Exploring a Basis Set of Intrinsic Functions Underlying Neural Computation by Symbolically Programming Recurrent Neural Networks

Daniel Calbick (daniel.calbick@yale.edu)
Interdepartmental Neuroscience Program, Yale University

Jason Z. Kim (jk2557@cornell.edu)
Department of Physics, Cornell University

Ilker Yildirim (ilker.yildirim@yale.edu)
Department of Psychology, Yale University

Abstract

Much like recurring cell types and circuit motifs, a basis set of functional primitives implemented in stereotyped neural dynamics can, in principle, be used to accomplish complex computation in the brain. Existing work using multitask-trained deep neural networks has established the emergence of functional modules in otherwise homogeneous networks. However, these approaches only allow for indirect exploration of the space of functional primitives due to the entanglement of training objectives, datasets, and architectures. Here, we take a direct, hypothesis-driven approach: inspired by classic work in computational neuroscience, we explore the basic building blocks of digital circuit functions, including oscillators, lag operators, and set-reset latches, as the basis set of functional primitives in the brain. Our approach is enabled by a new way of utilizing neural network computation. Instead of training, we program the weights and connectivity of RNNs to compute symbolically-specified functions. Using a standard multi-task learning battery, we show that these "intrinsic function RNNs" can support cognitive flexibility.

Keywords: intrinsic functions; neurobiology; programmable RNNs; biologically-plausible RNNs; multitask learning

Introduction

Neuroscience explores cell types and local circuit motifs, revealing that similar patterns recur across the cortex (Harris & Shepherd, 2015). In much the same way, and likely by building on this underlying organization, it is plausible that a set of reusable functional primitives implement complex computation in the brain, including learning, memory, and other aspects of cognition. Indeed, stereotyped neural dynamics, such as sequential activity and oscillations, can support diverse functions from decision-making and navigation to the generation of rhythmic behaviors (Harvey, Coen, & Tank, 2012; Marder & Calabrese, 1996). How, in formal engineering terms, can we explore such a basis set of intrinsic functions in the brain?

One approach is based on task-optimized deep neural networks (DNNs). Recent work using multitask-trained DNNs has shown the emergence of functional modules in architectures built from otherwise uniform units (e.g., Yang, Joglekar, Song, Newsome, and Wang (2019)). Despite the insights it provides, this approach allows only for an indirect way of exploring what might be the space of functional primitives implemented in neurobiology. The functional modules in a task-optimized DNN fall from an entanglement of task objectives, datasets, and architectures through training. Moreover, resulting functional modules often need reverse-engineering for how they may contribute to computation (Yang et al., 2019; Orhan & Ma, 2019). Finally, the reusability of these functional modules is limited (Márton, Gagnon, Lajoie, & Rajan, 2021); there is no principled abstraction for reassembling these modules for de novo tasks.

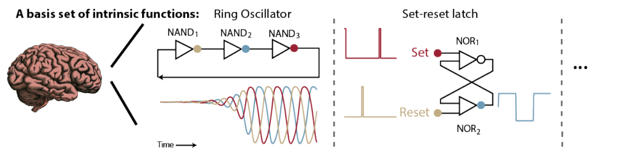

Figure 1: A basis set of intrinsic functions – based on building blocks in electronics – to support learning and memory.

Inspired by classic work in computational neuroscience, we explore basic building blocks from electronics as the basis set of intrinsic functions that give rise to learning, memory, and other aspects of cognition in neurobiology. Early work explored largely hand-designed circuit models (manually designing the weights and connectivity) to compute such functions as oscillators, phasors, and shifter circuits (Marder & Calabrese, 1996; Anderson & Van Essen, 1987; Touretzky, Redish, & Wan, 1993), in the service of understanding biological computation for rhythmic behavior generation, navigation, and invariant object recognition. However, as computational goals and architectures become more complicated and bigger, hand-designing networks become simply intractable. Indeed, backpropagation has been the workhorse of inducing useful features in artificial neural networks.

Here, we take a different approach: by building on recent work in physics of dynamical systems (Kim & Bassett, 2022), we symbolically program large (e.g., typical number of hidden units on the order of 1000) and biologically plausible recurrent neural networks (RNNs) to compute basic building blocks from electronics, such as oscillators, shifters, and set-reset latches (Fig. 1). This procedure does not involve any training data (no numerical input-output pairs), nor training (via backpropagation or similar); instead it requires a symbolic description of the function to be computed and thus enables direct, hypothesis-driven exploration of what might be the functional primitives of neurobiology. These symbolically programmed RNNs offer a new way to harness distributed computation, which we drive with sensory inputs to learn multiple tasks via simple linear regressions over their hidden states. We show that these "intrinsic function RNNs" differentially support cognitive flexibility, including pattern encoding, retention, and discrimination during multi-task learning.

A basis set of intrinsic functions

We hypothesize that complex computations in the brain can arise from a basis set of intrinsic functions and their compositions. This is not a novel hypothesis by any means – rather, it is a theme explored in several theories of neural computation (Marcus, Marblestone, & Dean, 2014). However, we make a concrete computational proposal for what this basis set should be and test it in non-trivial settings. Any such functional basis set should be *(i)* general purpose in the sense of being able to support a wide variety of tasks; *(ii)* drive-able by (sensory) inputs to support learning new tasks; and *(iii)* composable to yield even more complex computation and support learning of even more complex tasks.

We propose that these intrinsic functions in neurobiology can be viewed as the set of basic building blocks from elec-

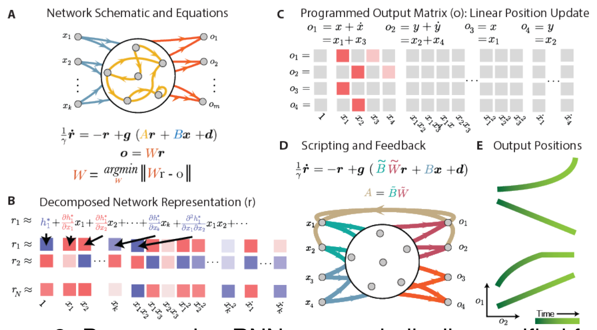

Figure 2: Programming RNNs on symbolically specified function, instead of training them on exemplars sampled from that function as is predominantly done. See text.

tronics – of which, in this work, we explore oscillators, shifters, and set-reset latches. The challenge here is to realize these building blocks in biologically plausible neural networks so that sensory inputs can drive and be composable across functions to support sophisticated computation and learning. Once this is accomplished, we can ask novel questions, such as whether an RNN oscillator circuit can support multitask learning and basic working memory functions.

Programming RNNs on symbolic functions To realize this hypothesis, we extend a recent method for programming, instead of training as is predominantly done, the weights and connectivity of RNNs to compute symbolically defined functions (Kim & Bassett, 2022). The class of RNNs we consider are shown in Fig. 2A with a network schematic and equations, including the network input $\boldsymbol{x} \in \mathbb{R}^k$, hidden state $\boldsymbol{r} \in \mathbb{R}^N$, and output $\boldsymbol{o} \in \mathbb{R}^m$, with weight matrices B (input weights), A (recurrence weights), and W (read-out weights) connecting these quantities. To program this RNN, we formalize its hidden state as a symbolic function in network inputs via linearization (Fig. 2B). We then program an output matrix o based on a symbolic specification of the function we wish the RNN to compute, which is illustrated for a linear dynamical system (Fig. 2C). The translation between the hidden state and this output matrix, the read-out weights W is simply solved for via a linear equation (Fig. 2A, bottom). Critically, these read-out weights can be connected back into the RNN to program the connectivity matrix of the RNN A and enable functional composability (Fig. 2D). In our linear system example, these feedback dynamics simulate trajectories (Fig. 2E).

Intrinsic function RNNs In this study, we explore three intrinsic function hypotheses: a shifter (i.e., a lag operator; Fig. 3A), a ring oscillator with three NAND gates sequenced via feedback (Fig. 1), and a set-reset latch with two NOR gates connected via feedback (Fig. 1). To study the computational efficiency of intrinsic functions, we also vary the size of hidden units (range: 30 to 1500 units). We consider as a baseline randomly connected RNNs.

Driving intrinsic function RNNs with sensory inputs and multitask learning Given a programmed RNN, we drive it using sensory and task inputs and test this RNN on multitask learning (Fig. 3B). Our overall architecture is shown in Fig. 3B. We expose three inputs to programmed RNNs. Two of these inputs encode sensory information, including a scalar fixation input and a scalar sensory feature. (In this work, the sensory feature is a random projection of a higher-dimensional sensory input indicating a direction signal.) The third input to the RNN encodes the task using one-hot coding, which sets the bias term, d (see inset equation, Fig. 3B). We store a bias term for each task by evolving the programmed RNN and registering its hidden state at every K time steps (K=1000). This procedure encodes the tasks using the expanse of the state-space afforded within the RNN dynamics.

For learning, we drive each RNN on 100 labeled example trials of each task – with each trial consisting of 65 time steps, during which fixation, sensory feature, and task are fed into the RNN. We then linearly regress the hidden states of the RNN onto labeled behavioral outputs. We test model performance on a held-out dataset. We emphasize that this simple learning strategy allows directly testing the ability of the proposed intrinsic functions to support multitask learning.

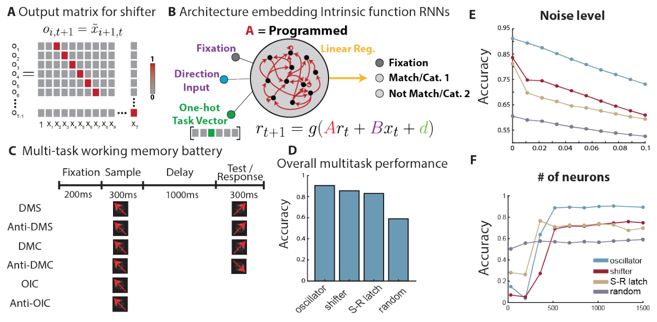

Figure 3: Intrinsic function RNNs differentially support multi-task learning. See text.

Results

Intrinsic function RNNs support multi-task learning We test intrinsic function RNNs on a standard multi-task learning battery with 6 tasks (Masse, Rosen, Tsao, & Freedman, 2022) (Fig. 3C); we follow Masse et al. (2022) for coding fixation and sensory inputs.For each functional hypothesis, we report the average accuracy across 3 RNNs. All three intrinsic function RNNs outperform a random network baseline (Fig. 3D). Importantly, these programmed RNNs show some robustness to noise – their performance gracefully decreases with increasing noise (Fig. 3E).

Intrinsic functions vary in computational efficiency We find that the computational efficiency – as a function of the number of hidden units – of the different intrinsic functions vary. The oscillator RNN was most efficient – typically yielding better performance for the same number of hidden units relative to other intrinsic functions.

Conclusion

We explored common building blocks in electronics and physics as a basis set of intrinsic functions to support computation in the brain. This direct hypothesis-driven exploration of intrinsic functions was enabled by a new way to harness the distributed computation in neural networks – programmable RNNs. Future work should expand the scope of intrinsic functions and tasks as well as make contact with empirical data.

Acknowledgments

We thank the Yale Center for Research Computing for their support in managing the Milgram computing cluster.

References

Anderson, C. H., & Van Essen, D. C. (1987). Shifter circuits: a computational strategy for dynamic aspects of visual processing. *Proceedings of the National Academy of Sciences*, *84*(17), 6297–6301.

Harris, K. D., & Shepherd, G. M. (2015). The neocortical circuit: themes and variations. *Nature neuroscience*, *18*(2), 170–181.

Harvey, C. D., Coen, P., & Tank, D. W. (2012). Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature*, *484*(7392), 62–68.

Kim, J. Z., & Bassett, D. S. (2022). A neural programming language for the reservoir computer. *arXiv preprint arXiv:2203.05032*.

Marcus, G., Marblestone, A., & Dean, T. (2014). The atoms of neural computation. *Science*, *346*(6209), 551–552.

Marder, E., & Calabrese, R. L. (1996). Principles of rhythmic motor pattern generation. *Physiological reviews*, *76*(3), 687–717.

Márton, C. D., Gagnon, L., Lajoie, G., & Rajan, K. (2021). Efficient and robust multi-task learning in the brain with modular latent primitives. *arXiv preprint arXiv:2105.14108*.

Masse, N. Y., Rosen, M. C., Tsao, D. Y., & Freedman, D. J. (2022). Flexible cognition in context-modulated reservoir networks. *bioRxiv*, 2022–05.

Orhan, A. E., & Ma, W. J. (2019). A diverse range of factors affect the nature of neural representations underlying short-term memory. *Nature neuroscience*, *22*(2), 275–283.

Touretzky, D. S., Redish, A. D., & Wan, H. S. (1993). Neural representation of space using sinusoidal arrays. *Neural Computation*, *5*(6), 869–884.

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., & Wang, X.-J. (2019). Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, *22*(2), 297–306.