**Large-scale modeling of the behaving brain**

Terrence C. Stewart and Chris Eliasmith

Abstract --- We review the methods used to construct Spaun, the first (and so far only) reasonably detailed brain model capable of performing cognitive tasks. Spaun has 2.5 million simple, spiking neurons with 60 billion connections (synapses) between them. These neurons are arranged to respect known anatomical and physiological contraints of the mammalian brain. The resulting model can perform eight different perceptual, motor, and cognitive tasks (see http://nengo.ca/build-a-brain/spaunvideos for video demonstrations). We built Spaun using a general tool set for building systems that compute particular functions using neuron-like components in a biologically constrained way.
Building such systems is critical for improving our understanding of how the brain works, and exploiting recent advances in neuromorphic (i.e. brain-like) hardware.

**Introduction**

We recently described what is currently the world's largest functional brain model, one that is capable of performing a variety of cognitive tasks (Eliasmith et al., 2012). This model, which we refer to as the Semantic Pointer Architecture Unified Network (or Spaun), consists of 2.5 million simulated spiking neurons whose properties and interconnections are consistent with those found in the human brain. The model receives input in the form of digital images on a virtual retina and produces output that controls a simulated arm. With this framework, Spaun is able to perform eight different tasks, including digit recognition, serial working memory, pattern completion, mental arithmetic, and question answering. Furthermore, it is able to switch between these tasks based on its own visual input, meaning that there are no external modifications made to the network between tasks (see http://nengo.ca/build-a-brain/spaunvideos for video demonstrations). This sort of cognitive flexibility is a hallmark of cognitive systems, but is difficult to achieve with traditional neural modeling approaches.

We achieved this using the Neural Engineering Framework (NEF), a mathematical theory that provides methods for systematically generating functional, biologically plausible spiking networks (Eliasmith & Anderson, 2003). We also employ the Semantic Pointer Architecture (SPA), a hypothesis regarding some aspects of the organization, function, and representational resources used in the mammalian brain (Eliasmith, 2013). We have made these methods highly accessible through the software tool Nengo (http://nengo.ca). The resulting NEF/SPA/Nengo combination has proved very effective for rapidly constructing biologically detailed neural models that can be related directly to specific behaviour.

The goal of this research is to understand how the brain works by reverse-engineering it. We do this by building biologically plausible models of cognitive processes. For us, these are models where the individual components (simulated neurons) can be made as similar to real neurons as desired, and where the large-scale anatomy and connectivity of the brain is respected. While the work described here uses the leaky integrate-and-fire (LIF) model of a neuron (the simplest and most common neuron model that produces spikes), all of the techniques apply to more detailed neural models. Different neurons in different brain areas have different biological and functional properties, and we use this neurological data to constrain our models. While we do not argue that this is the only way to build such models, we believe that the Neural Engineering Framework (NEF) is a general tool for implementing a very large set of algorithms in components like neurons, and that the Semantic Pointer Architecture (SPA) is one particular architecture that can be implemented with the NEF and that we believe is quite promising for matching human performance.
As a side effect of this reverse engineering goal, the Neural Engineering Framework provides a practical methodology for biomimetic computation.
In this paper we focus on the consequences and applications of this approach. For further information on the details of the methods, we point interested readers to recent descriptions of this work (e.g. Stewart & Eliasmith, 2014).

**The Neural Engineering Framework (NEF)**

The Neural Engineering Framework (NEF) is a general-purpose system for taking algorithms and implementing them using components such as spiking neurons. This can be thought of as a “neural compiler” where algorithms written in a high-level language are converted into neurons with connections between them. This compilation process works for arbitrary neuron types, and can be constrained in biologically realistic ways. Importantly, the high-level algorithms must be expressed in terms of vectors and functions on those vectors (including ordinary differential equations). The resulting neural networks approximate the desired functions, and the error of this approximation can be made arbitrarily small by increasing the number of neurons. This makes the NEF ideal for expressing algorithms typically seen in domains such as control theory, and determining their relevance to brain function.

While the NEF can be used to build arbitrary abstract systems such as controlled attractor networks, we have primarily used it to show how particular capabilities found in real animals might be implemented biologically. This has included path integration in rodents, working memory, and arm movements in monkeys, and decision-making in rats and humans. We have also taken into account biological constraints such as Dale's Principle (the principle that biological neurons are either inhibitory or excitatory, not both as often assumed by artificial neural networks) and incorporated biologically realistic learning rules to construct these networks.

The NEF is defined by the three main principles that address representation, computation, and dynamics in neurobiological systems. These three principles are sufficient to implement all of our neural models. To simplify the process of constructing these models, we have developed an open-source software package known as Nengo (Neural ENGineering Objects) that creates and runs these models. Models can be built in Nengo using a drag-and-drop graphical user interface or specified using the Python scripting language. Full details and documentation can be found online at http://nengo.ca.

**The Semantic Pointer Architecture (SPA)**

While the NEF specifies how to convert vector-based algorithms into spiking neural networks, a separate theory is needed to describe cognitive function in terms of vector-based algorithms. This is the purpose of the Semantic Pointer Architecture (SPA). The core of this approach a vector-based cognitive architecture: i.e., a set of basic functional components, each of which can be defined in terms of vector operations, and an organization of those components that can work together to implement cognitive algorithms. In addition to these components, we provide a hypothesis as to how structured representations (like sentences) can be represented using vectors and what basic operations need to be performed on those vectors to achieve memory, planning, pattern matching, and other behaviors.

We refer to our proposed form of neurally plausible representation as “semantic pointers.” We specify operations on semantic pointers that allow them to be useful for statistical processing (as often found in perceptual systems), structural processing (often found in more cognitive systems), and implementation of adaptive control algorithms (often found in motor systems). We couple this representation to an action selection system, in the form of a model of the basal ganglia, used to control the flow of information through the whole system. Together the components of the SPA have proven sufficient for building Spaun, a biologically plausible, unified model of the brain.

Spaun has been shown to match a wide variety of neural and behavioral data across many scales. Its components have been shown to reproduce spike patterns in the basal ganglia during a reinforcement learning task, single neuron tuning curves found in primary visual cortex, population spectrogram shifts during a working memory task, recognition accuracy on naturalistic stimuli (i.e., handwritten digits), and reaction times during a counting task, and similar results apply to the complete model as well. In short, because the model implements cognitive behaviors using spiking neurons that are anatomically and physiologically matched to their biological counterparts, its performance can be constrained by data from single cell physiology through to behavior. By making successful comparisons to diverse constraints, we begin to build the case that Spaun is capturing some important aspects of the biological mechanisms at work in real brains.

**Limitations and engineering applications**

While we believe that models like Spaun are moving us towards a better understanding of brain function, there remain many challenges ahead. It is important to keep in mind that Spaun has 40,000 times fewer neurons than the human brain. Consequently, it is still not clear how well the methods of the SPA will scale, despite encouraging initial results. Similarly, Spaun includes several simplifying assumptions regarding the number and kind of neurotransmitters, and physiological properties of individual neurons. Again, past work using the same methods has incorporated a wider variety of such properties than are found in Spaun, but it remains to be seen how additional biological detail will affect Spaun's functioning.

In addition to these limitations, it is clear that there are many kinds of brain function not well reflected in Spaun. For instance, the ability of the model to lay down new long term memories is minimal (this only occurs in one of the eight tasks). Similarly, there is little environmental interaction of the model: its single eye remains fixed, and it does not see its own output. Much work remains to be done to determine what additional functions are necessary to allow such a model to be embedded in a dynamic, open-ended environment. Relatedly, learning new cognitive behaviors in such an environment is a well-known challenge with few general, effective solutions. It has not yet been shown how these kinds of models can be used to effectively tackle such challenges.

From a computational perspective, simulating large-scale neural models on conventional computational hardware is difficult. For Spaun, it took approximately 2.5h of simulation time to generate one second of behavior on a high-end workstation. While we believe that this simulation can be made much more efficient (and in the latest version of Nengo we have made significant improvements), it is clear that alternate computing approaches would be advantageous.

A wide variety of these brain-inspired computing devices exist, all based around the idea of having a large number of simple neuron-like components whose spiking activity is based on the sum of their inputs. We have used the NEF as a general method for programming such neuromorphic computers. Examples of using the NEF in this way can be found on efficient digital architectures employing thousands or millions of ARM cores (the SpiNNaker project), analog architectures that directly incorporate forms of learning and hybrid architectures (the Neurogrid project). There are many benefits to this new computing paradigm, including orders of magnitude better power efficiency per computation, robustness to noise and variability, and massive parallelism.

Because the NEF was developed to address systems with these same properties (but in a biological setting), it has proven an effective means of programming such hardware, by indicating what the connection weights should be to achieve different computational results. For example, the NEF has been used to control a robot that can learn by treating training examples as the function to be approximated, to do operational space control on a 3-joint arm, and to implement a model of the rat hippocampus' path integration ability on a mobile robot. In all of these examples, the algorithms being implemented are well-suited to approximation using the NEF, and so are much more efficient when implemented on neuromorphic hardware than on traditional computing devices.

While current hardware implementations remain small scale compared to models like Spaun, we expect that in the near future, there will be a significant increase in the size of neuromorphic platforms available. Indeed, we expect that a full hardware implementation of Spaun will be completed within the next two years. This co-development of algorithms, programming techniques, and infrastructure in the neuromorphic space, provides fertile ground for designing and testing brain-like models. We believe that such models will allow us both to better understand biological brain function, and to develop a new class of solution to challenging information processing problems.

**Conclusion**

The theoretical methods and software suite we described form a comprehensive tool chain for connecting high-level behavior to low-level neural processes. The Neural Engineering Framework compiles algorithms expressed in terms of vectors and functions on those vectors (or their temporal derivatives) into a neural network that approximates those functions. The Semantic Pointer Architecture is a means of organizing neural models that is consistent with contemporary neuroscience. Nengo implements both the NEF and the SPA, allowing a user to specify the high-level function to be implemented (along with whatever neural constraints are appropriate). Nengo has also been shown to scale up well, as it was used to create Spaun, the world's largest functional brain simulation, with 2.5 million neurons and over 60 billion synapses.

Combined, these approaches provide a novel method for creating large-scale neural networks that can exhibit high-level cognitive behavior. Our ongoing research involves testing psychological theories by implementing them in neurons and comparing the model performance to human (and animal) performance. Importantly, we can do this comparison based not only on the overt behavior, but also on low-level neural measurements, such as firing patterns in different brain areas. This provides strong constraints on theories of brain function. For example, we find that overall reaction time is strongly connected with the neurotransmitter time constants. Furthermore, we believe accurate models of these neural functions could lead to improved understanding of how particular neural disorders (such as Parkinson's Disease or Alzheimer's Disease) produce their behavioural effects. More research, however, is needed to improve the neural details of these models such that it is possible to damage them in the same way that they are damaged in those diseases.

A more surprising consequence for this research is that it provides a novel method for programming highly parallel hardware. The human brain can be thought of as 100 billion interconnected processors (neurons). Each of these processors are slow, noisy, and can only compute one operation (the neural non-linearity), but by connecting them in different ways we can approximate a wide variety of functions. This is a new approach for neuromorphic engineering, and our ongoing collaborations are examining the possibilities for implementing complex cognitive algorithms efficiently.

Simulating the human brain is a monumental task and clearly beyond what a single research group can accomplish. This is why we are interested in helping to create general, open tools that can be applied to many different brain areas and many different kinds of cognitive tasks. We have developed tutorials and documentation to introduce the open-source Nengo software (available at http://nengo.ca). To aid instruction, we have included both a complete scripting system and an integrated drag-and-drop GUI interface for Nengo. We hope that being able to integrate ideas from psychology, neuroscience, and artificial intelligence and construct large-scale neural models that connect sensory systems, cognitive system, and motor systems makes for an exciting new approach to brain research. We believe these sorts of models will be extremely beneficial for understanding human cognition, treating brain disorders, and developing efficient parallel computation.

**Further Reading**

Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation and dynamics in neurobiological systems*. Cambridge, MA: MIT Press.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., & Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science*, *338*, 1202–1205. doi:10.1126/science.1225266

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. New York, NY: Oxford University Press.

Stewart, T. C., & Eliasmith, C. (2014). Large-Scale Synthesis of Functional Spiking Neural Circuits. *Proceedings of the IEEE*, *102*(5), 881–898. doi:10.1109/JPROC.2014.2306061



{Terrence C. Stewart} recieved a B.A.Sc degree in systems design engineering at the University of Waterloo in 1999, an M.Phil degree in computer science and artificial intelligence at Sussex University in 2000 and a Ph.D. degree in cognitive science from Carleton University in 2007.
He is a post-doctoral research associate in the Department of Systems Design Engineering with the Centre for Theoretical Neuroscience at the University of Waterloo, in Waterloo, Canada. His core interests are in understanding human cognition by building biologically realistic neural simulations, and he is currently focusing on language processing and motor control.
Dr. Stewart is a Behavioral & Brain Sciences Associate, a member of the Cognitive Science Society, and a founding member of the Biologically Inspired Cognitive Architecture society. He also co-chaired the 2013 International Conference on Cognitive Modelling, held in Ottawa, Canada.

{Chris Eliasmith} received a B.A.Sc. degree in systems design engineering in 1994 and a M.A. degree in philosophy in 1995 from the University of Waterloo. He received a Ph.D. degree in philosophy from Washington University in St. Louis in 2000.
From 2000-2001 he was a postdocoral researcher with Charlie Anderson in David van Essen's lab at Washington University Medical School. Since then he has been an assistant, associate, and full professor at the University of Waterloo. He is currently Director of the Centre for Theoretical Neuroscience at the University of Waterloo and holds a Canada Research Chair in Theoretical Neuroscience. He has authored or coauthored two books and over 90 publications in philosophy, psychology, neuroscience, computer science, and engineering venues.
Professor Eliasmith holds a P.Eng designation, is associate editor for the journal Cognitive Science, and a member of the Society for Neuroscience, the Canadian Philosophical Association, and the Society for Cognitive Science.