RF-LISSOM CUDA

- Giacomo Spigler

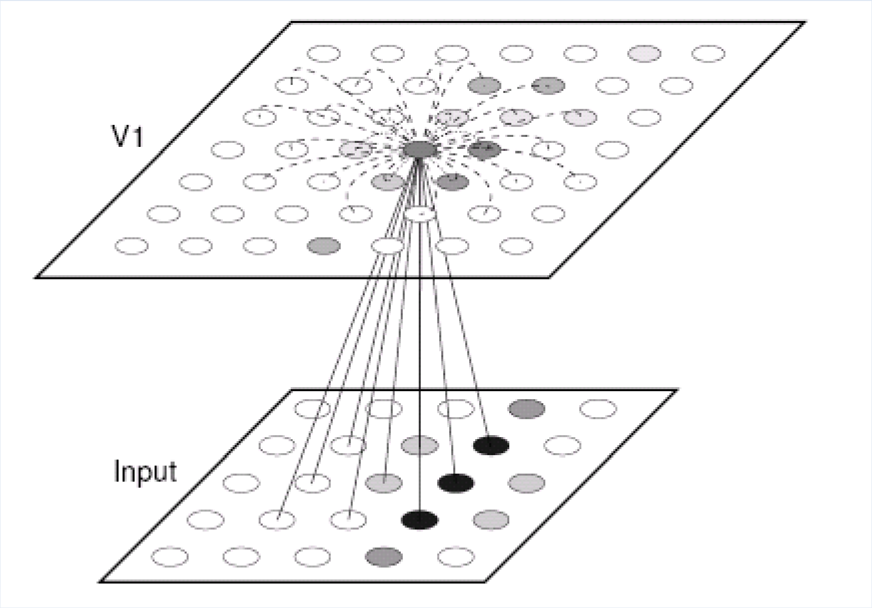
-An RF-LISSOM simulator which takes advantage of CUDA compatible GPUs.

-Up to 5x (10x peak) performance gain on GT200 GPUs

-Tesla compatible system (as many GPUs as you want can be used to speed up the simulation even more)

RF-LISSOM

-RF-LISSOM : Laterally Interconnected Synergetically Self-Organizing Map
-This special kind of SOM introduces lateral connections (short-range excitatory and long-range inhibitory) into the classical Self-Organizing Map framework, producing an accurate simulation of the neural cortex at neural columns level.
-This model proved successful in simulating visual phenomena occuring in Primary Visual Cortex V1, among whom we can list development of Orientation Selectivity, Ocular Dominance, Depth Perception, and a direct observation of the Tilt-After Effect.



-Moreover, the model can probably simulate visual processing in the brain up to the Object Recognition level, to Visual Segmentation and to Perceptual Grouping.

-The image on the left shows the typical connections being simulated by the model.

RF-LISSOM

-RF-LISSOM basic operations are quite easy to describe, but they actually need an enormous amount of memory and of computational power.
-RF-LISSOM can be divided into 3 steps: First Activation, Settling and Weights Adjusting.

-First Activation : Once an input is presented on the input layer, every LISSOM's neuron computes a first activation based on its current Afferent Weights. The result is then processed with a piecewise linear approximantion of the sigmoidal function.

-Settling : This phase consists of running a few (typical 9-13) steps, which will simulate lateral connections, and which will produce some stable blobs of neural activity, representing a stable state of the network (given the current input).

$$\eta_{ij}(t) = \sigma\left(s_{ij} + \gamma_{\mathrm{E}} \sum_{kl} \eta_{kl}(t-1) E_{kl,ij} - \gamma_{\mathrm{I}} \sum_{kl} \eta_{kl}(t-1) I_{kl,ij}\right)$$

n(t) is the firing strenght of neuron (i, j), and the 3 activations inside the sigmoidal function are: First Activation for neuron (i, j), Excitatory Activation with neuron (k, l) and Inhibitory Activation (neuron (k, l)).

RF-LISSOM

-Weights Adjusting : Weights are then modified according to the last neural activation recorded using Hebbian Rule.
They are then normalized to sum to 1; hence, the modification of a single weight it's compared to those of the other weights (of the same type, eg, afferent).
Every weight, on weights adjusting phase, either increases or decreases its **relative** strenght, gathering more importance.

$$w'_{pq,ij} = \frac{w_{pq,ij} + \alpha X_{pq} \eta_{ij}}{\sum_{uv}(w_{uv,ij} + \alpha X_{uv} \eta_{ij})}$$

CUDA Simulation

-Here are shown the main features of the CUDA implementation:

-Every layer contains three objects, called Projections, representing its different weights types: Afferent, Excitatory and Inhibitory

-Every Projection's function is almost fully interexchangable with the other ones (eg, adjusting afferent weights is accomplished with the same code as used to adjust inhibitory weights; the only things we change are parameters).

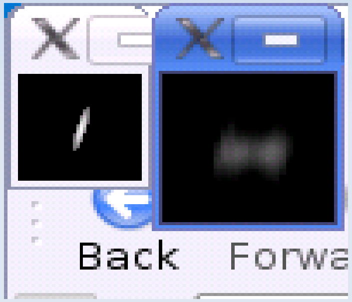
-RF-LISSOM is intrinsically a parallel problem, as Neural Networks always are; hence, the highly parallel architecture of CUDA and CUDA compatible GPUs lead to a great improvement in speed.

-The main bottleneck is memory: every iteration consists of tens of thousands, and peraphs millions of sparse access to weights in memory, which prevents from coalescing memory reads and writes.

-A partial solution to this problem has been found using Textures bounded to linear memory for fast, cached reads, and memory writes were reduced.

Examples

-After training on synthetic data (elongated gaussians), we can test both what the network looks like and how it behaves.
-Here is the network's activity when an input is presented (shown at the left) before training.



-And this is what it looks like after Self-Organization occured (please notice the formation of little blobs, which are tuned to that specific orientation and won't respond to any other simulus).

Examples

-Now, let's have a look at LISSOM's weights after 20000 training steps.

-Excitatory Connections are shrinked during training:

-Inhibitory Connections develop among the neuron's preferred orientation, and help producing the final stable blobs.

-Afferent Connections clearly show their preferred orientation.

Conclusions and Future Work

-Research on RF-LISSOM and on its CUDA implementation will lead to important results in understanding how the visual cortex, and probably the whole neural cortex work.

-Next improvements will be:

-Stable support for multiGPU system (it's currently working, but it needs some fixing)

-Support for CUDA Clusters (clusters whose nodes are embedded with CUDA compatible hardware), through MPI.

-Testing and Research on many LISSOM layers connected into a hierarchy.

-Object Recognition and research on how the neural cortex processes complex data from the real world, producing stable representations of the environment.

-We are currently investigating some tests to study Tommaso Poggio's Standard Model with a Model of the Visual Cortex produced by a synthesis of RF-LISSOM and current knowledge of neuroanatomy of this portion of the brain.

-In conclusion, CUDA allows us to process increasingly bigger areas of the brain in much less time than what it would take on a classical CPU, and costs are kept down: I am actually an high school student, and I have no source of fundings. However, with CUDA, I can gather access to the computational power I need!

References

1) ``Computational Maps in the Visual Cortex``, Miikkulainen, Bednar, Choe, and Sirosh (New York: Springer 2005) http://nn.cs.utexas.edu/computationalmaps/

2) RF-LISSOM: http://homepages.inf.ed.ac.uk/jbednar/rflissom_small.html

3) CUDACluster: http://cudacluster.nvidia.com

4) nVidia: http://www.nvidia.com

5) CUDA: http://www.nvidia.com/object/cuda_home.html

6) Poggio's CBCL: http://cbcl.mit.edu

7) Poggio's Standard Model: http://cbcl.mit.edu/projects/cbcl/publications/ai-publications/2005/AIM-2005-036.pdf

RF-LISSOM CUDA

END