Constructing neural-level models of behavior in working memory tasks

Zoran Tiganj (zorant@bu.edu)
Nathanael Cruzado (nac0005@bu.edu)
Marc W. Howard (marc777@bu.edu)
Center for Memory and Brain
Boston University

Abstract

Constrained by results from classic behavioral experiments we provide a neural-level cognitive architecture for navigating memory and decision making space as a cognitive map. We propose a canonical microcircuit that can be used as a building block for working memory, decision making and cognitive control. The controller controls gates to route the flow of information between the working memory and the evidence accumulator and sets parameters of the circuits. We show that this type of cognitive architecture can account for results in behavioral experiments such as judgment of recency and delayed-match-to-sample. In addition, the neural dynamics generated by the cognitive architecture provides a good match with neurophysiological data from rodents and monkeys.

Keywords: Cognitive architecture; Working memory

Introduction

Behavioral experiments provide important insights into human memory and decision making. Building neural systems that can describe these processes is essential for our understanding of cognition and building artificial general intelligence (AGI).

Here we propose a neural-level architecture that can model behavior in different cognitive tasks. The proposed architecture is composed of biologically plausible artificial neurons characterized with instantaneous firing rate and with the ability to: 1) gate information from one set of neurons to the other (Hasselmo & Stern, 2018) and 2) modulate the firing rate of other neurons via gain modulation (Salinias & Sejnowski, 2001). The architecture is based on a canonical microcircuit that represents continuous variables via supported dimensions (Shankar & Howard, 2012; Howard et al., 2014). The microcircuit is implemented as a two-layer neural network. The same microcircuit prototype is used for maintaining a compressed memory timeline, evidence accumulation and for controlling the flow of actions in a behavioral task.

A neural architecture for cognitive modeling

We sketch a neural cognitive architecture and apply it to two distinct working memory tasks. The architecture is composed of multiple instances of a canonical microcircuit. This microcircuit represents vector-valued functions over variables. These functions can be examined via attentional gates and then used to produce a vector-valued output. We first discuss the properties of the microcircuit.

Function representation in the Laplace domain

The microcircuit (Figure 1A) takes a set of inputs (top) and produces a set of outputs (bottom) with the same dimensionality. Let us refer to the input at time t as $\mathbf{f}(t)$. The heart of the canonical microcircuit is a set of units that represent vector-valued functions in the Laplace domain.

The first layer approximates the Laplace transform of the input $f(t)$ via a set of neurons which can be described as leaky integrators $F(s)$, with a spectrum of rate constants s. Each neuron in $F(s)$ receives the input and has a unique rate constant:

$$\frac{dF(s)}{dt} = \alpha(t)\left[-sF(s) + \mathbf{f}(t)\right], \tag{1}$$

where $\alpha(t)$ is an external signal that modulates the dynamics of the leaky integrators. If $\alpha(t)$ is constant, $F(s)$ codes the Laplace transform of $\mathbf{f}(t)$ leading up to the present. It can be shown that if $\alpha(t) = dx/dt$, $F(s)$ is the Laplace transform with respect to x (Howard et al., 2014). We assume that the probability of observing a neuron with rate constant s goes down like $1/s$. This implements a logarithmic compression of the function representation.

The second layer $\tilde{f}(\overset{*}{\tau})$ computes the inverse of the Laplace transform using the Post approximation. It is implemented as a linear combination of nodes in $F(s)$: $\tilde{f}(\overset{*}{\tau}) = \mathbf{L}_k^{-1} F(s)$. The operator $\mathbf{L}_k^{-1}$ approximates kth derivatives with respect to s. Because $\mathbf{L}_k^{-1}$ approximates the inverse Laplace transform, $\tilde{f}(\overset{*}{x})$ provides an approximation of the transformed function.

Accessing the function

The representation described above stores working memory as a vector-valued approximation of a function over an internal variable. We assume that this entire function can not be accessed all at once, but that one can compute vector-valued integrals over the function. The microcircuit includes a gating function $G(\overset{*}{x})$ that is externally controllable. The output of the microcircuit is:

$$\mathbf{O} = \sum_{\overset{*}{x}=1}^{N} G(\overset{*}{x})\tilde{\mathbf{f}}(\overset{*}{x}), \tag{2}$$

where N is the number of values of $\overset{*}{x}$ used to implement the function approximation. Note that this output depends on the state of the function representation, $\tilde{\mathbf{f}}(\overset{*}{x})$ and the current state of the gates $G(\overset{*}{x})$. We restrict $G(\overset{*}{\tau})$ to be unimodal. Gates can be set narrowly and then activated sequentially, allowing a scan of the function representation or many gates can be set broadly to sum across the $\overset{*}{x}$. This enables one to construct

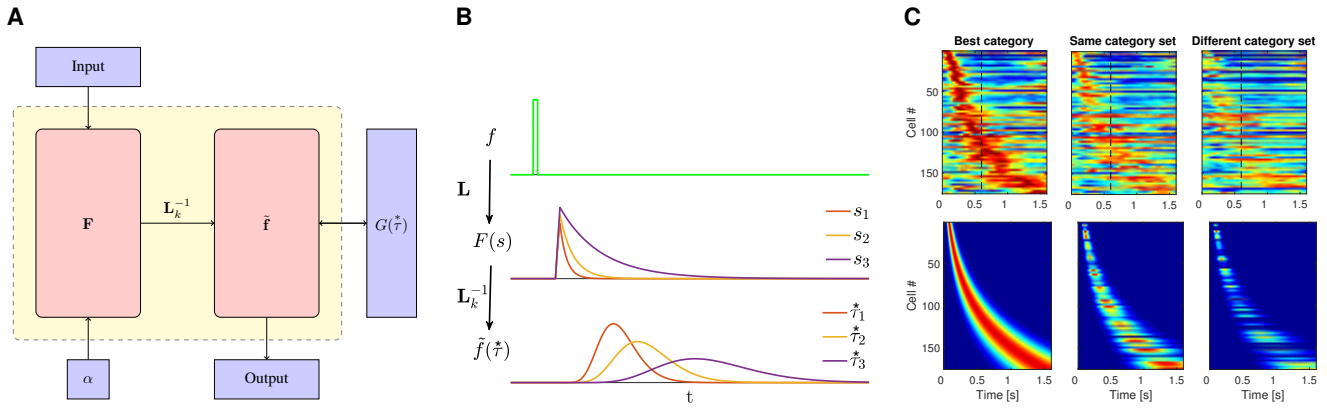

Figure 1: **Constructing a scale-invariant compressed memory representation through an integral transform and its inverse. A.** Microcircuit for representing variables via supported dimensions by implementing the Laplace and the inverse Laplace transform. **B.** A response of the network to a delta-function input. Activity of only three nodes in each layer is shown. **C** Top: During DMS task sequentially activated cells in monkey lateral prefrontal cortex encode time (via sequential activation) conjunctively with stimulus identity (firing rate encodes visual similarity of the stimuli - stimuli in "Best category" were visually more similar to stimuli in the "Same category set" than to stimuli in the "Different category set"). The three heatmaps show neural activity during the stimulus presentation (first 0.6 s) and the delay period (following 1 s) averaged across trials. (Taken from Tiganj et al. (in press)). Bottom: The model captures qualitative properties of the neural data.

cognitive models based on scanning (e.g., Hacker, 1980) or to construct global matching models (e.g. Donkin and Nosofsky (2012)).

Working memory: Functions of time

When $\alpha(t)$ is a constant, $\tilde{\mathbf{f}}$ maintains an estimate of $\mathbf{f}(t)$ as a function of time leading up to the present and we write $\tilde{\mathbf{f}}(\tilde{\tau}^*)$. If the input stimulus was a delta function at one point in the past, the units in $\tilde{\mathbf{f}}(\tilde{\tau}^*)$ activate sequentially with temporal tuning curves that are broader and less dense as the stimulus becomes more temporally remote (Figure 1B). Neurons with such properties, called time cells, have been observed in mammalian hippocampus (MacDonald, Lepage, Eden, & Eichenbaum, 2011) and prefrontal cortex (Tiganj, Kim, Jung, & Howard, 2017). Furthermore, different stimuli trigger different sequences of cells (Tiganj et al., in press), Figure 1C. Taken together $\tilde{\mathbf{f}}(\tilde{\tau}^*)$ can be understood as a compressed memory timeline of the past; the application of the Laplace transform in maintaining working memory in neural and cognitive modeling has been extensively studied (e.g., Shankar & Howard, 2012; Howard et al., 2014). The logarithmic compression in s values leads to logarithmic increases in the time necessary to sequentially access memory and a power law decrease in the strength of the match as stimuli recede into the past.

Evidence accumulation: Functions of net evidence

In simple evidence accumulation models, the decision variable is the sum of instantaneous evidence available during the decision-making process. In these models, a decision is executed when the decision variable reaches a threshold. By setting $\alpha(t)$ to the amount of instantaneous evidence for one alternative, we can construct the Laplace transform of the net amount of decision variable since an initialization signal was sent via the input $\mathbf{f}$. Inverting the transform results in a set of cells with receptive fields along a "decision axis" consistent with recent findings from mouse recordings (Morcos & Harvey, 2016). If no new evidence has been observed at a particular moment then $\frac{dF(s)}{dt} = 0$, thus all the units remain active with sustained firing rate. Large amount of evidence will, on the other hand, mean a fast rate of decay.

Cognitive control: Functions of planned actions

The program flow control activates sequence of actions necessary for completion of a behavioral task. For instance, a typical behavioral task may consist of actions such as attending to stimuli, detecting the probe, accumulating evidence and taking an appropriate action depending on which of the available choices accumulated more evidence. These operations require the ability to route information to and from the working memory and evidence accumulation modules. For instance, in order to compare a probe to the contents of memory, one might route the output of the working memory unit, filtered by a probe stimulus, to the $\alpha(t)$ of an evidence accumulation unit. Because various operations take place in series, we can understand them as a function of future planned actions. Rather than past stimuli, the vectors in $\mathbf{F}(s)$ and $\tilde{\mathbf{f}}(\tilde{\tau}^*)$ can be understood as operations that affect other units.

In practice, different cognitive models correspond to different initial states in $F(s)$ and $\tilde{f}(\tilde{x}^*)$. The actions will be executed sequentially by setting $\alpha(t) < 0$, winding the planned future closer and closer to the present. For instance, if the first step of a behavioral task is to wait for a probe, then that action (*wait for a probe*) will set the controller's $\alpha(t)$ to 0 until the probe is detected. Once the probe is detected, $\alpha(t)$ will be set to a default value of -1 so the neurons in the first layer will grow exponentially and the sequence loaded in $\tilde{f}(\tilde{\tau}^*)$ will continue evolving.

Integrating microcircuits into cognitive models

The three blocks described above: working memory, evidence accumulation and cognitive control are all constructed from the same microcircuit (Figure 1A). Each circuit has an input (which is unused for the evidence accumulation and the program control blocks), α and output. To demonstrate the utility of this approach, we connected the three blocks such that that program control block gates information from the working memory to the evidence accumulation block and monitors its output (Figure 2A).

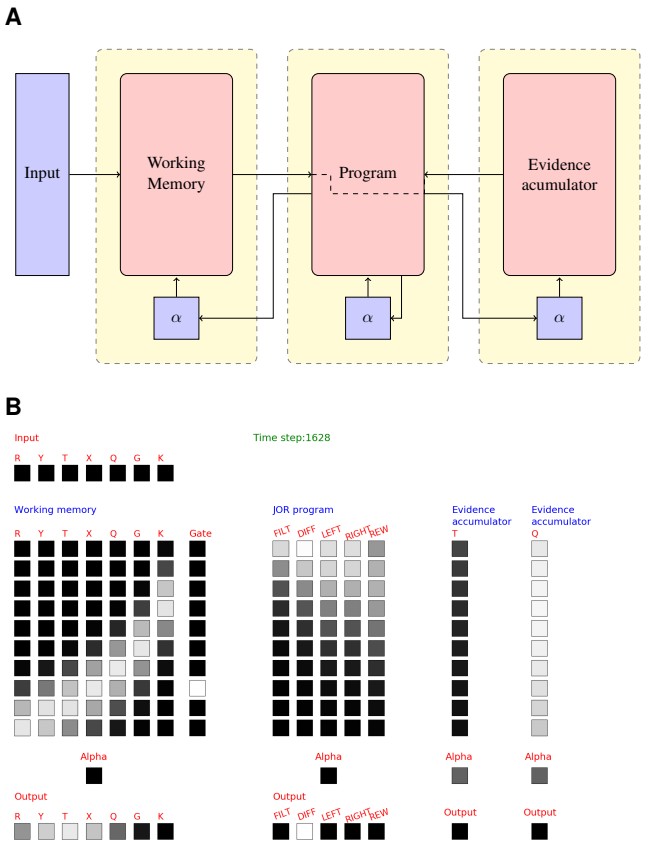

Figure 2: **A schematic of a neural-level circuit that can be used to model different behavioral tasks. A** A configuration of the circuit composed of three blocks: working memory, program and evidence accumulator, each implemented with the microcircuit shown in Figure 1**A**. Program block sequentially executes actions which include waiting for the probe, gating information from working memory to the evidence accumulator and reading the output of the evidence accumulator. **B** Example of a JOR experiment implemented with the proposed architecture. The implementation is done with microcircuits that correspond to those in Figure 2A. Each square corresponds to a single neuron. Shading reflects the activity of the neuron at a given time step; darker shading means less activity. At the time step shown on the plot, the sample and probe stimuli (T and Q) were already presented (they are stored in the working memory). Program control block sequentially gates the information from the working memory into the α neuron of the evidence accumulator (DIFF action in the program block), causing sequential activation in the accumulator. After the evidence accumulator reaches the threshold, program control continues execution by activating an appropriate action.

Results

We demonstrate performance of the proposed architecture on two classical behavioral tasks: Judgment of Recency (JOR) and Delayed-Match-to-Sample (DMS). We compare the results of the model with behavioral data (for JOR) and neural data (for DMS). Critically, even though these two tasks have very different demands, the neural hardware for the models is identical. The only difference is in the initial state of the program block. After initialization, each model runs autonomously and is self-contained.

In JOR subjects are presented with a random list of stimuli (e.g. letters or words) one at a time, and then probed with two stimuli from the list and asked which of the two stimuli was presented more recently. The classical finding is that the time it takes subjects to respond (reaction time) depends on the recency of the more recent probe, but not the recency of the less recent probe (Figure 3A) (Hacker, 1980; Singh & Howard, 2017). This result provides an important insight into how working memory is maintained, suggesting that the subjects maintain working memory as a a temporally organized, scannable representation. In other words, the result of the JOR experiment is consistent with a self-terminating backward scan along a temporally organized memory representation. Moreover, the reaction time is a sublinear function of the lag (Figure 3B) (Singh & Howard, 2017), suggesting that the working memory representation is compressed.

To model the JOR task, the first step of the model was to wait for the probe item to appear. After that, the gates were set to scan the memory representation sequentially from more recent to more distant past (Figure 2B). At each step, the value found in the memory was used to drive two evidence accumulators, one accumulator for each probe item. Once one of the two evidence accumulators reached a preset threshold, the program would continue executing and take an appropriate action (left or right choice). Variability in the reaction times was obtained by adding additive Gaussian noise to the evidence accumulation process. Results in Figure 3C indicate that the model captures well the aspect of the real data (Figure 3A) that suggests sequential scanning. In addition, the model is consistent with the data regarding compression of the memory representation (Figure 3B - data, Figure 3D - model).

In DMS subjects are presented with a sample stimulus followed by a delay interval, followed by a test stimulus. The action that subjects need to take (e.g. pressing a left or right button) depends on whether the two stimuli were the same or different. We modeled the task with the same components as for JOR task. The only differences were in 1) how the probe item was set (in DMS the second stimulus is by construction the probe, while in JOR the probe is marked by presenting two stimuli at the same time) and 2) what parts of the working memory were gated to the evidence accumulator (in DMS one accumulator accumulated evidence for presence of the probe item in the memory and the other accumulator accumulated evidence that any other item was found in the memory, while in JOR each of the two probe items had its own evi-

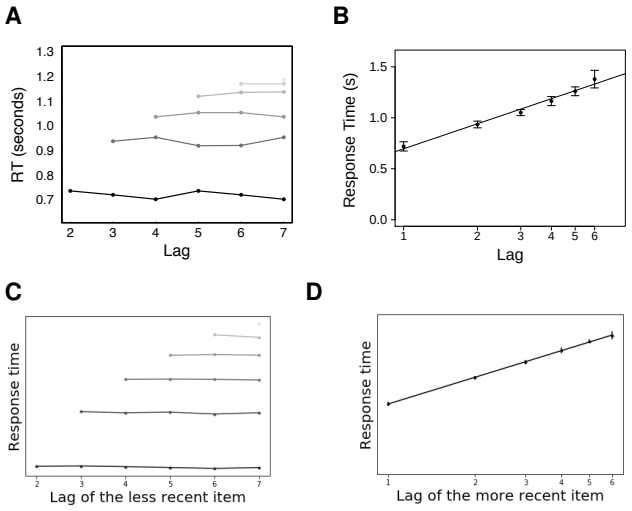

Figure 3: **The model captures behavioral results in the JOR task. A.** In the JOR task, median RT for correct responses depends strongly on the recency of the more recent probe but not the recency of the less recent probe. Shade of the line denotes lag of the more recent item, with the most recent item shown in black and the most distant item shown in the lightest shade of gray. (From Singh and Howard (2017).) **B.** In the JOR task, median RT varies sub-linearly with recency (x-axis is log-spaced). **C.,D.** Results of the model corresponding to **A.** and **B.** respectively.

dence accumulator). While simple in terms of behavior, DMS task is often done on animals while recording activity of individual neurons. Neural recordings during the delay period of this task show evidence for existence of stimulus-selective sequentially activated cells (Tiganj et al., in press) that correspond well to the neural activity produced by the sequential memory used here (Figure 1C).

Conclusions

Building neural models of behavioral tasks is an important step towards developing AGI. Here we provided an architecture that is based on realistic neural data and that can account for non-trivial behavior. In particular, the behavioral results of JOR task are consistent with the hypothesis that the subjects are scanning along a compressed timeline. The same architecture was used to model DMS task, resulting in neural representation of working memory that closely corresponds to the neural data. Critically, both of these tasks use the same neural hardware, differing only in the initial condition of the controller. This work is complementary with ongoing efforts of building cognitive architectures such as ACT-R (Anderson, Matessa, & Lebiere, 1997) and SOAR (Laird, 2012). The distinction of the present work is in its attempt to build such architecture with neuron-like units, similar to Eliasmith et al. (2012), but with a different type of neural representation. The present work commits to a specific type of representation: variables are represented as supported dimensions via neural tuning curves, tuned to a particular amount of elapsed time, accumulated evidence or a position in a sequence.

Acknowledgments The authors gratefully acknowledge support from ONR MURI N00014-16-1-2832, NIBIB R01EB022864 and NIMH R01MH112169.

References

Anderson, J. R., Matessa, M., & Lebiere, C. (1997). Act-r: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, *12*(4), 439–462.

Donkin, C., & Nosofsky, R. M. (2012). A power-law model of psychological memory strength in short- and long-term recognition. *Psychological Science*. doi: 10.1177/0956797611430961

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*(6111), 1202–1205.

Hacker, M. J. (1980). Speed and accuracy of recency judgments for events in short-term memory. *Journal of Experimental Psychology: Human Learning and Memory*, *15*, 846-858.

Hasselmo, M. E., & Stern, C. E. (2018). A network model of behavioural performance in a rule learning task. *Phil. Trans. R. Soc. B*, *373*(1744), 20170275.

Howard, M. W., MacDonald, C. J., Tiganj, Z., Shankar, K. H., Du, Q., Hasselmo, M. E., & Eichenbaum, H. (2014). A unified mathematical framework for coding time, space, and sequences in the hippocampal region. *Journal of Neuroscience*, *34*(13), 4692-707.

Laird, J. E. (2012). *The Soar cognitive architecture*. MIT press.

MacDonald, C. J., Lepage, K. Q., Eden, U. T., & Eichenbaum, H. (2011). Hippocampal "time cells" bridge the gap in memory for discontiguous events. *Neuron*, *71*(4), 737-749.

Morcos, A. S., & Harvey, C. D. (2016). History-dependent variability in population dynamics during evidence accumulation in cortex. *Nature Neuroscience*, *19*(12), 1672–1681.

Salinias, E., & Sejnowski, T. (2001). Gain modulation in the central nervous system: Where behavior, neurophysiology, and computation meet. *Neuroscientist*, *7*, 430–440.

Shankar, K. H., & Howard, M. W. (2012). A scale-invariant internal representation of time. *Neural Computation*, *24*(1), 134-193.

Singh, I., & Howard, M. W. (2017). Recency order judgments in short term memory: Replication and extension of hacker (1980). *bioRxiv*, 144733.

Tiganj, Z., Cromer, J. A., Roy, J. E., Miller, E. K., & Howard, M. W. (in press). Compressed timeline of recent experience in monkey lPFC. *Journal of Cognitive Neuroscience*.

Tiganj, Z., Kim, J., Jung, M. W., & Howard, M. W. (2017). Sequential firing codes for time in rodent mPFC. *Cerebral Cortex*, *27*, 5663-5671.