Leveraging Progress in Neurobiology for Computing Systems

Atif Hashmi[†], Hugues Berry[††], Olivier Temam[††], and Mikko Lipasti[†]
ahashmi@wisc.edu, hugues.berry@inria.fr, olivier.temam@inria.fr, mikko@engr.wisc.edu
[†]*University of Wisconsin-Madison, USA* [††]*INRIA, France*

Whether future technologies will be ultra-small CMOS transistors, nanotubes, or even individual molecules or biological cells, these elementary components all share several common properties: they come in great numbers, they won't be much faster or may even be way slower than current transistors, they may be hard to precisely lay out and connect, and they may be faulty. The key question, then, is how can one design and, more importantly, program a computing system using billions of such components while we are not even capable of harnessing a few hundreds traditional cores ?

Now, when one considers these properties, it is almost irresistible to observe that nature has found a way to harness a huge number of elements with similar properties to realize complex information processing tasks. While suggesting that we design computing systems which somehow imitate parts of the brain is such an old *cliché* that most computer scientists are embarrassed to bring it up, biologists may force us to reconsider and rectify this behavior. The fact that biologists have made tremendous progress in understanding the working of parts of the brain [2] is not yet well-known to computer architects. It could be time to leverage some of this progress for designing at least special-purpose computing systems. And the simple fact remains that large-scale biological networks are far more efficient, time-wise and power-wise, than classic computing systems for a number of important applications.

Several decades ago, computer architects had teamed up with chemists (e.g., Grove, Moore, Noyce) to breed the first microprocessor. More recently, they have started investigating quantum computers based on the progress of physicists [3]. Similarly, they are well positioned to translate the progress of biologists into potentially usable computing systems. Naturally, both the elementary components and the resulting computing systems would be drastically different from current systems, but many of the research tasks at hand are similar in spirit: understanding how to combine and control elementary components hierarchically into increasingly complex building blocks, defining a programming approach for these computing systems, understanding their potential applications scope, understanding the appropriate design abstraction level allowing to manipulate billions of components without being overwhelmed by complexity nor missing key properties, and so on. Naturally, we do not suggest that such cortical microarchitectures can become general-purpose computing systems, in spite of their Turing completeness [1]. For now, we advocate their use for application-specific purposes, in the spirit of ASICs or moderately programmable systems.

A particularly promising research direction is learning to replicate cortical sensory tasks, even complex ones, such as vision, only using elementary components, such as hardware neurons, even if they are rigged with faults and defects. Biologists have now partially explained how such functions can emerge, and operate, within the brain. A key concept, drastically different from normal computing systems operations is the notion of *abstraction*, the ability to automatically extract more complex notions out of a large set of elementary data. For instance, it is now believed that the visual cortex creates abstractions corresponding to complex shapes, and to invariance properties of shapes (position, size, rotation), through a simple, hierarchical and repetitive arrangement of neurons and sensory/inhibitory connections. Even better, biologists have proposed detailed models showing how neurons should be connected for such abstraction operations to emerge. Because the necessary neural and synaptic organization is largely stochastic, and because additional neurons and synapses can directly translate into augmented capabilities (e.g., recognize more shapes or more nuances among subcategories of shapes), these architectures have intrinsic *scalability* properties. And they raise none of the complex programming challenges of parallel systems, as the programming consists in unsupervised learning: simply the repeated exposure to target data.

Using the example of vision processing, we have started constructing two computational models emulating the automatic abstraction process and applied it to simple classification tasks: a detailed neuron-level model validated by knowledge in biology, and a more high-level model validated by the first model but more computationally efficient. Both models can already be implemented using the same hardware neurons used for artificial neural networks, but the operations and structures of these bio-inspired models are significantly different from ANNs.

Our biologically plausible model is derived from the model of Riesenhuber et al. [4] which itself relies on physiological data of large-scale biological neural networks. However, in that model, the neurons are connected in a custom manner, with the purpose of breeding specific combinations of operations which let complex abstractions automatically emerge from the input data. For both biological plausibility *and* computational reasons, we introduce a novel model, similarly capable of achieving automatic abstraction, but

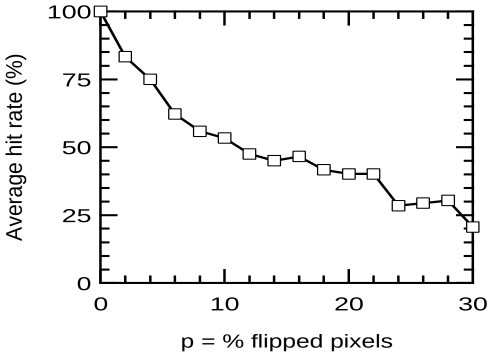

Figure 1. *Neuron based model's recognition efficiency and robustness on numerical digits.*

which accommodates a random topological layout of neurons and random connections among neurons. As neurons are composed together across layers, they correspond to increasingly complex abstractions. In practice each neuron connects to several tens of neurons of the lower layer on average, so abstractions emerge with even fewer hierarchy levels. Except for the connections between the input layer and the first layer, the neural connections can be both excitatory and inhibitory. This has the impact of creating complex *logical combinations* of abstracted information at any level of the hierarchy. To assess the efficiency and robustness of our networks, input images (digits) were blurred by randomly flipping an increasing number of pixels. Figure 1 shows the recognition hit rate as a function of the percentage of flipped pixels, p; the recognition hit rate is averaged over 200 image exposures for each probability value.

A major shortcoming of our biologically-plausible approach is it computational complexity. To recognize complex patterns, a large number of neurons is required which increases the computational demands of this model quite significantly. We use the concept of abstraction to solve the computational complexity problem. Our computationally efficient model implements the working of cortical columns as its basic fucntional abstraction. First, this model implements the transformations that affect the visual input as it is transferred from the retina to the primary visual cortex through the optical pathways. Second, we create a competitive learning based hierarchical network that uses transformed visual data as input. We tested our model with a sample of handwritten digits (0-9) obtained from the MNIST Database (http://yann.lecun.com/exdb/mnist). To study the performance of our hierarchy in the presence of noisy data, we added random noise to the digit dataset and exposed it to the hierarchy after it had been trained using the noise free dataset and evaluated the average recognition hit rate.

We studied the hit rates of the hierarchy by using following receptive field configurations for the Level 0 (L0) hypercolumns.

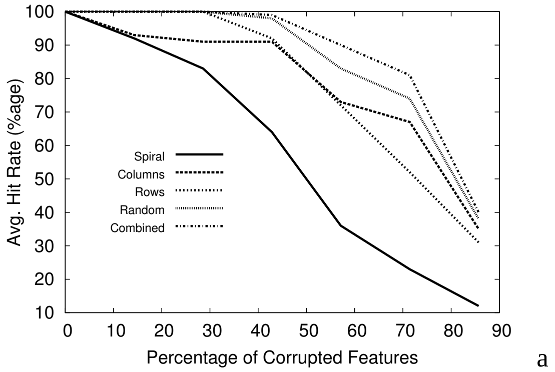

Figure 2. *Column based model's recognition efficiency and robustness on numerical digits for different L0 receptive field configurations.*

- **Rows:** Each row of the average LP image is used as the receptive field for each of the L0 hypercolumns.
- **Column:** Each column of the averaged LP image is used as the receptive field for each of the L0 hypercolumns.
- **Spiral:** Each L0 hypercolumn receptive field consists of unique pixels selected in a spiral manner.
- **Random:** Each L0 hypercolumn receptive field consists of randomly selected unique pixels.

We also constructed a complex hierarchical network that consists of three redundant networks each looking at the same average LP image of digits but using a different receptive field configuration and an additional layer at the top to associate all three networks. Here, we exploit the redundancy property of the neocortex and refer to this complex hierarchy as *combined*. Figure 2 shows the average recognition hit rate of the hierarchy as a function of percentage of randomly corrupted features for each of the above mentioned cases.

Due to the generic organization of large-scale biological networks, it is likely that the proposed neural architectures can be applied to other and/or more complex sensory tasks. And as we learn to scale and combine these cortical microarchitectures, we also aim at progressively approaching the complexity level of tasks handled by large-scale biological neural networks.

References

[1] G. V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[2] E. Kandel, J. Schwartz, and T. Jessell. *Principles of Neural Science*. McGraw-Hill, 4 edition, 2000.

[3] R. V. Meter, K. Nemoto, W. J. Munro, and K. M. Itoh. Distributed arithmetic on a quantum multicomputer. In *ISCA '06: Proceedings of the 33rd annual international symposium on Computer Architecture*, pages 354–365, Washington, DC, USA, 2006. IEEE Computer Society.

[4] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.