A Spiking Neural Architecture that Learns New Tasks

Niels Taatgen (n.a.taatgen@rug.nl)
Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence
Cognigron Center for Cognitive Systems and Materials
University of Groningen, Nijenborgh 9, 9747 AG Groningen, Netherlands

Abstract

Cognitive architectures based on neural networks typically use the Basal Ganglia to model sequential behavior. A challenge for such models is to explain how the Basal Ganglia can learn to do new tasks relatively quickly. Here we present a model in which task-specific procedural knowledge is stored in a separate memory, and is executed by general procedures in the Basal Ganglia. In other words, learning happens elsewhere. The implementation discussed here is implemented in the Nengo cognitive architecture, but based on the principles of the PRIMs architecture. As a demonstration we model data from a mind-wandering experiment.

Keywords: Spiking neural networks; Mind Wandering; Basal Ganglia; PRIMS; Nengo; Skill Acquisition

Model code: https://github.com/ntaatgen/NengoPRIMs

Introduction

Symbolic cognitive architectures are very powerful in producing flexible task performance. Part of task performance is the ability to carry out steps in sequence. Although a production system, the typical symbolic solution to sequential behavior, is a straight-forward solution, it is less clear how it is carried out by the brain. The brain structures that are typically implicated in sequential behavior are the Basal Ganglia and the Thalamus[1]. For example, numerous ACT-R studies map model activity onto brain areas, of which procedural memory is mapped onto the Basal Ganglia (Anderson et al., 2004). Several neural network architectures that include sequential behavior have forwarded proposals for possible Basal Ganglia implementations (Stocco, Lebiere, & Anderson, 2010; O'Reilly & Frank, 2006; Eliasmith et al., 2012). However, these implementations impose quite some constraints on production rules. In the Stocco et al. implementation, the amount of information that can be transferred between modules is limited to a single item of information. The Eliasmith et al. solution does allow for the transfer of multiple items, but has not clear way in which the procedural knowledge is learned. In addition, one may wonder whether all human procedural knowledge, which is often quite task-specific, can be stored in a structure as small as the Basal Ganglia.

The work presented here is not a completely new proposal for sequential behavior, but builds on the Eliasmith et al. (2012) solution in Nengo, ACT-R (Anderson, 2007) and the PRIMs theory (Taatgen, 2013). A common idea among these theories is that procedural knowledge involves controlling the flow of information between different cognitive modules. For example, in order to perform an Aural-Vocal task in which a number has to be spoken based on the pitch of a tone (i.e., when you hear a low tone you have to say "One", when you hear a middle tone you have to say "Two", etc.), an Aural module determines the pitch, a Declarative memory module determines the mapping from pitch to number, and a Vocal module speaks the number. The role of procedural knowledge is to take the result of the Aural module and feed this into the Declarative module, and once the Declarative module successfully produces a result, move that result to the Vocal module.

If we assume that the knowledge to carry out a procedural task such as the aural-vocal task is encoded in the Basal Ganglia, we have a problem. Tasks such as the aural-vocal task, and also more complicated tasks that are typically part of psychological experiments, can typically be carried out by subjects after a short instruction and very little practice, even though they have never done these tasks before. It is therefore not very likely that they train their Basal Ganglia in that short period for this specific purpose. We therefore have to look for a solution that uses existing representations in the Basal Ganglia to do new tasks. To develop such a solution, it is useful to look at the PRIMs architecture (Taatgen, 2013). In PRIMs, procedural knowledge is decomposed into a fixed set of primitive operations. Each of these operations either makes a single comparison, or performs a single action by transferring one knowledge element from one module to another. Because the set of PRIMs is finite, we can imagine a Basal Ganglia model that is capable of carrying out any of the PRIMs, and is therefore in principle capable of performing any sequential task that can be defined in terms of PRIMs.

In this paper, I will first describe the overall architecture of the Nengo/PRIMs model. It resembles the Spaun model, a Nengo model that is capable of carrying out a range of tasks Eliasmith et al. (2012). The main difference between the two is that Spaun's procedural knowledge is hardcoded in the Basal Ganglia, whereas the Nengo/PRIMs model only encodes PRIMs in the Basal Ganglia, and uses a memory system to trigger the correct PRIM at the right moment. I will then use it to model an experiment by Smallwood et al. (2011).

[1] To save space and improve readability, I will refer to the Basal Ganglia/Thalamus combination as just the Basal Ganglia for the rest of the paper.

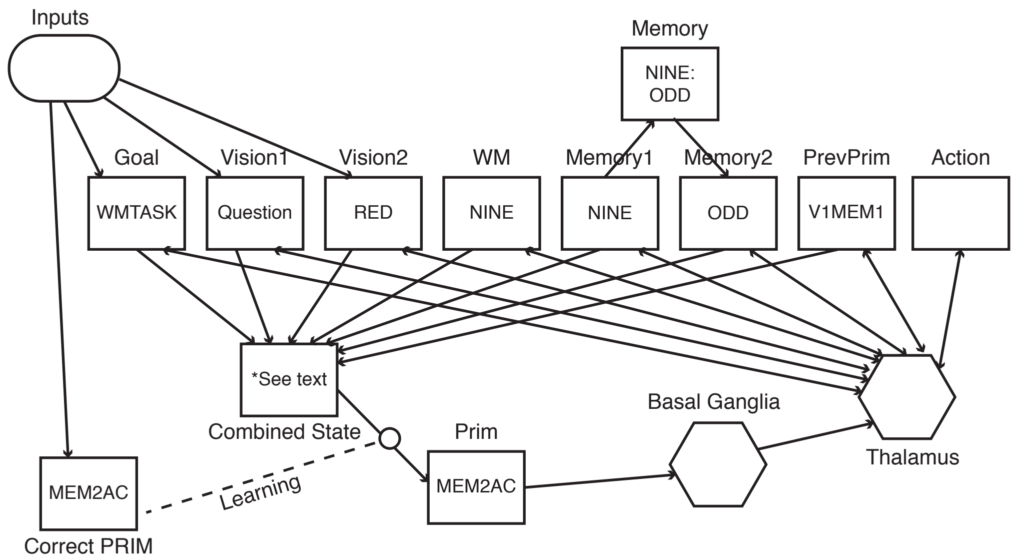

Figure 1: Overview of the Nengo/PRIMs model. Rectangles represent slots that can hold a single semantic pointer. Hexagons are more complex neural structures. The rounded rectangle provides the inputs to the network at scheduled times.

Overview of the System

Nengo basics: Semantic Pointers

Nengo is a neural network architecture based on spiking neurons. Clusters of neurons are used to represent vectors of numbers, and mappings between these clusters can calculate functions. For example, we can define a cluster of 100 spiking neurons to represent the vector $\begin{pmatrix} x \\ y \end{pmatrix}$, and connect this to another cluster of spiking neurons that will calculate and represent $\begin{pmatrix} x^2 \\ y^2 \end{pmatrix}$.

The next step is to let these vectors represent symbols. For example, a particular vector of numbers can represent the color RED (we use 128 dimensional vectors in the model here). A symbol, represented by a vector of numbers, is called a *semantic pointer* in Nengo. Semantic pointers can represent simple symbols, but can also be convolved to create more complex representations. For example, we can represent a red ball by the following vector:

$$\mathrm{REDBALL} = \mathrm{COLOR} \odot \mathrm{RED} + \mathrm{SHAPE} \odot \mathrm{ROUND}$$

Structure of the Model

With semantic pointers Nengo is capable of representing quite powerful knowledge structures, which can be manipulated with the appropriate mappings between clusters of neurons. The structure we will use is depicted in Figure 1. Each of the rectangles in the Figure represents a cluster of neurons that holds a single semantic pointer (we will call them "slots" in this paper). The horizontal row of rectangles represents a set of slots that hold information related to particular cognitive modules, similar to buffer slots in ACT-R. For illustration purposes, some values have been put into the boxes. They are related to the experimental task to be discussed later. The *Goal* slot represents the current task. It, together with the visual input, is set by a separate process represented by the rounded rectangle. This process sets the values in these slots to particular values at particular times in the task. In the example, the goal is set to WMTASK, and the visual input is set to a red question mark.

The *WM* (working memory) slot can hold a single item of information. Contrary to the other buffer slots, which information decays away if not fed by another process, the WM slot maintains its value until replaced. The three *Memory* slots represent a limited long-term declarative memory. An item can be placed in Memory1, after which an associative memory (Memory) finds the associate memory that is then placed in Memory2. In the example in the Figure, memory is used to determine that NINE is ODD. The *Action* slot is used to set the model's action. In the Figure it is not connected to anything, but it should be connected to an appropriate motor system, comparable to what has been done in Spaun (Eliasmith et al., 2012). Finally, *PrevPrim* refers to the previous step the system has executed, because this will be part of the input for determining the next step.

The model takes cognitive steps by transferring information between the slots. These steps are represented by cognitive operations that are basically quite simple: a symbol (semantic pointer) that represents the source and destination slots. For example, V1MEM1 means: copy the contents of Vision1 to Memory1. MEM2AC means: copy the contents of Memory2 to Action. The desired action is placed in the *Prim* slot, after which the Basal Ganglia carries out that action. The Basal Ganglia follows the standard Nengo implementation,

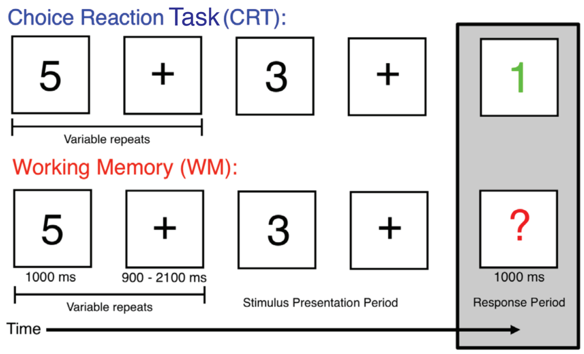

Figure 2: Design of the Smallwood et al. (2011) experiment.

Table 1: Timing of the Inputs node. The V1 and V2 columns are fed into Vision1 and Vision2, respectively, and the CRT prim or WM prim is placed in the ”Correct Prim” slot when learning is switched on.

t_{start}	t_{end}	V1	V2	CRT prim	WM prim
0.0	0.3	Digit	Black	FOCUS	V1WM
0.3	0.5	Blank	Blank	FOCUS	FOCUS
0.5	0.8	Digit	Black	FOCUS	V1WM
0.8	1.0	Blank	Blank	FOCUS	FOCUS
1.0	1.3	Digit or question	Red	V1MEM1	WMMEM1
1.3	1.6	Digit or question	Red	MEM2AC	MEM2AC
1.6	1.7	Blank	Blank	FOCUS	FOCUS

and has a rule for each of the possible PRIMs.

Although the PRIMs architecture also has primitive operations to test conditions, the Nengo/PRIMs model will achieve this in a different way. The role of conditions is to determine, given the state of the system, what actions need to be carried out. Here we achieve this goal in a slightly different way: by learning a mapping between the contents of all the slots and the Prim slot. We do this by combining all slots in a single semantic pointer:

$$\text{Combined} = \text{G} \odot \text{WMTASK} + \text{V1} \odot \text{Question} + \text{V2} \odot \text{RED} + \ldots$$

This combined semantic pointer is then mapped onto a PRIM semantic pointer.

Learning

The advantage of changing conditions into a more abstract mapping is that they can be learned instead of programmed. The current model uses supervised learning, which why there is a *Correct PRIM* slot that is set by the input process. Whenever the model produces a PRIM on the basis of the combined state (initially random), that PRIM is compared to the Correct PRIM, after which the weights that map the combined state onto the Prim slot are adjusted based on the error using prescribed error sensitivity (PES) learning (MacNeil & Eliasmith, 2011).

A Model of Mind Wandering

As an illustration of the model explained above, I will present a model of a task by Smallwood et al. (2011). In the experiment, subjects had to do two different tasks. In the Choice Reaction Task (CRT), subjects were presented with a sequence of digits that were interleaved with fixation crosses. Digits were presented for 1000ms, and the fixation cross for a variable duration between 900 and 2100ms. As long as the digits were black, no response was needed. After 2–5 black digits, a colored digit would appear, to which a response had to be made depending on whether the digit was odd or even. In the Working Memory task (WM), subjects were also presented with a sequence of 2–5 black digits, except that a colored question mark would appear instead of a colored digit. At that point subjects had to respond whether the last digit they saw was odd or even. Because subjects do not know when the question mark would appear, they had to remember the black digits. Occasionally, instead of the colored digit or question mark, subjects would be presented with a so-callled thought probe, to which they had to respond whether or not they were attending the task, or were thinking about something else. Smallwood et al. found that in the CRT, subjects were thinking about something else 68% of the time, whereas in the WM task they did so in 51% of the cases.

Models of the CRT and WM Task

In order to be able to do the tasks, the Basal Ganglia had to be prewired to carry out primitive actions. Primitive actions consisted of a source slot and a destination slot. For example, V1MEM1 would transfer the contents of the Vision1 slot to the Memory1 slot, and WMMEM1 would transfer the contents of the working memory slot to the Memory1 slot. For efficiency reasons, not all possible combinations were implemented, but a modest superset of the operations needed to do both tasks: V1MEM1, MEM2AC, V1WM, WMMEM1, MEM2WM, WMAC. A second function of the Basal Ganglia is related to learning, and was only active during learning: whenever a primitive action had completed its action, the learning signal would be suppressed. The reason is that we wanted to associate the operation with the state before the operation had been carried out, and did not want an association with the state after the operation (otherwise it would learn to repeat the operation).

A second piece of knowledge the network needs is which numbers are odd and which are even. An winner-takes-all associate memory was implemented in the Memory part of the model. Therefore, if a Semantic Pointer representing a number is placed in Memory1, ODD or EVEN would appear in Memory2.

The input node in the network feeds the input into the Vision slots of the network, and, during the training period, the correct PRIM into the Correct Prim slot. The timing of the model is not yet completely consistent with the real experiment, but compressed in time, and restricted to just two black digits before the colored digit or question mark. Table 1 shows the schedule for what is presented by the input node to both Vision slots, and the correct PRIM operator that needs to be carried out at that point, which is send to the Correct Prim slot to be used in the learning process.

The timing of the experiment is not consistent with human experiment, because many of the processes in Nengo are a lot faster in simulated time, but a lot slower in real time. Visual perception nor actions do take any time in this model, and memory retrieval is extremely fast. On the other hand, simulating a large model like this takes quite some real time, which means that for simulation purposes this is a reasonable compromise.

The general idea in the CRT is that the model does not need to do anything until it sees a red digit. It then should execute V1MEM1 to determine whether the digit it sees is odd or even. After the memory has retrieved ODD or EVEN, it should execute MEM2AC to make the retrieved parity into an action. We are assuming here we have an action system that can interpret this as an action.

The WM model needs to do a bit more work: every time a black digit is presented, it should store that digit in working memory with a V1WM action. Once it sees a colored question mark, it should transfer the item from working memory into the memory retrieval system: WMMEM1. Similarly to the CRT, the result of that retrieval should be transferred to the action slot, MEM2AC.

Whenever the model does not need to do anything, the table shows FOCUS. Although this is indeed represented in the Prim slot, there is no rule in the Basal Ganglia to carry it out (because it doesn't do anything). However, the Basal Ganglia are not just waiting, but has a carries out a "default" action, which will be the basis for Mind Wandering.

Modeling Mind Wandering itself

As has become clear in the previous section, the Basal Ganglia are not always engaged in task-related actions. To model mind wandering, we added a default action to the Basal Ganglia that initiates and perpetuates mind wandering as long as it does not receive a specific instruction from the Prim slot. This option is more or less standard in the Nengo Basal Ganglia model, because you have to specify a default action for it to do if no other action is sufficiently supported.

The idea is, following some existing symbolic models (Taatgen et al., submitted), that Mind Wandering consists of a train of thought simulated by a sequence of declarative retrievals. To mimic this in a simple way, we added a number of extra associations to the memory that also produces the mapping between numbers and parity. More specifically, we added that EPISODE maps onto CRY, CRY maps onto REDEEM, and REDEEM maps onto LAUGH. The default Basal Ganglia action is to feed EPISODE to Memory1, and also copy the contents of Memory2 into Memory1. This means that if there are no active PRIMs (either because it is set to FOCUS, or when there is no specific PRIM active), EPISODE is placed in Memory1, which will in turn lead to the retrieval of CRY, which is fed back into Memory1 leading to the retrieval of REDEEM, etc.

Training

Training consisted of 40 learning blocks, each with a CRT trial and a WM trial. A trial lasted 1.7 simulated seconds following Table 1. After 40 trials the training input was blocked, after which and additional 20 blocks were simulated and used to determine the results.

Results

The critical mapping that the model needs to learn is between the combined state of the system and the PRIM to be executed. Figure 3 shows the input to the Basal Ganglia, which represents the strength of each of the PRIMs in the Prim slot. The graphs shows the average of the 20 performance trials after learning. On the left side of the graph the WM task is shown, where the V1WM prim becomes active whenever there is a black digit. During the short periods between the digits, there is no PRIM that is active enough to exceed the 0.3 threshold, which means that the model will initiate Mind Wandering during this (very brief) period. When the red question mark is presented, the WMMEM1 PRIM is activated, transferring the contents of working memory to a memory retrieval. When the answer has been retrieved from memory, the MEM2AC PRIM is activated to transfer the retrieval to the action slot. The interesting aspect of last action is that the PRIM becomes active earlier than during training (approximately at 1.2 seconds instead of 1.3 seconds), which indicates that the learning has made sure that the PRIM has been keyed to a successful retrieval.

For the CRT we can see that the model does nothing when black digits are presented, even though the V1WM PRIM becomes active, but at a subthreshold level (indicating some transfer from the WM task). When the red digit comes up, the V1MEM1 PRIM becomes active, initiating the memory retrieval and subsequently the MEM2AC PRIM. It is clear that in the CRT the model has much more opportunity to mind wander. This can be seen slightly more clearly in the Thalamus output graph (Figure 4, where a winner-takes-all competition has produced a winning action in each of the stages.

The next question is how much Mind Wandering these decisions actually produce. To get an impression, we need to look at the activity in Memory. Figure 5 shows the activity of various memory items in a sample trial, measured in the Memory2 slot. We can see mind wandering by the activation of the CRY, REDEEM and LAUGH semantic pointers, while task-related activity consists of activation of ODD and EVEN. Obviously, there is a lot more Mind Wandering going on than the Basal Ganglia results suggest. The reason is that after the Basal Ganglia initiates Mind Wandering, it can dominate the activity in the memory system for a while as long as it is not needed by the task (following the threaded cognition

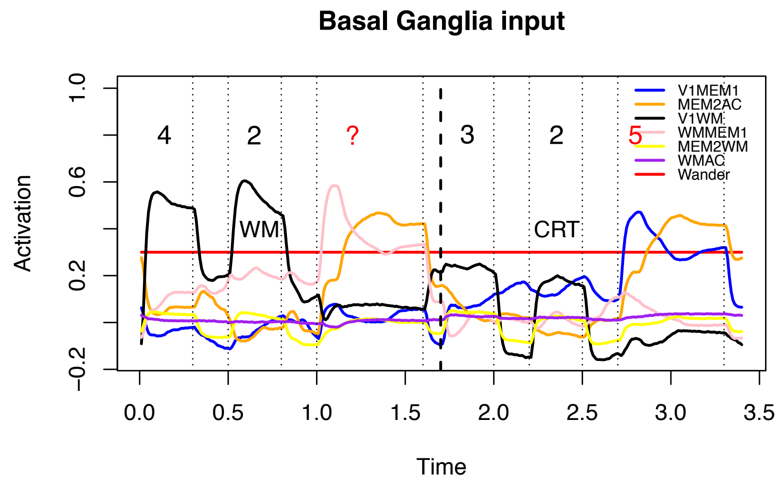

Figure 3: Input to the Basal Ganglia, showing the activation of each of the PRIMs. The WM task is between time 0 and 1.7, the CRT between 1.7 and 2.4. Representative stimuli that are presented to the model are displayed at the top of the Figure. The red horizontal line is the activation of the Wander action: this is not a real activation, but a default action if none of the PRIMs exceeds the 0.3 threshold.

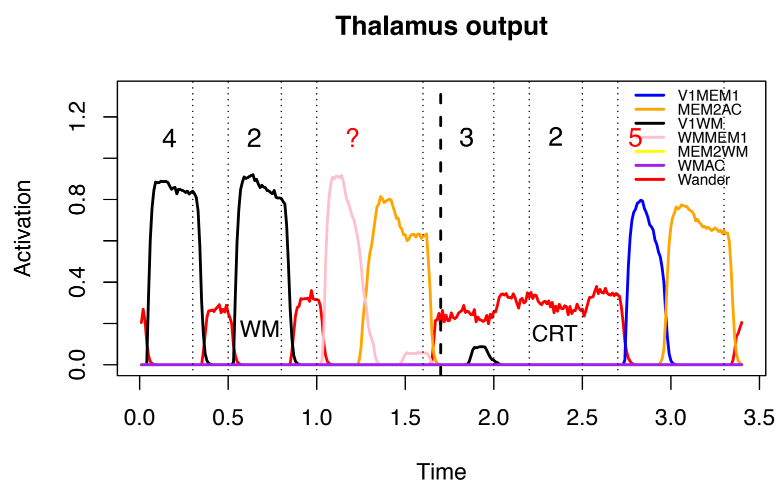

Figure 4: In the output of the Thalamus we can see which action is selected, which is the highest value of the input.

multitasking theory, (Salvucci & Taatgen, 2008)). Nevertheless, in the CRT Mind Wandering is supported by the Basal Ganglia for a much longer period, which is reflected in more memory activity.

If we calculate the proportion of Mind Wandering over all the model output (after training), we see that the Memory output matches the data most closely (Figure 6). We have to take these results with a grain of salt, though, because the timing of the experiment does not match the real experiment.

Discussion

The main purpose of this work was to demonstrate that sequential tasks can be learned by a spiking neural network following principles derived from symbolic architectures. In this model it is no longer necessary to store all procedural knowledge in the Basal Ganglia, but is stored in an associative memory that can be located elsewhere, probably in the prefrontal cortex (Cole, Bagic, Kass, & Schneider, 2010). A key difference with regular production models (and also Spaun), is that it does no test conditions explicitly, but instead learns a mapping between the cognitive system's state and the action to be performed. This has two advantages: sequential matching of production rules in a neural network is cumbersome. In order to do this in parallel, production rules already need to be relatively hard-wired, which makes flexibility a greater challenge. The second advantage is that it is much easier to learn new productions.

Still, there is a lot of work still to be done. The actions this model can make are elementary PRIMs. However, in the full PRIM theory, elementary PRIMs cluster together into general purpose operators. The most probable place for this kind of learning are the Basal Ganglia. Moreover, we used

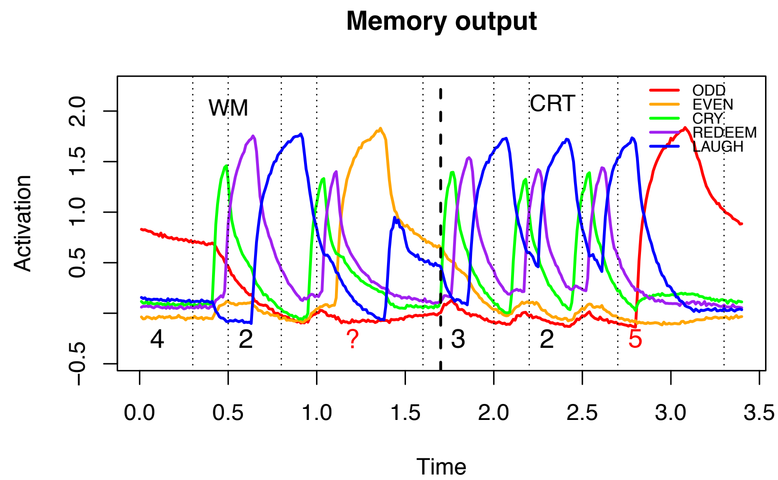

Figure 5: The contents of Memory2 during the experiment, showing which of the facts in Memory is most active.

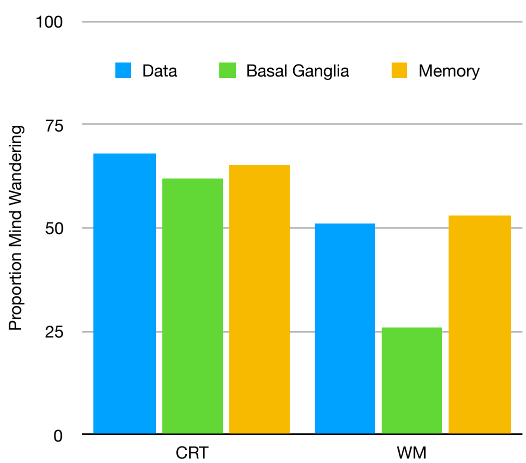

Figure 6: Proportion Mind Wandering in the data, in the model's Basal Ganglia action output, and in the model's Memory output.

supervised learning in this model. It is unclear where such a learning input would come from, and therefore a form of reinforcement learning is a better alternative.

The model's mind wandering is a nice demonstration (also showing the model can fit some data), but the Mind Wandering itself is now modeled as a "default strategy". Instead, it should also be modeled using primitive operations that compete with task-related operators (similar to Taatgen et al, submitted).

Acknowledgments

I wish to thank Terry Stewart, Sean Aubin, Alexander Serb, and Sverrir Thorgeirsson for their help and good discussions for this project.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford university press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological Review*, *111*, 1036–1060.

Cole, M. W., Bagic, A., Kass, R., & Schneider, W. (2010). Prefrontal Dynamics Underlying Rapid Instructed Task Learning Reverse with Practice. *Journal of Neuroscience*, *30*(42), 14245–14254.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., … Rasmussen, D. (2012, nov). A large-scale model of the functioning brain. *Science*, *338*(6111), 1202–5. doi: 10.1126/science.1225266

MacNeil, D., & Eliasmith, C. (2011). Fine-tuning and the stability of recurrent neural networks. *PloS one*, *6*(9), e22885.

O'Reilly, R. C., & Frank, M. J. (2006). Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia. *Neural Computation*, *18*(2), 283–328.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, *115*(1), 101–130.

Smallwood, J., Schooler, J. W., Turk, D. J., Cunningham, S. J., Burns, P., & Macrae, C. N. (2011, dec). Self-reflection and the temporal focus of the wandering mind. *Consciousness and Cognition*, *20*(4), 1120–1126. doi: 10.1016/j.concog.2010.12.017

Stocco, A., Lebiere, C., & Anderson, J. R. (2010). Conditional routing of information to the cortex: A model of the basal ganglia's role in cognitive coordination. *Psychological Review*, *117*(2), 541–574.

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, *120*, 439–471.

Taatgen, N. A., van Vugt, M. K., Daamen, J., Katidioti, I., Huijser, S., & Borst, J. P. (submitted). The resource-availability theory of distraction and mind-wandering.