Croft, "Information Filtering and Information Retrieval: Two Sides of the Same Coin?," *Communications of the ACM*, December 1992, Vol. 35, No. 12, December 1992, pp. 29-38.

22. R. H. Creecy, B. M. Masand, S. J. Smith, D. L. Waltz, "Trading MIPS and Memory for Knowledge Engineering," *Communications of the ACM*, Vol. 35, No. 8, August 1992, pp. 48-64.

23. IJ. J. Aalbersberg, "Incremental Relevance Feedback," *Proc. 15'th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, NY, 1992, pp. 11-22.

24. E. Shapiro and D. H.D. Warren, eds., "The Fifth Generation Language Project: Personal Perspectives, *Communications of the ACM*, Vol. 36, No. 3, March 1993, pp. 46-103.

25. T. Toffoli and N. Margolus, *Cellular Automata Machines*, MIT Press, Cambridge, Mass., 1987, p. 5.

26. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. National Academy of Sciences*, Vol. 79, pp. 2554-2558.

27. D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science*, Vol. 9, pp. 147-169.

28. S. Amari and K. Maginu, "Statistical Neurodynamics of Associative Memory," *Neural Networks*, Vol. 1, 1988, pp. 63-73.

29. K. Knight, "Connectionist Ideas and Algorithms," *Communications of the ACM*, Vol. 33, No. 11, November 1990, pp. 72-74.

30. J. A. Anderson and E. Rosenfeld, eds., *Neurocomputing*, MIT Press, Cambridge, Mass., 1988.

31. D. E. Rumelhart, J. L. McClelland and the PDP Research Group, *Parallel Distributed Processing*, Vols. I-II, 1986.

32. I. Aleksander, ed., *Neural Computing Architectures*, MIT Press, Cambridge, Mass., 1989.

33. S. S. Iyengar and F. B. Bastani, eds., "Special Section on Self-Organizing Knowledge and Data Representation in Distributed Environments," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 4, No. 2, April 1992.

34. M. Minsky, *The Society of Mind*, Simon and Schuster, NY, 1986.

35. D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks," *Phys. Rev. Lett.*, Vol. 55, No. 14, 30 September, 1985, pp. 1530-1533.

36. M. Zak, "Terminal Attractors in Neural Networks," *Technical Support Package* for *NASA Tech. Brief*, Vol. 15, No. 7, July 1991, p. 1.

37. A. M. Segre, "Applications of Machine Learning," *IEEE Expert*, June 1992, pp. 30-34.

38. J.E.T., "Meaning," in *The Oxford Companion to the Mind*, Oxford Univ. Press, Cambridge, UK, 1985, pp. 450-454.

39. R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, Macmillan Publishing, N.Y., 1978, pp. 99-102.

40. N. S. Friedland and Azriel Rosenfeld, "Compact Object Recognition Using Energy-Function-Based Optimization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 7, July 1992, pp. 770-777.

41. Eugene L. Margulis, "N-Poisson Document Modelling," in *Proc. 15'th ACM SIGIR Conf. on Research and Development in Information Retrieval*, ACM Press, 1992, pp. 177-189.

42. R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, Fourth Ed., Macmillan, NY, 1978, pp. 99-109.

43. P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Probability Theory*, Houghton Mifflin, Boston, Mass., 1971, pp, 129-130, 146-7, 230-3.

44. S. Geman and D. Geman, Section IV, pp. 724-26.

45. J. M. Bugajski and J. T. Russo, "Data Compression with Pipeline Processors Having Separate Memories," U.S. Patent No. 5,245,337, Sept. 14, 1993.

Volume Serving and Media Management in a Networked, distributed Client/Server Environment

Ralph H. Herring and Linda L. Tefend

EMASS®Storage Systems
Solutions from E-Systems
P. O. Box 660023
2260 Merritt Drive
Dallas, TX 75266-0023
Phone: (214) 205-6478
Fax: (214) 205-7200
lindat@Emass.Esy.COM