Neural Network Scalable Spiking Simulator

N2S3 is an open-source neuromorphic hardware simulator with the following properties:

- **Event-Based**: Event based processing, in contrast to clock-based processing allows the local processing of each event, and high accuracy of timestamp.
- **Scalable**: N2S3 uses the Akka project to enable massively concurrent programming. All simulator entities are (potentially) actors. This model fits well with the neural network processing paradigm. Thus a simulation is concurrent and can be distributed on a cluster.
- **Extensible**: N2S3 is designed to easily integrate new models and tools.
- **Portable**: N2S3 is written in the Scala programming language. It can thus run on any computer compatible with the Java Virtual Machine.
- **Open-Source**: This project is under the CECILL-B licence.

Why a Spiking Model?

Spiking models are a good compromise between the biology fidelity and computation efficiency.

- Simple models with few parameters
- Spike communication reduces bandwidth, and so energy consumption
- Good combination with STDP for unsupervised learning
- Possible energy efficient hardware implementation with memristors or even CMOS

Different Network Topologies in N2S3

Each application requires an adapted network topology. E.g.,

- Feed Forward Neural Networks
- Convolutional Neural Networks
- Reservoir Computing (in progress)

Currently working Input Formats

N2S3 is already integrated with the following input formats:

- Address Event Representation (AER)
- MNIST
And can be easily extended to read new formats.

## Release Notes

### Version 1.1

N2S3 1.1 is a quality improvement release. The highlights of this version are the introduction of a new Domain Specific Language (DSL), a reorganization of the existing neuron models and lots of cleanups.

**Core:**

1. Introduction of a DSL to simplify the task of creating a simulation
2. The different models N2S3 supports have been reorganised:
   1. Neuron Models are now located in the models.neurons package. The featured models are:
      1. Izhikevich ([http://www.izhikevich.org/publications/spikes.pdf](http://www.izhikevich.org/publications/spikes.pdf))
   2. Synapse Models are now located in the models.synapses package. The featured synapses are:
      1. Inhibitory Synapse
      2. Simplified STDP
      3. Standard STDP
      4. Static Synapse
      5. Ternary Synapse

**Cleanups:**

1. Removal of lots of compilation warnings
2. Removal of unused classes from the support package, old inputs
3. Removal of the deprecated connections creation in ConnectionPolicy and subclasses
4. Removal and cleanup of commented code

### Version 1.0

N2S3 is now feature complete, well tested and fully documented. External users are welcome!

N2S3 1.0 is mainly an infrastructure release oriented towards the ease of development and distribution. This version builds on version 0.2 with ~165 new commits! There are also some news in the core of the simulator and the documentation. Here you have the major changes:

**Infrastructure:**

1. N2S3 is now an SBT multi-project [1]
3. Supporting the system’s distribution:
   1. Clustering support
   2. Refactoring to support the dynamic deployment of actors in different cluster nodes.
   3. Two deployment strategies were implemented so far: Random and concrete node.
4. Fixes in the core to make it distributable. E.g., fixed serialization of messages, global state.
5. The Continuous Integration service was extended and enhanced [4]
6. A new input mechanism was implemented [5, 6]

Core:

1. Preliminary version of supervised learning algorithm SpikeProp
2. Preliminary version of Supervised Xor, an example of learning the Xor function with SpikeProp
   (Error-backpropagation in temporally encoded networks of spiking neurons)
3. Added bio-inspired model i.e., a model with parameters similar to those in the nature.

Examples:

1. Examples of Mnist/Freeway/sevenSegment with the bio-inspired model.
2. Example of Masquelier's experiment: Spike Timing Dependent Plasticity Finds the Start of
   Repeating Patterns in Continuous Spike Trains
3. Preliminary version of an example of direction detection with reservoir computing (using
   supervised STDP and only 2 directions)

Documentation:

1. The core scaladocs were entirely revisited
2. Technical documentation about the core implementation can be found in [7]

We are waiting for your contributions, And do not hesitate to share!

The N2S3 team