Neural Networks 79 (2016) 37–52

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Engineering neural systems for high-level problem solving

Jared Sylvester, James Reggia *

Department of Computer Science, University of Maryland, A.V. Williams Building, College Park, MD 20742, United States

ARTICLE INFO

Article history:
Received 16 December 2015
Received in revised form 11 March 2016
Accepted 17 March 2016
Available online 31 March 2016

Keywords:
Neuroengineering
Attractor neural networks
Gated neural networks
Top-down vs. bottom-up AI
Neural network problem solving

ABSTRACT

There is a long-standing, sometimes contentious debate in AI concerning the relative merits of a symbolic, top-down approach vs. a neural, bottom-up approach to engineering intelligent machine behaviors. While neurocomputational methods excel at lower-level cognitive tasks (incremental learning for pattern classification, low-level sensorimotor control, fault tolerance and processing of noisy data, etc.), they are largely non-competitive with top-down symbolic methods for tasks involving high-level cognitive problem solving (goal-directed reasoning, metacognition, planning, etc.). Here we take a step towards addressing this limitation by developing a purely neural framework named GALIS. Our goal in this work is to integrate top-down (non-symbolic) control of a neural network system with more traditional bottom-up neural computations. GALIS is based on attractor networks that can be "programmed" with temporal sequences of hand-crafted instructions that control problem solving by gating the activity retention of, communication between, and learning done by other neural networks. We demonstrate the effectiveness of this approach by showing that it can be applied successfully to solve sequential card matching problems, using both human performance and a top-down symbolic algorithm as experimental controls. Solving this kind of problem makes use of top-down attention control and the binding together of visual features in ways that are easy for symbolic AI systems but not for neural networks to achieve. Our model can not only be instructed on how to solve card matching problems successfully, but its performance also qualitatively (and sometimes quantitatively) matches the performance of both human subjects that we had perform the same task and the top-down symbolic algorithm that we used as an experimental control. We conclude that the core principles underlying the GALIS framework provide a promising approach to engineering purely neurocomputational systems for problem-solving tasks that in people require higher-level cognitive functions.

1. Introduction

Most artificial intelligence (AI) and cognitive modeling systems fall into one of two general groups: systems that take a symbolic, top-down approach, and those that adopt a neural, bottom-up approach. The divide between these two strategies is both long-standing and, at times, quite contentious. This conflict is regrettable because the two different strategies are in many ways complementary rather than competitive: each of the two approaches has its own relative strengths and weaknesses. For example, while neural systems excel at problems that involve pattern matching, incremental learning, low level control, fault tolerance, and/or processing noisy data, they are less adept at handling higher cognitive functions such as goal-directed reasoning, meta-cognition, and planning. Top-down symbolic methods are largely just the opposite. This complementarity has been recognized in the past (Reggia, Monner, & Sylvester, 2014) and leveraged effectively in a number of cognitive architectures (e.g., Sun & Naveh, 2004).

The current limited abilities of neural architectures to capture critical aspects of high-level cognition put them at a tremendous disadvantage relative to symbolic AI techniques when trying to engineer neurocomputational systems for high-level problem-solving tasks. Such problem solving by people depends on *cognitive control*, the process of managing other cognitive processes (Schneider & Chein, 2003). Examples of cognitive control include such *executive functions* as shifting attention, response selection, working memory maintenance, goal setting, and inhibition of irrelevant signals. Executive functions are primarily associated with prefrontal cortex in the primate brain, and substantial recent work has focused on localizing these functions to specific individual prefrontal regions (Burgess, Dumontheil, & Gilbert, 2007; Koechlin & Summerfield, 2007).

* Corresponding author.
E-mail addresses: jared@jsylvest.com (J. Sylvester), reggia@cs.umd.edu (J. Reggia).

http://dx.doi.org/10.1016/j.neunet.2016.03.006
0893-6080/

The limited ability of neurocomputational methods to support higher-level cognitive/executive functions is somewhat surprising in that the human brain handles such issues routinely, thereby establishing that neural computations have the capacity to do so. In the work that we describe here, while we take inspiration from human cognition and neuroscience, we are not trying to create accurate models of either. Instead our primary focus is on how to construct/engineer neural network systems for problem-solving tasks that are competitive with top-down symbolic AI problem-solving systems. Developing purely neurocomputational systems for high-level problem solving could ultimately provide several significant practical advantages. For example, when compared to traditional top-down AI, neural computation is fault tolerant, and it has the potential for great speed due to its inherent parallel processing (Haykin, 2009; Reggia et al., 2014). The latter is particularly true given the recent increasing availability of parallel computing hardware such as neural network chips and GPU clusters. Further, neurocomputational methods have the ability to learn and adapt, something that will be increasingly important in future AI systems of all kinds.

Studying neurocognitive architectures involving cognitive control is currently viewed as an important research direction (Roy, 2008), and the importance of developing neural computational methods for cognitive control is likely to substantially increase over the next decade as work on developing large-scale brain and neurocognitive models accelerates. By *large-scale models* we mean recent and ongoing research efforts to create neuroanatomically grounded simulations of all or major portions of human/mammalian brain structure and function, or at least major subsystems of the brain that span multiple cortical regions. These models vary from extremely large networks of biologically-realistic spiking neurons to those that are more abstract, based on a higher level of components such as cortical columns, or are focused on simultaneously supporting human cognitive functions (e.g., de Garis, Shuo, Goertzel, & Ruiting, 2010; Eliasmith et al., 2012; Townsend, Keedwell, & Galton, 2014; Weems & Reggia, 2006; Winder, Cortez, Reggia, & Tagamets, 2007). They are often inspired by the view that the brain is organized as a network of regions that are inter-connected via well recognized pathways. Specifically, the primate cerebral cortex is organized as a distributed network of interacting cortical regions, exhibiting both functional integration and functional segregation (Bressler & Menon, 2010; Sporns, 2011; van Essen, Anderson, & Felleman, 1992). *Large-scale region-and-pathway models* inspired by this viewpoint consist of components (modules) that are neural network simulations of individual brain regions, e.g., Wernicke's and Broca's areas and the arcuate fasciculus that connects them (Monner & Reggia, 2013; Weems & Reggia, 2006). Work on large-scale neurocognitive models is increasing in part due to recent major funding initiatives (Europe's Human Brain Project, US BRAIN Initiative, etc. (Abbott, 2013)).

A fundamental question arises in constructing large-scale neurocognitive architectures like these: Is there an identifiable minimal set of generic, region-level functions and interactions that can be used to construct neural architectures that provide cognitive control of human-like problem-solving behaviors? As one possible approach to answering this question, we have recently proposed the GALIS framework (Sylvester, Reggia, Weems, & Bunting, 2013), where GALIS is an acronym for "Gated Attractors Learning Instruction Sequences". The methods used in GALIS are intended to provide a general purpose framework within which models for specific higher-level problem solving tasks can be instantiated and trained using solely subsymbolic methods. While it takes inspiration from the human brain and cognition, GALIS is *not* intended to be a veridical model of either the brain or human reasoning. The central issue that GALIS addresses concerns how one can adopt and extend purely neurocomputational methods to engineer high-level cognitive control of the sort that can currently only be readily modeled using top-down symbolic approaches. GALIS assumes that one is interested in constructing a large-scale region-and-pathway model of some aspect of human-level cognition that is inspired by the organization of the cerebral cortex, and perhaps other subcortical brain regions. An implication of this assumption is that model brain regions must learn not only the facts about a specific instance of a task, but also the procedure or "instruction sequence" that is needed to perform that task in general. This focus on making problem solving dependent on patterns stored in the network's memory, rather than on the network's structure or "hardware", differs from many previous models of cognitive control, and is intended to make GALIS models more generalizable: Their behavior can be changed by adjusting which behavioral sequences are learned rather than by adjusting the structure of the model itself.

GALIS answers the fundamental question above by adopting two principles. First, GALIS assumes that each region in the cortical network can be conceptualized as an attractor neural network — a dynamical complex system whose activity is driven towards certain preferred states. Attractor networks have been used previously in cognitive control models (Farrell & Lewandowsky, 2002; Hoshino, Usuba, Kashimori, & Kambara, 1997; Jones & Polk, 2002), but usually operate only with fixed-point attractors. In contrast, GALIS' attractors are designed to enable switching between attractor states in ordered sequences. This is critical if procedural information of the sort readily handled by top-down symbolic AI methods is to be accommodated in memory: procedures by their very nature must have temporal extents and their component steps must be performed in a specific order (Ismail & Shapiro, 2000). While multiple techniques have been used to add similar dynamism to attractor nets (Brown, Preece, & Hulme, 2000; Horn & Opher, 1996; Winder, Reggia, Weems, & Bunting, 2009), GALIS uses an approach to learning of temporally asymmetric connection weights that we recently developed (Sylvester, Reggia, & Weems, 2011; Sylvester, Reggia, Weems, & Bunting, 2010a).

Second, GALIS assumes that each cortical region cannot only exchange information with other cortical regions in the form of activity patterns, but can also gate other regions' functions and interactions. By *gating* here we mean that one cortical region can modulate the functions of other regions, or open/close the flow of information between other regions. The inspiration for adopting gating as a central aspect of neural problem-solving systems comes from past neuroscience research and neurobiologically-realistic computational models suggesting that gating is an important aspect of brain dynamics (Frank, Loughry, & O'Reilly, 2001; O'Reilly & Frank, 2006; Sherman & Guillery, 2006; Singer, 2011; Womelsdorf & Fries, 2009). While there have been numerous theories posited for the biological structures that could directly or indirectly underly such cortical gating, GALIS is agnostic about the particular physiological implementation. Rather, we take the existence of some such mechanism as given and implement gating as direct interactions between model cortical regions and their connecting pathways.

In summary, the basic hypothesis being explored via GALIS is that large-scale region-and-pathway models based on (1) representing procedures/programs as temporal sequences of attractor states, and (2) allowing model regions to gate the behavior of other model regions, provide a sufficient purely-neurocomputational framework for engineering autonomous problem-solving systems that can be competitive with the more traditional top-down symbolic problem-solving systems that are used in AI. While our previous work with GALIS was encouraging in showing that it could successfully support models for simple working memory applications such as the *n*-Back task used in psychological testing (Sylvester et al., 2011, 2013), such applications were too simple to seriously support this hypothesis; they had no need to generate