New Brain Simulator II Open-Source Software

Charles J. Simon(0000-0003-2858-6501)

Future AI, Washington, DC 20001, USA
charles@futureAI.guru

Abstract. This paper introduces the open-source software project, Brain Simulator II, simplifying experimentation into various facets of AGI. The software seamlessly marries spiking neural networks with symbolic AI algorithms. It supports a large array of simple neurons (of various models) and groups of neurons collected into "Modules", backed by custom software. 3D and 2D simulators allow a virtual entity to move about, have binocular vision and touch, and merge this information with spoken input. Information is captured in a Universal Knowledge Store module which represents information in links between nodes. Continuing development will enhance these capabilities.

Keywords. Artificial General Intelligence, Spiking Neuron Model

1 Focus of the Brain Simulator II Project

The focus of the Brain Simulator II is to facilitate experimentation into various facets of AGI beginning with biologically plausible techniques. The platform merges information from any number of sources such as sight, sound, and touch so an artificial entity can be tried out in a unified environment. This multi-sensory approach allows for experimentation which can contribute to AGI development.

The platform provides a large array of neurons interconnected with synapses, plus neuron areas declared as "Modules" as a shortcut to creating networks and implementing more complex computation. The remainder of this paper describes the project development status as of February 2020. Additional features are being developed including the ability for the entity to move objects in its environment to allow experimentation with goals, planning, and intentionality.

2 Implementation

2.1 The Basic Neuron Models

The program represents an array of artificial neurons interconnected by synapses. The default neuron type is an "Integrate and Fire" spiking model [Abbot 1999] which aggregates weighted synapse inputs and fires when a threshold is reached. This model is extremely efficient and has been tested in real time with a million neurons on a desktop

CPU. Other neuron models include a “leaky integrate and fire” model [Dutta 2017] and others. Additional neuron models can be created in a few lines of code.

2.2 The Neuron User Interface

The neuron display can be used to build neural circuits to explore the capabilities and limitations present in small clusters of neurons. Examples include small Hebbian learning and decoding neural pulse streams. The user interface can display relative timings of selected neuron firings.

In the neuron array, colors represent the firing state of each neuron. Optionally, individual synapses can be shown and edited. The complete state of the network can be edited, saved, and automatically restored like a document.

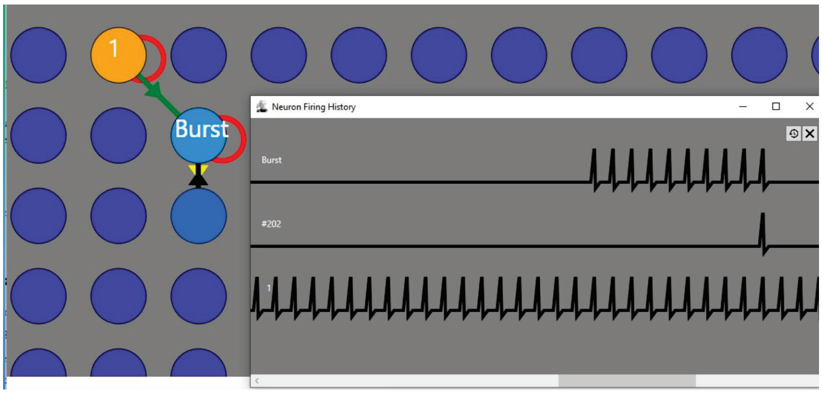


Fig. 1. The basic neuron model is “Integrate and Fire”. Small neural circuits can be created at the individual synapse level and firing history can be displayed.

2.3 Modules

Any cluster of neurons can be grouped together into a module backed by custom software and (optionally) a dialog box. A module can perform any desired computation but also can manipulate neurons and synapses throughout the network and may communicate directly with other modules via method calls. Each module has a primary method which is called once for each time-slice of the neuron simulator.

There is no *requirement* for biological plausibility within modules. For example, the binocular Vision module receives its input in the form of arrays of neural signals but estimates distances with a few lines of trigonometry rather than any biologically plausible technique. Visible features are then added with direct calls to the Internal Model module which also uses trigonometry to emulate the functionality of the brain’s Grid Cells [Haftig 2012] to handle the entity’s motion and rotation within the model.

2.4 Module Library

The current library of over thirty modules includes the following:

Simulators. The digital entity currently operates within a simulated 2D environment. The simulator supports physical objects, binocular vision, two-appendage touch, motion, rotation, aroma, and collisions. A simple file-command module allows for repeatable sequences of individual neuron firings for testing. A 3D version of this simulator has also been written.

Sensory Modules. For aroma, touch, vision, and speech-recognition, each module processes input from a specific sense. For example, the vision module handles color and uses bit patterns from its two “eyes” to approximate depth. The touch module can establish more accurate depth but cannot process color. The speech-recognition module uses the computer’s intrinsic speech library to fire neurons which represent a sequence of individual phonemes.

Universal Knowledge Store (UKS). This general-purpose knowledge graph supports any number of properties, relationships, etc. All relationships are many-to-many and relationships can be weighted so the knowledge store can learn over time. In keeping with the biological plausibility objective, information is represented in edges connecting the nodes and nodes themselves do not typically contain information at all.

Each node in the graph is associated with two neurons, one causes a node to be “activated” and another fires if the node is activated internally. For example, UKS input neurons fire in response to phonemes received from the speech engine and are separate from those which are connected to the speech synthesizer to enable speaking.

Internal Reality Model. This module is a layer above the UKS which handles physical objects in the “known world”. Input from surroundings via various senses is collected in the internal model. For example, distance information estimated from binocular vision can be corrected or superseded by touch information. An aroma can make some objects more attractive than others. Spatial relationships are maintained relative to the entity’s point of view as the entity moves or turns. Merging the information from multiple senses builds up a model with a better “picture” of the entity’s surroundings and lets it remember things which are not currently visible and imagine possibilities such as an alternate point of view.

Various Learning Modules. These operate on the UKS. Over time, it can correlate object properties with spoken words and can correlate situations, behaviors, and outcomes. Limited learning-by-imitation allows the system to learn to speak words and phrases after initially hearing itself speak random phonemes.

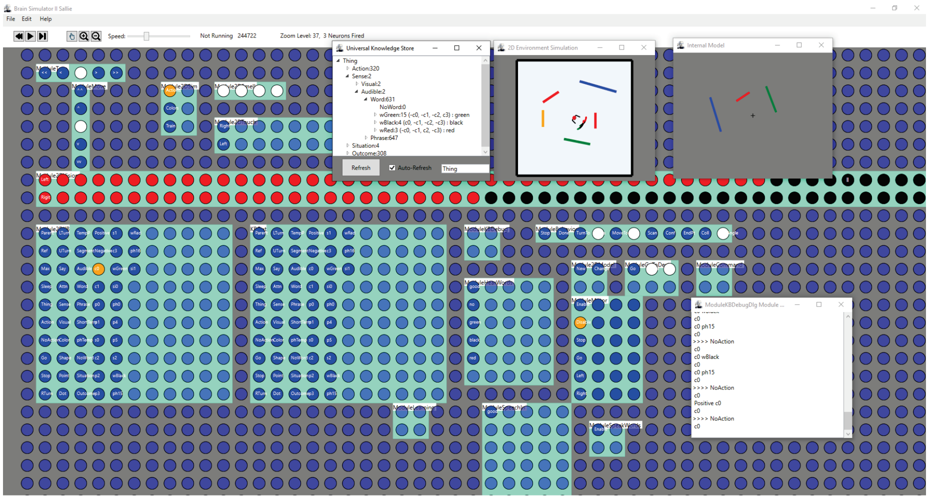
Behaviors. A library of primitive behaviors lets the entity interact with its environment. Primitive behaviors can be combined into sequences to create more complex behaviors. A similar module accepts the same primitives and interfaces with a mobile robot.

3 Project Status

The Brain Simulator II encompasses components of a variety of AGI models. [Laird 2012, Miller 2015, Simon 2018] The combined modules currently create a simple digital entity named "Sallie". Sallie can move through a simulated environment and use binocular vision to estimate distances which build an internal model of her surroundings which she can use to plan paths. She can learn to associate spoken words with colors and learn to associate spoken commands with behaviors.

Consistent with the incremental development strategy, there is an end-to-end process which forms the basis for future development in learning object comprehension and more interesting behaviors with many features yet to be filled in.

Applications. A collection of 10 small applications have been developed including navigating mazes and learning to talk via imitation. Applications share many common components. For example, the UKS structure which supports maze navigation also supports reinforcement learning for commands. That is, for a given situation, there is a collection of possible actions each leading to an outcome. In the maze application, the outcome is a destination. For reinforcement learning, the outcome is the state of an external reward signal which allows the system to learn "right" vs. "wrong" responses.



```
Thing
  Action:320
  Sense:2
    Visual:2
    Audible:2
      Word:631
        NoWord:0
        wGreen:15 (-c0, -c1, -c2, c3) : green
        wBlack:4 (c0, -c1, -c2, -c3) : black
        wRed:3 (-c0, -c1, c2, -c3) : red
      Phrase:647
  Situation:4
  Outcome:308
```

☑ Auto-Refresh: Thing

```
c0
c0 ph15
c0
>>>> NoAction
c0
c0 wBlack
c0
c0 ph15
c0
>>>> NoAction
c0
Positive c0
c0
>>>> NoAction
c0
```

Fig. 2. A screenshot of the Brain Simulator II shows the neural array with modules, some of which have specific dialog windows. The engine controls are in the upper left.

4 Unique in this Software

The basic neuron model calculates individual neural spikes and modules implement higher-level functionality which could conceivably be implemented in spiking neurons as well. Some deviations from this idea are noted and may take specific advantage of characteristics of computers which make them more efficient than biological brains. As an example, the UKS allocates new nodes and edges as needed; in a biological brain, most connections are allocated in early brain development. [Stiles 2010]

Further, combining neural and symbolic AI could prove one technique for creating AGI [Mao 2019]. The program provides valuable infrastructure to ease AGI experimentation.

5 Conclusions and Future Research

Several insights have already been gleaned from this system which will be the topics for additional publications. For example, a form of the UKS was initially created in neurons and it was observed that a UKS node requires at least seven neurons and many more if sequential information is stored. To the extent the human brain stores information in a UKS-like structure, this puts a limit on the amount of information a brain can contain.

Planned near-term development includes: Improved and expanded sensory inputs, expansion of language capabilities, and the ability for the entity to move objects in its environment. This will allow exploration into how the entity learns the basic physics of objects and uses this knowledge to plan object motions to achieve goals.

The software is available under the MIT license which allows virtually any use at no cost (including commercial). Available for download at: `http://brainsim.org`.

References

L. F. Abbott (1999), Lalique's introduction of the integrate-and-fire model neuron (1907), Brain Research Bulletin, Vol. 50, Nos. 5/6, pp. 303–304, 1999

Dutta, S., Kumar, V., Shukla, A. *et al.* (2017) Leaky Integrate and Fire Neuron by Charge-Discharge Dynamics in Floating-Body MOSFET. Scientific Reports **7,** 8257 (2017). `https://doi.org/10.1038/s41598-017-07418-y`

Hafting, T.; Fyhn, M.; Molden, S.; Moser, M. B.; Moser, E. I. (2005). "Microstructure of a spatial map in the entorhinal cortex". *Nature*. **436** (7052): 801–806. Laird, J (2012), The Soar Cognitive Architecture, MIT Press.

Mao, J., Gan, C. Kohli, P., Tennenbaum, J., Wu, J. (2019) The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences from Natural Supervision, ICLR 2019 conference paper.

Miller, M.S.P (2015). *Building Minds with Patterns*, Piaget Modeler, Beverly Hills, CA.

Simon, C. (2018), *Will Computers Revolt? Preparing for the Future of Artificial Intelligence,* FutureAI, Washington, DC.

Stiles, J. & Jernigan, T.L., The Basics of Brain Development, Neuropsychology Rev (2010) 20: 327. `https://doi.org/10.1007/s11065-010-9148-4`