 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

A Proposal and Implementation of a Neural Network Based Hierarchical Temporal Memory to Realize Cognitive Functions

[1]Rajesh Babu, [2]Dr. Jitendra Kumar
[1]Rakshpal Bahadur Management Institute, Bareilly, UP, India
[2]Drownacharya Engineering College, Gurgaon, Haryana, India

Abstract
Hierarchical Temporal Memory (HTM) is a recent innovation in cognition science. Developed in 2005 by Numenta Inc., an artificial intelligence research firm in the US, HTMs attempt to capture the way the human brain learns and infers its environment. One of the most notable characteristics of this model is the consideration of the hierarchical organization of objects in the world. Data in the world is made up of elementary features that aggregate in successive layers to form perceivable objects. This data can be visual, auditory or from other abstract spaces such as stock markets and scientific studies. The amount of raw data that the brain is exposed to throughout its lifetime is beyond imagination. However the brain is known to use a very noble and systematic approach to handle the perception, storage, and inference of this data. Several studies in neuroscience and psychology indicate that the brain makes use of the hierarchies that features in the world exhibit in their organization to form objects. Hence, for instance, 'corners' and 'lines' can aggregate to form a 'table' object in the visual world. These elementary features, however, can use a different aggregation to form a 'chair' object. The same is true for data in other types of worlds such as audio. HTMs directly apply a similar handling of world data for their cognition. Furthermore, the structure of HTMs, made up of data processing nodes arranged in a hierarchical tree, mimic the physical arrangement of cortical layers in the brain.

Keywords
Artificial Intellegence, HTM, Newal Network, Cognitive Function

I. Introduction

"The study and design of intelligent agents" is the modern definition of artificial intelligence (AI) where an 'intelligent agent' is a system that perceives its environment and takes actions which maximizes its chances of success [28]. The philosophical roots of AI date back as early as ancient Greece. The development of AI has been strongly linked to the study of logic, mathematics, statistics, neuroscience, as well as the improvement of computer hardware and programming languages.

Within the current framework of study various streams of study exist. One such study, symbolic AI incorporates cognitive simulation, logical AI and knowledge based AI. Sub symbolic AI, with pattern recognition and the concept of learning, covers a range of 'computational intelligence' topics. Other more recent approaches include the intelligent agent paradigm and intelligent agents. Other schools of philosophy tend to categorize AI into 'strong AI' that addresses systems capable of general level intelligence and 'weak AI' with systems that exhibit intelligence in a predetermined area of purpose [28].

In recent times AI development has steadily gained attention as the number of possible application scenarios and the power of computational platforms grow. Large research projects dedicated to studying the subject for its own sake rather than for practical purposes are very common. Apart from the analytical and theoretical research there has also been a huge flow of resources into developing powerful platforms. Most notable is the IBM blue gene project which is used to simulate brain functions [26-27]. It seems indispensable to acquire a competent platform to run these systems if a truly intelligent machine needs to be implemented.

A. The Science of Cognition

'Cognition' is a scientific term that draws significance from the natural processes of perception, memory, recognition, planning, motor behavior, and is taken as a manifestation of intelligence in natural entities [25]. However the term is nowadays being widely applied in the science of AI as well.

Several schools of thought and modeling paradigms exist within the study of cognition as applied to AI. Among these the emergent models are approaches embracing connectionist systems, dynamical systems, and enactive systems, all based to a lesser or greater extent on principles of self-organization. In particular connectionist systems involve an interactive set of dynamically adapting units and include those based on neural networks, which are a subject of this work.

Connectionist systems rely on parallel processing of non-symbolic distributed activation patterns using statistical properties, rather than logical rules, to process information and achieve effective behavior. Biological neural networks (BNNs) are considered to be the main inspirations behind artificial neural networks (ANNs), which are at the forefront of the connectionist model.

Neural networks have emerged in different structures and working principles. Self organizing maps (SOM), radial basis functions (RBF), the Boltzmann machine and the perceptron model are among the popular ones [23-24]. The most widely used model of neural network, however, is the multi-layer perceptron (MLP). Even though various irregular structures and operating modes exist for the MLP, the regular fully connected structure with a feed forward operation is the default configuration for different applications. Given the proper topology and operational parameters, the MLP has the ability to cope with extended input set size and high precision in approximating untrained input patterns to nearest outputs. This has made it the subject of research in various recognition applications.

B. Literature Survey of Cognitive Models

Due to the relatively recent nature of the study and the lack of a general consensus on many of its philosophical and operational theories, AI is one of few areas of science that is documented with mainly hypothetical and contradicting publications [2].

The research in cognitive models is no exception. In order to assess the relative advantages of each model and the feasibility of merging two of the models, namely neural networks from the connectionist pact [12,23] and HTMs from the hierarchical ones, a fairly wide review of many of the models was conducted.

Hierarchical modeling of cognitive structures had its own emergence with first links to non cognitive computing applications [4]. Such works involved the modeling of a dynamic memory environment to implement in machines of fast access cache architecture in specific applications. Although the research is limited to computational platforms it demonstrated some of the shortcomings of monotonous storage architectures. As an alternative to artificial cognition models, the theory of hierarchy did not emerge until recent years. In the paper, brain-inspired conscious computing architecture [10], the author attempts to model the brain with a very large topology neural network model with no sub-structuring of hierarchical type or otherwise, while others [16-17] try to demonstrate the concepts of feedback in neural networks when applied to visual recognition problems with a simplified circuit as a building block of a larger monotonous architecture.

Neural networks themselves, even though having an older history, have passed several application attempts in cognitive models. A description of one model [9] addresses prefrontal visual cortex of the brain as a model for a visual cognitron by a direct mapping of the working memory structures of the neurons. Another similar research [7-8] also attempts to directly map the physical structures of neuronal circuits in the brain to an artificial memory system. These researches base their motivation on the notion of the natural brain as the most efficient cognitive memory architecture currently available.

The mapping of the brain in functional behavior rather than structure did not get attention until the recent work of Dileep George and Jeff Hawkins, invariant pattern recognition using Bayesian inference on hierarchical sequences (2004) [13]. The theory is a Bayesian inference model of cognition employing a hierarchical chain of decisions. It did not develop into the more robust theory of hierarchical temporal memory until late 2005 [18, 20].

Other works involved employing neural network structures in non-monotonous organization, albeit not truly hierarchical. Mostefa et al. [1], Lin et al. [11], Miikkulainen et al [5] try to model forms of cognition with different architectures that involve the use of neural networks as an inference engine.7

It is notable also that an evolution based neural network model has been explored by Garis et al. [3] as a research into a task survival based evolution of a potential artificial brain. As a most recent development, hierarchical temporal memory [19] expand the theory of hierarchy in terms of functional modularity as well as storage. This model provides a direct match to the natural process of cognition in an intuitive manner. The concept of hierarchical storage of world data features and their aggregations to attain an efficient storage and retrieval system has gained strong acclaim from researchers in the field. Furthermore a functional analogy of the cortical layers of the brain and this model has been observed [15].

II. Hierarchical Temporal Memory

Hierarchical Temporal Memory (HTM) is a machine learning (artificial cognition) model developed by Jeff Hawkins and Dileep George of Numenta, inc. That models some of the structural and algorithmic properties of the neocortex as Bayesian networks [15]. It is widely considered a novel approach for proposing functions of the cortical layers. It is considered by Numenta as a new computational paradigm based on cortical theory.

A. Background Concepts

It is a long discovered fact that the neocortex, the part of the brain responsible for cognitive functionalities, works on a common algorithm for tasks that vary from vision to language [8]. A single algorithm is used in the classification of sensory input data from different types of biological sensors. Thus the various traits of behavior and perception are processed in the neocortex as a single type of input data and only maintain their unique processing in the perception or motor areas of the brain allocated to the specific sensor. HTMs share this important characteristic of the biological system. The very concept of data classification in HTMs depend on 'cause discovery'. This is to mean that every separate input to the HTM network is considered to have its own cause. The input, thus, is taken as part of a larger input that has not been fully sensed yet by the particular unit that senses this input. This dictates the absolute necessity of time in learning causes and comprises the 'temporal' nature of this model. Data inputs that consistently occur closer together in time (i.e. One after the other) are taken as being part of a larger cause. This corresponds directly to reality where parts of a 'cause' occur sequentially in time to the receptor. Input patterns that occur in spatially close formation (e.g. One besides the other) are also considered to have a special relationship with each other. This other concept comprises the 'spatial' nature of the model. Hence the model synchronizes the spatio-temporal nature of the real world in a single cognition framework.

Another key concept of the model is the concept of 'hierarchy'. In the real world objects (visual, auditory, or any other) can be observed to have a hierarchy of sub-objects that are shared among higher level objects. For instance a simple diagram of a car and a house may share lower level objects such as horizontal or vertical lines. Hence the upper level objects share these lower level objects during learning or inferencing by the HTM unit. In the actual world, however, this hierarchical composition of objects may be much more complex. In addition other input data type such as audio may not have an easily readable cause-object relationship. This 'cause sharing' of higher level objects from lower level ones results in the storage efficiency of HTMs. It is worthy to note also that the biological brain has also been a center of research in the area of storage as the amount of data it emulates as storing and its sheer size and retrieval speeds do not appear to match.

Mapping the exact 'parent-child' looking relationship of real world objects, HTMs are modeled in a tree shaped hierarchy of nodes, with each node having its own spatial and temporal classification functionalities and operate with the same algorithm as the others. Each node has input and output vectors. The input vector is fed to the node for spatio-temporal analysis and the result is written on the output vector. Nodes at the lower level of the hierarchy, i.e. Leaf nodes, are the first stages of reception for world data. In a visual task, for instance, they can be fed row pixel data of the sensed object. The nodes in the level immediately above the receptors are fed the outputs of two or more child (receptor) nodes depending on the tree architecture. These nodes in turn perform spatio-temporal analysis on their input using the same generic algorithm and output the result for higher level nodes. Hence as one goes up the hierarchy nodes classify higher and higher types of objects (e.g. From pixels, to lines, to corners and to shapes in visual case). The most important thing in HTMs is that the causes discovered by each node is not pre-determined or programmed by the designer of the application.

III. Current Implementation

The current developer of HTMs, Numenta Corporation, has a basic implementation of the concept in each of the identical nodes of the HTM tree [18].

Each separate computing unit of HTMs, called node, has two distinct parts and two main operations corresponding to these parts. The first section, the spatial pooler, is responsible for accepting input data and making spatial classifications based on the pattern. The term 'spatial' should not be misleading in that the pooler is useful for visual input only; rather it means the position of the input pattern's elements with respect to each other. The spatial pooler uses a distance metric computation to classify distinct spatial patterns. Hence a greater distance metric value tends to gather more patterns into a single group, whereas a smaller one assigns each different pattern its own classification. This technique is used to deal with noisy inputs. After classifying input patterns, the pooler begins generating outputs for each incoming input based on the classified categories. The output of the spatial pooler is a vector set of beliefs as to where the current input is classified. Hence a normalized belief vector is written on the output of the spatial pooler which in turn in is the input of the next stage of the node.

The next stage, the temporal pooler, undertakes a more complex operation. The basic job of this pooler is to classify the categories generated by the spatial pooler into a set of temporally adjacent clusters and make temporal inference based on these clusters. Classification decisions are made based on the consistence of a pattern in appearing close in time to another pattern. For this purpose various data structures are employed, the main one being the temporal matrix. This matrix maintains a record of all the patterns the pooler came across and the number of appearance of each pattern after the pattern that appeared previously. At each arrival of a pattern the specific entry in the matrix that corresponds to the arriving pattern row which is under the previously appearing pattern column is incremented with a chosen value. For better performance a non linear priority function is used to increment values of these entries in precedence of late appearance.

After the construction of the matrix is complete the pooler runs a clustering algorithm on the matrix to classify the categories into groups. This classification operation results in groups of patterns that appear closer in time to each other more than the rest. These groups will serve as easy references upon inference operation of the node in determining which feature of an input space is currently being observed. In a visual perception application, or instance, the patterns are classified into clusters of 'vertical line', 'horizontal line', 'lower left corner' and so on. This is possible because the patterns in each of these clusters are more likely to appear closer in time than the patterns in the rest of the clusters. The algorithm for this clustering is the most important one in determining the performance of the node since these clusters provide the means for position/focus/orientation (or all, depending on the desired purpose) invariance to the categorization. Hence a' line' will be classified as a 'line' no matter what position or which orientation the perceived line has.

After training of the node is over, it is said to be ready for inference (recognition). The spatial pooler has to be trained first and put into its own inference mode before the temporal pooler is trained. In the HTM hierarchy lower levels are trained before upper (parent levels). When a level has finished training it is put into inference mode and serves as an input generator for its parent level, which in turn is put into training mode. A number of optimization techniques (or shortcuts) can be employed for faster performance of the network training. Node cloning is used to clone a trained node to all other nodes in the level instead of training each one in turn.

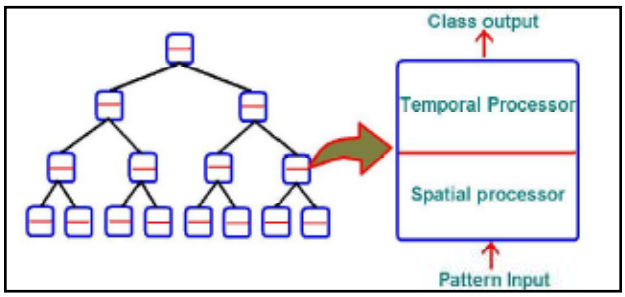

Fig. 1: Implementation of the HTM Network

A. Limitations

As is discussed in the structure and operations of the current HTM implementation, the model is made up of several processing and storage areas to analyze, classify, store and infer patterns. All of these operations are computation based routines that search and match patterns according to explicit criterions and appearance history.

The HTM model is noble in providing autonomy to a cognitive engine in that no pattern cluster organization or type of members in a cluster are predefined or guided by the designer. In contrast the natural adjacency and frequency of appearance of patterns (and hence features) in the real world are guides to the formation of these clusters. This leads not only to the discovery of distinctive features of objects but also the hierarchical relationship between them, resulting in a tremendous amount of conserved memory and efficient operation.

However this autonomous characteristic remains limited in the outermost topology of the network. The internal structure of the nodes themselves is a pure computation based implementation. Patterns are analyzed for elements, discriminated for noise with well known vector dissimilarity relations, stored in tables, and searched for matches. This approach has a few serious limitations.

1. Problem in Dealing with Untrained Patterns

Computational algorithms in general rely on explicit associations of data to make decisions. Upon encountering unforeseen data, they tend to introduce error or default to a preset output [22]. The current HTM nodes make use of the spatial noise discriminator functionality, which is designed to discriminate against random noise elements limited in extent, to deal with unforeseen data as well. A Bayesian belief output is computed for the top categories. When the encountered data deviates beyond the threshold of the discrimination function for the beset match, an error signal is generated and the object is tagged as 'unknown' in the output. One may argue that increasing the threshold value may solve the problem. However the purpose of the discrimination function is to filter out random noise and increasing the threshold, in anticipation of unforeseen variants of patterns, results in the clustering together of unrelated patterns. Thus at best, the implementation is a tradeoff between noise intolerance and feature approximation.

B. Inconvenience of Node Algorithms for Parallel Running Machines

Apart from the requirements of explicit data association in the algorithms and their problem in dealing with unpredictable patterns, the search-and-match routines make the current node implementation a bottleneck in exploiting parallel running systems that are expected to be the main platforms in future cognition

modeling. Present research in such machines reveals that the design tendency is towards systems made up of a large number of processing nodes running in parallel and possibly in hierarchical organization [27].

Although single nodes of the HTM can be assigned to such processing nodes of these machines and exploit their powers, the true advantages may not be harnessed until single processes in each HTM node can run in data independence manner with other processes and assigned its own processing resources.

Hence the researcher argues that the current HTM model is limited in its ability to be a future model for implementing in such dedicated systems and a further re-organization of the node structures is required to meet this goal.

IV. Proposed Solution

As indicated, the proposed solution for this problem in this work is a hybrid model of the HTM and the well known connectionist model of neural networks. Thus, the original HTM model itself was found to be the best match to the hierarchical organization of objects in the modeled world while neural networks are known to have the advantages of parallel architectures and better output approximations.

Pattern classification is done primarily in the spatial processor of the HTM nodes. The current implementation of the node structures is illustrated below in fig. 2.

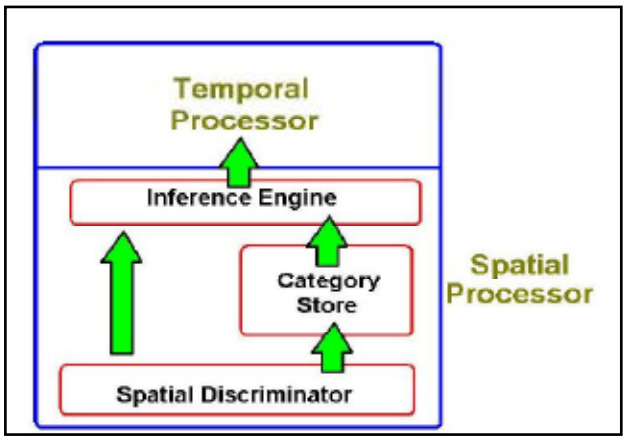

Fig. 2: Functional Structures of the Spatial Processor in the Current HTM Node

The inference engine is the main component of the processor in the task of pattern inference. It compared and matches patterns that arrive with table entries in the category store and forward an inference decision to the temporal processor. This makes it the responsible component in dealing with uncategorized data that comes from the discriminator at inference time. As such it has been the main focus of enhancement in the form of neural network data structures. Neural networks by nature are not only classifier engines but also store the learned data in the form of weight values between the neurons. This eliminates the need for the category store as well. The proposed model is shown below in fig. 3.

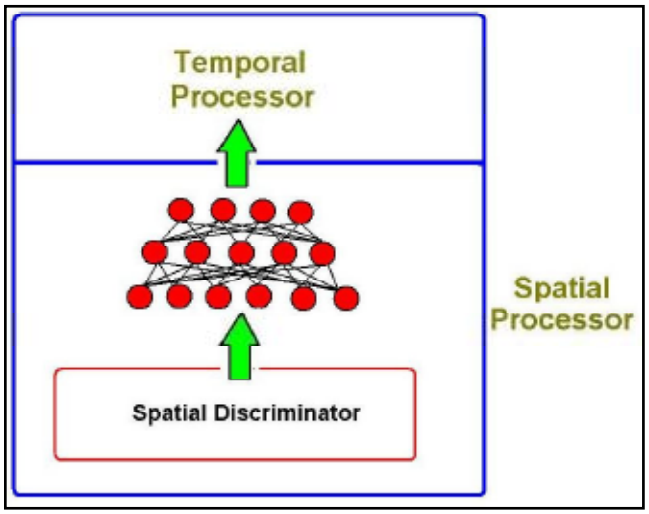

Fig. 3: Proposed Modifications to the Spatial Processor of the HTM Node

The fig. 3, illustrates the replacement of two of the components of the spatial processor in the node. The neural network, when the node is in training mode, is trained for the input patterns and their associated representative category outputs. Thus the various spatial pattern both with and without noise are stored as weight values in the network. When the node is in inference mode, the network is provided with inputs and its output is directly fed to the temporal processor part of the node. In this mode, unforeseen input patterns such as patterns with noise are approximated to the nearest matching output categories.

The output approximation of neural networks is expected to offer superior results in comparison to any other analytic computation approaches such as the Bayesian belief output comparison used in the original model by Numenta.

V. Conclusion and Recommendations

From the results that emerged from the analysis of both the operational characteristics and output performance of the proposed model, it is evident that encouraging improvements were achieved by merging neural networks with a structured model of cognition. Although some more optimization remains to be done with regard to the stability and storage capacity of the MLP neural network, the typical cognition tests showed the following gains in performance.

- ~ 4.2% recognition rate improvement in shape variation test
- ~ 9.6% average recognition rate improvement in noise tolerance tests

Shape variation and noise discrimination are considered among the few tests that are known to require higher cognitive functionalities for which the brain is the typical reference. These results, in addition to the suitability of the proposed model for more robust implementation on parallel architecture machines, stand in favor of the new model.

Thus:

- Based on the study and analysis of the operational characteristics in the neural network based model, results typical to any theoretical model based on the HTM principles were found. This illustrates that there is indeed a solid possibility that such a hierarchical organization can be applied to neural networks and hence a truly brain-like processing power can be obtained provided the necessary platform and resources.
- Based on the output comparisons of the original and the

new model under several conditions, it was found out that neural network based systems can attain improved levels of recognition accuracy in conditions that represent real world variances in input.

References

[1] Mostefa Golea, Mario Marchan (1990),"A Growth Algorithm for Neural Network Decision Trees", [Online] Available: http://www.iop.org/EJ/abstract/0295-5075/12/3/003.

[2] Eleine Rich, Kevin Knight,"Artificial Intelligence", 2nd Ed., pp. 3-27. Tata McGraw-Hill, 1991.

[3] Hugo De Garis, Sung-Bae Cho, Michael Korkin, Arvin Agah (1998),"Designing An Artificial Brain With 10,000 Evolved Neural Net Modules", [Online] Available: http://www.citeseer.ist.psu.edu/49849.html. [Last Accessed June 2007]

[4] J. Djordjevic, A. Milenkovic, S.Prodanovic (1999),"A Hierarchical Memory System Environment", [Online] Available: http://www.ncsu.edu/wcae/ISCA1998/ milenkovic.pdf.

[5] Risto Miikkulainen, James A. Bednar, Yoonsuck Choe, Joseph Sirosh (2000),"A Self-Organizing Neural Network Model Of The Primary Visual Cortex", [Online] Available: http://www.citeseer.ist.psu.edu/137449.html

[6] Asim Roy (2000),"Artificial Neural Networks - A Science in Trouble, Arizona State University, School of Information Management", [Online] Available: http://www.portal.acm.org/ft_gateway.cfm?id=846192&type=pdf&dl=portal&dl=ACM.

[7] Dean V. Buonomano, Michael Merzenich (2001),"A Neural Network Model of Temporal Code Generation and Position-Invariant Pattern Recognition", [Online] Available: http://www.neco.mitpress.org/cgi/reprint/11/1/103.pdf.

[8] David Voge (2002),"A Neural network Model Of Memory And Higher Cognitive Functions In The Cerebrum", [Online] Available: http://www.bbsonline.org/documents/a/00/00/11/77/bbs00001177-01/Vogel_HigerFnct.pdf.

[9] Gwendid T. Van der Voort, Van der Kleij, Marc de Kamps, Frank van der Velde (2003),"A Neural Model of Binding and Capacity in Visual Working Memory", [Online] Available: http://www.springerlink.com/index/kc9kxcru7q3101ym.pdf.

[10] Włodzisław Duch (2003),"Brain-Inspired Conscious Computing Architecture", School of Computer Engineering, Nanyang University of Technology", [Online] Available: http://cogprints.org/3319/1/03-Brainins.pdf

[11] LIN Jie, JIN Xiao-gang, YANG Jian-gang (2004),"A hybrid neural network model for consciousness", [Online] Available: http://www.zju.edu.cn/jzus/2004/0411/041119.pdf.

[12] Gert Westermann, Denis Mareschal (2004),"Connectionist modeling", [Online] Available: http://www.cnbc.cmu.edu/~plaut/papers/pdf/Plaut00chap.conn.pdf.

[13] Dileep George, Jeff Hawkins (2004),"Invariant Pattern Recognition using Bayesian Inference on Hierarchical Sequences", [Online] Available: http://www.stanford.edu/~dil/RNI/DilJeffTechReport.pdf.

[14] Bin Zhang, Sargur N. Srihari (2005),"Properties of Binary Vector Dissimilarity Measures", [Online] Available: http://www.cedar.buffalo.edu/papers/articles/CVPRIP03_propbina.pdf. [Last Accessed June 2007]

[15] Dileep George, Jeff Hawkins (2005),"A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex", [Online] Available: http://www.stanford.edu/~dil/invariance/Download/GeorgeHawkinsIJCNN05.pdf.

[16] Xiao-Jing Wang (2005),"A Microcircuit Model of Prefrontal Functions: Ying and Yang of Reverberatory Neurodynamics in Cognition", [Online] Available: http://assets.cambridge.org/97805216/72252/frontmatter/9780521672252_frontmatter.pdf.

[17] M. W. Spratling, M. H. Johnson (2006),"A Feedback Model of Perceptual Learning and Categorization", [Online] Available: http://www.cogprints.org/4885/1/vis_cog06.pdf.

[18] Jeff Hawkins, Dileep George (2006),"Hierarchical Temporal Memory - Concepts, Theory, and Terminology", [Online] Available: http://www.numenta.com/Numenta_HTM_Concepts.pdf.

[19] Numenta Inc. (2006),"Hierarchical Temporal Memory, Comparison with Existing Models", [Online] Available: http://www.numenta.com/for-developers/education/HTM_Comparison.pdf.

[20] Dileep George, Bobby Jaros (2006),"The HTM Learning Algorithms", [Online] Available: http://www.numenta.com/for-developers/education/Numenta_HTM_Learning_Algos.pdf.

[21] Numenta Inc.(2007),"Numenta Platform for Intelligent Computing Programmer's Guide Version 1.0.3", [Online] Available: http://www.numenta.com/for-developers/education.php.

[22] Numenta Inc. (2007),"Zeta Algorithms Reference, Version 1.2", [Online] Available: http://www.numenta.com/for-developers/software/pdf/nupic_gettingstarted.pdf.

[23] Wikipedia [Free online Encyclopedia]. Connectionism. [Online] Available: http://en.wikipedia.org/wiki/Connectionism

[24] Wikipedia [Free online Encyclopedia]. Neural Networks, [Online] Available: http://en.wikipedia.org/wiki/Neural_networks.

[25] Wikipedia [Free online Encyclopedia]. Cognition. [Online] Available: http://en.wikipedia.org/wiki/Cognition

[26] Wikipedia [Free online Encyclopedia]. Blue Gene. [Online] Available: http://en.wikipedia.org/wiki/Blue_gene

[27] Wikipedia [Free online Encyclopedia]. Supercomputers [Online] Available: http://en.wikipedia.org/wiki/Supercomputers

[28] Wikipedia [Free online Encyclopedia]. Artificial Intelligence. [Online] Available: http://en.wikipedia.org/wiki/Artificial_intelligence

Rajesh Babu received his Post Graduate degree in Physics from Avadh University, Faizabad (U.P.), India in 1984, Master of Computer Application degree from Technical University, Punjab, India in 2005 and M.Tech. (IT) degree from Karnataka State University, Mysore, (Karnataka) India in 2010. He was Asst. Professor and Associate Professor with Deptt. of Computer Science of Bareilly College, Bareilly affiliated with MJP Rohilkhand University, Bareilly (U.P.) India during 1989 to 2006. At present he is working as Professor in Computer Application Department with Rakshpal Bahadur Management Institute, Bareilly (U.P.) India, which is approved by All India Council for Technical Education and affiliated with Gautam Buddh Technical University, Lucknow (U.P.) India. His research interest include Implementation of Neural Network based Hierarchical Temporal Memory to realize Cognitive Functions.