OpenCog NS: A Deeply-Interactive Hybrid Neural-Symbolic Cognitive Architecture Designed for Global/Local Memory Synergy

Ben Goertzel

Novamente LLC
1405 Bernerd Place, Rockville MD 20851

Abstract

A deeply-interactive hybrid neural-symbolic cognitive architecture is defined as one in which the neural-net and symbolic components interact frequently and dynamically, so that each intervenes significantly in the other's internal operations, and the two form a combined dynamical system at the time-scale of each component's individual cognitive operations. An example architecture of this nature that is currently under development is described: OpenCog NS, based on integration of the OpenCog cognitive architecture (which incorporates symbolic, evolutionary and connectionist aspects) with a hierarchical attractor neural network (HANN). In this integrated architecture, the neural and non-neural aspects each play major roles, and the depth of the interconnection is revealed for example in the facts that symbolic reasoning intervenes in the process of attractor formation within the HANN, whereas the HANN plays a major role in guiding the individual steps of logical inference and evolutionary program learning processes.

Introduction

Neural net and symbolic AI systems tend to have different strengths and weaknesses. We will argue below that the distinction betwen neural and symbolic systems is a vague one; but even accounting for this, the differences in strengths and weaknesses remain. Neural nets tend to be especially good at recognition of patterns in high-dimensional quantitative data, among other things. Symbolic systems tend to be good at abstract reasoning and syntax processing, among other things. Both kinds of systems can be good at generalization and analogy, and at credit assignment, but in different contexts and in different ways. Hardware-wise, neural nets make better use of GPUs, but symbolic systems better exploit the non-brain-like precision of digital computer processors.

These differences have spawned a host of "neural-symbolic" AI systems combining aspects of both paradigms. The various neural-symbolic systems described in the literature (Garcez et al, 2008; Hammer and Hitzler, 2007) are quite diverse, and don't have that much in common aside from the fact of encompassing both neural and symbolic aspects, and the motivation of "getting the best of both worlds."

I will describe here a novel approach to neural-symbolic integration called *deeply-interactive hybrid neural-symbolic cognitive architecture*. It's *hybrid* because it involves two separate components, one neural and one symbolic, with neither in a primary role; it's *deeply-interactive* because the two components are tightly dynamically bound with each other in their real-time internal operations. We will argue that this sort of architecture is well-suited to achieve an important property called "global/local synergy," which means roughly that each key type of memory (sensory, declarative, procedural, episodic) has both global-memory and localized-memory oriented sub-stores attached to it, and the cognitive processes associated with these sub-stores interact in a synergetic manner.

Finally, we will describe in detail a particular example architecture designed according to this approach: OCNS (OpenCog Neural-Symbolic), a hybrid of the existing OpenCog Prime architecture (which combines symbolic, evolutionary and connectionist aspects) with a hierarchical attractor neural net.

The hypothesis underlying this work is that deeply interactive hybrid neural-symbolic architecture may be the best way to leverage existing computer hardware and algorithms toward advanced artificial general intelligence.

Globalist versus Localist AI Systems

The distinction between "connectionist"/"neural" and "symbolic" AI systems, which seemed relatively clear in the 1970s and 80s, has become dramatically fuzzier as AI technology and theory have advanced.

On the one hand, some semantic network and production rule systems contain spreading activation and represent some knowledge emergently rather than locally, giving them many of the key characteristics of connectionist systems. Inspired by this, the ACT-R cognitive architecture, originally a symbolic production system, was experimentally reimplemented using a connectionist architecture (Lebiere and Anderson, 1993). And when one connects an uncertain logic based system directly to sensors and actuators, rather than using formally-encoded knowledge, the sense in which the resulting system is more "symbolic" than a neural network becomes subtle and unclear (Goertzel, et al, 2006).

On the other hand, some neural net systems have complex formal neurons (Aizenberg and Morega, 2006) and/or highly structured dynamics (Pollack, 1991), which bring them fairly far from the sphereof brain-modeling and into the domain of carefully engineered, special-purpose AI systems.

Nevertheless there are still some clear and meaningful distinctions to be made, between many of the AI systems typically classified as "symbolic", and many of the AI systems typically classified as "connectionist" or neural." Perhaps the most critical distinction is between systems where memory is essentially global, and those where memory is essentially local.

This distinction is most easily conceptualized by reference to memories corresponding to categories of entities or events in an external environment. In an AI system that has an internal notion of "activation" – i.e. in which some of its internal elements are more active than others, at any given point in time – one can define the internal image of an external event or entity as the fuzzy set of internal elements that tend to be active when that event or entity is presented to the system's sensors. If one has a particular set S of external entities or events of interest, then, the degree of memory localization of such an AI system relative to S may be conceived as the percentage of the system's internal elements that have a high degree of membership in the internal image of an average element of S.

In this sense, a Hopfield neural net (Amit, 1992) would be considered "globalist" since it has a low degree of memory localization (most internal images heavily involve a large number of system elements), whereas a typical logic-based knowledge store would be considered "localist" as it has a very high degree of memory localization (most internal images are heavily focused on a small set of system elements).

Of course, this characterization of localization has its limitations, such as the possibility of ambiguity regarding what are the "system elements" of a given AI system; and the exclusive focus on internal images of external phenomena rather than representation of internal abstract concepts. However, our goal here is not to formulate an ultimate, rigorous and thorough ontology of memory systems, but only to pose a "rough and ready" categorization so as to properly frame our discussion of certain types of hybrid neural-symbolic systems.

It might be conceptually better-founded to discuss "globalist-localist" AI systems (defined as systems containing separate but interacting globalist and localist components) than "neural-symbolic" AI systems. For it is in principle quite possible to create localist systems using formal neurons, and also to create globalist systems using

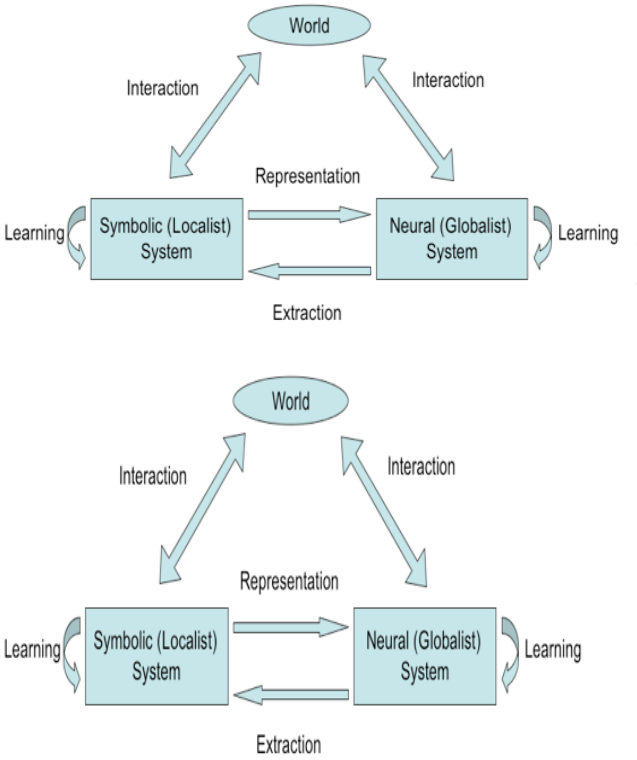

Figure 1. Generic neural-symbolic architecture

Bader and Hitzler categorize neural-symbolic systems according to three orthogonal axes: interrelation, language and usage. "Language" refers to the type of language used in the symbolic component, which may be logical, automata-based, formal grammar-based, etc. "Usage" refers to the purpose to which the neural-symbolic interrelation is put. In our Figure 1 we use "learning" as an encompassing term for all forms of ongoing knowledge-creation, whereas Bader and Hitzler distinguish learning from reasoning.

Of Bader and Hitzler's three axes the one that interests us most here is "interrelation", which refers to the way the neural and symbolic components of the architecture intersect with each other. They distinguish "hybrid" architectures which contain separate but equal, interacting neural and symbolic components; versus "integrative" architectures in which the symbolic component essentially rides piggyback on the neural component, extracting information from it and helping it carry out its learning, but

playing a clearly derived and secondary role. We prefer Sun's (2001) term "monolithic" to Bader and Hitzler's "integrative" to describe this type of system, for reasons I will shortly make clear: some hybrid neural-symbolic systems can be intensely "integrative" in the common usage of this term.

Within the scope of hybrid neural-symbolic systems, there is another axis which Bader and Hitzler do not focus on, because the main interest of their review is in monolithic systems. We call this axis "interactivity," and what we are referring to is the frequency of high-information-content, high-influence interaction between the neural and symbolic components in the hybrid system. In a low-interaction hybrid system, the neural and symbolic components don't exchange large amounts of mutually influential information all that frequently, and basically act like independent system components that do their learning/reasoning/thinking periodically send each other their conclusions. In some cases, interaction may be asymmetric: one component may frequently send a lot of influential information to the other, but vice versa. However, the systems that interest us most here are symmetrically highly interactive ones.

In a symmetric high-interaction hybrid neural-symbolic system, the neural and symbolic components exchange influential information sufficiently frequently that each one plays a major role in the other one's learning/reasoning/thinking processes. Thus, the learning processes of each component must be considered as part of the overall dynamic of the hybrid system. The two components aren't just feeding their outputs to each other as inputs, they're mutually guiding each others' internal processing.

One can make a speculative argument for the relevance of this kind of architecture to neuroscience. It seems plausible that this kind of neural-symbolic system roughly emulates the kind of interaction that exists between the brain's neural subsystems implementing localist symbolic processing, and the brain's neural subsystems implementing globalist, classically "connectionist" processing. It seems most likely that, in the brain, localist symbolic functionality emerges from an underlying layer of globalist neural dynamics. However, it is also reasonable to conjecture that this localist symbolic functionality is confined to a functionally distinct subsystem of the brain, which then interacts with other subsystems in the brain much in the manner that the symbolic and neural components of a symmetric high-interaction neural-symbolic system interact.

Neuroscience speculations aside, however, our key conjecture is that this sort of neural-symbolic system presents a promising direction for artificial general intelligence research. In later sections I will give a more concrete idea of what a symmetric high-interaction hybrid neural-symbolic architecture might look like, exploring the potential for this sort of hybridization between the OpenCogPrime AGI architecture (which is heavily symbolic in nature) and hierarchical attractor neural nets.

Multiple Memory Types and Cognitive Synergy

The OpenCogPrime (OCP) architecture, and OCNS which builds on it, are founded on the distinction between multiple types of memory: the declarative, procedural, sensory, and episodic memory types that are widely discussed in cognitive neuroscience (Tulving and Craik, 2005), plus attentional memory for allocating system resources generically, and intentional memory for allocating system resources in a goal-directed way.

One of the core principles underlying OCP is "cognitive synergy" (Goertzel, 2009) which states that an intelligent system should have different cognitive processes corresponding to each type of memory, and that these processes should interact synergetically, so that when one of them gets stuck, it can appeal to the others for help.

One may extend this notion of cognitive synergy into a notion of "global/local synergy" – referring to systems that contain both globalist and localist memory sub-stores corresponding to each memory type; and have cognitive pocesses corresponding to each of these memory sub-stores, which interact synergetically. A natural hypothesis is that symmetric high-interaction neural-symbolic systems are a promising route to global/local synergy.

The globalist/localist dichotomy should not be overstated: OCP, for example, contains both globalist and localist aspects to its memory and learning. However, it is is clearly more localist than an attractor neural net.

OpenCog Prime: An Integrative Cognitive Architecture for General Intelligence

The OCP architecture has been summarized in a recent conference paper (Goertzel, 2009a) and we will not repeat that summary here, but will only note the key data structures and cognitive algorithms of the system, as correlated with the memory types listed above:

Memory Type	OpenCogPrime data structure OpenCogPrime cognitive process
Declarative	The AtomTable: a special form of weighted, labeled hypergraph -- i.e. a table of nodes and links (collectively Atoms) with different types, and each weighted with a multi-dimensional truth value
	Probabilistic Logic Networks for uncertain inference; concept creation heuristics
Attentional	Atoms in the AtomTable are weighted with AttentionValue objects, which contain both ShortTermImportance (STI) values (governing processor time allocation) and LongTerm Importance (LTI) values (governing memory usage).

	Economic Attention Allocation (ECAN) propagates and updates AttentionValues based on system goals and Hebbian learning.
Procedural	"Combo" tree structures embodying LISP-like programs, in a special program dialect intended to manage external and internal actions
	MOSES (probabilistic evolutionary learning) and hillclimbing.
Sensory	A collection of specialized sense-modality-specific data structures
	Pattern mining heuristics
Episodic	An internal simulation world that allows the system to run "mind's eye movies" of situations it remembers, has heard about, or hypothetically envisions.
	Heuristics for launching simulations based on declarative knowledge; MOSES and pattern mining for extracting declarative patterns from simulations
Intentional	Goals are represented by Atoms stored in the AtomTable; there is a separate table indicating which Atoms are top-level goals
	PLN for goal refinement and abstraction; ECAN for directing actions and resources based on goals

Table 1. The OpenCogPrime data structures used to represent and process the key memory types

Hierarchical Attractor Neural Networks

OCP could be integrated with a variety of different neural network (NN) architectures. For the purpose of this paper, I will articulate a class of neural nets that is neither completely general nor extremely specific. Many of the ideas to be presented here are in fact more broadly applicable beyond the NN architecture described here.

The following assumptions will be made about the HANN (Hierarchical Attractor Neural Network) to be hybridized with OCP:

- It consists of a network of neurons, endowed with an activation spreading and learning algorithm, whose connectivity pattern is largely but not entirely hierarchical (and whose hierarchy contains both feedback, feedforward and lateral connections)
- It contains a set of input neurons, receiving perceptual inputs, at the bottom of the hierarchy
- It has a set of output neurons, which may span multiple levels of the hierarchy. The "output neurons" indicate control signals to actuators, which may be internal or external.
- Other neurons besides I/O neurons may potentially be observed or influenced by external processes; for instance they may receive stimulation
- Link weights in the HANN get updated via some learning algorithm that is "statistically Hebbian," in the sense that on the whole when a set of neurons get activated together for a period of time, they will tend to become attractors. By an attractor I mean: a set S of neurons such that the activation of a subset of S during a brief interval tends to lead to the activation of the whole set S during a reasonably brief interval to follow
- As an approximate but not necessarily strict rule, neurons higher in the hierarchy tend to be involved in attractors corresponding to events or objects localized in larger spacetime regions

Examples of specific neural net architectures satisfying these requirements are the visual pattern recognition networks constructed by Hawkins (2004), Granger (2006), and Arel et al (2009). The latter appears to fit the requirements most snugly due to having dynamics better suited to the formation of a complex array of attractors, and a richer methodology for producing outputs.

Incorporating HANNs Into OpenCog Prime via a Deeply-Interactive Hybrid Architecture

The essential nature of the OCP/HANN integration suggested here, labeled OCNS (OpenCog Neural-Symbolic), is conveyed in Figure 2. It involves crosslinking and dynamically coupling the two systems fairly tightly so as to form a composite nonlinear dynamical intelligent system.

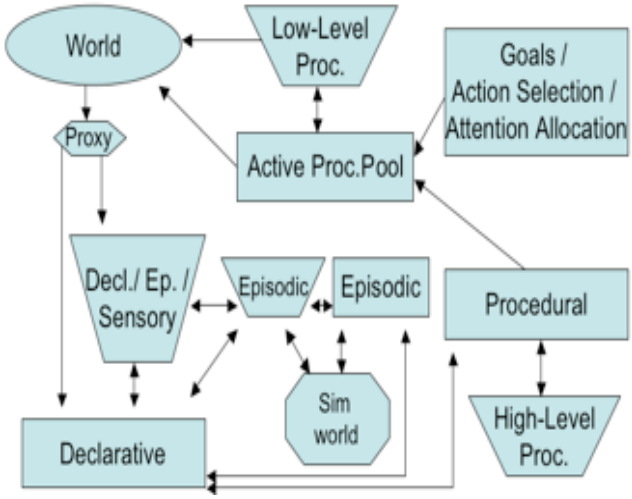

Figure 2. OpenCog Neural-Symbolic architecture. Rectangles are OCP memory/processing components; trapezoids are HANNs. All 4 HANNs are assumed interlinked but these links are not shown. The "Goals..." box links to all boxes (for spread of attention) and these links are not shown.

OCNS involves four separate HANNs corresponding to different memory types:

1. a "primary" HANN that handles sensory and declarative and some episodic knowledge, connected via MemberLinks with OpenCog's AtomSpace

2. a “high-level procedural” HANN used for procedure learning, that connects with MOSES
3. a “low-level procedural” HANN embodying low-level procedures invoked by Combo trees
4. an “episodic” HANN used for episodic learning, that connects with a DB of episodes and an internal simulation world

The activation-spreading and learning dynamic may be provisionally assumed the same for each HANN. The crux of OCNS architecture is the way these HANNs are interrelated with existing OCP cognitive processes in pursuit of global/local synergy.

The intended operation of OCNS is best explained via enumeration of memory types and control operations.

Declarative Memory

The key novel declarative knowledge mechanism in OCNS is the linkage of HANN attractors to OCP ConceptNodes via MemberLinks. This is in accordance with the notion of glocal memory (Goertzel et al, 2009), in the language of which the HANN attractors are the maps and the corresponding ConceptNodes are the keys. Put simply, when a HANN attractor is recognized, MemberLinks are created between the NN nodes comprising the main body of the attractor, and a ConceptNode in the AtomTable representing the attractor. MemberLink weights may be used to denote fuzzy attractor membership. Activation may spread from NN nodes to ConceptNodes, and STI may spread from ConceptNodes to NN nodes; a conversion rate between NN activation and STI currency must be maintained by the OCP central bank, for ECAN purposes.

Sensory Memory

OCNS uses the primary HANN to store memories of sense-perceptions and low-level abstractions therefrom. MemberLinks may join concepts in the AtomTable to percept-attractors in the NN. If the primary HANN is engineered to associate specific neural modules to specific spatial regions or specific temporal intervals, then this may be accounted for by automatically indexing ConceptNodes corresponding to attractors centered in those modules in the AtomTable’s TimeServer and SpaceServer objects, which index Atoms according to time and space.

Procedural Memory

The role of the low-level procedural HANN is to learn procedures such as low-level motion primitives that are more easily learned using NN training than using more abstract procedure learning methods. For example, a Combo tree learned by MOSES in OCP might contain a primitive corresponding to the predicate-argument relationship *pick_up(ball)*; but the actual procedure for controlling a robot hand to pick up a ball, might be expressed as an activity pattern within the low-level procedural HANN. A procedure P stored in the low-level procedural HANN would be represented in the AtomTable as a ConceptNode C linked to key nodes in the HANN attractor corresponding to P. The invocation of P would be accomplished by transferring STI currency to C and then allowing ECAN to do its work.

On the other hand, OCNS’s interfacing of the high-level procedural HANN with the OCP ProcedureRepository is intimately dependent on the particulars of the MOSES procedure learning algorithm. MOSES is a complex, multi-stage process that tries to find a program maximizing some specified fitness function, and that involves doing the following within each "deme" (a deme being an island of roughly-similar programs)

1. casting program trees into a hierarchical normal form
2. evaluating the program trees on a fitness function
3. building a model distinguishing fit versus unfit program trees, which involves: 3a. figuring out what program tree features the model should include; 3b. building the model using a learning algorithm
4. generating new program trees that are inferred likely to give high fitness, based on the model
5. return to step 1 with these new program trees

There is also a system for managing the creation and deletion of demes.

The weakest point in the current MOSES implementation appears to be step 3. And the main weakness is conceptual rather than algorithmic; what is needed is to replace the current step 3 (which is based on Pelikan’s (2005) hBOA) with something that uses long-term memory to do model-building and feature-selection, rather than (like the current code) doing these things in a manner that's restricted to the population of program trees being evolved to optimize a particular fitness function.

In OCNS we propose to resolve this issue via replacing step 3b (and, to a limited extent, 3a) with an interconnection between MOSES and the procedural HANN. A HANN can do supervised categorization, and can be designed to handle feature selection in a manner integrated with categorization, and also to integrate long-term memory into its categorization decisions.

Episodic Memory

OCNS handles episodic knowledge via a combination of:

- using some traditional computing based approach to store a large database of actual experienced episodes [including sensory inputs and actions; and also the states of the most important items in memory during the experience]
- training a large HANN to summarize the scope of experienced episodes.

Such a network should be capable of generating imagined episodes based on cues, as well recalling real episodes. The episodic HANN would serve as a sort of index into the memory of episodes, There would be HebbianLinks from the AtomTable into the episodic HANN.

Action Selection and Attention Allocation

OCNS chooses actions using OCP's action selection mechanism, which selects procedures based on which ones are estimated most likely to achieve current goals given current context, and places these in an "active procedure pool" where an ExecutionManager object mediates their execution.

Attention allocation spans the two components of OCNS. Attention flows between the two components due to the conversion of STI to and from NN activation. Furthermore, Hebbian learning is a cross-component dynamic. This is where the assumption that the HANN obeys "statistical Hebbian learning" comes in. Links in the HANN may be reinforced via Hebbian learning, via having the nodes they join simultaneously activated.

In this manner assignment of credit flows from GoalNodes into the HANN, because this kind of simultaneous activation may be viewed as "rewarding" a NN link. So, the HANN may reward signals from GoalNodes via ECAN, because when a ConceptNode gets rewarded, if the ConceptNode points to a set of neurons, these neurons get some of the reward.

Conclusion

I have identified a previously uncharacterized category of AI architecture: symmetric high-interaction hybrid neural-symbolic systems. wehave then given a detailed description of one possible architecture falling into this category: OpenCog Neural-Symbolic (OCNS), a hybrid of the existing OpenCogPrime architecture with a hierarchical attractor neural net. As OCNS has not yet been implemented its properties are obviously somewhat speculative; even as an architecture specification, however, it does serve to exemplify the sort of things that can be built if one wishes to explore hybrid neural-symbolic systems in which the two components interact very tightly. We believe this is a fundamentally different category of system than the neural-symbolic systems that are more commonly explored, with some potentially very desirable properties, including global/local memory synergy.

References

Aizenberg, Igor and Claudio Morega (2006). Multilayer Feedforward Neural Network Based on Multi-valued Neurons (MLMVN) and a Backpropagation Learning Algorithm, Soft Computing 11-2

Amit, David (1992). Modeling Brain Function. Cambridge UniversityPress.

Arel, Itamar, Derek Rose, and Robert Coop (2009). A Biologically Inspired Deep Learning Architecture with Application to High-Dimensional Pattern Recognition. BICA-2009

Bader, Sebastian and Pascal Hitzler (2005). Dimensions of Neural-symbolic Integration - A Structured Survey. In: S. Artemov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb and J. Woods (eds). We Will Show Them: Essays in Honour of Dov Gabbay, Volume 1. International Federation for Computational Logic, College Publications, 2005, pp. 167-194.

Garcez, Artur S. d'Avila , Luis C. Lamb and Dov M. Gabbay (2008). Neural-Symbolic Cognitive Reasoning. Cognitive Technologies. Springer

Goertzel, Ben (2009). OpenCog Prime: A Cognitive Synergy Based Architecture for Virtually Embodied Artificial General Intelligence. ICCI-2009

Goertzel, Ben (2009). Cognitive Synergy: A General Principle for Feasible Artificial General Intelligence. ICCI-2009

Goertzel, Ben, Joel Pitt, Rui Liu and Cassio Pennachin (2009). Glocal Memory. Submitted for publication.

Granger R (2006) Engines of the Brain: The computational instruction set of human cognition. AI Magazine 27:15-32

Hammer, Barbara and Pascal Hitzler (Eds) (2007)., Perspectives of Neural-Symbolic Integration. Studies in Computational Intelligence, Vol. 77. Springer

Hawkins, Jeff and Sandra Blakeslee (2004). On Intelligence. Times Books.

Heljakka, Ari, Ben Goertzel, Welter Silva, Izabela Goertzel and Cassio Pennachin (2006). Reinforcement Learning of Simple Behaviors in a Simulation World Using Probabilistic Logic, in Goertzel, Ben and Pei Wang (Eds.), Advances in Artificial General Intelligence, IOS Press

Lebiere, C. & Anderson, J. R. (1993). A Connectionist Implementation of the ACT-R Production System. In Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society, pp. 635-640.

Pollack, J.B. (1991) The Induction of Dynamical Recognizers. Machine Learning, 7, 227-252.

Pelikan, Martin (2005). Hierarchical Bayesian Optimization Algorithm. Springer.

Sun, Ron (2001). Hybrid systems and connectionist implementationalism. In Encyclopedia of Cognitive Science. MacMillan Publishing Company

Tulving, Endel and Craik (2005). The Oxford Handbook of Memory. Oxford University Press