Cortical Columns: Building Blocks for Intelligent Systems

Atif G. Hashmi and Mikko H. Lipasti
Department of Electrical and Computer Engineering
University of Wisconsin - Madison
Email: {ahashmi@wisc.edu, mikko@engr.wisc.edu}

***Abstract*— The neocortex appears to be a very efficient, uniformly structured, and hierarchical computational system [25], [23], [24]. Researchers have made significant efforts to model intelligent systems that mimic these neocortical properties to perform a broad variety of pattern recognition and learning tasks. Unfortunately, many of these systems have drifted away from their cortical origins and incorporate or rely on attributes and algorithms that are not biologically plausible. In contrast, this paper describes a model for an intelligent system that is motivated by the properties of cortical columns, which can be viewed as the basic functional unit of the neocortex [35], [16]. Our model extends *predictability minimization* [30] to mimic the behavior of cortical columns and incorporates neocortical properties such as hierarchy, structural uniformity, and plasticity, and enables adaptive, hierarchical independent feature detection. Initial results for an unsupervised learning task–identifying independent features in image data–are quite promising, both in a single-level and a hierarchical organization modeled after the visual cortex. The model is also able to forget learned patterns that no longer appear in the dataset, demonstrating its adaptivity, resilience, and stability under changing input conditions.**

I. Introduction

CURRENT generation computing systems, even though computationally quite powerful, suffer from three major issues. First, by any meaningful definition of the term, they are not intelligent. Second, the exponential increases in performance that we have enjoyed for decades may not last much longer because of power dissipation overhead. Third, over time, the basic structural unit of a processing system, i.e. a transistor, is becoming more and more faulty due to technology scaling[2].

One of the main reasons for the lack of intelligence in contemporary processing systems is that all these systems are based on the Von Neumann model which by definition lacks any intelligence. This suggests that processing paradigms other than the Von Neumann model must be explored in order to develop truly intelligent systems. We also have to live with the fact that in the future transistors will become more and more unreliable. Thus, we need to develop processing models that are inherently fault-tolerant. These observations lead us to study a biological computing system: the *human neocortex*. It is the most powerful and flexible natural computing system and is highly fault tolerant.

The neocortex is unique to mammals and is highly developed for humans; it accounts for about 76% of the mass of the human brain [33]. It is one of the most powerful, fascinating, and complex natural processing system that is yet to be completely understood. A neuron, the basic structural unit of the neocortex, is orders of magnitude slower than a transistor, the basic structural unit of modern computing systems. The average firing interval for a neuron is around 150ms to 200ms [21] while transistors can operate in less than a nanosecond. Still, the neocortex performs much better on pattern recognition and other learning based tasks than contemporary high speed computers. One of the main reasons for this seemingly unusual behavior is that certain properties of the neocortex–like independent feature detection, attention, feedback, prediction, and training data independence–make it a quite flexible and powerful parallel processing system.

Another interesting property of the neocortex is that it accomplishes its tasks using seemingly unpredictable and unreliable basic components. Studies by researchers in neuroscience show that neurons appear highly unpredictable. A single neuron might or might not fire under identical circumstances [4]. Even in the presence of such an unpredictable basic structural unit, the neocortex accomplishes its tasks very efficiently.

Many efforts have been made to construct systems that mimic the working of neocortex. Some examples include artificial neural networks [15], Bayesian networks [10], Hebbian learning models [19], Dean's model [6], and hierarchical temporal memories [14], [9]. Most models incorporate some of the properties of neocortex, but still do not appear to be as powerful, efficient, and flexible as the neocortex because of a number of reasons. First, these models do not accurately mirror the anatomical structure of the brain. Second, they do not model the basic and most important properties of the brain like plasticity, prediction, and feedback. Third, these models can be quite dependent on the type of data that is used to train them. Fourth, most artificial intelligence models consider neurons to be the basic functional unit of the neocortex. Even though neurons are the basic structural unit of the neocortex, they are not necessarily the basic functional unit [35], [16]. [14],[9] and [6] compare their model with the properties of the neocortex but they do not exactly implement the underlying properties of the cortical columns. Specifically, they lack the lateral connections, feedback, and predictions that play a very important role in learning and development of the neocortex.

In this paper, we assert that cortical columns should be considered the basic functional unit to construct a flexible and biologically plausible intelligent system. We develop

a functional model that uses the concept of *predictability minimization*[30] to construct a basic unit with behavior and properties quite similar to a cortical column. We then establish the biological basis of our basic functional unit by comparing it with the structural properties of mini-columns and hyper-columns. We simulate a network of 8 modules, each of which has a receptive field in the form of a 2x2 grid and is exposed to events like vertical, horizontal, and diagonal lines. Over time, each of the units trains itself to identify an independent event, supplying dimensionality reduction. Any remaining units that provide only redundant information (once all independent dimensions have been identified), keep thrashing as they search for independent feature that may appear in future. We then model a hierarchical organization of these networks–modeled after the visual cortex–which reliably detects higher-order shapes in its receptive field. Our model also shows the ability to forget previously learned patterns that stop occurring in the dataset the network is being exposed to. The main advantage of *learning to forget* is that it frees up resources that can be reassigned. Otherwise, the network may grow indefinitely and eventually fail.

The main contributions of our work are as follows:

- We propose cortical columns as the basic functional unit of an intelligent system.
- We present a biologically plausible computational model that extends the idea of predictability minimization to mimic the functional attributes of a cortical column.
- We establish the biological basis of our model by comparing it with the structural properties of a cortical columns.
- We present a biologically plausible hierarchical model for an intelligent system that recognizes complex shapes and also shows the ability to forget.

II. Neocortex: The Executive Processing System

The neocortex is the part of the brain that is responsible for perception, language, imagination, mathematics, arts, music, planning, and all the other aspects necessary for an intelligent system. It contains virtually all our memories, knowledge, skills, and experiences.

A very interesting property of the neocortex is its apparent structural uniformity [23], [24]. It is composed of millions of seemingly-identical functional units that are called *cortical columns*. Because of this property, the regions of the cortex that handle auditory inputs appear very similar to the regions that handle visual and other inputs. This uniformity suggests that even though different regions specialize in different tasks, they employ the same underlying algorithm.

The neocortex is also hierarchically structured[3], [16], [23]. This means that there is an abstract sense of *above* and *below* in the hierarchy. There are afferent paths from sensory inputs to the lower cortical regions (lower levels of the hierarchy) and from the lower cortical regions, information flows to the higher ones. Moreover, based on the information being received, the higher cortical regions generate activity which is communicated to the lower cortical regions via feedback paths [20]. For example, the visual cortex has 4 levels in its hierarchy. These levels are V1, V2, V4, and IT [26]. Visual information from retina enters the V1 and flows up the hierarchy while the feedback information from IT flow down the lower layers in the hierarchy [38], [14].

The main purpose of these feedback paths is to modulate the response of the lower cortical regions based on the predictions made by the higher ones [8]. These feedback paths that bring predictive information from the higher regions to the lower ones are one of the most important and powerful features of the neocortex. Research shows that the number of feedback paths taking predictions down the hierarchy is significantly greater than the number of feedforward paths going up the hierarchy [20]. This clearly suggests the importance of the feedback predictive paths.

Attention is another important neocortical feature. Paying attention to something may be achieved by means devoting more cortical units to it in order to extract maximum information out of it. If we see something unusual or something novel, we focus on it and spend time and effort understanding it. Once we understand that unusual or novel input, we no longer pay extra attention to it.

Plasticity is one of the most powerful features of the neocortex. The neocortex continuously makes new connections and keeps updating the synaptic weights based on the data being fed to it [26]. Plasticity contributes significantly to the efficient, fast, and online learning process of the brain.

Some contemporary intelligent systems partially model the hierarchical property of the neocortex [27], [32], [12], [31], [36], [7], [34], [37], [14], [9], but they *do not* model the structural and functional uniformity and the feedback connections that influence the output of the lower layers. Similarly, none of these systems model the plasticity aspect of the neocortex. In general, there is no notion of attention in any of these intelligent systems. We argue that these properties must be incorporated into a truly intelligent system and propose a framework that enables that.

III. Cortical Column: Basic Functional Unit of Neocortex

The neocortex is composed of thousands of identically structured functional units called cortical columns. Anatomically, a cortical column consists of six layers [3], [29], [16]. Feedforward information from lower cortical regions is received by Layer IV. Layer IV sends this information to Layers II and III which communicate it to higher cortical regions. Similarly, feedback information from the higher cortical regions is received by Layer I and is transfered to Layers II and III. Layers II and III then communicate that information to the Layers V and VI. Layers V and VI then transfer this information to lower cortical regions. Apart from these vertical paths, which convey information up and down the hierarchy, there are also horizontal paths between the cortical columns at the same level. Layers II and III of the cortical column connects the cortical columns at the same hierarchical level with each other. Mountcastle [24] suggests that a column is formed by many mini-columns

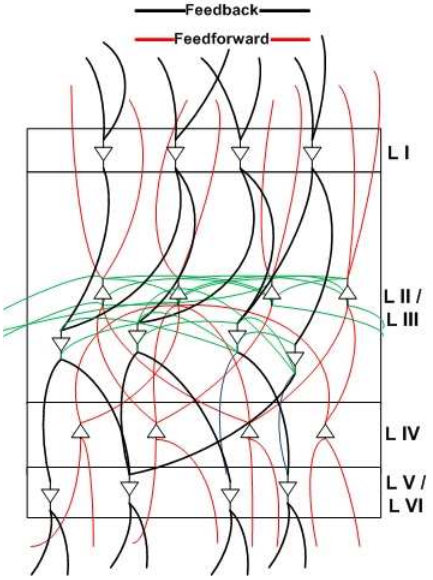

Fig. 1. Six layered structure of a cortical column.

bound together by short-range horizontal connections. Figure 1 shows the structural level diagram of a cortical column. In the figure, neurons are represented with triangles.

We hypothesize that these horizontal links are responsible for dimensionality reduction, which plays a critical role during the learning process. Cortical columns monitor the activity of nearby columns within the same network using these horizontal paths, and can modify the synaptic connections of their neurons to identify features from the data that are not being detected by other columns [11]. At the same time, the columns might also use these horizontal connection to determine if they are generating redundant information i.e. information that is being generated by some other column in the same network. By doing so, each of the columns in the same network learns to detect independent features from the input data. One of the classical examples of this behaviour is the primary visual cortex. Columns in the primary visual cortex train themselves to identify edges of different orientation from the image formed at the retina [17], [18], [22]. Each of the edges can be treated as an independent feature. A direct consequence of independent feature detection is that it reduces the dimensionality of the data. Once the columns train themselves to identify independent features from the input, it is very easy to identify columns that are providing redundant information. The outputs of columns that generate redundant information can be ignored, and the columns themselves can be pruned and reassigned to other tasks.

Another interesting property of cortical columns is that even though they are identical, they appear to be data-independent: no matter what type of data is used to train them, the columns can learn to extract independent features from that data. For example, even though there are distinct regions in the neocortex that learn to identify visual, auditory, and somatosensory data, these regions are all structurally very similar, if not identical.

Most artificial intelligence systems tend to model neurons as the basic functional unit [15], [10], [19]. Even though neurons are the basic structural unit of the neocortex, we feel they are not the most useful abstraction for a basic functional unit. There are example in the literature from the neuroscience community like [35] that support our hypothesis. A single neuron fires based on the action potentials it receives through the synapses in its dendritic tree. Long-term potentiation and depression (LTP and LTD) modify these synaptic weights so the neuron can fire in response to different inputs. However, the mechanism that governs LTP and LTD is not well understood. We argue that the horizontal connections in a cortical column constitute this mechanism, and propose a model based on predictability minimization that mimics LTP and LTD, and results in a computationally powerful behavior similar to what has been observed in cortical columns. Hence, a system that has basic functional blocks modeled on the properties of a cortical column can be much more flexible, powerful, and biologically plausible than a conventional neural network. Several studies suggest that cortical columns play a very important role in learning and recognition [35], [29], [13], [28]. Cortical columns are a practical and useful abstraction layer that separates the low level structural properties of the neurons (i.e. their activation, properties of synaptic connections, effects of neurotransmitters on their behavior, etc.) from their function or behavior. Developing models for the structural properties of neurons is critical to understanding their behavior, but to develop large-scale intelligent systems, modeling these low level structural properties may not be essential.

IV. A Computational Model for Cortical Columns

We believe that for any artificial system to be as powerful and flexible as the neocortex, its fundamental functional unit should be modeled to represent the functional properties of a cortical column. In an effort to develop a system that is motivated by the functional properties and interconnections of cortical columns, we build a model that extends the basic idea of predictability minimization (PM) proposed by Schmidhuber [30]. Originally, PM was proposed to automatically identify factorial codes but there are inherent properties of PM that make it a very good candidate for mimicking the functional properties of cortical columns.

A. Predictability Minimization

The main idea of predictability minimization suggests that there are multiple modules within a network. These modules are connected to each other via horizontal paths. Each of the modules has two main components: a predictor unit and a code unit. The code unit is like a conventional perceptron, adjusting its weights to fire only for particular inputs. The predictor unit for each module gets its inputs from the outputs of all the code units of all the other modules in the network. Based on these inputs, the predictor tries to predict the output of its own code unit. At the same time, the code unit tries to avoid the prediction made by its predictor unit. Each

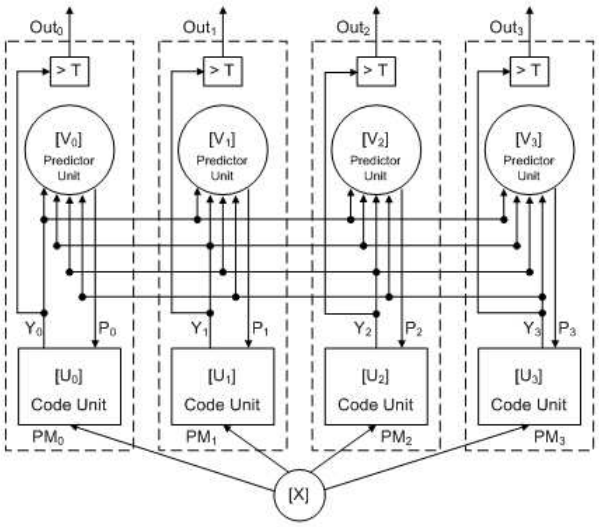

Fig. 2. Block level diagram of a PM based system with 4 modules.

code unit tries to react to the environment in such a way that minimizes its predictability. Thus, the predictor and the code units co-evolve to identify independent features from the input data.

The predictor and the code units can be trained using any of a number of algorithms, as long as the key principle behind PM is realized: the predictor units try to predict the output of their corresponding code units based on the outputs of all the other code units in the network while the code units try to avoid that prediction. One of the simplest means to achieve this behavior is by adjusting the predictor unit's weights to minimize the error between the predictor's output and the corresponding code unit's output and to adjust the code unit's weights to maximize the error between the code unit's output and the predictor unit's output.

Figure 2 shows the general structure of such a system. In the figure, X is the input to the system, Y^i is the output of the code unit of the i^{th} module and P^i is the output of its predictor unit. U^i is the weight vector of the code unit and V^i is the weight vector of the predictor unit of the i^{th} module.

$$Y^i = \sum_{j=1}^{M} U_j^i X_j^i \tag{1}$$

Y^i is actually the dot-product between the weight vector of the code unit and the input vector X. Here, M is the number of elements of the vectors U^i and X^i. Similarly,

$$P^i = \sum_{j=0, j\neq i}^{N} V_j^i Y^j \tag{2}$$

Here, N is the number of PM modules. P^i is actually a linear combination of the outputs of the code units of all the other PM modules in the network.

To update the weights, the predictor units try to minimize the same objective function that the code units try to maximize. This objective function is,

$$E^i = (P^i - Y^i)^2 \tag{3}$$

Using gradient descent, the predictor unit of the i^{th} module updates V^i in order to minimize E^i while the code unit updates U^i to maximize E^i. Thus, both the code unit and the predictor unit co-evolve battling against one another and result in modules that train themselves without supervision to identify independent features from the data.

An important aspect of any learning system is its stability criterion. To ensure the stability of our PM network in the steady state, we use the following rule to update the weights of the code units.

$$\hat{U}^i = U^i + \frac{\partial E^i}{\partial U^i} \times \frac{1}{\prod_{j=1}^{M} |U_j^i|} \tag{4}$$

As the elements of U^i become strong, they resist any change because of the $\frac{1}{\prod_{j=1}^{M} |U_j^i|}$ term in the learning rule. Finally, the i^{th} PM module fires as a whole in response to a feature only if the output of its code unit is greater than a predefined threshold i.e. $Y^i > T$

B. Predictability Minimization and Properties of Cortical Columns

The properties of a PM module suggest that it very closely models the functional properties of a cortical column. Since the code unit of each of the modules tries to avoid the predictions made by its predictor unit, it trains itself without supervision to identify independent features from the training data. Second, the PM modules that provide redundant data thrash in search of features that are not being identified by other modules. In steady state, the PM modules providing redundant information can be pruned from the network. Finally, a PM network is not dependent upon the nature of the training data. No matter what type of features are present in the data–visual, auditory, somatosensory, etc.–a PM network can train itself to identify independent features from that data. The learning rule that we developed to update the code unit's weights mimics the learning behavior of a cortical column. Columns that detect a feature very strongly always resists changes within the synapses of their neurons due to any new features.

The predictor units mimic the functionality of Layers II and III of the cortical columns by observing the behavior of columns for a certain event and communicating that information across the network. On the other hand, the code units mimic the functionality of the Purkinje cells found in the cerebellum [1]. Purkinje cells modify their synaptic weights if they receive simultaneous Excitatory Post Synaptic Potentials (EPSP). In simple terms, if a Purkinje cell receives simultaneous activations at two of its inputs it modifies its behavior so that those simultaneous activations are avoided in the future. This is quite similar to the working of the code unit, which modifies its behavior so that it is not activated for the inputs that result in the activation of the corresponding predictor unit (in effect, applying LTD to synapses that co-occur with the output of its predictor unit). These properties of a PM network make it very flexible and powerful and at the same time biologically plausible. Figure 3, shows the

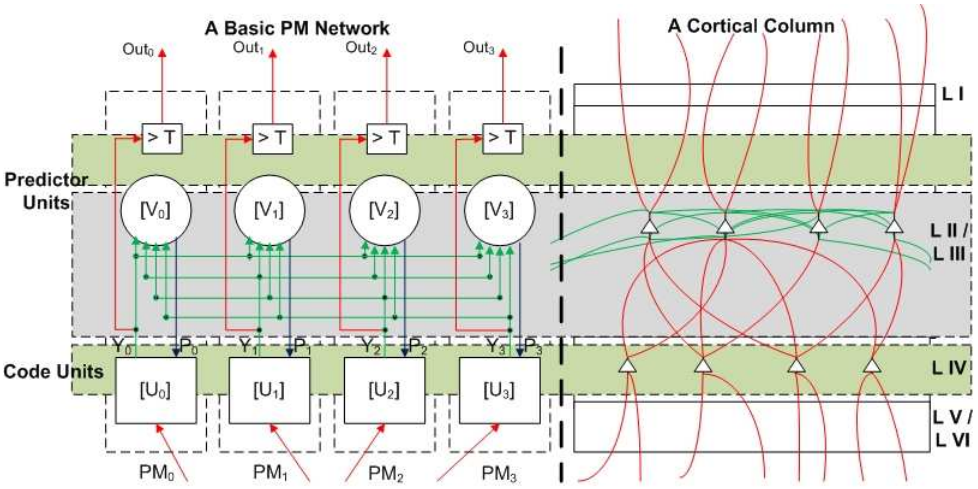

Fig. 3. Mapping between a PM network and Feedforward circuitry of a Cortical Column.

mapping between different components of a PM network and a cortical column.

C. Simulation Methodology and Results

We developed an infrastructure in C++ to simulate the working of single- and multilevel PM networks. The network was initially trained using the data set shown in Figure 4. Each dataset element is represented by a 2x2 matrix having +1.0 for white and -1.0 for a black quadrant.

Initially, we studied a single level PM network. The single-level configuration with 8 PM modules was initialized with random weights in each code and predictor unit. After 500 training epochs/iterations, the network showed the following behavior.

- Six out of eight PM modules trained themselves to detect the features shown in Figure 4.
- The weights of the code units of these modules had very strong correlations with the feature they were firing for.
- Code units of the remaining two modules kept thrashing during further training iterations after the steady state was achieved by the network.

These initial results clearly verify the properties of the PM network mentioned in the previous section and show the power of such a network.

Figure 5 shows the number of iterations taken by each of the eight code units to lock itself to identify a feature. In the figure, the number of iterations are along x-axis while y-axis has the absolute value of the code unit's weight. The i^{th} unit is said to lock itself to identify an independent feature when $\forall_j |U^j| = 20.0$. To keep the code unit's weights from becoming very large, they are clamped at 20.

From Figure 5, we can see that modules 0–3 lock themselves to identify an independent feature after a very few learning epochs. Modules 4 and 5 show some initial instability because they try to lock themselves to identify a feature that is being detected by some other unit, but quickly stabilize. Unit 6 and 7 keep trying to identify independent features but keep on thrashing as all of the six independent features have already been identified by some other module.

V. A Hierarchical Model for an Intelligent System

In order to identify complex features, we need to have a hierarchical organization of PM networks. Lower hierarchical layers identify simple features and communicate their output to the higher cortical regions. Based on the output of the lower PM layers, higher PM layers train themselves to identify complex independent features. There can also be feedback paths from higher layers to lower layers. Using these feedback paths, the higher layers can change the value of the threshold parameter of the lower layer to modulate their response for a certain input based on the prediction made by the higher PM layer. A typical hierarchy of PM modules is shown in Figure 6. The inherent

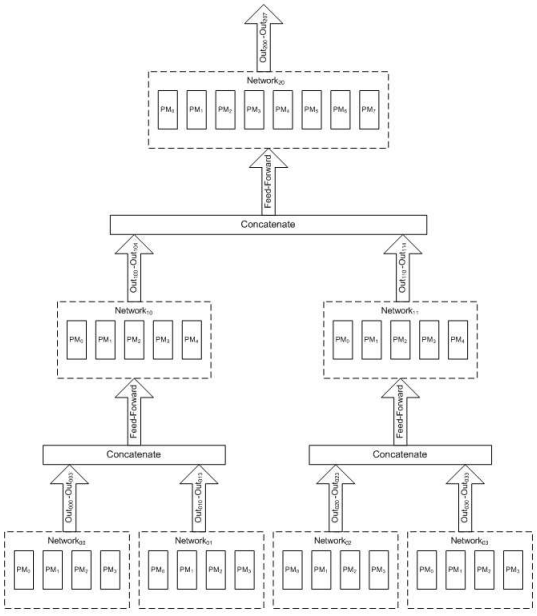

Fig. 6. A hierarchal structure of PM networks.

properties of a hierarchical PM network match neocortical properties like structural and functional uniformity, feedback,

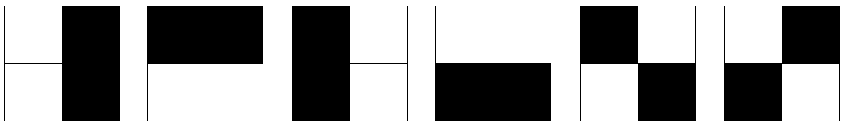

Fig. 4. Dataset used to train a single layer PM network of 8 modules.

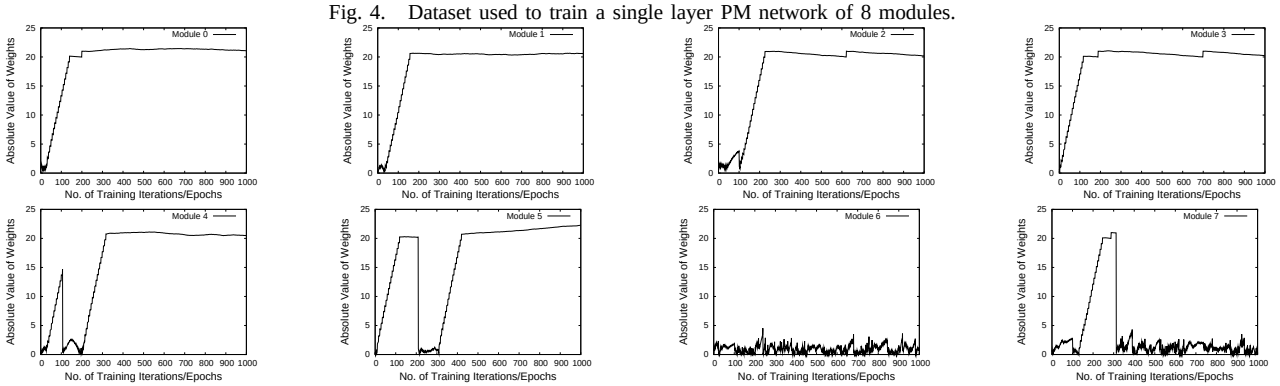

Fig. 5. Gradual increase in the absolute values of the code unit's weights of the 8 modules as the units lock themselves to identify independent features from the input data. The graph shows the number of learning iterations/epochs taken by each of the 8 modules to lock itself to identify an independent feature.

prediction, attention, and plasticity. Each of the hierarchical layers consists of a network of PM modules each running the same algorithm. This conforms to the structural and functional uniformity of the neocortex. The feedback paths from the higher PM networks to the lower PM networks take predictions from higher layers to the lower ones. It is very simple to simulate attention is such a system. To extract more information from a dataset, we can allocate more PM modules to different layers of the hierarchy to identify more independent features. If these newly added modules stabilize to identify a new feature, we have successfully extracted more details out of the data. Otherwise, the newly added units will keep on thrashing which will suggest that the dataset has no new features in it, and the units can be reclaimed for other purposes. Finally, adding new PM modules at runtime to extract more independent features from the dataset and removing modules representing redundant information, suggests that the system is capable of structural (or dendritic) plasticity.

A. Simulation Methodology and Results

To test the operation of a hierarchy of PM networks, we created a feedforward PM hierarchy of 3 levels similar to the one shown in Figure 6. Level 0 has eight PM networks, each network having 8 modules. Level 1 has four PM networks, each network having 10 modules. Level 2 has two PM networks, each network having 10 modules. Level 3 has one PM network and has 10 modules as well. To date, we have not utilized prediction or feedback paths in this hierarchy, and leave that to future work. The hierarchical PM network is trained using datasets shown in Figure 7. Each element of the dataset is represented using a 8x8 matrix with values either being +1.0 or -1.0. Initially, this matrix is split into eight 4x2 matrices and each of these matrices is used as a input to one of the networks in the Level 0 of the hierarchy. The output of the networks at Level 0 is concatenated and is used as an input to the networks at Level 1. Similarly, the outputs of the networks at Level 1 are concatenated and are used as an input to Level 2 network. Finally, the outputs of Level 2 are concatenated and are used as input to Level 3. After 2000 learning iterations/epochs, the following behaviors were observed:

- Each of the eight networks at Level 0 of the hierarchy trained itself to identify basic independent features from the dataset.
- The networks at Level 1 trained themselves to identify complex features based on the output of multiple Level 0 networks.
- The networks at Level 2 trained themselves to identify complex features based on the output of multiple Level 1 networks.
- The Level 3 network trained itself to identify each of the elements of the dataset shown in Figure 7.

The results for four out of eight Level 0 networks, two out of four Level 1 networks and the only network of Level 3 are shown in Figures 8, 9, and 10. Other networks in the hierarchy showed similar behavior. Due to space limitations, we have not shown the results for all the networks in this paper.

B. Learning to Forget

A very important feature of the neocortex is that it forgets. If certain patterns that we learn to identify no longer occur, the columns that detect those patterns start to forget them. This means that if a pattern does not appear for a long time, the column allocated to detect that pattern will start to change its behavior and after some time it will stop detecting that pattern. The main advantage of learning to forget is that it frees up resources. This is very important, since otherwise the network may grow indefinitely and eventually fail. We add this property to our model by using a decaying function

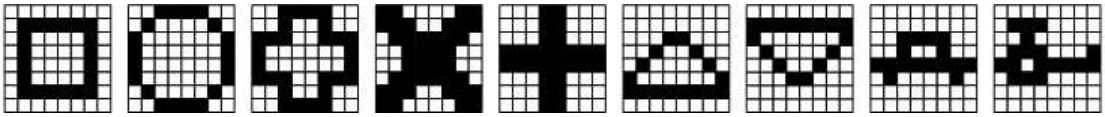

Fig. 7. Dataset used to train the hierarchical structure of PM networks.

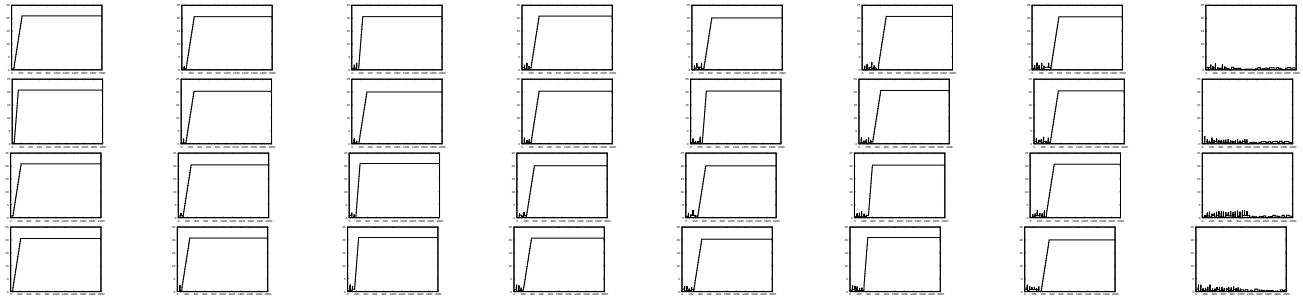

Fig. 8. Number of iterations taken by PM modules in 4 out of 8 networks at Level 0 of the Hierarchy to lock on to identify independent features from the input data. 7 PM modules from each network lock on to identify a feature while one module from each network keeps on thrashing.

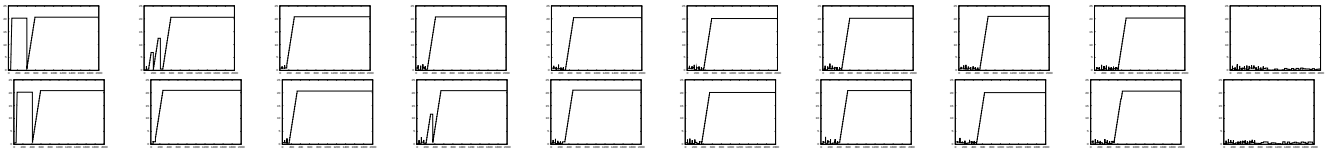

Fig. 9. Number of iterations taken by PM modules in 2 out of 4 networks at Level 1 of the Hierarchy to lock on to intermediate shapes from the input data. 7 PM modules from network 0 lock on to identify a feature while 3 modules from network 0 keep on thrashing while 9 PM modules from network 1 lock onto identify a feature while 1 keeps on thrashing.

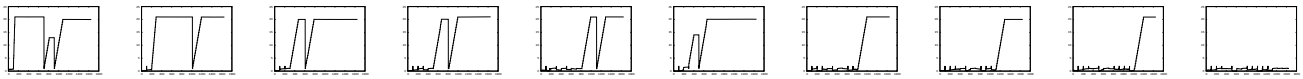

Fig. 10. Number of iterations taken by PM modules at Level 3 of the Hierarchy to lock on to identify the shape from the input data. There is just 1 PM network with 10 modules at level 3 of the hierarchy. 9 PM modules from network 2 lock on to identify a feature while 1 module from network 2 keeps on thrashing.

that decays the weights of the code units if the pattern they are trained to detect stops appearing.

We tested the *learning to forget* property in the hierarchical PM network. The behavior of PM modules of Level 3 of the hierarchical network is shown in Figures 11. Initially, the network was trained with the 9 independent features shown in Figure 7. After the initial training iterations, 9 modules at Level 3 of the hierarchy started identifying each of the 9 independent features. Then, we removed one of the features from the dataset and continued the training process. The module identifying that feature started to forget the feature that was not being seen by the network anymore and eventually, the weights were reset to detect a new feature. After some training epochs, we reintroduced that feature in the dataset and a PM module in level 3 started identifying that feature. The forgetting property is reflected throughout the hierarchy.

VI. Conclusion and Future Work

Current generation computing systems suffer from three main issues. They do not perform intelligent computing, due to power dissipation issues they will no longer see frequency scaling, and due to technology scaling the basic component of processing systems is getting quite unreliable [2]. This leads computer designers to explore processing models that are inherently intelligent and fault-tolerant, and no other processing system is as intelligent and as fault-tolerant as the human neocortex.

Researchers have put in a lot of effort to develop intelligent systems influenced by the properties of the human brain. These systems include artificial neural networks[15], Bayesian Networks[10], Hebbian learning models[19], Hierarchical Temporal Memory[14], etc. There are two major shortcomings of these systems: first, they focus on neurons as functional building blocks, rather than cortical columns; second, these systems do not model some properties of the neocortex that are, in our view, essential for intelligent systems. These properties include feedback, prediction, attention, plasticity, independent feature detection, dimensionality reduction, and training data independence. In this initial paper, we justify our hypothesis that cortical columns are the basic functional unit of the neocortex and that intelligent systems that model cortical columns as their basic functional unit can demonstrate neocortical properties essential for intelligence. Then, we present a model that extends the idea of Predictability Minimization to develop modules that mimic the working of cortical columns. Using our simulation infrastructure, we develop a network of these modules and we show that these modules are capable of

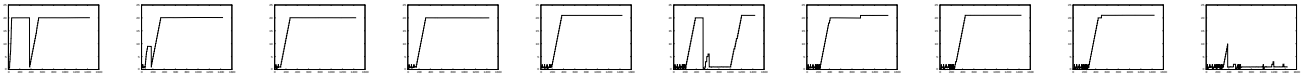

Fig. 11. Number of iterations taken by PM modules at level 2 of the hierarchy to lock on to identify the shape from the input data. There is just 1 PM network with 10 modules at level 2 of the hierarchy. 9 PM modules from network 2 lock on to identify a feature while 1 module from network 2 keeps on thrashing. 1 module shows forgetting behavior once a pattern stops appearing.

independent feature detection, dimensionality reduction, and training data independence. Networks of these structurally uniform modules can be arranged in the form of a hierarchy to detect complex independent features from more complex data set. Our model also shows the ability to forget things which is an important feature of the human neocortex.

Our framework is quite capable of incorporating additional critical properties like feedback, prediction, attention, and plasticity, but our simulation infrastructure does not yet implement these properties. In the future, we plan to incorporate these properties within our hierarchical simulation infrastructure. We want to study and understand the flexibility and power added to the system due to these properties. Another direction that we want to explore is to use our PM networks for Independent Component Analysis (ICA)[5] tasks. The inherent properties of the PM network like independent feature identification suggest that it can perform such tasks quite efficiently.

REFERENCES

[1] A. Arata and M. Ito. Purkinje cell functions in the in virto cerebullum isolated from neonatal rats in a block with the pons and medulla. *Neuroscience Research*, 50:361–367, 2004.

[2] S. Borkar. Designing reliable systems from unreliable components: The challenges of transistor variability and degradation. *IEEE Micro*, 25:10–16, 2005.

[3] W. Calvin. Cortical columns, modules, and hebbian cell assemblies. In Michael A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 269–272. MIT Press, Cambridge, MA, 1998.

[4] M. Carandini. Amplification of trial-to-trial response variability by neurons in visual cortex. *PLoS Biology*, 2:e264, 2004.

[5] C. Clopath, A. Longtin, , and W. Gerstnerm. An online hebbian learning rule that performs independent component analysis. In *Proceedings of Neural Information Processing Systems*. Neural Information Processing Systems, 2007.

[6] T. Dean. A computational model of the cerebral cortex. pages 938–943, 2005.

[7] M. Escobar and J. Ruiz del Solar. Biologically-based face recognition using gabor filters and log-polar images. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2, pages 1143–1147. International Joint Conference on Neural Networks, 2002.

[8] D. Felleman and D. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1:1–47, 1991.

[9] D. George and J. Hawkins. A hierarchical bayesian model of invariant pattern recognition in the visual cortex. In *Proceedings of International Joint Conference on Neural Networks*, volume 3, pages 1812–1817. IEEE International Joint Conference on Neural Network, 2005.

[10] Z. Ghahramani. Learning dynamic bayesian networks. In C. Giles and M. Gori, editors, *Adaptive Processing of Sequences and Data Structures*, pages 168–197. Springer-Verlag, Berlin, 1997.

[11] S. Grillner. Bridging the gap from ion channels to networks and behavior. *Current Opinion in Neuroscience*, 9:663–669, 1999.

[12] S. Grossberg. How does the cerebral cortex work? learning, attention, and grouping by laminar circuits of visual cortex. *Spatial Vision*, 12:163–185, 1999.

[13] S. Grossberg. Towards a unified theory of neocortex: Laminar cortical circuits for vision and cognition. In J. Kalaska P. Cisek, T. Drew, editor, *Computational Neuroscience: From Neuron to Theory and Back Again*, pages 79–104. Elsevier, Amsterdam, 2007.

[14] J. Hawkins and S. Blakeslee. *On Intelligence*. Henry Holt & Company, Inc., 2005.

[15] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ, USA, 1999.

[16] J. Hirsch and L. Martinez. Laminar processing in the visual cortical column. *Current Opinion in Neurobiology*, 16:377–384, 2006.

[17] D. Hubel and T. Wiesel. Receptive fields, binocular interactions and functional architecture in cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

[18] D. Hubel and T. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology*, 195:215–243, 1968.

[19] R. Kempter, W. Gerstner, and J. Hemmen. Hebbian learning and spiking neurons. *Physical Review*, 59:4498–4514, 1999.

[20] F. Kimura, M. Fukuda, and T. Tsumoto. Ach suppresses the spread of excitation in the visual cortex revealed by optical recordin: Possible differential effect depending on the source of input. *European Journal of Neuroscience*, 11:3597–3609, 1999.

[21] G. Kreiman, C. Koch, and I. Fried. Category-specific visual responses of single neurons in the human medial temporal lobe. *Nature Neuroscience*, 3:946–953, 2000.

[22] T. Lee, D. Mumford, R. Romero, and V. Lamme. The role of primary visual cortex in higher level vision. *Vision Research*, 38:2429–2454, 1998.

[23] V. Mountcastle. An organizing principle for cerebral function: The unit model and the distributed system. In G. Edelman and V. Mountcastle, editors, *The Mindful Brain*. MIT Press, Cambridge, Mass., 1978.

[24] V. Mountcastle. The columnar organization of the neocortex. *Brain*, 120:701–722, 1997.

[25] V. Mountcastle, A. Berman, and P. Davies. Topographic organization and modality representation in first somatic area of cat's cerebral cortex by the method of single unit analysis. *American Journal of Physiology*, 183:646, 1955.

[26] J. Nicholls, A. Martin, B. Wallace, and F. Fuchs. *From Neuron To Brain*. Sinauer Associates Ins, 23 Plumtree Road, Sunderland, MA, USA, 2001.

[27] B. Olshausen, C. Anderson, and D. Van Essen. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *The Journal of Neuroscience*, 13:4700–4719, 1993.

[28] F. Rohrbein, J. Eggert, and E. Korner. A cortex-inspired neural-symbolic network for knowledge representation. In *Proceedings of International Workshop on Neural Symbolic Learning and Reasoning*, volume 230. International Workshop on Neural Symbolic Learning and Reasoning, 2007.

[29] G. Roth and U. Dicke. Evolution of brain and intelligence. *TRENDS in Cognitive Sciences*, 5:250–257, 2005.

[30] J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4:863–879, 1992.

[31] J. Schmidhuber. Neural predictors for detecting and removing redundant information. In H. Cruse, J. Dean, and H. Ritter, editors, *Adaptive Behavior and Learning*, pages 135–145. Center for Interdisciplinary Research, Universität Bielefeld, 1994.

[32] W. Singer and C. Gray. Visual feature integration and the temporal correlation hypothesis. *Annual Reviews Neuroscience*, 18:555–586, 1995.

[33] H. Stephan, H. Frahm, and G. Baron. New and revised data on the volumes of brain structures in insectivores and primates. *International Journal of Primatology*, 35:1–29, 1981.

[34] S. Stringer and E. Rolls. Invariant object recognition in the visual system with novel view of 3d objects. *Neural Computation*, 14:2585–2596, 2002.

[35] K. Tsunoda, Y. Yamane, M. Nishizaki, and M. Tanifuji. Complex objects are represented in macaque inferotemporal cortex by the combination of feature columns. *Nature Neuroscience*, 4:832–838, 2001.

[36] S. Ullman and S. Soloviev. Computation of pattern invariance in brain-like structures. *Neural Networks*, 12:1021–1036, 1999.

[37] H. Wersing and E. Korner. Learning optimized features for hierarchical models of invariant object recognition. *Neural Computation*, 15:1559–1588, 2003.

[38] L. Wiskott and P. Berkes. Is slowness a learning principle of the visual cortex. *Zoology*, 106:373–382, 2003.