Multi-Layer Cortical Learning Algorithms

Pulin Agrawal
Department of Computer Science
The University of Memphis
Memphis, USA
pagrawal@memphis.edu

Stan Franklin
Institute for Intelligent Systems
The University of Memphis
Memphis, USA
franklin@memphis.edu

***Abstract*— Hierarchical Temporal Memory (HTM) is a model with hierarchically connected modules doing spatial and temporal pattern recognition, as described by Jeff Hawkins in his book entitled *On Intelligence*. Cortical Learning Algorithms (CLAs) comprise the second implementation of HTM. CLAs are an attempt by Numenta Inc. to create a computational model of perceptual analysis and learning inspired by the neocortex in the brain. In its current state only an implementation of one isolated region has been completed. The goal of this paper is to test if adding a second higher level region implementing CLAs to a system with just one region of CLAs, helps in improving the prediction accuracy of the system. The LIDA model (Learning Intelligent Distribution Agent - LIDA is a cognitive architecture) can use such a hierarchical implementation of CLAs for its Perceptual Associative Memory.**

Keywords— Cortical Learning Algorithms, predictive coding, LIDA, feedback message, pattern recognition

I. Introduction

Cortical Learning Algorithms (CLAs) constitute an attempt by Numenta Inc. to create a computational model of perceptual analysis and learning inspired by the neocortex in brains [1]. CLAs are used in the second implementation of a general framework for perceptual learning called Hierarchical Temporal Memory (HTM) [1-5]. The CLAs are a set of algorithms operating on a data structure. The data structure and algorithms, together, achieve some degree of spatial and temporal pattern recognition. The data structure used is a collection of columns of cells, called a region. A cell in a column is a neuron like entity, which makes connections to other cells, and aggregates their activity to determine its state of activation. Fig. 1 illustrates the structure of a region.

The Hierarchical Predictive Coding Model (HPCM) [6] is a technique for message passing between levels of a hierarchy doing perceptual analysis, where predictions about the next incoming input are sent down and prediction errors are transmitted up in the hierarchy. In this technique, each level of the hierarchy has a generative model. The generative model in each of these levels builds a model of its input domain in the terms of hypothetical causes of its inputs, and tries to predict the next input. These levels are stacked to form a Hierarchical Generative Model [6].

The CLAs, as described in the white paper [1], do the same thing with its input domain that a generative models does in the Hierarchical Predictive Coding Model described above. It tries to build a model of its input domain based on the statistics of the input and tries to predict it.

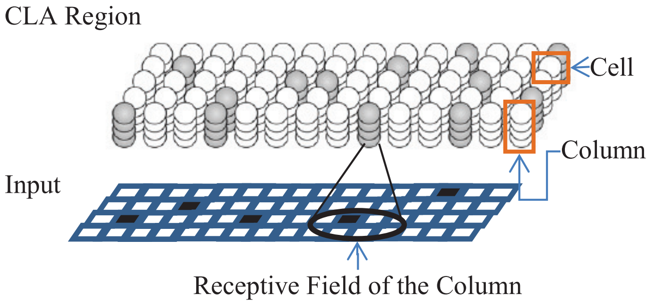

Fig 1. A CLA region is a column of cells. A column has a receptive field over the input. It aggregates the input activity in its receptive field and competes with the neighboring columns to decide whether to become active.

The idea of using predictive coding in HPCM was inspired by the work of Karl Friston et al. [7-9] in formalizing the brain as a system trying to minimize a quantity called 'free-energy'. 'Free-energy' of a system is a quantity dependent on the error the system has in predicting its environment. The CLAs try to predict their environment, and also do error-minimization. Thus, they can be broadly classified as a flavor of implementations based on the Free-Energy Principle [8] like the HPCM.

When the inferences made by a system are conditioned upon the statistics of the input it receives, then the system may be performing Bayesian inference. There has been growing evidence that brains have systems that can be thought of as doing Bayesian inference [16-20]. It is important for a system like CLAs to be able to perform Bayesian Inference because it gives this system the ability to operate in situations of uncertainty. Bayesian Inference improves the performance in the situations of uncertainty by providing a top-down influence that is based on the knowledge of a relatively abstract information gathering process. A one region system, as described in the CLAs white paper, does not have the tools to do Bayesian Inference. A two region CLAs could provide a possible implementation of a Bayesian inference system modeled after those in the brain. The higher region in a two region CLA will look at the activity of the cells in the lower region (input to higher region) and learn the temporal statistics of its activity pattern. The higher region then provides predictive inferences to the lower region about cell activity of the lower region. Lower region can use these inferences to improve its predictions of input activity. This type of message passing mechanism (like Hierarchical Predictive Coding Model [6]) is a possible implementation of Bayesian inference in brains.

978-1-4799-5375-2/14/$31.00 ©2014 IEEE

The white paper on CLAs proposes an implementation of only one CLA region. In this paper, we explore a two region hierarchy of CLA regions. We hypothesize that a two region hierarchy will improve the prediction accuracy of the system in comparison to only one region as described in the CLA white paper [1]. It was the thrust of our work to first come up with a mechanism of message passing between the proposed two regions of the desired two region system. One major constraint, making this task non-trivial, was to make this message passing mechanism general enough to be scalable to a multi-level hierarchy with multiple regions feeding into one region.

The fundamental data structure described in the CLA white paper is based on the structure of the cortical region. There is a wide consensus among neuroscientists about the hierarchical organization of the brain [10] and cortical regions in it [11-14]. The benefit of a hierarchy—as described in On Intelligence [15]—is that the higher levels of the hierarchy can extract more abstract knowledge about the environment. This is because the lower layers do some spatio-temporal grouping of the input that reduces the workload at the higher levels. The higher levels can then use these patterns from lower layers to do their own spatio-temporal grouping. Thus, more abstract patterns can be learned to make inferences or predictions about the lower levels and eventually the input at the lowest level. Such predictions would be otherwise impossible to achieve at a level of abstraction of knowledge that the lower levels can work on. These ideas are in support of our hypothesis.

II. Two Region Cortical Learning Algorithms

In this paper, we explore CLA systems with two regions. Such a system is a three level hierarchy, consisting of an input level and two CLA regions, each region being a level in the hierarchy. We build the interconnections between the levels by taking hints from the literature [1, 6] as described above. They include a feedback message-passing method of the predictive coding technique to implement the top-down connections between the levels of this three level hierarchy.

A. Model of the System

The system consists of a three level arrangement. The first level is an input level. The two levels above the input level are structured after the description of a region in the Cortical Learning Algorithms' paper [1].

The input to the system at one time instant is a Boolean vector from the input level, called the input vector, as shown in Figure 2. This input vector is generated by reading the state of sensors of the immediate environment. The state of the entities of the system, viz. columns and cells, constitute the internal representation of the stimuli. The output is a Boolean vector. It is the prediction of the next input stimulus made by the system, given the current input. This output is stored in a buffer called prediction vector. Fig. 2 depicts the stacking of the cortical regions, and a comparison of the system in the CLA paper and the system described in this section.

1) The input level.

The input level has a Boolean vector, called the input vector. The sensors get activated from the environment, and populate the input vector. The input level also contains a Boolean vector of the same dimension as the input vector called the prediction vector, to store the predictions about the input vector. The prediction vector is set by the next higher level. The process, by which this prediction vector is set, is described in the next sub-section, 'The higher levels'. For examples of input that this system can receive, see Figure 4.

2) The higher levels.

The higher levels consist of CLA regions. We have kept most of the workings of the original CLAs region intact, except for some essential modifications so that it can incorporate predictions from a higher level. The level immediately below a higher level acts as a source of input for the higher level. The vector containing the state of all the cells of a lower level acts as the input vector for the higher region. Both the lower region and the higher region work as described in the next paragraph.

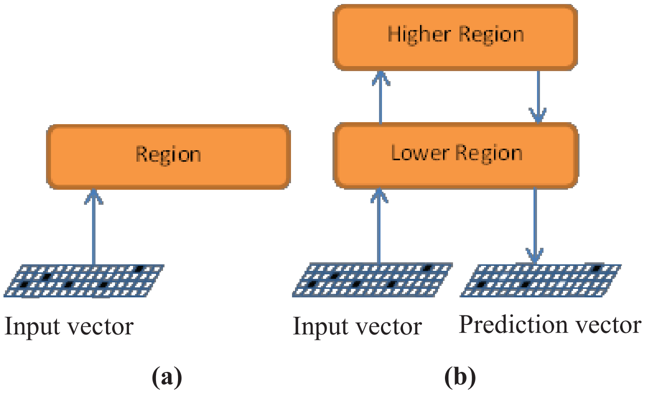

Fig 2. (a) shows the architecture of the system in the CLA paper. The region looks at the input vector to form spatio-temporal patterns. (b) shows the architecture of the system in this paper. Each level looks at the activity of the level below it to form spatio-temporal patterns, and sends a prediction about the lower level activity as feedback message.

A CLA region processes in two phases – a spatial pooling phase and a temporal pooling phase. The spatial pooling phase is for recognizing simultaneous patterns. These are the patterns in the input vector, which occur simultaneously (at one time instant). The spatial pooling phase uses a Boolean vector for building an internal sparse distributed representation of the input. This is done by looking at the input activity, and turning approximately 2% of the bits to an 'on' state in this Boolean vector, based on the spatial pooling algorithm described in the CLA paper [1]. Fig. 3 illustrates a couple of input representations after spatial pooling. The temporal pooling phase then uses such internal representations to recognize patterns of input activity over time, using the temporal pooler algorithm described in the CLA paper [1]. We added to the original CLA where the knowledge of the temporal pooler is used to generate a prediction about the next input stimulus.

The higher level region generates a prediction about the lower level region activity. This prediction is used to set the state of the cells in the lower level region. The lower level region uses this prediction to improve its knowledge. The lower level region then generates its prediction of its next input stimulus. This prediction is stored in a buffer called the prediction vector of the whole system, as described in the beginning of this section.

The prediction vector, thus generated, and the input vector are used to determine the prediction accuracy of the system. Prediction accuracy measures how well the prediction vector matches the input vector in the next input. It is calculated as the normalized Hamming Distance (cardinality of XOR) of the prediction vector and the input vector at the next time step. This distance is normalized by the number of pixels in the input vector (or the prediction vector) to obtain a percentage, called the ErrorIndex.

B. Hypothesis

We hypothesize that, after training, a system with two regions, called System 2 (S2)—as described in the section above—is more accurate at making predictions about the next input stimulus as compared to a system with just one region, called System 1 (S1). The ErrorIndex will serve as the random variable under consideration to formalize our hypothesis. Therefore, we state the null and the alternate hypotheses for the population means of ErrorIndex over all possible six image sequences of 16x16 1-Boolean pixel images, obtained after training, for S1 (m_{S1}) and for S2 (m_{S2}), as follows:

- H0 (null hypothesis): $m_{S1} \leq m_{S2}$
- H1 (alternate hypothesis): $m_{S1} > m_{S2}$

To test this hypothesis we sample the common input space of S1 and S2. Based on the law of large numbers, we will take 100 independent measurements of the random variable ErrorIndex to get 100 data values from each sample. Since our sample size is large enough, the Central Limit Theorem will be applicable. Thus, we will be able to perform an independent samples one-tailed Student's t-test to compare the means of these samples, based on our research hypothesis. We will use α=0.05 significance level for the t-test.

C. Experiment

The experimental setup that was used to test our hypothesis is described in this section. It involves a comparison of the two systems, S1 and S2. System 1 (S1) has an input level and a CLA region. System 2 (S2) has one input level and two CLA regions, as described in the aforementioned sub-section 'Model of the System'. We will be using 16x16 1-Boolean pixel images as input vectors for the experiment. Fig. 4 shows a couple of example images that were fed to the systems, S1 and S2, during the experiment.

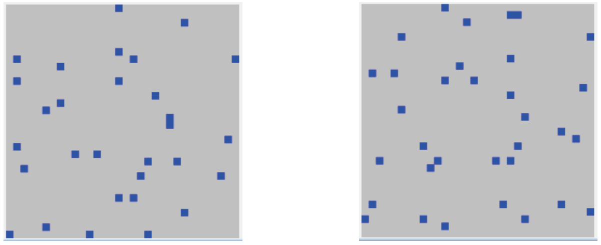

Fig 3. These images are examples of input representations (size 32x32) of a pattern after spatial pooling by the first region. Each blue square is an active column and the inactive columns are in gray.

Before running the experiments, we determined the number of cycles required for training the system as follows. For that, we trained the system on several 16x16 1-Boolean pixel image sequences. Each sequence had six images in it. We recorded the number of cycles it took for ErrorIndex to settle down within a threshold range when a sequence was presented in a loop. The mean recorded number of cycles was 1406. The standard deviation of the recorded number of cycles was 92. So we decided the training period should be of 1500 cycles (mean plus the standard deviation of recorded number of cycles). We rounded off the upper bound on the mean of the number of cycles to the higher value, to ensure completion of training.

The procedure for doing the experiment will be described now. We generate the control dataset and the test dataset as follows:

Control Dataset (D1): We create a sequence of 1500 randomly generated 16x16 1-Boolean pixel images as input to the system for training the systems once. The number of images in a sequence is the same as the number of cycles required for training, to minimize any temporal information that might occur due to repetition. We trained the systems independently a 100 times; therefore we created a set of 100 such sequences to form D1.

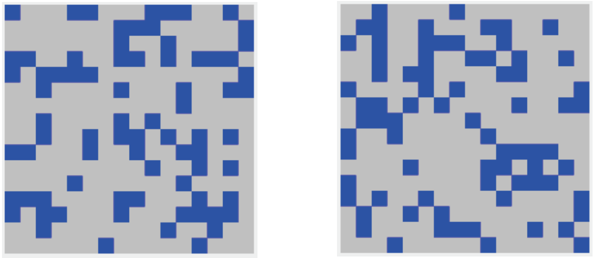

Fig 4. These are examples of images that were shown to the systems. Each image is a 16x16 1-Boolean pixel image.

Test Dataset (D2): We created a sequence of six randomly generated 16x16 1-Boolean pixel images looping through 1500 cycles. We trained the systems independently a 100 times; therefore we created a set of 100 such sequences to form D2. The repetition of a sequence causes an item in this set to deliver the required temporal information for the systems to learn.

To perform the control experiment we choose one sequence from D1. We input this sequence of images, one image at a time, to both the systems, until the desired number of training cycles complete (1500 cycles). We stop the training and we input an image from that sequence. Then we use the prediction vector, thus generated, and the next input in the sequence to take a measurement of the ErrorIndex from both the systems. We record this as one data point of the sample of ErrorIndexes for each of the systems. We iterate over this process a 100 times, choosing a different sequence from D1 and reinitializing both the systems each time. After this process we get 100 independently obtained data points for each sample of ErrorIndex, from S1 and S2, called ES1 and ES2 respectively.

We follow the same procedure to obtain the samples ES1 and ES2 for test dataset D2.

III. Analysis and Results

Comparing the means of the two samples of ErrorIndex obtained from S1 and S2 will tell us if there is a significant difference between the two samples. An independent samples one-tailed t-test is a suitable candidate to test the difference in means of the two samples thus obtained. The Shapiro-Wilk Normality test can be used as the applicability test to check if the sample data meet the assumptions of t-test, i.e. the samples are normally distributed. The samples passed the Shapiro-Wilk Normality test. Then we conducted the t-test to compare the samples ES1 and ES2 from the control dataset D1 and the test dataset D2. Results for both of these tests are mentioned in the following sub sections. The usual meaning of the symbols for a t-test and Shapiro-Wilk Normality test apply i.e. t is the test statistic for t-test, W is the test statistic for normality test, df refers to the degrees of freedom, p means the probability of obtaining the test statistic within the confidence interval, m means the population mean, $\bar{m}$ means the sample mean and σ means the sample standard deviation.

A. *Analysis of samples from D1*

Shapiro-Wilk Normality test gave a p-value of 0.7764 for W = 0.9913 for ES1 and p-value of 0.2068 for W = 0.9825 for ES2. P-value > 0.05 for both ES1 and ES2 allowed us to conduct t-test. An independent-samples t-test was conducted to compare ES1 and ES2 from D1. There was a significant difference in the scores for ES1 ($\bar{m}_{ES1}$= 0.4817407, σ_{ES1}= 0.01275459) and ES2 ($\bar{m}_{ES1}$= 0.46482, σ_{ES2}= 0.02149648); t(df= 161.32)= 6.7606, p = 1.194e-10. This shows that ErrorIndex from S1 was greater than from S2. The difference between the means of the two samples was $\bar{m}_{ES1}$- $\bar{m}_{ES2}$= 0.017. The two samples tested here are shown in the Fig. 5.

The proposed system S2 showed a reduction in the prediction error by just 0.017. This reduction in error can be attributed some random patterns present in the sequence of 1500 images which the system was able to exploit to improve its predictions.

B. *Analysis of samples from D2*

Shapiro-Wilk Normality test gave a p-value of 0.1657 for W = 0.9812 for ES1 and p-value of 0.1701 for W = 0.9814 for ES2. P-value > 0.05 for both ES1 and ES2 allowed us to conduct t-

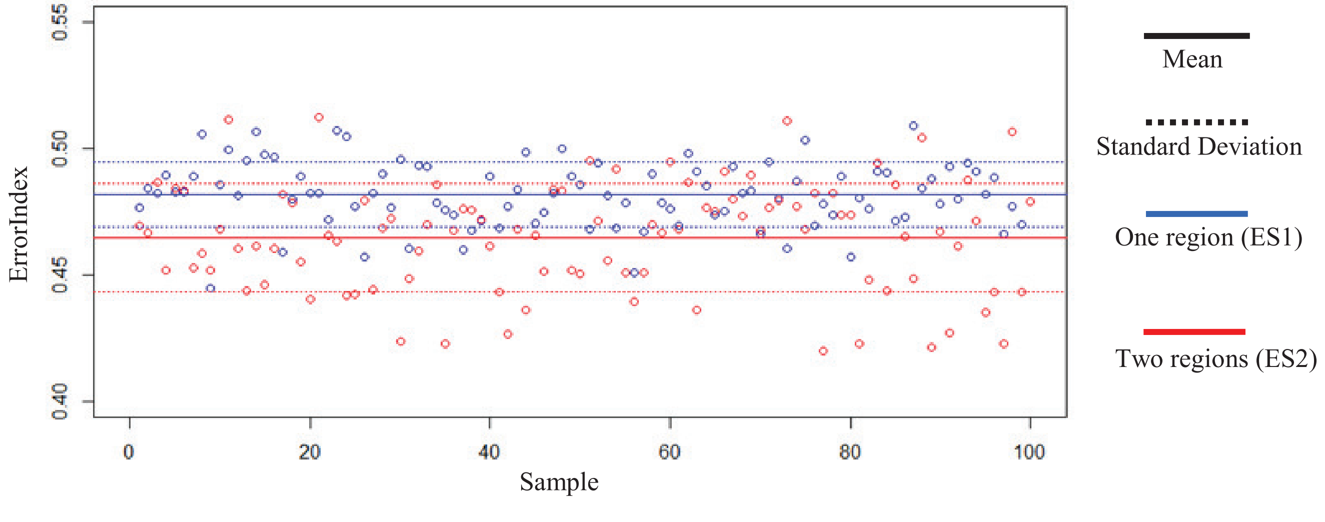

Fig 5. The prediction accuracy is not any better for two regions (red) as compared to one region (blue) since the mean lines (solid lines) are not significantly far apart and the standard deviation lines (dotted lines) for the two systems cross each other.

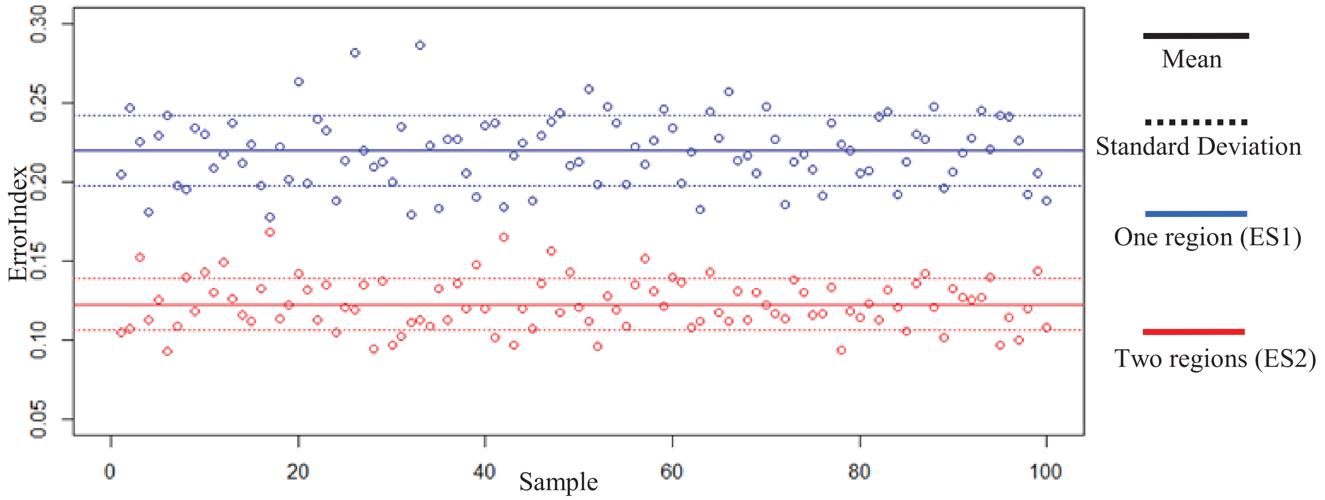

Fig 6. The prediction accuracy is better for two regions (red) as compared to one region (blue) since the mean lines (solid lines) are significantly far apart and the standard deviation lines (dotted lines) for the two systems do not cross. A one-sided t-test confirmed that the mean of the sample ES2 of the two region model S2 is indeed significantly lower.

TABLE I. RESULTS OF T-TEST

Dataset	Sample	Mean ($\bar{m}$)	S. Dev. (σ)	t	df	p-value
D1	ES1	0.48174	0.01275	6.7606	161.32	1.194e-10
	ES2	0.46482	0.02149			
D2	ES1	0.2194	0.02209	35.5581	180.3	< 2.2e-16
	ES2	0.1225	0.01597			

test. An independent-samples t-test was conducted to compare ES1 and ES2 from D2. There was a significant difference in the scores for ES1 ($\bar{m}_{ES1}$= 0.219485, σ_{ES1}= 0.02209806) and ES2 ($\bar{m}_{ES2}$= 0.1225183, σ_{ES2}= 0.01597893); t(df=180.3)= 35.5581, $p < 2.2e\text{-}16$. This shows that ErrorIndex from S1 was significantly greater than from S2. The difference between the means of the two samples was $\bar{m}_{ES1}$- $\bar{m}_{ES2}$= 0.111778348. The two samples tested here are shown in the Fig. 6.

The proposed system S2 showed a reduction in the prediction error by 0.1117. The reduction for the test dataset was much more than the control dataset. This shows that the characteristics of the test dataset, i.e temporal pattern,was exploited to improve the prediction.

The results of the t-test for both the datasets are shown in Table 1. The data that we obtain provide strong evidence in favor of our alternate hypothesis, and against the null hypothesis. The p-value from the t-test is less 0.05, therefore we cannot reject our research hypothesis that the mean ErrorIndex obtained from System 1 is higher than the mean ErrorIndex obtained from System 2. Since we are measuring the error in prediction, we say that System 1 has higher error in prediction than System 2. Thus, we can say that System 2 predicts better than System 1. The addition of the second layer does improve the prediction accuracy of the system as measured by the ErrorIndex.

IV. CONCLUSION

The tests described in this paper suggest that a two region hierarchy of CLAs can be built by adding feedback connections based on the predictive coding scheme. Based on the t-test performed, we concluded that the prediction accuracy of the proposed system with two regions is better than just one region as described in the CLAs paper [1].

Since the addition of a second region in the hierarchy is able to improve the performance, we can expect that by adding more regions in a hierarchy, the hierarchy will be able to extract more abstract patterns. This will enable a more sophisticated Bayesian Inference, hopefully improving the performance of this system in uncertain situations that might arise in the process of making predictions. This has also given us hope of being able to fuse representations from two different lower level regions into one higher level region. Since the two lower level regions can take input from two different sensory systems/modalities, we may be able to generate rich representations of the environment. This might help in improving the perceptual system of the LIDA architecture [21-23]. Such a kind of hierarchical representation may allow us to make Perceptual Associative Memory (PAM) [24, 25] more efficient because the hierarchical temporal patterns might allow more extensive event representations [25] to be learned. This can also help in carving out the design for a sparse coding based vector representation for the LIDA system.

This topic still needs further exploration in many directions. It will be interesting to see the improvements that a higher region can deliver in recognition of spatial patterns. Also, we still need to determine the limit of additional levels that will continue to improve the performance of the system or if there exists any such limit at all. This system is not currently equipped to handle real-world data, but running this system on a real-world dataset will provide useful insights about the performance of this system in real-world applications. In fact, creating a multi-level hierarchy is the first step towards making this system better equipped for real-world data.

Acknowledgment

We want to acknowledge the huge amount of time and effort put in by the Cognitive Computing Research Group at The University of Memphis in conceptualizing and designing the LIDA Framework. We would especially like to thank Ryan James McCall for building the Cortical Learning Algorithms framework upon the LIDA Framework that enabled all the experiments mentioned in this paper. A lot of useful feedback on statistics from Lih Y. Deng and Stephen Strain helped in improving the quality and content of this paper. Thank you!

References

[1] J. Hawkins, S. Ahmad and D. Dubinsky. Hierarchical temporal memory including HTM cortical learning algorithms, v. 0.2. *Unpublished* 2011. Available: www.numenta.com.

[2] D. George, *How the Brain might Work: A Hierarchical and Temporal Model for Learning and Recognition,* 2008.

[3] D. George and J. Hawkins, "Towards a Mathematical Theory of Cortical Micro-circuits," *PLoS Comput Biol,* vol. 5, pp. e1000532, 10/09, 2009.

[4] J. Hawkins and S. Blakeslee, "A new framework of intelligence," in ***On Intelligence***, 1st ed.Anonymous United States: Times Books, 2004, pp. 58.

[5] J. Hawkins, D. George and J. Niemasik, "Sequence memory for prediction, inference and behaviour," *Philos. Trans. R. Soc. Lond. B. Biol. Sci.,* vol. 364, pp. 1203-1209, May 12, 2009.

[6] R. P. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nat. Neurosci.,* vol. 2, pp. 79-87, Jan, 1999.

[7] K. Friston, N. Trujillo-Barreto and J. Daunizeau, "DEM: a variational treatment of dynamic systems," *Neuroimage,* vol. 41, pp. 849-885, 2008.

[8] K. Friston, "The free-energy principle: a unified brain theory?" *Nat. Rev. Neurosci.,* vol. 11, pp. 127-138, Feb, 2010.

[9] K. J. Friston, J. Daunizeau, J. Kilner and S. J. Kiebel, "Action and behavior: a free-energy formulation," *Biol. Cybern.,* vol. 102, pp. 227-260, Mar, 2010.

[10] C. Zhou, L. Zemanová, G. Zamora, C. C. Hilgetag and J. Kurths, "Hierarchical organization unveiled by functional connectivity in complex brain networks," *Phys. Rev. Lett.,* vol. 97, pp. 238103, 2006.

[11] D. J. Felleman and D. C. Van Essen, "Distributed hierarchical processing in the primate cerebral cortex," *Cerebral Cortex,* vol. 1, pp. 1-47, 1991.

[12] J. L. McClelland, B. L. McNaughton and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory." *Psychol. Rev.,* vol. 102, pp. 419, 1995.

[13] V. B. Mountcastle, *Perceptual Neuroscience: The Cerebral Cortex.* Harvard University Press, 1998.

[14] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nat. Neurosci.,* vol. 2, pp. 1019-1025, 1999.

[15] J. Hawkins and S. Blakeslee, *On Intelligence.* United States: Times Books, 2004.

[16] E. Jaynes, *How does the Brain do Plausible Reasoning?* Springer, 1988.

[17] G. Hinton, P. Dayan, A. To and R. Neal, "The helmholtz machine through time," in *International Conference on Artificial Neural Networks (ICANN-95),* 1995, pp. 483-490.

[18] P. Dayan, G. E. Hinton, R. M. Neal and R. S. Zemel, "The helmholtz machine," *Neural Comput.,* vol. 7, pp. 889-904, 1995.

[19] P. Dayan and G. E. Hinton, "Varieties of Helmholtz machine," *Neural Networks,* vol. 9, pp. 1385-1403, 1996.

[20] T. Madl, S. Franklin, K. Chen, D. Montaldi and R. Trappl, "Bayesian Integration of Information in Hippocampal Place Cells," *PloS One,* vol. 9, pp. e89762, 2014.

[21] S. Franklin and F. Patterson Jr, "The LIDA architecture: Adding new modes of learning to an intelligent, autonomous, software agent," *Pat,* vol. 703, pp. 764-1004, 2006.

[22] S. Franklin, T. Madl, S. D'Mello and J. Snaider, "LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning," *IEEE Transactions on Autonomous Mental Development,* 2013.

[23] U. Ramamurthy, B. J. Baars, S. Franklin and S. K. D'Mello, "LIDA: A Working Model of Cognition," in *Proceedings of the 7th International Conference on Cognitive Modeling,* Edizioni Goliardiche, Trieste, Italy; Eds: Danilo Fum, Fabio Del Missier and Andrea Stocco; pp. 244-249, 2006.

[24] S. Franklin, "Perceptual memory and learning: Recognizing, categorizing, and relating," in *Proc. Developmental Robotics AAAI Spring Symp,* 2005, .

[25] R. J. McCall, S. Franklin, D. Friedlander and S. D'Mello, "Grounded event-based and modal representations for objects, relations, beliefs, etc.." in *FLAIRS Conference,* 2010, .