Cognitive Modelling with the Neural Engineering Framework

Chris Eliasmith (celiasmith@uwaterloo.ca)
Terrence C. Stewart (tcstewar@uwaterloo.ca)
Centre for Theoretical Neuroscience, University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1

Abstract

The Neural Engineering Framework (NEF; Eliasmith and Anderson, 2003) provides a general methodology for developing efficient and realistic neural models that perform a specified task. The framework consists of three quantified principles, one for each of representation, computation, and dynamics in neural systems. Adopting these principles provides a method for generating connection weights between groups of neurons that represent and transform state variables. In short, the NEF provides a neural compiler: a method for taking a high-level description of a neural system and deriving a plausible organization of realistic neurons that realize this system. Our tutorial introduces the principles of the NEF and demonstrates how they apply to cognitive modeling. This is done through the use of Nengo, a GUI neural simulation system, which supports an adjustable level of neural accuracy, Python scripting, and the analysis of the resulting models.

Keywords: Nengo; neural engineering; neural representation; control theory; neural cognitive modelling

The Neural Engineering Framework

As cognitive models become more complex, there is an increased demand for details at both the low and high levels. Traditionally, focus in cognitive modeling has been on higher levels of abstraction. As a result, researchers typically posit a high-level organizational structure which allows them to consider the information that needs to be represented and the transformations that are required for implementing hypothesized algorithms. Ideally, however, a cognitive model should also make detailed predictions as to the firing rates of neurons implementing the model, their tuning curves, connectivity, neurotransmitters, and other properties.

The Neural Engineering Framework (or NEF; Eliasmith and Anderson, 2003) provides a novel approach to addressing this typical gap in cognitive modeling. It is based on three principles of neural engineering:

1. Neural representations are defined by the combination of nonlinear encoding (exemplified by neuron tuning curves) and weighted linear decoding.
2. Transformations of neural representations are functions of variables that are represented by neural populations. Transformations are determined using an alternately weighted linear decoding.
3. Neural dynamics are characterized by considering neural representations as control theoretic state variables. Thus, the dynamics of neurobiological systems can be analyzed using control theory.

Each of these principles is considered under the assumption that neural systems are subject to significant amounts of noise. Therefore, any analysis of such systems must account for the effects of noise.

The core idea of the NEF is to consider any cognitive system as containing a large number of representations which change over time. How these representations change is dependent both on the external stimuli and on the other representations within the system. A particular neural group can represent (via its spike pattern) a single scalar, a vector, or even a function. These representations are inherently noisy, and accuracy will be dependent on various neural properties (although representational error has been shown to be inversely linearly related to the number of neurons used).

To understand how these representations change (i.e. define a transformation of a representation), the NEF provides methods for defining weighted axonal projections. For instance, a given group might represent the product of the values being represented by two other groups which are projected to it (i.e. $x(t) = y(t)*z(t)$, where each variable is represented by a neural population). Importantly, we can use the NEF to derive the linearly optimal connection weights to perform a wide variety of linear and nonlinear transformations. Doing so makes it clear that the accuracy of these transformations is intimately related to the observable tuning curves of the neurons involved. This leads to models that are orders of magnitude more efficient than other approaches to neural representation, and which are a closer match to observed neurological data (e.g. Conklin & Eliasmith, 2005; Fischer, 2005).

Applications

Initially, the main applications of this approach were in the domains of sensory and motor systems. This has included the barn owl auditory system (Fischer, 2005), rodent navigation (Conklin & Eliasmith, 2005), escape and swimming control in zebrafish (Kuo & Eliasmith, 2005), and the translational vestibular ocular reflex in monkeys (Eliasmith et al., 2002). However, these same principles are now being applied to higher-level cognitive models. A direct extension of the visual working memory model (Singh & Eliasmith, 2006) has led to a neural model of the ACT-R goal buffer (Stewart, Tripp, & Eliasmith, 2008). More crucially, the use of Vector Symbolic Architectures (Gayler, 2003) has allowed for the representation and manipulation of structured symbol trees by these neural models. This neurally realistic cognitive architecture (Stewart & Eliasmith, 2009a) resulted in a model of the Wason card task (Eliasmith, 2005) and ongoing work producing an efficient production system using realistic neural constraints (Stewart & Eliasmith, 2008; 2009b).

The NEF provides an exciting new tool for cognitive modelers as it provides a technique for producing direct neural predictions from a given high-level algorithmic description of a cognitive model. Furthermore, it leads to important theoretical results as to the relationships between neural properties and the high-level algorithms they are capable of implementing (e.g. the relationship between neurotransmitter re-uptake rate and the time constant of neural transformations).

These consequences are also very general, as the NEF provides techniques that can be applied to any cognitive model. It provides a structure for organizing the high-level description of a model, such that it can be implemented by realistic spiking neurons, providing meaningful data in terms of the expected spike patterns, time course, and accuracy. We have made use of it in a wide variety of contexts, and we have developed tools that support the creation and analysis of these models. These tools can be applied to many existing models to incorporate low-level neural details into existing modeling research.

Software and Simulations

We have developed Nengo <nengo.ca>, a freely available open-souce Java-based neural simulator that supports the NEF. This allows for hand-on examples of the theoretical concepts underlying the NEF. Using a point-and-click interface, we can create neural group, configure them to represent scalars and vectors, adjust their neural properties, and simulate their spiking activity over time. We can also connect these neural groups via synapses so as to perform linear and nonlinear transformations on these values, and store information over time. These are the basic mechanisms required for a wide range of algorithms, and form the basis for our models of sensorimotor systems and working memory. Nengo can also be programmed using a Python interface, allowing for quick creation of complex models (Stewart, Tripp, & Eliasmith, 2009).

Furthermore, these basic tools can be used to implement the theory of Vector Symbolic Architectures (Gayler, 2003) using NEF. This involves using high-dimensional fixed-length vectors to represent symbols and symbol trees. The nonlinear operation of circular convolution is used to manipulate these symbol trees. This can be seen as a non-classical symbol system, capable of performing the operations required for symbolic cognition. The result is a scalable and efficient neural cognitive architecture, constructed from these basic neural components.

References

Conklin, J. & Eliasmith, C. (2005). An attractor network model of path integration in the rat. *Journal of Computational Neuroscience 18*, 183-203.

Eliasmith, C. (2005). *Cognition with neurons: A large-scale, biologically realistic model of the Wason task.* 27th Annual Meeting of the Cognitive Science Society.

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

Eliasmith, C., Westover, M.B., & Anderson, C.H. (2002). A general framework for neurobiological modeling: An application to the vestibular system. *Neurocomputing 46*, 1071-1076.

Fischer, B. (2005). *A model of the computations leading to a representation of auditory space in the midbrain of the barn owl*. PhD thesis. Washington University in St. Louis.

Gayler, R. (2003). *Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience.* ICCS/ASCS International Conference on Cognitive Science, 133-138.

Kuo, D. & Eliasmith, C. (2005). Integrating behavioral and neural data in a model of zebrafish network interaction. *Biological Cybernetics*. *93*(3), 178-187.

Singh, R., & Eliasmith, C. (2006). Higher-dimensional neurons explain the tuning and dynamics of working memory cells. *Journal of Neuroscience*, *26*, 3667-3678.

Stewart, T.C. & Eliasmith, C. (2008). *Building production systems with realistic spiking neurons.* 30th Annual Meeting of the Cognitive Science Society.

Stewart, T. C. & Eliasmith, C. (2009a). *Compositionality and biologically plausible models*. In Oxford Handbook of Compositionality, W. Hinzen, E. Machery, M. Werning, eds. (Oxford University Press).

Stewart, T. C., & Eliasmith, C. (2009b). *Spiking neurons and central executive control: The origin of the 50-millisecond cognitive cycle*. 9th International Conference on Cognitive Modelling.

Stewart, T. C., Tripp, B., & Eliasmith, C. (2008). *Supplementing Neural Modelling with ACT-R*. 15th Annual ACT-R Workshop.

Stewart, T. C., B. Tripp, & Eliasmith, C. (2009). Python scripting in the Nengo simulator. *Frontiers in Neuroinformatics*. *3*(7), 1-9.