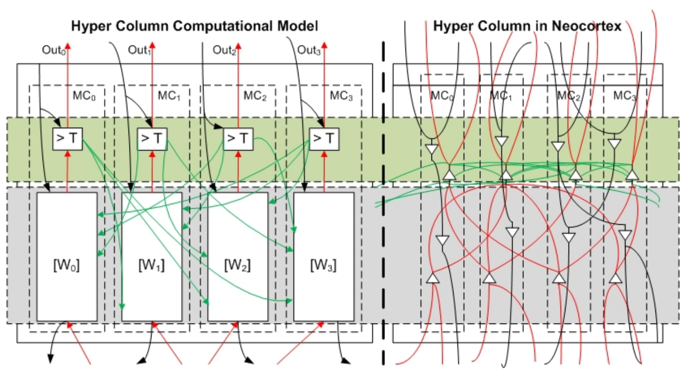
Discovering Cortical Algorithms

Atif Hashmi and Mikko Lipasti
Department of Electrical and Computer Engineering, University of Wisconsin – Madison
Email: ahashmi@wisc.edu, mikko@engr.wisc.edu

Since its creation, the classical Von Neumann model of computation has been a relatively nice fit for general purpose as well as application-specific computational systems. In recent years reliability, power dissipation, and error-tolerance challenges in current and future implementation technologies have raised concerns about the continued viability of the Von Neumann model. These concerns have made computer architects open to investigating alternative computational models better suited to cope with the current issues encountered in technology evolution. With these concerns in mind, it is hard for one not to observe that in the form of the human neocortex, nature has found a way to harness a large number of elements with properties reminiscent of future hardware technologies to realize complex information processing tasks. Apart from being an extremely robust parallel processing system, the human neocortex is inherently error-tolerant and power-efficient. Due to neuroscientific discoveries, understanding of various properties of the neocortex has significantly increased over the recent years. This suggests that it could be time to leverage some of this progress for investigating designs of special-purpose computing systems modeled after the neocortex.

While neurons are the basic structural component of the neocortex, cortical or hyper columns, as proposed by Mountcastle [2], provide a very useful implementation abstraction for a computational system. These columns are replicated all over the neocortex to form hierarchies [3] suggesting that the neocortex utilizes a common, wide spread, and repeatedly applied underlying algorithm to perform complex processing tasks. Thus, understanding the architecture and behavior of these columns and the way they interact with each other can help us in implementing power efficient and error-tolerant cortically-inspired computational models.

Figure 1 compares the basic unit of our cortically inspired computational model [1] with the structure of a hyper-column. Structurally and functionally, there are number of commonalities between our model and a neocortical hyper-column. First, within each hyper-column there are multiple mini-columns. These mini-columns interact with each other via the horizontal inhibitory connections and feed-forward excitatory connections and learn to identify independent features in their receptive field. The mini-columns also receive excitatory and inhibitory feedback signals from higher cortical regions. These hyper-columns are arranged in hierarchies to realize complex processing tasks.



In our model, feedback serves two purposes. First, it is used to generate invariant representations. Using a variant of supervised learning, feedback connections notify the lower level mini-columns to pool variations of a pattern together. For example, the mini-columns at the top level receive feedback from the supervisor and pool different lower level mini-columns that fire for different variation of a pattern. Once the top level is stable, feedback to the lower levels indicate that the mini-columns at the lower level should start pooling their child mini-columns. This process continues until pooling is transferred to the mini-columns at the lowest level in the hierarchy. A primary advantage of our feedback-based pooling mechanism is that it frees up resources. The amount of pooling must be balanced by an un-pooling mechanism so that desired independent features are not pooled together. Second, feedback can also be used for making future prediction based on previous experiences. This is useful in situations where context can be used to make correct decisions, particularly with noisy or degraded inputs.

We have tested our cortically inspired computational models with synthetic images and hand-written digits images obtained from MNIST database. Our model shows 100% recognition for training images. Second, our feedback based pooling algorithm effectively generates invariant representations and also improves the resource utilization of our model. In the future, we are planning to model feedback based prediction and temporal learning in our model.

References

1. A. Hashmi and M. Lipasti, Cortical Columns: Building Blocks for Intelligent Systems, Proceedings of the Symposium Series on Computational Intelligence, 21-28, 2009.
2. V. Mountcastle, An Organization Principle for Cerebral Cortex: The Unit Model and the Distributed System, The Mindful Brain, MIT Press, 1978.
3. V. Mountcastle, The Columnar Organization of the Neocortex, Brain, 120:701-722, 1997.