Research Institute for Advanced Computer Science
NASA Ames Research Center

IN-62
43084
P52

Total Recall in Distributive Associative Memories

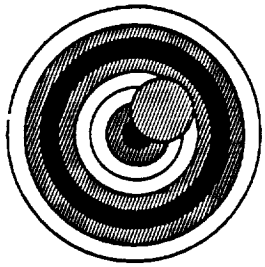
Douglas G. Danforth

RIACS Technical Report 91.03

January 23, 1991

(NASA-CR-188884) TOTAL RECALL IN DISTRIBUTIVE ASSOCIATIVE MEMORIES (Research Inst. for Advanced Computer Science) 52 p
CSCL 09B
G3/62
N92-11699
Unclas
0043084

Total Recall in Distributive Associative Memories



Douglas G. Danforth

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

ABSTRACT

Iterative error correction of asymptotically large associative memories is equivalent to a one-step learning rule. This rule is the *inverse* of the activation function of the memory. Spectral representations of nonlinear activation functions are used to obtain the inverse in closed form for Sparse Distributed Memory (Kanverva, 1988), Selected-Coordinate Design (Jaeckel, 1989), and Radial Basis Functions (Poggio, 1989).

The Research Institute for Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 311, Columbia, MD 21044, (301) 730-2656.

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

Table of Contents

List of Figures

1.0 Introduction

In this report, I study issues governing learning in associative memories which use patterns to recall other patterns. The patterns can be simple or complex, composed of one or many features. The features can refer to sensory input or high-level mental constructs. The paper is theoretical and treats these patterns as abstract vectors in n-dimensional spaces. There are many questions that can be asked about such systems, however, I focus on the following.

How can *precise* patterns be recalled when information is *distributed* throughout an associative memory? In what way can the effects of *interference* between patterns in the memory be *undone*? When an associative memory is implemented as an *artificial neural network* what *learning rule* will produce high capacity and precise recall? How does the issue of *generalization* relate to the issue of storage capacity?

These questions are examined for associative memories which have an unlimited number of memory locations. This extreme case reveals there exist very efficient single-step learning rules for high quality recall. The extreme-case rules are precisely determined yet are not simple nor intuitively obvious. I examine these rules in an effort to *understand* of the basic processes governing the behavior of distributive systems.

I have found this investigation to be insightful for it indicates the interplay between storage[1] and recall necessary to realize high quality memories. The picture that emerges is an *onion of* alternating layers of excitation and inhibition about each *neuron* of the memory (hence the logo on the title page). This picture is reminiscent of the on-center off-surround network found in the visual cortex of higher vertebrates (Malsburg, 1973). Such structures are necessary for producing sharp *edges* in distributive systems. Sharp edges are necessary for precise discrimination in small regions when there are high correlations between disparate input patterns.

The asymptotic results derived raise the question of whether there exist analogous rules for nonasymptotic memories. This question is left to a later publication (Danforth, 1991).

1.1 An example of total recall

One might well ask whether it is even possible to realize high quality recall in a distributive system. To illustrate that it is possible to realize *perfect* recall consider the following simple example.

Assume an associative memory has input x where x is a pattern. For this example assume the pattern is binary valued. Also assume there are but two bits in x (x_1 and x_2). Assume for each of the four possible inputs the output patterns $y = A, B, C$, and D are associated. It is not necessary to specify what these patterns are or what their dimensionality is, only that

1. The inverse rules derived here can equally well be applied to the recall process. If recall uses the inverse then learning uses the rule and visa versa.

ity is, only that they are numeric. They do not have to be distinct. Assume there is a memory location for every possible input pattern (the asymptotic case). The specified associations are depicted below.

In a distributive memory, when the pattern $y = A$ is associated with the pattern $x = (0, 0)$, it is not placed just at $(0, 0)$ but is written at several locations. This is partly motivated by the desire for fault tolerance under hardware failure. See Figure 2.

The rule used for the distribution of pattern A in Figure 2 writes it in all locations within a Hamming distance of one of $(0, 0)$. If this rule is applied to each of the patterns A,B,C, and D to be written into memory, the following configuration results (Figure 3).

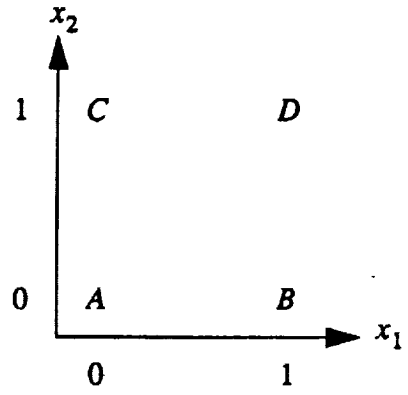


Figure 1 Desired associations.

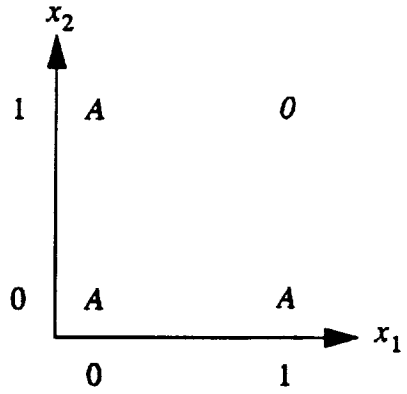


Figure 2 Distributed storage.

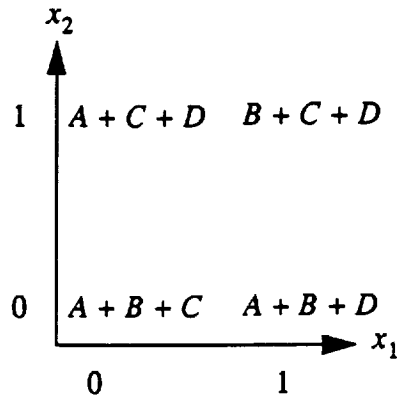


Figure 3 Multiple writes.

The contents of a memory location are taken to be the sum, e.g. $A + B + C$, of the patterns written into it. Memory locations accumulate information by adding to their present store. How can the desired pattern be untangled when this memory is read? If the same rule is used for writing and reading, one would expect to read from all locations within one bit of, say, $(0, 0)$. How is this done? The contents can be pooled (summed) to produce

$$\begin{aligned} y &= M(0,0) \\ &= (A + C + D) + (A + B + C) + (A + B + D) \\ &= 3A + 2B + 2C + 2D. \end{aligned}$$

It is not at all obvious that the pattern A which we wish to recover from location $x = (0, 0)$ will predominate in this sum. In fact, if all of the other patterns in memory are actually equal to Q $(B = C = D = Q)$, for some Q, then $y = 6Q + 3A$ and the pattern Q predominates over the pattern A (six versus three).

So it appears that distribution followed by pooling is not a good way to store and recover patterns precisely. However, one might think the pattern Q *should* be what is read from location $(0,0)$. The predominance of Q over A indicates A may be in "error". To accept Q is to allow the memory to *correct* errors by smoothing local irregularities. But what if the datum A is not error but signal? Then the signal has been lost in the *noise* of Q.

I will examine the case where all data are considered signal to be retained. For well-separated data the distribute-and-pool process meets the needs of robustness and generalization. If the data are not well separated then noise can swell to overcome the signal.

Is there an alternative way to distribute information such that, on recall, the signal predominates? The answer to this question is yes, and the way this distribution and recall process interact is illuminating.

Consider Figure 4 where the write rule has been changed. Locations within one bit of *(0, 0)* now receive the pattern *A/3* whereas the location two bits away in Hamming distance receives twice the negative of this value. Applying this new rule to each of the remaining patterns produces Figure 5.

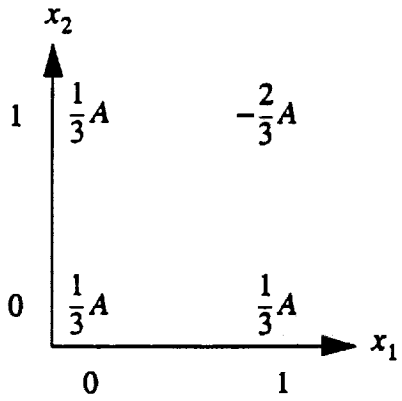


Figure 4 A new write rule.

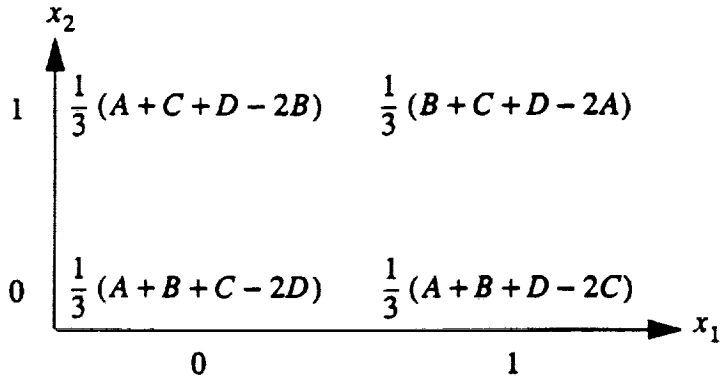


Figure 5 Multiple writes using new rule.

If the memory is now read at *(0, 0)* with the original pooling rule (the contents of all locations within a Hamming distance of one are added), then

$$
\begin{aligned}
y &= M(0,0) \\
&= ((A + C + D - 2B) + (A + B + C - 2D) + (A + B + D - 2C))/3 \\
&= (A + A + A + B + B - 2B + C + C - 2C + D + D - 2D)/3 \\
&= (3A + 0 + 0 + 0)/3 \\
&= A
\end{aligned}
$$

and the desired signal is recovered without error. The patterns B, C, and D at *(0, 1)*, *(1, 0)*, and *(1, 1)* are likewise recovered exactly. So by changing the write rule but retaining the read rule it is possible to compensate for the process of distributing information within the memory. How was the write rule determined? The answer to this question is the main topic of this report.

It can be argued that shuffling information around in a memory which *spans* all possible input patterns is not very enlightening since it is precisely the vastness of the number of these patterns (2^{1000}, say) that makes it impossible to construct such a memory. Having said this, it is still beneficial to understand what form the shuffling and encoding must take in order to retrieve patterns from a dense memory (one with all possible locations present). It will be shown that knowledge of this dense case provides information about the *ensemble average* of *sparse* memories. It will also be shown that error correction in artificial neural nets is related to this process of weighting and shuffling of information. In addition, the analysis of the asymptotic case provides a concrete place to stand for further analysis of smaller implementable memories.

2.0 Overview

This report develops a methodology for constructing single-step write-rules for asymptotically large single-hidden-layer artificial neural systems based on a *spectral representation* of their nonlinear activation functions.

In section 3.0 notation is developed for reading and writing using nonlinear operators.

In section 4.0 it is shown that repeated writing of errors into a memory implements a series expansion of the inverse of its activation function.

In section 5.0 the general notation of the spectral representation for activation rules is introduced and the general form of the inverse derived from it.

In section 6.0 the inverse is determined for Sparse Distributed Memory (Kanerva, 1988).

In section 7.0 the Selected-Coordinate Design (Jaeckel, 1989) is discussed and its inverse derived.

In section 8.0 Radial Basis Functions (Poggio, 1989) are introduced and an expression for their inverse is presented.

The report ends with section 9.0 giving a summary and conclusions of this research.

PRECEDING PAGE BLANK NOT FILMED

3.0 Notation

An *associative memory* is a system for constructing input-output mappings from examples of input-output pairs. The memory has a construct called a *location* with a set of input weights, $\dot{x}$, and a set of output weights, $\dot{y}$. The vector $\dot{x}$ corresponds to the *address* of the location in a Random Access Memory (RAM) and the vector $\dot{y}$ corresponds to the *data* stored at the location. A location is *activated* by an input pattern x if the address decoder or *activation function*, A, deems that the pattern is sufficiently similar to the address $\dot{x}$ of the location. If the location is activated, information y either can be stored at the location (weights $\dot{y}$ modified) or retrieved from the location (weights $\dot{y}$ contribute to the output from memory). The activation rule, A, is the *kernel* or fundamental aspect that characterizes an associative memory. This activation rule will be written symbolically to bring forth its functional dependence on input, x, address, $\dot{x}$, and dimensionality of the input space, n. When specializing to specific forms for Sparse Distributed Memory and Selected-Coordinate Design, other parameters will be introduced. The activation rule is written as

$$a = A_{x,\dot{x}}(n). \qquad \text{(EQ 1)}$$

By writing the activation rule as a function of two arguments (the input vector and the input weights), a very broad class of functions can be represented. It includes the standard linear threshold unit where the inner product of x and $\dot{x}$ is compared to a threshold. It also includes functions that may depend in a complex manner upon the interplay between input and weights. The result of activation, the quantity a, can be binary valued or real valued (e.g., sigmoid functions). The unspecified complexity of A has the potential of modeling not just a node in a single hidden layer but also a node deep within a multilayer system when the dimensionality of $\dot{x}$ is allowed to be far larger than that of x. In this paper it is assumed A is the activation rule for a memory with a single hidden layer.

Although the *results* of activation are treated *linearly*, it needs to be stressed that the activation rule A can be a highly *nonlinear* function of its arguments. Many of the results derived in this paper are independent of the exact form of this nonlinearity.

3.1 Sparse sampling

The pairing of input pattern, x, with output pattern, y, leaves unspecified whether the pair is actually observed in any finite sample. A notation is adopted here that depicts y as a direct function of x, namely, y_x (for discrete input spaces) or $y(x)$ (for continuous input spaces). This notation is taken to mean that at any input point x, y_x is the observed output at that point. If x does not occur in the sample, y_x is zero. If x occurs multiple times in the sample, y_x is the sum of the y's at that point. In like manner $\dot{y}_{\dot{x}}$ is the sum of the data stored at address $\dot{x}$. If there is no location in memory with address $\dot{x}$, $\dot{y}_{\dot{x}}$ is the zero pattern. With this understanding, the patterns x and $\dot{x}$ are allowed to range over all their possible values with y_x and $\dot{y}_{\dot{x}}$ constraining the associated values to those observed.

PRECEDING PAGE BLANK NOT FILMED

3.2 Projections

The issue of sparse *data sampling* can be specified cleanly in terms of *projection operators*. I define the operator Π to be the projection of the space onto the subspace spanned by sample data y. The action of Π on y leaves y unchanged. Since the operator is a projection it is idempotent (repeated application of a projection is the same as a single application). So

$$\Pi y = y, \qquad \text{(EQ 2)}$$
$$\Pi\Pi = \Pi.$$

For discrete spaces, one can think of the operator Π as the identity matrix with holes along the diagonal. The nonzero diagonal elements specify the input patterns that occur in the data.

In like manner, sparse *memory locations* can be represented as a projection, $\dot{\Pi}$, of the full space of all possible memory locations onto those actually occurring in the memory, $\dot{y}$. We then have

$$\dot{\Pi}\dot{y} = \dot{y}, \qquad \text{(EQ 3)}$$
$$\dot{\Pi}\dot{\Pi} = \dot{\Pi}.$$

These projection operators are mentioned here for completeness of exposition. They will be sparingly used in the body of this report but will play a stronger role in future research directed at the issue of generalization in sparse memory systems.

3.3 Recall

The process of recall from a distributive memory is taken to be a weighted summing of information from the locations of the memory conditionalized on the input pattern x. The weighting is determined by the activation rule A where

$$s_x = \sum_{\dot{x}} A_{x,\dot{x}} \dot{y}_{\dot{x}}. \qquad \text{(EQ 4)}$$

The result of this weighted pooling is a pattern, s_x, called the sums. The sums may be further processed for other stages of analysis or for cascading memories. In this work, the sums are considered the output from the memory.

If x is an observation point with y_x then for perfect recall it is desired to have $s_x = y_x$.

3.4 Storage

The process of storage in a distributive memory is taken to be a weighted summing of information from observations conditionalized on the input pattern x'. The weighting is determined by an activation rule B (to be determined in terms of A) where

$$\dot{y}_{\dot{x}} = \dot{\Pi}_{\dot{x},\dot{x}} \sum_{x'} B_{\dot{x},x'} y_{x'} \qquad \text{(EQ 5)}$$

and

$$\dot{\Pi}_{\dot{x},\dot{x}} = \begin{cases} 1 & \dot{x} \text{ present,} \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(EQ 6)}$$

Each observation, $y_{x'}$, is weighted by the B activation rule and written into location $\dot{x}$ if it is present. The quantity $\dot{y}_{\dot{x}}$ accumulates these values.

3.5 Operators

The previous equations can be encapsulated concisely by using operator notation. By this I mean that for discrete spaces the quantities are considered vectors and matrices and for continuous spaces the quantities are functions with one or two arguments. Products are taken to be inner products (summed or integrated) over their appropriate spaces.

The values stored in memory are written as

$$\dot{y} = \dot{\Pi} B y \quad \text{(writing to memory)} \qquad \text{(EQ 7)}$$

and the values recalled from memory are written as

$$s = A\dot{y} \quad \text{(reading from memory)} \; . \qquad \text{(EQ 8)}$$

The sums, s, can now be expressed directly in terms of the observations, y, as

$$s = A\dot{\Pi} B y. \qquad \text{(EQ 9)}$$

3.6 Ensemble averages

The operator $\dot{\Pi}$ plays a fundamental role in actual implementation of associative memories for it specifies in address space (weight space) where hidden nodes are placed. If these locations are *randomly* chosen subject only to the constraint the *expected* number of them is fixed, then the expected behavior of the associative memory can be derived. Let p be the probability a specific address will exist in a uniformly randomly chosen memory. The expected value of $\dot{\Pi}$ is then just a multiple of the identity matrix

$$E(\dot{\Pi}) = pI \qquad \text{(EQ 10)}$$

for fixed positive scalar constant p. This means, over all models with randomly chosen hidden nodes, the expected sums for a fixed training set can be derived from the product of the activation rule and a memory with all possible locations present. Hence, the operator $\dot{\Pi}$ will not be considered in the remainder of this paper and the sums will be written as

$$\boxed{s = ABy.} \qquad \text{(EQ 11)}$$

The results that follow therefore hold for uniformly randomly chosen hidden nodes or dense memories where all possible hidden nodes are present.[2]

3.7 Identity transformations

If one could find a B such that AB were the identity transformation, then the sums, s, would be exactly the data, y, and recovery of data would occur without error. Note that reading at points x which were not sampled would yield the zero pattern. For systems that generalize this is not desirable, whereas for systems with perfect recall, it is.

I now turn to the motivation for considering inverses arising from the practice of using error correction in artificial neural networks.

2. Random placement of locations for the Selected-Coordinate Design has been shown (Danforth, 1990) to give good results for single-talker discrete-speech digit recognition.

4.0 Series expansion of inverse operators

A frequently used learning rule in artificial neural network research is back-propagation (Rumelhart, 1986). It repeatedly cycles the differences (errors) between desired output and actual output back into the network to adjust internal weights between layers of the network.[3] The form of associative memories considered in this paper have a single hidden layer where the weights between the input and hidden layer are fixed. Only the weights between hidden and output layer are adjustable. Therefore, knowledge of the derivative of the activation rule is not needed since propagation of errors through more than one layer is not used.

What I now show is when errors are used to cyclically adjust the weights in a system with one hidden layer where all possible hidden nodes are present, the resultant state of the memory is identical to that produced by a single-step inverse operator.

The following derivation assumes (in accord with current neural net literature) the activation rule used for recall from memory is the same one used for storage in memory.

Let $\dot{y}(t-1)$ be the data stored at location $\dot{x}$ (with $\dot{x}$ unrestricted) at time *t-1*. If one writes into this location the difference between the desired signal y and the sums generated by a read operation with activation rule A, the value of the data at time t will be given by

$$\dot{y}(t) = \dot{y}(t-1) + A^T e(t-1) \qquad \text{(EQ 12)}$$

where

$$e(t-1) = y - A\dot{y}(t-1). \qquad \text{(EQ 13)}$$

The operator A^T is the *write* operator and is the transpose of the *read* operator A (the transpose distinction is made here for purity and is usually not stipulated since most activation rules are symmetric). The write operator weights and pools the errors from distant patterns to a specific hidden location for modification. The read operator weights and pools all hidden locations to produce the output sums for comparison with the desired signal, y. Recall y is the whole data set. That is, y_x is the output pattern associated with input pattern x. If x does not occur in the data set then y_x is the zero pattern. Now (EQ 12) represents one epoch of training (note the error signal is buffered until all values are calculated; then every location is updated at once). Regrouping of terms gives

$$\dot{y}(t) = A^T y + (I - A^T A)\,\dot{y}(t-1). \qquad \text{(EQ 14)}$$

When carried to time *t=0*, this recursion yields the sum

$$\dot{y}(t) = \sum_{\tau=0}^{t-1} (I - A^T A)^{\tau} A^T y \qquad \text{(EQ 15)}$$

3. A similar technique is used by Prager (1989) to adjust the weights of an associative memory.

where it has been assumed the memory is filled with the zero pattern at time zero. In the limit of an infinite number of epochs the above series expansion is *formally* equivalent to

$$\dot{y} = \lim_{t \to \infty} \dot{y}(t), \qquad \text{(EQ 16)}$$

$$= \sum_{\tau=0}^{\infty} (I - A^T A)^{\tau} A^T y,$$

$$= [I - (I - A^T A)]^{-1} A^T y,$$

$$= [A^T A]^{-1} A^T y,$$

$$= A^{-1} y.$$

That is, *if* the series converges then it converges to the inverse of the operator. This formal argument reveals the relationship between writing errors into an associative memory and the use of an inverse operator for the learning rule. They are equivalent. This equivalence holds only when all possible hidden locations are present (no restriction on domain of $\dot{x}$)[4] and an infinite number of error-correction cycles is used. It indicates there exist strategies (inverse rules) that lead to very rapid (a single epoch) learning of specific patterns.

4. For sparse sampling and sparse memory there is an equivalent statement concerning the matrix derived from the activation rule evaluated on the lattice of observation points.

5.0 Spectral representation of nonlinear operators

Given an activation rule, A, for an associative memory, one wishes to find a second rule, B, which will act as the inverse of A. This is expressed as

$$\sum_{\dot{x}} A_{x,\dot{x}} B_{\dot{x},x'} = \delta_{x,x'} \qquad \text{(discrete spaces)} \qquad \text{(EQ 17)}$$

or

$$\int A(x,\dot{x}) B(\dot{x},x') d\dot{x} = \delta(x-x') \qquad \text{(continuous spaces).} \qquad \text{(EQ 18)}$$

The quantity δ is the Kronecker delta function for discrete spaces and is the Dirac delta function for continuous spaces. The operator, B, acts as the write-rule into memory and the operator A acts as the read-rule from memory. The intermediate states, $\dot{x}$, absorb information written to them in a distributed fashion and then, during reading, recombine that information.

The determination of the operator which cancels the effects of pooling can be determined for both discrete and continuous spaces in terms of the associated eigenvalue problem for the operator A. The problem is written

$$A\Psi = \Lambda\Psi \qquad \text{(eigenvalue problem)} \qquad \text{(EQ 19)}$$

where Ψ is the eigenvector associated with A and Λ is its eigenvalue[5]. For binary input spaces of dimension n, the number of components of Ψ is equal to 2^n. For continuous input spaces of dimension n, the eigenvalue problem is a homogeneous integral equation of the second kind (Smithies, 1962) and the eigenvectors are eigenfunctions of n parameters. I use eigenvector notation here with the understanding inner products can readily be interpreted as integrals for continuous spaces. Orthogonality is easily shown to hold for real symmetric operators A, (interchange of input pattern and address pattern leaves the value of activation unchanged).

$$\Psi_a^T \Psi_b = \delta_{a,b} \qquad \text{(orthonormal eigenvectors).} \qquad \text{(EQ 20)}$$

The superscript T stands for the transpose of the vector, and the quantities a,b specifies which member in the set of eigenvectors is under consideration.

If the eigenvectors form a complete set (basis), then an arbitrary vector (function) in the space can be written as a linear combination of them as

$$y = \sum_a c_a \Psi_a \qquad \text{(eigenvector expansion).} \qquad \text{(EQ 21)}$$

5. Uppercase lambdas are written for the full eigenvalues since they are usually the weighted cumulation of shell eigenvalues. Shell eigenvalues are written as lowercase lambdas (see section 5.2).

The identity operator is written as the sum over all outer products of the eigenvectors as

$$I = \sum_a \Psi_a \Psi_a^T. \qquad \text{(EQ 22)}$$

That this is true can be proved using the orthonormality of the eigenvectors, where

$$Iy = \left(\sum_a \Psi_a \Psi_a^T\right)\left(\sum_b c_b \Psi_b\right) = \sum_{a,b} c_b \Psi_a \Psi_a^T \Psi_b, \qquad \text{(EQ 23)}$$

$$= \sum_{a,b} c_b \Psi_a \delta_{a,b} = \sum_a c_a \Psi_a = y \qquad \text{(for all } y\text{)}.$$

It also can be shown the operator A is expressible in terms of Ψ and Λ as

$$A = \sum_a \Lambda_a \Psi_a \Psi_a^T \qquad \text{(EQ 24)}$$

since

$$A\Psi_b = \left(\sum_a \Lambda_a \Psi_a \Psi_a^T\right)\Psi_b = \sum_a \Lambda_a \Psi_a \delta_{a,b} = \Lambda_b \Psi_b. \qquad \text{(EQ 25)}$$

The inverse of A (when it exists) is now written as

$$A^{-1} = \sum_a \Lambda_a^{-1} \Psi_a \Psi_a^T \qquad \text{(EQ 26)}$$

since

$$AA^{-1} = \left(\sum_a \Lambda_a \Psi_a \Psi_a^T\right)\left(\sum_b \Lambda_b^{-1} \Psi_b \Psi_b^T\right) = \sum_{a,b} \Lambda_a \Lambda_b^{-1} \Psi_a \Psi_a^T \Psi_b \Psi_b^T, \qquad \text{(EQ 27)}$$

$$= \sum_a \Lambda_a \Lambda_a^{-1} \Psi_a \Psi_a^T = \sum_a \Psi_a \Psi_a^T = I.$$

5.1 Eigenvector interpretation

The eigenvalue problem is interesting unto itself for it specifies the *set* of patterns recalled without change by read operations (except for a constant multiplier Λ). It is the collective set of patterns that has this property and not just a single pattern $\Psi_{a,x}$ at x. If one were to write into memory a collection of patterns specified by the eigenvector Ψ_a then the distributed internal representation would simply be $\Lambda_a \Psi_a$. Reading from memory would retrieve the data multiplied by the eigenvalue squared. Therefore, there exist data sets that are *natural* to an associative memory. The sets, however, are highly specific and are unlikely to occur in any actual experimental environment. They do, however, form a basis for arbitrary data sets and it is this property that is exploited in determining the write-rule that is the inverse of the read-rule of the memory.

5.2 Radial kernels and shell operators

An activation rule dependent only on the distance between its arguments will be called a *radial* kernel. Radial kernels can be written as

$$A_{x,\dot{x}}(n) = f(d_n(x, \dot{x})) \qquad \text{(radial kernel).} \tag{EQ 28}$$

The function f is a scalar of one argument and d_n is a distance metric for vector arguments of n dimensions.

A radial kernel can be decomposed into a linear combination of more primitive *shell* operator $S(n, \rho)$. A shell operator is activated with unit value when its components are separated by a distance exactly equal to ρ. The shell operator is written as

$$S_{x,\dot{x}}(n, \rho) = \begin{cases} 1 & d_n(x, \dot{x}) = \rho, \\ 0 & \text{otherwise.} \end{cases} \tag{EQ 29}$$

A radial kernel is then simply a weighted sum of these shell operators with ρ ranging over its permissible values as

$$A(n) = \sum_{\rho} S(n, \rho) f(\rho). \tag{EQ 30}$$

If one can solve the eigenvalue problem for S and the eigenvectors are not a function of ρ (as will be shown to be the case) then the eigenvalues for a general radial activation rule will simply be a linear combination of the eigenvalues of the S operator weighted by the radial dependence function f

$$\Lambda(n) = \sum_{\rho} \lambda(n, \rho) f(\rho). \tag{EQ 31}$$

Since the eigenvectors of S are not a function of ρ (to be shown) it follows that the eigenvectors of S are also the eigenvectors of A and the task thereby reduces to solving the eigenvalue problem for S. The eigenvectors of S become the fundamental basis set for representing arbitrary memory configurations in all memories with radial activation rules.

5.3 Eigenvalue problem for shell operators

The task now at hand is to solve the following eigenvalue problem

$$S(n, \rho)\Psi(n, \rho) = \lambda(n, \rho)\Psi(n, \rho) \tag{EQ 32}$$

with S a shell operator defined by (EQ 29). The eigenvector Ψ has been explicitly tagged with n and ρ dependence. Now if $S(n, \rho)$ commutes with $S(n, \rho')$ for some other ρ' then

it is possible to find a *common* set of eigenvectors for both operators (Merzbacher, 1961). If this is true for any two values of ρ it is true for all values of ρ and there is a single common set of eigenvectors independent of ρ.

5.4 Proof shell operators commute

Let S' be a shell operator with activation radius ρ'. Then S and S' are said to *commute* if

$$SS' = S'S \tag{EQ 33}$$

or

$$\sum_u S_{x,u}(n,\rho)S_{u,x'}(n,\rho') = \sum_v S_{x,v}(n,\rho')S_{v,x'}(n,\rho). \tag{EQ 34}$$

From the definition of S this is equivalent to requiring

$$\sum_u I\,[d_n(x,u)=\rho \;\&\; d_n(u,x')=\rho'] = \sum_v I\,[d_n(x,v)=\rho' \;\&\; d_n(v,x')=\rho] \tag{EQ 35}$$

where I is the *indicator* function which equals one if its argument is true and is zero otherwise. The indicator function on the left-hand-side will be one for those points u which are ρ distant from x *and* ρ' distant from x'. Does there always exist a v (for every u) that is ρ' distant from x and ρ distant from x'? If so then the above equation will be true. The one-to-one transformation[6]

$$u+v = x+x' \tag{EQ 36}$$

satisfies this requirement since

$$d_n(x,v) = \|x-v\| = \|x-(x+x'-u)\| = \|u-x'\| = d_n(u,x') \tag{EQ 37}$$

and

$$d_n(v,x') = \|v-x'\| = \|(x+x'-u)-x'\| = \|x-u\| = d_n(x,u). \tag{EQ 38}$$

Therefore, it has been shown the S operators commute and the eigenvalue problem can be written as

$$S(n,\rho)\Psi(n) = \lambda(n,\rho)\Psi(n) \tag{EQ 39}$$

with a common set of eigenvectors $\Psi(n)$ independent of ρ.

6. It should be noted boundary conditions on the space can make it impossible to satisfy this symmetry constraint. It is assumed here this is not the case.

6.0 The Kanerva model

Pentti Kanerva (1988) has presented a sparse distributed memory model of associative memories that uses patterns composed of long bit strings. He derives many interesting results from this abstract model which is based on the standard random access memory of present day computers. The Kanerva model is specified by an activation rule (kernel) of the form

$$A_{x,\dot{x}}(n,r) = \begin{cases} 1 & d_n(x,\dot{x}) \le r, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{(EQ 40)}$$

$$\text{for} \quad x, \dot{x} \in \{0,1\}^n.$$

The quantity r is a free parameter of the model and is called the activation radius. The metric, d_n, is the Hamming distance between patterns x and $\dot{x}$ where both are binary valued vectors. The weights, $\dot{x}$, are fixed and are called the *address* of a location (hidden node). This activation rule is a radial kernel where the radial dependence function f is a step function. It is written

$$f(\rho, r) = \begin{cases} 1 & \rho \le r, \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(EQ 41)}$$

This rule has a sharp discontinuity at distance r between x and $\dot{x}$. The rule is not differentiable and so gradient descent methods which rely on first derivatives of the activation rule for error minimization can not be applied. This does not mean error signals can not be used to modify the contents of memory (see for example Prager, 1989) only that gradient techniques can not be used. One might be lead to believe this "hard thresholding" activation rule would make it impossible to find an inverse write rule, however, this is not the case as I now show. The decomposition of radial kernels into sums of shell operators can now be specialized to the Kanerva model as

$$A(n,r) = \sum_{\rho=0}^{n} f(\rho, r) S(n, \rho). \qquad \text{(EQ 42)}$$

6.1 Eigenvalue problem for the Kanerva model

The task reduces to solving the eigenvalue problem for a shell that uses *Hamming* distance between *binary* vectors.

$$S_{x,\dot{x}}(n,\rho) = I\left[\sum_{i=1}^{n} d(x_i, \dot{x}_i) = \rho\right] \qquad \text{(EQ 43)}$$

where x_i, $\dot{x}_i$ are bits and d is one if they differ and is zero if they are the same. A recursive decomposition can now be applied to $S(n, \rho)$ in order to represent it in terms of $S(n-1, \rho)$. With this decomposition it is possible derive a set of recurrence relations for the eigenvalues and eigenvectors for S. Now

$$S_{x,\dot{x}}(n, \rho) = I\left[d(x_n, \dot{x}_n) + \sum_{i=1}^{n-1} d(x_i, \dot{x}_i) = \rho. \right] \qquad \text{(EQ 44)}$$

If the nth components x_n, $\dot{x}_n$ are equal then evaluation of the indicator function falls upon the evaluation of the second term which is identical to $S(n-1, \rho)$. If the nth components differ then evaluation of the indicator functions falls again on the second term but this time with reduced activation radius equal to $\rho - 1$. These results can be summarized by partitioning the shell operator for n dimensions into four parts, one for each of the possible x_n, $\dot{x}_n$ combinations, namely

$$S(n, \rho) = \begin{bmatrix} S(n-1, \rho) & S(n-1, \rho-1) \\ S(n-1, \rho-1) & S(n-1, \rho) \end{bmatrix}. \qquad \text{(EQ 45)}$$

The dimensions of the operator $S(n-1, \rho)$ are 2^{n-1} by 2^{n-1} and grow by a factor of two as n increases by one. If one could establish the relationship between the solutions of the eigenvalue problem for $S(n, \rho)$ in terms of those for $S(n-1, \rho)$ and knowing the solutions for $n=0$, one would have the solutions for all n. This indeed is what will be done. I now state that an eigenvector for n dimensions is built up from a partition of eigenvectors of $n-1$ dimensions as

$$\Psi_{\pm}(n) = \frac{1}{\sqrt{2}} \begin{pmatrix} \Psi(n-1) \\ \pm\Psi(n-1) \end{pmatrix}. \qquad \text{(EQ 46)}$$

Note normalization and orthogonality hold for n dimensions if they hold for $n-1$ dimensions, since

$$\Psi^T_{\pm}(n)\Psi_{\pm}(n) = \Psi^T(n-1)\Psi(n-1) = 1, \qquad \text{(EQ 47)}$$

$$\Psi^T_{\pm}(n)\Psi_{\mp}(n) = 0.$$

Completeness also follows if the set of eigenvectors for $n-1$ dimensions is complete. To see this note there are 2^{n-1} orthonormal eigenvectors for a binary space of $n-1$ dimensions. For each one of these, two are created for n dimensions; one which is constructed by the concatenation of equal vectors with a plus sign (and weighted by the inverse of $\sqrt{2}$) and another which is constructed by a concatenation with a minus sign. Each of these new 2^n vectors is orthonormal (see above) and so the set spans the space. It remains to show $\Psi(n)$, as given by (EQ 46), is indeed an eigenvector of the shell operator $S(n, \rho)$ as given by (EQ 45). The eigenvalue problem may now be written as

$$\begin{bmatrix} S(n-1,\rho) & S(n-1,\rho-1) \\ S(n-1,\rho-1) & S(n-1,\rho) \end{bmatrix} \begin{pmatrix} \Psi(n-1) \\ \pm\Psi(n-1) \end{pmatrix} = \lambda_{\pm}(n,\rho) \begin{pmatrix} \Psi(n-1) \\ \pm\Psi(n-1) \end{pmatrix}. \quad \text{(EQ 48)}$$

One can quickly see that the operator applied to the partitions yields a multiple of the full vector since the vectors for *n-1* dimensions are by definition eigenvectors of the operators for *n-1* dimensions. The upper and lower partition yield the same relationship between the eigenvalue for *n* dimensions in terms of those for *n-1* dimensions, namely

$$\lambda_{\pm}(n,\rho) = \lambda(n-1,\rho) \pm \lambda(n-1,\rho-1). \quad \text{(EQ 49)}$$

Now (EQ 46) and (EQ 49) are the desired recurrence relations for the eigenvectors and eigenvalues of the Kanerva model. To derive a closed form expression for them begin by defining the eigenvector for *n=0* (no bits) as equal to one, hence

$$\Psi(0) \equiv 1. \quad \text{(EQ 50)}$$

It then follows

$$\Psi_{+}(1) = \frac{1}{\sqrt{2}} \begin{pmatrix} +1 \\ +1 \end{pmatrix} \quad \text{(EQ 51)}$$

and

$$\Psi_{-}(1) = \frac{1}{\sqrt{2}} \begin{pmatrix} +1 \\ -1 \end{pmatrix}. \quad \text{(EQ 52)}$$

One can see as *n* increases, vectors will be composed of strings of alternating signed 1's. Let's continue this process for one more step to help reveal an analytic expression of a vector's components:

$$\Psi_{++}(2) = \frac{1}{2} \begin{pmatrix} +1 \\ +1 \\ +1 \\ +1 \end{pmatrix}, \quad \text{(EQ 53)}$$

$$\Psi_{+-}(2) = \frac{1}{2} \begin{pmatrix} +1 \\ -1 \\ +1 \\ -1 \end{pmatrix}, \quad \text{(EQ 54)}$$

$$\Psi_{-+}(2) = \frac{1}{2} \begin{pmatrix} +1 \\ +1 \\ -1 \\ -1 \end{pmatrix}, \quad \text{(EQ 55)}$$

$$\Psi_{-}(2) = \frac{1}{2}\begin{pmatrix} +1 \\ -1 \\ -1 \\ +1 \end{pmatrix}. \qquad \text{(EQ 56)}$$

The analytic form that satisfies the vector expressions is

$$\boxed{\Psi_{\alpha,x}(n) = \frac{1}{\sqrt{2^n}}(-1)^{\alpha^T x}} \quad \text{(Kanerva model eigenvectors).} \qquad \text{(EQ 57)}$$

The label α distinquishes different eigenvectors. It is a bit string residing in the same *n*-dimensional space as that of x (the input bit pattern space). The quantity $\alpha^T x$ is the inner product between the bit vectors α and x. It is the sum of the logical *AND* of the bits. The (-1) to this power retains only the information of whether this inner product has even or odd parity.

To see (EQ 57) satisfies (EQ 46), let α_n and x_n be single bits (the nth bit of a bit vector of length n) and α', x' bit vectors of length *n-1* then:

$$\Psi_{\alpha_n\alpha',x_nx'}(n) = \frac{1}{\sqrt{2^n}}\begin{bmatrix} (-1)^{(\alpha_n \cdot 0 + \alpha'^T x')} \\ (-1)^{(\alpha_n \cdot 1 + \alpha'^T x')} \end{bmatrix}, \qquad \text{(EQ 58)}$$

$$\Psi_{\alpha_n\alpha',x_nx'}(n) = \frac{1}{\sqrt{2}}\begin{bmatrix} \frac{1}{\sqrt{2^{n-1}}}(-1)^{(\alpha'^T x')} \\ (-1)^{\alpha_n}\frac{1}{\sqrt{2^{n-1}}}(-1)^{(\alpha'^T x')} \end{bmatrix}, \qquad \text{(EQ 59)}$$

$$\Psi_{\alpha_n\alpha',x_nx'}(n) = \frac{1}{\sqrt{2}}\begin{bmatrix} \Psi_{\alpha',x'}(n-1) \\ (-1)^{\alpha_n}\Psi_{\alpha',x'}(n-1) \end{bmatrix}. \qquad \text{(EQ 60)}$$

As α_n flips between zero and one the lower partition's sign changes between +1 and -1 thereby showing (EQ 57) does indeed satisfy the recurrence relation for the shell eigenvectors.

The eigenvectors that have been found to satisfy the shell operator for binary spaces are not new. In fact these functions were investigated by Hadamard and now bare his name (Harwit, 1979). For each α there is a Hadamard function Ψ_α the collection of which form a basis for the binary space. These functions are the eigenvectors of all activation rules that depend only radially on the binary weights (address) of a hidden node and the binary input pattern. I now turn to the issue of an explicit form for the eigenvalues. Since each ei-

genvector is labeled by a bit string α it is also necessary to do so for each eigenvalue. Therefore,

$$\lambda_\alpha(n, \rho) = \lambda_{\alpha'}(n-1, \rho) + (-1)^{\alpha_n}\lambda_{\alpha'}(n-1, \rho-1) \qquad \text{(EQ 61)}$$

for α_n a bit and α' a bit-string of length *n-1*. The boundary conditions for λ are chosen to be zero for ρ outside of the interval [0,*n*] of possible distance values. Hence

$$\lambda_\alpha(n, \rho) = 0, \qquad \rho \notin [0, n]. \qquad \text{(EQ 62)}$$

To solve (EQ 61) it is expedient to transform it into a generating function by multiplying by an arbitrary parameter, *t*, raised to the power of ρ and then summing over all ρ from zero to *n*. Let $g_\alpha(n, t)$ be such a generating function expressed as

$$g_\alpha(n, t) \equiv \sum_{\rho=0}^{n} \lambda_\alpha(n, \rho) t^\rho. \qquad \text{(EQ 63)}$$

It follows (using the above boundary conditions) that:

$$g_\alpha(n, t) = g_{\alpha'}(n-1, t) + (-1)^{\alpha_n} t g_{\alpha'}(n-1, t), \qquad \text{(EQ 64)}$$

$$g_\alpha(n, t) = [1 + (-1)^{\alpha_n} t]\, g_{\alpha'}(n-1, t), \qquad \text{(EQ 65)}$$

$$g_\alpha(n, t) = \prod_{i=1}^{n} [1 + (-1)^{\alpha_i} t], \qquad \text{(EQ 66)}$$

and so finally,

$$g_\alpha(n, t) = (1-t)^{\|\alpha\|} (1+t)^{n-\|\alpha\|} \qquad \text{(EQ 67)}$$

where $\|\alpha\|$ is the Hamming weight of α (the number of bits equal to 1 in the bit-string). The beauty of the generating function representation is that it reveals the order-independence of the bits in the eigenvalue label α. The generating function can be used to determine many interesting properties of the eigenvalues. An explicit expression for the eigenvalues can now be obtained by expanding the factors (EQ 67) in powers of *t*, collecting like powers, and equating the coefficients of these powers with the eigenvalues. A set of expansion coefficients, *C*, are defined here which will be useful later in the expression of the inverse Kanverva model. They are

$$C_\rho(a, b) = \sum_{k=0}^{\rho} (-1)^k \binom{a}{\rho - k} \binom{b}{k}. \qquad \text{(EQ 68)}$$

These coefficients have an alternating sign in front of the binomial coefficients and are closely related to the hypergeometric distribution (Feller, 1968). The shell eigenvalues can now be written as

$$\lambda_\alpha(n, \rho) = C_\rho(n - \|\alpha\|, \|\alpha\|). \quad \text{(EQ 69)}$$

So, the eigenvalue problem for a shell operator with binary valued arguments and Hamming distance activation has been solved. The eigenvectors for the Kanerva model are the same as these eigenvectors. The eigenvalues are the sum of the shell eigenvalues up to the activation radius, r.

$$\Lambda_\alpha(n, r) = \sum_{\rho=0}^{r} \lambda_\alpha(n, \rho), \quad \text{(EQ 70)}$$

$$\boxed{\Lambda_\alpha(n, r) = \sum_{\rho=0}^{r} \sum_{k=0}^{\rho} (-1)^k \binom{n - \|\alpha\|}{\rho - k} \binom{\|\alpha\|}{k}} \quad \text{(Kanerva model eigenvalues).} \quad \text{(EQ 71)}$$

6.2 Inverse of the Kanerva model

Having solved the eigenvalue problem for the Kanerva model there still remains the issue of whether it has an inverse. This is equivalent to asking whether any of the eigenvalues are zero. Interestingly, the answer to this question is determined by whether the *parity* of a binomial coefficient is even or odd. If it can be shown that the parity of an eigenvalue is odd then that eigenvalue can not be the number zero. The recursion relation for the shell operator can be used to show the parity of shell eigenvalues is not a function of the eigenvalue label α. Let π be a parity operator defined as

$$\pi(a) = \begin{cases} +1 & a \text{ even}, \\ -1 & a \text{ odd}. \end{cases} \quad \text{(EQ 72)}$$

With this operator the parity of the sum of two numbers is the product of the parity of the numbers and the parity of the negative of a number is just the parity of the number, so:

$$\pi(a + b) = \pi(a)\pi(b), \quad \text{(EQ 73)}$$

$$\pi(-a) = \pi(a). \quad \text{(EQ 74)}$$

The parity of the recurrence relation of the shell eigenvalues then becomes:

$$\pi(\lambda_\alpha(n,\rho)) = \pi(\lambda_{\alpha'}(n-1,\rho) + (-1)^{\alpha_n}\lambda_{\alpha'}(n-1,\rho-1)), \quad \text{(EQ 75)}$$

$$= \pi(\lambda_{\alpha'}(n-1,\rho))\pi((-1)^{\alpha_n}\lambda_{\alpha'}(n-1,\rho-1)),$$

$$= \pi(\lambda_{\alpha'}(n-1,\rho))\pi(\lambda_{\alpha'}(n-1,\rho-1)).$$

It can be seen the parity for n dimensions is not a function of the nth bit of α. Since this is expressed recursively in terms of the parity for n-1 dimensions it follows that the parity is not a function of any of the bits of α. As such, knowledge of the parity of $\lambda_\alpha(n,\rho)$ for one value of α yields the parity for all values of α. Choose α equal to the string of all 0's with Hamming weight zero, $\|\alpha\| = 0$, then the eigenvalue for this α reduces to a binomial coefficient expressed as

$$\lambda_0(n,\rho) = \binom{n}{\rho}. \quad \text{(EQ 76)}$$

In summary, the parity of a shell eigenvalue with label α is the same as the parity of the eigenvalue with label 0 which is the same as the parity of a binomial coefficient, so

$$\pi(\lambda_\alpha(n,\rho)) = \pi(\lambda_0(n,\rho)) = \pi\left(\binom{n}{\rho}\right). \quad \text{(EQ 77)}$$

The parity of the binomial coefficients for n up to 16 is presented in the following diagram. It is noticed for $n=2^k-1$ (e.g. 0,1,3,7,15, etc.) the parity is constant for all ρ and equal to -1. It also appears that the diagram is self recursive; the top triangle down to 2^k-1 is copied below itself to the left and to the right with the intermediate spaces filled with +'s.

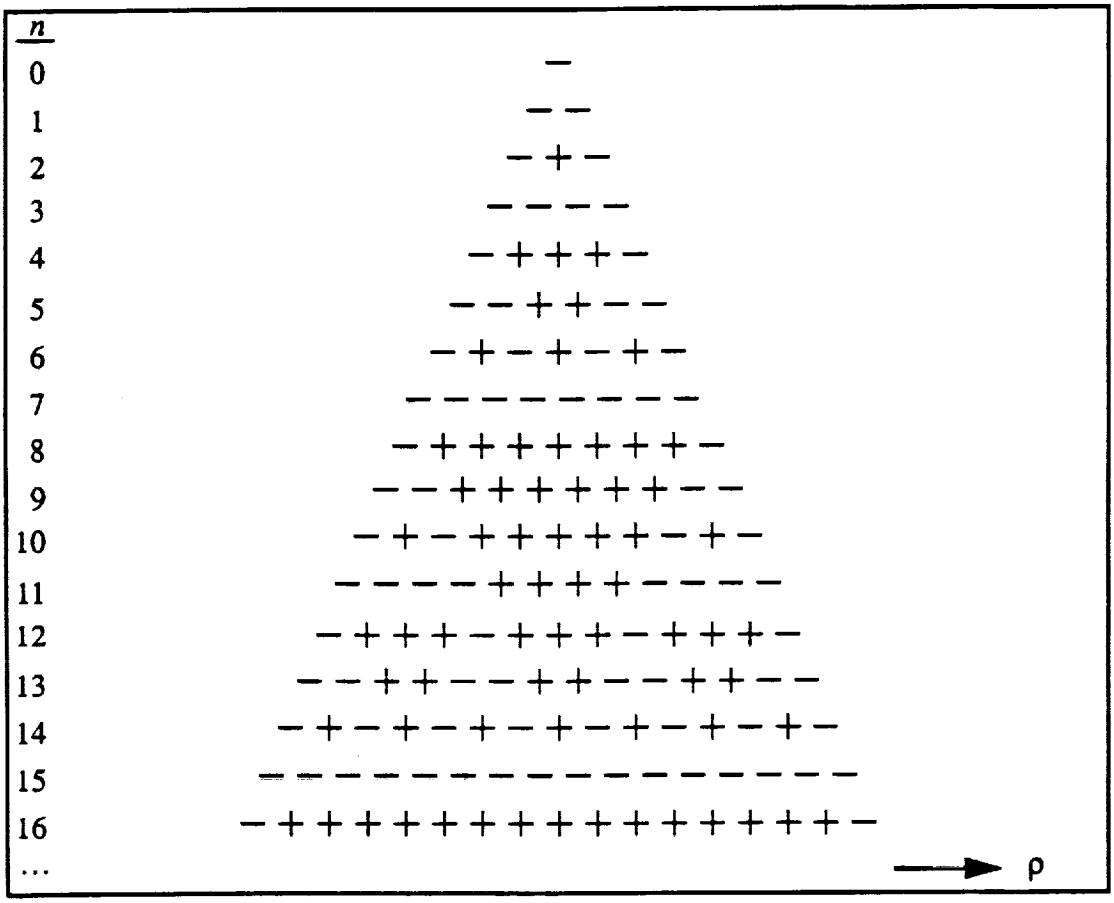


Figure 6 Parity table for binomial coefficients.

It is speculated the parity of a binomial coefficient is given by the rule

$$\pi\left(\binom{n}{\rho}\right) = \begin{cases} -1 & \bar{n}_B \wedge \rho_B = 0, \\ +1 & \text{otherwise} \end{cases} \qquad \text{(EQ 78)}$$

where ρ_B is the base-2 bit-vector representation of ρ, $\bar{n}_B$ is the bit compliment of the base-2 representation of n, $\wedge$ is the logical AND of bits and 0 is a bit string of 0's (all representations are considered padded with sufficient bits to represent the largest number). For example, n=12, ρ=4 implies n_B=1100, $\bar{n}_B$=0011, ρ_B=0100, and $\bar{n}_B \wedge \rho_B$=0000. So the parity rule says 12 choose 4 should have odd parity. Looking back at the parity diagram it can be seen this is true. If this rule is true then when n is one less than a power of two the binary representation of n has all one-bits and since ρ is less than or equal to n the sets $\bar{n}_B$ and ρ_B are disjoint and their logical AND is zero for all bits. Now the Kanerva model makes use of the sum of the first r shell eigenvalues. The parity of this sum is the product of the parity of individual eigenvalues which is the product of the parity of the corresponding binomial coefficients. The parity of the sum of the binomial coefficients is

equal to this product and so can be used to evaluate the parity of the sum of the shell eigenvalues.

$$\pi(\Lambda_\alpha(n, r)) = \pi(\sum_{\rho=0}^{r} \lambda_\alpha(n, \rho)), \qquad \text{(EQ 79)}$$

$$= \prod_{\rho=0}^{r} \pi(\lambda_\alpha(n, \rho)),$$

$$= \prod_{\rho=0}^{r} \pi(\binom{n}{\rho}),$$

$$= \pi(\sum_{\rho=0}^{r} \binom{n}{\rho})$$

which is the parity of the *cumulative* binomial distribution. The parity table for the cumulative binomial coefficients is given in the next diagram.

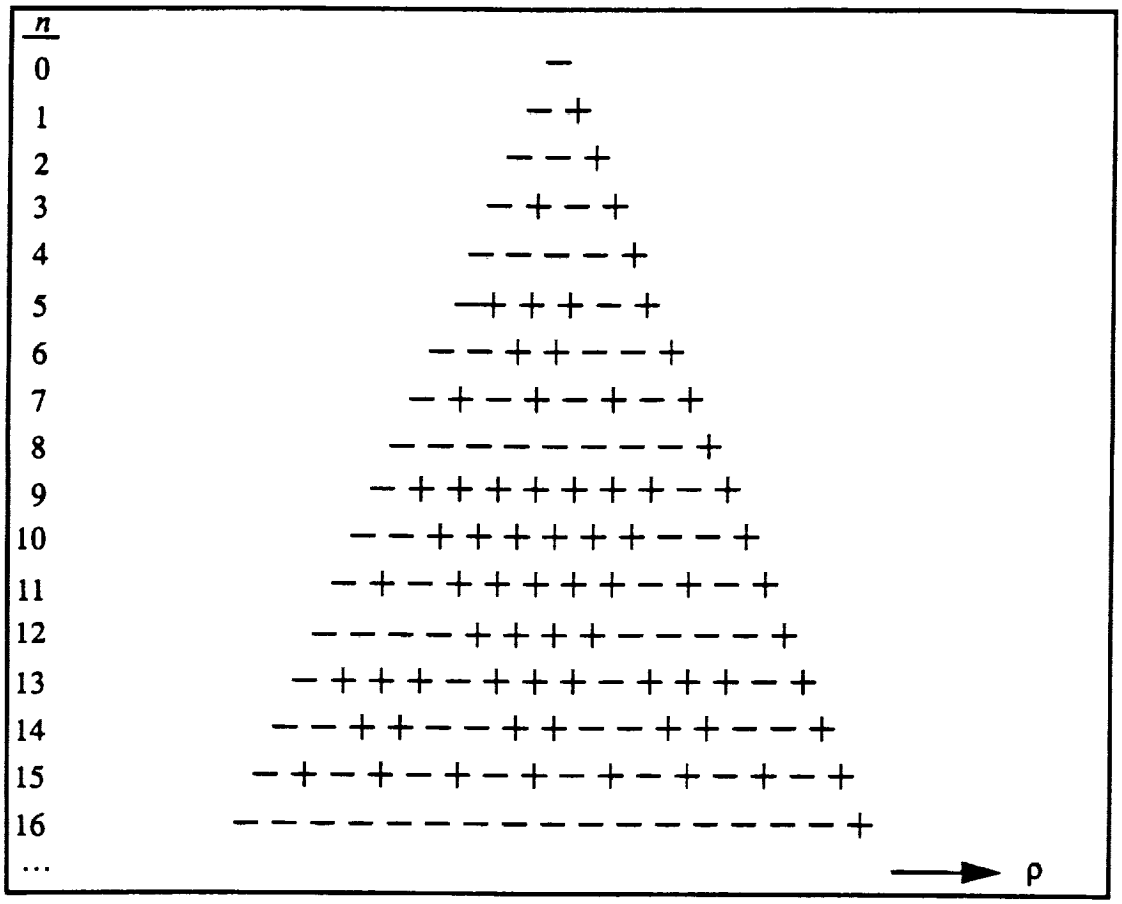


Figure 7 Parity table for cumulative binomial coefficients.

It can be seen for $n=2^k$ (e.g., 1,2,4,8,16, etc.) the parity is odd for all coefficients accept for the last entry where $r=n$. As such, no eigenvalue for the Kanerva model can be zero for any activation radius less than n for n a power of two. It is now possible to state the following theorem for the Kanerva model:

> **If the dimensionality of the input space, n, is a power of two *and* the activation radius, r, is less than n then the Kanerva model has an inverse.**

There are other combinations of n and r for which the Kanerva model has an inverse, however, they are not as easily specified.

Knowing the Kanerva model has an inverse we are in a position to derive an explicit expression for it. Recall that an arbitrary invertible operator can have its inverse expressed as a linear combination of the outer products of its eigenvectors with the inverse of its eigenvalues. If A^{-1} is the inverse of the Kanerva model then

$$A^{-1}(n, r) = \sum_{\alpha} \Lambda_{\alpha}^{-1}(n, r) \Psi_{\alpha} \Psi_{\alpha}^{T}. \qquad \text{(EQ 80)}$$

That the Kanerva activation rule is only a function of the distance between its elements would lead one to suppose its inverse would also have this property. This turns out to be true and can be demonstrated by breaking the sum over α into two sums, one over $w = \|\alpha\|$ and the other over all labels with fixed Hamming weight $\|\alpha\|$.

$$A^{-1}_{\dot{x}, x}(n, r) = \sum_{\alpha} \frac{1}{\sum_{\rho=0}^{r} C_{\rho}(n - \|\alpha\|, \|\alpha\|)} \frac{(-1)^{\alpha^T \dot{x}}}{2^{n/2}} \frac{(-1)^{\alpha^T x}}{2^{n/2}}, \qquad \text{(EQ 81)}$$

$$= 2^{-n} \sum_{\alpha} \frac{(-1)^{\alpha^T (\dot{x} + x)}}{\sum_{\rho=0}^{r} C_{\rho}(n - \|\alpha\|, \|\alpha\|)},$$

$$= 2^{-n} \sum_{w=0}^{n} \frac{\sum_{\alpha | \, \|\alpha\| = w} (-1)^{\alpha^T (\dot{x} + x)}}{\sum_{\rho=0}^{r} C_{\rho}(n - w, w)}.$$

The expression in the numerator of the last equation, interestingly, can be represented in terms of the expansion coefficients, C, and the hamming distance, d, between the vectors $\dot{x}$ and x. Now

$$\sum_{\alpha | \, \|\alpha\| = w} (-1)^{\alpha^T (\dot{x} + x)} = \sum_{\alpha | \, \|\alpha\| = w} \prod_{i=1}^{n} (-1)^{\alpha_i (\dot{x}_i + x_i)}, \qquad \text{(EQ 82)}$$

$$= \sum_{\alpha | \, \|\alpha\| = w} \prod_{i=1}^{n} (-1)^{\alpha_i (\dot{x}_i - x_i)}.$$

If a component of $\dot{x}$ and x are equal then their sum is either zero or two. In either case this sum times the component of α is even. The quantity (-1) to an even power is 1 therefore components of $\dot{x}$ and x which are equal do not affect the overall product. Only components that differ have an affect. It follows one may replace the sum of $\dot{x}$ and x by their difference in (EQ 82)and retain the same value for the product.

Now $\dot{x}$ and x differ on d bits (define the number of bit differences as d). Since we are summing over all possible vectors α with w one-bits it does not matter where in the x vectors the differences occur therefore we may conceptually move them to the beginning of the x vectors.

$$|\dot{x} - x| = \underbrace{\overbrace{11\ldots1}^{d}\,00\ldots0}_{n} . \qquad \text{(EQ 83)}$$

The sum over α can be decomposed into two parts where k bits of α fall in the 1's region and w-k bits fall in the 0's region. There are d choose k ways this can happen for the first case and n-d choose w-k ways this can happen in the second case. The first case contributes $(-1)^k$ and the second contributes unity to the overall product. One then sees

$$\sum_{\alpha \mid \|\alpha\| = w} \prod_{i=1}^{n} (-1)^{\alpha_i (\dot{x}_i - x_i)} = \sum_{k=0}^{w} (-1)^k \binom{n-d}{w-k}\binom{d}{k}, \qquad \text{(EQ 84)}$$

$$= C_w(n-d, d).$$

It can be shown that

$$\binom{n}{w} C_d(n-w, w) = \binom{n}{d} C_w(n-d, d). \qquad \text{(EQ 85)}$$

Combining these results a final expression for the inverse of the Kanverva model is obtained.[7]

$$A^{-1}_{\dot{x},x}(n, r) = 2^{-n} \sum_{w=0}^{n} \binom{n}{w} \frac{C_d(n-w, w)}{\binom{n}{d} \sum_{\rho=0}^{r} C_\rho(n-w, w)}, \quad \text{(inverse of Kanerva model)} \qquad \text{(EQ 86)}$$

where

$$d = d_n(\dot{x}, x). \qquad \text{(EQ 87)}$$

The factors in this inverse expression can be given some interpreted when one considers the inverse's behavior with the Kanerva operator along the diagonal which we know should be equal to 1. The way the terms collapse show the interdependence of the parts of the expression.

7. The complexity of this expression caused me to seek verification of its correctness via nonanalytic means. I programmed the function and for small values of n (up to eight) found that indeed the product of the Kanerva kernel with its inverse gives the identity matrix.

$$(A^{-1}A)_{x,x}(n,r) = \sum_{x'} A^{-1}_{x,x'}(n,r)A_{x',x}(n,r), \qquad \text{(EQ 88)}$$

$$= \sum_{d=0}^{n} \sum_{x' \mid d_n(x,x')=d} A^{-1}_{x,x'}(n,r)A_{x',x}(n,r),$$

$$= \sum_{d=0}^{n} A^{-1}_{x,x'}(n,r)A_{x',x}(n,r) \sum_{x' \mid d_n(x,x')=d} 1,$$

$$= \sum_{d=0}^{n} \binom{n}{d} A^{-1}_{x,x'}(n,r)A_{x',x}(n,r),$$

$$= \sum_{d=0}^{r} \binom{n}{d} A^{-1}_{x,x'}(n,r),$$

$$= 2^{-n} \sum_{w=0}^{n} \binom{n}{w} \frac{\sum_{d=0}^{r} C_d(n-w,w)}{\sum_{\rho=0}^{r} C_\rho(n-w,w)},$$

$$= 2^{-n} \sum_{w=0}^{n} \binom{n}{w},$$

$$= 1.$$

To develope a sense of what the inverse looks like, plots of the activation rule and its inverse for n=8 are presented below (the thin line is the activation rule for the Kanerva model and is not in the same scale as the thicker inverse rule):

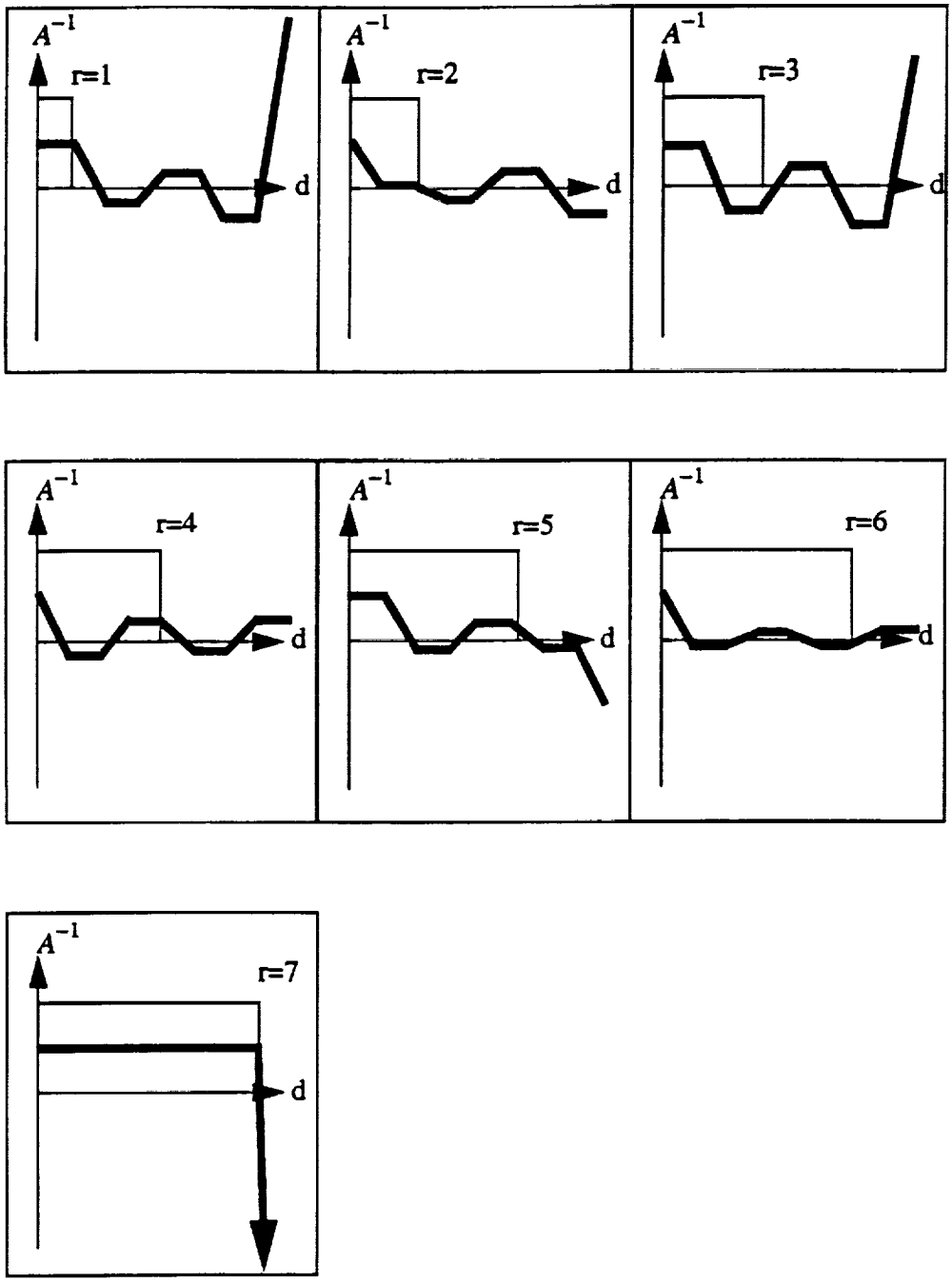


Figure 8 Radial dependence of the inverse of the Kanerva model, n=8.

For $r=0$ the inverse is just the identity transformation and this plot is not presented. For $r=n$, it was shown no inverse exists (all points in the space receive exactly the same information and there is no way to recover the original data). The plots show the form of the distance dependence of the inverse operator (write-rule) for $r=1$ to $r=n-1$ ($n=8$). It can be seen there is a general oscillatory behavior of the inverse. Some regions enfold data with a positive sign and some regions enfold it with a negative sign.

Figure 9 depicts reading at a location (indicated by the uniformly-shaded gray disc) which is offset from the center of a write operation. The sum over the disc of the inverse-weighted data is zero.The magnitude and amplitude of the inverse rule is such that when information is recalled from memory, using the hard thresholding rule of the Kanerva model, competing patterns around the read point cancel exactly to leave only the data written at the point.

The inverse of a activation rule has been obtained. The rule is quite simple and yet its inverse is not. The oscillatory behavior is reminiscent of the Fourier transform of a step function which is a damped sinusoid. In the inverse case, however, the oscillations need not decrease with increasing distance. This concludes the discussion of the Kanerva model.

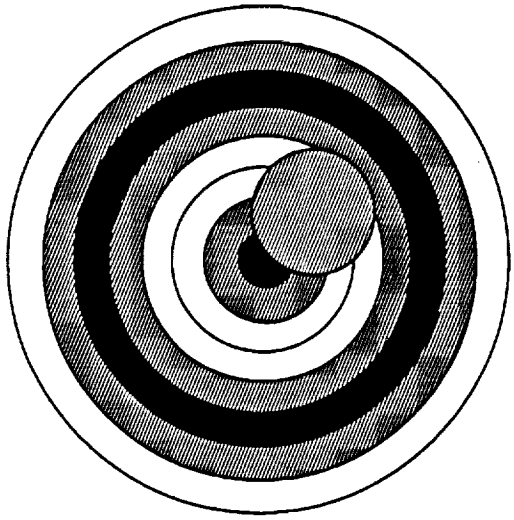


Figure 9 Reading off-set from a write point.

7.0 The Selected-Coordinate Design

Louis Jaeckel (1989) has put forth an alternative design to the Sparse Distributed Memory of Pentti Kanerva. The design is in the same spirit as Kanerva's model, however, there are some major differences. In the Selected-Coordinate Design (SCD) a hidden location will be activated only if there is an exact match between the input and its ternary valued weights. Ternary in the sense *don't care* values are introduced. With don't care values attention to the input is restricted to a subset of *selected coordinates*. Each hidden location has its own subset which are randomly chosen and have random bit values. If an input pattern matches the values of the selected coordinates exactly then the hidden location is activated. For notation convenience I now treat a "bit" as an element of the set $\{-1,+1\}$ and a don't care value as the quantity "0". The activation rule for the Jaeckel model can be specified as

$$A_{x,\dot{x}}(n) = \begin{cases} 1 & x_i\dot{x}_i \geq 0,\ i = 1, 2, \ldots, n, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{(EQ 89)}$$

$$\text{for } x, \dot{x} \in \{-1, 0, 1\}^n.$$

If x and $\dot{x}$ differ on a bit then one of them must be a don't care value (0). Notice I have extended the range of *input*, x, to also include the value zero. This is for symmetry and allows a ready solution to the eigenvalue problem. A zero value for input can be considered a "don't know" bit. If all input is known then a zero value will never occur. A don't know matches any -1,0, or +1 value.

7.1 Eigenvalue problem for the Selected-Coordinate Design

The exact match criterion of the Selected-Coordinate Design allows a factoring of the activation rule into a product of indicator functions. This factoring immediately reveals the eigenvalue problem can be solved as a product of functions. Each function is a 3x1 vector solution of a 3x3 matrix, whose eigenvalues are readily calculated. I now go through the steps in obtaining the eigenvectors and eigenvalues for the Selected-Coordinate Design.

Write the activation function as

$$A_{x,\dot{x}}(n) = \prod_{i=1}^{n} I\,[x_i\dot{x}_i \geq 0]\,. \qquad \text{(EQ 90)}$$

Each factor in the product expression has the same form and is the activation rule for a single bit. This one-bit activation rule can be specified as a 3x3 matrix indicating whether activation will occur for each of the three possible hidden states of the first bit and each of the three possible states of the input for the first bit. Let $A(1)$ be this 3x3 matrix then

PRECEDING PAGE BLANK NOT FILMED

$$A(1) = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \qquad \text{(EQ 91)}$$

That is, activation will not occur for $x\dot{x} = -1$. The eigenvalue problem for $A(1)$ is solved by the eigenvector ψ with eigenvalue λ. The problem is specified as

$$A(1)\psi = \lambda\psi \qquad \text{(EQ 92)}$$

and has nontrivial solutions only if the determinant

$$\begin{vmatrix} 1-\lambda & 1 & 0 \\ 1 & 1-\lambda & 1 \\ 0 & 1 & 1-\lambda \end{vmatrix} \qquad \text{(EQ 93)}$$

is zero. This leads to the cubic equation in λ

$$(1-\lambda)^3 - 2(1-\lambda) = 0, \qquad \text{(EQ 94)}$$

which has the three solutions

$$\lambda_a = 1 + a\sqrt{2} \text{ for } a = -1, 0, 1. \qquad \text{(EQ 95)}$$

For each a there is a corresponding three component eigenvector ψ_a given by

$$\psi_a = \frac{1}{\sqrt{2+2a^2}} \begin{bmatrix} 1 \\ a\sqrt{2} \\ 2a^2 - 1 \end{bmatrix}. \qquad \text{(EQ 96)}$$

These three eigenvectors can also be grouped and displayed in the symmetric 3x3 matrix ψ where

$$\psi = \begin{bmatrix} \frac{1}{2} & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} \\ \frac{1}{2} & \frac{-1}{\sqrt{2}} & \frac{1}{2} \end{bmatrix}. \qquad \text{(EQ 97)}$$

The full eigenvector for the Selected-Coordinate Design is now given as the product of elements of ψ, namely

$$\Psi_{\alpha, x}(n) = \prod_{i=1}^{n} \psi_{\alpha_i, x_i}, \tag{EQ 98}$$

$$\alpha_i \in \{-1, 0, 1\} \tag{EQ 99}$$

where α is a "bit" string acting as a label for a specific eigenvector. There are 3^n such eigenvectors. The eigenvalues Λ_α for the full n-bit space are the product of the one-bit eigenvalues λ, namely

$$\Lambda_\alpha = \prod_{i=1}^{n} \lambda_{\alpha_i}. \tag{EQ 100}$$

These are further expressed as:

$$\Lambda_\alpha = \lambda_-^{n_-} \lambda_0^{n_0} \lambda_+^{n_+}, \tag{EQ 101}$$

$$= (1 - \sqrt{2})^{n_-} 1^{n_0} (1 + \sqrt{2})^{n_+},$$

$$= (1 - \sqrt{2})^{n_-} (1 + \sqrt{2})^{n_+}.$$

The n's satisfy the equation

$$n = n_- + n_0 + n_+ \tag{EQ 102}$$

where n_-, n_0, and n_+ are the count of the number of -1's, 0's, and +1's, respectively, in the bit string α.

7.2 Inverse of the Selected-Coordinate Design

Recall the inverse of a matrix operator can be expressed as

$$A^{-1} = \sum_\alpha \Lambda_\alpha^{-1} \Psi_\alpha \Psi_\alpha^T. \tag{EQ 103}$$

It can be seen from (EQ 100) that none of the eigenvalues for the Selected-Coordinate Design are zero and so the SCD has an inverse. Since the eigenvectors and the eigenvalues factor for each bit it follows that the inverse also factors into a product

$$A^{-1}_{\dot{x},x}(n) = \prod_{i=1}^{n} A^{-1}_{\dot{x},x}(1) \qquad \text{(EQ 104)}$$

of one-bit inverses. This product is found to yield[8]

$$A^{-1}(1) = \begin{bmatrix} 0 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 0 \end{bmatrix}. \qquad \text{(EQ 105)}$$

The $A^{-1}(1)$ matrix has constant values along 45 degree diagonals. This corresponds to the constraint that $\dot{x}_i + x_i$ is a constant. In fact, an element of the matrix is a function only of the absolute value of this sum. Note it is the sum and not the difference between the components that determines the behavior of the inverse. As such, the inverse of the Selected-Coordinate Design can not be expressed as a function of the *distance* between $\dot{x}$ and x which depends upon their difference. This fact is not intuitively obvious since the Selected-Coordinate Design's activation rule can be expressed as a function of their difference. From the product form for the full inverse, with the insight the sum of components determines the inverse's value, it is possible to write a more explicit form for the inverse, namely

$$\boxed{A^{-1}_{\dot{x},x}(n) = (-1)^{k_0}(1)^{k_1}(0)^{k_2}} \qquad \text{(EQ 106)}$$

where the k_t are the count of the number of terms equal to t.

$$k_t = \sum_{i=1}^{n} I[t=|\dot{x}_i + x_i|], \qquad \text{(EQ 107)}$$

$$n = k_0 + k_1 + k_2. \qquad \text{(EQ 108)}$$

Curves of constant inverse are no longer hyperspheres (cubes) as in the Kanerva model but are curves with constant k values. If any two corresponding bits (± 1) of the input and the hidden location are the same then the store for that location is not modified (inverse is zero) otherwise the datum is either add or subtracted from the store. This completes the discussion of the Selected-Coordinate Design.

8. It should be noted for the simple form of the Selected-Coordinate Design it is not necessary to go through the eigenvalue problem to obtain this inverse.

8.0 Radial activation rules

I now turn to associative memories with real-valued input and output that have activation rules only a function of the *distance* between input and weights. Such systems have been considered by Poggio (1989) under the name of (Generalized) Radial Basis Functions. In this section I develop the inverse for such systems. The task at hand is to find a set of eigenfunctions and eigenvalues which satisfy radial activation rules for real-valued input. One can go quite far in this task without specifying the exact form of the radial dependence. It will be shown spherical Bessel functions play a fundamental role in the radial dependence and Gegenbauer polynomials play a role in the angular dependence.

8.1 Eigenvalue problem for radial activation rules

The task is to solve the following operator problem with n-dimensional real-valued variables and an activation rule that is a function only of the distance between its arguments:

$$A\Psi = \Lambda\Psi, \tag{EQ 109}$$

$$\int_{Volume} A(n, x, \dot{x})\Psi(\dot{x})d\dot{x} = \Lambda(n)\Psi(x)$$

where

$$x, \dot{x} \in R^n \tag{EQ 110}$$

and

$$A(n, x, \dot{x}) = f(\|x - \dot{x}\|_n) \tag{EQ 111}$$

for some scalar function f. Now (EQ 109) is the multidimensional analogue of a homogeneous linear integral equation of the second kind (Smithies, 1962) whose solution can be determined from the zeros of the Fredholm determinant. Rather than follow that path here, it is possible to decompose the operator in terms of shells and then apply Gauss's theorem for integration over surfaces. One then obtains a differential equation that is separable in the radial and nonradial components.

As with the Kanerva model, the radial activation rule can be decomposed into a superposition of shell operators

$$A(n, x, \dot{x}) = \int_0^\infty f(\rho)S(n, \rho, x, \dot{x})d\rho \tag{EQ 112}$$

where

$$S(n, \rho, x, \dot{x}) = \delta(\rho - \|x - \dot{x}\|_n). \tag{EQ 113}$$

The quantity δ is the Dirac delta function (Friedman, 1956) and has the property that for every continuous function φ,

$$\int_{-\infty}^{\infty} \delta(x)\varphi(x)dx = \varphi(0). \qquad \text{(EQ 114)}$$

Operators S with different values of ρ commute and so a common set of eigenfunctions independent of ρ can be found. That is,

$$\int_{Volume} S(n, \rho, x, \dot{x})\Psi(\dot{x})d\dot{x} = \lambda(n, \rho)\Psi(x) \qquad \text{(EQ 115)}$$

with the eigenvalues of A given in terms of the eigenvalues of S by

$$\Lambda(n) = \int_0^{\infty} \lambda(n, \rho)f(\rho)d\rho. \qquad \text{(EQ 116)}$$

Note the generality of this expression. Nothing has been said about the exact form of f. It need not be monotonically decreasing. It only need be sufficiently smooth so the integral exists. This decoupling of the exact form of radial dependence of the activation rule from the shell eigenvalue problem is very powerful for it allows the determination of the *eigenfunctions* for *all* radial activation rules.

The task of solving the eigenvalue problem for A is now reduced to the task of solving the eigenvalue problem for S.

8.2 Radial shell eigenvalue problem

Now, the integral over a volume of a delta function that is a function only of the radial component becomes an integral over a surface area. Write the dummy variable of integration as

$$\dot{x}_i = x_i + ru_i \qquad \text{(EQ 117)}$$

where r is the distance between $\dot{x}$ and x and u is a unit vector pointing from x to $\dot{x}$. Note the differential volume element can be written as $d\dot{x} = r^{n-1}(dr)\ (d\Omega)$ where $d\Omega$ is a differential solid angle. One finds:

$$\begin{aligned}\int_{Volume} S(n, \rho, x, \dot{x})\Psi(\dot{x})d\dot{x} &= \int_{Volume} \delta(\rho - \|x - \dot{x}\|)\Psi(\dot{x})d\dot{x}, \qquad \text{(EQ 118)}\\ &= \int_{\Omega}\left[\int_0^{\infty} \delta(\rho - r)\Psi(x + ru)r^{n-1}dr\right]d\Omega,\\ &= \int_{\Omega} \Psi(x + \rho u)\rho^{n-1}d\Omega.\end{aligned}$$

The shell eigenvalue problem becomes

$$\int_{\Omega} \Psi(x+\rho u)\rho^{n-1} d\Omega = \lambda(n,\rho)\Psi(x). \qquad \text{(EQ 119)}$$

This integral form can be transformed into a set of differential equations for Ψ and λ by first differentiating (EQ 119) by ρ. In this process the derivative of Ψ under the integral sign must be taken. This leads to the gradient of Ψ dotted with the unit surface normal u:

$$\frac{\partial}{\partial\rho}\Psi(\dot{x}) = \sum_{i=1}^{n} \frac{\partial \dot{x}_i}{\partial\rho}\frac{\partial}{\partial \dot{x}_i}\Psi(\dot{x}), \qquad \text{(EQ 120)}$$

$$= \sum_{i=1}^{n} u_i \frac{\partial}{\partial \dot{x}_i}\Psi(\dot{x}),$$

$$= u \cdot \nabla\Psi.$$

So

$$\frac{\partial}{\partial\rho}[\Psi\rho^{n-1}] = (n-1)\rho^{-1}[\Psi\rho^{n-1}] + (u\cdot\nabla\Psi)\rho^{n-1}. \qquad \text{(EQ 121)}$$

Therefore, we find

$$\frac{\partial}{\partial\rho}\int_{Area}\Psi da = (n-1)\rho^{-1}\int_{Area}\Psi da + \int_{Area}(u\cdot\nabla\Psi)\,da. \qquad \text{(EQ 122)}$$

Now one may exploit Gauss's theorem (Richards, 1959) which holds for arbitrary dimensional spaces so that

$$\int_{Area}(u\cdot\nabla\Psi)\,da = \int_{Volume}\left(\nabla^2\Psi\right)d\dot{x}. \qquad \text{(EQ 123)}$$

That is, the surface integral of a vector field dotted into the normal of the differential element of the surface is equal to the volume integral of the divergence of the vector field over the volume enclosed by the surface. In our case the vector field is the gradient of a scalar and so the divergence becomes the Laplacian, ∇^2 , of the scaler function.

$$\nabla^2 = \sum_{j=1}^{n}\frac{\partial^2}{\partial x_j^2} \qquad \text{(the Laplacian).} \qquad \text{(EQ 124)}$$

Now the Laplacian is taken relative to the argument of Ψ, namely $\dot{x}$. This is equivalent to the Laplacian relative to x evaluated at $\dot{x}$ since from (EQ 117)

$$\frac{\partial}{\partial\dot{x}}\Psi(\dot{x}) = \frac{\partial}{\partial x}\Psi(x+\rho u). \qquad \text{(EQ 125)}$$

Therefore, the Laplacian may be moved outside of the integral sign leaving a volume integral of Ψ as

$$\frac{\partial}{\partial\rho}\int_{Area}\Psi da = (n-1)\rho^{-1}\int_{Area}\Psi da + \nabla^2\int_{Volume}\Psi d\dot{x}. \qquad \text{(EQ 126)}$$

The volume integral of Ψ is the radial integral of the surface integral of Ψ. The surface integral is by definition the eigenvalue problem and can be replaced by its right-hand side. The radial integral I will leave in integral form for a moment:

$$\int_{Volume}\Psi d\dot{x} = \int_0^{\rho}\left[\int_{Area}\Psi(x+ru)da\right]dr, \qquad \text{(EQ 127)}$$

$$= \int_0^{\rho}[\lambda(n,r)\Psi(x)]\,dr,$$

$$= \left(\int_0^{\rho}\lambda(n,r)dr\right)\Psi(x).$$

The original differentiation of the area integral of Ψ with respect to ρ is equal to the derivative of the right-hand side of the eigenvalue equation, so

$$\frac{\partial\lambda}{\partial\rho}\Psi = (n-1)\frac{\lambda}{\rho}\Psi + \left(\int_0^{\rho}\lambda(n,r)dr\right)\nabla^2\Psi(x). \qquad \text{(EQ 128)}$$

A second differentiation with respect to ρ removes the integral and yields the differential equation

$$\frac{\partial^2\lambda}{\partial\rho^2}\Psi = (n-1)\frac{\partial}{\partial\rho}\left(\frac{\lambda}{\rho}\right)\Psi + \lambda\nabla^2\Psi(x). \qquad \text{(EQ 129)}$$

By dividing both sides of this equation by the product $\lambda\Psi$ and rearranging terms we obtain an equation whose left-hand side is a function only of ρ and whose right-hand side is a function only of x. The equation is

$$\frac{1}{\lambda}\frac{\partial^2\lambda}{\partial\rho^2} - \frac{(n-1)}{\lambda}\frac{\partial}{\partial\rho}\left(\frac{\lambda}{\rho}\right) = \frac{1}{\Psi}\nabla^2\Psi(x) = \text{constant}. \qquad \text{(EQ 130)}$$

The only way this equation can hold for all ρ and x is for each side to be equal to a constant, say, $-\omega^2$. The two differential equations, one for the eigenvalues and the other for the eigenfunctions, become

$$\left(\frac{d^2\lambda}{d\rho^2} - (n-1)\frac{d}{d\rho}\left(\frac{\lambda}{\rho}\right)\right) + \omega^2\lambda = 0 \quad \text{(radial eigenvalues)} \qquad \text{(EQ 131)}$$

and

$$\nabla^2\Psi(x) + \omega^2\Psi = 0 \quad \text{(radial eigenfunctions).} \qquad \text{(EQ 132)}$$

8.3 Solution of differential equations for radial activation rules

I now proceed with the process of finding the solution to the differential equations for the eigenvalues and eigenvectors of radial activation rules.

8.3.1 Radial eigenvalues

Differential equations of the following form (Gradshteyn, 1965, EQ 8.491.6)[9]

$$u'' + \frac{1-2\alpha}{z}u' + \left[\beta^2 + \frac{\alpha^2 - \nu^2}{z^2}\right]u = 0 \qquad \text{(EQ 133)}$$

have solutions

$$u = z^{\alpha}Z_{\nu}(\beta z) \qquad \text{(EQ 134)}$$

where the Z are any of the Bessel functions of the first kind, J, second kind, N (Neumann functions), or third kind H (Hankel's functions). When (EQ 131) is expanded we find it corresponds to a form of (EQ 133), namely

$$\lambda'' + \frac{1-n}{\rho}\lambda' + \left(\omega^2 + \frac{n-1}{\rho^2}\right)\lambda = 0. \qquad \text{(EQ 135)}$$

Using the appropriate associations one can then right down the following solution for λ as

9. There is a double misprint in one of the equations on page 971 of Gradshteyn. The correct form for equation 8.491.3. is

$$u'' + \frac{1-2\alpha}{z}u' + \left[(\beta\gamma z^{\gamma-1})^2 + \frac{\alpha^2 - \nu^2\gamma^2}{z^2}\right]u = 0, \qquad u = z^{\alpha}Z_{\nu}(\beta z^{\gamma}).$$

$$\lambda(n, \omega, \rho) = (\omega\rho)^{n/2} Z_{\frac{n}{2}-1}(\omega\rho). \quad \text{(EQ 136)}$$

The order of the Bessel function Z is $n/2-1$ which is integral or half integral depending upon the dimensionality of the space. For n odd, the solutions are called spherical Bessel functions (Sneddon, 1961, p120).

The full radial eigenvalues are then the integral over ρ of the shell eigenvalues weighted by the radial activation function f. This is expressed as

$$\Lambda(n, \omega) = \int_0^\infty \lambda(n, \omega, \rho) f(\rho) d\rho \quad \text{(radial eigenvalues).} \quad \text{(EQ 137)}$$

An explicit form for Λ will depend upon the specification of f.

8.3.2 Radial eigenfunctions

The differential equation for the eigenfunctions is a point-source form of the wave-diffusion equation (Richards, 1959) with no time dependence. Its solution in n-dimensional radial coordinates is given in term of Hyperspherical harmonics (Avery, 1989). The first two chapters of Avery's book are a lovely summary of the generalization of spherical harmonics to n-dimensions. I follow closely his derivations here. The generalized Laplacian operator ∇^2 can be written in the form

$$\nabla^2 = \sum_{j=1}^{n} \frac{\partial^2}{\partial x_j^2} = \frac{1}{r^{n-1}} \frac{\partial}{\partial r}\left(r^{n-1} \frac{\partial}{\partial r}\right) - \frac{L^2}{r^2} \quad \text{(EQ 138)}$$

where L^2 is the *generalized angular momentum* operator, defined by

$$L^2 \equiv -\sum_{i>j}^{n} L^2_{i,j} \quad \text{(EQ 139)}$$

and

$$L_{i,j} \equiv x_i \frac{\partial}{\partial x_j} - x_j \frac{\partial}{\partial x_i}. \quad \text{(EQ 140)}$$

The eigenfunctions, $Y_{l,m}(\Omega)$, of the generalized angular momentum operator L^2 satisfy

$$L^2 Y_{l,m}(\Omega) = l(l+n-2) Y_{l,m}(\Omega) \quad \text{(EQ 141)}$$

and are called *hyperspherical harmonics*. They are a pure angular function, independent of the hyperradius, r. They are also orthonormal since

$$\int_{\Omega} Y_{l,m}(\Omega) Y_{l',m'}(\Omega) d\Omega = \delta_{l,l'} \delta_{m,m'}. \qquad \text{(EQ 142)}$$

The hyperspherical harmonics also satisfy a *sum rule*

$$C_l(\dot{u} \cdot u) = K_l \sum_m Y^*_{l,m}(\dot{\Omega}) Y_{l,m}(\Omega) \qquad \text{(EQ 143)}$$

where

$$K_l = 2 \frac{\left(\frac{n}{2} - 1\right) \pi^{n/2}}{\Gamma(\frac{n}{2}) \left(\frac{n}{2} - 1 + l\right)} \qquad \text{(EQ 144)}$$

and $\dot{u}$, u are unit vectors in the directions specified by $\dot{\Omega}$, Ω. The C's are called *Gegenbauer polynomials* and they satisfy the same differential equation as the hyperspherical harmonics.

We now look for a solution of (EQ 132) by the method of separation of variables where the eigenfunction is written as a product of hyperspherical harmonics and a radial function, $R(r)$, to be determined. This is written as

$$\Psi_{\omega,l,m}(x) = R_\omega(r) Y_{l,m}(\Omega). \qquad \text{(EQ 145)}$$

When we do this the differential equation becomes

$$\left[\frac{1}{r^{n-1}} \frac{\partial}{\partial r} \left(r^{n-1} \frac{\partial}{\partial r} \right) - \frac{L^2}{r^2} \right] (R_\omega(r) Y_{l,m}(\Omega)) + \omega^2 R_\omega(r) Y_{l,m}(\Omega) = 0. \qquad \text{(EQ 146)}$$

Applying the operations to each function followed by division by RY yields

$$r^2 \left(\frac{\frac{1}{r^{n-1}} \frac{\partial}{\partial r} r^{n-1} \frac{\partial}{\partial r} R_\omega(r)}{R_\omega(r)} + \omega^2 \right) = \frac{L^2 Y_{l,m}(\Omega)}{Y_{l,m}(\Omega)} = \text{constant}. \qquad \text{(EQ 147)}$$

The radial and angular dependencies have been separated. The equation must hold for all values of the independent variables r and Ω which can only be satisfied if the equation is equal to a constant. In this case, however, we know the constant must be equal to the eigenvalue for the angular function $Y_{l,m}$ which is $l(l+n-2)$. Therefore, the differential equation for the radial dependence becomes

$$R''_{\omega,l}(r) + \frac{(n-1)}{r} R'_{\omega,l}(r) + \left(\omega^2 - \frac{l(l+n-2)}{r^2}\right) R_{\omega,l}(r) = 0 \qquad \text{(EQ 148)}$$

where the index l has been included to show R's dependence upon the angular momentum parameter. With appropriate boundary conditions specified for R the parameter ω will also be constrained.

The differential equation for R is remarkably similar to that for λ and yields similar Bessel function solutions, namely

$$R_{\omega,l}(r) = (\omega r)^{-\left(\frac{n}{2}-1\right)} Z_{\frac{n}{2}-1+l}(\omega r) \qquad \text{(EQ 149)}$$

where now the order of the Bessel function is also a function of the angular momentum, l, and the power of the hyperradius, r, is now negative.

8.3.3 Boundary conditions

In electro-statics there usually are natural constraints on the class of functions at the boundary of the domain under consideration. Either the electric field goes to zero on the boundary or its derivative does. For associative memories and neural networks the issue of boundary conditions is not usually considered. Since I do not have a good sense of what conditions may be appropriate for associative memories I will arbitrarily consider functions with Dirichlet boundary conditions and let the reader modify the solutions from here on for his own situation if this assumption is not appropriate.

Dirichlet boundary conditions (Jackson, 1962, p15) specify the value of the function on a closed surface. This value I take to be zero. If the input vector, x, is constrained to lie within a sphere of radius a then by taking a large enough the restriction to zero values on a does not markedly restrict the values of the function at interior points.

The condition the eigenfunction is zero on the boundary $r=a$ places a restriction on the allowable values of ω through the zeros of the Bessel function.

$$R_{\omega,l}(a) = 0 \qquad \text{(Dirichlet boundary condition)} \qquad \text{(EQ 150)}$$

implies

$$Z_{\frac{n}{2}-1+l}(\omega_{k,l} a) = 0 \qquad \text{(EQ 151)}$$

where

$$\omega_{k,l} = \frac{z_{k,l}}{a} \qquad \text{(EQ 152)}$$

with $z_{k,l}$ a zero of the Bessel function. For each value of l (and n) there are a countably infinite number of zeros (k values) of Z. I now label the radial eigenfunctions with k and introduce a normalization factor N. One can now write

$$\boxed{R_{k,l}(r) = N_{k,l}(\omega_{k,l}r)^{-\left(\frac{n}{2}-1\right)}Z_{\frac{n}{2}-1+l}(\omega_{k,l}r)} \qquad \text{(EQ 153)}$$

where

$$\int_0^a R^2_{k,l}(r)r^{n-1}dr = 1 \qquad \text{(EQ 154)}$$

and

$$N_{k,l} = \omega_{k,l}^{\frac{n}{2}-1}\left[\int_0^a Z^2_{\frac{n}{2}-1+l}(\omega_{k,l}r)rdr\right]^{-1/2}. \qquad \text{(EQ 155)}$$

The R functions also satisfy the following orthogonality condition

$$\int_0^a R_{k,l}(r)R_{k',l}(r)r^{n-1}dr = \delta_{k,k'} \qquad \text{(EQ 156)}$$

since the Bessel functions satisfy[10]

$$\int_0^a Z_{\frac{n}{2}-1+l}(\omega_{k,l}r)Z_{\frac{n}{2}-1+l}(\omega_{k',l}r)rdr = N^{-2}_{k,l}\delta_{k,k'}. \qquad \text{(EQ 157)}$$

The eigenfunctions are now written as

$$\boxed{\Psi_{k,l,m}(x) = R_{k,l}(r)Y_{l,m}(\Omega).} \qquad \text{(EQ 158)}$$

Using the quantized values for the frequencies the eigenvalues for the shell operator are expressed as

10. See Gradshteyn, p634, eq 5.54.1. for an expression for arbitrary Bessel functions. (EQ 157) holds for $Z=J$, Bessel functions of the first kind.

$$\lambda_{k,l}(n,\rho) = (\omega_{k,l}\rho)^{n/2} Z_{\frac{n}{2}-1}(\omega_{k,l}\rho) \qquad \text{(EQ 159)}$$

which when combined with the radial activation function yields the quantized eigenvalues for the full problem.

$$\Lambda_{k,l}(n) = \int_0^\infty \lambda_{k,l}(n,\rho) f(\rho) d\rho \qquad \text{(discrete radial eigenvalues).} \qquad \text{(EQ 160)}$$

The complete solution of the eigenvalue problem for radial activation rules has been obtained. I now turn to the task of finding the inverse activation rule for radial activation functions.

8.4 Inverse of radial activation rules

The inverse of the radial activation rule can now be expressed in terms of the spectral representation of the operator as previously stated in (EQ 26). Using the eigenvalues and eigenvectors found in the last section we see

$$A^{-1}(\dot{x}, x) = \sum_{k,l,m} \frac{\Psi_{k,l,m}(\dot{x})\Psi^*_{k,l,m}(x)}{\Lambda_{k,l}} \qquad \text{(EQ 161)}$$

becomes

$$A^{-1}(\dot{x}, x) = \sum_{k,l,m} \frac{[R_{k,l}(\dot{r})Y_{l,m}(\dot{\Omega})]\,[R_{k,l}(r)Y_{l,m}(\Omega)]^*}{\Lambda_{k,l}}. \qquad \text{(EQ 162)}$$

Rearranging order of summation and collecting terms which are only a function of the index m gives

$$A^{-1}(\dot{x}, x) = \sum_{k,l}\left(\frac{R_{k,l}(\dot{r})R^*_{k,l}(r)}{\Lambda_{k,l}}\right)\left(\sum_m Y_{l,m}(\dot{\Omega})Y^*_{l,m}(\Omega)\right). \qquad \text{(EQ 163)}$$

The hyperspherical harmonics' sum rule can now be applied to yield an expression in terms of Gegenbauer polynomials, C. The expression is

$$A^{-1}(\dot{x}, x) = \sum_{k,l}\left(\frac{R_{k,l}(\dot{r})R^*_{k,l}(r)}{\Lambda_{k,l}}\right)\left(\frac{C_l(\dot{u}\cdot u)}{K_l}\right). \qquad \text{(EQ 164)}$$

The index k is applicable only to the first factor so regrouping gives

$$A^{-1}(\dot{x}, x) = \sum_{l}\left(\sum_k \frac{R_{k,l}(\dot{r})R^*_{k,l}(r)}{\Lambda_{k,l}}\right)\left(\frac{C_l(\dot{u}\cdot u)}{K_l}\right). \qquad \text{(EQ 165)}$$

One is now in a position to substitute the explicit Bessel function form for the radial dependence of the eigenvectors. This yields the expression

$$A^{-1}(\dot{x}, x) = (\dot{r}r)^{-\left(\frac{n}{2}-1\right)}\sum_l\left(\sum_k \frac{Z_{\frac{n}{2}-1+l}(\omega_{k,l}\dot{r})Z^*_{\frac{n}{2}-1+l}(\omega_{k,l}r)}{\Lambda_{k,l}\int_0^a Z^2_{\frac{n}{2}-1+l}(\omega_{k,l}r)r\,dr}\right)\left(\frac{C_l(\dot{u}\cdot u)}{K_l}\right). \qquad \text{(EQ 166)}$$

One knows from symmetry the inverse, like the rule itself, must be a function only of the *distance* between its arguments. This is partly revealed by the argument of the Gegenbauer polynomial which is a function only of the inner product between the directions of $\dot{x}$ and x. An orthogonal transformation of the space (rotation) applied to both $\dot{x}$ and x leaves their

lengths ($\dot{r}$, r)and the inner product ($\dot{u}$, u) unchanged. The inverse is therefore invariant under such a transformation.

This completes the discussion of the inverse for radial activation rules. For each explicit form of radial dependency, $f(r)$, there will be an inverse activation rule as given by (EQ 166). Further work is necessary to investigate this expression for, Gaussian, exponential, and spherical activation rules.

9.0 Summary and conclusions

In previous sections were presented the derivations of the inverse activation rules for three associative memory models. I summarize the results here.

9.1 The Kanerva model

The Sparse Distributed Memory model (Kanerva, 1988) uses a hard threshold activation rule with binary input and binary weights. Any pattern closer than r in Hamming distance to the weights of a hidden node will activate the node with unit activity; otherwise the node's activity will be zero. The Sparse Distributed Memory activation rule is written

$$A_{x,\dot{x}}(n,r) = \begin{cases} 1 & d_n(x,\dot{x}) \le r, \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(activation rule)} \qquad \text{(EQ 167)}$$

The inverse activation rule is not simple.

$$A^{-1}_{\dot{x},x}(n,r) = 2^{-n} \sum_{w=0}^{n} \binom{n}{w} \frac{C_d(n-w,w)}{\binom{n}{d} \sum_{\rho=0}^{r} C_\rho(n-w,w)} \qquad \text{(inverse activation rule)} \qquad \text{(EQ 168)}$$

with

$$C_\rho(a,b) = \sum_{k=0}^{\rho} (-1)^k \binom{a}{\rho-k} \binom{b}{k}, \qquad \text{(EQ 169)}$$

$$d = d_n(\dot{x},x).$$

As can be seen from Figure 8 there is a general oscillatory behavior to the inverse (it bears some resemblance to the Fourier transform of a step function). This oscillatory behavior creates (in the multidimensional space) layers with alternating sign. The strength and sign of a layer determines the factor a datum will be multiplied by before being added into the memory in that region of space.

That is, about each hidden node is an "onion" with layers of alternating sign that determines how that node treats data to be written to its own local store (output weights). Unlike the activation rule that is localized about each hidden node, the effects of the inverse rule spread throughout the space. This is analogous to the behavior of a function and its Fourier transform where concentration of the function in a localized region produces a spreading in the transform space. Hence, learning with the inverse rule adjusts all of memory. Another analogy of the inverse operator can be drawn to the "Mexican hat" type of on-center-off-surround neural response in the eye. Such a response can be understood as a

general neural property necessary to form sharp boundaries in vision. It localizes effects in a distributive system in the same way the inverse localizes memory.

9.2 The Selected-Coordinate Design

The Selected-Coordinate Design (Jaeckel, 1989) is similar to the Kanerva model except that only a subset of an input pattern is attended to by each hidden node. This is accomplished by including "don't care" bits with the binary values that an address (input weights) of a hidden node can take. Such "sparse" sampling of input is also found in the sampling of mossy fibers by granule cells in the cerebellum (Marr, 1969). Another difference is that among the "care" bits an exact match must occur for the hidden node to be activated. The activation rule for the Selected-Coordinate Design is

$$A_{x,\dot{x}}(n) = \begin{cases} 1 & x_i\dot{x}_i \geq 0,\ i = 1, 2, \ldots, n, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{(activation rule)} \qquad \text{(EQ 170)}$$

$$x, \dot{x} \in \{-1,0,1\}^n.$$

The exact match condition makes it easy to derive the inverse of the Selected-Coordinate Design where it is found

$$A^{-1}_{\dot{x},x} = (1)^{k_1}(0)^{k_2}(-1)^{k_3} \qquad \text{(inverse activation rule).} \qquad \text{(EQ 171)}$$

The k_s are the count of the number of components of $|x + \dot{x}|$ which equal s. This write rule has the property that only points, x, which are *dissimilar* to a hidden location, $\dot{x}$, are written to $\dot{x}$'s store. Data will be added to the store if the number of "care" bits in $\dot{x}$ is even, otherwise they will be subtracted.

9.3 Radial activation rules

Radially dependent rules for function interpolation have been considered in the past and a generalized form in the context of neural networks has been considered by Poggio (1989). Radial activation rules can be stated as

$$A(x, \dot{x}) = f(\|x - \dot{x}\|) \qquad \text{(activation rule)} \qquad \text{(EQ 172)}$$

where f is an arbitrary real valued function of a single variable. It is shown the eigenvectors are radially Bessel functions, angularly Gegenbauer polynomials, and the eigenvalues are a convolution of Bessel functions with the radial activation rule. The inverse activation rule for radial functions is

$$A^{-1}(\dot{x}, x) = (\dot{r}r)^{-\left(\frac{n}{2}-1\right)} \sum_l \left(\sum_k \frac{Z_{\frac{n}{2}-1+l}(\omega_{k,l}\dot{r}) Z^*_{\frac{n}{2}-1+l}(\omega_{k,l}r)}{\Lambda_{k,l} \int_0^a Z^2_{\frac{n}{2}-1+l}(\omega_{k,l}r) r dr} \right) \left(\frac{C_l(\dot{u} \cdot u)}{K_l} \right) \quad \text{(EQ 173)}$$

(inverse activation rule)

where

$$\Lambda_{k,l}(n) = \int_0^\infty \left((\omega_{k,l}\rho)^{n/2} Z_{\frac{n}{2}-1}(\omega_{k,l}\rho) \right) f(\rho) d\rho \ , \quad \text{(EQ 174)}$$

$$\omega_{k,l} = \frac{z_{k,l}}{a},$$

$$Z_{\frac{n}{2}-1+l}(z_{k,l}) = 0.$$

9.4 Conclusions

The use of iterative error correction in artificial neural networks and associative memories appears to be a series expansion of some operator. When this intuition is rigorously formulated for asymptotically large single-layer neural networks it is found error correction, indeed, is a series expansion of the inverse activation rule of the network.

Closed form expressions for the inverse show information must be spread throughout memory in an oscillatory manner to cancel the effects of competing patterns.

The presence of an inverse also shows write-rules exist for very rapid single-step learning.

The methodology of iterative error correction does not provide an understanding of the form of the weights resulting from its application whereas inverse activation rules specifies precisely the weightings that must be applied to the data to give total recall. It is the knowledge that such exact specifications exist which is the major result of this paper.

The existence of an inverse in the asymptotic case, causes one to seek the same type of analysis for practical distributive memories. The lattice of input observation points in such systems can act as a starting point of this analysis. The similarity of lattice points (as determined by the activation function between the points) forms a matrix that can be inverted under certain circumstances. The analysis of this case and others will be the subject of future work (Danforth, 1991).

It is expected drastic methodological changes will not be necessary to increase the quality of recall and generalization in practical systems. It is simply a question of *what* changes

should to be applied. Theoretical analysis is the guide that can show us which passage will lead us to the appropriate changes.

In the domain of neural networks (still new and in search of solid mathematical principles), I found this research to be refreshingly concrete.

10.0 References

1. Avery, J. (1989). *Hyperspherical Harmonics, Applications in Quantum Theory.* Boston: Kluwer Academic Publishers.
2. Danforth, D. G. (1990). An Empirical Investigation of Sparse Distributed Memory Using Discrete Speech Recognition. RIACS Technical Report 90.18.
3. _____ (1991). Generalization in Distributive Associative Memories. RIACS Technical Report 91.xx (in preparation).
4. Feller, W. (1968). *An Introduction to Probability Theory and Its Applications*, Vol I. New York: Wiley & Sons.
5. Friedman, B. (1956). *Principles and Techniques of Applied Mathematics.* New York: Wiley & Sons.
6. Gradshteyn, I. S. and I. M. Ryzhik (1965). *Table of Integrals, Series, and Products.* New York: Academic Press.
7. Goldstein, H. (1950). *Classical Mechanics.* Reading, MA: Addison-Wesley.
8. Harwit, M. and N. J. A. Sloane (1979). *Hadamard Transform Optics.* Academic Press, New York.
9. Jaeckel, L. A. (1989). An Alternative Design for a Sparse Distributed Memory. RIACS Technical Report 89.28.
10. Kanerva, P. (1988). *Sparse Distributed Memory.* Cambridge, MA: MIT Press.
11. Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik* 14:85-100. Reprinted in J. A. Anderson (ed.), *Neurocomputing, Foundations of Research* (Cambridge, MA: MIT Press, 1988).
12. Marr, D. (1969). A theory of cerebellar cortex. *Journal of Physiology* 202.
13. Merzbacher, E. (1961). *Quantum Mechanics.* New York: Wiley & Sons.
14. Poggio, B. (1989). A Theory of Networks for Approximation and Learning. Massachusetts Institute of Technology A.I. Memo No. 1140.
15. Prager, R. W. and F. Fallside (1989). The modified Kanerva model for automatic speech recognition. *Computer Speech and Language, 3,61-81.*
16. Richards, P. I. (1959). *Manual of Mathematical Physics.* New York: Pergamon Press.
17. Rumelhart, D. and J. L. McClelland (1989). *Parallel Distributed Processing.* Cambridge, MA: MIT Press, Vol 1.
18. Selby, S. M. (1970). *Standard Mathematical Tables.* Cleveland, Ohio: The Chemical Rubber Co., Eighteenth Edition.
19. Smithies, F. (1962). Integral Equations. Cambridge Tracts in Mathematics and Mathematical Physics, No. 49. Cambridge: University Press.
20. Sneddon, I. N. (1961). *Special Functions of Mathematical Physics and Chemistry.* New York: Interscience Publishers.