A Study on Associative Neural Memories

B.D.C.N.Prasad[1]
Dept. of Computer Applications,
P V P Siddhartha Institute of Technology
Vijayawada, India
bdcnprasad@gmail.com

P E S N Krishna Prasad[2]
Dept. of Computer Science & Engineering
Aditya Engineering College
Kakinada, India
surya125@gmail.com

Sagar Yeruva[3]
Department of Computer Applications,
St. Peter's Engineering College, Hyderabad.
sagaryeruva@yahoo.com

P Sita Rama Murty[4]
Dept. of Computer Science & Engineering
Sri Sai Aditya Institute of Science & Technology
Kakinada, India
psramam.1@gmail.com

Abstract— **Memory plays a major role in Artificial Neural Networks. Without memory, Neural Network can not be learned itself. One of the primary concepts of memory in neural networks is Associative neural memories. A survey has been made on associative neural memories such as Simple associative memories (SAM), Dynamic associative memories (DAM), Bidirectional Associative memories (BAM), Hopfield memories, Context Sensitive Auto-associative memories (CSAM) and so on. These memories can be applied in various fields to get the effective outcomes. We present a study on these associative memories in artificial neural networks.**

Keywords-Associative memories; SAM; DAM; Hopfield model; BAM; Holographic Associative Memory (HAM); Context-sensitive Auto-associative Memory (CSAM); Context-sensitive Asynchronous Memory (CSYM)

I. INTRODUCTION

Learning is the way we acquire knowledge about the world around us, and it is through this process of knowledge acquisition, that the environment alerts our behavioral responses. Learning allows us to store and retain knowledge; it builds our memories.

Aristotle stated about memory: first, the elementary unit of memory is a sense image and second, association and links between elementary memories serve as the basis for higher level cognition. One author stated, memory stands for the elementary unit and association for recollection between elementary units.

In a neurobiological context, memory refers to the relatively enduring neural alterations induced by the interaction of an organism with its environment. Without such a change, there is no memory. The memory must be useful and accessible to the nerves system that influences the future behavior.

Memory and Learning are intricately connected. When a particular activity pattern is learned, it is stored in the brain where it can be recalled later when required. Learning encodes information. A system learns a pattern if the system encodes the pattern in its structure. The system structure changes as the system learns the information. So, learning involves change. That change can be represented in memory for future behavior.

Over the past century the psychologists have studied learning based on fundamental paradigms***: non-associative*** and ***associative***. In *non-associative learning* an organism acquires the properties of a single repetitive stimulus. In *associative learning* [Edward Thorndike, B.F. Skinner], an organism acquires knowledge about the relationship of either one stimulus to another, or one stimulus to the organisms own behavioral response to that stimulus.

On the neuronal basis of formation of memories into two distinct categories: STM (short term memory) and LTM (long term memory). Inputs to the brain are processed into STM's which last at the most for a few minutes. Information is downloaded into LTM's for more permanent storage. One of the most important functions of our brain is the laying down and recall of memories. It is difficult to imagine how we could function without both short and long term memory. The absence of short term memory would render most tasks extremely difficult if not impossible - life would be punctuated by a series of one time images with no logical connection between them. Equally, the absence of any means of long term memory would ensure that we could not learn by past experience.

The acquisition of knowledge is an active, on going cognitive process based on our perceptions. An important point about the learning mechanism is that it distributes the memory over different areas, making them robust to damage. Distributed storage permits the brain to work easily from partially corrupted information.

II. ASSOCIATIVE MEMORIES

The associative memory models[4], an early class of neural models that fit perfectly well with the vision of cognition emergent from today brain neuro-imaging techniques, are inspired on the capacity of human cognition to build calculus makes them a possible link between connectionist models and classical artificial intelligence developments.

Our memories function as an **associative** or **content-addressable**. That is, a memory does not exist in some isolated fashion, located in a particular set of neurons. Thus memories are stored in *association* with one another. These different sensory units lie in completely separate parts of the brain, so it is clear that the memory of the person must be distributed throughout the brain in some fashion.

We access the memory by its contents not by where it is stored in the neural pathways of the brain. This is very powerful; given even a poor photograph of that person we are quite good at reconstructing the persons face quite accurately. This is very different from a traditional computer where specific facts are located in specific places in computer memory. If only partial information is available about this location, the fact or memory cannot be recalled at all.

Traditional measures of associative memory performance are its *memory capacity* and *content-addressability*. Memory capacity refers to the maximum number of associated pattern pairs that can be stored and correctly retrieved while content-addressability is the ability of the network to retrieve the correct stored pattern. Obviously, the two performance measures are related to each other.

It is known that using Hebb's learning rule in building the connection weight matrix of an associative memory yields a significantly low memory capacity. Due to the limitation brought about by using Hebb's learning rule, several modifications and variations are proposed to maximize the memory capacity.

A. *Model*

Associative memory maps[4,6] data from an input space to data in an output space. In general, this mapping is from unknown domain points to known range points, where the memory learns an underlying association from a training data set.

For non-learning memory models, which have their origin in additive neuronal dynamics, connection strength's are "programmed" a priori depending upon the association that are to be encoded in the system. Sometimes these memories are referred to as matrix associative memories, because a connection matrix W, encodes associations $(A_i, B_i)_{i-1}^{Q}$, where $A_i \in B^n$ and $B_i \in B^n$. If (A_i, B_i) is one of the programmed memories then B_i is called the association of A_i. When A_i and B_i are in different spaces then the model is ***hetero-associative memory.*** *i*.e. it associates two different vectors with one another. if $A_i = B_i$, then the model is ***Auto-associative memory.*** i.e. it associates a vector with itself. Associative memory models enjoy properties such as fault tolerance.

Associative Neural Memories

Associative neural memories are concerned with associative learning and retrieval of information (vector patterns) in neural networks. These networks represent one of the most extensively analyzed classes of artificial neural networks.

Several associative neural memory models have been proposed over the last two decades. These memory models can be classified into various ways depending on

- Architecture (Static *versus* Dynamic)
- Retrieval Mode (Synchronous v*ersus* Asynchronous)
- Nature of stored association (Auto-associative *versus* Hetero-associative)
- Complexity and capability of memory storage

1) *Simple Associative Memories*

One of the earliest associative memory models is the correlation memory [Anderson, 1972; Kohonen, 1972; Nakano, 1972]. This correlation memory consists of a single layer of **L** non interacting linear units, with the l^{th} unit having a weight vector $w_l \in R^n$. It associates real values input column vectors $x^k \in R^n$ which corresponding real valued output column vectors $y^k \in R^n$ according to the transfer eq.:

$$y^k = Wx^k \tag{1}$$

Where $\{x^k, y^k\}, k = 1,2,....n$ a collection of desired associations and W is is an L×N interconnection matrix whose l^{th} row is given by w_l^T. This associative memory is characterized by linear matrix vector multiplication retrievals. Hence it is referred to as a ***linear associative memory*** [1](LAM). This LAM is said to be *hetero-associative* because y^k is different (in encoding and/ dimensionality) from x^k. If $y^k = x^k$ for all k, then this memory is called *auto-associative*.

The correlation memory is a LAM that employs a simple recording or storage recipe for loading *m* associations $\{x^k, y^k\}, k = 1,2,....n$ into memory. The recording recipe is responsible for synthesizing *W* and is given by

$$W = \sum_{k=1}^{m} y^k (x^k)^T = YX^T \tag{2}$$

Where *W* is the Correlation Matrix of *m* associations.

2) *Simple Nonlinear Associative Memory Model*

The binary-valued associations[1] $\mathbf{x}^k$, {-1, +1} N and $\mathbf{y}^k$ {-1, +1} *L* and the presence of a clipping nonlinearity *F* operating component wise on the vector Wx, according to

$$y = F[Wx] \tag{3}$$

relaxes some of the constraints imposed by correlation recording of a LAM. Here, **W** needs to be synthesized with the requirement that only the sign of the corresponding components of $\mathbf{y}^k$ and $\mathbf{Wx}^k$ agree. Next, consider the normalized correlation recording recipe given by:

$$W = \frac{1}{n}\sum_{k=1}^{m} y^k \left(x^k\right)^T \quad (4)$$

This automatically normalizes the x^k vectors. Now, if one of the recorded key patterns $\mathbf{x}^h$ is presented as input, then the following expression for the retrieved memory pattern can be written:

$$\tilde{y}^h = F\left[y^h + \frac{1}{n}\sum_{k \neq h}^{m} y^k \left(x^k\right)^T x^h\right] = F\left[y^h + \Delta^h\right] \quad (5)$$

3) Optimal Linear Associative Memory (OLAM)

The correlation recording recipe does not make optimal use of the LAM interconnection weights. A more optimal recording technique can be derived which guarantees perfect retrieval of stored memories $\mathbf{y}^k$ from inputs $\mathbf{x}^k$ as long as the set $\{\mathbf{x}^k; k = 1, 2, ..., m\}$ is linearly independent (as opposed to the more restrictive requirement of orthogonality required by the correlation-recorded LAM). This recording technique leads to the optimal linear associative memory[1,7] (OLAM). For perfect storage of m associations $\{\mathbf{x}^k, \mathbf{y}^k\}$, a LAM's interconnection matrix $\mathbf{W}$ must satisfy the matrix equation given by:

$$Y = WX \quad (6)$$

This equation can always be solved exactly if all m vectors $\mathbf{x}^k$ (columns of $\mathbf{X}$) are linearly independent, which implies that m must be smaller or equal to n. For the case $m = n$, the matrix $\mathbf{X}$ is square and a unique solution for $\mathbf{W}$ in Equation (6) exists giving:

$$W^* = YX^{-1} \quad (7)$$

Which requires that the matrix inverse $\mathbf{X}^{-1}$ exists; i.e., the set $\{\mathbf{x}^k\}$ is linearly independent. Thus, this solution guarantees the perfect recall of any $\mathbf{y}^k$ upon the presentation of its associated key $\mathbf{x}^k$.

B. Dynamic Associative Memories (DAMs)

Associative memory performance can be improved by utilizing more powerful architectures than the simple ones considered above. As an example, consider the auto associative version of the single layer associative memory employing units with the sign activation function and whose transfer characteristics are given by Equation (3). Now assume that this memory is capable of associative retrieval of a set of m bipolar binary memories $\{\mathbf{x}^k\}$. Upon the presentation of a key $\hat{\mathbf{x}}^k$ which is a noisy version of one of the stored memory vectors $\mathbf{x}^k$, the associative memory retrieves (in a single pass) an output $\mathbf{y}$ which is closer to stored memory $\mathbf{x}^k$ than $\hat{\mathbf{x}}^k$. In general, only a fraction of the noise (error) in the input vector is corrected in the first pass. Intuitively, we may proceed by taking the output $\mathbf{y}$ and feed it back as an input to the associative memory hoping that a second pass would eliminate more of the input noise. This process could continue with more passes until we eliminate all errors and arrive at a final output $\mathbf{y}$ equal to $\mathbf{x}^k$. The retrieval procedure just described amounts to constructing a recurrent associative memory with the synchronous (parallel) dynamics given by

$$x(t + 1) = F[W\, x(t)] \quad (8)$$

Where $t = 0, 1, 2, 3, ...$ and $\mathbf{x}(0)$ is the initial state of the dynamical system which is set equal to the noisy key $\hat{\mathbf{x}}^k$. For proper associative retrieval, the set of memories $\{\mathbf{x}^k\}$ must correspond to stable states (attractors) of the dynamical system. In this case, we should synthesize $\mathbf{W}$ (which is the set of all free parameters w_{ij} of the dynamical system in this simple case) so that starting from any initial state $\mathbf{x}(0)$, the dynamical associative memory converges to the "closest" memory state $\mathbf{x}^k$.

DAM has several variations, which are presented below:

1) Hopfield model

The Hopfield model[6,8] is a distributed model of an associative memory. The *Hopfield Model* was proposed by John Hopfield of the California Institute of Technology during the early 1980s.

The dynamics of the Hopfield model is different from that of the *Linear Associator Model* in that it computes its output recursively in time until the system becomes stable. We presented a Hopfield model with six units, where each node is connected to every other node in the network is given below.

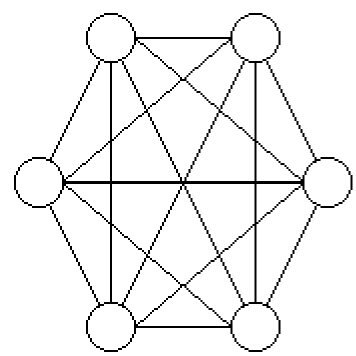

Figure 1: Hopfield model

Unlike the linear associator model which consists of two layers of processing units, one serving as the input layer while the other as the output layer, the Hopfield model consists of a single layer of processing elements where each unit is connected to every other unit in the network other than itself. The connection weight matrix $\boldsymbol{W}$ of this type of network is square and symmetric, i.e., $w_{ij} = w_{ji}$ for $i, j = 1, 2, ..., m$. Each unit has an extra external input I_i. This extra input leads to a modification in the computation of the net input to the units.

Unlike the linear associator, the units in the Hopfield model act as both input and output units. But just like the linear associator, a single associated pattern pair is stored by computing the weight matrix as follows:

$W_k = X_k^T Y_k$, where $Y_k = X_k$

$$W = \alpha \sum_{k=1}^{p} W_k \qquad (9)$$

to store p different associated pattern pairs. Since the Hopfield model is an auto-associative memory model, patterns, rather than associated pattern pairs, are stored in memory.

After encoding, the network can be used for decoding. Decoding in the Hopfield model is achieved by a collective and recursive relaxation search for a stored pattern given an initial stimulus pattern. Given an input pattern $\boldsymbol{X}$, decoding is accomplished by computing the net input to the units and determining the output of those units using the output function to produce the pattern $\boldsymbol{X'}$. The pattern $\boldsymbol{X'}$ is then fed back to the units as an input pattern to produce the pattern $\boldsymbol{X''}$. The pattern $\boldsymbol{X''}$ is again fed back to the units to produce the pattern $\boldsymbol{X'''}$. The process is repeated until the network stabilizes on a stored pattern where further computations do not change the output of the units.

If the input pattern $\boldsymbol{X}$ is an incomplete pattern or if it contains some distortions, the stored pattern to which the network stabilizes is typically one that is most similar to $\boldsymbol{X}$ without the distortions. This feature is called ***Pattern Completion*** and is very useful in many image processing applications.

During decoding, there are several schemes that can be used to update the output of the units. The updating schemes are *Synchronous* (parallel), *Asynchronous (*sequential), or a combination of the two (*hybrid*).

Using the synchronous updating scheme, the output of the units are updated as a group prior to feeding the output back to the network. On the other hand, using the asynchronous updating scheme, the output of the units are updated in some order (e.g. random or sequential) and the output are then fed back to the network after each unit update. Using the hybrid synchronous-asynchronous updating scheme, subgroups of units are updated synchronously while units in each subgroup updated asynchronously. The choice of the updating scheme has an effect on the convergence of the network.

Hopfield (1982) demonstrated that the maximum number of patterns that can be stored in the Hopfield model of m nodes before the error in the retrieved pattern becomes severe is around $0.15m$. The memory capacity of the Hopfield model can be increased as shown by André cut (1972).

Hopfield model is broadly classified into two categories:

- Discrete Hopfield Model
- Continuous Hopfield Model

2) Brain-state-in-a-Box model

The "brain-state-in-a-box"[4,6] (BSB) model is one of the earliest DAM models. It is a discrete-time continuous-state parallel updated DAM. The BSB model extends the Linear Associator model and is similar to the Hopfield Model in that it is an Auto-associative model with its connection matrix computed using outer products in the usual way. The operation of both models is also very similar, with differences arising primarily in the way activations are computed in each iteration, and in the signal function used. The BSB model stands apart from other models in its use of the linear threshold signal function.

3) Bi-directional Associative Memory (BAM)

Kosko (1988) extended the Hopfield model by incorporating an additional layer to perform recurrent auto-associations as well as hetero-associations on the stored memories.

The network structure of the Bi-directional Associative Memory model[4,7] is similar to that of the linear associator, but the connections are bidirectional, i.e., wij = wji, for i = 1, 2, ..., m and j = 1, 2, ..., n. Also, the units in both layers serve as both input and output units depending on the direction of propagation. Propagating signals from the X layer to the Y layer makes the units in the X layer act as input units while the units in the Y layer act as output units. The same is true for the other direction, i.e., propagating from the Y layer to the X layer makes the units in the Y layer act as input units while the units in the X layer act as output units. Below is an illustration of the BAM architecture.

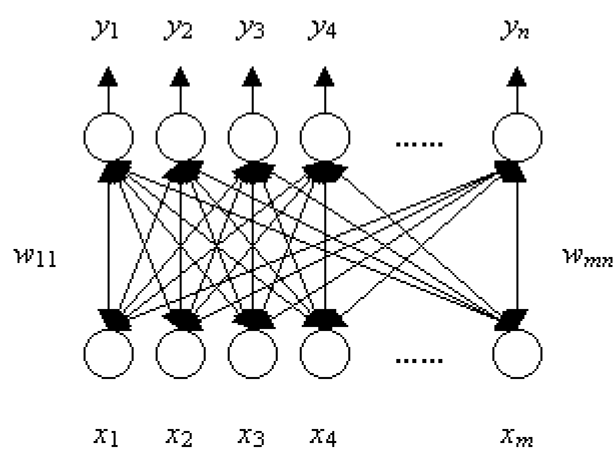

Figure 2: BAM model

Just like the linear associator and Hopfield model, encoding in BAM can be carried out by using:

$$W_k = X_k^T Y_k \qquad (10)$$

to store a single associated pattern pair and

$$W = \alpha \sum_{k=1}^{p} W_k \qquad (11)$$

to simultaneously store several associated pattern pairs. After encoding, the network can be used for decoding.

In BAM, decoding involves reverberating distributed information between the two layers until the network becomes stable. In decoding, an input pattern can be applied either on the X layer or on the Y layer. When given an input pattern, the network will propagate the input pattern to the other layer allowing the units in the other layer to compute their output values. The pattern that was produced by the other layer is then propagated back to the original layer and let the units in the original layer compute their output values. The new pattern that was produced by the original layer is again propagated to the other layer. This process is repeated until further propagations and computations do not result in a change in the states of the units in both layers where the final pattern pair is one of the stored associated pattern pairs. The final pattern pair that will be produced by the network depends on the initial pattern pair and the connection weight matrix.

Several modes can also be used to update the states of the units in both layers namely *synchronous*, *asynchronous*, *and a combination of the two*. In *synchronous* updating scheme, the states of the units in a layer are updated as a group prior to propagating the output to the other layer. In *asynchronous* updating, units in both layers are updated in some order and output is propagated to the other layer after each unit update. Lastly, in synchronous-asynchronous updating, there can be subgroups of units in each layer that are updated synchronously while units in each subgroup are updated asynchronously.

Since the BAM also uses the traditional Hebb's learning rule to build the connection weight matrix to store the associated pattern pairs, it too has a severely low memory capacity. The BAM storage capacity for reliable recall was given by Kosko (1988) to be less than *minimum* (*m*, *n*), i.e., the minimum of the dimensions of the pattern spaces. A more recent study by Tanaka et al (2000) on the relative capacity of the BAM using statistical physics reveals that for a system having *n* units in each of the two layers, the capacity is around 0.1998n.

Mostly, BAM can be classified into two categories:

- Discrete BAM

In a discrete BAM, the network propagates an input pattern ***X*** to the ***Y*** layer where the units in the *Y* layer will compute their net input.

- Continuous BAM

In the continuous BAM, the units use the sigmoid or hyperbolic **tangent** output function. The units in the *X* layer have an extra external input I_i , while the units in the *Y* layer have an extra external input J_j for $i = 1, 2, ..., m$ and $j = 1, 2, ..., n$. These extra external inputs lead to a modification in the computation of the net input to the units.

4) Context Sensitive Auto Associative Memory model (CSAM)

The matrix correlation memories can be very efficiently modulated by contexts in the case in which the key vector and a vectorial context are combined confirming a Kronecker product. The existence of multiplicative contexts enlarges in many directions the cognitive abilities of the correlation memories. One of the abilities of the context-sensitive associative memories is the possibility to implement all the basic logical operations of the propositional calculus. Moreover, these memories are capable of computing some fundamental operations of modal logic. The theory of logic and the theory of context-dependent associative memories converge to operator formalism.

This model is referred to as Context dependent auto-associative memory neural network [2,3], which is more powerful algorithm that suits to compute the clinical and laboratory factors effectively. Here, we could use the Kronecker product matrix as memory representation in the network structure. The model of this algorithm is presented below.

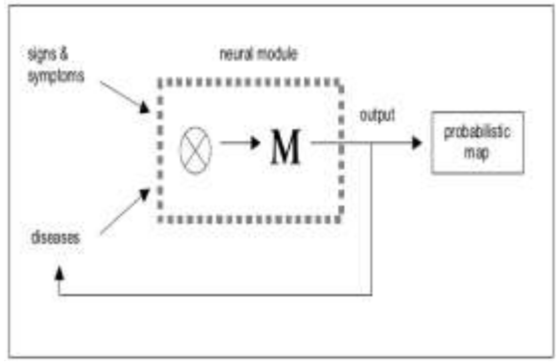

Figure 3: Context-sensitive Auto Associative model

The neural module receives the input of two vectors: one representing the set of possible diseases up to the moment and the other vector corresponding to a new sign, symptom or laboratory result. The action of the neurons that constitute the neural module can be divided into two sequential steps: the Kronecker product of these two entries and the association of this stimulus with an output activity pattern. This output vector is a linear combination of a narrower set of disease vectors that can be reinjected if a new clinical data arrives or can be processed to obtain the probability attribute to each diagnostic decision.

A context-dependent associative memory M acting as a basic expert system is a matrix

$$M = \sum_{i=1}^{k} d_i \left(d_i \otimes \sum_{j(i)} s_j \right)^T \tag{12}$$

Where di are column vectors mapping k different diseases (the set {d} is chosen orthonormal), and sj(i) are column vectors mapping signs or symptoms accompanying the i disease (also an orthonormal set). The sets of symptoms corresponding to each disease can overlap. The Kronecker product between two matrices A and B is another matrix defined by

$$A \otimes B = a(i, j).B \tag{13}$$

Denoting that each scalar coefficient of matrix A, a(i, j), is multiplied by the entire matrix B. Hence, if A is nxm dimensional and B is k x l dimensional, the resultant matrix will have the dimension nk x ml.

5) Context Sensitive Asynchronous Memory Model (CSYM)

Context-sensitive asynchronous memory[10,11] is a priming-based approach to memory retrieval. It exploits feedback from the task and environment to guide and constrain memory search by interleaving memory retrieval and problem solving. Solutions based on context-sensitive asynchronous memory provide useful answers to vague questions efficiently, based on information naturally available during the performance of a task.

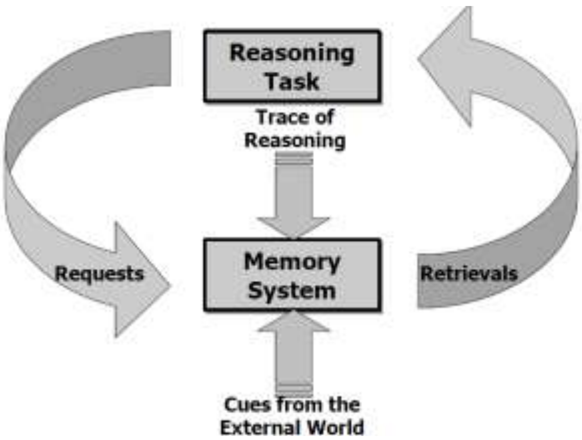

Figure 4: *A Context Sensitive Associative memory*

Reasoning is an important biological activity. Emerged at some point of biological evolution, reasoning is a neural function with a decisive role for the kind of life style that the human beings have developed. The knowledge of the external world, in fact one of the highlights of human culture, is a direct consequence of the capacity of reasoning.

A context-sensitive asynchronous memory differs from these "traditional" approaches by simultaneously providing a means to allow reasoning to proceed in parallel with memory search as well as a means to guide ongoing memory search.

- Asynchronous retrieval**:** Asynchronous retrieval is the autonomous processing of memory retrieval requests. A memory system can perform asynchronous retrieval by using reified retrieval requests and a retrieval monitor which operates in conjunction with the agent's task controller. Asynchrony logically includes spontaneous retrieval (retrieval at the discretion of memory) and anytime retrieval (retrieval on demand of the reasoner).

- Context-sensitivity**:** Context sensitivity is using feedback to guide memory search. Context sensitivity can be achieved through a process, called context-directed spreading activation, which operates hand in hand with the agent's working memory.

6) Holographic Associative Memory (HAM)

In 1990 Sutherland in his pioneering work, presented the first truly holographic associative memory[20] with holographic representation and learning algorithm analogous to correlation learning. It is a two-dimensional (2-D) generalized multidimensional phased representation.

Here information is mapped onto the phase orientation of complex numbers operating. It can be considered as a complex valued artificial neural network. Generally speaking, holographic networks are very suitable for those problems where stimuli are long vectors with symmetrically (uniformly) distributed arguments. A longer stimulus vector assures a greater learning capacity, i.e. a greater number of stimulus-response associations that can be learned. Symmetry in arguments assures accuracy in reproducing the learned stimulus-response associations.

Holographic memory is a storage device that is being researched and slated as the storage device that will replace hard drives and DVDs in the future. It has the potential of storing up to 1 terabyte or one thousand gigabytes of data in a crystal the size of a sugar cube.

Advantages of Holographic Memory Systems

Aside from having a tremendous amount of storage space for data, holographic memory systems also have the ability to retrieve data very quickly, up to a 1 gigabyte per second transfer rate

The main difference between holographic and conventional neural networks is that a holographic neuron is more powerful than a conventional one, so that it is functionally equivalent to a whole conventional network. Consequently, a holographic network usually requires a very simple topology consisting of only few neurons. Another characteristic of the holographic technology is that it represents information by complex numbers operating within two degrees of freedom (value and confidence). Also an important property is that holographic training is accomplished by direct (almost non-iterative) algorithms, while conventional training is based on relatively slow "back-propagation" (gradient) algorithms. A *holographic neuron* is sketched in *Figure 5* . As we can see, it is equipped with only one input channel and one output channel. However, both channels carry whole vectors of complex numbers. An input vector S is called a stimulus and it has the form

$$S = \left[\lambda_1 e^{i\theta_1}, \lambda_2 e^{i\theta_2}, \lambda_3 e^{i\theta_3}, \ldots\ldots, \lambda_n e^{i\theta_n}\right] \qquad (14)$$

An output vector R is called a response and its form is

$$R = \left[\gamma_1 e^{i\phi_1}, \gamma_2 e^{i\phi_2}, \ldots\ldots\ldots, \gamma_m e^{i\phi_m}\right] \qquad (15)$$

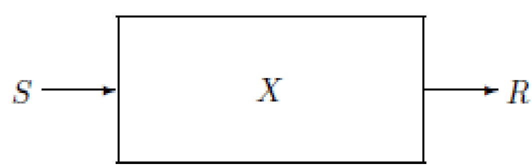

Figure 5: *A Holographic neuron*

All complex numbers above are written in polar notation, so that moduli (magnitudes) are interpreted as confidence levels of data, and arguments (phase angles) serve as actual values of data. The neuron internally holds a complex $n \times m$

matrix $X = \lfloor x_{jk} \rfloor$ which serves as a memory for recording associations.

We present the *basic learning process*. Learning one association between a stimulus *S* and a desired response *R* requires that the correlation between the *j*-th stimulus element and the *k*-th response element is accumulated in the (*j; k*)-th entry of the memory matrix. More precisely:

$$x_{jk} += \lambda_j \gamma_k e^{i(\phi_k - \theta_j)} \qquad (16)$$

The same formula can be written in the matrix-vector form:

$$X += \bar{S}^T R \qquad (17)$$

Here $\bar{S}$ denotes the conjugated transpose of the vector *S*.

Mostly Holographic memory is of Different models, but we are considering two of them;

- Dynamically Structured Holographic Associative Memory (DSHAM).
- Composite Holographic Associative Memory (CHAM).

III. Applications

A. Human Resources Management

1) Employee Selection and Hiring - predict on which job an applicant will achieve the best job performance.

Input data: information about an applicant: personal information, previous jobs, educational levels, previous performance, etc.

2) Employee Retention - identify potential employees who are likely to stay with the organization for a significant amount of time based on data about an applicant.

Input data: applicant's hours of availability, previous jobs, educational levels and other routine information.

3) Staff Scheduling - predict staff requirements for restaurants, retail stores, police stations, banks, etc.

Input data: time of year, day of week, pay-days, holidays, weather, etc.

4) Personnel Profiling - forecast successful completion of training program; identify employees most suitable for a certain task.

Input data: background characteristics of individuals.

B. Medical

1) Medical Diagnosis - Assisting doctors with their diagnosis by analyzing the reported symptoms.

Input data: patient's personal information, patterns of symptoms, heart rate, blood pressure, temperature, laboratory results, etc.

2) Detection and Evaluation of Medical Phenomena - detect epileptic attacks, estimate prostate tumor size, detects patient breathing abnormalities when a patient is under anesthesia, etc.

Input data: patient's personal information, breathing rate, heart rate, patterns of symptoms, blood pressure, temperature, etc.

3) Patient's Length of Stay Forecasts - forecast which patients remain for a specified number of days.

Input data: personal information such as age and sex, level of physical activity, heart rate, blood pressure, temperature and laboratory results, treatment procedures, etc.

4) Treatment Cost Estimation

Input data: personal information such as age and sex, physiological data, the use of drug or other therapies, treatment procedures, number of recurrences after first treatment, etc.

C. Financial

1) Stock Market Prediction - predict the future movement of the security using the historical data of that security.

Input data: Open, High, Low, Close, Volume, technical indicators, market indexes and prices of other securities.

2) Credit Worthiness - decide whether an applicant for a loan is a good or bad credit risk.

Input data: applicant's personal data, income, expenses, previous credit history, etc.

3) Credit Rating - assign credit ratings to companies or individuals based on their financial state.

Input data: current financial state indicators and past financial performance of a company or individual.

4) Bankruptcy prediction - classify a company as potential bankruptcy.

Input data: company characteristics and business ratios, such as working capital/total assets, retained earnings/total assets, earnings before interest and taxes/total assets, market value of equity/total debt, and sales/total assets.

5) Property Appraisal - evaluate real estate, automobiles, machinery and other property.

Input data: property parameters, environment conditions as well as appropriate demographic, ecological, industrial and other factors.

6) Fraud Detection - detect and automatically decline fraudulent insurance claims, client transactions, and taxes.

Input data: transaction parameters, applicant's information and other data of past incidents.

7) Price Forecasts - forecast prices of raw materials, commodities, and products.

Input data: previous price movements, economical indicators, market indexes.

8) Economic Indicator Forecasts - forecast economic indicators for the next week, month, and quarter.

Input data: social and economical indicators, time-series data of an indicator.

D. Sales and Marketing

1) Sales Forecasting - predict future sales based on historical information about previous marketing and sales activities.

Input data: historical data about marketing budget, number of ads, special offers and other factors affecting sales.

2) Targeted Marketing - reduce costs by targeting a particular marketing campaign to the group of people which have the highest response rate. Avoid wasting money on unlikely targets.

Input data: information about customers and their response rate.

3) Service Usage Forecasting - forecast the number of service calls, customer transactions, customer arrivals, reservations or restaurant covers (patrons) in order to effectively schedule enough staff to handle the workload.

Input data: season, day-of-week, hour of the day, special events in the city/area, marketing budget, promotional events, weather, etc.

4) Retail Margins Forecasting - forecast the behavior of margins in the future to determine the effects of price changes at one level on returns at the other.

Input data: retail prices, expenditures at the retail level, marketing costs, past margin values, price variability and other market characteristics.

E. Industrial

1) Process Control - determine the best control settings for a plant. Complex physical and chemical processes that may involve interaction of numerous (possibly unknown) mathematical formulas can be modeled heuristically using a neural network.

2) Quality Control - predict the quality of plastics, paper, and other raw materials; machinery defect diagnosis; diesel knock testing, tire testing, beer testing.

Input data: product/part/machinery characteristics, quality factor.

3) Temperature and force prediction in mills and factories

Input data: previous values of temperature, force and other characteristics of mills and factories.

F. Operational Analysis

1) Retail Inventories Optimization - forecast optimal stock level that can meet customer needs, reduce waste and lessen storage; predict the demand based on previous buyers' activity.

Input data: characteristics of previous buyers' activity, operating parameters, season, stock, and budgets.

2) Scheduling Optimization - predict demand to schedule buses, airplanes, and elevators.

Input data: season, day-of-week, hour of the day, special events in the city/area, Weather, etc.

3) Managerial decision making - select the best decision option using the classification capabilities of neural network.

Input data: initial problem parameters and final outcome.

4) Cash flow forecasting - maximize the use of resources with more accurate cash flow forecasts.

Input data: accounts payable, accounts receivable, sales forecasts, budgets, capital expenditures, stock, season, operating data, etc.

G. Data Mining

1) Prediction - use some variables or fields in the database to predict unknown or future values of other variables of interest.

2) Classification - map (classify) a data item into one of several predefined classes.

3) Change and Deviation Detection - uncover certain data records that are in some way out of the ordinary records; determine which cases/records suspiciously diverge from the pattern of their peers.

4) Knowledge Discovery - find new relationships and non-obvious trends in the data.

5) Response Modeling - build a neural network based response model.

6) Time Series Analysis - forecast future values of a time series.

IV. CONCLUSION

We presented a survey on various associative memories which are being used in various applications in recent trends. There are few points that we would like to mention through this article:

Memory is not just a passive store for holding ideas without changing them; it may transform those ideas when they are being retrieved. There are many examples showing that what is retrieved is different from what was initially stored.

Simple Associative memories are static and very low memory so that they cannot be applied in the applications where high memory is required.

Dynamic Associative memories such as Hopfield, BSB, and BAM are Dynamical memories but they are also capable of supporting very low memory, so they cannot be applied in the applications where high memory requirements are there.

A simple model describing context-dependent associative memories generates a good vectorial representation of basic logical calculus. One of the powers of this vectorial representation is the very natural way in which binary matrix operators are capable to compute ambiguous situations. This fact presents a biological interest because of the very natural way in which the human mind is able to take decisions in the presence of uncertainties. Also these memories could be used to develop expert agents to the recent problem domain.

Holographic memories are being used to build the many advanced memory based agents like memory cards, USB Drives, etc., Context asynchronous memories are being used to develop experience based agents. Context-sensitive asynchronous memory enables agents to efficiently balance resources between task processing and memory retrieval. By incrementally searching the knowledge base, a context-sensitive asynchronous memory supports anytime retrieval of the best answer found so far and thus enables agents to satisfy.

Finally, we conclude that Context Sensitive Auto Associative memory and Asynchronous Memory and Holographic Associative Memory can be used to solve the real applications which we mentioned.

V. FUTURE SCOPE

Among these memories we have used Context sensitive auto-associative memory model to implement the expert system for medical diagnosis of Asthma. We are planning to implement the medical based expert systems with the use of Context Sensitive Asynchronous Memory Models (CSYM) and also to implement mining based applications including time series analysis.

REFERENCES

[1] "Fundamentals of Artificial Neural Networks" – Mohadmad H. Hassoun.

[2] "A Comparative Study of Machine Learning Algorithms as Expert Systems in Medical Diagnosis (Asthma)" – Dr. B D C N Prasad, P E S N Krishna Prasad and Y Sagar, CCSIT 2011, Part I, CCIS 131, pp. 570–576, 2010.

[3] "Context-Sensitive Auto-Associative Memories as Expert Systems in Medical Diagnosis" – Andres Pomi and Fernando Olivera, BioMed Central.

[4] "Neural Networks, A Classroom Approach" – Satish Kumar.

[5] "The Moral Demands of Memory" - Jeffrey Blustein.

[6] "Neural Network Design" – Martin T. Hagan, Howard B. Demuth, Mark Beale.

[7] "Neural Networks and Fuzzy Systems - A dynamical Systems Approach to Machine Intelligence" – Bart Kosko.

[8] Anderson JA, Cooper L, Nass MM, Freiberger W, Grenander U:Some properties of a neural model for memory. AAAS Symposium on Theoretical Biology and Biomathematics 1972 [http://www.physics.brown.edu/physics/researchpages/Ibns/Cooper%20 Pub040_SomePropertiesNeural_72.pdf]. Milton, WA. Leon N Cooper Publications.

[9] "An Introduction to Neural Networks" – James A. Anderson.

[10] "Neural Networks Ensembles, Cross Validation and Active Learning" – Anders Krogh and Jesper Vedelsby, MIT Press[1995].

[11] Mizraji E: Context-dependent associations in linear distributed memories. Bulletin Math Biol 1989, 51:195-205.

[12] Valle-Lisboa JC, Reali F, Anastasía H, Mizraji E: Elman topology with sigma-pi units: An application to the modelling of verbal hallucinations in schozophrenia. Neural Networks 2005,18:863-877.

[13] Mizraji E, Pomi A, Alvarez F: Multiplicative contexts in associative memories. BioSystems 1994, 32:145-161.

[14] Pomi-Brea A, Mizraji E: Memories in context. BioSystems 1999,50:173-188.

[15] Mizraji E, Lin J: A dynamical approach to logical decisions. Complexity 1997, 2:56-63.

[16] Mizraji E, Lin J: Fuzzy decisions in modular neural networks. Int J Bifurcation and Chaos 2001, 11:155-167.

[17] Pomi A, Mizraji E: A cognitive architecture that solves a problem stated by Minsky. IEEE on Systems, Man and Cybernetics B (Cybernetics) 2001, 31:729-734.

[18] "Context-Sensitive Asynchronous Memory"- Anthony G. Francis, Jr.

[19] "Associative Memories in Medical Diagnostic"- Jorge M. Barreto, Fernando de Azevedo, Carlos I. Zanchin, Lycia R. Epprecht.

[20] "Context-Sensitive Associative Memory: Residual Excitation" in Neural Networks as the Mechanism of STM and Mental Set - Victor Eliashberg

[21] "Retrieving Semantically Distinct Analogies" - Michael Wolverton.

[22] "Improving Array Search Algorithm Using Associative Memory Neural Network" - Emad Issa Abdul Kareem and Aman Jantan.

[23] "Hippocampal Auto-Associative Memory" - Nicolas P. Rougier.

[24] "Phrase Detection and the Associative Memory Neural Network" - Richard C. Murphy.

[25] "Human Recognition in Passive Environment using Bidirection Associative Memory" - Vivek Srivastava and Vinay kumar Pathak.

[26] "Generalized Asymmetrical Bidirectional Associative Memory" - Tae-Dok Eom, Changkyu Choi, and Ju-Jang Lee.

[27] UCI Machine Learning Repository:

[28] "Characteristics of MultiDimensional Holographic Associative Memory in Retrieval with Dynamically Localizable Attention" – Javed I . Khan.

[29] "Bidirectional Associative Memory Neural Network method in the Character Recognition" - Yash Pal Singh, V.S.Yadav, Amit Gupta, Abhilash Khare.

[30] Cross SS, Harrison RF, Lee Kennedy R: Introduction to neural networks. The Lancet 1995, 346:1075-1079.

[31] E. Oja, "A simplified neuron model as a principal component analyzer," J. Math. Biol., vol. 15, pp. 267–273, 1982.

[32] D. Psaltis and F. Mok, "Holographic Memories," Sci. Amer., Nov. 1995, pp. 70–76.

[33] H.-M. Tai and T. L. Jong, "Information storage in high-order neural networks with unequal neural activity," J. Franklin Inst., vol. 327, no.1, pp. 16–32, 1990.

[34] M. Wenyon, Understanding Holography. New York: Arco, 1978.

[35] B.Widrow and M. E. Hoff, "Adaptive switching circuits," IRE WESCON Convention Rec., pt. 4, pp. 96–104, 1960.

[36] D. Willshaw, "Holography, associative memory and inductive generalization," in Parallel Models of Associative Memory, G. E. Hinton and J. E. Anderson, Eds. Hillsdale, NJ: Lawrence Erlbaum, 1985.

[37] McKerrow, P.J. (1991). Introduction to Robotics. Sydney, Australia: Addison-Wesley.

[38] McNamara, T.P. (1992). Priming and constraints it places on theories of memory and retrieval. Psychological Review, 99, 650-662.

[39] McNamara, T. P., & Diwadkar, V. A. (1996). The context of memory retrieval. Journal of Memory and Language, 35, 877-892

[40] Moore, A.W., & Atkeson, C.G. (1995) Memory based neural networks for robot learning. Neurocomputing, 9(3) pp. 243-269.

[41] http://archive.ics.uci.edu/ml/machine-learning-databases/

AUTHORS PROFILE

Dr. B D C N Prasad, currently is a Professor & Head of Department of Computer Applications at Prasad V. Potluri Siddardha Institute of Science and Technology, Vijayawada, Andhra Pradesh, India. He received Ph.D. in Applied Mathematics from Andhra University, Visakhapatnam,India in 1984. His research interests includes Machine Intelligence, Data Mining, Rough Sets and Information Security in Computer Science and Boundary value problems and Fluid Dynamics in Mathematics. He has several publications in mathematics and computer science in reputed national and international journals. He is a member of ISTAM , ISTE and also he is a national executive member of Indian Society for Rough Sets.

Mr. P E S N Krishna Prasad, currently is a Research Scholor under the guidance of Dr. BDCN Prasad in the area of Machine Intelligence and Neural Networks. He is working as Associate Professor in the Department of CSE, Aditya Engineering College, Kakinada, Andhra pradesh, India. He is a member of ISTE. He has presented and published papers in several national and International conferences and journals. His areas of interest are Artificial Intelligence, Neural Networks and Machine Intelligence.

Mr. P Sita Rama Murty, currently is a Research Scholor, in the area of ATM networks and Information Secuirty. He is working as Assistant Professor in the department of CSE, Sri Sai Aditya Institute of Science and Technology, Kakinada, Andhra Pradesh, India